

Mindful Design

How and Why to Make Design
Decisions for the Good of Those
Using Your Product

—
Scott Riley

Apress®

Mindful Design

How and Why to Make Design
Decisions for the Good of Those
Using Your Product

Scott Riley

Apress®

Mindful Design

Scott Riley
Nottingham, UK

ISBN-13 (pbk): 978-1-4842-4233-9
<https://doi.org/10.1007/978-1-4842-4234-6>

ISBN-13 (electronic): 978-1-4842-4234-6

Library of Congress Control Number: 2018963593

Copyright © 2019 by Scott Riley

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr

Acquisitions Editor: Louise Corrigan

Development Editor: James Markham

Coordinating Editor: Nancy Chen

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit <http://www.apress.com/rights-permissions>.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484242339. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

To The Mum.

Table of Contents

About the Author	xi
About the Technical Reviewer	xiii
Acknowledgments	xv
Introduction	xvii
Chapter 1: Attention and Distraction	1
The Attentional System	1
Sensory Distraction	2
Emotional Distraction	3
Task-positive Mode	4
The Default Mode Network.....	6
The Cost of Distractions	9
Decision-making	9
Inclusivity and Equality.....	10
Habit-changing	13
Attention-grabbing	13
Attentional Economics	16
Taking the Inside Out	17
The Cognitive Miser	19
Categories and Cognitive Economy.....	21
Applying Attention to Design.....	27
Summary.....	28

TABLE OF CONTENTS

- Chapter 2: Vision, Perception, and Aesthetics 31**
 - The Iconic Abstraction Scale..... 34
 - Illustration and Photography 36
 - The Gestalt Principles of Perception 39
 - Proximity 40
 - Similarity 42
 - Continuity 44
 - Common Fate 46
 - Closure 47
 - Common Region 48
 - Putting It All Together 49
 - The Self-fulfilling Myth of Color Psychology 50
 - The von Restorff Effect..... 51
 - Codifying Color 52
 - Animation..... 57
 - Limitations of Contrast..... 63
 - Convention, Signifiers, and Metaphor 66
 - In Search of the Soul of Design..... 73
 - Genuine Aesthetics 74
 - Slow Down and Make Things..... 75
 - Summary..... 76

- Chapter 3: Learning and Memory 79**
 - A Note on Goals..... 80
 - Learnable vs. Educational Applications 82
 - How We Learn 83
 - Designing a Learnable Environment 85
 - The Storage and Retrieval of Files or Data..... 86
 - The Creation of Media 88
 - Sharpening Our Mental Models..... 91
 - Levels of Processing..... 97
 - Where and When We Teach 98

Steps to Learnability.....	110
Putting It All Together	117
Summary.....	118
Chapter 4: Expectation, Surprise, and the Musical Interface	121
Fourth Dimensional Design.....	122
Melody, Harmony, and Rhythm.....	124
Melody.....	124
Harmony	130
Rhythm	131
Tension, Anticipation, and Surprise.....	133
Schemas.....	133
Tension as Schematic Violation.....	138
The Spectrum of Surprise	142
Surprise: the Remedy to Habituation.....	144
Happy Complexities	144
The Musical Interface	148
Summary.....	156
Chapter 5: Reward and Motivation	159
Common Mistakes of the Reward Slingers.....	161
Trying to Manufacture Reward	161
Overreliance on Carrot and Stick.....	161
Forgetting about Emotion	162
Mandating Emotional Impact.....	162
The Origin of Wanting.....	163
Dopamine.....	164
What the Heck Is It?	165
What the Heck Does It Do?	165
Reward Seeking.....	166
The Pocket Slot Machine	168
Reward in Games	171

TABLE OF CONTENTS

- The In-Game Slot Machine 171
- Intrinsic Rewards..... 173
- Rewards as Feedback 177
- Humanistic Motivation 184
 - The Self-Determination Theory 185
- The Ethics of Rewards 187
- Summary..... 188
- Chapter 6: Designing Environments 191**
- What Is UX?..... 191
 - Design Intangibles..... 193
 - The Folly of UX: Control and Linearity..... 196
- What Is a Digital Environment? 198
- Balancing Linearity and Nonlinearity 198
 - Linearity..... 199
 - Designing Linearity..... 200
 - Nonlinearity 205
 - Balancing the Two 211
- Design Your Moments 212
- Design as a Function of State 214
- Summary..... 220
- Chapter 7: The Design Stack 221**
- Research and Planning 222
 - Customer Research 223
 - Observation Research 224
 - Competitor Research..... 232
 - Customer Service Is Your Friend 233
 - User Personas 233
 - The Important Questions 238
 - The Common Deliverables..... 238

Scribbling and Sketching	238
Design Briefs and Feature Lists.....	240
Success Metrics and Goals	244
Wireframes	245
The Important Questions	251
The Common Deliverables.....	251
Testing and Iterating	252
User Testing.....	252
Iterating after Testing	256
The Important Questions	257
The Common Deliverables.....	257
Summary.....	258
Chapter 8: Emotion Mapping	261
Thinking in Systems and Moments.....	263
System Thinking	263
Documenting Systems	265
State Diagrams.....	265
Mental Models.....	271
Empathy Maps.....	279
Putting It All Together	283
Chapter 9: The Anatomy of an Interaction.....	285
Thinking in Moments	286
Finding Your Moments	288
Moments Tell a Story	289
Moments Make Us Money	289
Moments Constrain Our Interface.....	289
Documenting Your Moments	293
Moments Rely on Intent.....	293
Moments Are (Mostly) Stand-alone	294
Moments Change State	294

TABLE OF CONTENTS

- Moment Stories..... 295
 - A Moment’s Beginning..... 296
 - Points of Friction 297
 - The Moment of Truth 298
 - A Reflective Ending..... 298
- Storyboarding Moments..... 301
 - Pick a Moment and an Empathy Map..... 305
 - Plot Out the Story Flow..... 305
 - Tell the Story..... 306
 - End the Story Properly..... 307
 - Annotate 309
 - Moment Mapping 310
- The Anatomy of an Interaction 313
 - Framing 315
 - Acknowledgment..... 316
 - Waiting 319
 - Respond..... 322
- Zooming In and Out..... 325
- Chapter 10: Responsible Implementation..... 329**
 - “Black Hat” Everything—Purposeful Pessimism..... 330
 - Facebook’s Celebration of Death 332
 - The Role of Choice 334
 - Slack: A Case Study In Open Design..... 339
 - People Are Not Edge Cases..... 346
- Index..... 351**

About the Author

Scott Riley is a designer and developer, specializing in interface and interaction design. He is devoted to exploring the intersection of design, brain science, and mental health. Scott works with companies around the world, trying to inject mindfulness and compassion into the systems they design. You can find details of all his work on his website: www.scott.is and his writing on Medium: https://medium.com/@scott_riley. But you should absolutely not follow him on Twitter @scott_riley.

About the Technical Reviewer



Zach Inglis is a web design and developer hybrid. He started teaching himself design and programming 19 years ago and hasn't looked back since. As one half of design agency Superhero Studios (www.superhero-studios.com), Zach takes care of web and print design, branding, and business strategy for a wide range of large and small clients. Zach was also co-organizer of HybridConf, an inclusive conference dedicated to bridging the gap between design and development.

Acknowledgments

Many thanks to the amazing researchers, scientists, academics, and industry professionals whose studious experiments and eloquent literature made this book a possibility.

Thanks to the Apress team, especially to Louise Corrigan, Nancy Chen, and James Markham for their continued support, feedback, and motivation—without you all, this book would still be a garbled mess in a random Markdown document. Thanks for making it real.

Thanks to Zach Inglis for being a wonderful friend and an only-mildly-annoying technical editor. Without your unique brand of compassion, kindness, and unwavering proclivity to stand up for others, this book would be a cynical husk of what it is now.

Infinite thanks to my family: Lynn (The Mum), Molly, Lili, Astrid, Nan, Gags, Pedro, Patch, and Frankie. Not a day goes by where I don't marvel at how lucky I am to have you all around me.

Immeasurable thanks to my partner, Roxie Ablett, for inspiring me to be a better human, for the countless pep talks, for picking me up after every exhausted breakdown, for teaching me linguistics, and for never judging me too harshly for my love of Oxford commas. You're the best thing that's ever happened to me. I love you.

Thanks to Chad Tomkiss for being the most incredible of friends, a constant source of support and love, bare music suggestions, and some of the most cherished memories of my entire life.

Thanks to Vic Bell. Words cannot express how fortunate I am to have you in my life. Your candid transparency is inspiring and you mean more to me than you'll ever know.

Thanks to Dan Kolodziej for being a lush soul and for letting me guinea pig my workshop techniques on real-life projects. I promise to never tell anyone about your Cream tattoo.

Thanks to Laura Bee. Your optimism and love for the world is almost as contagious as your accent. Your support throughout these last few years has kept me going.

Thanks to Meg Lewis for teaching me that it's okay to embrace yourself; for being the most badass, inspirational designer I know; and for always bringing compassion, love, and incredible thumb heads to the world of design.

ACKNOWLEDGMENTS

To Jade Pearn for being the most insightful and empathetic product partner I've ever had the pleasure to work with, and for being a treasured friend through some of the most difficult times of my life.

Thanks to Glen, Vic, Spadge, Greg, Dan, Dexter—and I guess James too—from my Simple as Milk family. Working with you has enriched my life in so many ways.

Double thanks to Spadge Nunn for giving me the initial spark some two years ago to even consider writing this book. Your enthusiasm and encouragement helped make this possible. I owe you a bath bomb.

Thanks to Stevie Buckley for being an okay friend every now and then, but mostly for introducing me to Barry's Tea.

Thanks to Ashley Baxter for the eternal support, trust, and friendship. Working with you has shown me time and time again how compassion, dedication, and grit are often the most important tools we have.

Finally, thanks to everyone out there who is fighting against oppression, inequality, oligarchy, and untempered capitalism. To the transgender rights activists, the antifascists, the intersectional feminists, the democratic socialists, the outspoken #BLM members, and every other compassionate activist in the tech scene and beyond. Your bravery is inspirational, and I hope in our lifetime we can help fix the world.

Introduction

I want to preface this silly book of mine with some musings on our role and responsibilities as designers on this weird old planet of ours. While I'm loathe to add to the arrogant noise that lives in the dusty corners of the idea that being a designer somehow makes us special—different from the 'normal' people who don't BS their way through meetings and make derivatives of everyone else's work. In some way, it's impossible to ignore the fact that design decisions, even (especially) in their absence, impact the world. As such, we find ourselves in a position where our decisions, ideas, failings, and oversights will, invariably, affect other humans.

When we consider that any interface is a product of thousands of these decisions, ideas, failings, and oversights, we are faced with little choice but to accept the responsibility that comes with that.

Design, in its most abstract sense, is emotional manipulation. It is a mechanism for eliciting changes in the brain. This doesn't, however, make design unique, or evil, or inherently problematic. In fact, to me, it further highlights its propensity for banality.

Music, art, video games, food, literature, film, photography, and myriad other indulgences to the senses all instigate their own, visceral reactions—lighting up neurons, exciting receptors, uncovering buried memories, and exploring the pathways that form new ones. Whether knowingly or unknowingly to their creator, they are odes to the plasticity of our brain, to the malleability of our mind.

Design's quirk—what makes it something of an outlier among the arts—lies in its desire to knowingly and accurately manipulate. We want to *make* people tap the right buttons, we want to *make* people share links, we want to *make* people use our interfaces in certain ways. And, usually, we want to do these things in the ways that make our products the most profitable.

While “real” art encourages emotional exploration and ambiguity, basking in the in-betweens, design strives for surety, for absolution. This is evident in the buzzword-laden landscape of how we talk about our work. We might *funnel* people into our product, then *convert* them, and subsequently *activate* them into paying customers.

We often fall into the trap of treating interfaces—or flows—as linear, constraining our design thinking into a neat little one-lane road of successful micro-interactions

INTRODUCTION

with various milestones along the way. We even create nice little user flow diagrams, like an electronics engineer planning the nodes their signal will pass through between origin and destination. We sort them into personas and categories and deal with them in averages and denominations, often blighted and constrained by our own biases and assumptions.

The question here is, bluntly, where does real humanity fit into this equation? When “empathy” is our buzzword, our yardstick for abstract adeptness as a designer, why are we diluting our understanding and appreciation of humans down into decision trees and reductive personas? Why can your average designer write an atomic “user journey,” but tell you little to nothing about the wonders of the mind? Is it its response to intrinsic motivation or stress? Is it its unfathomable ability to adapt? Its beautiful fragility?

Through the reading of this book, however deep you delve, I’d like to propose we flip conventional design vocabulary, in relation to people, on its head. Rather than thinking about how we can manipulate people to buy into our products—using product-centric verbs like *activation* and *conversion*—I’d like to explore what products and interfaces can give back to the people who use them. Not necessarily in a broad, feature-level sense of addressing needs and solving problems, but in a more abstract, emotional one. I want to introduce new questions to our decision workflow, mainly, “*How could this make people feel?*”

We all have the ability to knowingly add to the world, yet we often refuse to acknowledge just how much we can potentially take from it. Design can solve problems, but we rarely accept that it can create them as well. It can empower millions, but we rarely accept that it can equally, aggressively oppress. Design is a responsibility.

I feel, passionately, that an understanding of the mind—however basic—is increasingly essential to anyone who wishes to design. If you champion design’s ability to empower, if you fetishize its capacity to “change the world”—consequently, I feel you have no choice but to accept its potential to damage, to accede that your decisions can alienate, that your negligence can marginalize.

This is no attempt to denigrate the pursuit of design, nor is it a doom-and-gloom psycho-novella about the impact of poor design. My intention with this book is to provide a spark. To distill the often-saturated worlds of cognitive psychology and neuroscience down to concepts and knowledge that I feel can best inform our pursuit, as designers, to empower and positively impact real humans. To pass on a sense that has grown and been shaped by the study of these fields. A sense of togetherness, of a self-imposed obligation to be diligent and cognizant of our design decisions’ impact on others.

Most importantly, if this book achieves anything, my hope is it makes you give at least one more damn than you gave before you started reading it.

CHAPTER 1

Attention and Distraction

Attention is a precious mental resource. Every day we are bombarded with decisions and trivial things we need to remember. *What should I wear today? Should I even get dressed? Can I just hide under the duvet until June? Where did I leave the keys? Is there enough milk left for cereal?* Even though we can seemingly instantly and instinctively answer most of these questions, there is a very real cost attached to each and every one of them.

Behind the scenes, our brain—through a process of selective attention—is constantly assessing our environment and deciding what occurrences or stimuli are important enough to bring in to our conscious mind, forcing us to actively pay attention. This process of attentional filtration is one of the most interesting and enlightening facets of human cognitive evolution, and, as designers, we must endeavor to hold ourselves accountable for our impact on this process.

We must understand that there is invariably and demonstrably a mental cost to distraction and interruption and, likewise, to any period of held attention. We must acknowledge that part of our job is to consider and analyze these costs when making design decisions. We'll see in the coming pages just how fallible and vulnerable the human attentional system can be. We'll explore the idea that most people—through merely existing, thinking, and working—place themselves under a notable cognitive load. We'll also explore technology's strange and often damaging fetishization of attention. To do all this, we need to first understand the various concepts and connections that make up our attentional system.

The Attentional System

Ever worked from a coffee shop? Did you notice all the stimuli there? The buzz of coffee grinders, the rumbling of espresso machines, the waft of fresh coffee? The incessant and energetic chatter, the clinking of ceramic, and the almost-too-quiet hum of whatever music is playing? Perhaps you merely categorized all of this as a jumbled entanglement

of “background noise” and proceeded about your business. Yet, were you still able to do good work? Did you find yourself almost impervious to this cacophony of stimuli?

This is selective attention at work. When you need to focus your attention on a specific, important task, your brain adeptly shuts out these low-priority environmental stimuli, banishing them to your subconscious and allowing you to affix your attention to the task at hand.

This is an amazing function of the brain. Think of all that’s going on around you in that kind of environment—every stimulus within it is a potential focus of our attention. Even so, we’re able to primarily direct our attention to our work. When we’re focused, we pay very little attention to the everyday buzz of our environment. Auditory stimuli blend together to form a blanket of sweeping background noise and visual stimuli are ghosted into our periphery. Everything that isn’t related to our task can feel worlds apart from our current self in that current moment.

Sensory Distraction

Look at Figure 1-1 and imagine being in this coffee shop.



Figure 1-1. *Traditional coffeeshop environment vying for our attention*

Now, what would your immediate reaction be to a loud crash? Say, someone dropping a tray of precious, overpriced lattes all over the floor. It's highly likely, unless your hearing is impaired, that your focus would be taken away from your current work and you would experience a rather immediate shift of attention to the source of the crash.

Your brain has just alerted you to an environmental stimulus that it deemed irrepressible from your conscious attention. It has, quite actively and quite instantly, distracted you from the task at hand and diverted your attention toward something it has deemed an essential attentional focal point. The crash was visceral and unexpected, and it contrasted with the humdrum background noise of the coffee shop. This kind of interruption, part of the startle response, is innate and defensive, and it triggers one of the most instantaneous reactions that the brain is capable of performing.

Our startle response is low-level, innate, and deeply rooted in our survival throughout prehistory. It provides a rather crude example of how our attention can be ripped from a task-positive mindset. Other subtler environmental distractions are what we often find ourselves facing in our daily lives. Buzzing and pinging notifications on our devices. A colorful outfit in a sea of cheap suits. The appetizing smell of good food. All have the ability to distract us when they are processed by our senses, and they all have an inclination to interrupt our current focus.

Emotional Distraction

Consider another scenario in the same coffee shop—you're working away contently in a state of prolonged focus on a new design, engrossed in your task, when unexpectedly you hear your name mentioned from a couple of tables away. Suddenly finding that one edgy serif that looks good in IK Blue doesn't matter—you look away from your screen and toward the source. You're allured by this speaking of your name.

This is very different from the crashing of dropped mugs or the contrast of a colorful outfit against boring attire, but equally as alluring. We have a very large emotional investment in our own name, and hearing it spoken is an occurrence most of us find difficult to ignore. But what does this tell us about our environmental processing? The mind has a fascinating ability to selectively subdue environmental stimuli while still, to some degree, processing that information. On the surface, the conversation where your name was mentioned was part of the general sensory nothingness that our brains are rote to relegate to background noise. If you were quizzed at any moment before

your name was mentioned as to the nature of the conversation, you'd probably remark that you weren't listening and that you didn't hear any of it. Yet, it wasn't ignored, and it wasn't silenced; you just didn't know you were hearing it. We might not consciously register or acknowledge any previous word in that conversation, but as soon as we hear our name, we're alerted to it. Our focus shifts. The brain, essentially, decides what in our environment is immediately important to us and, more often than not, we acquiesce.

Both types of situations exemplify our brain's attentional filter at work. They're responses to sensory (the loud crash or the colorful outfit) and emotional (our name being spoken) distractions. By dragging us out of a focused mindset and diverting our attention toward the source of a stimuli the brain feels is important (either biologically or evolutionarily bound to our survival instinct or something that we've internalized as emotionally substantial), the brain is micromanaging exogenous and endogenous inputs and attempting to surface the correct, most important issues. Sometimes these distractions keep us alive; other times they're genuinely and categorically useless. Sometimes your brain is a tour guide, diverting your attention to the contrasting wonders of your world. Other times, it's a cat bringing you another dead mouse from the garden, or your kid bringing home another finger-painting for the refrigerator. (It's okay—you don't have to pretend it's good. No one here is going to judge.)

Task-positive Mode

Behind the scenes, what is actually happening between our task and our interruption? While our mind is focusing on our work, we're in what is known as "task-positive mode"—a state of mind exemplified by the intense focus on a single task. In this mode, our conscious attention is directed toward our work. However, buzzing away in our subconscious, our attentional system is constantly analyzing our environmental stimuli, deciding what to hide from our conscious attention and what should be presented up front as important and critical. When something occurs that our attentional system deems worthy of our focus, mental resources are shifted and a new object of attention is revealed.

This is evolutionary, bottom-up human behavior and has long been integral to our survival as a species. This innate function of our attentional system dates back to prehistory, honed over tens of thousands of years during which failure to react to a clear environmental hazard was less likely to result in getting coffee on your shoes and more likely to result in getting your face eaten off by an actual bear.

We'll discuss top-down and bottom-up processing in more detail throughout this book, but for now, they can be loosely defined as the following:

- *Bottom-up*: Fast, instinctive and intuitive, subconscious.
- *Top-down*: Slower, more reflective, conscious.

In his excellent book, *Thinking Fast And Slow*, Nobel Prize winner Daniel Kahneman discusses a two-system model of human thought. System 1, he suggests, works heavily on heuristics and cognitive biases. It is “intuitive thinking.” System 2, on the other hand, is “rational,” working at measured, more meticulous levels.

Kahneman presents a compelling argument for the notion that we're still impulsive by nature, preferring heuristic to contemplation and instinct to pontification (whether that's good or bad for us or what the ideal balance between instinctual and purposeful might be is a source of endless debate). While we like to think of ourselves as attuned and intellectual animals, we're far more instinctive and irrational than most of us wish to imagine. Our mind still hasn't shaken off the cobwebs of our reliance on survival instinct.

In our modern lives, with our modern comforts, we're much less likely to be in situations where actual bears are trying to eat our face and are much more likely to be in situations where we need to apply critical and empathetic thought to succeed in life. Still, millennia of surviving bear-face-eating have our brains wired a certain way.

Back to our coffee shop one last time. If you've ever experienced the kind of unavoidable distractions discussed earlier, you probably would have spent a few post-distraction minutes more aware of those previously dulled stimuli. Perhaps you listen in on a conversation, or actively pause and listen to whatever song is playing. It's probably that one Bon Iver song that you only ever hear in coffee shops, but there it is—the fleeting focus of your attention.

Or perhaps, rather than something previously meaningless catching your attention, you found yourself lost in thought for some untold moments. Your mind, previously dedicated to a specific task, is now drifting from thought to thought, idea to idea. Whatever the resulting focal point (or lack thereof) is, it almost feels as if our instantly focusing on the initial distraction has “ripped” us out of our focused mode and, depending on our discipline and mindset, we face a potentially uphill battle to get back on track and achieve our previous, task-focused state.

These scenarios provide examples of just a few of an infinite number of daily life's attentional undulations. Perhaps on some days we never achieve that focused state and spend our time, for better or worse, daydreaming and mentally meandering. Maybe

on others, we get into a true state of flow and work for hours on end, cocooned from environmental stimuli and life's incessant distractions. More likely, however, we'll be modulating between the two states.

The Default Mode Network

Beyond an evolutionary, focus/interrupt imperative, our attentional system also allows for switching between an intrinsically task-focused mode to a more reflective one—one that incubates and allows moments of introspection, creativity, empathy, and nonlinearity. In foregoing our surroundings and perceptions when nothing requires our direct focus or attention and looking inward, we're placating our ego's desire to pontificate on ourselves, putting ourselves in a mindset to philosophize and internalize or simply allowing our mind to wander and our thoughts to drift. This state of daydreaming occurs within what is known as the default mode network.

The default mode network was a landmark discovery for neuroscience and cognitive psychology, sparking a wave of new thinking and new questions about how the brain operates in conscious resting states. This network, tying together discrete neural networks in various areas of the brain, becomes active when we have no specific task at hand to focus on, or when we're not having to immediately react to a situation in our direct environment. This mode of thinking—of being—is widely seen as our brain's psychological baseline. In a nutshell, our minds wander and we daydream until we need to do something, and once we're done, we're back to meandering.

While healthy individuals are quite able to focus single-handedly on a task, "zone in" on problems, and generally get things done, more often than not we humans find ourselves in a state of mind-wandering, self-narrating, future-planning, reflecting, and just outright daydreaming. Neuroscientists call this type of thinking stimulus-independent thought—essentially a category of images and thoughts that are outside of, and unrelated to, our immediate environmental stimuli. In their 2007 study, *Wandering Minds: The Default Network and Stimulus-Independent Thought*, Malia F. Mason and colleagues showed a correlation between subjects' moments of stimulus-independent thought and activity in the various brain regions that form the default mode network. When our mind is "at rest," our thoughts are drifting, amorphous blobs of introspection. Freed from the need to react to the environment or focus on an intense task, our minds produce unique images, create melodies that have never before existed, and transport ourselves through mental time and space—allowing us to imagine ourselves in the future and to reflect upon our past.

While there is evidence that our default mode network is responsible for this stimulus-independent thought, the reason why remains elusive. In concluding their study, Mason and colleagues offer a philosophical and erudite set of possible explanations for the mind's propensity to wander and deviate from assigned or assumed goals—from providing us with a baseline state of arousal to get us through remedial tasks; to engaging in “spontaneous mental time travel” in order to bring coherence to experiences we've had, are having, and are yet to have (I am personally hugely interested in the notion of our mind forming a temporally cohesive “self”); to, finally, the suggestion that maybe we're all just overthinking this BS and the mind wanders “simply because it can.”

The nature of the default mode network remains a somewhat controversial subject and is as much a source of philosophical debate as it is one of neuroscience and cognitive psychology. While there remains no direct evidence of the network's association with creativity (the cognition of creativity is unto itself a field of immense complexity), there is growing research and burgeoning theories that link the mind's resting state with divergent thought (a simplification of what goes on in our mind when we're “being creative”). The phenomenon of “insights” occurring when we remove ourselves from a problem is widely reported. Think of how many times you attempted to solve a tricky problem for hours on end and gave up for the day only to solve it that night via a seemingly spontaneous eureka moment 40 seconds into your relaxing bath.

Given that this daydreaming network may be invoked during spontaneous thought, improvisation, self-projection, and empathetic thought, it's hard to *not* imagine it having a net impact on creativity and self-actualization. In *Ode to Positive Psychology*, Scott Barry Kaufman and Rebecca L. McMillan posit that these feelings are the intrinsic and personal “rewards” of stimulus-independent thought. While the mind may not be actively engaging with a task—on the path to achieving a specific and known goal—daydreaming opens us up to highly gratifying personal acknowledgements; potential creative insights; and self-rewarding, top-down introspection. While to some observing neuroscientists and psychologists, this “task negative” mode of thought is seen as inefficient and counterproductive, to their subjects—to us—it presents a myriad of intrinsic, highly personal rewards. Kaufman and McMillan suggest that “we need a new focus and new metrics” when studying and measuring the personal psychology of our mind's default, resting state. Viewing these intrinsic discoveries and moments of self-projection and unbridled imagination as increasingly important, compared with goal-focused tasks, is a compelling notion.

The Danger of Default Mode

This daydreaming, default mode is not all positive reflection and creativity. For clinically depressed individuals, the daydreaming, projection, and self-analysis of this mental mode can instead be taken over by feelings of guilt and shame. This creates a horrendous situation where an unoccupied, depressed mind defaults to what is known as depressive rumination—a constant and churning negative association with one’s self, one’s past experiences, and one’s future prospects. In their meta-analysis of studies on the default mode network’s association with depression, Dr. J. Paul Hamilton and colleagues portrayed that depressive ruminations were essentially “hijacking” our self-reflection and introspection processes. If you’re interested in the real details of this, Hamilton and his colleagues suggest that an overactive default mode network unto itself is not an indicator or predictor of major depressive disorder, but that a “functionally united” subgenual prefrontal cortex (sgPFC) and default mode network “often predicts levels of depressive rumination,” so our default mode network is as apparently responsible for insight and reflection as it is rumination.

The “mental time travel” the default mode network seemingly allows us to perform is also not always a fantastical window into an endless play of plays either. For every exciting projection of ourselves into the future as the atomic “me” we strive to be, there’s the shame-ridden journey to the past—where we accidentally called our teacher *Mum* or our sexual partner *Adrian*. In our daydreaming, we’re just as capable of negativity and self-deprecation as we are positivity and self-indulgence.

Similarly, when we look at our focused mode, we can just as easily drift into dangerous territories. Being overly devoted to a single task can lead to a rather damaging myopia and a state of tunnel vision. Furthermore, a notable number of social anxiety episodes are triggered by intense hyper-focus on a specific reaction, phrase, or gesture. Hyper-focus on a task or activity can also often cause us to lose track of time, result in mental over-exertion, and lead to stress and irritability. At the extreme end of the scale, there have been numerous cases of otherwise-healthy people so engrossed with high-pressure work and demanding tasks that they suffer health complications, some of which prove to be tragically fatal.

While these may seem to be extreme examples and notably low on the subtlety scale, the real point here is that there is, for every individual, a range on the spectrum between hyper-focus and perpetual mind-wandering that constitutes as “balanced” for them. A disruption in either direction to that balance is something we must be wary of and at least be able to empathize with.

The Cost of Distractions

The Internet is a hotbed of distraction. Social media actively profits from “eyes on pixels,” and news outlets rely on revenue from increasingly obnoxious advertisements or through “pay walls”—often presented in their own unique brand of obtrusiveness—that block content for non-paying readers. Our phones are vibrating (or when they’re not, we might just treat ourselves to some phantom vibrations to let us believe they are) to alert us to any and all possible bits of information or occurrences that might be of use to us. In modern times, attention is a powerful currency, and there are many apps, advertisers, and products out there that will snatch it from us without deliberation or morals.

It’s important to note that our attentional system, like a muscle, has a finite and depleting amount of energy available to it. Neurons are organic parts of a living ecosystem and, just as our muscles require and consume more glucose when put under stress, so too do the neurons in our brain. By asking our brains to switch focus throughout our daily life, we’re actively depleting these energy stores. The constant depletion of this energy without the requisite replenishment from a good old rest can result in damaging levels of mental fatigue, irritability, and burnout.

This is one of those things that, once I’d learned it, was hugely eye opening and cause for genuine pause and reflection. How often had a mistake or poor design decision in a product I’d designed been the one tiny but critical bit of cognitive effort that caused someone’s burnout? Quite melodramatic, right? But the point remains that small but frequent acts of attentional switching and cognitive load slowly sap us of mental energy. We’re faced with more information in our day-to-day lives than ever before and, as our technology evolves far faster than our brains, we’ve entered a point of incessant information overload, arriving hand-in-hand with perpetual decision overload.

Decision-making

It seems, too, as though every bit of content we see online is an entanglement of decisions. Content is no longer viewed as *just* content. It is social currency. It exists, at least in part, to be shared. By attaching actions beyond simply consuming the content, we’re raising myriad questions and decisions. *Do I read this? Do I trust it? Do I like it? Should I respond? Report it? Should I ever, ever read the comments?* By simply asking our brain to make these decisions, alongside the continual attentional-hopping from one distraction to the next, we’re slowly sapping ourselves of mental energy. If that sounds

somewhat exhausting, consider how many individual “chunks” of such content you might see on one platform in, let’s say, 15 minutes of browsing. Now, extrapolate that over however many social networks you use. Now further extrapolate that over how long you spend browsing social media during an average day. That’s a potentially huge amount of attentional switching and decision-making for something so seemingly trivial, and that’s before we even take into consideration the cognitive load of notifications.

While we’re shifting our attention like this, we’re depleting our brain’s nutrients. By asking our brain to focus on different things and to make a slew of decisions, we’re forcing our prefrontal cortex to consume glucose, an energy source of limited supply. Once this is depleted, our ability to focus drastically lowers and we make irrational and impulsive decisions. Deprived of its cognitive fuel, our unfocused brain releases adrenaline and cortisol, hormones that are inherently tied to stress and anxiety. Once we enter this state of depletion, we’re cognitively hamstrung, unavoidably bottom-up thinkers, and we make more mistakes more often. The cure for this is proper rest and replenishment. Allowing the mind to wander, taking a break to eat, and getting a good night’s sleep are all remedies for a nutrient-starved mind and, somewhat ironically, are all things that distractions, interruptions, and notifications often keep us from indulging in.

And what about notifications? Glenn Wilson has shown that simply through knowing you have an unread email in your inbox, your effective IQ during task-positive focus can be lowered by as much as 10 points. This reduction (almost double that attributed to casual marijuana use) is attributed to a compelling need to respond, as noted by Dr. Wilson and colleagues’ subjects. Further on this subject, Gloria Mark and colleagues conclude in *The Cost of Interrupted Work: More Speed and Stress* that constant attentional switching and interruptions cause people to exhibit “more stress, higher frustration, time pressure and effort.”

Inclusivity and Equality

We must also acknowledge that attention is a privilege. Neurodivergent individuals—those of us who are deemed “non-neurotypical”—will often find their attentional system and central executive hampered, imbalanced, or abnormal in some way. Poverty, too, can correlate with reduced cognitive function. A 2013 study by Anandi Mani, Sendhil Mullainathan, Eldar Shafir, and Jiaying Zhao showed that the cognitive impact of poverty can be similar to that of a 13-point IQ decrease, suggesting that “poverty-related concerns consume mental resources, leaving less for other tasks.” This raises huge ethical questions in how we design for inclusivity and equality. While we may assume

that our products and interfaces are not cognitively over-burdening, and we might even attempt to validate that through user testing, this thinking and testing is borderline pointless if we're not considering vulnerable, exhausted, and impaired people.

One of my biggest pet peeves with user testing is how often the environments these tests are conducted in are overtly vacuous. That's before we take into consideration the inherent bias that comes from most companies handling and conducting their own testing. Obviously, if you're making decisions on the back of testing sessions that don't account for a cognitive load (simulated, at the very least) and attentional depletion, you are doing a disservice to many of the people who might rely on your product. Your creation might seem perfectly fine when you observe a facilitated testing session from behind a two-way mirror with little to distract your subjects other than your product and their internal, individual inclination to daydream, but how usable is it under load? Try giving your subjects math problems, of varying difficulties, to consider during the test. This is a common tactic used in neuroscience and psychology experiments to provide a vulgar-yet-utilitarian simulation of cognitive load. Or, even better, have them perform the test on a system that is pinged with varying levels of intermittent notifications, as one may experience in real life. Test your interface often, under various levels of distraction and depletion, and you've already made a step toward a more-inclusive practice.

As designers, part of our job is to apply objectivity and empathy to our decisions, not just design what we like. We need to extend that beyond the most obvious, visceral aspects of our work and attach a more holistic importance to how we consider our use, and required levels, of human attention. The idea that, as professionals, we should be able to pick the right color palette for, say, a kids' TV show website without simply defaulting to our own preconceived notion of a "good" palette is nothing controversial. Yet it's much easier to fall into the trap of assumptions when it comes to the invisible and the in-betweens in our work. Understanding that our products are going to be used on devices that are, by default, loaded with potential distractions is important and needs to be a consideration throughout our interfaces.

In fact, by creating for connected devices, it can often feel that we're already at a disadvantage. We know we'll be creating something that will invariably live on a device with a multitude of other apps, all vying for attention and all using their own tactics to attempt to grab that attention. However, if we're diligent with our decisions and honest about how much attention our product or feature actually warrants, we have made the first steps to building a product that people can decide to ignore when they need to and that is conducive to focus when it needs to be. Some of the best products out there make no noise at all and, when used, allow for faster and deeper immersion in a task.

iA Writer is a perfect example of this. It doesn't send a flurry of "notice me" notifications when not in use, and it has a beautifully designed interface that is all focus and Zen when in use. Everything about the application is designed to streamline on-screen writing, to help you fall into a state of task-positive focus, and to help you continue that focus throughout your writing session. When typing, the ability to only focus on your current paragraph or sentence is a keystroke away (Figure 1-2).

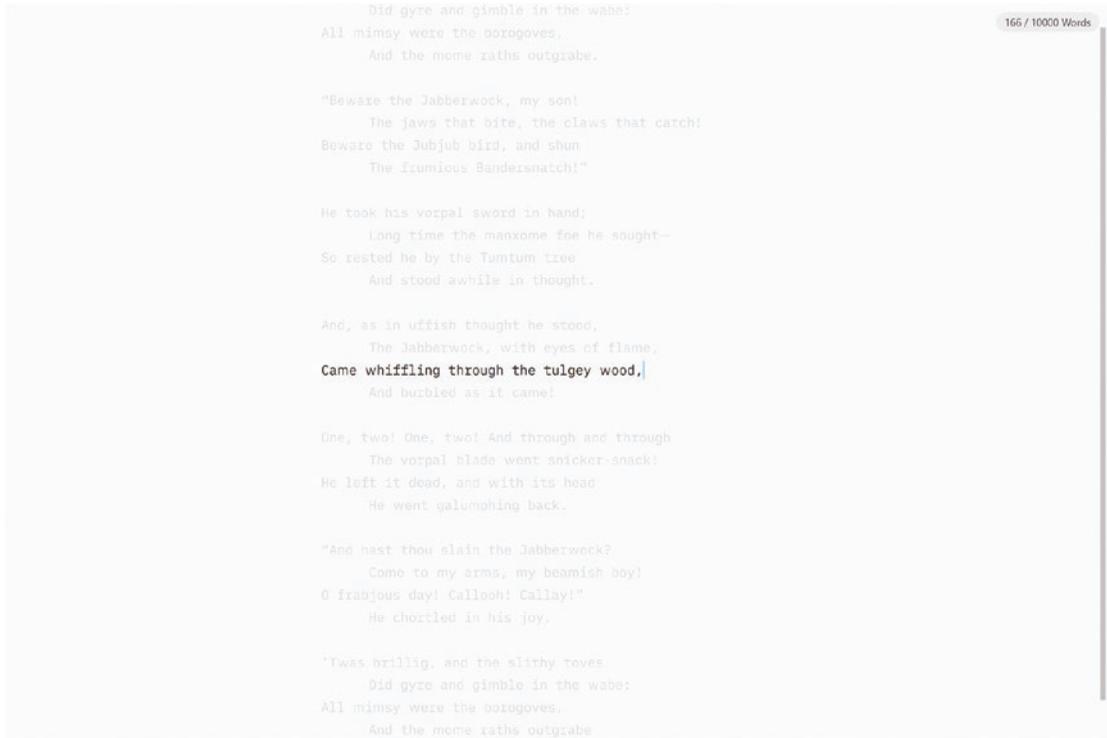


Figure 1-2. iA Writer's focus mode in action

The surrounding words fade into insignificance and your current focus is obvious. This simple, single feature allows for such in-the-moment consciousness and focus that it's actively more difficult to be distracted when using it. It's selective attention in microcosm. By actively fading out all but the most crucial aspect of the interface, iA has used design to (knowingly or not) mimic the spotlight of human attention. Design has already done some of the work for us, and we'll soon see that our brains really, really like that.

Habit-changing

On the other side of the attention-requirement spectrum are products that one would expect to be disruptive and distracting—at the right times, of course. The two most prominent examples I can think of are habit-changing products—think Duolingo for learning a new language through daily practice or apps such as Offtime and Moment that aim to mitigate social media addiction—and what I like to call “smoke alarm” products: apps that allow for truly impactful or destructive actions, where distraction before an important action is actually a critical (or even the only) feature. Just like a smoke alarm, you’re grateful for such blatant alert when it’s needed, but you most definitely don’t want them going off at 3 am just for the sake of it.

Examples of such apps are pretty limited and generally involve mitigating or protecting from some form of emergency or potential disaster. On a personal level, these could be connected glucose monitors for diabetic individuals that notify the wearer and selected emergency contacts when the individuals may be at risk. On a more distributed level, we can point to alert systems for natural disasters or systems such as the ill-fated Hawaii Missile Alert system that, in 2018, was incorrectly used outside of test mode (by some poor soul whose attentional system was obviously not quite in perfect shape that day), putting an entire US state on unnecessary alert.

Attention-grabbing

This dichotomy, however, is a rather naive. The fact is that the vast majority of applications will cover a wide, situationally dependent range of this spectrum of attentional need. iA Writer is an atomic example. Its prime purpose is to encourage and facilitate long-form writing, something that is only truly possible when distractions are dimmed and we’re in a focused mindset. Similarly, single-purpose, critical alert applications occupy the other extreme. The reality is that most apps will have aspects that require both simple, low-effort consumption and purposeful friction or distraction to instigate slower thinking and garner focused attention.

A prime example of this is GitHub’s repository deletion process. As the most prominent collection of free and open source software, GitHub makes it possible for millions of people around the world to use and reference your code. By deleting a repository, there’s a very real chance that a project many people rely on will cease to exist. Naturally, GitHub wants to prevent such occurrences—hence, their rather obtrusive delete flow (Figure 1-3).

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository. [Learn more.](#)

Source
GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more.](#)

None ▾ Save

Theme Chooser
Select a theme to publish your site with a Jekyll theme using the master branch. [Learn more.](#)

Choose a theme

Danger Zone

Make this repository private Make private
Hide this repository from the public.

Transfer ownership Transfer
Transfer this repository to another user or to an organization where you have the ability to create repositories.

Archive this repository Archive this repository
Mark this repository as archived and read-only.

Delete this repository Delete this repository
Once you delete a repository, there is no going back. Please be certain.



Figure 1-3. *GitHub’s danger zone*

To delete a repository on GitHub, you follow some pretty standard steps—there’s a settings screen that’s split into useful categories and tabs. Some way down that page is a section, with a big red border, aptly titled “Danger Zone.” Just in case you were still wondering if they meant business, here they are throwing Kenny Loggins at us. The “Danger Zone” is home to destructive actions that GitHub really, really doesn’t want you accidentally performing. Still, a decent smattering of red coupled with Kenny Loggins’s greatest hit might not be enough, and GitHub has one final trick up its sleeve to make absolutely certain you want to delete your repository. You have to actively type the full name of the repository into a text input to confirm the deletion (Figure 1-4).

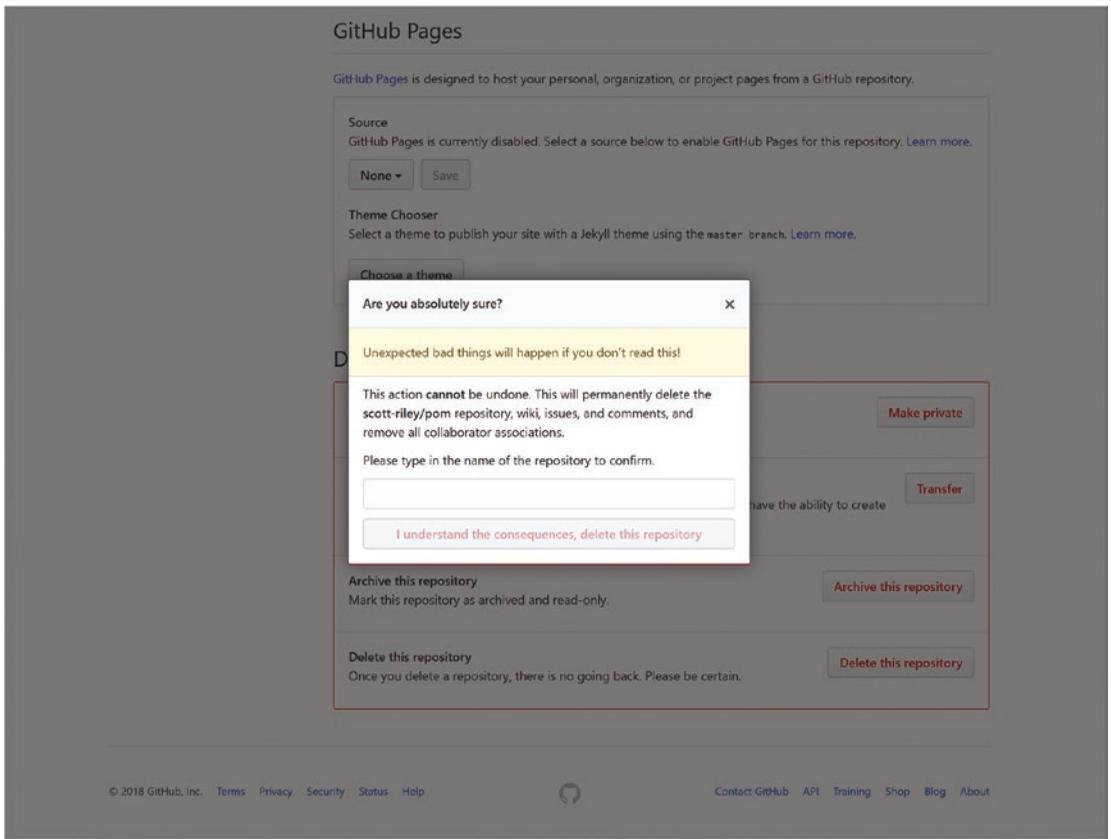


Figure 1-4. *GitHub forces you to type the full repository name in order to confirm deletion*

Inattentive blindness means we miss a lot in our environment, and this is amplified when using a device that is itself a cause of many distractions and interruptions. It's feasible and fair to assume that, in certain situations, we may be too distracted, tired, burned out, or stressed to notice the red, to notice the Danger Zone, and to remain on auto-pilot and accidentally hit the wrong button. I think we've all lost work due to similarly "dumb" reasons before. What GitHub does is prevent this with a task that is extremely difficult to *not* focus on. It grabs your attention, with zero subtlety, through actively reducing simplicity. By forcing you to slow down (typing is hard, relatively speaking), the designers at GitHub are doing their best to make damn sure you actually want this codebase gone.

This is GitHub's "smoke alarm," and it has its merits. It goes to show that there is a time and place for instigating an attentional switch. While there is a cost associated with slowing the pace down and being this disruptive, it's most certainly not as detrimental

as the emotional effects of accidentally deleting a codebase. This also works because it occurs in a relatively clean, distraction-free interface. GitHub’s baseline is that of subtlety, bordering on utilitarianism, and when such a chilled-out interface goes from 0 to Kenny Loggins this quickly, the *change* itself is as important a primer as the aesthetics and the content.

The “are-you-absolutely-sure-you-want-to-delete” paradigm is probably the most available example of attempted attention-grabbing in digital interfaces. It’s also probably handled badly way, way more often than it’s handled well. Simply burying your delete functionality deeper into your navigation and/or throwing a “red” version of your app’s typical alert modals asking someone if they’re sure they want to delete is often not enough. The goal is to slow people down, to help people engage in more-deliberate top-down thinking. This can only be achieved if you understand the baseline from which you’re working. If your app is already full of interruptions, or advertisements, or simply just poorly-designed and cluttered, you’re operating from a different baseline than an app that is well-designed and conducive to focus. You have two options here: you can lower your attentional baseline or you can attempt to make your “smoke alarms” more impactful. In case you can’t guess, I much prefer the former.

Attentional Economics

Again, and plainly: attention is finite, limited, and valuable. As designers, I believe we must see ourselves as a kind of attentional economists, as ridiculous (and boring) as that sounds. While a deep and scientific knowledge of attentional models is not essential, an intuitive and empathetic approach to “cognition as a currency” can be hugely impactful.

Trace the lineage of a design decision far enough back and you will always hit a presupposed “need” for attention. Our errors are red because red is a danger color, and danger colors cause a distraction powerful enough to capture attention. We animate objects when their state changes because we believe that new state to be important, and change—especially when interpolated through motion—is a sure-fire path to grabbing attention. By acknowledging the fact that directing attention away from Thing A and toward Thing B has a very real cost attached to it, we’re able to rationalize and hopefully empathize with that switch—especially, for example, if Thing A is a compelling long-form article and Thing B is a bouncy red advertisement with an auto-playing video. (We’d actually want to double that distraction cost, at least, given that there’s an inkling of hope that our reader would continue after that full-on sensory attack.)

The point of this discussion is to not push for a complete moratorium on attention-grabbing and notifications while we figure things out. There are, after all, many apps and products out there that integrate seamlessly into our daily life that the odd well-timed notification can genuinely enhance. The point is that we must take an honest and effortful inventory of the attention we're asking from people who use the things we create. Too often we convince ourselves that our products are deserving of smoke-alarm levels of attention when the reality is far more humbling.

What good, responsible design gives us is the luxury of cognitive headroom. By operating efficiently and simplifying complex tasks, an uncluttered interface puts itself in the best position to be impactful when necessary. And this isn't restricted to destructive actions. Small degrees of mindful friction or slight difficulty bumps here and there can actively promote learning and memorization—it's part of why education apps are so heavily gamified these days. Another benign example of attention-grabbing is guiding someone's attention toward a success message after they complete their task. Positive feedback is another integral component of affect and education, so if you have the mental space to sprinkle in some cutely animated success message, then you absolutely should!

Taking the Inside Out

It would also be undeniably remiss of me to not focus on the beauty and potentially evolutionarily critical role that digital devices, the Internet, and design can play in this age of monetized attention. For all the pings and buzzes and popups-inside-of-popups, digital interfaces, proliferated content, and technological advancements have supercharged humankind's most incredible propensity as a diverse and distributed species: we take what's in our brains, yank it out of our brains, and put into the external world. And then we share it. This act, one of the most incredible and cherished aspects of a connected world, is that of creating distributed knowledge. In fact, while it's easy to rain doom-and-gloom on the state of the internet as an omnipresent nuisance, its role in our species' ability to get 'stuff' out of our brains and into a consumable format is astounding. It's sparked a cornucopia of new styles and means of creation, proliferation, categorization, markup, formatting, micro formatting, expression and consumption. It plays into this fascinating, innate desire to teach and learn.

Since shortly after prehistory, as we made our first steps into becoming an agricultural- and trade-centric species, one of the main tenets of human evolution

has been our ability to use our external world as an extension of our minds. By writing something down, we're removing our brains' need to memorize and rehearse that information. Beyond our ability as individuals to unburden our minds of trivial information, we've gone further and further down this rabbit hole of externalizing our internal processes. Now cultures, societies, and industries hinge on the distributed teachings of our ancestors and peers. Arguably our greatest accomplishment as a species lies in our desire to use the external environment to dump, record, and share our knowledge, facts,, and memories.

We've continued that tradition through books, newspapers and later broadcast media—creating a form of distributed intelligence, knowledge, and memory that, to me, is the integral, defining factor of humankind. The Internet has given us a platform like no other to partake in this act, largely away from the traditional propaganda-leaning, government-controlled multimedia deliverance of recent generations. That is something to cherish, defend, and maintain.

On top of the Internet's growing impact as our species' source of distributed knowledge, the evolution of digital interfaces and devices has also allowed us to further our efficiency in creating that content. From simple, sync-able note-taking apps to the most convoluted issue-tracking-sprint-planning-project-managing systems imaginable, we have the tools and the knowledge to create, well ... tools for knowledge. Even social media, for all the—often deserved—ire that is thrown its way, plays a massive role in the proliferation of distributed information.

Design's role in this age of perpetual information overload is to organize content and present features in ways that reduce the cognitive load on those who have a desire or need to interact with them. By understanding and utilizing the “stack” of attentional management, intelligent heuristics, and emotional design, a task that would have previously demanded a lot of temporal and attentional investment can be simplified to the point where it becomes almost automatic, painless, and even (whisper it quietly now) delightful. We'll delve deeper into the mind's process of learning and habit-forming later in this book, but if we briefly define the process of learning and mastery as “practicing a task enough times for it to become automatic, requiring negligible cognitive effort to perform,” then I hope it's apparent how important good design can be in speeding up this process. After all, if our goal is to reduce the cognitive effort to the point of mechanization, the positive and instant net gain of design and technology's ability to make complex tasks more intuitive, easier to learn, or even automated entirely is absolutely compelling.

This, to me, sums up the power and beauty of positive, mindful design. When I search my mind for the reasons I still love design, this constant challenge of pushing toward reducing the cognitive load of modern life, even in the most minuscule, humblest of ways, is always front and center. I believe that through positive and empathetic learning and application of the tenets of cognitive psychology that most impact our field—attention, perception, motivation, learning, and reward—and by responsively utilizing the many shortcuts and affordances favored by the human mind, we can make conscientious and healthy design decisions.

While the following chapters of this book will go more in depth into the heuristics and specific cognitive effects of their respective topics, the remainder of this chapter will attempt to look into attention as a whole and discuss a few of the overarching attentional biases and heuristics that we can look to apply in our work.

The Cognitive Miser

Remember earlier in this chapter when I suggested that part of our job is to be “attentional economists”? I’d absolutely agree that that suggestion makes me sound like a ridiculous human, but it turns out that we’re all apparently subconsciously economizing our own cognitive currency. The phrase “cognitive miser” constitutes a theory of social cognition that, ultimately, boils down to the notion that we’re all cheapskates when it comes to our mental energy expenditure. This theory, introduced by Susan Fiske and Shelley Taylor—expanding on the work done by the likes of Daniel Kahneman, Amos Tversky, and their colleagues on the subject of heuristics and attentional biases—suggests that, at our core, we consistently default to easier methods to solve problems and understand our world.

We explored the notion that the mind has two modes of processing: bottom-up, where we respond intuitively and instinctually; and top-down, where we are more analytical and sophisticated in our thought. The cognitive miser theory is rooted in the notion that the low-effort, bottom-up processes will generally win out in situations where they can be safely applied. The vast majority of our decisions and problem-solving involve some form of simplification, shortcut, or heuristic. Simply put, we’re very lazy, and if we have the opportunity to apply a simpler heuristic to a problem or a decision, then we almost always will. This defaulting to low-effort processing, like so many of the brain’s complex processes, wildly varies in usefulness depending on the context and the individual.

As an example, imagine you're at a restaurant and you're trying to pick what meal to order. A top-down approach to this decision would likely be to slow down and think about your requirements, perhaps calculating how many calories you've consumed today and what a reasonable further number of calories would be; to consider whether you've eaten enough protein; to take each option that meets your nutritional needs and compare its price to the other options that fit your requirements, and to maybe ask your server for some paper and a pen to work it all out, and oh look now everyone hates you. More likely than not, you've probably engaged in what Nobel Prize winner Herbert Simon coined as *satisficing*—the notion that, in most noncritical situations in life, we're inclined to select something that appears “good enough” rather than performing the rather effortful task of finding the absolute best candidate from a list or group.

Chances are—when choosing food from a menu—you'll engage in this act. The process of satisficing doesn't ignore critical information or result in random picks—if you're on a high-protein diet, or you're vegan, or you're on a strict budget, then you'll actively steer clear of the options that obviously defy your diet or budget—but beyond that, your choice is quite likely going to be good enough, but not ideal. We engage in this act all the time. We might not categorically know that we've picked the best tool for a specific job, but we're generally content to opt for something that checks enough boxes to be a decent option, favoring it over others because of its availability or through some other cognitive bias.

Satisficing is what allows us to settle on sufficient candidates for the mundane decisions in our life without wasting precious time and cognitive effort processing them beyond their most obvious characteristics. Did you wear the right socks today? Chances are you barely paid any attention to that decision. Perhaps if it was cold, you opted for the first pair you found that was thick enough to keep your toes warm. Or maybe you wanted a specific color, or material, or length. What you probably didn't do is empty your drawer of socks, lay each one down individually on the floor, stand in the epicenter of your newly formed sock circle, and perform a deep CSI-style forensic examination of each sock's ability to stand up to the trials and tribulations of the upcoming day. Unless you did do that. In which case, you're probably my hero.

There's a real danger in assuming that people will perform a truly effortful cost/benefit analysis when using our products or interfaces. Even more so when they're choosing between them. Instead, we must embrace satisficing, along with the many other mental shortcuts our brains make when cognitive effort is at a premium. I believe that what we *really* mean when we say that good design simplifies is that good design

strips away information to a density level that allows for optimal satisficing. By managing the amount of content and features we present to someone at any one time, we're maximizing the possibility that a non-ideal process will result in a positive decision. Of course, the flip side to this is that through obfuscating information that is useful to a person but detrimental to a company (cancelling your account on a subscription service, for example), our efficient-but-naive decision-making process can be exploited. Such nefarious behavior has led to myriad *dark patterns* across digital products—little nuggets of malpractice that lay in wait to exploit our fast-paced decision-making. If you get far enough through this book, we'll have a proper little sit-down and talk about why these eventually pan out to benefit no one, but for now, just know that they exist, people hate them, and the people responsible for them are resounding twits.

Satisficing is not the only energy-saving shortcut our brains take. In fact, discovering new heuristics and cognitive biases is seemingly a pastime of many a cognitive psychologist. While there are a great deal of humorous and esoteric biases—such as the IKEA effect, a very real bias that suggests we place more value on something we've assembled ourselves, even if it comes out as complete trash—there are also some that are critical to our understanding of how humans make sense of our world, especially when the information we need is missing or too cognitively demanding to process in time to act.

Before delving into the more integral heuristics and how we can use them responsibly, we should really look into one of the most fundamental aspects of how we organize our world: *categorization*. Categorization is as integral a mental process as any and has a profound impact on our relationship with, and the notions of our place in, the universe.

Categories and Cognitive Economy

I spent a decent chunk of my early career designing and building websites for local councils in the United Kingdom. While it wasn't the most riveting of industries (unless you love Internet Explorer), it did expose me to sites absolutely brimming with content. I'm talking thousands of pages, hundreds of services, and an average of (probably) 60 million different forms. It also meant that I was designing for, potentially, *everyone*. When you work on wide-service government sites, there really is no "target demographic" beyond a probably shared location.

Such vast swathes of content present an extremely interesting categorization problem—how do you organize all that stuff? First, like most organizations, every local government is generally split into departments, each dealing with their own aspects of civil service. You might have a Libraries team that is responsible for keeping library opening times' content up-to-date, a team that's responsible for bins and recycling, one for tax, and so on. Like I said, riveting stuff. The immediate inclination in such an organization is to keep that content categorized by departments. This usually works well for the organization and its constituents. If you've just moved and you want to check when your recycling bins are collected, it's fair to assume that you'll find that information under a category such as "Recycling." This is pretty scalable too; the Council Tax category usually collects information on the cost of tax, surfaces features for paying online, forms to make enquiries—it's all pretty neat and trivial. So much so, that at the time I was making these sites, there was a nice, handy, centralized list of categories that almost every local authority site would use. It made it easy for all the different departments to be responsible for their content without having to sift through everything from other departments. It made CMS vendors' lives a little easier too. So you'll often see things like "Council Tax," "Licensing & Permissions," "Births, Deaths & Marriages," "Health & Wellbeing," and so on. Pretty straightforward, right?

Absolutely not. In what sane, empathetic world does someone lump births, deaths, and marriages into the same category? Imagine having to register a death certificate for a recently deceased loved one and being asked to sift through content about newborn children and new marriages. Or, on the flip side, needing to register the birth of a child and being asked to use your own attentional filtration to ignore all the scary death content. Designers love User Personas, so here's one: Ms. A has just lost her husband and has been informed she needs to register his death. She goes on her local authority's site. She is knowingly shown options about marriage and asked to consciously acknowledge that content in order to filter it out so that she can find and fill in one esoteric form.

This is a prime example of neglectful design. At some point, the internal categorization structure of an organization was preferred over the sensical grouping of content for those accessing a site. It's a common pitfall, too. After all, within an organization, you use certain internal words, phrases, structures, and hierarchies all the time. When moving fast, it's easy for that kind of structure to "feel right" and for you to move on. Oftentimes this miscategorization is a simple oversight, and the cognitive effort required to "translate" it is rather low, so it goes unchecked in user-testing, idiosyncratic

at worst. However, from a real-world, top-down perspective, poor categorization can have an emotional impact ranging anywhere from mildly frustrating to actually rather harrowing.

Aside from wanting to get eight years of frustration out in long form, why am I telling you this? Simply put, categorization is part of our evolution and continued survival. Through sensible, culturally sensitive, and well-considered grouping, we can take even the most complex, monolithic masses of content and turn it into a guided tour of knowledge and progression.

While design's role in an interface is not static—that is to say, sometimes its purpose is to intently obfuscate, sometimes to alert, sometimes to be invisible—its need to effectively communicate structure and hierarchy is almost omnipresent. For such a fluid, undulating concept, the core of almost any design decision lies in grouping and manipulating content based on its current importance in a given situation—to highlight the content that is relevant and to quieten the content that is not. Good design reduces the cognitive burden of information processing by optimistically performing a lot of the tasks we'd ask of our attentional system in the first place. The power of good categorization in this process cannot be understated.

It's also important for us to remove the somewhat narrow lens with which we view categories. Generally, when designers and developers think of "categorization," we're thinking in terms of a literal taxonomy. Think news sites with categories such as Sports and Politics. If it helps, replace categories with "groups." It's not 100% semantically correct, but it's good enough.

When we make trivial decisions, we almost always work based on categories. Certain objects have certain characteristics that cause us to mentally group them together. Sometimes these are based on physical characteristics. For example, we generally call all those weird floppy-feathered things with wings and hard noses "birds." Other times, we'll group things by function, for example, matches, lighters, and flint are physically dissimilar, but can be categorized as "things that we use to make fire." This is known as *functional equivalence*.

Finally, by grouping and categorizing items in the world around us, with the filter of our current needs and desires, we're able to very quickly make a decision about their application for a specific role. In the real world, we may be faced with the task of hammering a picture hook into the wall. The absolute best tool for this job is likely to be a hammer (probably a very specific kind of hammer, but let's not get carried away). Let's say we're not sure where our hammer is and, for whatever reason, next to us we have

a book, a sandwich, a shoe, a brick, and our pet tortoise. Without satisficing, we'd just go look for our hammer, but that'd likely take longer to find than it would for us to just improvise. So we satisfice; the book is paperback and a bit too flimsy to hammer a nail, and a sandwich is out of the question no matter how well-toasted it is. So we're left with shoe, brick, and tortoise in a very strange, situationally dependent category of "things I can hammer a hook into the wall with." Your shoe is expensive, and poor Frankie doesn't deserve to pay for your laziness by being bashed into a wall, so the brick it is.

While this is a rather crude example, it shows how the human mind can very quickly use the affordances it knows (or expects) objects to possess and group them into functional categories. Without our ability to acknowledge that bricks are hard, and hard things can be used to bash other things, we'd be stuck looking in every room for that hammer we haven't used for a year. Now, think about the ways those items are related outside of the lens of the task at hand. There's practically nothing that intuitively links a shoe to a brick, yet based on our needs we are able to recognize their suitability to a task.

Bringing this into our interfaces, we must be aware that various tools and features that might not innately be seen as linked could, depending on context, be intuitively categorized by someone performing a certain task within our interface. By grouping potentially discrete features together in a context-aware UI, we could greatly increase efficiency and reduce cognitive effort. We see this kind of context-dependent categorization all over interfaces, and it's a great way to keep people focused and on track. Twitter is a good example of this. Muting or blocking someone might be categorized as "actions performed against a user," and sharing or reporting a tweet as "actions performed against a tweet." However, in a situationally dependent category of "actions performed when a tweet is awful as heck," muting and blocking the tweet's associated user are quite clearly members of this category. By presenting these as auxiliary actions that can be performed from the context of a tweet itself, the interface allows us to perform the required subtask without a context shift (Figure 1-5).

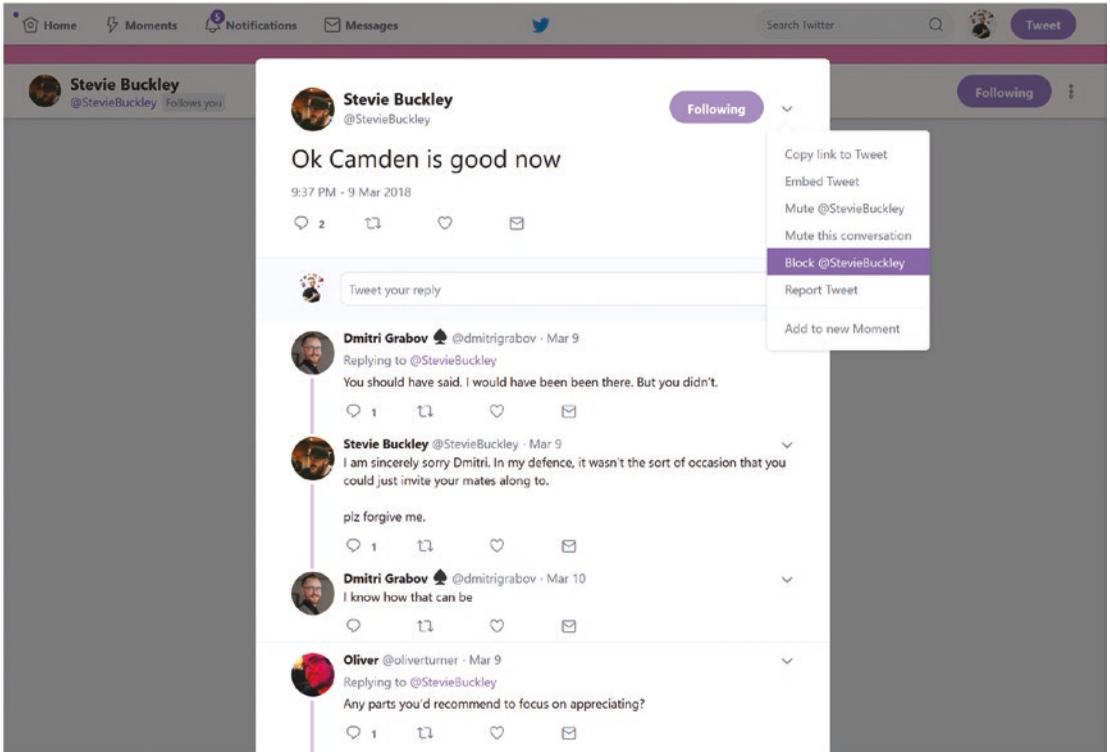


Figure 1-5. *Performing person-based actions within the context of a Tweet*

Categories also go hand-in-hand with reduced cognitive effort. This is especially true for larger interfaces, where the need to individually process every action or item would be an incessant assault on our attention. Through the process of good categorization and an interface that correctly communicates those categories, we allow the mind to disregard entire “chunks” of an interface at a time. If you’re using a product that has a wide range of features, but you only care about a categorized subset of those features, being able to zone an entire group of information out at once is infinitely less taxing than asking your brain to individually process every tool you have available to you.

A somewhat extreme example of this can be seen in InVision’s web app. InVision’s feature list is sprawling and, on paper, rather daunting. You can—deep breath now—build prototypes through clickable hotspots that link your screens together, comment on specific areas of a design, reply to someone else’s comments, preview individual screens, preview the entire prototype, inspect the various components that make up each screen and their attributes as a reference when coding it up, manage the workflow of individual screens in a kanban style board, invite people to collaborate, share individual screens,

and share the entire prototype. Phew. What’s more, this is just for the prototype side of the app. There’s a whole suite of sub-products in the same overarching app (Figure 1-6).

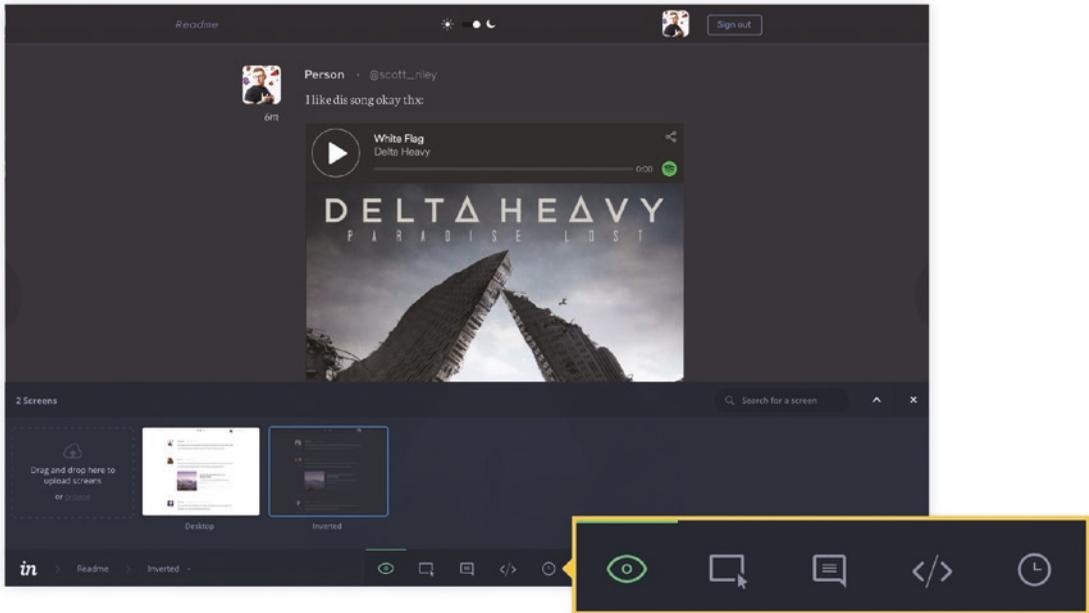


Figure 1-6. InVision’s available modes

What’s impressive about InVision is how easily it is to completely disregard the stuff you don’t need. InVision’s tactic for this is to present you with “modes.” Each mode presents and hides different subsets of features. In edit mode, you’re creating hotspots and deciding where they link to; in comment mode, you’re able to see, reply to, resolve and delete comments; in preview mode, the interface disappears almost entirely and you’re able to navigate a prototype as a simulacrum of the real app; in inspect mode, you’re given a whole UI for viewing the constituent parts of a design and the attributes most relevant to those who are coding the design up. The designers at InVision know that they have a behemoth of a feature set, and they have done a wonderful job in managing this.

What we see here is structurally a million miles from the Zen-like harmony we saw in iA Writer’s focus mode, yet they’re both prime examples of good design mimicking the mind’s innate proclivities. By preemptively categorizing and proactively juggling the visibility of whole suites of features in InVision’s case or through selectively lowering the prominence of unimportant information in iA Writer’s case, the interface is a precursor

to the likely cognitive functions that would take place within the mind itself. We're a species that thrives on making our internal manifest in the external. Just as we've evolved to annotate and distribute our brains' content to our external worlds, design is in a position to enrich this aspect by becoming a conduit for cognitive *function* itself.

This, to me, is the core premise of mindful design. And, yes, it sounds incredibly lofty, but through a concise understanding of the brain's attentional system, its ingrained desire to categorize, and its penchant for shortcuts and heuristics in decision-making, we can set ourselves the goal of mimicking these in our interfaces to the best of our abilities. And we can start small! Your interfaces don't have to instantly become cognitive savants; you can do better than most by simply categorizing well and being economical with attention. In fact, there's an inherent risk to trying any of this. It only takes a small overstep or a modicum of naivety to come out the other side and produce an interface that's too obtuse, hampered by a desire to obfuscate that results in key features being hidden too deep. Like everything else, there's a balance to be found.

Applying Attention to Design

We've seen that our mind's mental modes are never inherently positive or negative, that daydreaming can be as much about rumination as it is about creativity. So, too, we must acknowledge that our brains' inclinations toward shortcuts and biases as a whole are both essential and wildly detrimental, depending on circumstance and application. If we're unscrupulous or careless in applying this knowledge, we can actively harm and alienate vulnerable people. Our responsibility as designers lies first and foremost in the facilitation, protection, and agency of the people who use the stuff we make. This is increasingly important as tech (and, by proxy, design) infiltrates more and more aspects of human life. The current climate of venture-capital-backed startups; fast-paced design and iteration; an un-unionized, unregulated industry; and a hyper-focus on growth, money, and attention can often come to blows with the notions of little things like ethics and responsibility.

Remember:

- Think of your structure and categories from an emotional perspective as well as a thematic one. What makes sense to you or your internal team is often a source of frustration when asked in the real world.

- By grouping objects and content astutely, we allow the brain to disregard an entire “block” rather than having to selectively filter every item.
- Structure and hierarchy form the backbone of a good UI. By surfacing what’s important and filtering out what isn’t, we can replicate the mind’s attentional system to a degree.
- The state of an interface is changing all the time. Our goal isn’t to present a static structure of information, but to have our interfaces adhere to the perceived goals of the person using it at any given time.
- Change and importance are the two key factors we can utilize to guide someone’s attention around our interface. With good use of contrast, hierarchy, and animation, we can ease people through tasks that would otherwise be taxing.

Attention is precious and limited. We operate under a genuine constraint in a zero-sum game, and a huge part of good design lies in the simplification of the dense and the difficult.

Summary

In this chapter, we learned that the human mind is often found in a state of daydreaming, where thoughts often pass fleetingly from one to another. This is known as the default mode network, and it can be viewed alternatively as a “task negative” mode, where the mind is free to wander in the absence of mentally demanding tasks. We discussed how apps such as iA Writer, with its “focus mode,” can replicate the attentional filtering that the brain performs, fading unnecessary information into the background and highlighting the information we care about.

We also explored the cost of distractions—something that we should be especially aware of given the addictive and often over-reliant relationship we have with our devices. Switching attention from one task to another has a very real cost attached to it, and a mind that is neither left to wander nor allowed to partake in prolonged, focused flow is a mind rendered inefficient by the very tools that are supposed to allow for and accommodate increased efficiency. GitHub’s repository deletion “Danger Zone” and subsequent warnings, as well as the requirement to actively type the repository name to fully confirm deletion, were used as examples of how “attention-grabbing” might be used as a preventative measure.

The Cognitive Miser theory offers an explanation for how we conserve mental energy, acting somewhat as economists of our own attention. Part of our role as designers, where applicable, can be to see ourselves as an external layer in this equation, acting as gatekeepers of attention economics by preemptively limiting the information presented at any one time. Satisficing, we learned, is a means of shallowly analyzing the information in front of us to make a decision that we *believe* will “do the job” rather than painstakingly performing a cost/benefit analysis of every option presented to us.

We learned that humans, by nature, categorize the world around us. This is especially evident in our remarkable ability to functionally, or temporarily, categorize items that, when taken out of context, lack any kind of similarity. Our example of deliberating between a brick, a shoe, and a tortoise as members of the functional category “things we can hammer a picture hook into a wall with” shows how categorization allows us to present novel and creative solutions to problems.

If you take one thing from this chapter to apply to your design process, make it the understanding that there is a real cost associated with attentional switching and that distractions and held attention are not free. Be as objective as possible when balancing the attentional requirements of people against the business goals your product is attempting to achieve.

CHAPTER 2

Vision, Perception, and Aesthetics

In 1872, a painting was painted. Now, in the 1800s, lots of paintings were painted. But this particular one was an important one—and not just because it’s my favorite painting of all time. This painting is Claude Monet’s *Impression, Soleil Levant* (Figure 2-1) and I love it, in part, because it embodies the Impressionist movement, one of the most beautiful, and beautifully dysfunctional, marriages of art and science in history. I’m going to spare you the full meandering history of Impressionism and post-Impressionism, but I’d implore you to do some further reading if you’re in any way interested in art, bohemia, or anarchy. What is important to understand is the insight Impressionism gives us into some of our brain’s perceptual processes.



Figure 2-1. Claude Monet, *Impression, Soleil Levant*—1872, oil on canvas, Musée Marmottan Monet, Paris (Public Domain)

Around the same time that Manet, Monet, Bazille, Renoir et al. were hunched in Cafe Guerbois, twizzling fancy moustaches, cursing the memory of Napoleon, and plotting the downfall of the bourgeoisie (the “Batignolles” were reportedly staunch anarchists and bohemians), science was making its first strides into understanding the separation between sensory input and perception, namely that “what we see” is not what we see. While our understanding of the intricacies of perception has become more refined, more rounded, and deeper than these seminal discoveries and theories, the science ‘*du jour*’ in the late 1800s still rings true: the eye and the retina are only half of the equation when it comes to perception. Without a form of organization and resolution in the mind, the things we “look at” would be perceived as little more than visual mush. Impressionism played to this reality and embodied, to its core, the true nature of human perception. Perception is how we organize our environment, how we take what we “sense” and transform it into what we “perceive”—and it is hugely impacted by our own experiences of the world. What we know, what we expect, and how we feel all influence what we perceive.

Vision is the “bottom-up” sensory process of transmitting light information to the brain. Light is captured, turned into neural signals, and then sent to the brain for processing. The brain then takes these signals and applies some form of order to them—it catalogues, organizes, and contextualizes what we see into objects with meaning and associations; this is perception. And everyone’s perception is different. It’s theorized that varying amounts of this sensory data are lost before they reach the visual cortex, and we actually spend a huge amount of our time mentally filling in gaps, making assumptions, and hypothesizing based on what we receive from our senses.

When you look at *Impression, Soleil Levant*, what do you see? I see a river; I see a few scattered boats with people in each, some seated, some standing; I see birds in flight over the river; I see a bridge, a sunrise, and the reflection of that sunrise in the water; I see a scene and a moment in time. Remove the romance though, look slightly closer, and it’s really a whole mess of visible brushstrokes. Even now, Monet’s work looks almost unfinished. (At the time, his style was shunned, almost heretic in its brazenness.) Before Impressionism, fine art was on a crusade of realism. Brushstrokes were deft and unnoticeable; the work was aspirational and hyper-realistic; important people were painted in important situations; and the mainstream, at the time, was, quite ironically, rooted in the avoidance of harsh reality through art escapism. The Impressionists sought to bring their own form of realism to art; to bring it down to earth; to paint the peasants and the “bohemes”; to capture moments, not fantasies. Part of this philosophical realism was the principle that to paint these moments—to acquiesce to the ephemerality of natural light, to paint with an obsession for minimizing the number of brushstrokes—was to realize, on canvas, the nature of human perception. To me, Impressionism embodies almost every “modern” thought we have on design through its striving for minimalism and its refusal to patronize.

There’s certain magic to this work, along with the neo- and post-impressionism and abstract art that followed it, and that magic lies in just how well it explores the workings of our mind’s eye. When we observe a work like this, we’re almost compelled to insert ourselves into it. The soul of an Impressionist painting cannot be found on the canvas; it’s in the mind of everyone who’s filled in its blanks. Impressionism and abstract art only truly occur in our minds. We use our past experiences and our knowledge of the world to extrapolate form into meaning. Without this extrapolation, we would see the artwork how we would see the world without perception—as a mess of lines and color and incoherence. However, by asking us to fill in the gaps, to connect the lines, and to blend the colors ourselves (Impressionism used hard contrasts and vibrant colors often,

literally expecting the observer to mentally mix them), Impression leaves room for our own affect, our own emotions—it is exploration on canvas, and it has changed art forever.

The Iconic Abstraction Scale

Creeping further toward modern day now, in his quite excellent book *Understanding Comics*, Scott McCloud theorizes that the more “iconic” an image is, without losing its meaning, the higher chance that we’ll apply to it some of our self by attaching our own inferences. Similar to the act of observing Impressionist art, when we see simple illustrations, we’re asked to fill in some blanks and to come to our own conclusions. McCloud uses the term “iconic abstraction” to denote the level of deviation from real-world form—the premise being simpler, more cartoon-like forms represent higher levels of iconic abstraction from a real object than, for example, a photograph. (See Figure 2-2.)

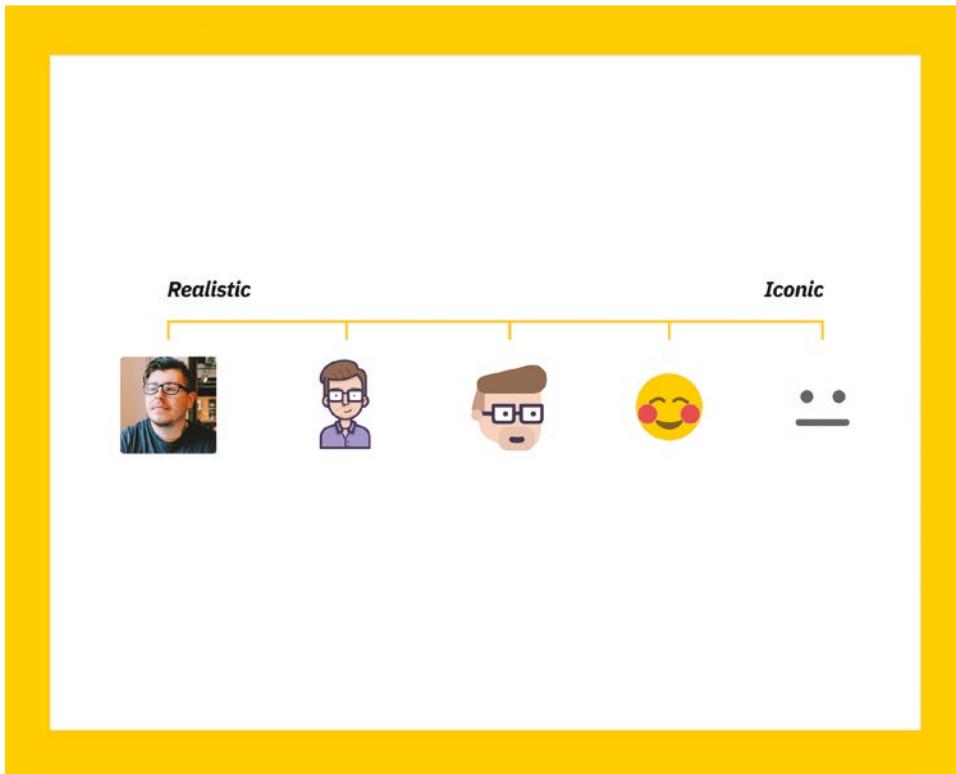


Figure 2-2. An example scale of iconic abstraction

While there's little in the way of empirical evidence for this theory, I do believe it's a wonderful meditation on just why we're so enamored with the abstract. Artwork that, without our ability to fill in blanks and create associations, would essentially be discarded as a cacophony of lines and brushstrokes become more than the moment or object it strives to represent. There exists enough conceptual space in these works for us to impart our own ideas and weave the threads of our own stories. I've long said (while acknowledging the inherent bias from the fact that some of my favorite humans are illustrators) that illustration is one of the most integral aspects of a modern brand, website or interface, and I think McCloud's theory and our willing cultural embrace of impressionist and abstract art gives us a hint as to the real power of emotional illustration.

Furthermore, what we "see" when we look at these works actively changes based on our working mental models of the world. Ask yourself this: when looking at *Impression, Soleil Levant*, how, do you imagine, would your perception of this image change if you were familiar with the river on which it is based? If, as part of your daily routine, you spent every afternoon in the Le Havre port, saw endless sunsets like the one depicted, and attached endless memories and associations to those moments? Conversely, what if you'd lived a life in which you had never seen a boat or even a bridge? How would you make sense of Monet's work without the requisite mental model? This is the power of abstraction. It nods toward a form that cannot exist without the inferences of our mind's eye. It's why we can see the sunset and its reflection in Monet's masterpiece, or the face in that doorknob we walk past every day, or Jesus Christ burned into our toast. When there is a form to be found, we will usually find it. (See Figure 2-3.) In fact, even when there is no explicit depiction of form, we can always convince ourselves there is.



Figure 2-3. *Can you unsee the “face”?*

Illustration and Photography

In modern interface design, our abstract art generally takes form as illustration and iconography. Unlike photography and videography (from herein I’ll be referring to both as simply “photography”), which provide us with a recreation of a moment or object in verism, our more iconic media—illustration especially—leave these “gaps” in which we can fill in our own stories. When presenting people with simple illustrations and icons, we’re not only able to imply meaning with fewer stimuli to process—reducing the overall visual complexity of our work—but we’re also giving people space to further interpret what they see. There is, too, a fine balance between this “pseudo-Impressionist” approach and a purely contemplative abstract approach. In our work for screens, we tend to lean more toward clarity and further away from chaos and open-to-interpretation works. How realistic or iconic your representative media (icons, illustrations, photography, videography, and animations) should be is as much a brand

consideration as it is anything else. An illustration is often more iconic, communicating meaning through a relatively small number of lines and shapes, while photography tends to be more explicit and aspirational and is used heavily in travel, fitness, fashion—essentially anywhere that speaks to our intrinsic desires.

Great photography in this sense can conjure up some powerful emotions. Aspirational photography can play to what is known as “mimetic desire”—the idea that we don’t really desire something until we see another human doing or using it. We constantly compare ourselves to others, for better and for worse, and seeing powerful, happy, and competent people can really hit home emotionally—provided the subjects are relatable enough for your audience. This style of photography, along with the dreaded “shock value” photograph, should be used consciously and responsibly, or not at all.

Another huge factor, especially with aspirational photography, is that of diversity. If you’re using photography in an attempt to uplift and empower and you omit an underrepresented group, you’re making a pretty plain and clear statement that you don’t care about that group’s members. The whole act of one’s relating to a piece of media relies on the ability to find or place a manifestation of one’s self within. If your imagery excludes, your potential impact diminishes. That’s not to say that by simply adopting a more iconic approach, you magically have no commitment to representation—just that the effects are starkest with the explicit nature of photography. However, if “further market reach” is the phrase it takes to convince you that representation and diversity matter, then I’d recommend putting this book down, leaving your bubble, and embracing some of the world’s cultures a bit. Seriously. Come back soon!

There exists, as lovely as it would be, no simple equation or checklist to help decipher whether an interface is better off using photography, illustration, or a combination of both. One reductionist, but sometimes useful, idea to consider is the notion that photography captures “specific” emotions and moments, while illustration allows for a greater range of potential inference. If we use McCloud’s scale of iconic abstraction and bastardize it a little, on the realistic side we’d have “explicit, single emotion, specific moment” and on the iconic side, we’d have “implicit, varied emotions, ephemeral” (Figure 2-4).

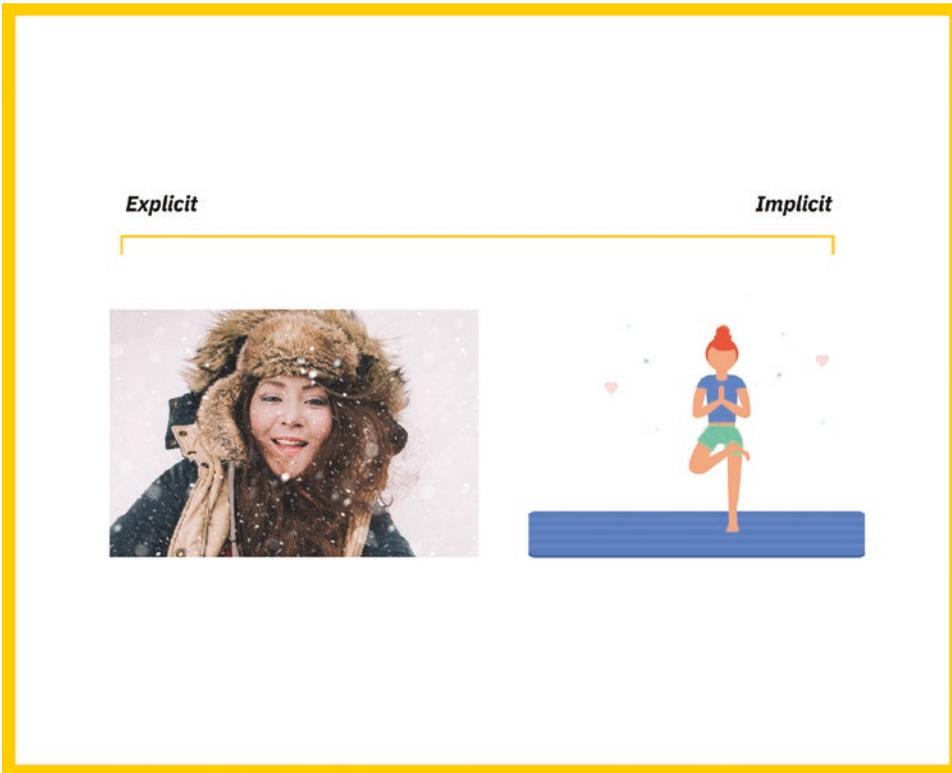


Figure 2-4. *The Explicit/Implicit Scale, appropriated from McCloud’s scale of Iconic Abstraction (Photo credit: Ashley Baxter, used with permission)(Illustration: Copyright 2017, Katerina Limpitsouni, unDraw, MIT licensed)*

That’s not to say that photography cannot be emotionally ambiguous and abstract and that illustrations can’t be focused or explicit—far from it! Artistic intent and human experience and inference both have a huge impact on these qualities.

The third of our triptych of digital media is iconography. While icons share many features with—and are often technically indistinguishable from—illustrations, they tend to serve a different purpose. Icons exist to simplify messages and actions, to signify state, or to attach an indicator of tone to feedback and responses. They provide us with a kind of visual mnemonic and a signifier for different ideas and actions that can occur within our interfaces. As a general shortcut, we can view icons as communicating an “idea” or an “action,” and illustrations communicating an “object” or a “scene.” While this is not always correct and while we have the rather ambiguous ideas of iconic illustrations and illustrative icons to contend with, this heuristic is more often applicable than not. We’ll delve into the subject of icons and their role in reducing cognitive load in Chapter 3.

The Gestalt Principles of Perception

The Gestalt Principles of Perception are a set of guidelines that aim to explain some of the inherent biases and approaches humans apply as part of our environmental perception. If I didn't lose you in Chapter 1, you hopefully know how important groups and categories are in helping us make sense of our world. Gestalt goes a good way to describe some of the more idiosyncratic ways we process and group perceptual information, or “visual field organization” as it's come to be known.

Gestalt is to visual design what Disney's “12 Principles” are to animation and motion design—everyone assumes that everyone else knows them, but everyone who writes about visual processing writes about Gestalt just in case. And thus the cycle continues. It's important to clarify, too, that these principles are not an all-encompassing “psychology of visual design.” The fact that Gestalt is relatively simple to illustrate and that its principles have quite admirably stood the test of time has led to a rather simplified notion that Gestalt defines perceptual psychology.

Another thing I want to emphasize with Gestalt is that, traditionally, it's used as a means of informing how we create “harmonious” and well-grouped designs. We'll explore in detail the idea of harmony in design a little later in this book, but as a free spoiler, harmony is both “necessary” and “boring.” What I'd like to focus on a little more than usual when we think about Gestalt is the yin/yang nature of these principles. By flipping them on their heads, we're able to see ways of creating visual tension, interesting layouts, and engaging content. We're also able to explore how “violating” these principles can grab attention. Of course, the yin to this yang is quite simple: we *must* be operating from a harmonious baseline for this tension to have the desired effect. Breaking the rules in an environment that doesn't seem to have any gets you nowhere but further down a spiral of chaos. Reference these principles well, use them as part of your tool set when creating your harmonious baseline, and then flip them on their head—when it's applicable and responsible to do so—if you need to create some interest, guide the eye, or bestow importance on an element of your design.

Proximity

The principle of proximity states that objects that are closer to each other are seen as grouped. This is probably the universal principle of visual design and goes a good way to informing why whitespace and layout are so integral to harmonious design. Figure 2-5 shows proximity in effect—chances are you’ve naturally grouped these objects based on proximity alone.

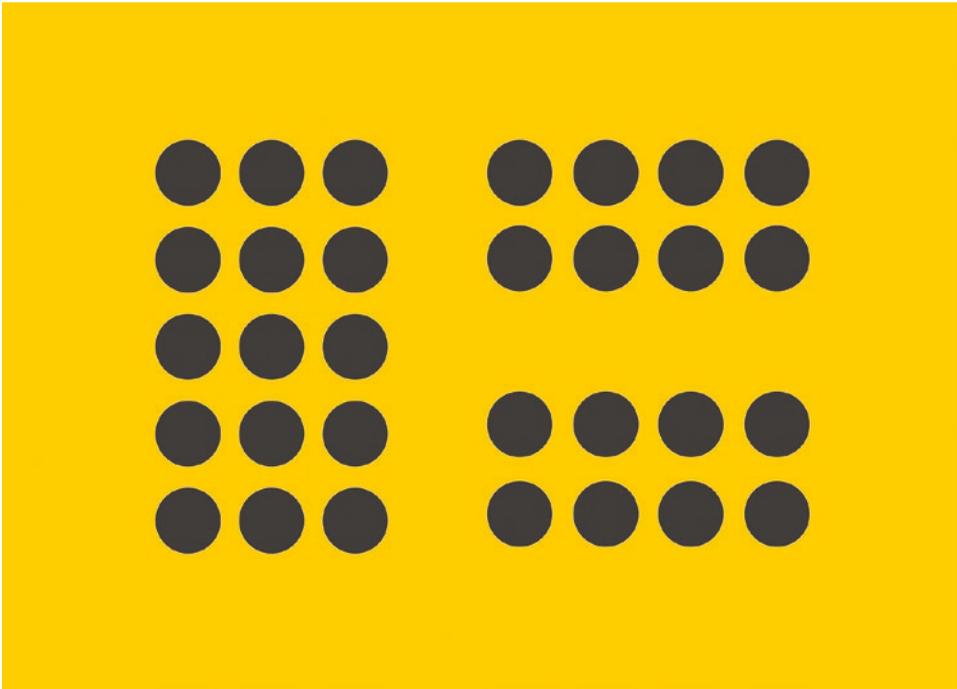


Figure 2-5. *Proximity in action*

In design, proximity informs some of our most basic guidelines. For example, when setting type, ensuring that a heading is closer to the paragraph it introduces than it is to the preceding paragraph (Figure 2-6) enforces the relationship between the two, detaching it from the less-relevant content.

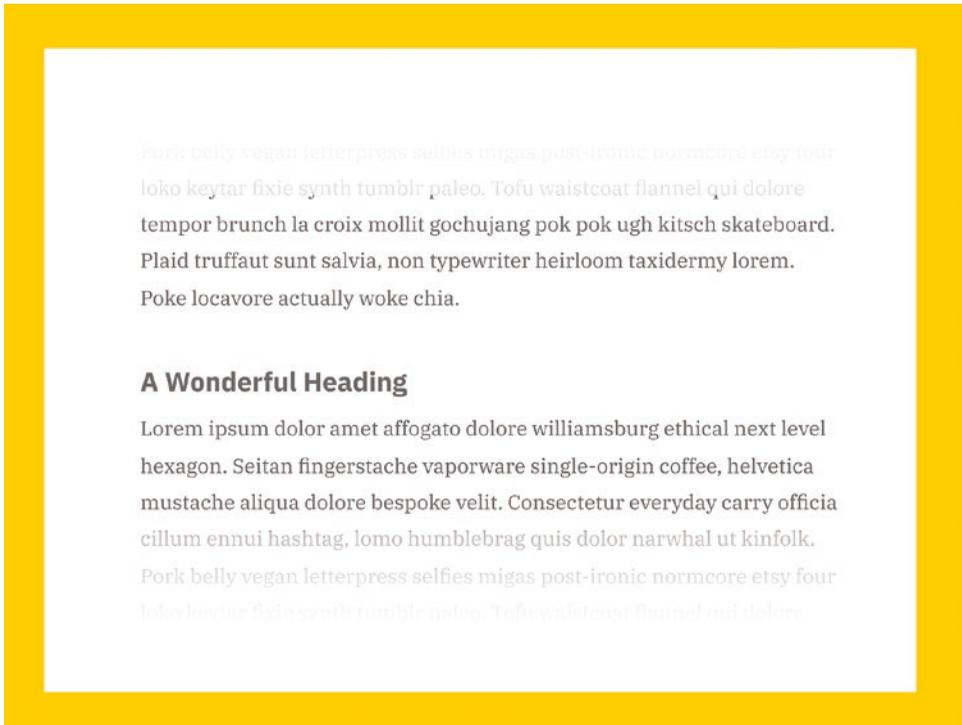


Figure 2-6. *The heading's proximity to the second paragraph indicates relatedness, as does its relative separation from the first paragraph*

The power of this principle lies in the fact that it allows us to communicate grouping, structure, and hierarchy without adding any extra elements to our designs. Many simpler designs can communicate relationships and hierarchy through proximity alone.

Similarity

The Principle of Similarity suggests that items with similar appearances and physical qualities will be grouped together. When you look at Figure 2-7, it's quite likely that you'll categorize the shapes into vertical columns of circles and triangles.

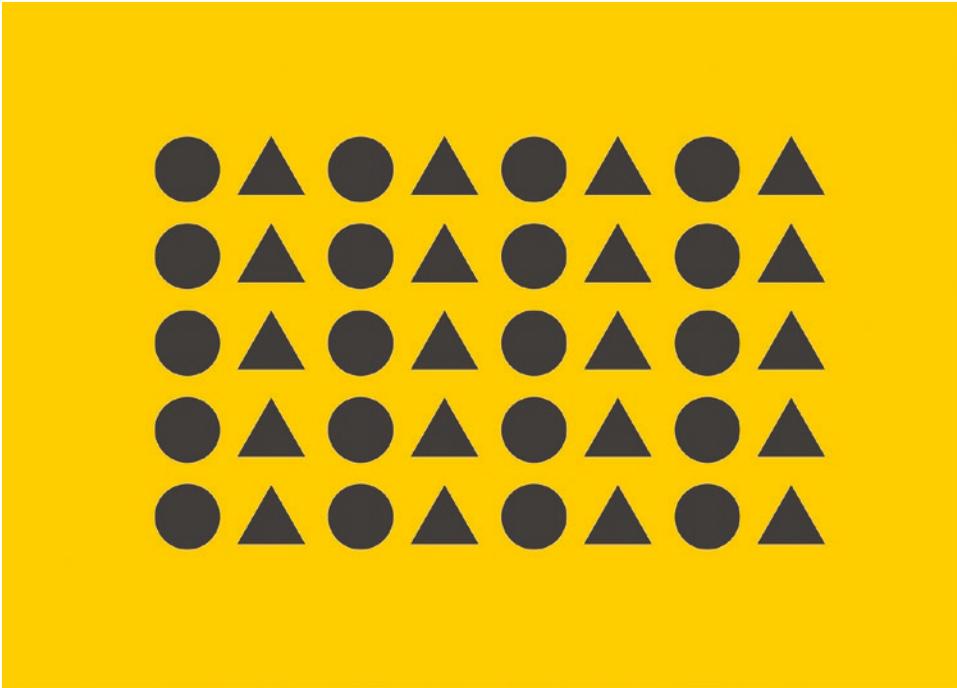


Figure 2-7. Similarity enforcing the perception of columns

Even though the shapes are spaced equally and are of roughly the same visual weight, your brain groups them together by their most obvious similarities. In Figure 2-8, however, it's much more likely you'll perceive horizontal rows.

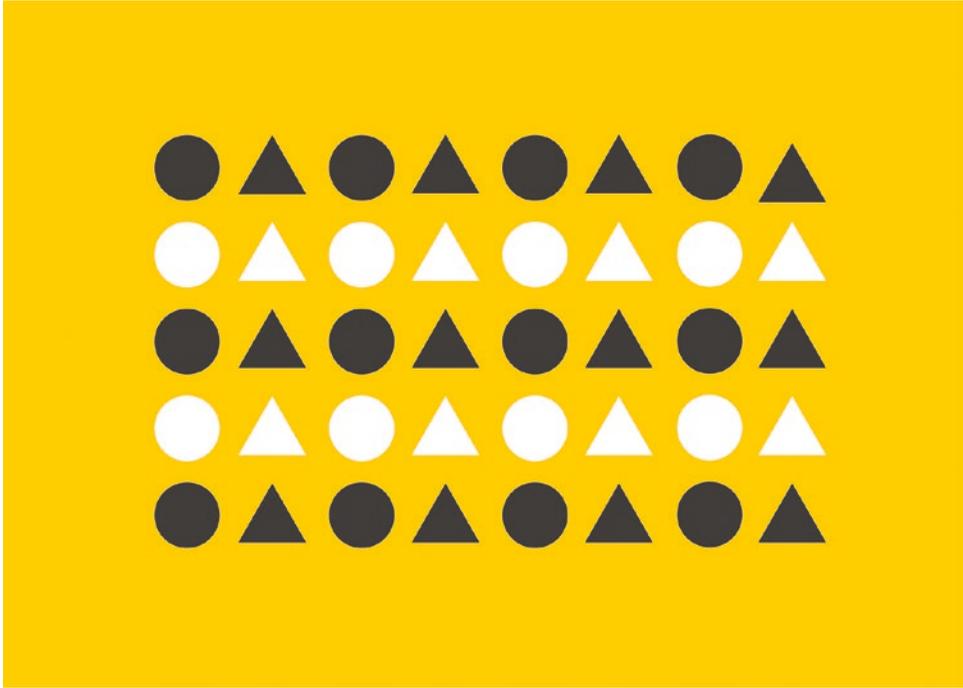


Figure 2-8. *Color similarity overriding the initial perception of columns and suggesting rows*

In this case, the color (horizontal) similarity is initially far more obvious than the shape (vertical) similarity.

In design, similarity can also help describe why we perceive different-colored words in a body of text as links—by categorizing them by their similarities, we allow them to imply paradigms throughout an interface, such as the text links in Figure 2-9.

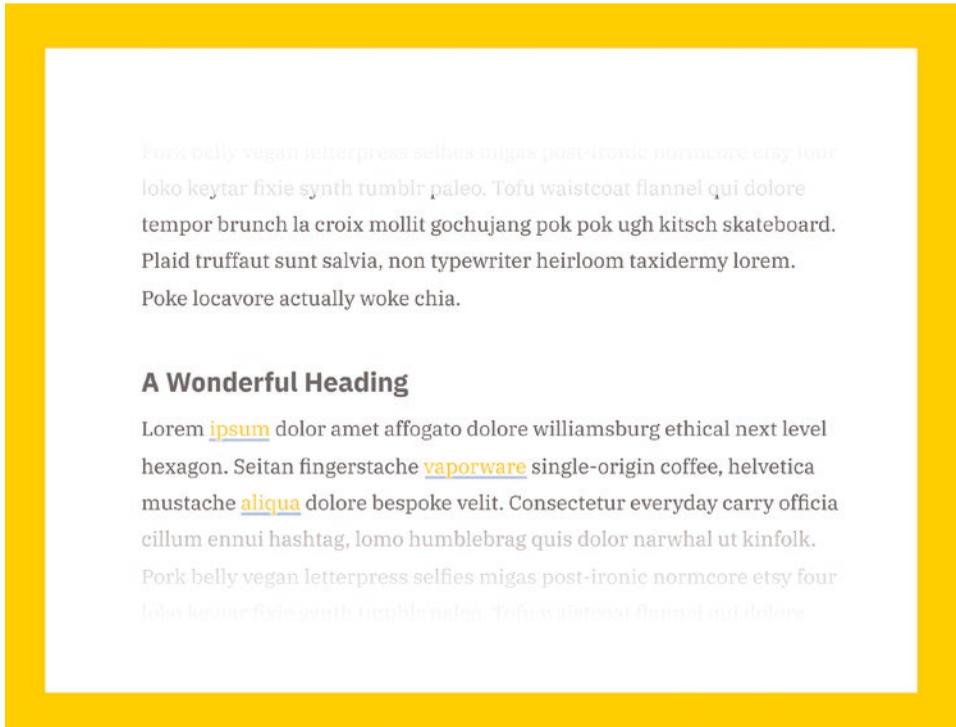


Figure 2-9. *Similarity as a signifier—we recognize the text links in the paragraph as grouped and communicating a common functionality*

This is known as a *signifier*, and similarity is a key aspect of communicating these paradigms by allowing people to codify and associate these visual properties with potential action and function.

Continuity

While proximity and similarity encompass many of the basic concepts we use most commonly in design, they don't fully account for one of the most important: "alignment." The Principle of Continuity suggests that elements that are aligned with each other are perceived as grouped due to our natural desire to visually "follow through" an element along a specific path or in a specific direction, as seen in Figure 2-10.

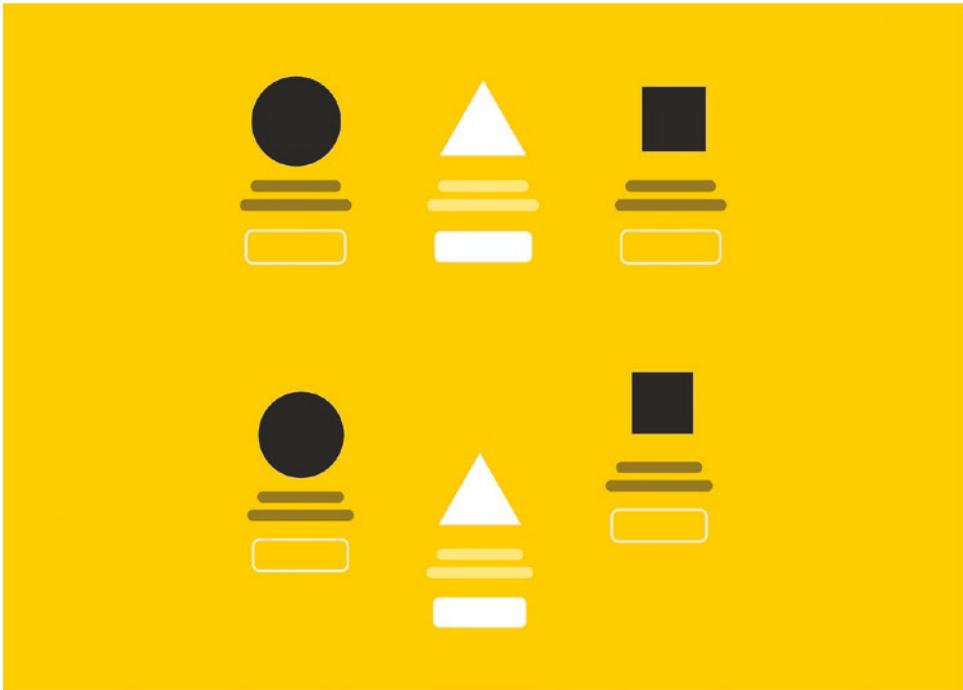


Figure 2-10. *Continuity helps us group the top row of elements together—the second row, not so much*

The first row in Figure 2-10 shows an extremely common row of elements and, while proximity and similarity have a say in their grouping, possibly the strongest indicator that they're a group is their alignment along the vertical axis. The second row shows a rather exaggerated misalignment between the elements, making it much more difficult to perceive them as a group.

Continuity can be enhanced with flowchart-style lines or arrows or hinted at with smaller design flourishes should you need to communicate a cohesive group that doesn't strictly align along an axis. Figure 2-11 shows an example of this; by drawing out a "path" that the eye can follow to connect each element to the next, their relatedness is communicated in spite of their misalignment.

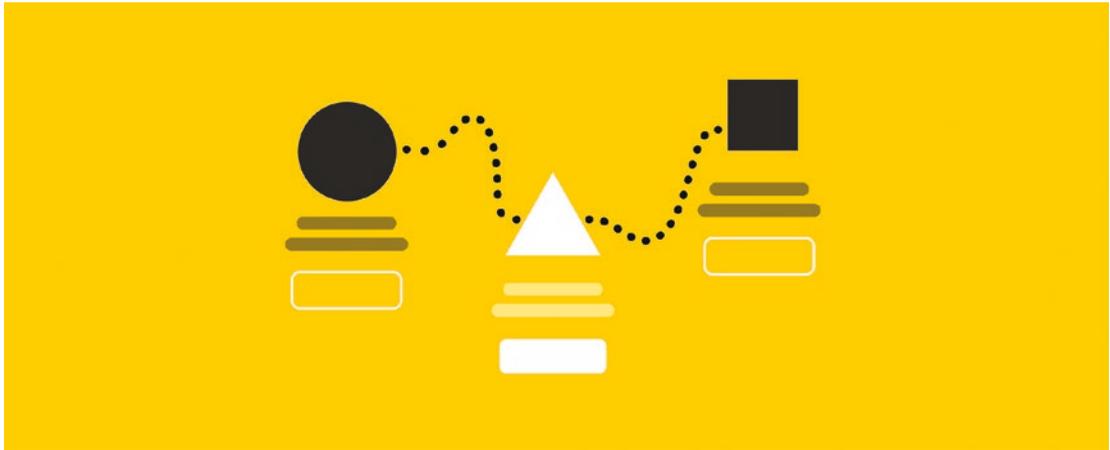


Figure 2-11. *Continuity Inception—the line connecting the elements helps signify their grouping, and the line itself isn’t a line. Continuity has us filling the gaps between the circles to perceive a line*

The path itself is also an example of continuity, with the brain perceiving the separate shapes of this path as a continuous line.

Common Fate

The Principle of Common Fate shows that items that move along a similar or shared path are seen as related—an insight that should heavily inform our animation design decisions. A remarkable example of common fate in nature can be observed in the flocking of birds, particularly that of starlings, turning and twisting in unison—appearing as a single unit. Common fate becomes exceptionally useful when we need to group nested items inside an already similar group. In situations like this, where similarity and proximity are important to maintain, we can use common fate to move one group of items along one path and another along a different path. This can be seen in Figure 2-12.

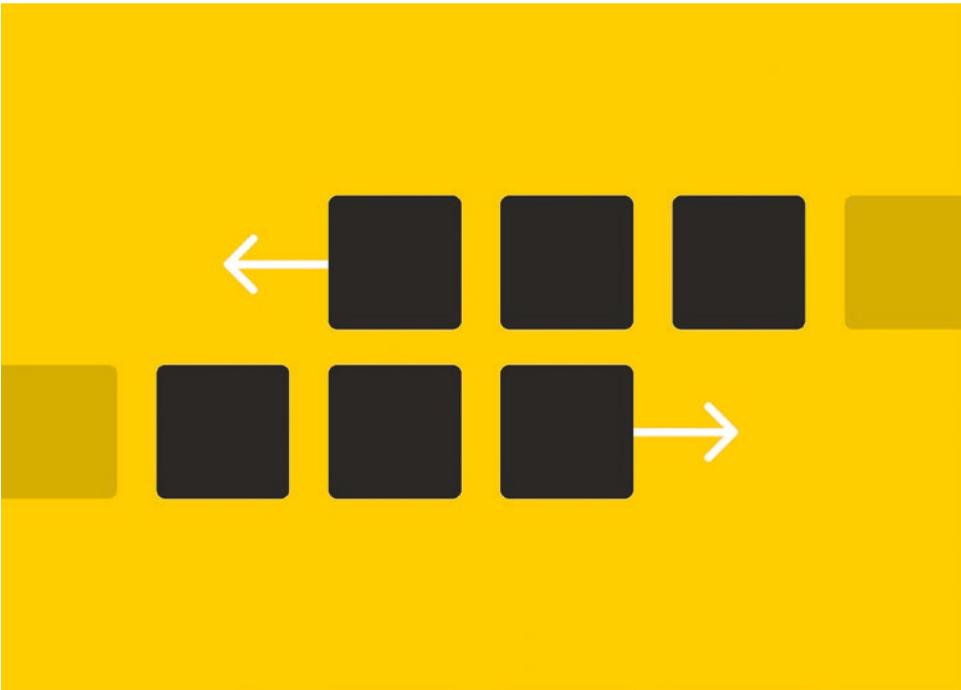


Figure 2-12. *Common Fate will help separate the top row from the bottom, even though it's highly likely they'd be perceived as a cohesive group without their opposing movement directions*

In Figure 2-12, the first row of elements and the second row are visually similar and grouped close together in order to appear connected, yet the different movement of the rows—imagine the top row animating in from right to left, the bottom from left to right—communicates a further intergroup relationship. This approach works really well in indicating a subtle hierarchy among elements of the same type.

Closure

Closure is an extremely interesting perceptual trait and used extensively and cleverly in logo and icon design. The Principle of Closure shows that we'll "fill in" forms that don't actually exist—provided there's a "hint" to that form. In Figure 2-13, it's likely you're able to perceive the "S" in the word "closure" and the diamond, square, and triangles in the shapes beneath it, even though those shapes don't actually exist.

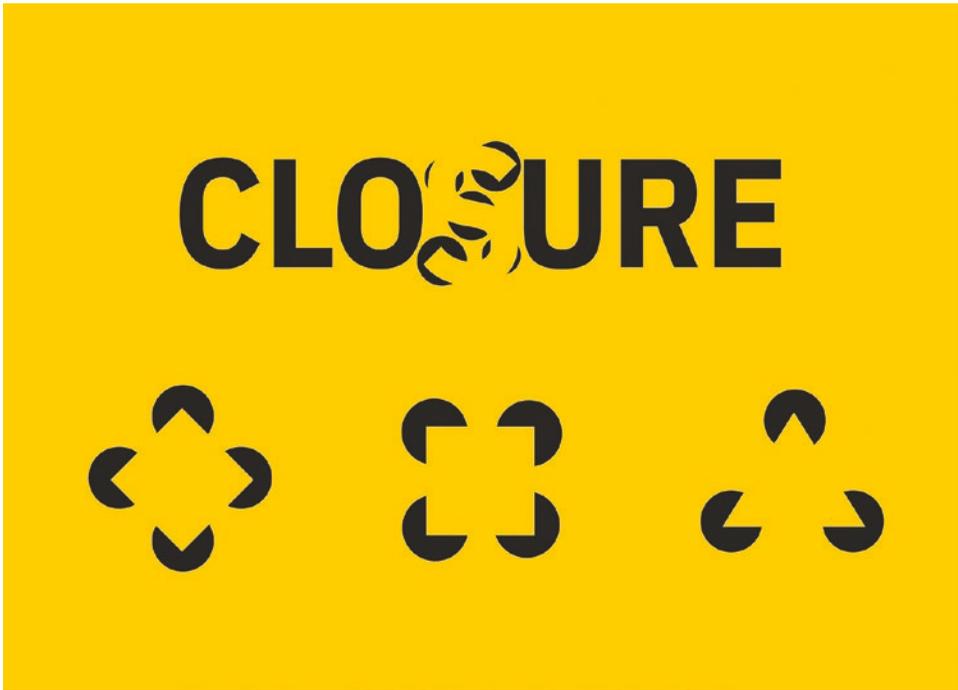


Figure 2-13. Closure allows us to perceive forms that don't actually exist

We have little use for closure in interface design, but it's extremely useful in iconography, allowing a form to be communicated using negative space and a much lower visual complexity.

Common Region

While the Principle of Common Region isn't part of the "classic" principles of Gestalt, it's a concept of perceptual grouping that is used extensively in visual design. Proposed by Stephen E. Palmer in 1992, the Common Region Principle states that, quite simply, elements that are contained within the same region are seen as grouped. Figure 2-14 shows the same grid of shapes as we saw in Figure 2-7 when discussing similarity, but with alternating rows contained within white rectangles.

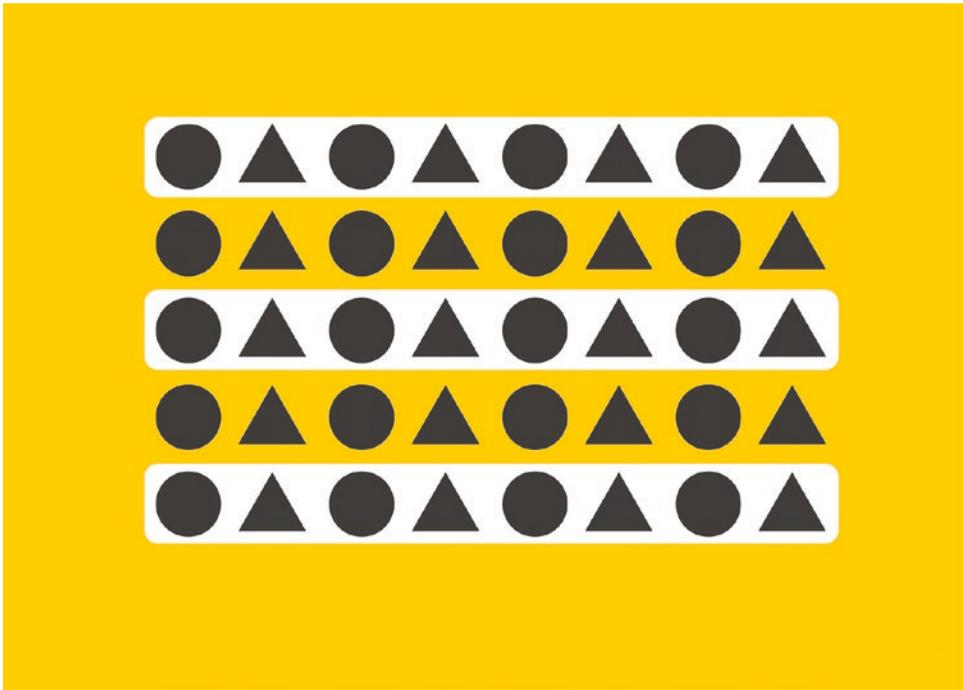


Figure 2-14. *Common region allows for very explicit groupings*

Just as we saw when the rows' shapes' colors were changed, the common region appears to “override” the subtler similarity and quite clearly communicates a row-based organization.

Putting It All Together

The Gestalt principles are rarely used in isolation, and while the examples provided are quite cut-and-dry, they're certainly not binary rules, with the only potential outcomes being grouped or ungrouped elements. We'll almost always “double-up” on our use of these principles. For example, putting similar-looking elements closer to one another in the same bounding box and aligned across the horizontal axis makes use of many of these principles in one layout section. Furthermore, we'll rarely have a single level of hierarchy to communicate—dense interfaces are likely to feature many groups which are, themselves, comprised of smaller groups, and so on. Fortunately, these principles have been shown to be impressively scalable and can explain groupings far more complex than we're accustomed to showing, or seeing, in even the most complex interfaces.

Like most of the concepts we've explored in this book, Gestalt—along with the broader, expanding principles of visual field organization—is a tool of cognitive economy, and it further shows how important design can be as a means of preemptively doing some of the brain's processing for it. By understanding how the human mind perceives items to be grouped or distinct, we can ensure that we effectively and intuitively position and move our interface elements to communicate hierarchy and reinforce relationships.

The Self-fulfilling Myth of Color Psychology

Depending on your focus as a designer, you may have been exposed, to varying degrees, to the concept of “color psychology.” Perhaps you have been introduced to the idea that colors themselves can make people feel and think in certain ways. Green and blue are “calming,” red is “exciting” and “edgy,” and orange is “creative.” The misconception that generally gets reported here is that these associated emotions are somehow emotional “responses” to that color. This is generally not the case. While these colors can hint at specific moods, they do not themselves possess notable degrees of emotional proclivity. In reality, we'd be pretty useless as a species if our emotional stability was so innately tied to something as ubiquitous and varied as color.

As with most design tools at our disposal, the reality of our emotional attachment to color is far more complex and far more reliant on context and culture than anything evolutionarily innate or neurological. We *are* evolutionarily inclined to respond to “contrast” of many kinds, something that color can have a profound impact on, but the notion that certain colors can make us feel certain emotions is overplayed and overhyped.

Red is probably the ultimate example of this. There are more studies on red than the rest of the primary colors combined. This isn't surprising. Red is interesting in that of all studied mammals, only primates are able to see it; it's not farfetched to suggest this is or was an evolutionary advantage. Red is the subject of many color psychology theories, including the notions that we are more inclined to notice red objects because it's the color of our own blood, that people are seen as more sexually attractive when wearing red, and that we've evolved to notice it as it signified ripe fruit among green foliage, and myriad other notions, varying from the completely sensible to the ridiculously absurd.

The thing with the color red is that very few things in terrestrial nature are naturally red, and a good deal of those that are—such as ripe tomatoes or chili peppers—aren't actually red for their entire lifespans. Contrast this with green. While we're currently, as a

species, seemingly on a mission to industrialize and deforest green into extinction—it’s “everywhere” in our natural landscapes. As is brown, and gray, and blue. Red stands out against the average palette of the natural world. It’s far more likely that we notice red simply because it’s noticeable than to assume we have an innate, Freudian, hyper-sexualized desire or necessity to seek the color out. In fact, due to the Von Restorff Effect, we don’t *need* to seek such a naturally contrasting color out—in most cases, we simply cannot ignore it.

The von Restorff Effect

The von Restorff Effect explains what I believe to be one of the key underlying principles of visual design: the technical conclusion of everything we do boils down to accommodating *contrast*. Contrast is how we communicate importance. Regardless if we’re assessing colors, layout, typography, animation, language, loudness, intensity, shape, or any other tool of our trade—effectively making important things stand out at the right time and to the right degree constitutes the practical layer of design. We discussed in the previous chapter how change and importance are huge factors when it comes to the subject of our attentional focus, and we primarily communicate this through contrast (Figure 2-15).



Figure 2-15. *The von Restorff Effect explains why the three red chilis may be instantly more recognizable and memorable*

The von Restorff Effect, also known as the “isolation effect” and discovered by Hedwig von Restorff, is a visual bias theory that suggests when presented with various similar items, the most different item will be easier to notice and recall. The counterpoint to this is that the homogeneous elements become less memorable, and perceived as less important, in the presence of the isolation effect’s trigger. At a glance at the photograph in Figure 2-15, you probably very quickly noticed the red chilis. There’s also a good chance that, if you were to look away, you’d remember their features to a notable level of detail. However, the green chilis likely almost blurred into one another even though they’re as varied in shape as the two red ones. In the presence of much more contrasting stimuli, they’re grouped, homogenized, and disregarded. This bias is probably the component of cognitive psychology that has the universal effect on understanding the technicalities of visual design, so much so that its ubiquity makes it essentially common sense: when something stands out from its peers, we remember it better, and we deem it more important.

If we view contrast as one of the fundamental conceptual elements of design, the von Restorff Effect, along with Gestalt, forms our psychological primer into making informed visual design decisions. Just like Gestalt, the isolation effect only “works” in an environment of limited stimuli. If we look at the dozens of ways we can make an element contrast with its neighbors and surroundings—vividness of color, size, spacing, framing, motion, and countless others—it’s pretty intuitive that the “busier” an element’s surroundings are, the more work we have to do to make it stand out and the more cognitive effort we require from people to process that information and infer the importance we’re trying to imply.

What the von Restorff Effect tells us, above all else, is that it is contrast that grabs our attention. Red objects are usually the most vivid objects in our surroundings—they stand out *because they stand out*, and much of the “emotion” we believe we apply to this color is, in fact, due to our own biases, associations, and cultural conventions. Color itself does not possess some mystical, emotional power outside of internalized association and preference. We largely create our own associations, and these associations can be learned and unlearned, or even rewired entirely, depending on context, content, and exposure to a convention.

Codifying Color

We’ll explore the ideas of communicating and introducing learnable UI concepts later in this book, but something to take away now is that we can, to a degree, introduce and communicate the use and importance of any interface paradigm

(color, depth, typography, etc.) to someone in a way that “marks” that paradigm as meaning a specific thing, or conveying a specific use. Video games are wonderful at this, using abstract and idiosyncratic icons to represent complex tasks or concepts. One game that does this wonderfully with color alone is *Horizon: Zero Dawn* in its frankly masterful use of the color yellow.

Toward the start of *Horizon*, you’re presented with some pretty standard “first few missions as onboarding” stuff (if you’re not a gamer, don’t worry, this concept will be explored in much more detail throughout this book)—designed to help you get a grip on the game’s mechanics, idiosyncrasies, and interface. At some point, you’re shown a ladder that you can climb. This ladder is bright yellow and contrasts notably with the natural, earthy, and lush palette of *Horizon*’s environments. The vividness of the ladder’s color and the fact that it’s isolated from its surroundings makes it stand out—and as part of our onboarding into the game and the affordances we associate with the ladder (we know, from our exposure to the world, that ladders, rope, etc., are all members of the functional category “things we can climb”), we’re taught, quite effectively, that we can climb this. Figure 2-16 shows how such a contrasting color can signify meaning.



Figure 2-16. *Horizon: Zero Dawn* (Sony Interactive Entertainment America LLC, Guerrilla B.V.) using yellow as a signifier, allowing us to attach meaning and causality to the color

Later on, as the game opens up and becomes less linear (“open world” games generally allow for unrestricted exploration of an expansive map), you start to notice little yellow notches in mountains and ledges tailed in scuffed yellow paint. Of all there is to love about *Horizon*, for it’s a wonderful game brimming with small details to appreciate, this is the thing that really stuck with me. From that initial onboarding experience I had, I was able to deduce that yellow means climbable. And every time I tried it, and it worked, I felt good as heck. By planting that seed in my head and “marking” things quite naturally—you’re never presented with a big you-can-climb-this-you-know? tip or anything that runs the risk of ripping you out of the fantasy world you’re in—*Horizon* was able to present to me an environment that felt truly empowering to explore. Exploration in *Horizon* is a large part of the game’s value, and its map is vertically expansive. From underground ruins to sky-scraping mountain peaks, you’re able to scale and submerge to a wondrous variety of heights. All of which means, really, you do a heck of a lot of climbing in the average play-through of *Horizon*. By refusing to patronize its audience, yet still clearly communicating what is and isn’t climbable, the game offers a masterful insight into the human mind’s ability to catalog and codify concepts together with simple perceptual cues. Without ever being told it, we intuit that yellow means climbable, and this is reinforced every time we try it and successfully prove our intuition correct. We’re never patronized by over explanation. It’s assumed we’ll understand their inference, and it *feels* like we’re making our own discoveries, with just a nudge here and there from the game’s UI and design decisions.

To me, this experience, and countless others like it, embodies a very important concept when it comes to color: it’s almost useless without context. In seeing something that contrasts with its environment and being able to interact with an object of that color in a specific and consistent way, we’re able to manufacture our own, very specific, affordances. In “this” environment, in “this” situation, “this” aesthetic communicates “this” effect. Importantly, we generally codify that aesthetic-effect relationship within the confines of that environment. I didn’t take my learning from *Horizon* and apply it to different environments; the “climbable” association I had with yellow in the game didn’t translate outside of its bounds, and I never encountered any embarrassing situations where I was compelled to acrobatically scale every McDonald’s sign I saw. I was simply shown a relationship between aesthetics and effect and codified that relationship in that distinct environment. One of our most intriguing and powerful heuristics is our

willingness to accept a simpler signal as a proxy for a more complex one. This can quite often have us falling short on the more philosophical or contemplative issues in life, but as a tool of cognitive economy, our ability to associate a visceral stimulus with something as complex as a series of performable actions is pretty wonderful.

This idea is a huge part of why we shouldn't be imposing too many arbitrary limitations on ourselves when it comes to color or any tool in our kit. We know, through the isolation effect, that it is an element's overall contrast against its environment, not necessarily something as reductionist as its color, that makes it notable. We also know that, if done correctly, we can create causality and association with simple visual cues—allowing something as simple as a color to communicate something complex and enabling. All of this points to the notion that color is simply part of a much more complex and contextual ecosystem of design decisions. Without context, color is a lot less powerful than we've traditionally been taught to believe. It's the associations and attachments we have and develop toward colors, along with their role in the contrast equation, that give them their power.

Back briefly to the subject of red. We're often told to stay away from it as a brand color or as a primary interface color due to its ubiquity as “the danger color.” This idea isn't unwarranted—we've manufactured a world where red means “stop,” and we've further propagated this through historic UI patterns such as red error states and red as a signifier for destructive functionality. However, as we explored above, context is so, so important. Such a sweeping and generalizing principle is useful simply as a random creative constraint. As far as we're aware, people aren't regularly frozen in front of Coca-Cola cans wondering what's so dangerous about them, people aren't regularly mistaking the YouTube play button for an error message and having a cognitive-dissonance-induced breakdown every time they try to watch a video online, and kids aren't scared to play with Legos because the logo reminds them of the crimson blood of corporal beings and their own tragic mortality.

In fact, YouTube's interface as a whole is a prime example of how red can be used effectively as a primary UI color while not harming usability or cognitive economy in any discernable way. Almost everything of importance on YouTube is red. It's the primary color of the logo, the primary CTA color for subscribe buttons, the color of the completed percentage of loading progress bars, the color of the elapsed time indicator of a video's play bar, and the color of the Play button itself for embedded videos. (See Figure 2-17.)

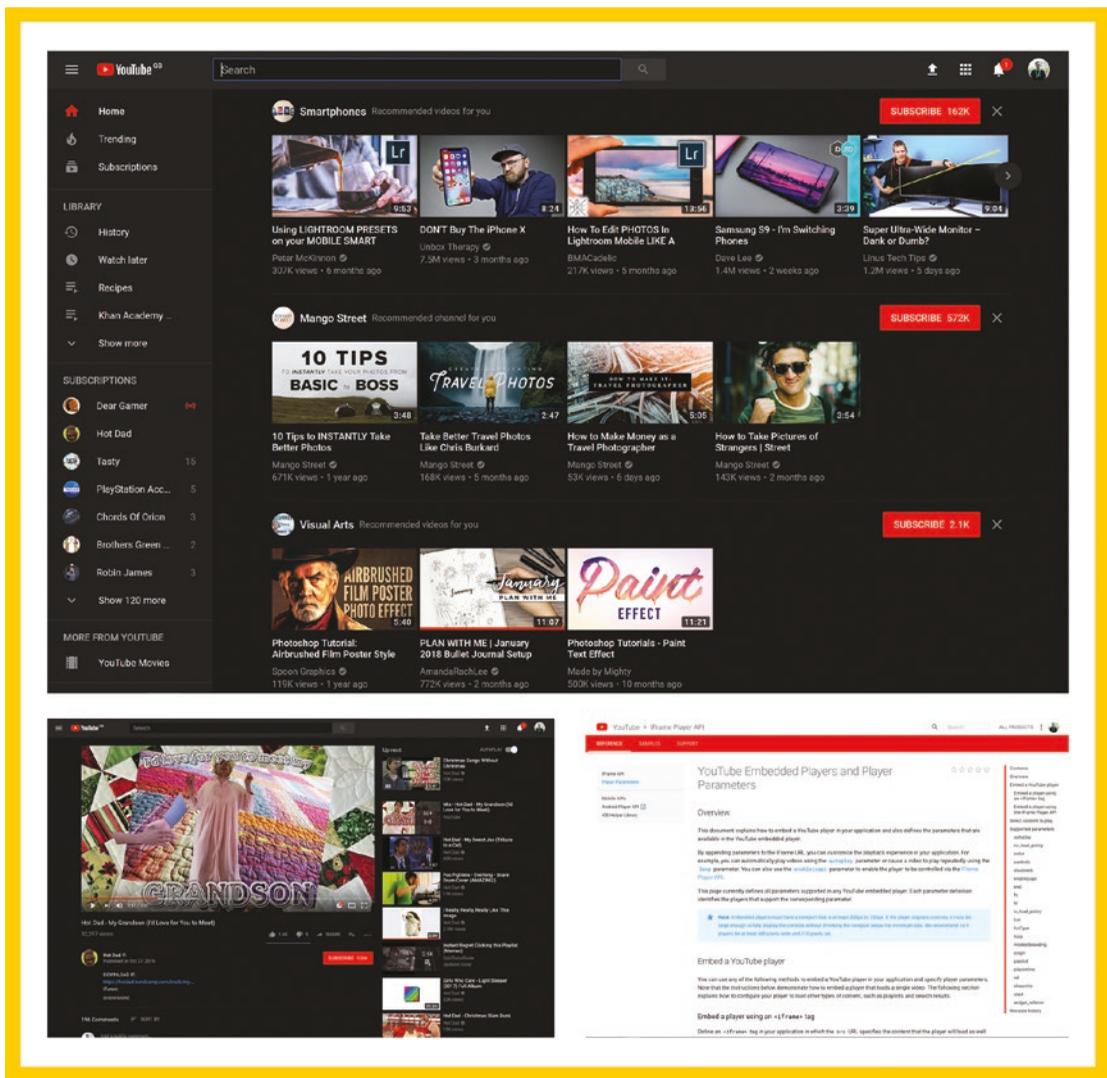


Figure 2-17. YouTube’s liberal use of red as a primary interface color

YouTube as a self-contained product has access to many billions of usage data points, which multiplies exponentially when you consider it’s a Google-owned product. That is not to say that Google, or any Google-owned product, isn’t capable of being poorly designed, but given the oft-reported notion that “CTA color” can have a radical impact on conversion rate, and the first-class-citizen status of YouTube in Google’s product ecosystem, I think it’s safe to assume that this idea has been tested rather extensively. “Stay away from red” is reductionist, prescriptive, and pointlessly constraining.

We need to take a few steps back and question the wisdom of our current accepted notions of primary interface colors. While the heuristic of “green is good, red is bad” is a useful shortcut and a tool that serves us extremely well anytime we have to provide visual feedback, I feel we can be a lot more explorative and much less prescriptive with these colors as part of our interface and branding. Having said that, we’ll explore a little later in this chapter how convention and paradigms are powerful, helpful notions in digital design, and color absolutely plays a part in this.

Finally, while colors’ ability to make us feel a certain way is hugely overplayed and a massive generalization, the idea that they communicate and reinforce certain characteristics is far from false. By marrying suitable color choices to a brand or interface that communicates specific characteristics through its other aesthetic attributes, we can really reinforce a consistent vibe. While this is far from a brand design book, and I am even further from a brand designer, part of an interface’s job is to act as an extension of a product’s brand, personality, and character—to carry on a conversation in a tone befitting of this character. The colors you choose for an interface, alongside the ever-changing cultural and contextual associations one might have with them, further juxtaposed with design elements (such as typeface selection and animation) that reinforce similar personality aspects can be a huge impact on the authenticity and perceived emotional message of a design and brand. This subject could quite easily be the basis of its own entire book. Just know, for now, that the sweeping statements you might be used to hearing regarding the psychology of colors likely range from oversimplified reductionism to outright myth.

Animation

Animation and digital interfaces have a long and rather strange relationship. Until relatively recently, our exposure to animations—particularly on the Web—was limited to unskippable Flash intros and attention-sapping banner advertisements. However, as the Web, our tools, and our devices and browsers have matured to allow for seamless, smooth, and performant animations, we’re witnessing the real boons of interface animations—both as a means of reinforcing personally and as a tool for reducing cognitive load.

With animation finally being viewed as a first-class citizen in our design processes (albeit at the time of writing this book, still far too often thrown in as an afterthought to an already-finished design) we’re quickly learning to embrace the impact it can have on our interfaces. Something to be wary of—an interaction design trope if you will—is

how often people use “delight” and “has some animation” interchangeably. Design has institutionalized delight as a currency in recent times and for some reason, a few quick bounces and a springy animation here and there is, apparently, all it takes to achieve that.

Nevertheless, animations that are consistent with the personality of the rest of your design—and used responsibly as a means of reducing the cognitive effort required to process an interface state—can turn a dry, minimalist interface into one that exudes charm and character. Just as with any other aesthetic concept, animation has many sides. There’s the functional, utility-focused idea of animation as a means of providing contrast; the notion of animation as a tool of effective cognitive economy and as a learning aid; and the more abstract, emotional side, focused on communicating character traits and eliciting an emotional response. Animation will be a key concern throughout Part Two of this book, where we delve into the practicalities of interaction and interface design, but for now, let’s take a quick look at each of these sides.

First, when discussing visual perception, animation is probably the single most-contrasting aspect of any interface. If something moves, especially against a still backdrop, it’s almost impossible for our brain to *not* register it as important. In many cases, our jobs revolve around communicating the importance of just a single element or group—say, a sign-up form on a landing page—and an extremely effective way to do this is to give it some form of animation. However, we need to be cognizant of the impact animation can have on our effective baseline of stimuli. Of all the tools at our disposal, animation is the most likely to bring us into the territory of “visual mush” (very scientific, I know). Animations need to *make sense* just as much as color and typography choices do, and it’s very easy to fall into the trap of animating for animation’s sake.

Note As you might be able to guess, communicating animations using only words and images is hard as heck. For convenience, any animations discussed will have links to example videos in their captions. You’re encouraged to check them out at your leisure.

The Common Fate Principle of Gestalt is probably the most useful insight we can apply to animation as a tool for grouping and cognitive economy. As we explored, items that move similarly to one another are viewed as grouped. We can take advantage of this to help reinforce the perception of grouped elements. For example, if we need to animate two separate blocks of content simultaneously (this happens often if we wish to communicate a change in state, for example, changing an active tab)—we can very

easily communicate this contextual separation by simply animating one block in from the left, and the other from the right. This is one of the most common and effective uses of common fate, and it allows for a more subtle or consistent static visual design within an interface that still effectively communicates its groupings—extremely useful for interfaces that might serve very similar content, separated only by conceptual categories, such as bloglike tools like Medium or Podcast browsers.

Furthermore, animation is extremely effective when used as a “signifier.” An element’s potential interactivity can be communicated through animation better than almost any visual cue we could provide. A super-common example of this approach is mobile apps that have horizontal scrolling areas (Figure 2-18). These apps will often have these scrollable elements begin at their “final” scroll position (usually all the way scrolled to the right) and then animate an auto-scroll to the “starting” position when the element is in view.

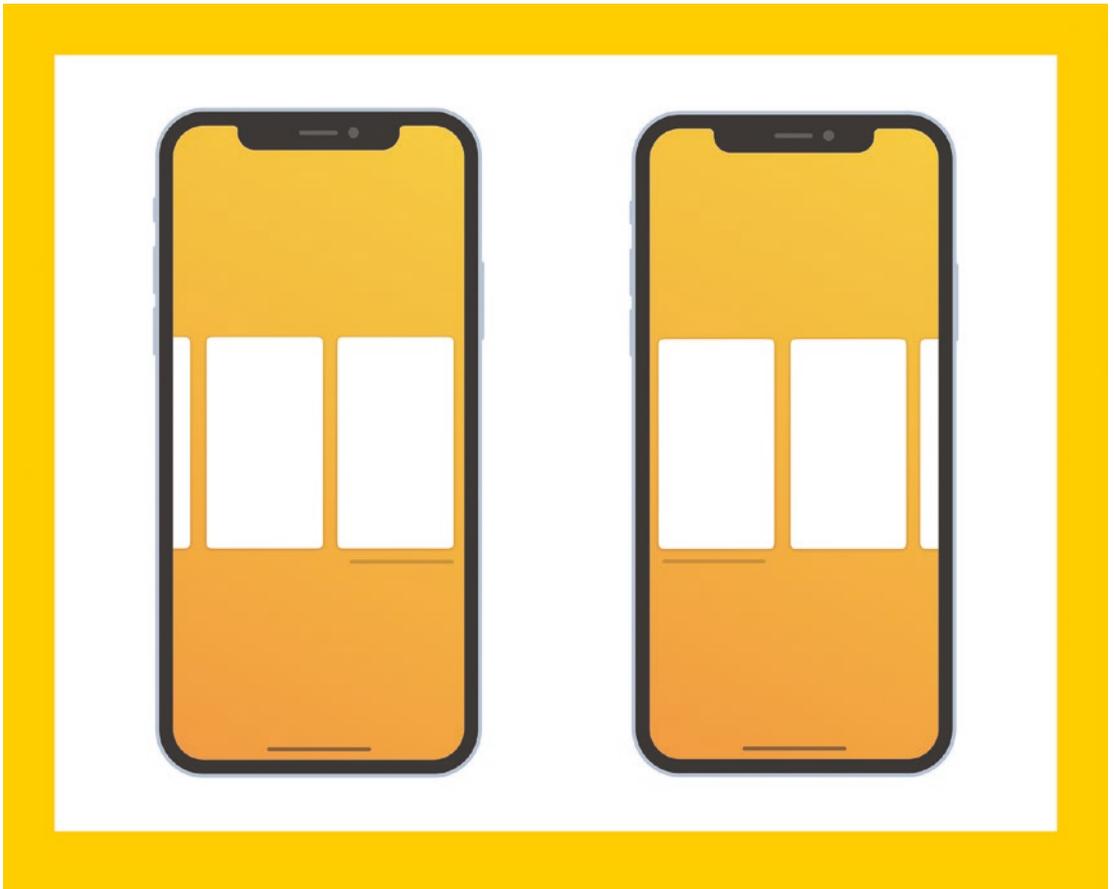


Figure 2-18. Start and end states of an animation that signifies an element can be scrolled Video: <https://vimeo.com/300085213>

This communicates quite emphatically the *scrollability* of the particular element and, while this is often reinforced with “peeking” (as seen in the “start” state), the animation itself probably offers the most cognitively available signifier possible. *Show, don't tell*, as the saying goes.

Animation itself can also be used to communicate a state, such as the iOS home screen reshuffling shown in Figure 2-19. All the icons on the screen appear to be energetically vibrating, hinting (albeit rather conspicuously) that, in this state, icons can be dragged around at will.

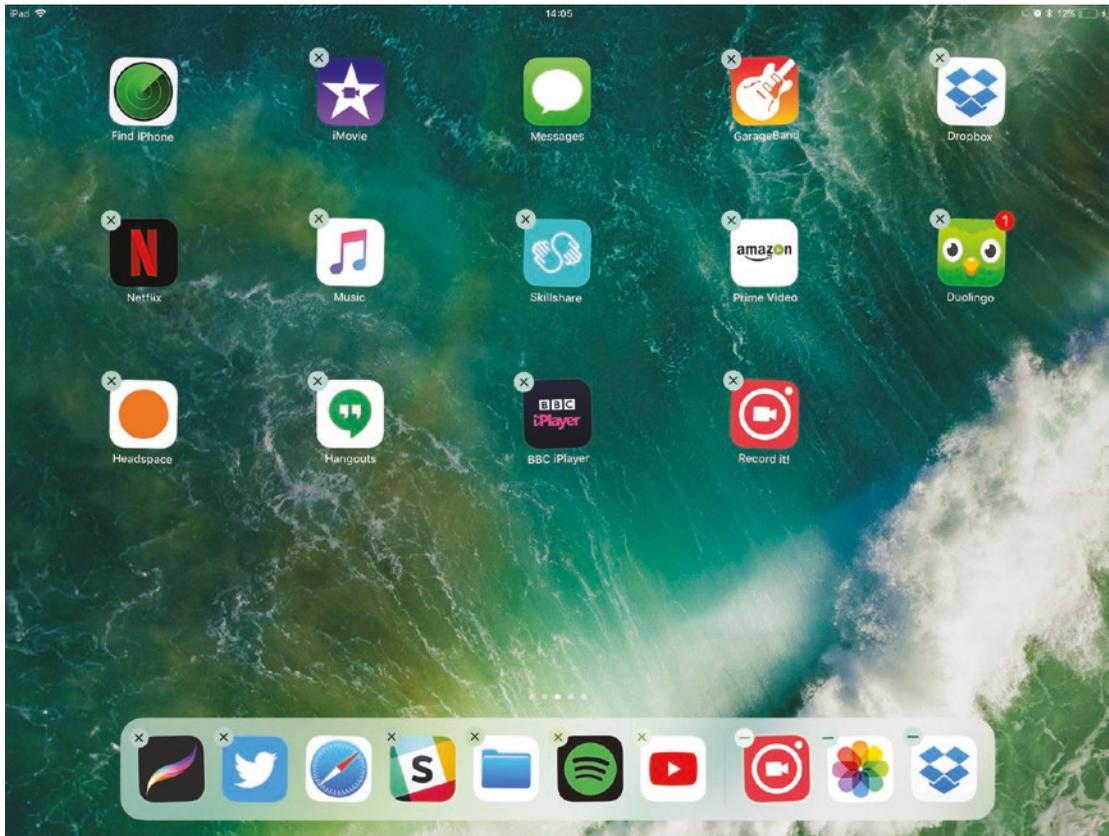


Figure 2-19. When icons are being reshuffled on an iOS home screen, they dance around, signifying they can be dragged. Video: vimeo.com/262019825

Animation plays a huge role in communicating feedback—which itself is a vital tool for effective learning and memorization. By definition, an interaction has some form of impact. It changes the application state in certain ways. An integral component of good interaction lies in effectively communicating this state change, and this is where animation can really shine.

Let's take a simple to-do application, with a list of current to-do items and a form at the top to add a new item (Figure 2-20). All our to-do items are visually similar because we're not sadists and we learned all about Gestalt, and our new to-do form is separated and ready for input. If we focus on the "create" interaction of this application, it takes us from one state (having our to-do typed in a form field) to another state (the to-do having been created, and added to the list).

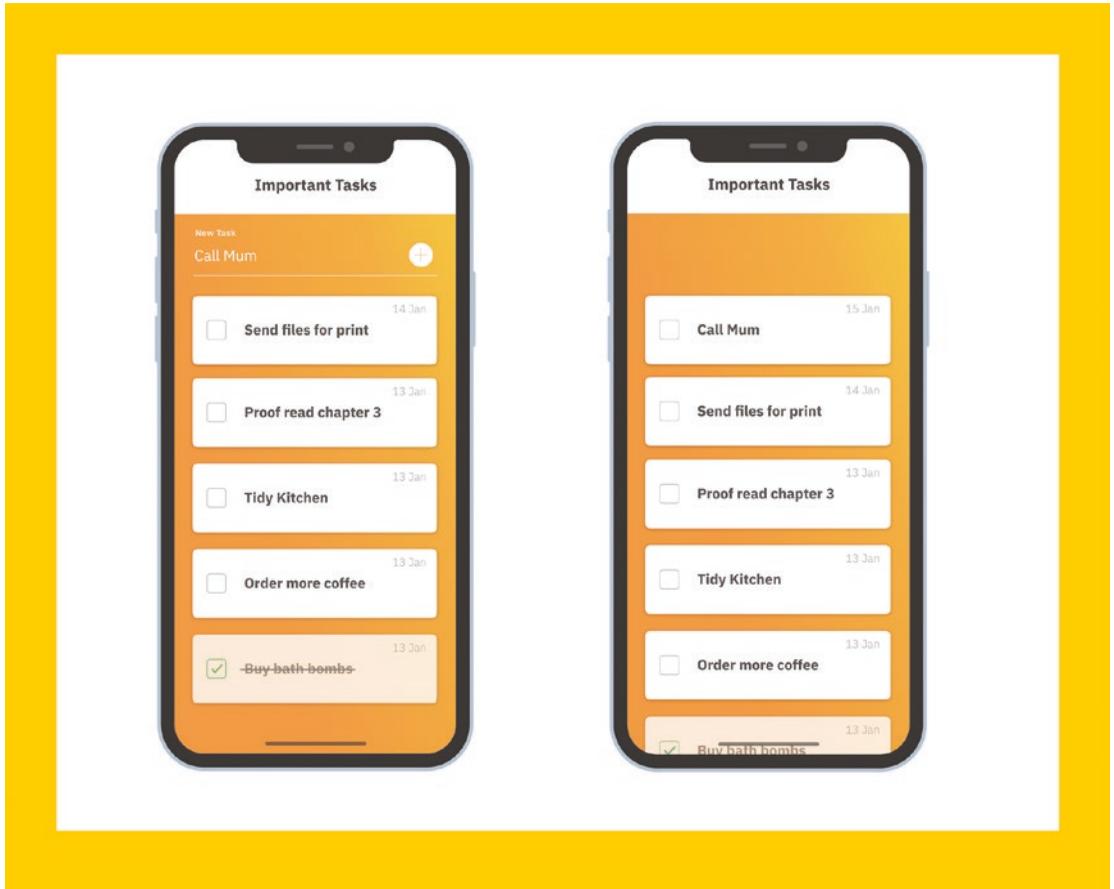


Figure 2-20. The “before” and “after” states of a simple to-do application. Notice how similar they look at first glance. Without any kind of animation to the “add” interaction, the success of the intended action is difficult to convey. Video: <https://vimeo.com/300087399>.

Because our items are visually similar and we're not separating new to-do items out in any way, our “after” state is extremely similar to our “before”—there's one extra item in a list of very similar items. However, if we were to animate the in-betweens of this

state transition, we can see just how integral animation is in these kinds of situations. By interpolating between the “before” and “after” states, we can effectively communicate that an interaction was successful. We can enhance this even further, if necessary, by animating in a simple “To-do added!” confirmation message, or by visually highlighting the new to-do for a short period of time. The key, however, is that we’re not left wondering if our intended interaction actually happened or not.

This is a very diluted example, of course, and most interfaces will likely be denser, with many more widespread changes in state to accommodate. In these cases, your animations will need to be far more complex in order to effectively interpolate between states. In Part Two of this book, we’ll delve deep into using concepts like delaying and staggering animations to avoid overburdening our perceptual and cognitive resources. Starting small and isolated like this, though, is a good way to ease us into the basic ideas.

Finally, there is an emotional side of animation. If we think of an interface partly as a conversation (regardless of whether it’s a trendy chatbot or not), then its animations can be seen as the subtle nods and hand gestures that psychologists know as “kinesics.” As emotional beings, we place a lot of value in nonverbal communication, almost always punctuating conversations with gesticulation. How animated our nonverbal gestures are provide subconscious insight to our current mood and our overall character—high-energy gestures can show we’re passionate, keen, and interested while more considered gestures can communicate grace, elegance, or stability.

The existence and sensibility of animations is only one side of their equation—their mannerisms and idiosyncrasies are the other. With digital interfaces, we generally have two options when it comes to controlling the characteristics of an animation. First, we have spring physics—which involves giving an element physical characteristics such as stiffness and damping, a start point, and an endpoint. Second, we have easing-based animations—like those found in CSS—where a duration is set and a bezier curve is used to describe an element’s acceleration and deceleration. Both of these options allow for great control over the feel of an animation. Generally speaking, a spring with more stiffness and bounce or an easing curve with more elasticity and overshoot will feel friendlier and more energetic, while a tighter spring or a subtler easing curve would be better suited to more elegant, formal tones. Think of these parameters as ways to control the character of your animations, but be wary of going too extreme on either end of the spectrum. Too loose and bouncy and your animations can come across as cheesy, annoying, or erratic. Too subtle or linear and they can appear bland or go unnoticed.

One final note about interaction animations—and this is pretty universal—they *must* be interruptible. Being locked into an animation, even if it’s less than a second long, is

a guaranteed cause of disruption. If you're animating a change in state and that state is reverted (for example, through an “undo” action) or changed in any way “before” the animation completes, then an interface needs to respond to this as and when it happens. Far too many interfaces use this naive way of animating transitions, turning what should be an informative and effective communication of change into a confusing mess of stacked animations and delayed responses.

Limitations of Contrast

In exploring the notion of contrast as a means of communicating importance, I hope it's somewhat clear that often the easiest way to “add” contrast to an element is to reduce the perceived importance of its surrounding elements. Just as “attention-grabbing” is best done to the backdrop of a harmonious and considered environment, so too must our various applications of contrast occur in an environment as free from chaos as is necessary. There exists a very real situational and individual threshold for contrast to be noticed, and a curve of rapidly diminishing returns when we try to make something stand out in a sea of increasing chaos. The counterpoint to this, however, is that there exists a just-as-important threshold of mundanity and homogeneity that, if dropped below, runs the risk of an interface that is just too outright dull to perceive, and thusly so. If we default too far toward the stoic, we're neglecting people almost as much as if we veer into stimuli-abundant absurdity. The good news is that the middle ground between these two thresholds is actually incredibly vast. There's a lot of wiggle room in design, and what is “dense” or “overburdening” is as much dependent on situation, context, and culture as it is anything else.

The science of—and abundance of theories surrounding—contrast is infuriatingly complex. It's one of those phenomena that we understand conceptually and qualitatively a lot more readily than we do explicitly or quantitatively. That is to say, it's much easier to note that “contrast describes the perceived prominence of one or more elements in relation to their surrounding environment” than it is to actually quantify the *degree* of contrast those elements might possess. Various attempts have been made to provide contrast equations and standards, with the WCAG's exhaustive and well-documented list of acceptable contrast ratios for accessibility probably the most well known to us as digital designers. While this kind of element-specific approach to contrast, alongside formulating integral rules and guidelines for your interface and style guide's approach

to visual contrast, is an important part of any design process, the role of contrast in our initial perception of an environment is conceptually a very different practice.

Contrast—as a means of communicating importance—and change—as a means of feedback and state communication—are two of the most integral considerations in our interface design process. The Weber-Fechner Law tells us something that I feel is already pretty intuitive: the more stimuli present in an environment, the more impactful any change must be in order for it to be perceived. This law (more accurately, a combination of two interrelated laws) can be applied to many aspects of human sense and perception, and it suggests that our threshold for perceived change is logarithmic, not linear.

Figure 2-21 shows this in action. It's quite clear that there are fewer shapes in the top left image (five) than there are in the bottom left (10). However, with a much more cluttered group of shapes, such as the 50 in the top right, the same increase of 5 in the bottom right is barely, if at all, perceptible.

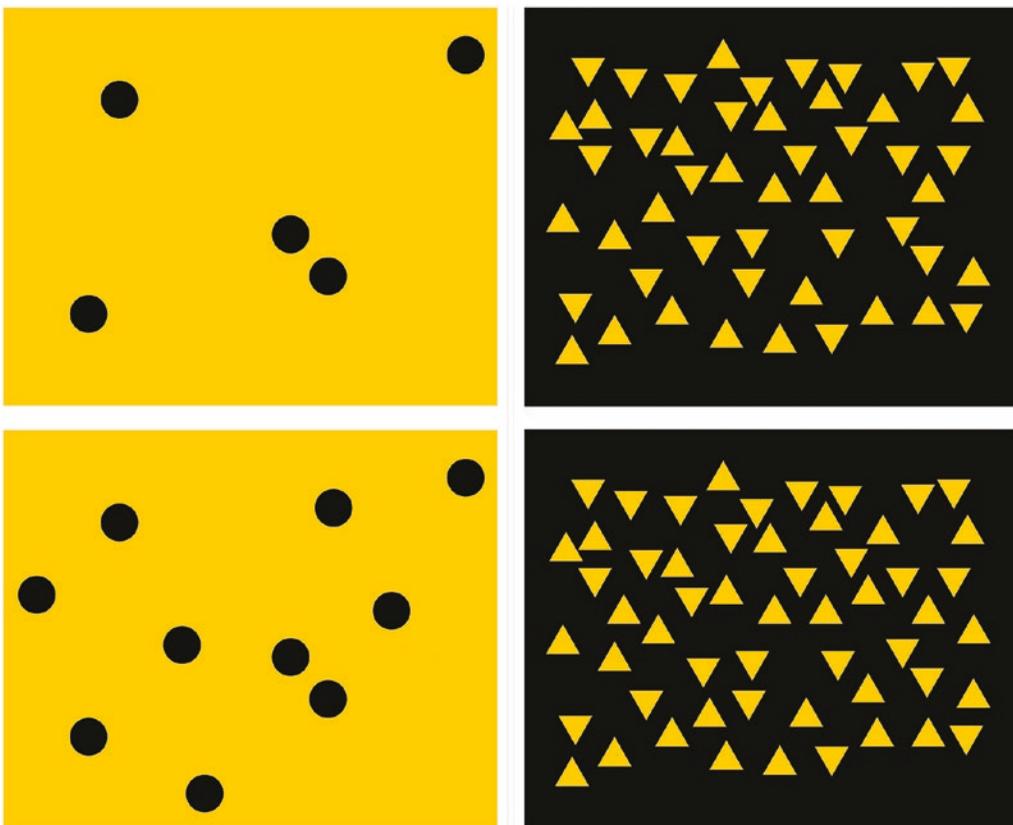


Figure 2-21. *The number of shapes changing is obvious between the left groups of shapes, but may not even be perceptible between the right groups*

This simple concept of perception portrays just how important a stable, calm, and distraction-free interface can truly be. By taking a minimalist approach to the number of stimuli present at any one time, we're able to introduce new concepts and convey importance with much more ease than we could in a poorly designed, distraction-riddled interface. Of course, every interface and product out there has its own baseline at which it can potentially operate—the magnitude of which is vastly dependent on the depth and complexity of features, the personality and tone it's trying to portray, and even the device and external environment in which it's used. A complex project management application is going to be inherently more cluttered than your average weather app, and in many cases, a vibrant, friendly tone will beget brighter colors and more boisterous animations than a stoic, elegant interface. Minimalism is only effective to the point where it doesn't negatively impact perceptual clarity or personality and the emotional, conversational aspects of our work.

Something we touched on in Chapter 1 and something that will be prevalent in many areas of this book is that it's imperative to understand that when something is brought to our attention or portrayed as important, if it's *not* deemed important by the observer, we run the risk of causing large amounts of cognitive dissonance. This is partly why modern online advertisements are so egregiously annoying. They bounce and auto-play and shout and trample all over an interface—and they almost always mean nothing to us. Even in interfaces that don't monetize attention through advertisements, we're likely to encounter various kinds of upsells and in-app purchases. An attention-grabbing stimulus that is not intrinsically important to the person who perceives it is, quite simply and by definition, a distraction. The more of these distractions you have, the more you dilute your interface and the more you run the risk of your actual important elements being missed due to your living on the tail end of the Weber-Fechner Curve. Again, the middle ground between stark and banal and stimuli-overload is pretty large—there is absolutely room for personality and vibrancy in our work. There's even room for *tasteful* advertisements, should they be required! Just know that any element you add to an interface performs two detrimental actions. First, it reduces the importance of any homogeneous or similar elements in its vicinity. Second, it increases the density of stimuli in your work. Both of these actions result in an increased cost of processing for the perceiver and more work for you when deciding how to convey the importance of other items going forward. This is the cost/benefit analysis we make as designers when it comes to presenting features and content. It's far from exciting, but a lack of consideration here can pointlessly constrain us further in our process.

Convention, Signifiers, and Metaphor

The concept of “affordance” is something that will inevitably arise when discussing the patterns and conventions of digital interfaces. I’m not going to dwell on the word for long—not because I want to be controversial, but because, really, almost all the times we talk about affordance in interface design—the idea that “form implies function”—we’re really talking about signifiers, metaphors or some form of convention. “Affordances” are often suggested to be our connection to our environment through the assimilation of possible action. The two classic examples that we wheel out to elucidate this concept are, first, the idea that we intuit whether a door should be pushed or pulled based on the physical traits of its handle, and second, that the handle of a mug or teapot implicitly communicates “this is the part you should hold.” In the case of the door, it’s posited that a vertical, pole-like handle “affords” pulling, while a flat panel affords pushing. In the teapot or mug example, the handle affords holding because it is a hand-shaped hole.

Throughout his career, James J Gibson proposed and refined a remarkable theory of ecological psychology. Dubbed “the Affordance Theory,” Gibson posited that, in perceiving our environment, we react instinctually and immediately to objects that suggest the possibility of action. When discussing his definition of affordance, Gibson suggests that “it implies the complementarity of the animal and the environment.” Probably the seminal text on design’s understanding and assumed implementation of affordances is found in Donald Norman’s *The Design Of Everyday Things*. Since its initial publication Norman has acknowledged, in a 1999 article for *Interaction*, that his definition of affordances was restrictive—suggesting the term “perceived affordances” as a more suitable term. Regardless of the dissonance between Norman’s appropriative definition and “traditional” Gibsonian affordances, these “perceived affordances” are the ones apparently most useful and available to us in our pursuit of intuition in design. Norman suggests that, when discussing affordances, “the designer cares more about what actions the user perceives to be possible than what is true.”

More importantly, however, Norman suggests that we’ve all really got the wrong end of the stick, and almost everything we categorize as an affordance on screen is either convention or metaphor. If this is all somewhat confusing and dramatic to you, don’t worry, you’re not alone; multiple papers and articles have been published attempting to settle this affordance debate. Fortunately for us, the fact we’ve embraced an incorrect definition of affordance means we can dodge the psychology community in-fighting altogether and focus on convention and metaphor.

Whenever we introduce an element to an interface, one of the core considerations we must make is how this element can communicate any interactive properties it may possess. How do we differentiate something that can be clicked or tapped from something that can be dragged? Going further, if our tappable element’s “tappability” depends on a particular application state—say, a form’s submit button being disabled until the form is valid—how do we communicate the (actually really complex) idea that “this item is tappable, but not yet”?

While we often attribute these inferences to Norman’s perceived affordances, usually these attributes are more likely implied through convention and metaphor. Convention has us looking at underlined, differently colored text amid a paragraph (Figure 2-22) and accepting it’s a link.

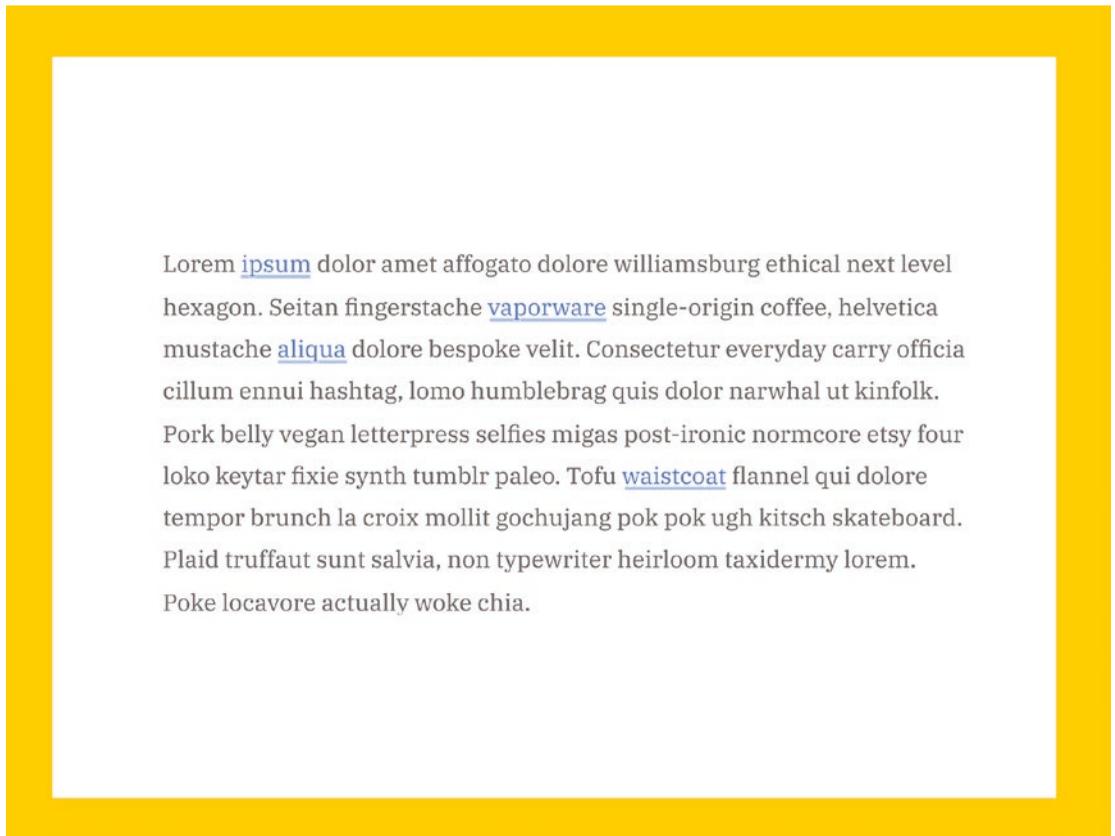
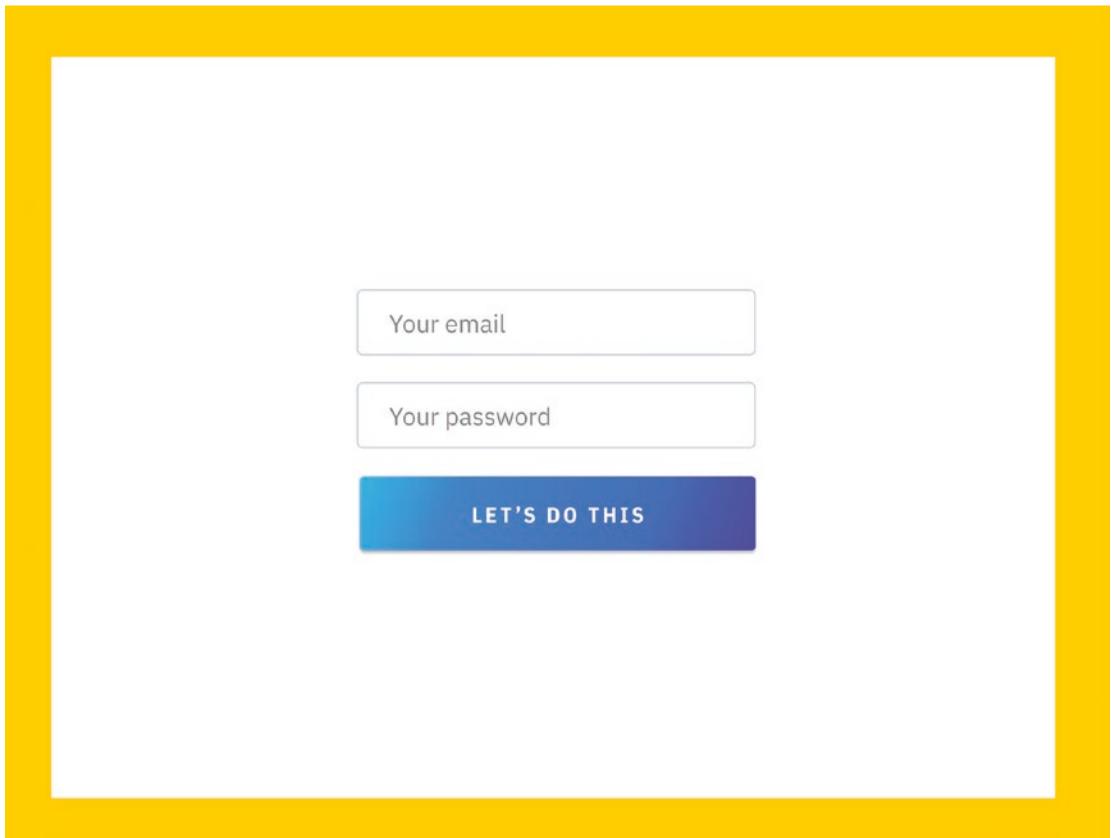


Figure 2-22. *Convention*

Blue, underlined links within paragraphs of text are one of the oldest examples of convention on the Web. As web design has evolved, the convention’s “rules” have shifted slightly—blue isn’t always used, nor is a solid underline—we generally perceive “stylistically different to the body text” as actionable.

Metaphor has us borrowing the real-world physical qualities of an object and replicating them in our interfaces—the classic example of early interface buttons mimicking the raised, enclosed appearance of physical buttons (Figure 2-23).



Figures 2-23. *Metaphor*

Button styles can make heavy use of metaphor, with early software buttons going to great lengths (at the time) to mimic the raised appearances of real-world buttons. Similar to the paradigm of text links, button styling has evolved and our rules for perceiving

something as a button have blurred over time, yet we often still make heavy use of real-world metaphor when designing buttons and their various states.

Both of these concepts combine when we look at the common disabled state of an interface button. It has a degree of visual similarity to a physical button, and it uses the conventional approach of dimming an element (Figure 2-24) to denote a “not-quite-yet” state of interaction.

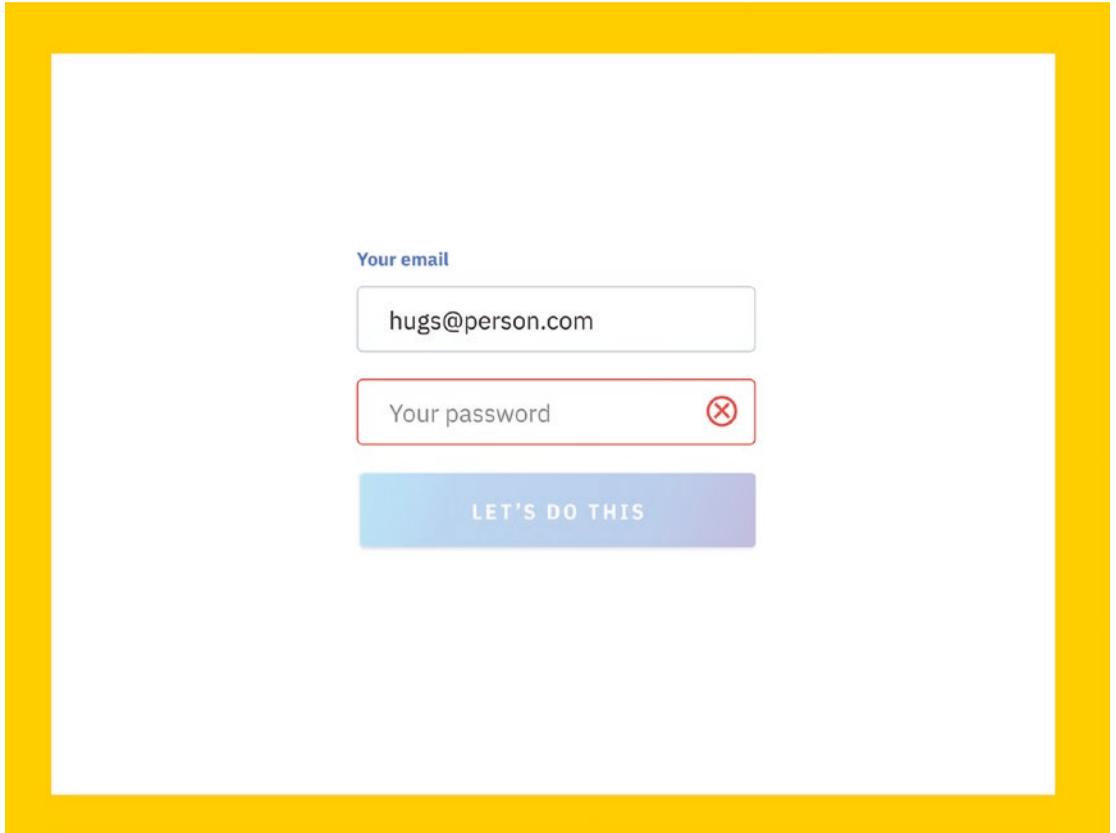


Figure 2-24. This button combines metaphor (the real-world representation of a pressable button) with convention (the accepted convention of “dimming: an element when it is disabled”) to communicate its properties (pressable) and state (currently disabled)

One thing that I do think flies under the radar a little in the quest for metaphorically sound interfaces is the idea of creatively compensating for lost affordances. By existing in the digital space, being interacted with on a device that handles the “real” affordances for us (a desktop computer translating keyboard and mouse input on-screen, for example), our interfaces lack a lot of the implicit physical affordances of the real world. You’re holding a ready-made example of this right now. If you’re reading the printed version of this book, you know roughly how many more pages you have left to struggle through. The weight of the unread pages gives a tacit implication as to how many you’ve made it through, and you’re able to compare the physical size of the stacks of read and unread pages. In the digital world, especially if we’re designing on top of abstracted layers such as browsers and operating systems, we lack the ability to effectively simulate this. If you’re reading the digital version of this book, you might be able to see the attempt to communicate this in the digital world. Certain e-readers allow for the enabling and disabling of progress indicators—a very digital convention—to communicate your percentage through this wondrous reading journey (Figure 2-25). This is a prime example of convention trumping metaphor.

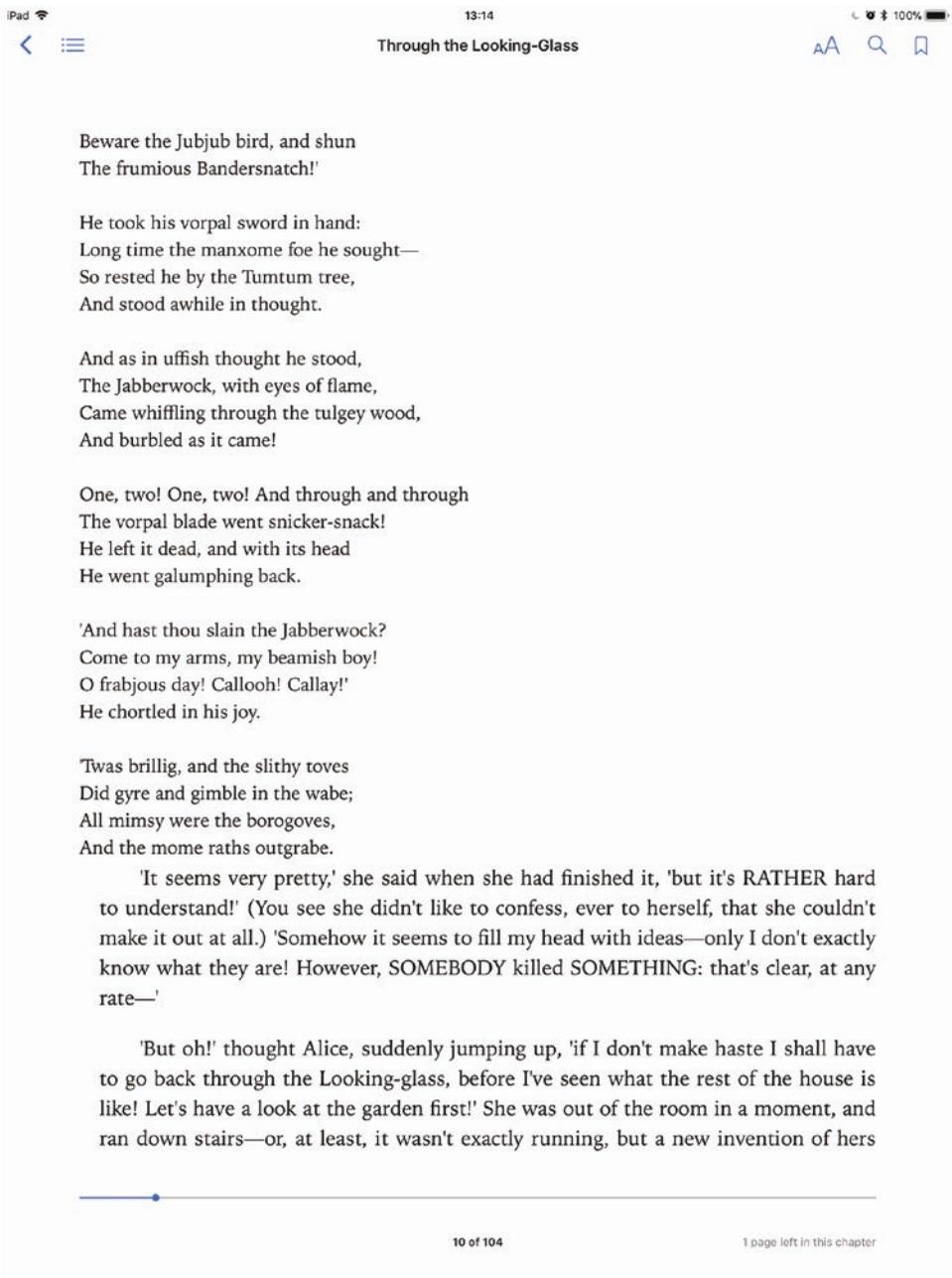


Figure 2-25. *The progress indicator signifier of digital e-readers replaces the tacit real-world affordance of using a book's physical properties as an indicator of completion*

Fortunately, design has mostly moved away from naive attempts at digital verisimilitude, accepting more and more that we can utilize—and even create—conventions that are unique and productive within the environment of digital interfaces. Through a rather natural, yet admittedly painfully slow (in technology terms) process of filtration, we harness a form of distributed cognition that allows for convention and pattern to be assimilated and subsequently embraced or rejected between large networks of humans. We have digitally native patterns and conventions of communicating and framing interactivity, indicating progress, intuiting states, and myriad other needs of a modern interface. While extensively cataloguing these various components is an arduous task and not this book’s purpose, I implore you to browse through some of the robust and inspiring design systems of products such as Shopify, Atlassian, Microsoft, and Salesforce (Figure 2-26).

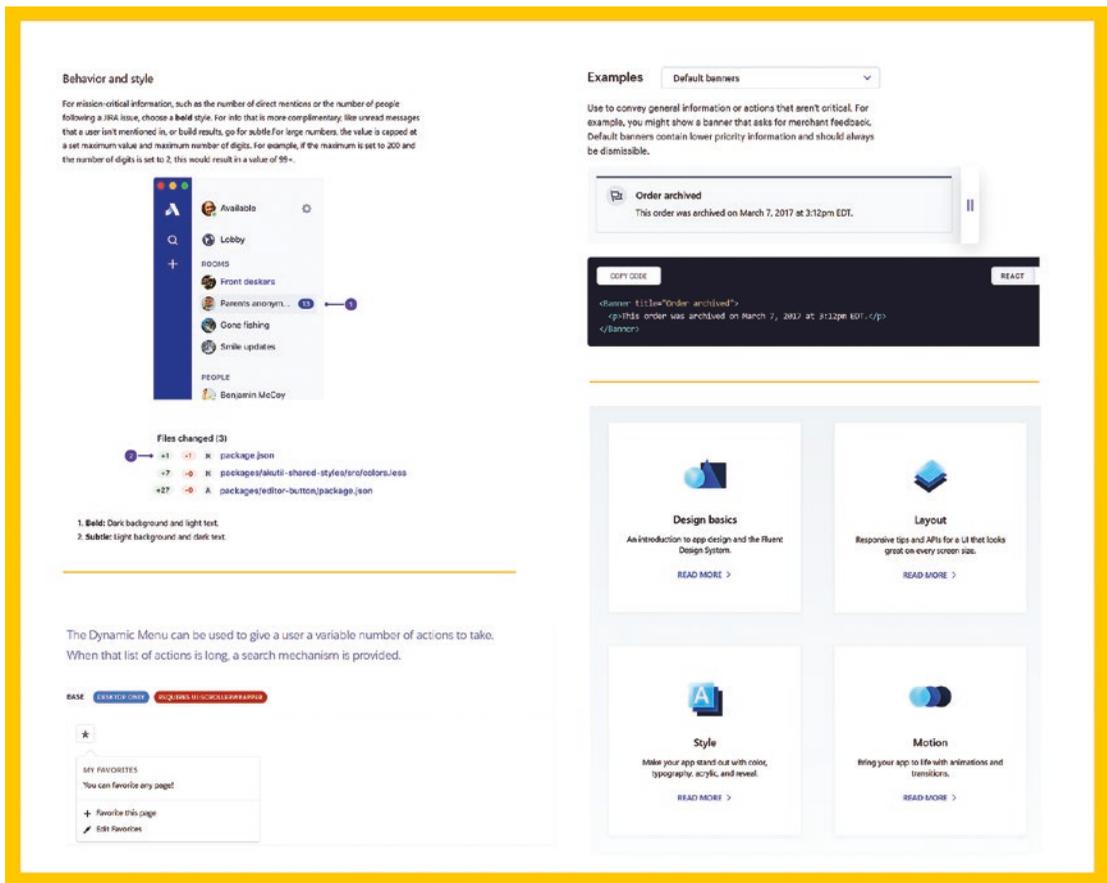


Figure 2-26. Atlassian, Shopify, Microsoft, and Salesforce (clockwise, respectively) all have amazing design system documentation sites that provide great insight and reference points

Finally, what about conventions' limitations on creativity? I don't think it's controversial to suggest that the requirement of convention stifles creativity. In fact, one of the most exhausting aspects of learning design is creating mental models and catalogs of all these conventions and human expectations. However, we must accept that a large part of the design practice's DNA is rooted in maneuvering within necessary and explicit constraints. The positive and enriching side to having such constraints is that we're given one fewer area in which to be creative for creativity's sake. Should we wish to break a convention, its ubiquity forces us to honestly and integrally justify that decision. As designers, we need to reconcile our relationship with constraints and accept that there's no greater justification for them than the acceptance within entire populations that, yes, this is how we interact with things.

In Search of the Soul of Design

One of the key purposes of visual design, beyond simple legibility and contrast, is “personality”. Personality is the emotional impact of our work—how it communicates with people, the characteristics it conveys, and its mannerisms. By understanding some of the implications of aesthetics on perceived personality and intuiting the rest, we are able to give our work soul. To give it a feel. Very unscientific, but very real notions. If design was as simple as Gestalt + von Restorff + Cultural Convention = Interface, then it would already be a robot's job, and—while this book is rooted in the understanding and proliferation of scientific concepts—I'm not in any way prepared to dilute design down to the purely “proven.” To do so would be a disservice to the craft.

The soul of a work of design is the human touch, the considered hand of the craftsperson shining through, and its a profound impact on emotional response and attachment to a product or interface. This intangible vibe is something that is always evolving within us as designers. It's those little details that—at the right moment, in the right way—communicate more than just function or causality, and reinforce personality. By using our skills, knowledge, and taste to produce designs that are witty, humorous, sympathetic, or myriad other aspirational characteristics, we find ourselves closest to the truth of our craft—we are here to serve emotion. A huge part of this is to appear—and more importantly to actually —*be* genuine.

Genuine Aesthetics

When dealing with the idea of personality in our interfaces, what we are essentially pursuing is a seamless blend of the key areas of aesthetics and perception. How something looks, moves, sounds, speaks, and responds—or, more specifically, how these factors are perceived—all add up to the notion of its overarching personality. As we explored when we looked at animation earlier, the general direction and energy of an element’s movement have a huge impact on the feel of that element. The same applies to color, typography, photography, and illustration style. The key here is that, when we make decisions within these areas, we are doing so with at least one eye on how they come together to communicate an overall tone or personality. A form of visual alchemy, if you will. Each constituent part contributes to something that emerges greater than the sum of said parts. Without this consistency in feel, we introduce unnecessary cognitive dissonance, lose authenticity, and run the risk of negating any possible emotional affect or attachment.

The Super Team Deluxe online shop (Figure 2-27) is a wonderful example of effectively communicating personality through design. The site is full of organic shapes, vibrant gradients, and adorable illustrations. Animations have a bounce and swagger to them, and the copy is maybe, probably, the best copywriting on the whole entire Internet. The site commits “hard” to this charming, irreverent, and fun personality. Every interaction oozes a certain playfulness—in line with the products they sell—with every design touch feeling neither egregious nor diluted. The site is usable, follows cultural conventions, and has a clear visual hierarchy, but the personality it exudes is what makes it such a joy to behold.



Figure 2-27. *The Super Team Deluxe web site, brimming with personality and potential for emotional connections*

Slow Down and Make Things

Technology creates connections, and design has a huge say in the social contract behind those connections. Our work is an infusion of emotion and craft an idea and giving a damn. Yet, the principle literary works of modern product are written by behavioral economists, marketers, and CEOs. We call people “users,” we want them “hooked” on our apps, and a “successful” product is that of artifice. And we “sprint” to make these, we “move fast and break things”—even though the things we’ve just broken are held together with string and chewing gum—and somewhere along the way, we’re sucked into the vapid void of late-stage tech capitalism. A world of shortcuts and loveless churned-out products, a scrapyards of failed startups, of bare-minimum wireframe-like interfaces, of not-my-problems and it’s-good-enough-to-launch-withs. We gladly release half-working applications, and they’re not-quite-so-gladly received by a customer base that is so used to ephemerality ad nauseum that they barely even register the fact they’re paying for a product team’s science experiment. Paying, at least, with their time,

attention, and mental reserves. But we don't have to conform to this—and we shouldn't. In pouring yourself into this craft, in finding the soul of your work, in wielding grace and empathy and mindfulness, you *will* create work that contrasts in this climate. There is room for warmth and emotional connection in our work.

Too often, as we “move fast and break things,” we err on the side of caution with aesthetics and design—favoring utilitarianism over personality. Design does not exist merely as a vehicle for usable components, sensical interactions, and appropriate visual categorization. While these aspects are the non-negotiable, bare-minimum considerations of a modern interface, the real art of design is in the charm, the sincerity, the *joie de vivre*. There's a special beauty to work that not only effortlessly choreographs contrast, causality, metaphor, and convention, but does so with an uncompromising and obvious personality. An interface with a soul is our legacy—it embodies and emboldens us. Let's not abandon that in the name of relentlessly sprinting toward feature completion.

Summary

In this chapter, we explored the notion that perception is affected and formed by our past experiences and schematic understanding of the world. The Impressionist movement took art away from hyper-realism and toward a style of painting that was more whimsical, more in-the-moment. This style allows us to explore a deeper, artistic representation of the mind's attentional system, presenting us blanks that we are asked to “fill in.”

We explored the idea of iconic abstraction, discussed by Scott McCloud in his excellent *Understanding Comics*, suggesting that the further we venture from realism, the more room we leave for personal affect and emotion to be found in an artwork. This might also impact the perception of our illustration and icon work in our interfaces, and could provide an interesting means of assessing where on the spectrum of realism (photography to simplistic illustration) we decide to place our representative media.

The Gestalt principles of perception provide us with useful guidelines for what is known as “visual field organization”—namely, how we instinctively perceive elements as grouped based on a number of relationship factors. Proximity tells us that items placed closer to one another are seen as “more grouped” than items that are further apart. Similarity shows us that we group items that have the same, or similar, visual properties together. Continuity shows us that items that are placed along a perceived “path” are

seen as related and can often be seen (in the case of our dotted line) as an entity unto themselves. Common fate tells us that elements that move in the same direction, or along the same path, can be grouped together through their motion alone. Closure shows that we're able to perceive elements that aren't actually present based on the appearance and boundaries of other shapes or elements. Finally, common region shows that elements that are contained in the same region are naturally grouped.

Gestaltism extends beyond analyzing perception, however, and presents us with one of the core themes of this book, the notion that the "whole" is a different, discrete entity to its constituent "parts." We see forests rather than lots of trees, for example. Gestalt, too, can be used as a tool of cognitive economy. By understanding the building blocks of visual grouping, we can ensure that related information is grouped together correctly within our interfaces—almost always as a combination of many of these guidelines.

We explored the idea that colors themselves do not possess any form of inherent, evolutionary importance outside of their relationship with an environment and the cultural significance that is applied to that color. This notion extends to many forms of perception—showing that through both our innate need to categorize and our openness to cultural convention, we're able to codify our perceptions in specific contexts and environments. We explored that the isolation effect (or The von Restorff Effect) may hold the secret as to why and how specific colors achieved their cultural significance over time. This effect shows that when we see an element that stands out from a homogenous environment, we will pay more attention, and more readily remember, that element, while the surrounding elements are "demoted" in importance. The combination of the isolation effect and our ability to codify our perceptions leads to interesting ways of communicating functionality or causality. We saw how, in *Horizon: Zero Dawn*, a somewhat complex and idiosyncratic mechanic of "climbing an object" was related to something as simple and universal as the color yellow. "Yellow means climbable" represented a domain-specific association between environment and action.

We briefly touched on the idea of "affordances"—more specifically, how we often confuse metaphor and convention as such. A combination of a metaphor and convention (exemplified by our use of a form button's disabled state) can help elements communicate their interactive properties. We also discussed how digital interfaces might accommodate for "lost affordances"—affordances or signifiers that exist and provide useful feedback in the physical world that we cannot attain in digital interfaces. We used the example of a progress indicator on a digital reader making up for the lost physical qualities of a real-world book—the weight and visual size of the still-to-read

pages provide a real-world method for us to estimate how much is left in our book. Rather than attempting to “stack” some strange metaphor of digital “pages,” the progress indicator represents a digitally native convention that is used to effectively communicate something that the real-world format of a book provides us.

Finally, we discussed how our design decisions can give our work a vibe and a personality of some kind. By spending time creating unexpectedly lovely visuals and animations, we’re able to explore the soul and humanity of our work. Slowing down and being true and genuine to the personality and conversation we’re trying to present (just as we saw with the Super Team Deluxe online store) can strengthen the connections we’re attempting to create with our work.

If you were to only take one thing from this chapter, I’d like it to be a clear understanding of the Gestalt principles and how they can be used to communicate visual grouping. We categorize by nature, and using some of our innate “rules” for perceptual grouping is the backbone of good visual design. The Gestalt principles may, to many, seem to simply be common sense, yet they really do represent the fundamentals of design. Furthermore, while each “rule” is simple when taken out of context, the real work comes in effectively combining them in the context of an entire interface, or application state. Just as Gestalt itself suggests, the whole is a separate entity to the sum of its parts—by learning how our mind perceives things to be similar, and thus to be grouped, we can start to figure out how similar they need to be.

CHAPTER 3

Learning and Memory

Rich people without wisdom and learning are but sheep with golden fleeces.

—Solon

To explore how humans learn as a species is to explore the very cores of cognition and emotion themselves. In its most reduced model, learning is simply the transmission of information from sensory input into long-term memory, with a few checkpoints along the way. In a wider sense, there are infinite ways in which we can learn things and seemingly infinite categories of learning we can partake in. Furthermore, learning is not *just* about committing a fact to long-term memory—learning a skill is as different from learning a fact as it is from developing a habit, yet all can be viewed as some form of learning or memorization. Some things, like our native language, are learned early on in life, without us really having a choice or making a conscious effort. Others, like playing a musical instrument or riding a bike, require enough deliberate practice for us to eventually treat the act as one of muscle memory. As we discussed in Chapter 1, the ability to delegate complex tasks to our subconscious plays a large role in cognitive economy. Once a task requires little to no conscious attention, it becomes less taxing to perform, allowing us to apply focus elsewhere—should we need or wish to—while we perform it.

Yet the role of cognitive economy in learning does little to explain the complex emotional and motivational structures behind the vast majority of our decisions or desires to learn. This is where goals come in to play. Aside from facts we pick up and remember seemingly at random, the overwhelming majority of the knowledge, skills, and habits we learn and develop over time are preceded by a *goal*. Sometimes that goal involves simply understanding more about the world we live in—studying biology, for example, to satisfy an innate curiosity of how the body and its systems function. Other times, that goal is more tangible or emotional in nature. We may fall in love with a piece

of music that inspires us to learn an instrument, which unto itself might reveal many different goals—reaching a level of proficiency to play that particular song or to create your own music in the vein of that piece. Almost every musician I know, including myself, can trace their musical career back to *that one song* that led to decades of deliberate practice sparked by a single emotional response. Our goals are almost always unique to and intrinsic to us. We may learn Japanese due to a desire to travel to Japan and engage in the culture in a deeper way, we may learn to use a drill so we can put up a shelf in order to display our favorite books, and we may even try to learn to do a sick backflip to impress our attractive neighbor, (The last one is *definitely* just a hypothetical example, I promise.) While we'll explore some of the neuroscience of learning, and indeed most of what we discuss will be through delving into the inseparable relationship of cognition and emotion when applied to learning, I want to make clear that it's the *goal* behind the need or desire to learn that we as designers should be focusing on. Our job is to get to the emotional epicenter of these goals and create something that helps move toward the achievement of them, motivating and inspiring where possible, to seamlessly solve the myriad problems and remove the myriad obstacles that are often encountered throughout any learning process.

A Note on Goals

The features and functions of our products should almost never be conflated with the goals of the people who use them. Get used to asking *why* a few more times than you currently do when exploring your feature set. If someone uses a product to send money to others, their goal is not necessarily to “perform a bank transfer,” and it's absolutely not to “tap the *transfer* button and then input the transaction details and then tap *send*.” We're often far too guilty of conflating interactions with goals, but there's always something deeper, something further removed (physically and metaphorically) from the screen that should inform our thinking. Why does someone need to transfer money? Perhaps to split a bar tab with their friend? How would your app change were you to design around that potentiality? Perhaps they send money to pay off a debt, or to pay a seller on an auction site, or maybe it's a charitable donation—all unique goals that one would expect to be framed and executed uniquely. Whatever the product, without understanding the real-world goals that inform its features, we limit ourselves to designing around the shallowest potentialities.

More often than we'd like to admit, a person's goals when using our work are far, far removed from actions that actually take place within our interfaces. We need to become more comfortable with accepting and embracing transience in our work. There's *always* an external driver to the actions that occur in our interfaces, and often the best thing we can do is to design so effectively for seamlessness that our interface becomes invisible. There is absolutely no shame in creating a wonderful environment for the "just-passing-through" crowd. Indeed, the modern-day fetishization of eyes-on-pixels and attention economies far too often skews our vision for our work. It suggests that the more "engaging" our products are, the better we're performing. But, generally, people aren't looking for *engagement*; they're looking for solutions to their problems. They're looking to be enabled and empowered in their pursuit of a goal. At the risk of generalizing, here's a hypothesis: most people want to engage with their banking app about as much as they want to engage with their alarm clock. That is, for the shortest time possible.

Even for media-streaming products such as Netflix and YouTube, our goal is not to "engage with the app." It's *escapism*, or it's *education*, or it's just outright *entertainment*. The fact that roughly 94% my interactions with Netflix involve scrolling through with my partner and taking it in turns to say "Nah, not tonight" to every suggestion we see doesn't mean Netflix has created an engaging interface. It means that, first, finding a good show is *hard work* and, second, the interface is *apparently* doing a poor job of presenting us with viable solutions to that problem.

One approach here is to jump headfirst into the rabbit hole of human emotion and motivation—to try and figure people out, get a grasp of what makes them tick, and to accede that the usefulness and impact of our applications (both positive and negative) come from, and manifest themselves in, the *real world*. It's our responsibility as designers, with the skills and knowledge we have, to do this level of due diligence in our approach to all our work, for without this initial research and understanding, how can we ever get to the core justifications for our design decisions? This chapter will be as much a meditation on this idea as it will be an exploration of human learning. I implore you to think about how you'll apply the concepts we explore throughout this book to better understand and serve intrinsic human goals—goals that exist far beyond the interactions our interfaces enable.

Learnable vs. Educational Applications

When I think of learning in tech, my first thought is to jump to the many e-learning apps that exist today—products like Memrise, Skillshare, Codecademy, and so on. And, while these learning apps offer extremely interesting insights into the behaviors behind human learning, they're specifically designed *educational* interfaces. A more universal concept is that of a *learnable* interface—and to avoid confusion I want to quickly separate the two. When I refer to *educational* apps, I'm specifically referring to products whose primary purpose is to teach people new skills—think Memrise for learning a language or Codecademy for learning how to code. When I talk about a *learnable* interface, I'm talking about *any* interface that presents its features and interactions in a way that allows one to internalize and remember its uses. Any interface can be learnable—it's an aspirational concept—but not all interfaces are, nor should be, educational. In fact, in scratching the surface of signifiers and metaphor in the previous chapter, I hope it's somewhat clear that even pursuing a *learnable* interface may often be a fool's errand.

The backbone of seamless design is the idea that intuition outdoes memorization. This idea forms the very basis of usability, and we rely on it every time we turn on a device and attempt to navigate its digital interfaces. When we explore the idea of learnability, we should assume that we've already ruled out the potentiality of making something intuitive. That is to say that either (a) the concept we're conveying has no universal signifier or intuitive mental model around it (think early touch device, or newer, purely gestural interfaces), or (b) that the non-universal mental model we're proposing represents an advantage that makes it worth learning. As I mentioned in Chapter 2, the acceptance among a population of a particular pattern or signifier portrays an extremely valid argument for convention; however, countless innovations, including the touch interfaces that revolutionized modern technology, would have never occurred if we *always* relied on or settled upon such conventions. Distributed cognition is a wildly interesting concept. While it doesn't completely explain the phenomena of a culture's acceptance of a specific interaction paradigm, it suggests that we can rely on the proliferation (or lack thereof) of concepts, mnemonics, signifiers, and such, as a barometer for success. What this essentially boils down to is a suggestion that if you're introducing a concept that can't be portrayed intuitively, your best way of confirming its efficacy is to simply *see if people bother to learn it*.

As is quickly becoming a mantra in this book, interface concepts that ask for the extra cognitive effort required in learning should be presented in an environment free of distraction and competing concepts, wherever possible. It's important to remember that

learnable concepts inherently start life as *unlearned*. It's not until we witness them in use that we can add their behavior and functionality to our cognitive map of the system we're using. Just like in the real world, the speed at which our concepts are learned and the resulting confidence in a person's understanding of that concept are highly reliant on the efficacy of the learning environment—in our case, the whole of our interface. By taking on board our knowledge of attention, cognitive economy, and effective grouping, we can frame and introduce our learnable concepts in environments that are free of distractions, sensibly structured, and as simple as possible to perceptually process. Further, by acknowledging and embracing the emotional impact of our craft, we can provide the essential tone and necessary character to the messaging around these interactions. These represent both the essential components of an effectively learnable concept and a broader culmination of interaction design concepts. We'll explore the anatomy of interactions in much greater detail in Part 2 of this book, but examining the value they add to the learning process represents a great springboard for later discussions.

How We Learn

While the most interesting and challenging aspect of designing around the human learning process lies, in my opinion, in the creation of a positive and enriching environment, it'd be remiss to not explore the basics of our learning processes first. The subject of human learning is extraordinarily broad with wide-reaching implications. Thus, the science of learning and memory is rife with debate, controversy, and ethical considerations. With that in mind, we'll cover some of the basic principles of learning and then zone in on where we can really make an impact with our work while leaving the shouting and debating to the professionals.

As mentioned at the beginning of this chapter, learning can be (over-)diluted into the act of transferring sensory data into long-term memory storage. Broadly speaking, we hear or read a fact or observe a technique, and through some form of practice or repetition, we commit it to memory. This process involves said sensory data traversing a few checkpoints along the way, with data loss or corruption often occurring between each stage. Furthermore, for most humans, this process is rarely a straightforward A-B-C flow of information; it's almost always a messy and chaotic back-and-forth between our different memory storage and retrieval structures. Remember in Chapter 1 when we discussed the notion of the brain as a huge network of networks? Well, really interesting things happen within this network when we learn something new. The nodes of our

network are our neurons, which are connected to other neurons through the passing of electrical signals through tail-like axons. Think of them as a typical network cable, passing signals from one node to another. When we learn something new, the brain shifts its structure, often creating new neurons and new connections.

Our brain is a living ecosystem of neurons and synaptic connectors—a universe wherein stars burn out and are born anew. If we were to observe the light show of the brain’s neural activity during learning, it’d be akin to watching a time lapse of our galaxy shifting over millennia. As we learn and meditate on new information, connections between neurons are reinforced, allowing them to better communicate. New connections between neurons are formed, existing connections are occasionally dimmed, and new neurons themselves may even be created. The more often a neural network associated with a specific task is strengthened and activated, the more efficient it becomes. This shifting structure at the brain’s cellular level is what is known as “plasticity”—the idea that our brains are living, malleable things, brimming with life and complexity while constantly changing until the day we die.

Furthermore, it’s this neural strengthening that allows us, over time, to relegate the performance of certain tasks to the “back of our minds.” Essentially, when the neural network used for a certain task is strengthened enough through consistent engagement, it allows for a level of efficiency in neural communication that requires very little conscious thought. When this happens, our default mode network is given more room to operate, we pay less attention to the task, and we can perform it intuitively and seamlessly.

Yet this process is far from perfect, and it has to happen countless times for a single stimulus to work its way through our attentional system, into our short-term or working memory, and finally to our long-term memory. All of that is only half the equation, too, for a memory lays dormant until we need to recall it, with this recall requiring a similar chain reaction of lighting neurons and synaptic activity. This results in some fascinating phenomena where memories that are stored are almost always slightly removed from the “pure” sensory input at the beginning of this chain reaction. We rarely remember anything in verism, seemingly relying on an acceptably low level of corruption to inform the truthfulness of what we recall. As you can probably guess, this opens up a scientific and philosophical minefield with implications ranging as far and wide as the objective trustworthiness of witnesses to crimes, the effectiveness of traditional educational exam structures, and the legitimacy of psychedelic experiences.

Designing a Learnable Environment

The “how” of learning, while a hugely interesting and rewarding topic to explore, doesn’t give us much to go on when it comes to informing our design decisions. What we should be focusing on is the various things that *impact* our learning. Helping memorization, encouraging experimentation, and forgiving mistakes are all things well within the abilities of modern interfaces and all things we should strive to be implementing, regardless of the kind of interface we’re designing. I believe there are several characteristics of an environment that help foster learning, but before we explore them, there’s an important task that should be performed every time we start work on an interface. We need to find the gaps between our users’ system’s model and the *mental model* of their system anyone who interacts with it has.

A mental model of a system is the culmination of all of the assumptions and intuitions someone has about it, including how its constituent parts perform and combine to make the *whole*. Mental models are often simplified, zoomed-out assumptions (Norman, 1988) and rely heavily on heuristics and often metaphor. We tend to form mental models of many of the core aspects of our lives, including social dynamics, sociological and political ideologies, economics, and scientific and mathematical utilities. Our mental models have a huge say in how we perceive the world, yet they are flexible and malleable in nature, constantly shifting as we better grasp the world around us. We can also quite easily use our mental models to our own detriment. Far too often we limit our perception of situations in our life based on a number of our existing mental models. Our worldview biases our perspective and can often lead to myopic reasoning, stubbornness, and dissonance. In the same vein, our unique combination of mental models can result in some quite innovative and refreshing ideas. Approaching a visual design problem with the mental models of cognitive psychology, for example, can lead to some rather creative, emotion-centered work. Similarly, approaching the problem with the mental models of an economist might lead to a very different, equally interesting result.

When it comes to interface design, the mental models we’ll usually be dealing with involve concepts such as the following.

The Storage and Retrieval of Files or Data

A classic example of mental models is that of a computer’s “folder” and “file” structure (Figure 3-1). While our data isn’t actually stored in a physical folder inside a computer, we recognize the metaphor and can apply a mental model to grasp it further. The mental model here usually includes things like “this data (documents, web site, etc.) lives in this container (folder, tab, etc.)” For the most part, an underlying model of the deeper system is not required to grasp the components that interface with it.

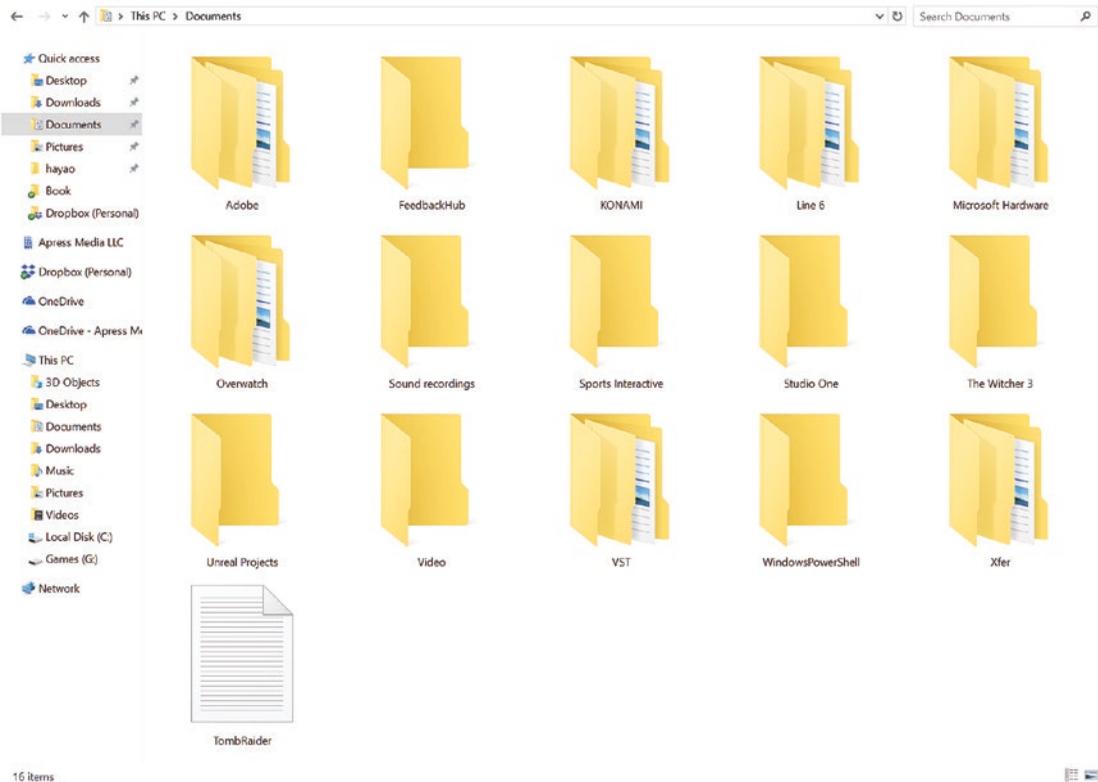


Figure 3-1. Windows 10’s file and folder structures

As stated, a computer doesn't have a bunch of physical folders that store physical documents, but the design of the interface abstracts the system's storage into an understandable model of the real world. Even if someone *was* to believe that computers performed this physical storage on our behalf, it really doesn't matter that they're wrong. Their model of the system is effective enough for them to never have to question its deeper workings. For someone who has to deal with the inner workings of a system, say, a computer technician, their model will be clearly different. In this case, they'd likely have to have a deeper understanding of the machinations of a system—that is, the fact that computers use digital storage, encoding, and decoding for data storage and retrieval—as opposed to the shallower understanding provided by the file/folder metaphor.

Similarly, when it comes to storing and retrieving data within, for example, a web or mobile application, the metaphors used to communicate an adequate model of the system don't necessarily need to inform people of the underlying server or database structure. In fact, one of the biggest mistakes we can make (as we touched on in Chapter 1) is assuming that people external to our organization will have a similar understanding of our category structure as us and our teammates. This also applies to communicating the models of our systems. If we assume that the underlying data structure, such as the schema of our database, is the “best” way to structure data externally, then we run the risk of communicating a too-technical, overly literal model. Where data “lives” and how it's sent there is usually of no concern to someone who just wants to manipulate or store it. Of course, if your app is specifically targeted toward developers and system administrators, it's a different story, but for non-technical users, aim for effective, non-technical models.

The Creation of Media

Design products, like Figma in Figure 3-2, portray a “canvas”-a centric interface containing features such as a pen tool, pencil tool, and various shape-drawing options.

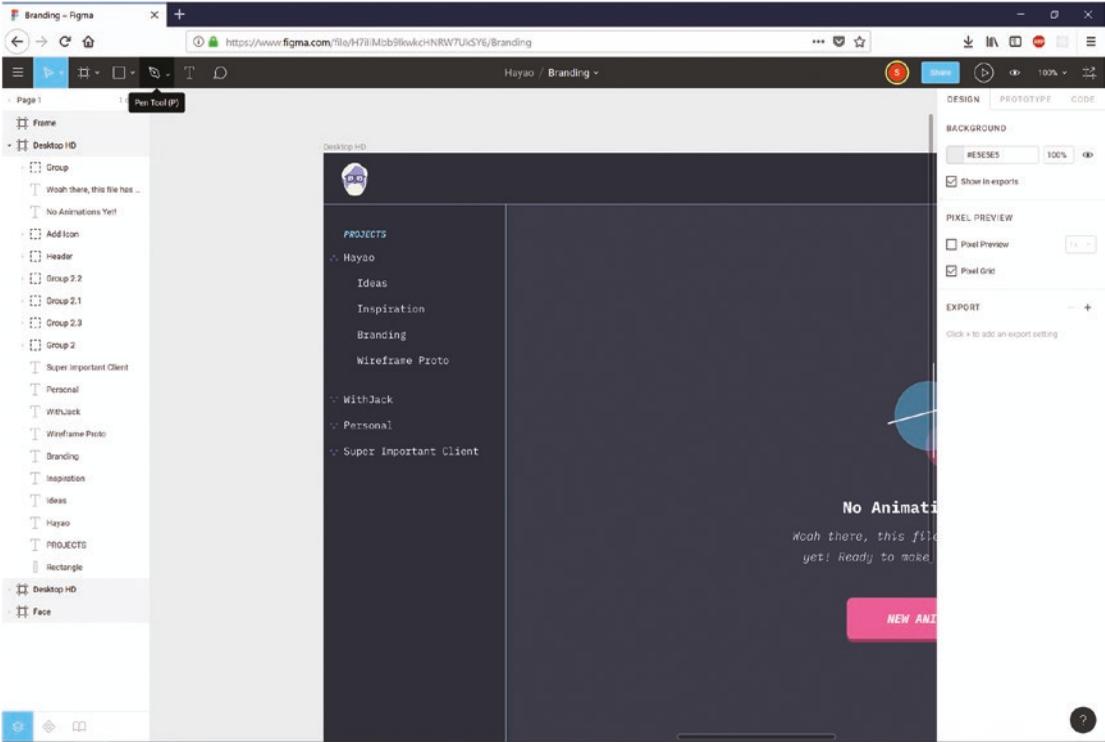


Figure 3-2. Figma

This makes Figma, and design tools like it, conducive to a mental model of the tools we use to create and draw in the real world.

Adobe's Lightroom (Figure 3-3) uses language and concepts related to the analogue world of developing film photography, again allowing for the application of a real-world mental model.

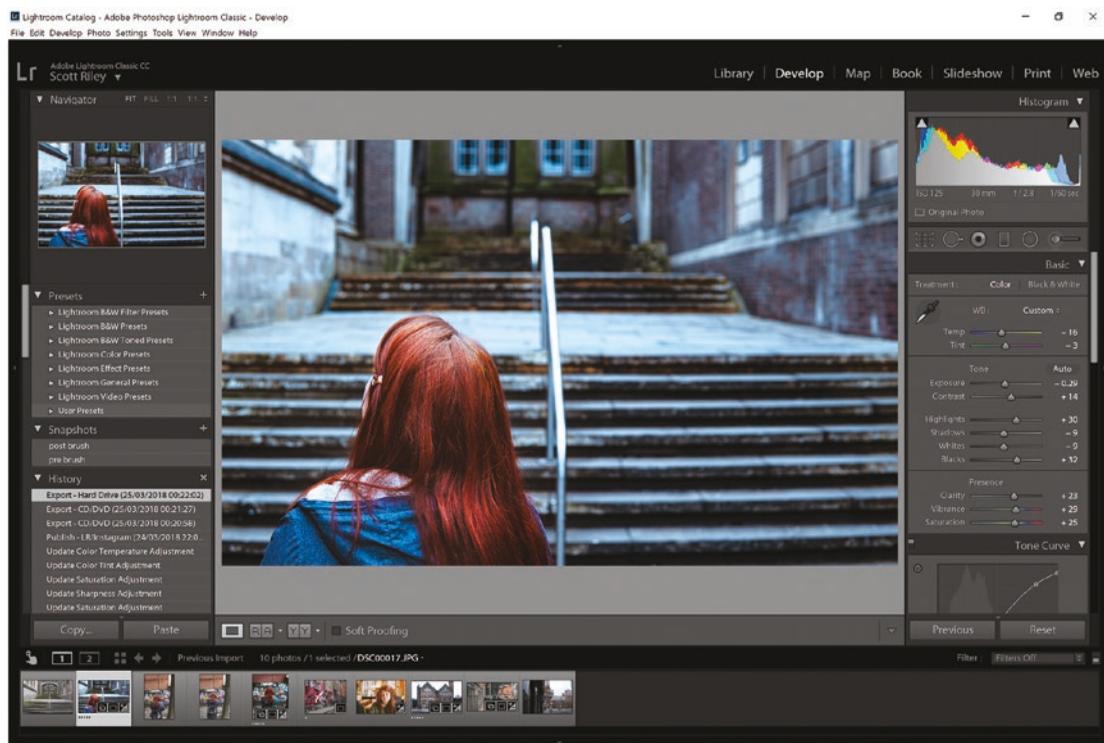


Figure 3-3. Adobe Lightroom

Lightroom has distinct modes such as “Library” and “Develop,” using labeling that would fit the mental models of someone already familiar with the medium.

Finally, Blackmagicdesign’s DaVinci Resolve (Figure 3-4) provides tools such as a “razor” tool—harking back to the days where movie reels were manually sliced and spliced.

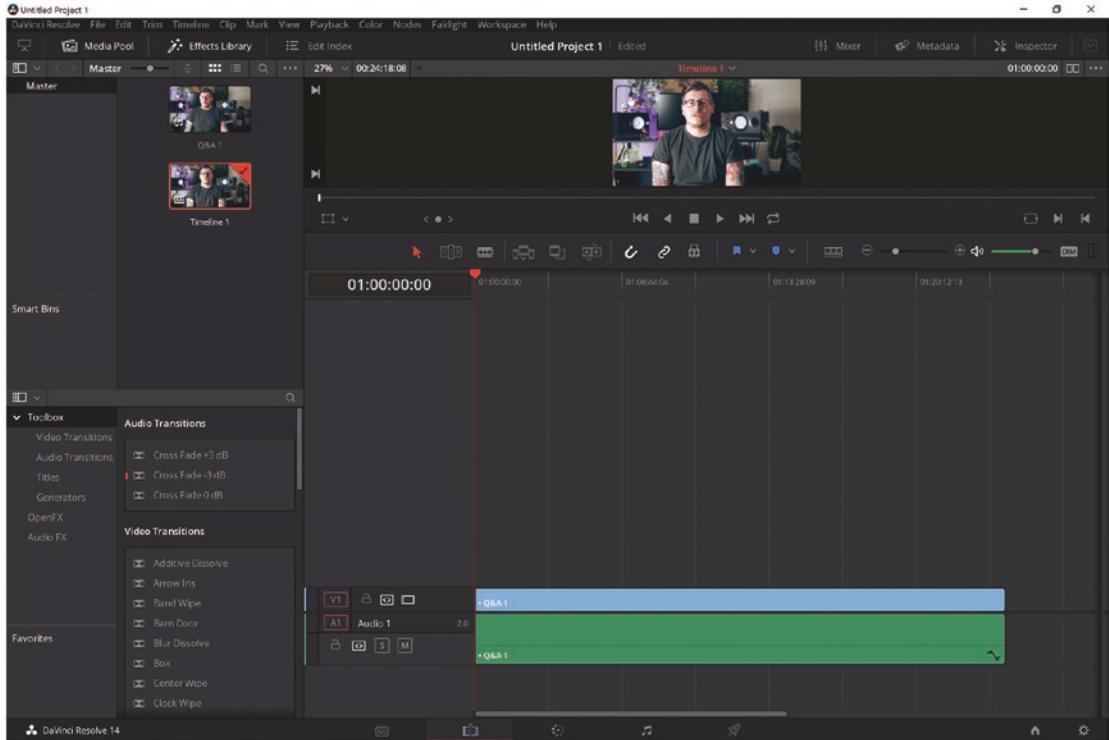


Figure 3-4. Blackmagicdesign DaVinci Resolve

These, of course, only begin to scratch the surface of some of the ways complex interfaces communicate their system’s models, but it shows how such interfaces harness potential real-world metaphor, especially in the media-creation realm.

What I find extremely interesting is how, in the examples shown, the mental models don’t completely fall apart if you’re not aware of the practices they represent. I’ve never in my life touched a reel of movie film—never mind sliced it with a razor blade—yet I understand that razors are used for cutting so when I see such an icon, I’m already intuiting its potential uses. The icon could just as easily have been a pair of scissors and I would have personally garnered the same idea, but it takes away the relationship an industry professional might have with that real-world tool. In deducing that I can

slice video or audio clips with the “razor” tool, I’ve come to the same conclusion as an industry professional probably would have. I’ve just done so by applying a different, less-efficient mental model to the interface.

A large part of our job as designers is to communicate and reconcile the *actual* model of the system (i.e., the thing on top of which we’re designing) with the mental models of the people who interact with it. If our system is a machine, then design is the ghost within it, translating machinations, or zeroes and ones, into real-world, emotionally imbued interactions. Our interfaces are outward representations of our systems’ internal workings, and they’re the first point of reference when people form their mental models. If we have a solid mental model of a system, we’re able to infer how its components can be interacted with and, most importantly, what the result of those interactions might be. Conversely, a poor mental model leaves us feeling lost and frustrated.

Modern products and interfaces are often hugely complex beasts—there’s no changing this. Often the best we can do with our skills is to ensure that this inherent complexity is managed and explained in ways that, over time, bridge any gaps in someone’s mental model of our systems. This is where the idea of *learnability* comes in to play. While convention and signifiers can communicate many functions of an interface at a glance, there will always be times when we need to work around an idiosyncrasy of our system. In these cases, there’s often no means of communication that can 100% intuitively demonstrate a potential action. Herein lies a probable “gap” in a mental model. The system does something in a certain way, which is difficult to explain using conventional interface components, thus creating an ambiguous “smudge” in a person’s mental model of the system. Our priority in this situation is to slowly and purposefully bridge that gap—to sharpen the smudge, if you will—so that the human mental model of our system becomes more useful.

Sharpening Our Mental Models

So far in this chapter, we’ve had a lot of necessary theoretical discussion, but it’s high time we started putting that to some practical use. How do we make sure that, first, the model we’re communicating of our system is as effective and seamless as possible? And, second, once we discover those inevitable gaps and smudges in people’s models of our product, how do we begin clarifying them? I believe the answer to both can, broadly, be summed up as “design an effective environment.” In fact, I believe this statement

embodies interface design as a whole, so much so that there's an entire chapter devoted to this later in this book. For now, let's briefly touch on the key areas that inform an environment's inherent learnability and how we might communicate an effective model of our product's underlying system.

Forgive Mistakes

Whenever we encounter something new, we're faced with a certain degree of trepidation that is usually relative to the complexity of the thing we're trying to comprehend. In the case of a relatively complex interface, one of the main sources of worry is that we'll accidentally delete or corrupt important information. A powerful solution to this can be seen in the—thankfully increasingly more commonplace—implementation of “undo” functionality for destructive tasks. When we looked at GitHub's “Danger Zone” approach to repository deletion back in Chapter 1, we saw an attempt to preempt and inform destructive behavior. What we didn't see is any level of forgiveness *after* the fact. Contrast this with Figure 3-5, which shows Google Inbox's “Undo” functionality after having sent an e-mail.

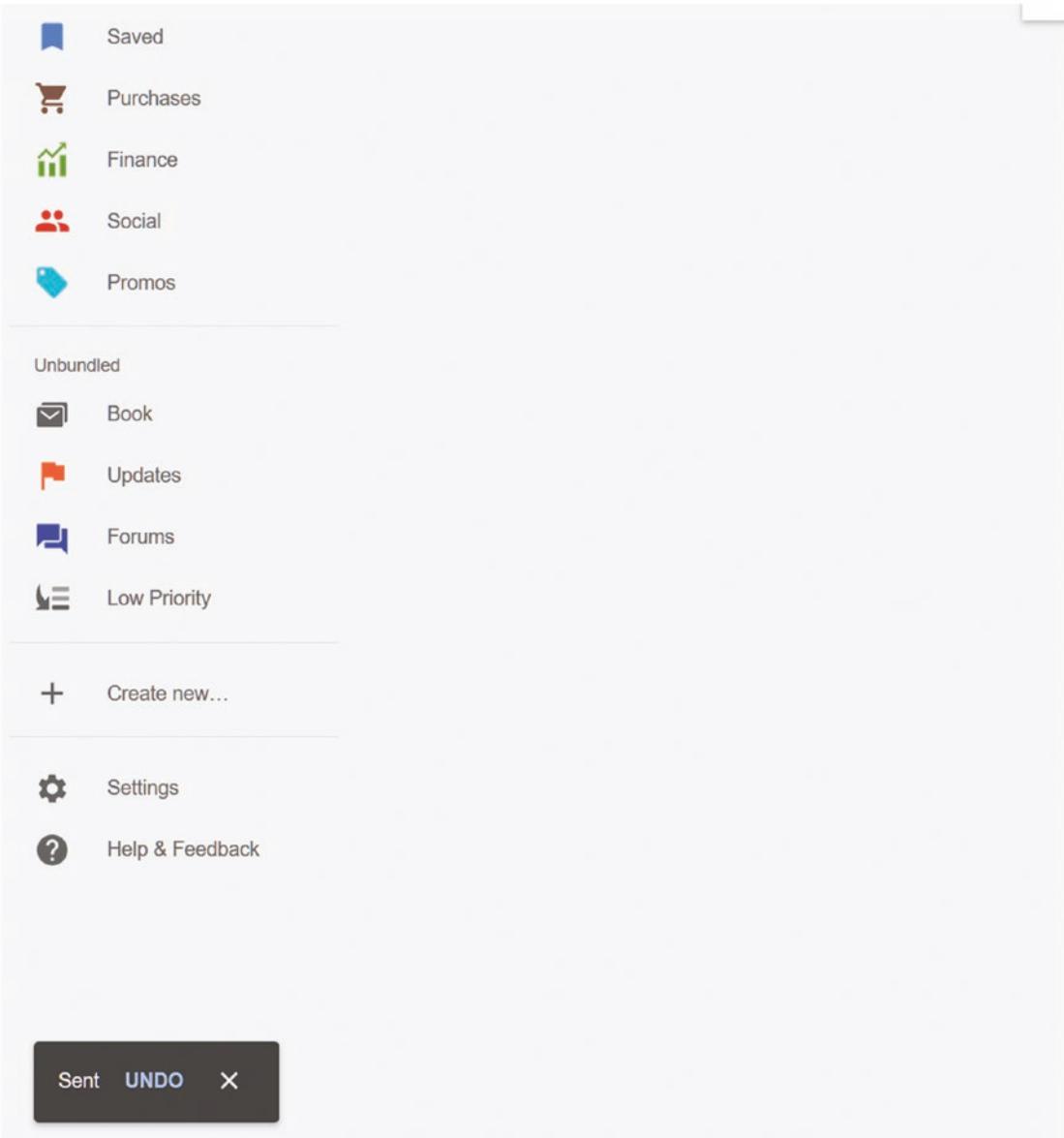


Figure 3-5. After sending an e-mail in Google Inbox, the sending is actually briefly delayed and you have the option to “undo” the action. This is useful if/when you spell someone’s name wrong or accidentally “reply all” with some juicy office gossip.

I think we've all encountered a situation where we've sent an important e-mail only to realize the second we hit send that it contains something less-than-professional—say, forgetting to attach a file or spelling the recipient's name incorrectly. This “forgiveness” that Inbox shows allows us to rectify our mistake like it never happened. This approach allows us to foster an environment wherein people finding their footing can explore and test their assumptions with less trepidation.

Unfortunately, the “are-you-extremely-very-sure-you-absolutely-want-to-do-this” obstruction is still a super common paradigm in interface design, which means that when we *do* offer undo functionality, it's quite likely we'll have to show and/or explain that it's possible.

There's a number of ways to accomplish this, and we'll explore some of the methods of communicating and teaching these concepts a little further on in this chapter. Just know that the “forgiving” model of change management is a far superior approach in relation to fostering an empowering and positive environment. When we're new to something, we *will* make mistakes—and we don't suddenly become “not new” to something as soon as we've gone through some fancy post-signup onboarding or snoozed through a webinar. How your interface handles mistakes (and how well you, as a designer, communicate this handling) directly impacts the learnability of the concepts within. We spend much of our lives learning through exploration and direct manipulation, and testing how things react and respond. In this modern age, where data storage and retrieval is exceptionally cheap and state management solutions are seemingly infinite, there's absolutely no excuse to continue with archaic, irresponsible handling of such important tasks.

Provide Positive Feedback

Enjoyment is one of the most powerful learning tools there is. When we encounter something that makes us react positively, we more readily remember not only the result, but also the actions. Our emotional state has been shown to have a notable impact on how well we remember an activity or action and, especially in a self-initiated, informal learning environment, the intrinsic emotional attachment we have to an object or an idea can be crucial in not only how well we remember it, but also how fondly or vividly. We also remember information more if we're surprised at the point of learning it—something that positive, personality-filled interaction design can actively achieve. This principle is both relatively simple and infinitely complex, namely because we can't ever directly *dictate* the emotion someone may feel at any given time. To some, an animated

mascot brimming with energy might be a complete delight, while, to others, it could be seen as tacky and annoying. This is the risk we take when we decide to not go with the middle-of-the-road our emotional design.

Clearly Communicate State

Application state can broadly be described as a “snapshot” of all the important variables within a system at any given time. These snapshots constitute the purest representation of an application’s model and are the most important thing to communicate to someone trying to achieve a goal within it. State is how we let someone know what actions and options are available to them at any given moment. Changes in state are how we let people know the results of their actions or whether or not their actions are still pending. An interface that effectively communicates current state and, just as importantly, effectively transitions *between* states is an interface that is prime for learning.

When we form our mental model of a system through its interface, we’re only able to reconcile that model if we actually attempt to *do* the things we *think* we can. If we see, for example, a number field with a range slider under it, we might infer that we can change the number by either typing in the input or dragging the slider. However, until we actually *attempt* this change, we simply cannot know for certain. Furthermore, let’s say this number and slider represent something such as the saturation of a photograph in an editing app. We’re actually reconciling a number of assumptions in one go: first, the assumption that changing the slider will change the number in the input (and vice-versa) and, second, that changing either of these fields will result in a perceptible change to the image that we’re editing. These are both logical assumptions (Gestalt, and all that). But, let’s say we tried to test them, and the change in the image was delayed by a second or two. For the sake of example, let’s say it’s a rather large image, and processing and displaying this change in saturation takes some time. We’re essentially transitioning between three states here: the initial *framed* state, wherein we’ve made our assumption as to the relationship between controls and their effects; the *transient* state, wherein any of the necessary processing and/or external communication to handle the action occurs; and the *reporting* state, where the results of the attempted action are communicated back to us. When we test an assumption we’ve made in an environment, the transitions between these states form our reference points. Without these being clear, effective indicators of state, we have little in the way of feedback. Going back to the example of the photo-editing app taking a long time to process an image—if the transient state was never catered for (that is, there was no indication the image was loading or processing),

we're likely to believe our assumption to be incorrect. Conversely, if we're immediately shown a processing indicator upon making our change, we're clearly shown this transient state and know that we need to wait a short time for our assumption to be proven correct or not.

The overall state of an application at any given time presents us with cues against which we're able to make and test assumptions in context—a vital part of the learning process. Without direct, instant changes in state to help us prove our assumptions, we're usually left questioning the interface or ourselves. If we approach an interface with a set goal (which we almost always do), then how well that interface communicates state determines how well we can perceive the status of our goal.

Maximize Cognitive Economy

We've already covered this from concept a number of angles, but cognitive economy is especially important when it comes to designing around learnable concepts. The overall complexity of your interface in any state that requires some form of learning is of huge importance as, by definition, learning requires substantial focus and cognitive effort. By operating at the harmonious baseline of as-low-as-possible cognitive effort that we've discussed throughout this book, we allow someone to devote the required amount of attention to the complex or idiosyncratic features we're asking them to learn.

Allow for Exploration and Experimentation

This last point is more of a culmination of the previous five, yet it bears repeating: an interface that can be effectively explored, where assumptions can be tested without fear of failure or negative reinforcement, is far more learnable than a poorly designed, linear interface. The idea of an explorative interface is something that I think is key to modern interface design. Where applicable, if there are multiple possible ways of achieving a specific result, we move away from the need for linear, learned paths—or user journeys—and toward environments that foster exploration and positive attachments.

The most important thing to consider when we want someone to learn our interfaces is that learning is most effective when done in context, with meaningful, observable results. While the far end of the exploration spectrum would look like a completely open, complex, and exciting environment, in interface design, we'll likely spend most of our time operating at the shallower end with “guided exploration” our likely optimal point. By allowing someone to experience *how* things work without removing them too

far from their usual contexts (we'll dig into this in the next section on onboarding) and clearly communicating the results of interaction, we put ourselves in the most optimal position to have our interface paradigms codified and better remembered.

In Part 2 of this book, we'll start tying all these ideas together in the form of designing effective environments and dissecting the interactions that occur within them. However, for the sake of the learning experience, it's important that there exists some breathing room within which people can carve their own path through our interfaces and toward their goals. By embracing the idea that we can cater both efficiency and exploration, and judging the balance we need between the two in any given situation, we set ourselves up not only to create a great learning environment, but also to create a great interface as a whole.

Levels of Processing

The “funnel” of memory from sensory input to long-term storage is far from a constant, invariable, or linear process. Instead, it's affected by seemingly infinite variables. Everything from our emotional attachment to the stimuli we're presented with to how long we wait to fall asleep after processing a stimulus can have a huge impact on the “quality” of the memory we store for that particular item. One of the most tested theories of memory “quality” is that—rather than memories strictly following an encoding, or stores/structures, process—they're directly dependent on the *depth* of processing performed at the time (Craik and Tulving, 1975).

In their 1975 study, Fergus I.M. Craik and Endel Tulving posited that when we perform deep, semantic processing—such as giving meaning to or creating associations with new information—we create “stronger” memories that are easier to recall. The implications of this study, along with a study three years prior that formulated much of the early workings of this theory (Craik and Lockhart, 1972), are rather fascinating. The notion that deeper processing of information encourages stronger memorization is the foundation of many learning techniques, such as mind maps and the famous “put-it-in-your-own-words” approach of the Feynman Technique. But what implication does this have on our approach to design?

Do Make Me Think

Quite simply, sometimes we just need to slow down and *think* when we're using an interface. The old adage of purely intuitive and completely friction-free experiences being the pinnacle of usability, while apt and still extremely relevant, doesn't cover all of our bases. For learnable concepts, a certain degree of friction—or at least a slowing

down of the fast-paced, bottom-up autopilot to encourage more top-down, deliberate thinking—can actually be a huge help. By gracefully or enjoyably *impeding* progress or thoroughly encouraging someone to interact with a complex feature in a meaningful way, we can go some way to, hopefully, ensuring that concept is better remembered. There’s a palpable degree of irony, if not hypocrisy, in advocating this approach in a book about design’s role in cognitive economy, but this is an optimistic, long-game technique. In slowing things down, just a touch, the first few times a difficult or idiosyncratic interaction is taking place, an interface can help us better remember how to perform it during subsequent uses.

This doesn’t mean that we should be going around and making our complex interactions even more difficult, nor does it provide us with an excuse to ignore convention and intuitiveness. Semantic processing is something that requires more focus and more mental energy to perform. To request this from someone is something that requires heavy justification.

Where and When We Teach

One of the key factors in teaching the concepts of a system is deciding on *when* and *where* we do so. With “traditional” products and interfaces, we generally have a few key areas where we can provide educational information or take people through guided interactions. When deciding what approach we take in guidance, it’s important to avoid anything that remotely resembles information overload. When presented with new concepts, we’re already likely to feel some form of blank-slate anxiety, so compounding this by asking for a lot of up-front effort in order to learn your concepts can quite quickly result in someone abandoning your product altogether. Let’s explore these key areas of guidance and, in the process, provide a number of “checkpoints” across which we can spread the effort of learning.

Onboarding

While more open, explorative interfaces are key to continued learning and creativity, throwing someone into such an environment without any priming or initial context will almost always be overwhelming. An extremely common practice in modern product design involves taking someone through an initial “onboarding” phase, usually straight after signing up for a service. This phase is often designed to introduce people to the concepts and key components of the product and interface, usually by separating them out in a somewhat linear fashion. It’s extremely tempting to use onboarding as a means

of communicating *all* of the key concepts of your product and equally tempting to isolate these into their own dedicated “steps.” However, we must be careful to avoid de-contextualizing actions. While onboarding is fantastic for set-and-forget style settings, it’s important to avoid removing components too far from their usual contexts, and even more important to avoid falling into the trap of mistaking “instruction” for teaching.

As discussed earlier, the more actively we engage with something, the more likely we are to remember it. This makes onboarding that contains some form of interaction essential. While it’s common to see flows like that shown in Figure 3-6, simply “stepped” screens that introduce the key features of an app without any actual interaction—labeled as “onboarding”—are really an extension of an app’s marketing at best.



Figure 3-6. Google Drive’s onboarding for their iPad app is just three stepped screens of informational content

While this approach isn't necessarily bad practice, it is pure instruction, which is not conducive to learning and often acts as a barrier to getting up and running an interface.

A technique employed by most onboarding processes is that of *progressive disclosure*. Progressive disclosure, as the name suggests, involves gradually or contextually disclosing new information and features to someone as they progress through the layers of a system. While progressive disclosure is a relatively broad interaction design concept and is not without its downfalls, its use in early-stage onboarding highlights many of its key benefits. If our onboarding process is the only real point in our interfaces where we restrict exploration and impose a somewhat linear flow, it puts us in control of the order in which we “disclose” our information. Again, it's important to be careful not to remove components too far from their usual contexts, but by limiting interactions early, on we're able to take advantage of this initial, controlled stage of learning.

Good onboarding will not only introduce the core concepts that form the underlying model of a system, but it will also allow people to explore their usage and causality. As we examined previously, testing assumptions is a core component of effective learning, and a solid onboarding phase must still allow for the flow of “assume, test, reflect.” An effective onboarding process will have people using our interfaces to perform real actions with components that communicate their effects clearly. Just as we saw in the previous chapter with *Horizon: Zero Dawn*'s use of yellow as a signifier for climbable objects, discovering this association for ourselves is a powerful way of learning how an interface communicates its causality. Now, unless we're designing a game, we likely lack a lot of the intrinsic motivation that's required for such an openly explorative onboarding process, but we can still garner some valuable insight from it—namely, that we're able to codify and associate causality without much in the way of explicit instruction. An onboarding process that focuses on presenting key features in context as much as absolutely possible in ways that allow for interactions that are consistent with the in-app behavior of the elements will almost always outperform one that is purely instructional.

In-App Guidance

Aside from the focused, pre-emptive teaching that we often attempt to perform in typical onboarding processes, an often-overlooked form of teaching is “in-app” teaching. There are many possible ways of implementing this approach, but the most common take the form of *interactive tours* and *contextual help*—and often the in-app learning includes both these approaches.

First, interactive tours are similar to onboarding in that they're intended to drip-feed information and progressively disclose features, yet they differ in the fact that they're *usually* only initiated when it's clear that someone wants to start a complex chain of interactions. It's important to note that in-app guidance can *be* your onboarding process. In fact, it's extremely common to forego a "controlled" onboarding process and present a fully functional interface up front and ready to explore, with in-app guidance scattered around where necessary. This approach to guidance is extremely useful if your surface-level interactions are relatively simple to understand and mostly intuitive, and the complexity of your interface lies a little deeper. You get to present a simple and concise interface up front and then interject when you feel the need arises.

A great example of this is Photoshop CC 2018's in-app guidance. I'd go as far to say Photoshop's new (in 2018, at least) in-app guidance is one of the most effective onboarding experiences of any app I've studied. When you first open Photoshop, you're presented with the "Learn" screen, seen in Figure 3-7. This screen alone utilizes *so many* of the concepts we've discussed in this chapter. First, the Photoshop team has identified some common tasks that people picking up the product for the first time might want to perform, such as removing imperfections from an image, changing the color of certain objects, and duplicating objects within a scene. This is an application that considers our *goals* from the very beginning.

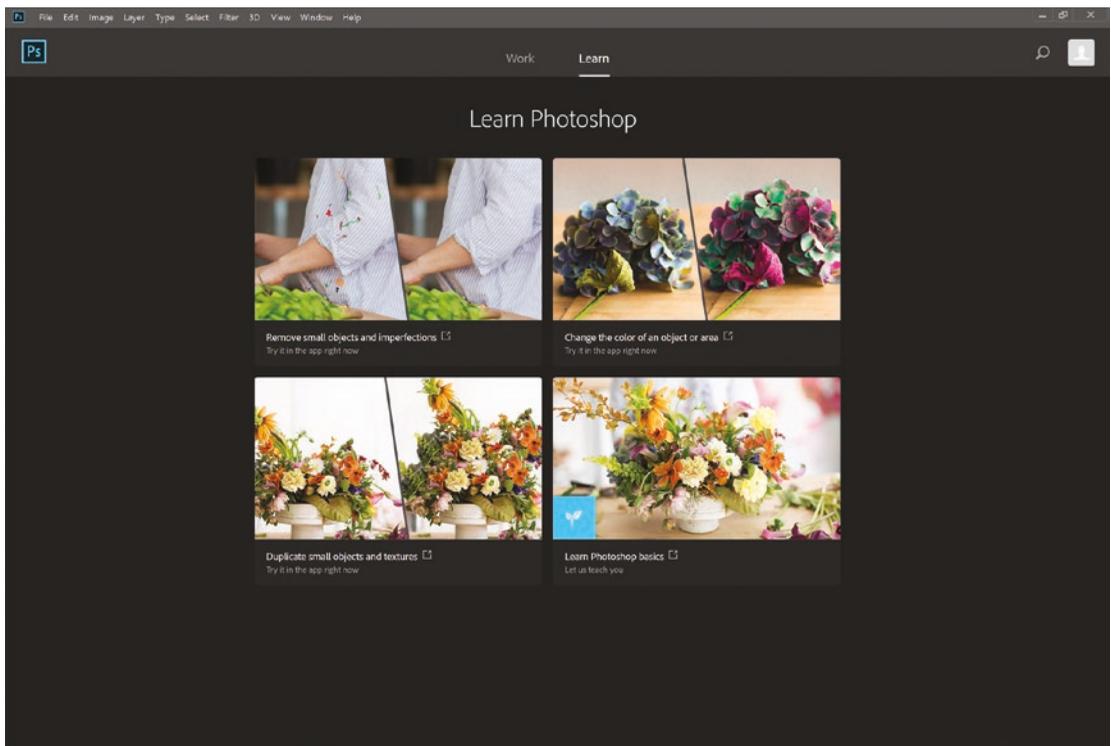


Figure 3-7. Photoshop CC 2018’s “Learn” screen

In addition to giving us the option of selecting one of these goals, the preview image for each goal actually shows a before/after comparison so we get to see the type outcome we might expect. This answers some of our most important questions right off the bat. The “try-it-in-the-app-right-now” microcopy is a descriptive, arguably even exciting, call to action.

Once we’ve selected our example, we’re shown the screen in Figure 3-8. It’s the Photoshop interface, fully explorable, but with a sample file opened already for us, a “Learn” panel open on the right providing us with detailed text, and a succinct tool tip telling us what tool to select.

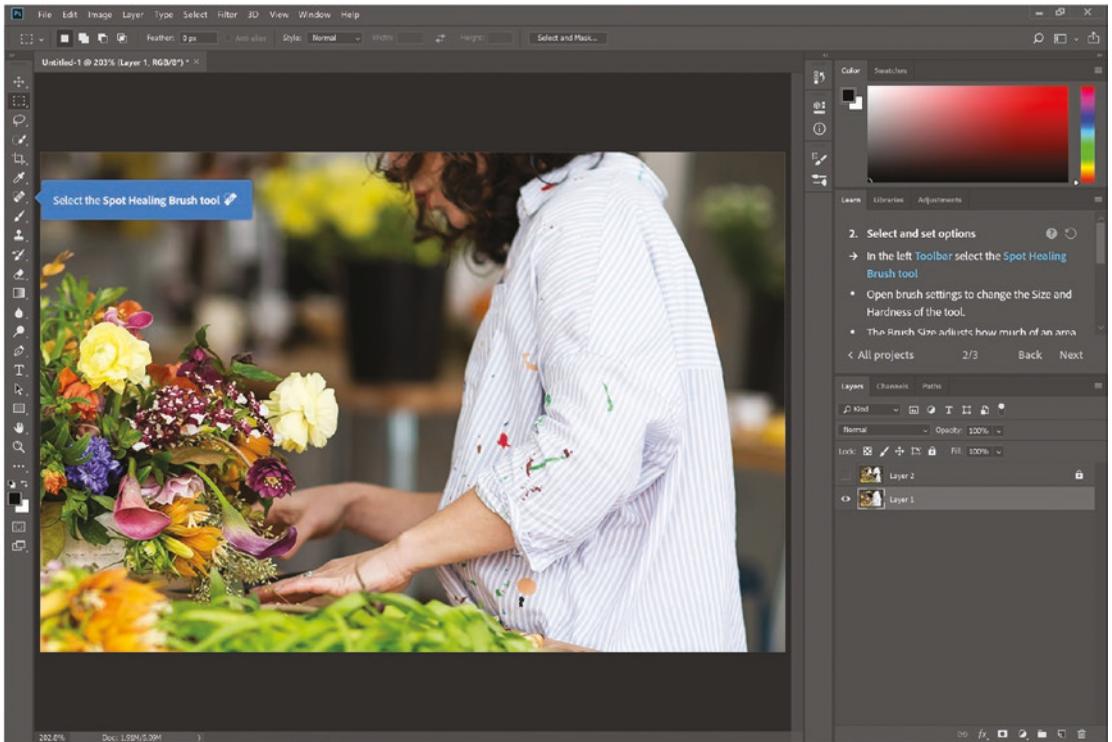


Figure 3-8. Photoshop’s interface with in-app guidance

In this case, it’s the “Spot Healing” brush tool. As an added bonus, when we hover over the tool in the sidebar, we’re shown a video of it in action (Figure 3-9), which allows us to see both its common usage and the potential results we can achieve.

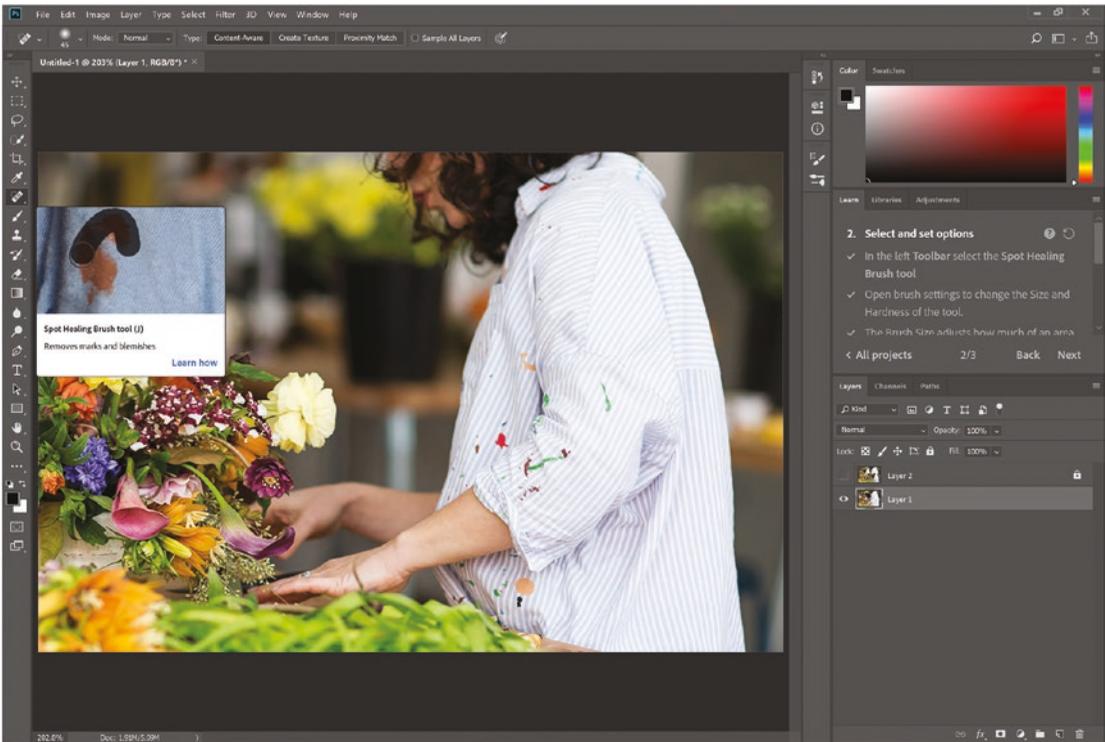


Figure 3-9. *Hovering over any tool for a short while shows an example video of the tool in use*

And once we select the tool, we’re given not only instructions as to how to customize it, but also a *reason why* we might want to do it. We’re asked to change our brush settings (Figure 3-10), and we’re told it’s because “we want a brush that’ll just cover the splotches on the shirt.”

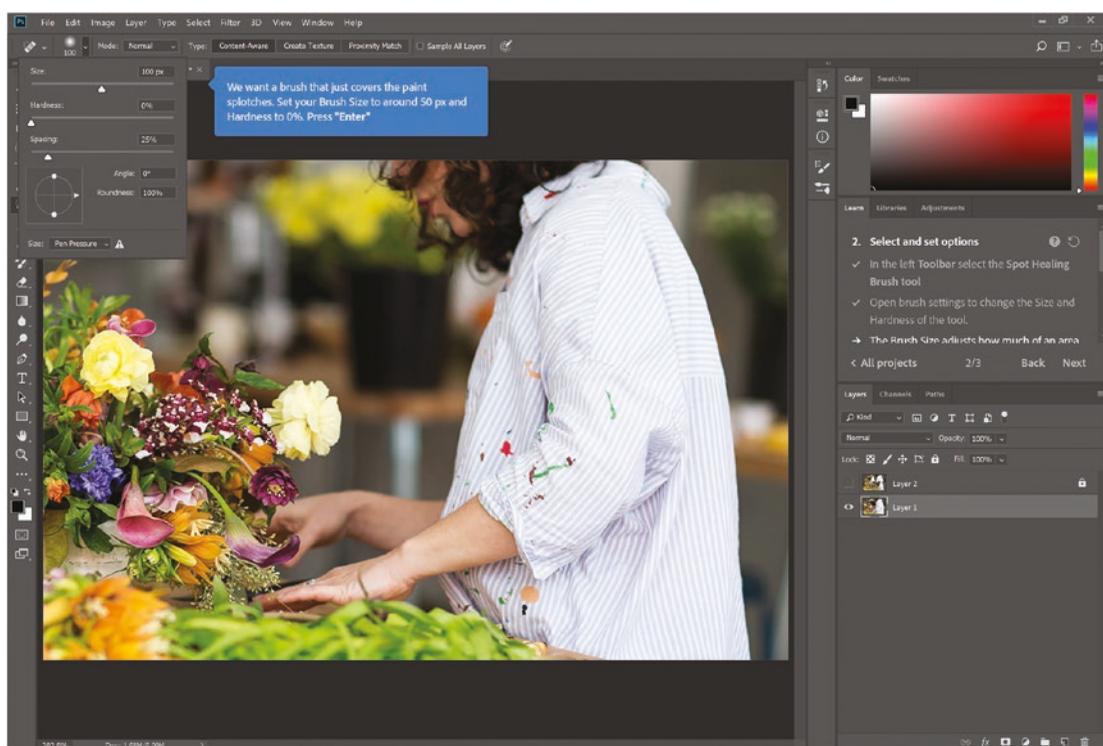


Figure 3-10. After selecting a tool, Photoshop’s interface walks us through configuring it

This reasoning is so important. By explaining *why* a certain parameter is being changed, not only is the causality of the settings explained to us, it’s also related back to a specific goal. In this case, the sliders, or text inputs, will change our brush hardness and size, and the reason we’re doing this is to “just cover” the splatches. We already stated our goal when we chose this particular tutorial to follow, so by relating configuration back to that goal, we’re left knowing that these settings changes are far from arbitrary practice.

Finally, once we've set and configured our tool, we're shown how to use it (Figure 3-11).

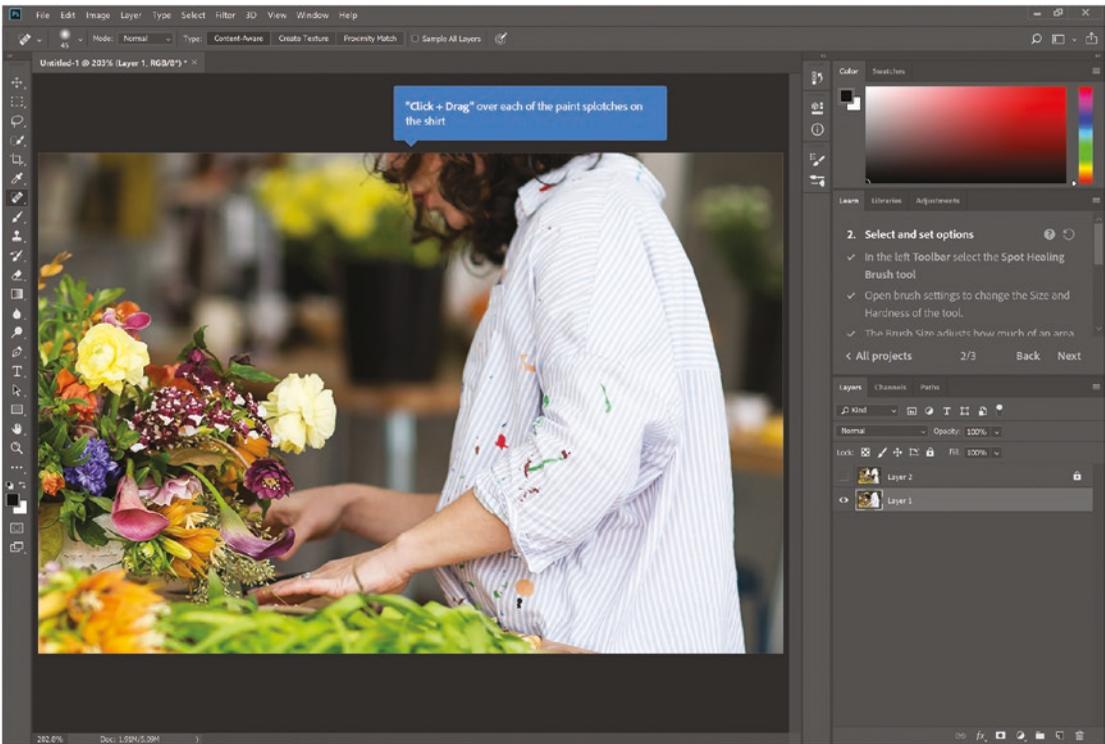


Figure 3-11. Tool tip instructions on how to use the selected tool

This is truly exemplary in-app education, and I think the Photoshop team deserves immense credit for their implementation of it. They've just taken us through an educational experience that first focused on our goals, then presented us with guidance, persisted with us when we decided we wanted to go off and explore, explained *why* we're changing the settings we are, and helped us perform the actions we were trying to do. All in the context of the full, unrestricted interface Photoshop provides. We've processed these concepts on a deeper level because (a) they're related to our goals, and (b) we're given some short guidance before actually using them. We've performed the same actions—in the same context—as we would if we were to do this “for real” with our own image and, quite wonderfully, we've been allowed to explore the interface away from the guidance whenever we felt like it.

Contextual help is a very similar concept, the main difference being it generally highlights single interactions, elements, or structures pre-emptively. The tool tip in Figure 3-12, from Google Docs, provides a good example of contextual help—highlighting the function of the icon while simultaneously showing the keyboard shortcut one could use to perform this action faster in the future.

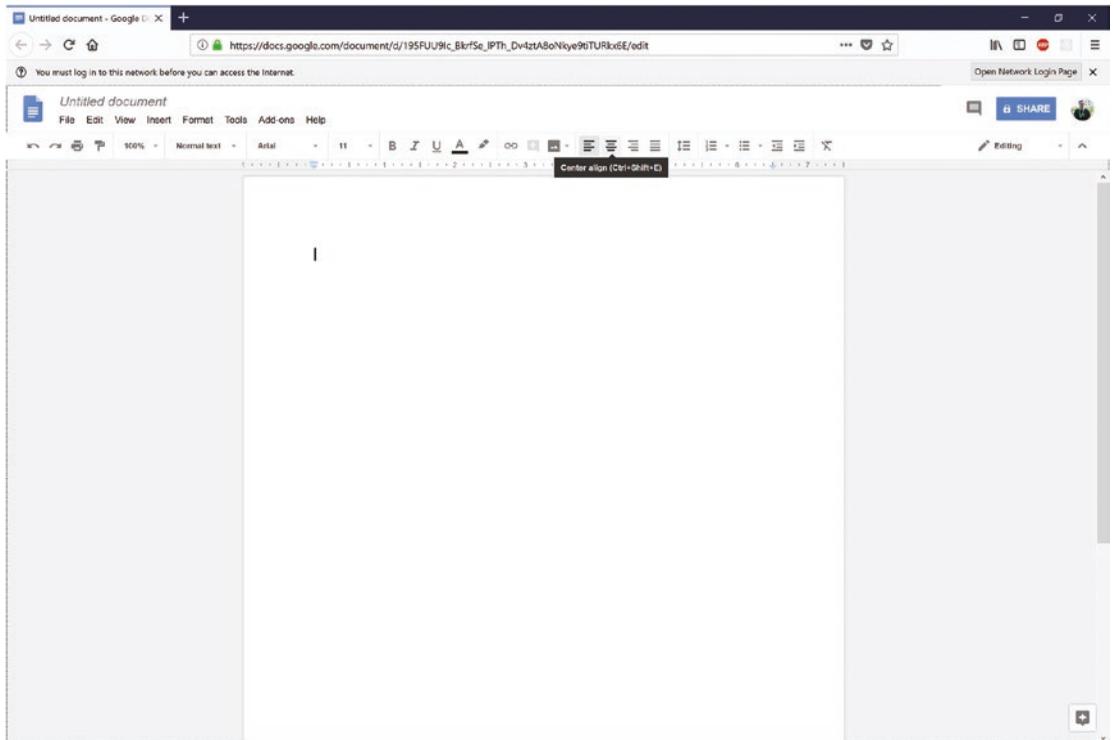


Figure 3-12. Google Docs’ contextual tool tip—describing the function of an icon and also showing the keyboard shortcut to perform this action faster next time

Contextual learning is great for subtle reminders or pre-emptive hints. Using small tool tips, coupled with excellent microcopy, we can provide guidance and feedback throughout our interface without glaring, attention-grabbing practices.

Figma’s onboarding, shown in Figure 3-13, is actually a better example of contextual help than it is effective onboarding. The tool tips provide concise and useful information about the actions that can be performed, but we’re never actually *guided through* the process.

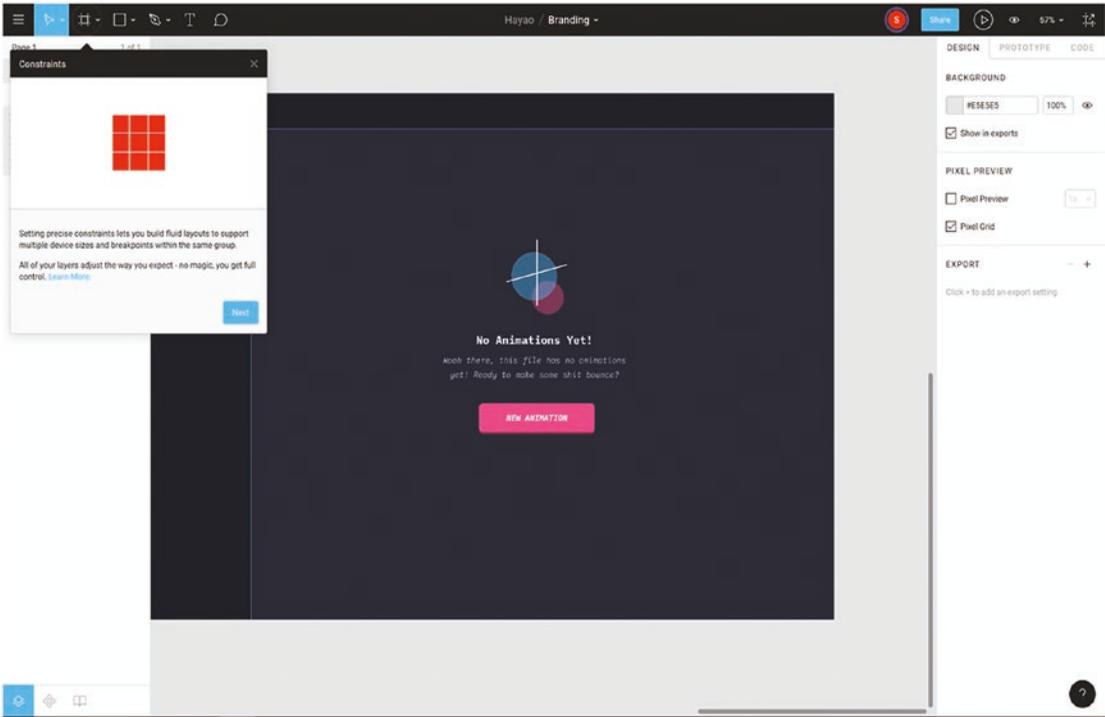


Figure 3-13. Figma’s onboarding, in this case, explaining to us the nature of Constraints

As instructional tools, they perform a good job of showing and describing the idiosyncrasies of specific tools in the interface. The imagery and animations are great, and the copywriting is sublime. However, it falls short as an effective onboarding experience given that we don’t really get to see, or more importantly *use*, the tool in action during this process.

A rather infamous example of the contextual help approach, which I believe feasibly set the “interactive assistant” phase back by at least a decade, is Microsoft’s Clippy mascot from back in the early days of Microsoft Office and its “Office Assistants.”

Clippy is a wonderful example of attention-grabbing and obstruction gone haywire—even if the underlying goals and ideas were commendable. The goal of Clippy was to assist people with everyday tasks, such as writing a letter, by popping up in the corner of the screen if it “detected” behavior that suggested such activity. I’m going to put what’s left of my reputation on the line here: I think Clippy was great. At least conceptually. If the Office Assistants had been designed differently, their process of offering pre-emptive guidance and useful information when deemed contextually appropriate to do so could have

been incredibly useful. As it turns out, Clippy proved too much of a pest, its annoyance outweighing its utility, and it was eventually removed from Microsoft's software, leaving behind only a legacy of annoyance and an entire category of dank memes.

Much of the same principles from onboarding apply to in-app guidance. When designing such processes, the biggest consideration remains the level of interactivity. As with onboarding, any tour or contextual guide that is focused around performing actions, rather than explaining them, represents an optimal incarnation. Our goal for this type of interaction should always be to allow for an ideal depth of processing to occur. Limiting actions, or directing focus to a specific setting or interaction, and allowing that interaction to be performed and observed is often far more effective than using up valuable space (both physical and cognitive) in an attempt to explain it. As we saw with Photoshop's guidance, it *is* possible to provide an effective, enjoyable learning experience while checking all those boxes.

External Teaching

Finally, it's fair to assume that many interfaces will never be fully learnable through automated guidance, regardless of how well designed our onboarding or in-app guidance is. In these cases, external resources can be used to great effect. While creating effective training material is not the focus of this book (there exists plenty of wonderful resources on the subject), it bears stating that well-made external documentation is incredibly useful. Whether this documentation comes in written form, video tutorials, or even face-to-face instruction often depends on the complexity of the system and the affordances of the environment in which it's taught or used. For example, if the system involves a lot of visual elements, such as a design tool or photo-editing application, video tutorials will likely be a great way to educate people. Being able to watch and, importantly, follow along with the use of an application can help us bridge any gaps or eliminate any inconsistencies in our mental models.

External education resources will likely be best utilized when explaining the longest, most-complex tasks of your application. Tasks that might take multiple hours to perform, such as coding up a web site, represent prime candidates for external learning resources. One consideration is how visible to make these resources within our actual interfaces. A common approach that is not without merits is to group all external learning into a "knowledge center" of sorts. This centralized repository of learning materials can be linked to quite unobtrusively in most styles of interface, and it provides a persistent mode of accessing help. "Where can I get help if I am stuck?" is also incidentally almost

certainly going to inform part of our mental model of a system, so a persistent means of accessing help resources is useful. However, we can combine this external information with more contextual guidance and attempt to find a good balance between both options. This is readily achieved by following the previous principles of designing contextual guidance. Rather than attempting to ease someone through actions with tool tips or modals, we can provide a link to specific pieces of our instructional content. When designing complex applications, we should always try and be aware of what educational content we have or what we might need. Doing so allows us to either present it at necessary points or to prepare and inform the creation of new content.

Steps to Learnability

Considering what we now know about mental models, memory, and learnability, we can put together a practical framework of sorts. While the following steps are multifaceted and will often involve a wide range of design considerations, we can at least give ourselves a general, flexible guideline process.

1. Understand our models.

The first step in designing a learning experience is understanding the various models that may be applied. Generally, we'll be comparing two archetypical models: the *designed* model, which we're in control of, and the *perceived* model, which is what an individual forms of our system through their interpretation of our interface. Both of these models represent an abstraction of the underlying system structure—the designed model being our ideal structure and the perceived model being a perceptual manifestation or translation of the designed model.

2. Find the gaps between them.

This is where traditional research methods are our friends. Conducting early-stage usability testing (with, if you remember from Chapter 1, various distractions and simulations of cognitive load) can give a fantastic insight into which areas of your designed model make sense and which cause dissonance or misconception. You'll also likely intuit—or better yet, have market research that shows—the average technical capabilities of your

audience, as well as perhaps the apps and services they interact with on a regular basis. This research should give you hints as to what areas of your designed model are likely to cause the most friction. Anything that breaks the assumed conventions of the apps and real-world tools your audience already uses is going to require some degree of behavioral change. If you can't afford to test in a good environment, using this market research can still give you a good way to understand the potential blanks between your mental model and your perceived model. Make sure to account for the filter of your own biases and that of your team, and only make these assumptions as a last resort.

3. Figure out if the gaps need bridging.

Before actually deciding that the concept we've determined to be a "blank" between our two models needs filling, we should determine if the effort involved, both from us and from anyone who uses our product, is actually necessary. Quite often in our field, we encounter ideas that are creative for creativity's sake—in other words, interactions or concepts that go against convention and expectation for the sole purpose of being unique or impressive. This isn't *always* bad. Shaking things up for the sake of doing so can often lead to some pretty profound results and is a valid form of exploration. However, if we encounter such concepts in our planning, we should absolutely question the necessity of such concepts. Furthermore, sometimes a feature is just too idiosyncratic to communicate with any degree of intuition. In such cases, the importance of this feature should be called into question, and we should never be afraid of completely dropping a feature if its complexity outweighs its usefulness.

An effective way of helping make these decisions at least somewhat objectively is to utilize a map like that show in Figure 3-14. The objective here is to honestly assess your proposed features set and categorize them along two axes: importance and intuitiveness.

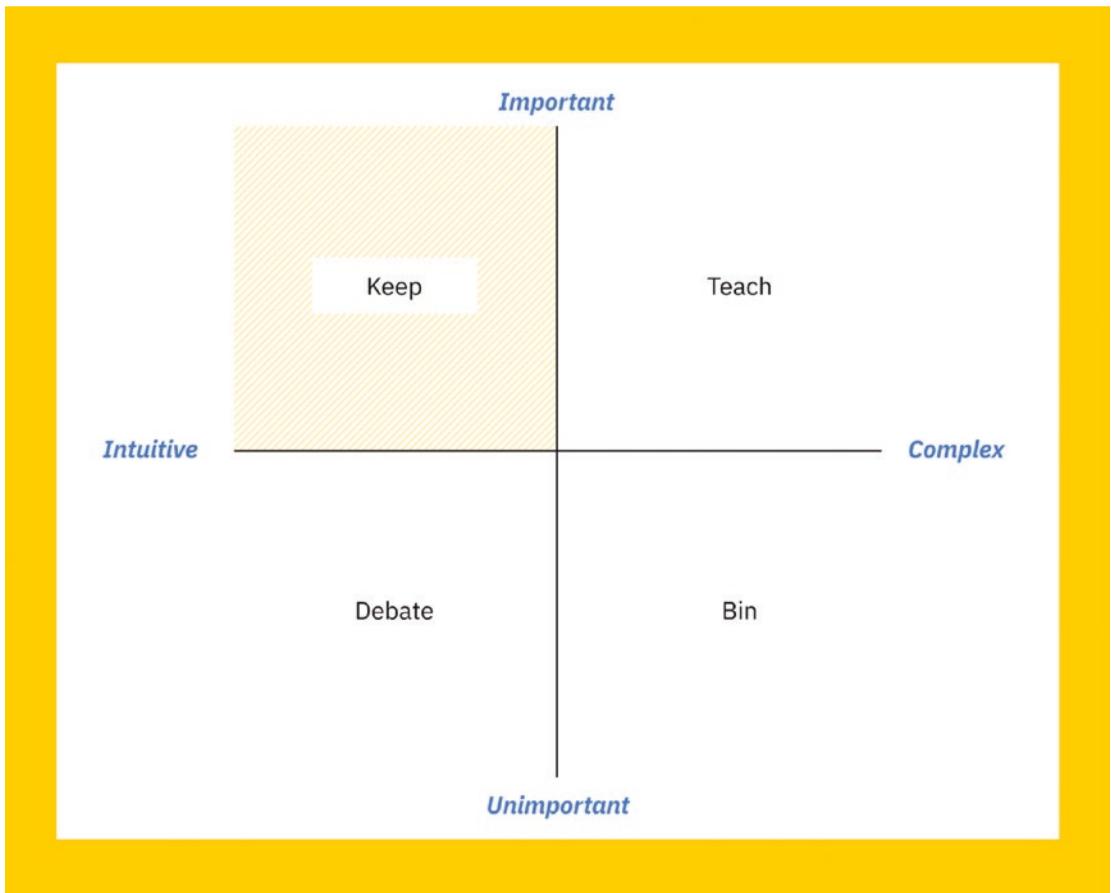


Figure 3-14. *An abstracted version of a feature map*

For the purpose of this kind of exercise, we can view *intuitiveness* as “the closeness between designed and perceived models.” A concept that is both important and intuitive is a win. One that is not important, but still intuitive, should still be questioned. After all, any element or concept we add into our interface can affect the mental effort required to perceive and process it. However, if this feature is something that adds a bit of character or satisfies a stakeholder (office politics are as much a design constraint as anything else, as much as we’d hate to admit it), then it represents a good candidate for use. Important but complex concepts form the basis of our learning approaches. It’s here that we’ll find the features and concepts that we are likely going to need to teach people. Finally, concepts that are unintuitive *and* unimportant should get in the bin.

This is, admittedly, a rather naive and abstracted approach. It's highly likely many of your concepts and features will exist in shades of gray. The point here is to use this exercise as a springboard for discussion and not to overthink the boundaries or strive for exact placement.

4. Understand the type of gap.

One last consideration before we actually jump in and start designing some lovely learning environments—we need to have a grasp of *why* this dissonance between two models exists. Is it a *knowledge* gap of some kind? For example, if we were designing an Agile-focused project-management app (because apparently we're masochists), we need to ask if a chunk of our audience does not have the technical knowledge of how Agile works in order to understand our more idiosyncratic concepts. Or is it a *skill* gap of sorts? While the required mechanical skill to use an interface isn't *usually* taxing, if we're working on a game or game-like product, we may actually encounter this kind of gap between models quite often. A skill gap generally means that some form of practice or repetition is required, and it puts us into the realm of gamification, which is its own beast all together. Aside from knowledge and skill gaps, there are also various *conceptual* gaps. Perhaps your interface handles a common interaction differently from your competitors' or the interfaces your audience are used to interacting with on a daily basis. In this particular case, it's important to show up front that this is as an uncommon pattern, and then highlight *why* it's done differently. Figma has, again, another great example of this. Its vector pen tool is quite different from what you'd get in Figma's competitors. The first few times you attempt to use it can feel quite strange if you're already familiar with the "traditional" approach. Figma pre-empts this with one of their onboarding tool tips (Figure 3-15) by letting us know that "vector networks" are quite different from what we're used to—better, if you ask Figma (and me) —and why we might want to give them a try.

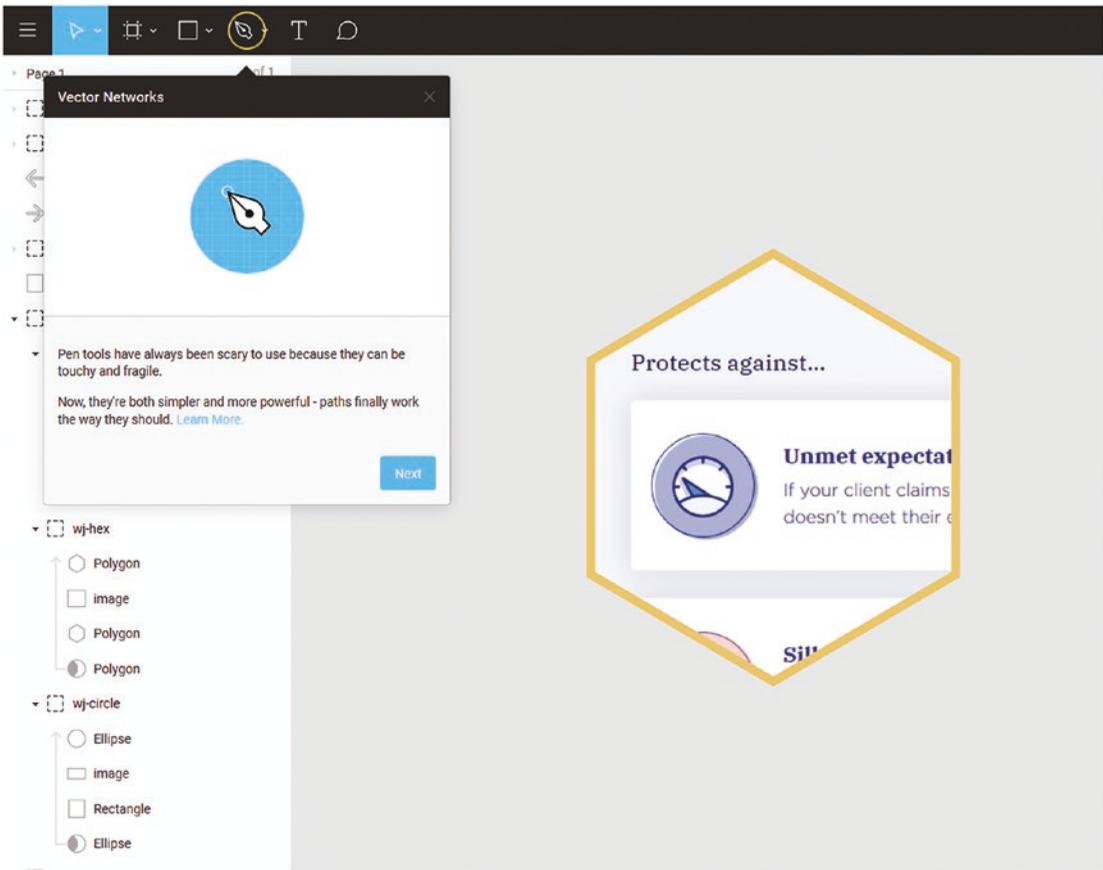


Figure 3-15. A tooltip in Figma introducing us to one of its core features

This not only brings our attention to one of Figma’s core design features, but it also encourages us to give it a try. Again, my biggest gripe with Figma’s approach is there’s no “learn-by-doing: to any of this. I’d love to see some form of guided “draw-some-stuff-with-the-pen-tool” process that satisfied a deeper level of processing, but as pre-emptive communication, this approach has its merits. I’d consider this gap a *convention gap*—a concept or interaction that deviates from standardized convention. Another conceptual gap emerges when we simply don’t have a relevant metaphor or model to apply to a concept. Sometimes, an idea or feature has no real-world comparative and no existing conceptual model—I’d call this a *null-comparative gap*, but only because I want to be credited for inventing a fancy phrase. These gaps are pretty

exciting because they can often be an indicator of innovation. If you genuinely believe the concept you're trying to convey is important and valuable and there truly exists no possible metaphor or accepted convention, you could be onto a pretty important discovery. Or it could just be a really, really bad idea. Sorry.

Quite often, the gaps we recognize will be a combination of these. Knowledge and conceptual gaps especially often go hand-in-hand, and bridging them requires both providing information and allowing interaction. The important thing here is not to spend a bunch of time labeling your potential gaps, but to understand the underlying causes behind them. Having this list sets us up perfectly to delve into designing, and asking the most important questions: *Where* and *when* do we communicate this?

5. Decide on the Where and When.

We've discussed this step in detail throughout this chapter, so hopefully this is the stage where we can start really utilizing and designing around our knowledge of learning and memory. Depending on the feature and the gap that it presents, we'll want to consider what method best allows us to communicate its use and properties. For important-but-complex features that are usually present in the "default" starting state of the application, it generally makes more sense to tackle these in some form of guided onboarding. For features that depend on context or application state, contextual help will likely be a great starting point. It might make a lot more sense to only try and teach specific concepts once the context in which they can be performed has been entered (e.g., Photoshop asking us to change our brush settings only when the Spot Healing brush tool was selected). For more complex, chained interactions, in-app guidance can be extremely useful, especially if it works toward the achievement of a goal. You're encouraged to revisit the relevant sections earlier in this chapter if you want to brush up on any of these learning areas!

6. Design your explainers.

Now we're really at the fun phase. First, a quick recap on what we've done so far. We've made steps to understand our two models, the designed and the perceived. We've documented potential gaps between the two models. We've attempted to objectively assess whether we *should* bridge the gaps. We've made efforts to understand the type of gaps we might be facing and, finally, we've formulated a plan as to where and when we present the learning information. All that's left for us now is to actually design this stuff.

When doing so, remember that learning-by-doing is one of the most powerful methods of learning, and I'd encourage you to, wherever possible, allow for real interaction during your guidance. Exploration will play a role in this too. Quite often you may defer to someone by presenting your concept to them and allowing them to test their assumptions against it. *Always* remember to explore the goals that may underpin these interactions too. If you can link the concept you're teaching back to a possible intrinsic goal, that interaction and its results will be far more meaningful.

This stage is where everything from the previous chapters comes into play—along with your own design skills and the constraints of any style guides or design systems you might have in place. When implementing in-app guidance especially, the Gestalt principles and the von Restorff Effect will greatly inform how one might group a component with its “explainer” and differentiate the explainer from the rest of the environment. These principles go a good way to describing why the typical “tool tip” approach is so ubiquitous in onboarding and guidance design. Our knowledge of attention will also be universally important during this process. As we explored earlier, some creative slowing down can help with learning and memory recall. Although I despise the term, onboarding and in-app guidance can feasibly be described as “attention manipulation” as, in their essence, these experiences attempt to encourage some degree of linearity in our focus. As is the universal caveat, these processes need to be approached

responsibly and with great care for the people we wish to have follow them. Remember the impact that design has on the voice and character of a system. Coupled with excellent microcopy, design can imbue a personality and positivity into our guided interactions that can have a notable impact on retention and recall of information. Finally, and at risk of sounding like a broken record, this guidance should form part of an interface that is operating at a baseline of low cognitive load and clear state communication, and as devoid of overt complexity as is possible. Plastering an already-chaotic interface with tool tips and explainers is rarely to end well.

Putting It All Together

All the previous concepts lead us to construct a pretty thorough set of principles to reference when creating an optimal environment for learning: our environments should aspire to provide an efficient, explorative, learn-as-you-go experience that forgives mistakes and fosters positivity. By using our knowledge of the mind's attentional and perceptive systems, we're able to design an efficient and intuitive environment. In adding these principles of effective learning, we're able to go beyond mere usability and cognitive economy and into the realms of empowerment and improvement. Add in the positivity and emotional power of accepting blame and providing encouragement, and we're able to do all of that and hopefully put smiles on faces in the process. This, to me, is where we start getting closer and closer to the real purpose of design. It's not usability and intuition. It's not aesthetics or character. Nor is it learning and exploration. It's all of those things, and so much more.

Design is the voice and effective model of an underlying system. It's the communication of, and transition between, any of its potential states. It's transferring the abstraction of complex and daunting tasks into an environment that is optimized for their performance. At its very best, great design makes people *feel good* when they perform complex tasks by seamlessly easing them between application states and toward their goals. In making the mundane automatic and the complex digestible and learnable, we can empower people in so many ways and should strive to make this a core tenet of our work.

Summary

In this chapter, we explored the various processes behind human learning and memory. We discussed the idea that an active decision to learn is almost always accompanied by a goal of some sort—learning an instrument so we can play our favorite song, for example. We also discussed the important differences between a “learnable” interface (an interface that is designed to be learned) and an educational interface (an interface that is designed to teach), and we limited our discussions primarily to the learnable. Educational applications often involve a highly specialized approach, usually involving an exploration of the intrinsic motivation behind the decision to learn. The design of such would be (and indeed is) the subject of its own entire book.

We first explored that learning, broadly, can be seen as the transference of information from short-term or working memory into our long-term memory structures. This flow is rarely a simple, linear process. We saw the brain as a network of neurons and synaptic connections, with various neural pathways being activated and strengthened through practice and repetition. Through consistent practice acts and concepts that seem difficult and complex at first can become second nature over time.

The bulk of this chapter was spent discussing the notion of mental models and their resolution against the underlying model of a system. Mental models are abstract representations of how we see the world. Underpinned by our schemas, they represent how we may form our expectations and understanding when interacting with a complex system. Various applications utilize expected mental models to communicate features and functionality. We saw this with the “razor” tool in video-editing suites and the “pen” tool in design applications.

I raised the idea that the learnability of an application is largely dependent on how we reconcile the system’s underlying features against the mental models that one might carry into its use. By finding and acknowledging “gaps” that exist between the actual, system model and a person’s assumed, mental model, we can look to purposefully bridge those gaps—thus bringing the mental model closer to a usable abstraction of the system’s workings. We explored that, to do this, the environments we design should forgive mistakes, provide positive feedback, clearly communicate state, maximize cognitive economy, and allow for exploration and experimentation.

We explored where and when we teach the learnable elements and concepts of our interfaces, including linear onboarding processes, which use techniques such as progressive disclosure to slowly introduce people to the more idiosyncratic interface elements. We looked at in-app guidance, where we saw Photoshop’s fantastic goal-based

instruction. Finally, we discussed external teaching, where we make use of knowledge bases, FAQs, and off-site content to teach broader concepts.

An important consideration when designing learnable interfaces is the general depth of processing that occurs in the learning of an interface element. Shallow processing, often in the form of simple explainer copy or tool tips, can result in poorer learning than deeper processing, such as how Photoshop has us achieving our goals and actually using the elements we're attempting to learn. Shallower processing requires much less cognitive effort, but sometimes, we just need to slow things down and ask for a deeper level of processing. This further highlights the importance of ensuring our interfaces are not full of overly complex, esoteric elements that require deeper processing to learn.

Finally, we discussed a framework of sorts to find and understand any gaps between a system's underlying structure and the broad mental models that people might possess based on our design work. This broadly revolved around understanding the system and mental models, finding the gaps between the two, deciding whether a feature's presence justifies the work required to reconcile any gaps, understanding the type of gap (knowledge, skill, or conceptual), deciding where and when to teach the knowledge necessary to bridge the gap, and finally designing the explainers themselves. We'll explore how this process can fit in around various stages of common design processes in Part 2.

If you only focus on one thing from this chapter, I'd propose getting comfortable with the idea of system models and mental models. Understanding that designers present a conceptual model of a system and that users reconcile their own mental models (and any knowledge they might have of the underlying system) against this is a key factor in designing both intuitive and learnable interfaces.

The early stages of a design process can often revolve solely around activities that are dedicated to presenting optimal conceptual models of a system—treating early-stage prototyping and user testing as opportunities to ensure your underlying models make sense to your audience can be hugely beneficial.

CHAPTER 4

Expectation, Surprise, and the Musical Interface

Music is organized sound.

—Edgard Varèse

While the link between design and music might seem tenuous, forced, or even nonexistent, music has proven to be one of the oldest and most informative media we can study—providing us with insights into areas of cognitive psychology and neuroscience as varied as cultural convention, anticipation, surprise, intuition, and emotion. When we break music down into its constituent components, such as melody, harmony, rhythm, and timbre, we’re able to explore one of the main tenets of art as a function of the mind: its core lies in the organization of stimuli.

Organization is what separates the individual noises that form a fragment of a musical piece from the meaningful, purposeful arrangement of these fragments. Similarly, it is organization that separates the individual brush strokes and colors of a painting from the intricate forms they’re intended to portray, or the discrete, isolated components of an interface from the usable and interactive whole. Furthermore, music is organized based on *relationships* (something we’ll explore in a little more detail throughout this chapter)—whether it’s the relationship between one note to the next to form melody, the relationship between two or more overlapping notes to form harmony, or the relationship between beat durations to form rhythm. Sounds, melodies, phrases, and movements are grouped, based on their relationships, to form an entity that is different to its constituent parts. Just as Gestalt’s approach to grouping and its assumption of a “global whole” inform the basis of our understanding of visual perception, so do musical elements appear relevant in how we perceive musical concepts.

We can explore how music plays on our *expectations* to elicit moments of suspense, interest, surprise, and intrigue. Exposure to particular kinds of musical concepts is an inherent part of cultural convention, with the differences in musical leanings being one of the most pronounced differences from culture to culture. Good musicians understand that by setting a certain baseline—establishing a certain note, phrase, or hook as “home”—deviations and explorations from this baseline can create tension, spark interest, and grab attention. (Hopefully, this is all starting to sound rather familiar!)

Finally, I think it’s important to discuss these concepts *outside* of design. We don’t create in a vacuum, nor should we study, teach, and learn in one—music is such a visceral, important part of our existence and provides myriad ways to marvel at the human mind. I hope you’ll allow me the indulgence of parallelism.

Fourth Dimensional Design

While we’ve touched on the importance of change in interfaces, as well as animation’s role in highlighting and communicating moments of change, we’ve yet to delve into the deeper notion of *progressive* art and media. A lot of “traditional” design education revolves around static representations of ideas—hardly surprising given modern design’s emergence from art and graphic design. However, interfaces are not static. By definition, an interface exists to provide some form of interaction with a system, almost always with a resultant change to the state of that system. While the insight that static art and graphic design provides us is crucial, it only offers part of the equation. To get the other parts we must look into progressive media, which we can view as media that uses *time* as a tool of organization or communication.

When we look at a static work of art or a piece of graphic design, the only thing that realistically changes over time is our *perception* of that work. If we’re being super pedantic, we can argue that the aging of the paper a poster is printed on, or the yellowing of a painting’s canvas, constitutes “change over time,” but realistically, the work is sufficiently static to be considered non-sequential and non-progressive. This static media captures a moment in time and locks an idea or thought down into the then-and-there of its creation—that is, its beauty and its limitation.

On the other end of this spectrum, we have music. While it’s possible to use sheet music or other forms of notation to portray a static *representation* of a piece of music, one cannot consume a musical composition without its occurrence over time. Music is, at least in part, the sequential arrangement and layering of sound. And while this may

seem like an obvious or even trivial notion, the sequential nature of music's organization means that its ideas and messages are communicated progressively. Rather than communicating in a static art form—like a painting—the ideas, message, and story of a musical piece evolve over time. This concept of using time and progression as a tool of creative organization is an incredibly powerful one, allowing us to explore how implicit and explicit divisions of time can be used to give life to sequential and progressive works.

Just like music, movies, animations, video games, comics, and novels, interaction design is not a static medium. While all these media utilize time to varying degrees of explicitness (music's *explicit* "division" of beats vs. comic books' *implicit* use of panels to communicate moment-to-moment transitions) and many of them can maintain their intentions to some degree if time is *removed* from the equation (a still from a movie can communicate a great deal of artistic intent, as can a screenshot of a video game), what they all have in common is that they are enhanced by, or reliant upon, the use and manipulation of time as a creative tool.

When we view time this way, we're quite literally given a whole new dimension to utilize in our work. We can stop looking at our interfaces as various stages of static screens and start looking at them as compositions that are played out over a somewhat measurable period of time, as moments and movements and transitions. Of course, this is nothing new. We've already discussed animations in Chapter 2, yet animations are an abstraction layer, merely a constituent part of the progressive nature of our work. Animations let us describe *how* an interface progresses from one state to another, with the "end" frame of our animation portraying the final state of an interaction. However, on a more theoretical level, time opens up the chance for us to build and release *tension*—to set, manage, and subsequently defy expectations. We'll explore this in much greater detail throughout this chapter because it's an integral part of the parallels we can draw between music and design.

This control of tension and time lets us tell stories through our work, whether these are literal and overt or subtle, implied stories. The approach of allowing our work to unfold and morph over time gives us a great deal of control—control that traditional graphic design has rarely allowed for. The "design-is-storytelling" angle has become something of a cliché, but there's absolutely some truth and value to it. As we discussed in Chapter 3, concepts such as progressive disclosure—the act of revealing different, often more complex, interaction devices throughout the prolonged use of an interface—are only made possible by accepting that our interfaces can, will, and should change over time. An understanding of *when* someone is in our interface, that is, at what point in implicit or progressive time they're currently experiencing it, is an important, albeit abstract, notion we should always keep mind.

Melody, Harmony, and Rhythm

Music, in an insultingly simplified nutshell, can be seen as the sequential arrangement and combination of notes over a period of time. This is one of those definitions that is of no use to anyone, though, and we need to explore some of the basic building blocks of musical composition in more detail to build a working definition.

Melody

When we talk about melody, we're really talking about pitch-based relationships.

Pitch, at a basic level, refers to the frequency at which a sound vibrates, usually noted in Hz. When we hear a note of a certain pitch, followed by another note of (potentially) a different pitch, and this continues over time until a "phrase" is completed, we're experiencing melody. Think of a phrase, by the way, as a musical "sentence." Pieces of music are often formed of many interconnected phrases that lead into and out from one another. Some pieces—especially nursery rhymes and modern pop pieces—are made up of a very few, repetitive phrases (often called "hooks"), while more avant-garde genres, such as improvised jazz, might be made up of hundreds of phrases, which barely, if ever, repeat.

The melody of a phrase is what you might whistle, hum, or sing along to, and it is usually the most memorable component of a piece of music. (This is assuming it actually has a melody. Many forms of sub-Saharan and Western African music, for example, are purely rhythm-based.) Our recognition of melody is, in part, how we can intuit that the three masterpieces that are "*Baa, Baa, Black Sheep*," "*Twinkle, Twinkle Little Star*," and "*A, B, C*" (how we're taught in Western culture to sing the alphabet) are essentially the "same song" with different words. The melodic relationships between the notes of these songs are identical. For example, the first two syllables ("*baa, baa*" or "*A, B*") of all three pieces are sung at the same pitch, with the next two ("*black sheep*" and "*C, D*") are sung at a higher pitch than the first pair. We're able to notice this similarity extraordinarily intuitively. (*If you're lucky enough to have not grown up on a westernized musical diet and have no idea of these nursery rhymes, or if you're just feeling somewhat nostalgic, you can watch two cartoon beavers present a mash up of all three songs at [youtu.be/SHWslMaL3Tc](https://www.youtube.com/watch?v=SHWslMaL3Tc). The Internet truly is incredible.*)

The placement of notes on a staff provides us with a good visualization of this theoretical “sonic distance” between pitches. In music, the “distances” between two notes (in other words, the degrees to which the notes’ core frequencies are separated) are known as *intervals*. In this case, the distance between the second and third note of the piece (G to D) is what is known as a “major third.” Music theory is notoriously esoteric, and there are lots of strange, often counterintuitive ways of labeling or structuring musical concepts. The next few sections will be pretty theory-heavy, but please do not worry if you get a little lost or overwhelmed by some of the musical concepts—it’s okay! We’re more concerned with how music thrives based on relationships than we are with explicit, arbitrary terms, theories, and rules. However, a good understanding of how Western scales are formed and learning some of music’s more general terminology when discussing relationships will help us discuss these concepts going forward. Deep breaths.

In most forms of music, there are 12 notes, most commonly given letter values between A and G, with a few “sharp” or “flat” (denoted as \sharp and \flat , respectively). Thus, variants of these notes are written as follows: A \sharp , C \sharp , D \sharp , F \sharp , and G \sharp (which are functionally equivalent to B \flat , D \flat , E \flat , G \flat and A \flat). This rather esoteric approach gives us a full octave of A, A \sharp , B, C, C \sharp , D, D \sharp , E, F, F \sharp , G, G \sharp , and A, completing a theoretical circle. (A sharp, or flat, note, by the way, is its own, discrete note. An A is no more or less related to A \sharp , in any functional sense, than an E is to an F.)

The distance between the first A and the last A is known as an octave. Playing these notes in sequential order produces a *chromatic scale* built on A. The distance between any note and its direct neighbor is known as a semitone, or a half tone. Various other note-labeling methods exist, too. If you have ever been in a choir, you may have learned to practice your singing scales with “Do Re Mi Fa Sol La Ti Do,” which was used traditionally across many central European countries. Can you tell yet that musicians and music theorists are sadists who thrive off ambiguity and confusion? Figure 4-2 shows a set of piano keys with these notes and a few example intervals labeled.

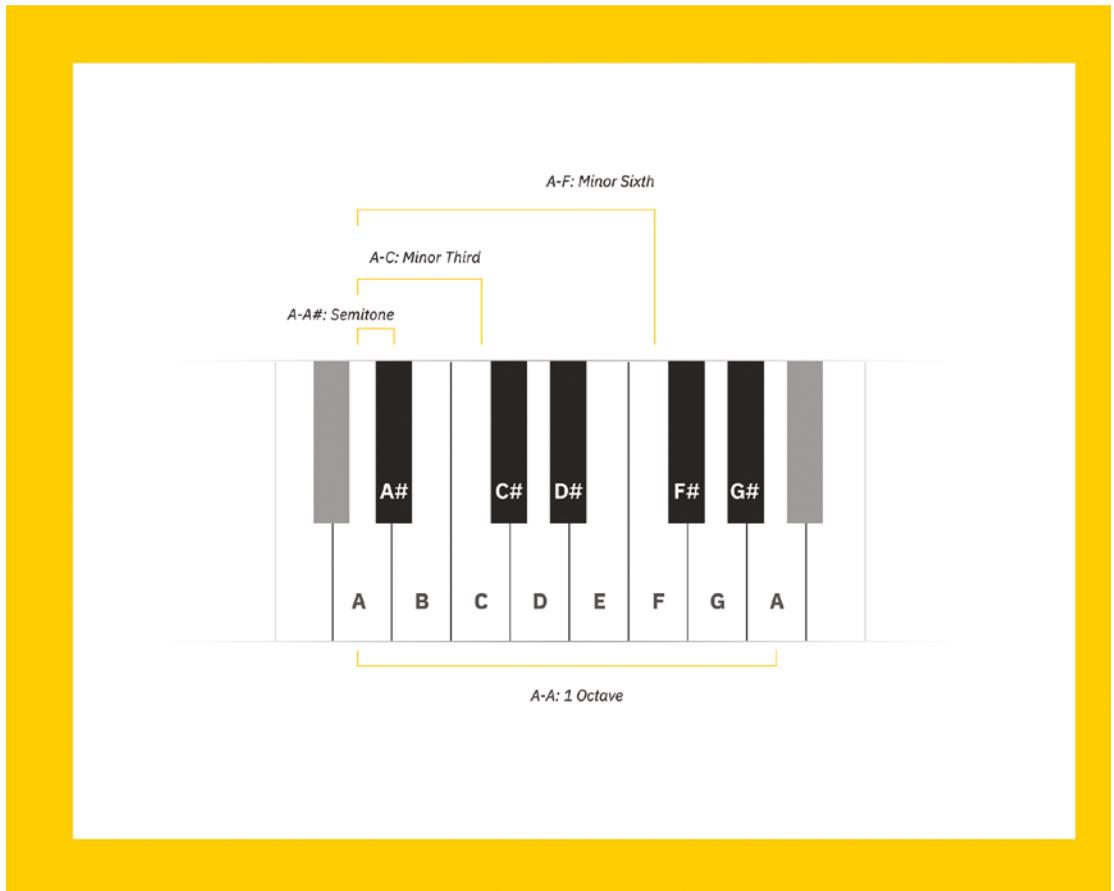


Figure 4-2. Music notes and intervals associated with piano keys. The distance between two adjacent notes is known as a semitone, or half step.

If you’ve ever played (or observed someone play) piano, you may have noticed that as you move from right to left along the keyboard, each note you play is higher than the one that came before. The full “journey” from the first A to the last A is our chromatic scale, and the distance between the first A and the second is our octave. The key points are that we have 12 notes to work with and the distance *from* one note *to* another is called an interval.

Very rarely, however, will a piece of Western music make use of all 12 notes. Western music is traditionally “tonal” music, which means pieces have a “key.” The key of a musical piece indicates a couple of things. First, it indicates what is likely the “home” note of a piece, and, second, it communicates which notes are “legal” members of that key.

So, a piece in the key of C Major tells us that C is the likely “home” note (more on this later) and the “legal” notes of the piece are limited to those found in the C Major scale. G# Minor tells us that G# will feel like home, and the legal notes are those of the G# Minor scale. A scale is a collection of notes, separated by defined intervals, that communicates a certain mood. In C Major, those notes are C, D, E, F, G, A, and B. If you’ve ever seen a jam band play, and they decide they’re going to improvise in the key of C, then aside from some “flavor” notes (known to music theorists as “accidentals”) here and there, they’re likely to restrict themselves to phrases constructed from those notes listed.

Rather than memorizing each note of every possible scale, though, musicians tend to learn scales based on their intervals. So when we look at C Major, we see that the distance from C to D is two semitones (or a “whole step”), as is the distance from D to E. From E to F, we have a “half step.” Then it’s whole, whole, whole, and half to give us the structural sequence of *whole, whole, half, whole, whole whole, half* to define a major scale. Now, if we start at any other note and follow this structure, we will get the major scale of that note. So G Major would contain G, A, B, C, D, E, and F#. By learning the underlying structural sequence of a major scale, musicians can essentially play or write in any key without learning 12 different note sequences. This only works because, in music, relationships rule all.

While explicit, provable knowledge of these aspects of music theory is likely useless to a non-musician, the fact that these rules govern the vast majority of Western music means that we *hear* and *expect* these rules to be followed as part of our intuitive understanding of music in general. Indeed, the key thing to take from this theoretical rambling is that these limitations and musical rules, based around sonic *relationships*, are what form our subconscious understanding of music.

When we listen to a melody, we’re hearing relationships between notes—and not just from each “pair” of notes to the next, but in the context of the entire melody so far. When we perceive a melodic phrase, every note, every core “part” of that phrase, turns the “whole” into something else—something it categorically *was not* before the introduction of that note. Again, without sounding like a broken record, this is the core tenet of Gestalt psychology.

When we talk about intervals, we’re documenting a specific relationship between frequencies of a sound. A “minor third” interval (for example, going from A to C) describes the relationship between these notes in a particular way—it communicates the *sonic distance* between the root frequencies of each note. For example, if an A note has a

frequency of 440Hz, its relationship to other notes can be described in the form of a ratio. It could be related to an E at 660Hz, or a ratio of roughly 3:2. It could be related to a C at 264Hz, or a ratio of 3:5, or a D at ratio 2:3.

Without delving into the *Da Vinci Code* levels of conspiracy and myth-busting that one invites upon one's self when discussing such matters, you might notice that these ratios all relate to the Fibonacci Sequence, which, in turn, relates to the Golden Ratio and Golden Spiral. It's also worth pointing out, at least for novelty values, that many of the typographic scales we use are based on the ratios provided by musical intervals. Figure 4-3 shows www.modularscale.com—a web site by Tim Brown and Scott Kellum that allows one to generate typographic scales based on a huge number of ratios.



Figure 4-3. Modular scale lets you choose “harmonious” typographic scales

With all the options aside from one (the Golden Ratio, funnily enough) being based on musical intervals, such as minor second, perfect fourth, and so on, modular scale shows an interesting and potentially useful direct application of the pure mathematics of music intervals to design.

Fundamentally, melody and phrasing are the driving forces of many forms of Western music. While genres such as ambient and noise focus on textural (or timbral) and reverberative qualities, and many forms of hip-hop eschew melody for rhythm (Public Enemy’s “Bring The Noise” is a fantastic example of how melody does not have to be the be-all and end-all of musical composition), it’s extremely difficult to find music that doesn’t revolve around melodic movements.

Harmony

Luckily for us, harmony and melody utilize the same underlying concept—intervals—just along a different axis. If we define melody as the intervallic relationship between *sequential* notes (for example, playing one note, *then* playing another some arbitrary time after the first), then harmony is the intervallic relationship between *concurrent* notes—that is, notes played *at the same time*.

In music, a combination of two or more notes played (usually, but not always, on the same instrument) at the same times is called a *chord*. If you’ve ever picked up a guitar, you might have learned a G chord early on in your musical career, and then gave up, maybe, because fingers weren’t designed to work in such stupid ways. This chord would have had you playing multiple notes, across multiple strings, at *roughly* the same time (fast enough to sound like one discrete note rather than an *arpeggio*). As with melody, the relationship between all the notes of a chord informs the harmonic structure of that chord. By layering specific intervals on top of each other, we can form chords with specific characteristics. The *root + major third + perfect fifth* combination, for example, gives us a major triad chord (“major” defining the intervallic relationship between the root and the *third*, “triad” because it’s formed of three notes). Changing a single interval by a single semitone in that equation gives us the combination of *root + minor third + perfect fifth*—or a *minor triad*.

Major chords—due almost exclusively to cultural convention—connote triumph, happiness, and irreverence, while minor chords can imply sadness, seriousness, and melancholy. One change to one interval, and we’re able to completely change the qualities of a chord. Of course, as we discussed when exploring melody and previously when discussing Gestalt, the *relationship* between multiple chords provides much more harmonic information than taking a single chord out of context to analyze its qualities. For example, minor chords show up *all the time* in otherwise “happy” songs, as do major chords in the melancholy landscapes of “sad” music. Without the context of a

piece's other chords, a single, out-of context chord—while containing more intervallic relationships than a single note—is still a relatively abstract and arbitrary fragment of a musical idea.

Just as melodic phrases occur over time, so too do the underlying chords that form the tonality of a piece. While the sequential transition from one note to another forms our melody, the movement from one *chord* to another is known as a “chord progression.” Chord progressions allow us to explore what is probably the most relevant concept we can borrow from music: the building and releasing of tension.

The concept of a “home” chord is something that almost every traditional piece of Western music will utilize. By communicating a certain chord as the tonic root of a piece, we essentially establish that chord as a musical base camp—returning here “feels right” (very scientific, right?) and any deviation from it creates some degree of tension or intrigue. There are myriad ways to make a chord “feel like home,” ranging from the not-so-technical “play it first, play it loud, and play it often” to more nuanced and subtle techniques. However, the core goal is to establish this tonic root. Once this root is established, we can move away from it to create tension, intrigue, and suspense, and return to it to resolve these feelings. *How*, and to what degree, we deviate from this home chord, as well as how long we take to return to it, greatly affects the tonal characteristics of a piece of music. Once we've got one more pesky definition out the way, we'll delve much deeper into the underlying emotional functions that permit this approach to tension.

Rhythm

Rhythm is music's explicit use of time, and it defines how much time a particular note or chord is played for as well as where any emphasis is placed. Rhythm can be thought of, in part, as the punctuation of a piece of music. While the notes of a melody might be our word choice and each phrase a sentence, rhythm is the essential punctuation that allows our syntax to breathe. Rhythm is what has us tapping and dancing along with music. When we read a piece of sheet music, we generally encounter *bars*. Bars, otherwise known as *measures*, are (somewhat arbitrary) ways of splitting a piece of music into phrases or sections.

Going back to “Baa, Baa, Black Sheep,” if we look instead at the rhythmic structure of the song, we can see that the first bar (*Baa, baa, black sheep*) consists of four notes of equal duration. In the second bar (*have you any wool*), each individual syllable until

wool are sang twice as fast as the four notes in the first bar, and the final *wool* is held, making it twice as *slow* as those in the first bar. Try slowly tapping these beats out if this labored written description is hard to grasp!

Without delving too deeply into rhythmic theory (as much as I'd love to), the first four beats of the song are called "quarter notes," or *crotchets*—a beat that takes up one-quarter of a bar (if that bar is in 4/4 time). The *have you any wool* beats, twice as fast as the quarter notes, are known as "eighth notes," or *quavers*. In a single bar of 4/4 (*again, if this doesn't make sense, don't worry—4/4 is a musician's way of saying "each bar of this piece contains the equivalent of four quarter notes"*), it'd take eight quavers to fill a bar. The final syllable of the second bar, *wool* (or *woooooooool*, to be precise) is what is known as a "half note," or a *minim*. This takes up as much time as two quarter notes, or four eighth notes. When we add our fractional notes together, for each bar, we get a nice, round one (four quarters in bar one, four eighths + one half in bar two).

The division and duration side of musical rhythm only forms part of the equation, however, because without *emphasis* to reinforce a piece's rhythmic qualities, we'd be left with a rather bland, plodding piece of music. Emphasis determines where we should place *accents* in a piece of music, giving it a distinct, recognizable pulse. Try singing "Baa, Baa, Black Sheep" out loud, while paying specific attention to which words you "naturally" emphasize (probably by singing them a little louder than the others). If you're super talented, try even tapping your feet along with singing. (Don't worry if you're reading this book in a public place! Do it anyway. You'll be the winner.)

Now, it's quite probable that the biggest emphases in your sing-along were placed on the first *Baa*, and the word *have*. It's also quite likely that you added a less-obvious emphasis to the words *black* and that last, prolonged *wool*. Now, this doesn't account for if you were feeling particularly rebellious or fancy with your rendition, nor does it accommodate for your having been taught that song by someone rather fond of syncopation, but the vast majority of us will "naturally" add emphasis in these places. This is due to our constant exposure to this prototypical rhythm forming part of our "rules" for Western music.

The overwhelming majority of contemporary western music is in a 4/4 time signature. This means that each measure of a 4/4 piece of music is made up of beat divisions that add up to four quarter notes. Furthermore, the overwhelming majority of songs in 4/4 time place primary emphasis on the first quarter note of every bar and secondary emphasis on the third quarter note. This combination of beat divisions and emphasis provide the rhythmic "feel" of a piece of music. In emphasizing the "right"

beats of a bar, such pieces can cater to our musical expectations. As you may have noticed in your earlier rendition, this emphasis feels ingrained into our brains—due to such an extensive, lifelong exposure to “4/4-with-an-emphasis-on-the-1-and-the-3” music, anything that differs from this can initially feel weird. One of the most common contemporary deviations from this rhythmic approach is found in reggae music. Throw on a popular reggae track (I recommend Peter Tosh’s “Legalize It,” if you’re short on reggae knowledge—just please don’t choose UB40) and try to tap along to the rhythm.

If you noticed you had a little difficulty matching the emphasis or rhythmic pulse of reggae music, that’s likely because it uses *syncopated* rhythms—the emphasis in reggae usually hits on the second and fourth beat of each bar. This is what gives reggae music its signature “bouncy” vibe. It’s still predominantly a genre based around 4/4 time, with contemporary Western instrumentation (guitar, bass, drums, and vocals), but simply changing the pulse of the beat gives it an “exotic” feel—at least in comparison to the average Western pop song.

Tension, Anticipation, and Surprise

If there’s something I’m positively certain you’ll agree with me on by now, it’s that “Baa, Baa, Black Sheep” is really, horrendously *boring*. There’s a reason such simplistic pieces are saved for young kids and meandering, example-driven music explainers in the middle of design books: it’s so stripped-down and basic that it’s almost completely dull. Which leads nicely into the fun part of dissecting music—the medium’s insightful and incredible application of emotional engagement through tension, anticipation, and surprise.

In order to explore the underpinnings of emotions like surprise, we must first understand how we build expectations of the world around us. One of the most elegant methods of explaining this is through the notion of *schema*.

Schemas

In Chapter 1, we spoke about the mind’s need to categorize. As a result of categorizing, we create and reference various *schemas*. A schema is a mental structure that consists of “rules” against which we can judge objects or stimuli from our environment. Schemas are similar, at least in definition, to the mental models we covered in Chapter 3. They both represent structures of the mind that allow for relation, abstraction, and categorization.

To hopefully disambiguate when I talk about mental models in this book, I'm talking specifically about a person's perception and understanding of an underlying *system*. Contrast this to schemas, which, in this context, represent a broader set of rules, or features of a category, that we may later use to classify something as a member, or nonmember, of said category. For example, our schema for *car* might include "has wheels, has seats, has a steering wheel, and has an engine." This approach to categorization allows us to quickly make decisions and predictions of objects and stimuli in our environment, based on things like physical appearance, tactile response, and auditory input. Furthermore, by developing rules and expectations for categories, we also appear to form *prototypical* members of categories based on our schema.

As part of an important and intriguing series of experiments around categorization starting in the 1970s, Eleanor Rosch presented and evolved her Prototype Theory. The Prototype Theory suggests that, rather than simply having a binary yes/no determinant as to whether or not an object or stimulus is a member of a category (it either meets the rules, or it doesn't—also known as the Aristotelian Model), we adopt the notion of *prototypes* (Rosch, 1973)—basing category membership on comparisons to central members that most closely adhere to the schema of our categories. Rosch's examples include the category of *furniture*, where it was shown that objects such as chairs and sofas are deemed "more central" (Rosch, 1975) to the furniture category than, say, stools or lamps. Thus, we can position chairs and sofas as "privileged" members of the category against which membership can be weighed for less prototypical, potential members.

The implication of this approach to graded categorization was substantial. The idea that categorization is central to our understanding of the world is rarely disputed, but the idea that we use and compare *members* of categories to help determine membership presents some fascinating insights into our nature. Particularly, it goes some way in explaining why our existence and understanding of the world is so inundated (and often plagued) by stereotypes and archetypes. By using prototypical members of a category as a heuristic determinant as to whether the thing we're analyzing "belongs" in that category, we invariably form—often incorrect, and often damaging—stereotypes for members of that category. While instantly jumping to a small chirpy sparrow or robin as opposed to a big, gangly ostrich when asked quickly to "think of a bird" isn't a damaging worldview at all (unless you're an ostrich), jumping to the image of a young woman when asked to "picture a nurse," or an old, white man when asked to "think of a scientist," for example, really is.

Schema and prototypes appear to show that our boundaries for categorization are not immutable rule sets, but fuzzy, evolving concepts. For a long time, the schema most people held for “woman” would seemingly only encompass biological females—a hugely problematic mindset that still, depressingly, exists in many people and subcultures in modern times. While many people are changing their schemas as to what constitutes an apparent member of a gender—the fact that transgender women *are women* should raise zero questions about transgenderism, and infinite questions about how fallible, arbitrary, and hurtful accepted schemas and stereotypes can be—and many still are rightfully questioning the binary nature of a gender paradigm in general, we must accept that this societal change is required, in part, due to our blinkered willingness to categorize in such ways. Our brains are, evolutionarily, quite outdated—and this is just one of many examples of how schemas and heuristic can hamper social progress. This does not, however, excuse or diminish bigotry and myopia. It merely serves to offer a potential explanation for some of their root causes.

This also suggests that our schemas are greatly informed by exposure to sociological events and cultural convention. When something challenges our schemas, we’re confronted with a situation that likely raises many questions, causes cognitive dissonance, and quite probably affects us emotionally. This is especially true if the schema in question is one that we’ve held for a long time, one that relates to a category that is of notable intrinsic importance, or one that underpins many other schemas that form the scaffolding for our worldviews. Many people, too, possess schemas that they believe to be immutable, resulting in a form of schematic myopia. Depending on the category in question, we might call these people “snobs” (for instance, jazz and classical music obsessives that refuse to acknowledge heavy metal or dub step as “music”) or even “bigots” (prejudiced individuals who appear devoted to their myopic opinions).

Most non-obsessive, non-bigoted humans, however, will possess schemas somewhere close to the middle of a mutability scale. That is to say, the boundaries for various categories are, to a degree, moveable and changeable in the face of evidence and experience. As well as being able to adapt and alter the underlying schema we hold for a category, prototype theory suggests that we also have “fuzzy” members of a category. Rather than a binary member/nonmember deduction, we have the notion of something being more, or less, of a category member, depending on its resemblance to a prototype and its adherence to the rules of our schema for said category.

Schemas, for better or worse, appear to form the basis of how we categorize objects and stimuli in our environment. However, another key area of schemas—and the area

that is pertinent to the discussion of concepts such as tension and anticipation—is that they greatly inform our *expectations*. This is something that musicians, knowingly or otherwise, take advantage of *all the time*. Remember the idea that music is organized sound? Part of that organization revolves around knowing and manipulating the expectations we've formed of music as a whole. Our schema for music is based on what we're exposed to throughout our lives, and it includes implicit rules for rhythms, melodies, and harmonies, as well as more esoteric expectations such as instrumentation and dynamic range.

When you sang “Baa, Baa, Black Sheep” earlier (you did sing it, right?) and I asked you to pay attention to where you placed your rhythmic accents, it was almost inevitable you'd been “taught” that nursery rhyme with those specific rhythmic emphases. In that sense, these simplistic nursery rhymes can be seen as prototypical of Western rhythmic structure. Our schema for a particular genre's rhythm, or even the entirety of musical rhythm known to us, is informed by our prolonged exposure to this one very specific approach to musical composition. This rhythmic structure, then, represents one of our core schematic expectations—an assumption that we take in to any scenario of music listening.

As we explored, the *vast majority* of western music has this rhythmic pulse to it. The first beat of every bar is usually the most energetic, followed by a generally low-energy beat, then a notable emphasis on the third beat preceding another low-energy beat to finish the bar, repeat ad nauseum—*DUM-dum -DA-dum, DUM-dum-DA-dum*, and on, and on. This rhythmic pulse can be heard in The Beatles' “*Yellow Submarine*,” in Black Sabbath's “*Paranoid*,” Beyoncé's “*Love On Top*,” and Kendrick Lamar's “*King Kunta*.” It's *everywhere*, regardless of genre or tempo, and a whole host of artists spend their entire career barely deviating from it. This pulse, coupled with non-changing 4/4 time signatures, could also quite feasibly represent the *only* approach to rhythmic composition the average listeners will hear in all the music they listen to. Through this kind of exposure, our schemas for many forms of music are littered with this rhythmic pattern.

Now think to how reggae, which is far from an avant-garde form of music, sounds somewhat “strange” or “exotic” when compared to the prototypical Western pop song. The instrumentation, timbre, melodic and harmonic constraints, and song structure of a prototypical reggae piece are extremely similar to those found in Western pop music, yet that one single shift in rhythmic emphasis defies our expectations and gives the genre its own unique vibe. Quite plainly, we're experiencing a form of schematic violation

when we encounter this. Western audiences have, over many years, built up an *idea* of what music is, and when faced with something that doesn't fit that idea, we're exposed to something *new*, often *challenging*, and sometimes *surprising*. This challenge, or schematic violation, provides a degree of necessary complexity to musical stimuli. By going against what we're used to, by defying our expectations, artists can provide us with novel and important stimulatory challenges.

Given that reggae mostly makes use of timbres, instrumentation, meter, and structures that Western audiences are used to, the syncopated rhythms provide just a slight deviation from our expectations. If we were to, say, replace the electric guitars found in many reggae pieces with a Japanese shamisen and swap the vocal melodies over to a West African balafon, we'd be further deviating from the "safe" expectations of a Western audience's schema for music.

Our exposure to instrumentation also plays a major role in our schema for music—especially the timbral qualities of the instruments we're *used* to hearing. While we haven't explored timbre much in this chapter, it is essentially a combination of harmonic overtones produced by an instrument that gives it its *sound*. A G# played on a piano sounds different from the same G# note played on a saxophone, for example. Our schemas for classical music might include typical orchestral instruments such as the piano, cello, violin, and French horn, while our schema for folk music will likely include the acoustic guitar, double bass, and fiddle.

When distorted electric guitars were first introduced to music, it created quite the stir. Many people were taken aback, even offended, at this fuzzy, distorted noise. In an age where acts like Tony Bennett, Dean Martin, and Frank Sinatra provided the sound of the times, this high-energy, visceral noise—with its jarring, micro-dissonant overtones—was quite the contrast.

Many others, however, were enamored. Intrigued and inspired by this sound and their sense of adventure aroused, a cultural revolution emerged and the chase for more distortion, louder amplifiers, and heavy, guitar-centric music started. From the early emergence of rock-and-roll to the high-energy, riff-based approaches of Deep Purple, Black Sabbath, and Led Zeppelin to the anarchist punk stylings of The Stooges and the Clash to the energetic hardcore of Black Flag and Bad Brains and on and on, entire genres of music and cultural movements emerged based around, as Frank Zappa wonderfully puts it, "the disgusting stink of a too-loud electric guitar." Similar movements occurred around the invention and popularization of audio synthesis, with many electronic, synth-driven artists going against the "real" musicians of those days, or

the grunge musicians of the late 1980s and early '90s taking rock and metal music away from the flashy, Van Halen-style, shred metal to a visceral, bare-bones genre. For every one of these emergent countercultures, there were many listeners rooted in their fixed schema, rolling out phrases such as “it’s just noise,” “that’s not real music,” and “I really wish Frank Sinatra was the only person ever allowed to make music” as thinly veiled excuses for schematic myopia.

These do represent notable deviations from popular schematic expectations, however, so it’s easy to see why many people rejected the concepts outright (and may go some way to showing why younger people tend to “get” new, different music easier and faster than older—they’ve spent less time validating their own myopic schema). However, this challenge to our preconceived notions of what constitutes a genre of music, or even music itself, is considered to be a major factor in music’s evocative and emotional underpinnings.

Tension as Schematic Violation

As we’ve explored, our schemas help inform our expectations. While being able to analyze our concurrent stimuli and make a logical prediction has worked wonders for our evolutionary survival, when it comes to processing noncritical stimuli (which, from modern, privileged positions, represents the majority of daily processing), we’re very easily bored by things that adhere too closely to our expectations. Movies rely on this all the time. By having the audience expect one thing (and often playing to that expectation) for the first acts of a movie, only to slowly unravel a conspiracy that completely defies that expectation, a much more exciting, memorable plot is played out. By carefully presenting us with enough information early on to form a theory or expectation, only to take us in a completely different direction, good movies create engaging, challenging experiences—and a decent plot twist or two can salvage even an otherwise terrible movie.

Music metaphorically throws plot twists at us all the time. And rather than taking two hours to do so, it often does this in a matter of seconds. When we hear music, especially if we’re paying direct attention and actively listening to a piece, our brain very quickly starts thinking of where the piece or movement might end up. Patterns start to emerge in the lines and phrases we hear, too, satisfying an innate desire of the auditory cortex to organize what we hear into safe, known, and expected patterns. As we explored when discussing harmony earlier, a major component of expectation in music lies in establishing a “home” chord or note, and traveling away from and back toward it in

various ways. This very act is made possible because we have expectations as to where a piece of music will go next, and this is constantly updating as we parse the sequential stimuli of a piece. Great composers will use this to dance between moments of intrigue (defying our expectations) and reward (the acknowledgment that we were “right” or the reversion to a harmonious baseline emotion). Another way at looking at this flow between intrigue and reward, using more “musically appropriate” terms, would be the concept of tension and resolution.

Tension in music is much like tension in many forms of progressive and sequential art, and it essentially boils down to these idea of flirtation with schema and pattern. Tension relies on using these ideas to create a kind of “useful complexity,” whereby we’re processing something that deviates from our expectations in novel ways—ways that suggest, retract, and reintroduce patterns, challenging our preconceived schemas and thus mentally arousing us. Musical tension is often achieved through first establishing this baseline of “home.” By framing and emphasizing a certain chord, note, or pattern, a composer can sow the seeds of expectation. What follows is generally a power dynamic wherein the listener, vulnerable and at the whims of the composer, allows their expectations to be teased and their emotions manipulated as the piece moves further away from the nice, established patterns of home and into stranger, schematically inconsistent territories.

The fulfilling of expectation in this process can be viewed more generally as the reinforcement of a schema, either through what we understand of music or genre as a whole, or through what the piece itself has communicated about its *own* baseline structures and patterns. When we imagine something like a piece of music to have rules and expect it to follow them, we’re left with a sense of reward when it does so: we’ve successfully predicted the direction of a piece; we’ve finally rediscovered the pattern we so craved.

Rules that are set early and adhered to often create this feeling of “correctness,” as can be seen in any song that has a “count-in” intro. Count-ins are one of the most obvious ways possible of setting expectations, explicitly calling out the rhythmic structure of the song before it begins. This can be heard in songs like Outkast’s “*Hey Ya*” and James Brown’s “*Get Up (I Feel Like Being A) Sex Machine*.”

Count-ins are common in live performances, where a member of the band (usually the drummer) counts “out loud” (either literally counting the numbers or tapping the rhythm on a drum or cymbal) a sample bar to give the rest of the band an idea of the tempo of the piece and an indicator for when to start playing. However, they serve a

secondary purpose, especially in music with a heavy focus on rhythm and movement, by setting the expectations of an audience. James Brown was a master of this, both explicitly (as heard in “*Sex Machine*,” “*The Boss*”) and implicitly (“*Get Up Offa That Thing*”) introducing the song’s rhythmic “rules” and setting expectations before the full band comes in. This works so well because James Brown’s music exists, primarily, to be danced to. If you can’t listen to James Brown without risking a little boogie, it’s highly likely you’re an *extremely boring* person.

Dancing and general outward displays of emotion and movement form a huge part of music. A notable amount of our brain’s response to music occurs when processing rhythm. There’s a direct link between music listening and the cerebellum—the area of the brain associated with movement, and it’s been shown that our brain’s rhythms actually sync in time with musical beat. Any physical response to music is largely taught out of us, with many Western cultures preferring the polite, white-bread response of simply observing musicians perform without much engagement. From seated, civil viewings of classical performances to that boring couple you know who pay to see Eric Clapton play and might potentially risk a head nod or two if no one’s looking (don’t deny this, everyone has these people in their life)—the white, middle class have long been taught to hold back any natural desire to move along with a piece of music.

When the late, great James Brown counts-in “*Sex Machine*,” he sets up a few expectations. First, by counting to four, he shows that the song is in 4/4 (unless it’s a waltz, 4/4 is the time signature of almost every “danceable” piece). We’re used to this. This is the time signature of most music. It’s an ingrained part of our schema. Second, the time between each number in the count tells us the tempo—how fast the song is going to be. This, very quickly, has us forming a schema for the song. We *expect* that the song will, at least at the beginning, adhere to these rules. Before we’re presented with the “bulk” of the musical experience, our auditory cortex already has the beginnings of a pattern.

Now, some artists, especially those who thrive on being avant-garde and unpredictable, might see this as an opportunity to immediately defy expectations, either by delaying the entry of the instrumentation (Frank Zappa did this *all the time*) or by immediately changing the tempo or meter. However, James Brown and his band played music that got people moving. To grossly and immediately defy expectations in such a way would be counterproductive, leaving an audience’s expectations (and movements) out of sync with the musicians’ performance. By setting and sticking to expectations for

rhythm, James Brown’s music gives us a sense of trust. It’s validating to perceive stimuli that we can predict, especially when those stimuli are designed to have us moving and expressing emotion.

Predictability, however, is often boring, and many areas of James Brown’s work defy expectations in their own way. While the key tenet of the music—the rhythm—is predictable and often maintained, the amazing range of vocal timbres, the masterful ebb and flow of dynamics, and the wide gamut of instrumentation and melodic ideas heard within Jams Brown’s works make for a varied and eclectic discography. While there’s little in Brown’s hits that violates “traditional” rhythmic schema, the sheer excitement I (still) get when he first turns a dulcet, low-pitched, smooth vocal note into an energetic, almost-screamed tone—dripping in vibrato and on the verge of completely falling apart changes the entire dynamic of a song. Similarly, when a piece that has been riding along with only vocals, bass, guitar, and drums for a while seemingly from nowhere introduces a loud, dynamic horn or brass section, it surprises us, defying our expectations and violating our schemas.

Dissonance is probably one of the most obvious and visceral schematic violations music can provide. Put simply, dissonance is the result of juxtaposing certain intervals that do not “fit” together. The introduction of incorrect or incompatible relationships to a previously harmonious whole checks both of the requisite boxes to be seen as novel; it eschews expected patterns, and it violates our schematic understandings. Dissonance is the vehicle of melodic and harmonious attention, our hearing of such challenging stimuli arouses our innate desire for novelty, and dopamine is released in response.

Everyone has their own thresholds for dissonance and novelty or complexity in general. To pit dissonance and tension as the brave saviors of music in the face of banal repetition and positive-feedback loops would be to only tell half the story. As much as we need the challenges of dissonance, so too do we need to safe and expected embrace of normalcy. In fact, over time and through repetition, once-novel musical ideas work their way into our schemas, into our stored patterns. The previous sources of surprise and violation slowly form parts of our stored patterns and our expected whole. As with almost every deep dive into the perceptual and attentional systems we’ve performed in this book so far, we require the *yin* of safe expectance just as much as we require the *yang* of dissonant novelty. Order and disorder, just like consonance and dissonance, just like expectation and surprise, do not exist away from, or in spite of, one another.

The Spectrum of Surprise

Remember the crashing mugs in the coffee shop in Chapter 1 that caught our attention through initiating a startle response? This response is surprise. Surprise is often categorized as an emotion, insofar as scientists can actually agree for 12 seconds on the actual definition of one, and often lasts for a very short period of time, usually giving way to other, longer-lasting emotions. Surprise and the startle response exist to “snap us out” of our current attentional focus, often toward the source of the startling stimuli. Broadly, surprise can be seen as the initial “switch” of attention that occurs when our expectations are defied. It’s that pesky lions-eating-our-face scenario all over again.

Like many emotions, surprise can occur with varying degrees of intensity and valence, meaning we can have big, bad surprises, such as a nasty wasp sting from out of nowhere, or little, happy surprises, like receiving a small, unexpected gift from a friend. In Chapter 2, I dropped the awe-inspiring knowledge that only primates can see the color red. The “surprising” thing about learning this, for me, was the implication that *bulls can’t see red*. After years of seeing the (awful and incomprehensible) matador-with-a-red-cape vs. an angry bull trope, I’d gone about my life believing incorrectly that the specific color somehow antagonized the bull. Now, this knowledge is, essentially, borderline useless to me. The surprise I felt was merely a violation of an accepted schema. It didn’t arouse my emotions and it didn’t make me feel particularly sad or happy, yet it still surprised me. In reality, I remember that single, silly fact plainly *because* it was surprising.

When something violates our expectations enough to instigate surprise, we remember the specifics of our situation much more readily. In a December 2008 article in *Scientific American*, Daniela Fenker and Hartmut Schütze suggest that the hippocampus, which plays a significant role in taking information from working and short-term memory into long-term memory, is the brain’s “novelty detector.” When the hippocampus is presented with new information, it compares it to our implied knowledge of the world (our acquired schema) and, if the two are incompatible, sparks a kind of dopamine feedback loop between the hippocampus and various brain regions. This is the accepted reasoning as to why novel (that is, surprising or expectation-defying) occurrences and learnings tend to stick with us much more easily and for a much longer period of time.

As well as the novel occurrence or stimuli *themselves* being more memorable, Fenker and Schütze give evidence that, in fact, stimuli and information that are present *alongside* the novel item are also more memorable. That is to say, in many cases, when we’re surprised by *a thing*, as well as that thing itself being more memorable, the stimuli

we experience around that time are as well. This notion might seem obvious if you think back to the last time you were truly surprised by something that occurred during your day-to-day activities.

As a personal example, I still vividly remember the emotion and stimuli I experienced when I discovered the results of the United Kingdom's shocking Brexit vote. A lot has changed in my life since then, yet I still remember the layout of the room I was in, down to where the light was coming in. I remember my emotional reaction and my immediate response to such a personally dismaying and surprising situation. I have a similar vividness of my memories of the day I found out that Donald Trump became the actual president of the United States of America. Perhaps this says a lot about my own naivety, in that I was sure enough that neither of these events would happen, to render their occurrence shocking, but they at least provide good working examples of the power of surprise, especially with regards to memory retention. For reference, these events occurred almost two years before the time of writing this chapter, and I struggle to remember what I had for dinner yesterday. Furthermore, these are two very negative, personal, and isolated examples of surprise and—by the very nature of their categorization as “very surprising things”—are not situations we often find ourselves processing. More common surprises (at risk of presenting an oxymoron) are the novel occurrences that elicit much tamer responses. It is in these moments that art forms like music, as well as good design and great copy, can really shine.

Surprise is a kind of gateway emotion. When we experience surprise, it's generally followed, quite quickly, by a more “opinionated” emotional cocktail. This could be the mixture of confusion and dismay at a shocking political result, or a blend of joy and excitement at a big lottery win. On a subtler level, less-intense forms of surprise can result in a wry smile or chuckle at a well-placed joke, or a feeling of exhilaration and excitement when a song transitions from sparse, ambient tones to a loud, energetic crescendo. Regardless of the emotions that follow, one thing is clear: we *remember* them better in the presence of novel events. This is the heart of what we know as *emotional design*; not only do we look to create positive moments throughout our interface, we must strive to make them *memorable* as well.

If mystery movies and crime TV shows rely on the intense surprise of, for example, *finally* revealing the killer, music relies on the subtler surprises: the “I wasn't expecting that, but I like it” transition from a basic 4/4 beat to a 10/8 breakdown—the unexpected, dynamic horn section that takes us into the chorus of an upbeat funk or soul track. Design, too, tends to operate at this level of surprise, favoring humorous copy and clever animations over pure, shock-value moments of awe and explosion.

Surprise: the Remedy to Habituation

An inevitability of the permeation of digital products and interfaces into every aspect of modern life is that of habituation. By this, I do not mean the forced introduction of a single product or application into the daily habits of an individual—an idea I find at its core quite deplorable. We spend our days performing largely homogenous actions, gestures, and interactions within largely homogenous interfaces that more or less meet our largely homogenous expectations. This is the double-edged sword of the habit-novelty dichotomy. On one level, the habituation of product usage and the resulting homogenous, expected interaction paradigms allow us standards with which to inform our work, much like the patterns that form our musical expectations. On a different level, however, this homogeneity leads to novelty-free, insipid interactions.

Purely functional interfaces—in this sense, interfaces that are constructed with meeting schematic expectation as the end goal—are the simplistic nursery rhymes of product design. Their reliance on previous habits and avoidance of any form of schematic violation have them teetering somewhere between boredom and irrelevance. Contrast this with interfaces that view fulfilling expectations as simply the price of admission and that acknowledge that, occasionally, *positively* defying expectations at the right moments to disrupt monotony and elicit surprise can be hugely beneficial acts. These interfaces, while a riskier proposition, stand to create a lasting, positive impact.

While “invisible” designs that “*don’t make me think*” are valuable goals, we should take care not to misrepresent these values as the championing of meek and insipid interaction design. Furthermore, by acknowledging that “invisible” and “habitual” are malleable *states* that our interfaces can be in—moments that can be deviated from and resolved back to over time or via interaction—we may start to notice areas where we can inject some positive, enjoyable surprises.

Happy Complexities

In Chapter 3, we looked at complexity in the form of a feature or interface where the system’s underlying model is difficult to communicate, or reconcile, against someone’s mental model. More generally, we can look at complexity as a form of schematic incompatibility. When something doesn’t match our preconceived expectations or understandings, it is, to us, *complex*. I feel that, as well as giving us an actual framework within which to judge complexity, this notion lets us look at difficult concepts away from the gaze of things like perceived intelligence or domain-specific expertise.

In taking this approach, we can take a step back from dangerous assumptions like “this person lacks the intelligence to solve this problem” and look toward the much less-judgmental assumption of “this person’s schema is incompatible with the behavior displayed by this object.” The former puts blame and impetus on the perceiver, leaving the creator unblemished. Conversely, the latter shows us that, when complexity is a function of schema, it presents itself as a problem that requires deeper thinking from the creator to navigate.

Furthermore, this approach to discussing complexity as an incompatibility between stimuli and schema lets us explore the idea of necessary, or “happy,” complexities. We’ve spent a great deal of this chapter analyzing how schematic violation can often result in excitement, intrigue, and surprise. If we abstract this out, we can suggest that a certain level of complexity is desirable, often necessary, for people to enjoy certain media. The notion of “desirable difficulties” (Bjork, 1994) presents this idea as a tool of learning, and we can look to our outgrowing of the simplistic movies, books, and music of our infancy to tackle the idea that complexity, to a certain extent, is salvation from the boring and mundane.

As children, we tend to lean toward preferring simple, repetitive music—with the abundance of nursery rhymes and “group” songs as well as popular TV shows and movies with simplistic musical underpinnings playing into this. Part of this is due to the fact that, at such a young age, our brains are still going through the process of creating and strengthening the neural networks that allow us to distribute our attention between multiple stimuli (Posner, 1990). At this stage of our neural development, it proves exceptionally difficult for us to process complex or concurrent stimuli, such as music that makes heavy use of dissonance and overlapping sounds. We appear to prefer simple, consonant music partly because the cognitive processing required is so low. We’re also still forming our schema for the proverbial syntax of music.

Music’s role as a language, or *protolanguage*, is a source of heavy debate across the sciences, so to call it a “language” in an academic sense is tantamount to kicking a hornet’s nest—but in our infancy (and actually some time before birth), we *are* processing music in many cases as we would language. We’re learning implicit structural rules and causalities, forming the basic syntax of musical communication, forming the schema that underpins the generative functions of music, and all the while *everything is new* to us. It stands to reason, then, that simplistic, repetitive music is our easiest, earliest foray into the realms of musical schema.

Yet as we grow, our ability to process multiple stimuli at the same time improves, our attentional filtration systems mature, and we're better equipped to separate our sensory inputs into discrete concepts in our brains. At this point, the oversimplified, prototypical music we've become used to becomes boring; it has served its purpose as a schematic primer, but we know it and its inherent structures all too well. We start noticing and appreciating the complexities of our world a little more, and with that comes a desire to be challenged—to not be exposed merely to the simplistic and predictable. Compare, say, the classic soundtracks of Disney movies to the very early prototypical nursery rhymes we teach infants. The majority of Disney movies are aimed toward a more “mature” audience of 4- to 12-year-old children. With this, comes the notion that more complexity is not only permissible, but also actively required to maintain engagement. Disney movies are part of many infants' first exposures to nonlinear story arcs with more complex emotional content and character development, so it's no surprise that the accompanying music is itself more evolved than the prototypical rhymes and lullabies that inform many a child's early musical schemas.

Yet, the soundtrack to *The Little Mermaid* is hardly Meshuggah. The harmonic, rhythmic, and tumbrel qualities of typical Disney pieces are still very much prototypical of Western music—even the *Mulan* soundtrack does very little beyond a few instrumentation cues to match soundtrack with thematic and cultural content. This is, in part, due to Disney's need to maintain its widespread appeal—to present overly challenging and overly complex ideas would inherently make their products more niche and be less universally appealing. Given their target audience and the fact that schema and neurological faculty are still evolving and maturing in the vast majority of it, Disney represents an abstraction of a milestone in cognitive and schematic development: an increased desire for complexity is offset by still-maturing faculties and schema.

As we reach adolescence and early adulthood, our mental faculties are generally approaching their peak abundance. The brain switches from mostly creating neural networks to a focus on strengthening and culling what we already have. This is where we start informing and developing a lot of our lifelong preferences and tastes. During the hormonal and emotional abundance of our teenage years, things like our emotional vulnerability, desire for social connection, and increasing focus on our sexuality and sexual desires all feed in to inform and reinforce the stimuli we reach out for and the art we consume. Cultures emerge around fandoms and shared interests. The desire to fit in, to be part of something, and to flock around likeminded people introduces external influences and attachments to our idea of what we “like.” We encounter new,

strange-to-us music and artwork that we never knew existed. During these times, our schema for things like music, art, movies, and video games are constantly challenged in consistently novel ways. This potent cocktail of emotional vulnerability and incessant exposure to new ideas makes for some of the most powerful memories and attachments we form in our entire lifetime, and it serves to greatly inform our worldview through schematic reconstruction and cultural exposure.

All of this leads to a hugely varied, highly personal tolerance to, and desire for, complexity. Those of us who constantly crave new experiences, who base their lives around being challenged and having our expectations defied, generally have a relatively high tolerance for complexity with a lower-than-average ability to tolerating the simplistic, limited, or boring. Those of us set in our ways, or who desire neither adventure nor challenge, tend to err on the side of the mundane, preferring media and input that meets expectations and provides predictable, digestible plot lines or conceptual structures. Figure 4-4 shows an example of two “affect/complexity curves.” The leftmost curve might represent an “adventurous” person, who thrives on schematic violation. The rightmost might illustrate a person who prefers the safety and niceness of schematic fulfillment.

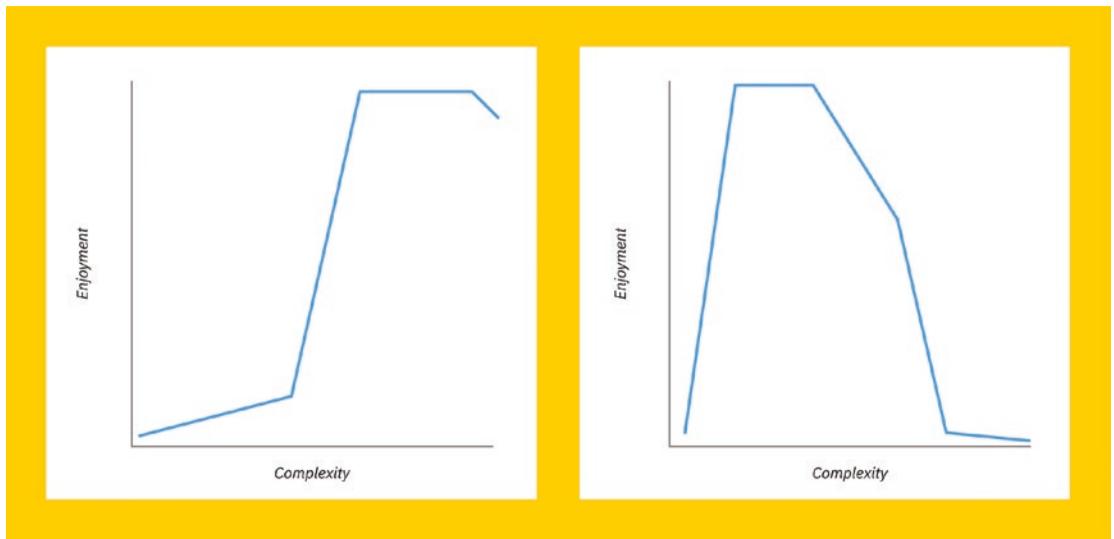


Figure 4-4. Representations of the desired complexity of two very different people

Yet, these curves do not tell the story of a person’s attachment to complexity, and there’s plenty of reasons to believe these curves change based on many factors. First, the media in question: I really dislike simplistic music, for example, and I much prefer music that provides a challenge to the listener; odd time signatures, non-standard structure and abrasive timbres appeal to me greatly. However, I *love* simplistic, “trashy” TV shows—I like predictable plots and one-dimensional characters and memorable one-liners. That’s not to say that I don’t appreciate sprawling, nonlinear television jaunts, or that I refuse to listen to anything in non-syncopated 4/4—just that I understand the points on the complexity curve where I’m *most satisfied* for particular media.

Second, mood, mental fatigue, cognitive load, and any concurrent activities will all impact our desired complexity curves. If I’m anxious or having a depressive episode but need to get something important done, I’ll listen to simplistic, calming ambient music or Disney soundtracks that I know by heart (yes, I’m that cool). If I’m working while watching YouTube, I might throw on a Let’s Play, where some esteemed YouTube Person simply plays through a video game. As we discussed in Chapter 1, circumstance and cognitive load affect our ability to actively pay attention to, or filter out, various stimuli. If the focus of our attention is something important, like our work, then it stands to reason that the desired complexity of background stimuli would be lower than if those stimuli were to be the focus of our active attention. Conversely, if I’m tidying my kitchen, my music choice might be something more complex to take my mind off the remedial task at hand—just as the default mode network may take over when we’re carrying out a boring or trivial task.

What this tells us is that *complexity is desirable*. Yes, the level of desirability is heavily dependent on the nature of the media, cultural convention, and the personal preferences of the person interacting with the piece, but very rarely will something that completely adheres to our schema be engaging or impactful.

The Musical Interface

Part of the emotional power of music is a sort of two-way contract between artist and listener—the notion that vulnerability is the price of admission to a novel and invigorating experience. Movies, books, video games, and comics, among many more media, are all art forms that benefit from this unwritten contract. But what about design? Given the state of digital technology and product design, the general anticipation or uncomfortable feelings we encounter are, quite rightly, more rooted in trepidation and

cynicism than they are in the adventurousness and suspended disbelief that give other progressive media their emotional canvas. Justifiably, rather than asking, “What kind of journey will you take me on?”—as one often would with a new piece of music or a new video game—when faced with digital products, our questions are more likely to be, “What data are you going to mine from me?” and “How often are you going to email me marketing trash?”

However, we can look to progressive media such as music and video games for inspiration in many areas. The first source of inspiration might be the use of time as an axis on which schematic inference and violation can occur. A huge part of what helps to make applications usable and enjoyable lies in not only the designer’s acceptance that their creations afford an experience that changes and adapts over time, but also that these changes and adaptations inform—and are informed by—a person’s schematic understanding on many levels.

We all have preconceived notions of what an “app,” in abstract, actually is. For those of us who do most of their work on the Web, perhaps an “app” is simply anything that has a digital interface. For people who only use a smartphone, their schema for “app” likely has a very different rule set; perhaps it’s something they find in an app store and install to their device. Second, there will exist a schema for the specific *type* of application. Someone’s preconceived idea of a calendar app might differ in expected functionality to another person’s. Finally, there will exist a schema for *your* application in particular, informed by a wide array of preexisting schema and by the expectations you help set. It’s important to acknowledge that, while malleable and still in its infancy, a schema will highly likely exist for your app before it’s even used, either through direct, in-depth feature descriptions on your web site or through abstract messages and broad-stroke feature descriptions on your off-site marketing.

What music shows us is that there lies opportunity in the fulfilling and violating of these preconceived schemas. Now, we’re likely going to want to do a whole lot more of the fulfilling and a whole lot less of the violating, given that we do not have the same emotional contract with our audience that movies, games, and music have, but there are still times where schematic violation can actually be rewarding. The flip side to the apparent default cynicism an average person holds toward digital products is that it carries with it certain expectations. Indeed, if the schema for a digital interface is “boring, data-mining, riddled-with-advertisements clusterf#@#,” by defying those expectations, which isn’t exactly hard with a bit of care, time, and mindfulness, we’re instantly able to differentiate our products from the boring, data-mining, riddled-with-advertisements clusterf#@#s.

Surprise, especially when manifested as the fail state of a negative prediction, is a huge factor not only in the memorability of an interaction, but also in the potential affect attached to that moment. By first *understanding* some of the default schemas that people might have of our interface (we'll explore how testing sessions can be better-utilized around this idea in Part 2 of this book) and how these might manifest themselves as expectations and predictions, we can determine areas of a schema that may result in negativity or complacency. Second, by pinpointing potential areas to *defy* these expectations, a good designer can take the lead of a good composer, injecting well-judged schematic violations to create the kind of positive surprise that is so well used throughout progressive media.

Defying expectation for the sake of it rarely works out without a huge deal of fortune. Setting the expectation that “this icon will save my work” only to have it actually pop up an animated dinosaur at the bottom of the screen for no apparent reason is, first, admirably weird, and second, likely to be an infuriating “surprise.” However, understanding where a schematic violation could be beneficial is something that is absolutely key to this part of a process.

And what about tension, consonance, and dissonance? While the general build-sustain-release approach that music takes with suspense and tension rarely translates well to interface design, where productivity and efficiency trump any kind of “emotional journey,” the notion of generating interest and dissonance through the deviation from an established “home” state, and intrinsic reward through the reversion to this state is something we can absolutely explore in our work.

While extremely poor uses of dissonance can be seen throughout modern applications in the form of unnecessary notifications and simply poor design, there *are* uses to this approach. When we talk about a harmonious baseline in design, we're essentially talking about the combination of a few concepts.

First, we're talking about an application that is in an expected state. Second, the interface's representation of that state is correct and accurate. Finally, that, through interaction with various areas or components of the interface, we can re-achieve this expected state should we need to.

The first point is somewhat trivial because it represents the ideal or atomic, un-errored state of our underlying system. If we were working on a bug-tracking app (Figure 4-5), for example, this state might be the “happy state” of having no active bugs. The second point relies on schematic expectations being fulfilled, not only by the system's underlying state, but also by *how* the design of the interface communicates this.

This would involve clearly communicating the expected state (0 bugs) in a harmonious way (that is, well designed, on-brand, free of distraction, and adhering to cultural convention) while still presenting the features and actions necessary for efficient use of the application (create a new bug, view all completed bugs, etc.).

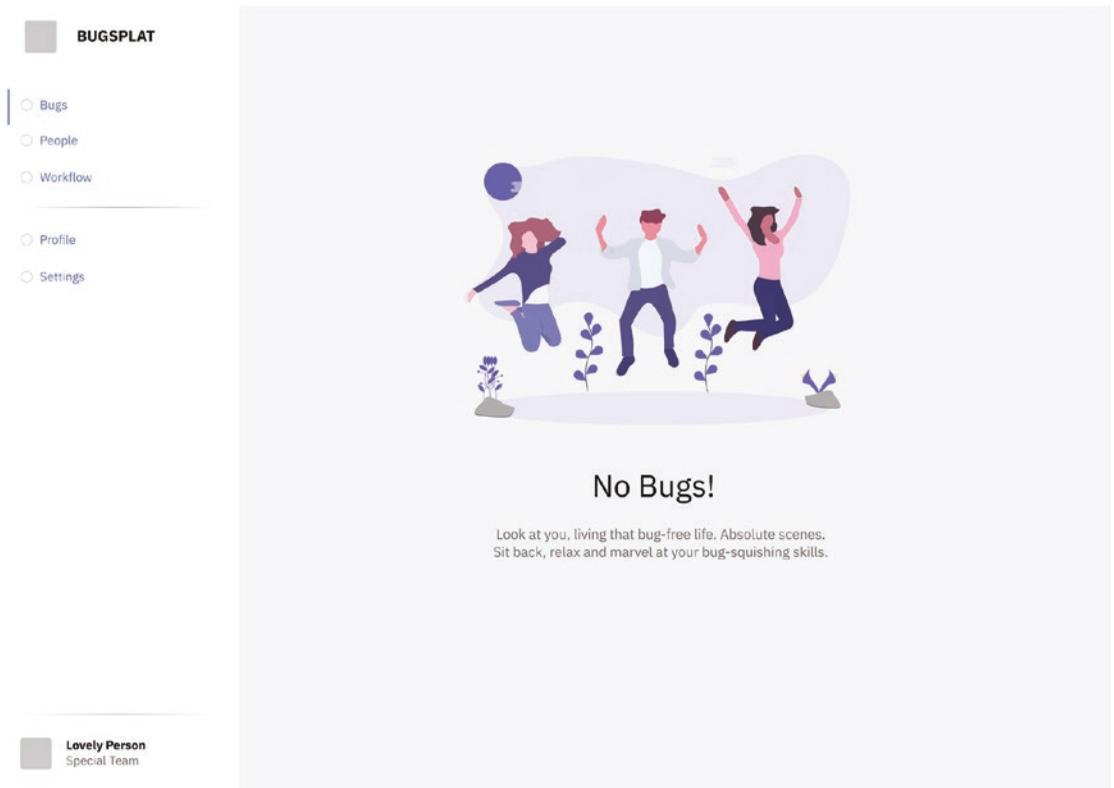


Figure 4-5. *A wireframe-like prototype of a hypothetical bug-tracker*

If we reinforce the expected, desirable nature of this state, we create the design equivalent of our metaphorical “home” chord: a baseline harmonious state, free from interruption and attention-grabbing, and deviation from which represents a degree of tension—a schematic violation that requires resolution. By effectively setting this baseline—in this case, say, raising a new bug—our interface can adapt and enter a state of necessary tension (Figure 4-6), where dissonance is used to portray a deviation from a harmonious state. This is where it makes sense to use certain implementations of attentional management—perhaps showing a notification badge in our “Bugs” sidebar or calling attention to the new bug by animating it into view.

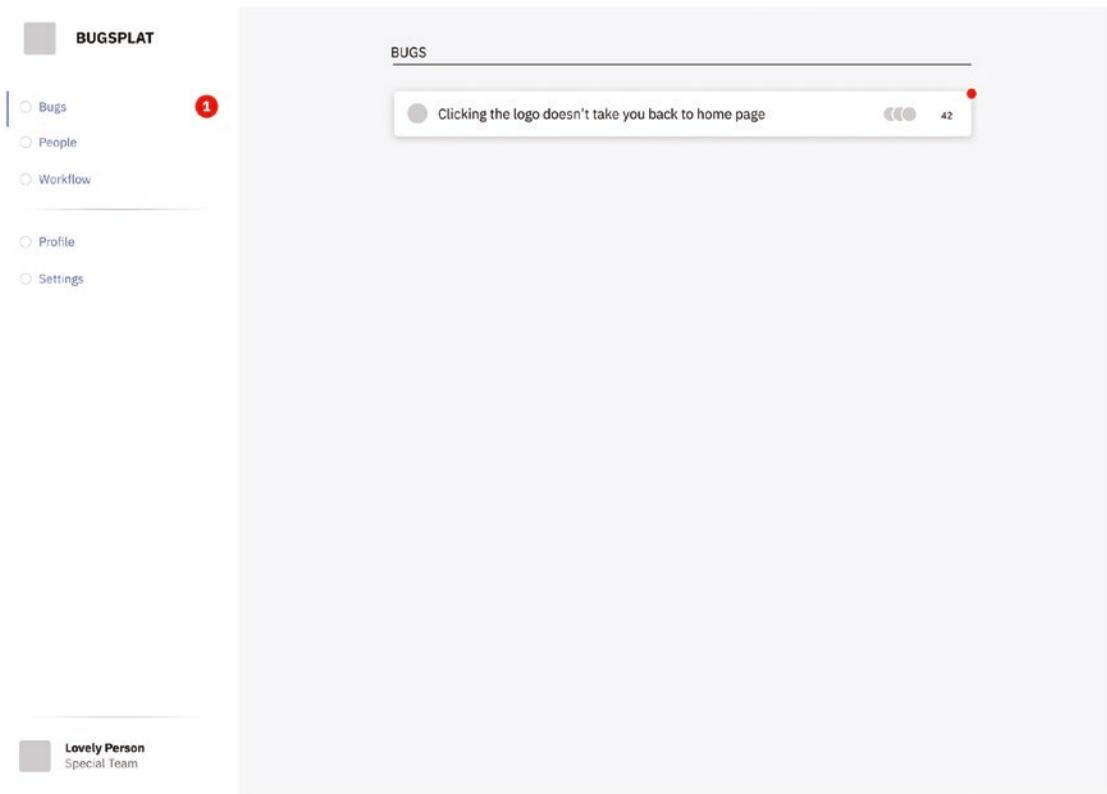


Figure 4-6. *Introducing a relevant amount of tension*

This now creates a state where our previous, harmonious baseline has been deviated from and work must be done to revert it to that state. By fixing the bug and marking it as such, we give the bug-fixer the rather understated reward of being the force that resolves this tension. Empty states in apps where empty is preferred (we want 0 active bugs in our bug tracker) are prime candidates for this approach, but so too can empty states themselves be the points of tension and dissonance. This is often the case in applications where the *creation* of content is the goal. In this case, we seemingly lose the luxury of being able to communicate our harmonious state up-front. We want our nonempty state to represent our harmonious baseline, with the tension coming from communicating our empty states.

A great example of bypassing this apparent limitation is the idea of “demo” content acting as a tutorial or introduction to the application. We discussed onboarding and in-app guidance in depth in Chapter 3, and this can be a great way to introduce people to the various states and “feels” of our interfaces. Trello (Figure 4-7) does this very well by presenting you with a demo project, full of example cards. You’re able to explore the various options and states of Trello in a “sandboxed” environment.

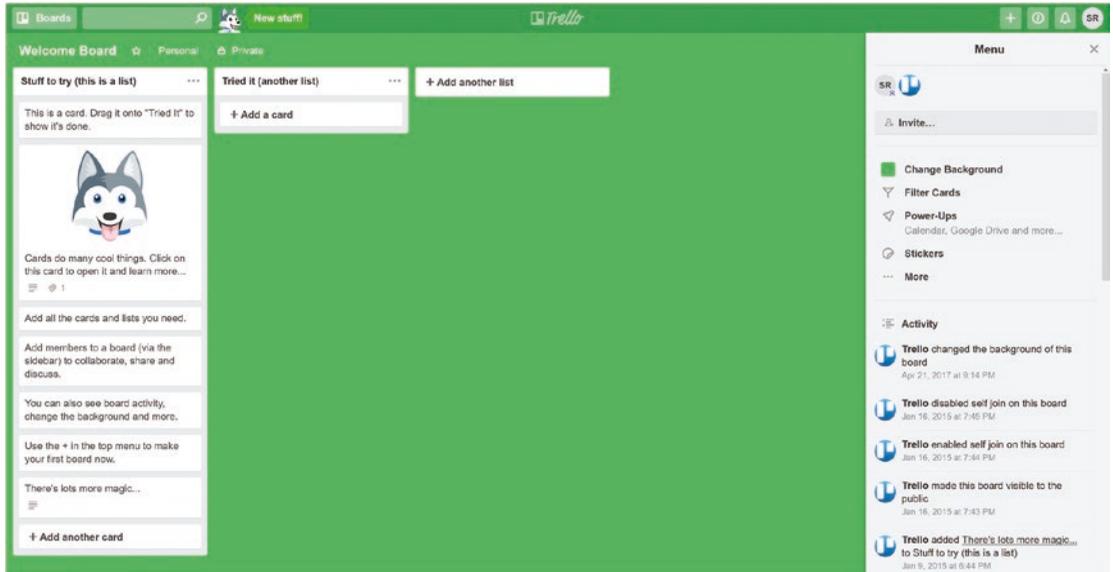


Figure 4-7. Trello’s example project

Not only does this get you into the bulk of the app’s features quickly, it bypasses the initial requirement of having to create content to explore the various states a project can be in. Given that Trello projects are often complex, it can take a good deal of effort to get into a state that is representative of the “normal” Trello state—that is, numerous cards in various states, with various labels, assigned to various people. Trello’s example project is an elegant way of saying, “This is the normal state of the app, this is your expected baseline.” And it works. Figure 4-8 shows the opposite of this with an empty Trello project.

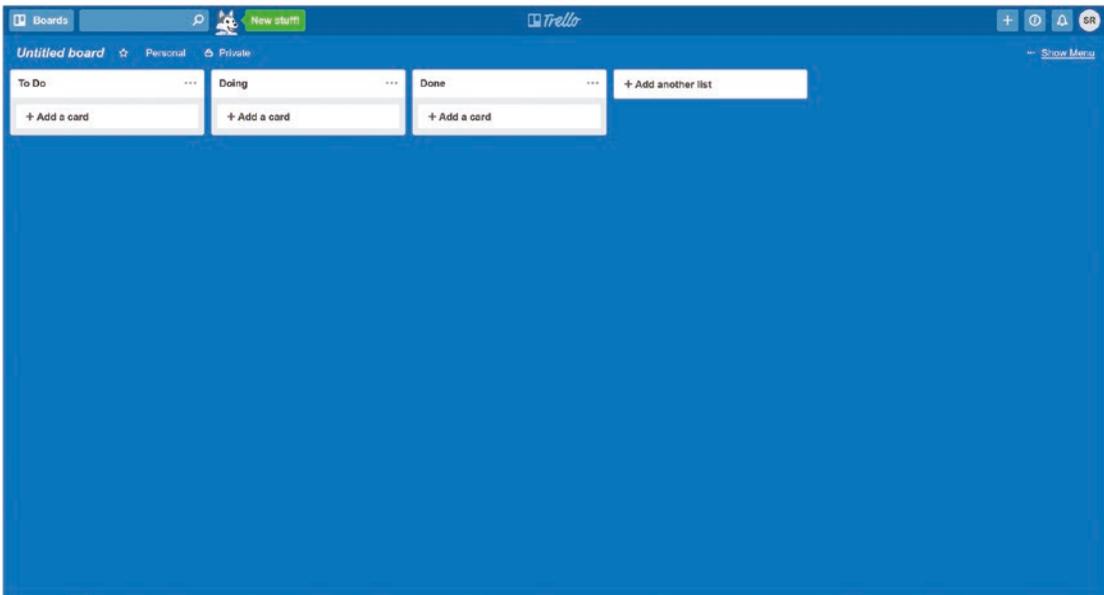


Figure 4-8. A new, empty Trello project

Empty or sparsely populated Trello projects look somewhat *strange*. There’s an awkward asymmetry to Trello’s UI when it’s not populated with content and, as we’ve already been exposed to the “filled” project state, we (hopefully) know that this doesn’t just come down to idiosyncratic design decisions. Put another way, our schema for a Trello project is not an empty, single-lane interface; it’s a populated and content-rich environment. The schematic violation presented by a sparse-feeling empty state would not exist if we were not initially presented with an idea of the “normal” state of a populated project. Without that initial exposure, the visual dissonance can easily be mistaken for just outright poor design rather than a creation of necessary tension. To resolve some of this tension, we’re invited to create new cards and, as we do so, we can slowly see that we’re bringing our *tense* state toward the predictable, safe notion of “home.”

It’s imperative, however, to acknowledge that tension without a clear resolution path is just *anxiety*. And that accidental tension and dissonance that remain unaddressed by a designer can very quickly tank the notion of a harmonious baseline. This entire exercise relies on the baseline state of our applications actually being desirable, fulfilling systemic expectations, and effectively being communicated as such. Use of dissonance as a tool for “nudging” behavior is tantamount to fallacy. If the underlying, intrinsic goals of an application’s “happy” state are neither met nor communicated, then any

tension we attempt to create is likely to manifest itself at best as annoyance and at worst as anxiety. This means that incessant notifications purely to “differentiate” your app icon from others or to create a degree of unresolved tension until the notification is read only “work” when notifications are a *violation* of your app’s schema.

For most applications, notifications are thrown around in abundance, and this soon forms part of people’s schemas, translating to an expected behavior. Suddenly, the tool used to attempt to convey a departure from an expected state (the big red blob of notification hell) becomes the de-facto, assumed base state of the interface and its underlying system. The same applies to attention-grabbing advertisements, creating an unavoidable and indelible dissonance between the actual, realized interface and the expectations one might have of it. This is part of why incessant and unnecessary notification badges and demanding, obnoxious advertisements aren’t just annoying, but actively damaging. The constant tension they create mean a harmonious baseline is rarely, if ever, achieved. Unresolved tension lingers and leaves us with a sense of apprehension and anxiety, emotions that we should see as absolute failure states should our application elicit them.

To summarize, music taps in to the rewarding nature of expected, safe, and predictable feelings of schematic adherence and pattern recognition, as well as the intrigue and surprise made possible by providing novel and enjoyable violations of an underlying schema. We can achieve similar, albeit diluted, results in our designs. By acknowledging the shifting, progressive nature of our interfaces, by accepting that we can morph away from, and back toward, an accepted, harmonious state—providing pleasant surprises along the way—we can disrupt the predictability and joyless monotony of habituated product use. Just like with music, relationships with our creations are built up and reinforced over *time*, and we have the luxury of time as an organizational and communicative tool that was rarely afforded to the static visual media that so informs our work today. By first understanding the nature of human expectation, and subsequently finding novel ways to play to and veer from these expectations over time, we can view our work as a composition that plays out over time and in response to human interaction.

Of course, there’s an important set of responsibilities that arise when we approach emotion and expectation in this way—namely, that too much dissonance, or too big a gap between schema and stimuli, can lead to uncomfortable and anxiety-inducing scenarios. Furthermore, we must wield tension and dissonance as we would any other form of attention-grabbing phenomena: mindfully and empathetically.

The overabundance of design “techniques” that create an undercurrent of *negative* tension with no clear path to resolution are not only plainly abhorrent, but they also serve only to further reinforce the idea that design is unworthy of emotional investment and engagement. While we do not have the luxuries afforded to the more entertaining art forms, we can still embrace the lessons they teach us about wonder, surprise, and exhilaration. Maybe one day, we’ll have set a new schema for digital products, one that brings us closer to providing real moments of excitement and enjoyment.

Summary

In this chapter, we used music to explore the mind’s proclivity toward forming schemas and concepts against which we categorize events, environments, and experiences. We discussed how music relies on cultural convention and emotional association to constantly blur schematic boundaries, making the unexpected enjoyable.

We also discussed the importance of seeing interface design as something that takes place moment-to-moment, existing as a sequential medium alongside music, video games, movies, and television, rather than as a series of static screens. We can view our interfaces as places where tension can be built and released, where applicable, to create interest and enjoyable complexities.

Our schemas and concepts lead us to perceive the world in certain ways, allowing us to make predictions and form expectations against our understanding of the world. It’s important to remember that these concepts are mutable and can change over time. Whenever we’re presented with a novel concept, a member of a category that we haven’t experienced before, it can go on to inform our schemas from then on. It’s important to attempt to understand the cultural conventions and worldviews that might form the basis of people’s schemas, similarly to how we wish to bridge the gaps between our system’s models and the mental models people form when using our interfaces.

We explored the idea of “happy complexities” and how people enjoy a certain amount of challenge and “newness.” Different people have different levels of desired complexity, and this can often be informed by cultural conventions.

Finally, we discussed how surprise, when used as the fail state of a negative prediction (defying the notion that “this application will be clunky and full of ads,” for example) can, occasionally, be a useful tool in our work, as well as how we can build and release tension through creating consonant and dissonant states in which our system can exist and transition.

The main thing I'd like you to take away from this chapter is the idea that our interfaces exist across moments and transitions and they can effectively build and release tension over time. Understanding our innate desire to form schemas and conceptual categories provides a valuable insight into how cultural convention and exposure can shape our perception of the world, but in creatively defying some of these schematic expectations—especially against a harmonious baseline—we can use time, progression, and transition to create a composition of sorts.

CHAPTER 5

Reward and Motivation

Insomuch as one *can* break the implicit fourth wall in a book like this, I'd like to do so and state that this chapter has been the most difficult to write. Not in the sense that the subject matter is the most difficult to grasp (although it's hardly a field of consensus and empirical evidence), nor that the research was particularly overwhelming, nor in the sense that it's difficult to draw parallels or explain the concepts in as plain a form of English as they allow. In truth, I knew this would be the most difficult before I even started this book, simply due to the *expectations* for such a subject matter.

This chapter will discuss dopamine, it will discuss “reward pathways,” and it will discuss the role emotions play both in rewarding behavior and as a source of motivation themselves. These three concepts, more than any others, have been popularized and banded around the design world—and the broader worlds of tech and business development—for a *long* time and with a *lot* of inaccuracy and misinformation.

Dress the literature up how you like, add in as many caveats and as many disclaimer paragraphs as you feel necessary, yet there is no escaping tech's fetishization of addictive products. While many books in the fuzzy category of design psychology attempt to toe the line, remain neutral in the presentation of evidence and theory, and eschew the moralistic debate in favor of some strange form of tech-libertarianism—this book does not, giving further reason for why this chapter was difficult to write. To present the accepted brain science around motivation and reward (and dopamine's role, in particular) would be a difficult, but perfectly doable, undertaking. To do so while staying true to the ethical considerations that form the underlying reasons for my writing this book is another matter entirely. So, here we are. A chapter that would, in many other books, be the first port of call is presented deep into our discussion for the very reasons that make it difficult to write.

First, I want to make clear the fact there exists far more utility in our understanding of the mind and that our understanding of reward is but one *part* of an important *whole*. Second, in the hope that the previous chapters have done enough to portray the ethical underpinnings of this book so far, we hope to understand the mind and its functions primarily to *serve* them and to never, ever do harm. I want to be explicit with this, because, at the time of writing this book, there exists a very real, very widespread misuse of the teachings of psychology, namely in the pursuit of profit and persuasive manipulation. One of our most overlooked roles as technologists and designers is that we can help democratize technology, that we can put tools of creativity and productivity into the hands of people who are often priced out, passed over, or otherwise oppressed. The current climate of tech capitalism teaches us that good design is design that drives profit or builds revenue; that is our schema for our work's impact.

A schema that prioritizes the bottom line over everything else is one that I, and by proxy this book, openly reject. As I discussed in Chapter 1, design as a proxy for a form of almost preemptive cognition is but one of the infinite applications of so-called design thinking that doesn't revolve directly around increased profit or more sales; it doesn't feed into attention economies. It's also something that doesn't resolve directly with most business metrics and, therefore, it is something that can be quite difficult to justify to an overeager or myopic CEO or manager. We'll discuss "tactics" (and how sad it is that we need to) for communicating this value in Part 2. But for now, if this preamble has you worried that I'm proposing a dramatic shift in how we approach design's role in startups and technology companies, then worry no more—I am.

Finally, and I promise we'll actually start talking about rewards and happy times soon, I'll politely ask that you discard as many of our preexisting ideas or assumptions around the whole *dopamine thing* going into this chapter and, while you're at the idea bin, just throw in any kind of lingering Maslow's hierarchy of needs triangle weirdness you have knocking around there too. Given how often these overhyped (in the case of dopamine) or inaccurate (in the case of need-based pyramids) concepts come up, get mistranslated, and finally get meme'd around social media and infotainment web sites, it's quite likely we've all got some long-held misconceptions about needs, motivations, and rewards. Don't worry, we'll top all that up next.

Common Mistakes of the Reward Slingers

Confession time. At various points in my design career, I've used the following phrases: "Dopamine, the reward chemical..." "It just needs a bit of dopamine..." and, perhaps most cringe-worthy, "gamify everything." If uttering them is not bad enough, I used them in somewhat popular blog posts and even at actual conferences in front of actual audiences. I, like many other designers beginning the casual schlep into design psychology, latched on to the boldest and most-exciting concepts from the literature at the time and decided that what the world really needed was more digital badges, progress bars, and gold stars. I was riding the wave of behavioral design and gamified interfaces and was an unapologetic advocate of surface-level gamification. I had inklings that these practices weren't as simple as designing the equivalent of a digital loyalty card. However, I was actively being employed to work on reward schemes and gifting flows so I was at least somewhat contextually bound to this research. Nonetheless, it would take me a long time and a lot more research to realize how shallow these implementations were.

When implementing supposedly rewarding features, I've noticed products tend to make one, some, or all of the following mistakes.

Trying to Manufacture Reward

While "synthetic" rewards do exist and can be used quite effectively, the shallow notion that in-product recognition (a digital sticker or a badge, for example) is enough to feel rewarding to a person should be challenged. We'll discuss a little later why this is the case, but it boils down, in part, to the disparity and interplay between extrinsic and intrinsic motivators. Quite simply, the initial effectiveness of a particular reward is, at least to some degree, a function of its importance to the person receiving it. As we'll see a little later, superficial or unimportant rewards can actively *discourage* certain activities.

Overreliance on Carrot and Stick

Operant conditioning, a component of the oft-maligned School of Behaviorism, relies on the notion that the valence of the result of interaction with our environment will determine our future behaviors. That is to say, the "badness" of touching an open flame suggests we'll avoid open flames in the future, and the "goodness" of eating a ripe piece of fruit means we will strive to do so more often. Behaviorism—a school of psychology

stemming from famous dog abuser Pavlov and later popularized with B.F. Skinner’s controversial work electrocuting rats—has been subjected to widespread scrutiny, so much so that Noam Chomsky’s criticism of it is seen as a seminal work of cognitive science writing (Chomsky, 1967). While the idea that we’re “conditioned” by the stimuli surrounding our actions is hardly controversial, the notion that rewards should be used to drive specific compulsive behaviors is dangerously myopic. As we’ll soon see, both Skinner and Pavlov have had profound impacts on long-standing theories of learning and motivation, with classical and, later, operant conditioning being common reference points when discussing reward-motivated behavior and impulse. However, we must avoid the trap of seeing this side of design purely as a blinkered manifestation of behavioral psychology.

Forgetting about Emotion

The more we abstract our idea of a reward away from an intrinsic value to a materialistic “thing,” the more we distance ourselves from understanding the basis of *why* things are rewarding in the first place. Similar to attempting to manufacture reward, it’s important to remember that amusement, happiness, and even solving a difficult challenge can all be seen as rewards unto themselves. In fact, you may be lucky enough to be working on a product that can be seen as rewarding to use, such as Spotify with its music discovery service, or any piece of software that is designed for creativity (photo-editing and manipulation apps, design tools, and music recording and editing applications, to name but a few). If the aforementioned carrot-and-stick approach is the behaviorists’ implementation of motivation and reward in design, then these inherent experiences of creativity, progression, and mastery are what we might attribute to the humanistic influence—technology as a tool for creativity, discovery, and self-actualization.

Mandating Emotional Impact

Many interfaces simply just do not need to be, or feel, rewarding in any way. The abundance of literature surrounding persuasive behavior and the clamor for “behavioral design” to become *a thing* have led us to believe that the end goal of our work should feel good, that we should look to *delight* and reward at every possible opportunity. In many places, this is simply not the case. As we’ll discuss shortly, motivation and reward require significant emotional and intrinsic investments and, should we wish to

implement them ethically (we should), a huge overhead on our part to correctly do so. The assumption that you have to work reward mechanisms into your interface lest it becomes unenjoyable or dull is a dangerous one. Emotions such as surprise and delight (two very different feelings, often grouped together arbitrarily to describe moments that constitute neither) require a degree of attention to manifest themselves. They also involve manufacturing prediction errors, like the tiny schema violations that make music interesting, which again bring with them their own levels of cognitive load as our underlying schemas and predictions are challenged. As I've hinted at throughout this book, often our work is best directed toward creating effective tools of cognitive easing. Good design and "persuasive" design, while not mutually exclusive, are not the same thing.

We'll discuss many of the reasons that define the aforementioned actions as mistakes throughout the rest of this chapter. To begin, let's take a look at the rather murky waters of motivation.

The Origin of Wanting

The science of motivation is genuinely fascinating, but within it lies very little consensus, many a controversial theory, and a whole Pandora's box of debate. A rather simplified definition of motivation would be to view it as the mind's mechanism for *wanting*, to create a desire to perform (or indeed avoid) a set of activities based on our attachment to their previously experienced outcome. That is the combination of various processes that encourage us to seek positive outcomes and avoid negative ones. From an evolutionary perspective, motivation can manifest as a drive to perform our biological needs. If we weren't motivated to eat, would we? The answer to this, apparently, is no. Rats and mice that have been rendered unable to produce dopamine (the primary driver of reward-seeking behaviors) simply do not seek food. Deprived of their source of motivation, they show no need or desire, and they simply lie down and starve.

This rather simplified approach forms the bulk of the teaching, at least in the design world, of motivation and reward activities in the brain. We'll discuss the intricacies of this very shortly, but the main headlines are as follows: we witness or are taught that a certain action has a positive outcome; we have a trigger of some kind that causes a neurological response (the most commonly discussed taking place in the dopaminergic mesolimbic pathway) and instigates *seeking* behavior; our seeking behavior leads us to perform that task; and, finally, that task has an outcome that *potentially* bestows

upon us the reward. Trigger, seeking and working, reward. How simple! In this case, the motivation is our survival instinct, our biological hunger. The seeking and work lead us to acquire, potentially prepare, and consume food. The outcome is, at least in part, the satiation and restoration of cognitive function that removing hunger brings.

Beyond that, though, things start to get interesting. It's quite clear that we have needs and desires that contrast with, and often contradict, our basic evolutionary needs. Our motivation for eating, for example, could go far beyond the satiety reward that comes from the elimination of hunger. Sugar addiction is a motivation to eat sweet foods. We may also be motivated to eat when we don't wish to, say out of some form of social politeness, like eating a slice of a co-worker's birthday cake even though you're full from lunch. These common actions, and dozens more like them, are what give psychologists little reason to buy into the popular notion that our needs are hierarchical. We can forego many of our basic needs in the name of—apparently—more superficial ones. In the case of eating when we're not hungry, we're defying a commonly discussed innate need for *homeostasis*—that is, a well-regulated, stable biological state. When we binge, the theory goes, we overfill, creating a biological imbalance that makes us lethargic and gives us potential temporary hormonal imbalances. This state is not conducive to escaping the face-eating lions that have plagued this book as our theoretical protohistorical nemeses.

What this points to, and what makes the whole discussion around glorifying *addictive* technologies so insidious, is that motivation and seeking aren't always good for us. When we think of motivation, we tend to misappropriate it as a feeling of productivity—the desire to get things done. Highly motivated people are those who can easily drag themselves up to go for a run, perhaps, or get to the gym at 5 a.m. And while these are indeed *manifestations* of motivation-driven behavior, we cannot forget the darker side to this: many of the same neural processes involved in these good habits are also involved in many debilitating compulsions—compulsions that nefarious, money-hungry companies are more than happy to take advantage of.

Dopamine

Dopamine is, rather curiously, something of a household name in many areas of the media. It's popped up everywhere, and with that kind of coverage comes a huge array of misconceptions. If, like me, you've tried (with the best intentions, I must add) to educate yourself on the subject, you've likely internalized a *lot* of misinformation and assumptions.

If you're one of the lucky few who have managed to avoid the hype, then the next section will aim to introduce you to the accepted current knowledge and a couple of subtly differing theories of dopamine's role in our motivational behaviors. For the rest of us, hopefully, a little unlearning will help us escape the pop culture traps we've previously found ourselves ensnared in.

What the Heck Is It?

Dopamine is classed as an *organic chemical*, and it plays different roles in the body and the brain. When we talk about dopamine in design, we're almost always talking about its role in the brain as a *neurotransmitter*.

A neurotransmitter is, in grossly simplified terms, a kind of “messenger” in the brain. Remember in Chapter 3 when we discussed that the brain is made up of hundreds of billions of neurons that are connected like a network? Well, there's a *gap* between one neuron's connector (the axon) and any of its neighboring neurons, known as the *synaptic cleft*. Neurotransmitters live in this gap and, when stimulated, allow signals from one neuron to “jump” this gap to reach another. Other neurotransmitters include glutamate, serotonin, epinephrine, and histamine.

What the Heck Does It Do?

That depends on where in the brain a neurotransmitter is released. In the case of dopamine, it's responsible for all sorts of things. Dopamine affects lactation, for one. It can affect motor activity as part of the basal ganglia motor group, wherein a deficiency in dopamine is linked with Parkinson's disease. On the other hand, an abundance of dopamine can result in agitated and impulsive movement. It also has a say in how we learn from our environment and is especially common to find in the discussions around the aforementioned operant conditioning. However, the “famous” area and function of dopamine are its relationship to reward-motivated behavior in the mesolimbic pathway.

The mesolimbic pathway is known as a *dopaminergic* pathway, which, aside from being a pretty wonderful word, tells us that this pathway, among other things, synthesizes dopamine. Other dopaminergic pathways include the nigrostriatal pathway (where our aforementioned motor control is regulated) and the tuberoinfundibular pathway (where dopamine regulates prolactin, thus affecting lactation). Catchy names, right? To combat the esoterica of neuroanatomical nomenclature, the phrase

“reward system” is often used as an oversimplification of the discrete areas in the brain responsible for reward- and motivation-associated behavior. This is rather reductionist, but for the purposes of brevity and getting to the meat of our discussion, we’ll use it here as a replacement for the more “formal” structures. We’re far more interested in the nature of motivation cognition than we are the specific brain biology that underpins it!

Now that we’ve defined our good friend dopamine and limited the scope of our discussion to its role in the reward system, we can start looking at the generally accepted role it plays in motivation cognition.

Reward Seeking

The most common description of reward-seeking behavior comes from the radical behaviorist B.F. Skinner’s findings on what he dubbed *operant conditioning*. Operant conditioning, an expansion on the Pavlovian ideas that came to be known as classical conditioning, essentially gives us insight into the behavioral underpinnings of reward-seeking behavior and comes from Skinner’s work with his now infamous Skinner boxes.

A Skinner box, or an *operant conditioning chamber*, is a structure that allows scientists to test the association between reward, stimuli, and action in animals. The classic Skinner box example comes from Skinner’s experiments on rats where a rat was placed in a box that contained a lever. When this lever was touched, food was released. The rats were left to discover this relationship by themselves and, once the pattern of action and reward was associated, they went straight for the lever to receive their reward on subsequent trips to the chambers. Further examples of Skinner boxes introduced electric floors, lights, and screens to test the role of punishment and stimuli association with reward, but the most significant change to the experiment conditions came in the form of reward variance.

By introducing an element of variance as to when, or whether, a reward would be distributed, the behaviorist theories of reward-motivation were able to account for chance and variance in the administration of rewards. This resulted in four commonly referenced forms of reward scheduling:

- **Fixed interval** scheduling involves administering rewards to a fixed, time-based schedule—for example, giving a reward every two minutes, without variance.
- **Fixed ratio** scheduling involves administering a reward after a set number of actions, say, on every fifth press of a lever.

- **Variable interval** scheduling means rewards are spaced apart by time (not the number of actions), but with no set or predictable time between each reward.
- **Variable ratio** scheduling is the process of spacing rewards apart based on a varying number of actions. For example, pressing a lever three times may net the first reward, the second might require nine presses, the third just two. Variable ratio scheduling's apparent ability to tap into our compulsive behavior is something that makes casinos, as well as social media apps, thrive in our society. And this is where we get to dopamine's role in the whole sordid affair.

In simple terms, dopamine is released during the *anticipation* stage of the reward cycle (not at the enjoyment stage!). With the dopamine-deprived rats we met earlier in this chapter, their brains' inability to synthesize the chemical meant that they did not perform this so-called reward-seeking behavior. An interesting result of the study of these same dopamine-free animals shows that, when they're actually presented with rewarding stimuli (sugar water inserted directly into a mouse's mouth, for example), they still react with a hedonic "liking" response. This discovery is one of many reasons why dopamine's misreported role as a "pleasure chemical" is challenged. While it is responsible for motivational behaviors, it appears to have no say in the hedonic nature of an experience.

Further studies into dopamine's role in reward-motivated behavior have shown that when *chance* is introduced (i.e., through adopting a variable ratio/variable time reward schedule), dopamine synthesis in the mesolimbic pathway is increased. This, arguably, gives us the chemical blueprint for maximizing compulsive behavior: establish an action/reward association and then vary the schedule to which the reward is administered to introduce chance.

The classic example of this implementation can be found in the design of the American darling, the slot machine. The accepted reasoning behind the slot machine's addictive nature is that, essentially, it is a human-operable Skinner box. We have the association with reward (I can win money!), we have the ease of action (I put my own money in and pull a lever—just like a Skinner rat!), and we have the super-wonderful variable ratio scheduling of rewards (I might not win on this spin, but who knows what spin will pay out?!). The slot machine is, to be frank, one of the most depressing inventions I think the world has ever produced outside of vehicles of war, weapons, and torture devices. Everything about it is designed to compel human beings to give

away as much money as absolutely possible. A slot machine is designed to entice and addict. Worse still, it offers the perfect analogy for how, at this time of writing, technology companies capitalize on supposedly reward-focused behaviors.

The Pocket Slot Machine

Tristan Harris, a former design ethicist at Google, equates the omnipresence of social media apps on people's phones to "putting a slot machine in a billion pockets" (Harris, 2016). As we explored above, slot machines work based on intermittent ratio rewards: insert cash, pull handle, watch reels spin, hope for a reward. Harris (and myself, for clarity) believes that acts such as checking our phone, refreshing our e-mail inbox, and refreshing social media applications in search of notifications is the technological equivalent of this behavior. Take out phone, pull on screen, watch a loader spin, hope for content. The parallels are palpable.

The majority of tech's relationship with reward-motivating behavior is wrapped up in this very shallow, arguably problematic relationship with behaviorism and compulsion. Relatively new practices such as persuasive design (there exists, believe it or not, a persuasive design laboratory in Stanford University) and behavioral design have reignited discussions around classical and operant conditioning. Crucially, these discussions are notably light on ethics and notably high on unfounded optimism. This act of psych necromancy presents old ideas in fresh contexts. In the rapidly prototyped world of Silicon Valley business practice, it represents a dangerous paradigm where compulsion and addiction are dressed up as "nudging" and "habit-forming." The controversial practices of conditioning and manipulation are touted as wins for an industry already demonstrably devoid of morals, unions, and anything close to resembling the Hippocratic Oath.

The gambling industry is regulated so heavily because it quite measurably costs people money and systematically ruins lives. The irresponsible application of such a thin slice of psychology is proven to prey on the vulnerable and the cognitively burdened. This regulation exists primarily due to the ease with which one can visualize the associated losses. Money put into a slot machine comes from the pockets of gamblers. The very notion that casinos are profitable shows this. Simple economics demonstrates that by capitalizing on our hardwired nature to seek reward, a business profits while a human loses. What is more worrying about our *digital* slot machines is twofold. *First*, the loss is currently impossible to measure or visualize—it manifests itself as directed

attention to a specific application, as stolen moments, as distractions from important tasks. It's a lot easier to watch a bucket full of quarters disappear before your eyes than it is to quantify the glucose expenditure of attentional misdirection, or the time and mental energy required to re-enter a state of deep working, deliberate practice or flow.

Second, the reward these applications posit is far more powerful than money. It's been shown that humans do not typically find money intrinsically rewarding or desirable. Instead, we understand the enjoyable nature of the things money allows us. What is inherently rewarding is an eclectic cocktail of satiety and desire for knowledge, interaction, and mastery, including social validation, information consumption, and learning progression. Rather unintuitively, the rewards offered to us by social media, e-mail, and infotainment web sites represent bite-sized chunks of many of our most compelling desires, whereas the intoxicating allure of slot machines provides money—at best an abstracted facsimile of a reward.

The intensity of our seeking behavior can be broadly viewed as a function of ease of action, perceived potential loss, and motivational salience. Essentially, if we place enough perceived importance on a pleasurable result, our incentive salience—how intensively we focus on achieving a result—can be high enough to encourage us to seek a reward, even in the presence of potential loss. Using Facebook as an example, someone “liking” a post of ours, or commenting directly on it, might be of relatively low salience in comparison to many things in life. However, the work required to reach this is extremely simple. Essentially, we make a post and we wait. Once our post is made, our rewards start to roll in. We might get one or two likes straight away from our ardent friends (Facebook allows you to receive notifications when a “close” friend posts.) Maybe over the course of the next few minutes, we get three or four more likes, one or two comments, and a “share” from people who are browsing their feed at the time you posted. This initial buzz of social recognition and involvement is often enjoyable, but rarely enjoyable enough to keep us satisfied beyond the first second or two of receiving that notification. Essentially, the reward is just not rewarding enough—we need more.

If I asked you now to predict how many likes, comments, or shares (or likes and retweets on Twitter) a basic, banal status update would get you (say if you posted on Facebook or Twitter about how much you're thoroughly enjoying this book), what would your answer be? I believe most active social media users have a rough idea of the engagement their posts generally receive. Every now and then, though, you might post something that breaks that mold and *goes viral*—the hallowed ground of the social media meme life. This *chance* of going viral could be motivation enough to keep posting

and posting. What we see here is almost an onion-skin-like approach to social rewards. There's *usually* a baseline amount of interaction that fulfills a very small, constant level of reward—the typical like/comment/share amounts you're used to. Then there's the chance factor of your post reaching a much wider audience than you usually do, sending a flood of interest and engagement your way. Now you have the same varying schedule of likes and comments and shares but at a *much faster* rate. The sad thing about all of this, however, is that social media relies on these rewards never being salient enough for the process of *liking* to distract you from the process of *seeking*. The entire ecosystem of products such as Facebook and Twitter have traditionally revolved around the reinforcement of seeking behavior through expertly tweaked micro doses of enjoyment.

In many areas of our lives, our seeking behavior leads to a satiating result. We're able to enjoy our result and feel content. This can absolutely apply to our online or digital lives too. Messaging apps let us have long, in-depth, and rewarding conversations with friends across the world; sites like Wikipedia let us consume and *create* democratized, distributed knowledge; and countless apps for musicians, artists, and writers exist to foster and inspire our creativity. However, contentment and satiety rarely allow for fast, conditioned feedback loops. Most of those tasks I mentioned involve a pretty drawn-out time investment, and their rewards are often savored over a matter of minutes, hours, or often even longer. Our post, like, refresh, and reciprocate conditioning to social media rarely, if ever, allow for this. Facebook needs shallow rewards, just enough to make you feel like the initial posting or the resulting semi-consistent checking of your phone is worth it. We're not *supposed* to enjoy these abstractions of social interaction much because that enjoyment and reflection time is better spent—apparently—browsing your timeline so you can be advertised to. This is the tight, tailored *dopamine loop* of social media—the carefully designed balance of reward salience; conditioned, low-impact response; and variable scheduling.

There is a compelling argument for the application of this knowledge in the formation of good habits—something that the proponents and practitioners of this area of design are quick to point out. However, this simply has not been the popular manifestation of this branch of behaviorism in tech. It is the manifestation of compulsion and addiction. Calling it habit-forming design is currently just a cute moniker for a glorified compulsion fetish. Similar to priming and subliminal messaging in advertising,

this attack on our biases and mental processes can lead to damaging behavior in the name of nothing but maximized profits. As long as Silicon Valley and the carousel of products it produces remains unregulated, the responsibility for the mental health and the eradication of nefarious manipulation in the name of growth, profit, and funding fall on the leaders and creators within the industry. At the time of writing, the neglectful pursuit of profit continues apace.

Add this all together and the current landscape is inarguably bleak for those of us who do not wish to be venture capitalists, tech CEOs, or willing proliferators of addictive experiences. We have popular branches of design that apply the same teachings as casinos to their design thinking, operating in an attention economy where eyes on a screen and misdirected attention translate abstractly into profit. They create and iterate on applications that provide *just enough* chunks of intoxicating content in an attempt to keep us in a reward-seeking loop—all with the backdrop of an industry that fetishizes unfettered growth above all. This is the culmination of what appears to be the tech industry's best efforts in bringing brain science into design—a manipulative application of century-old findings from a school of psychology that has been teetering on extinction for decades. And there's a lab for it. In Stanford.

Reward in Games

It's not just modern tech companies that make extensive use of this knowledge, though. Video games have been long-time proponents of classical and operant conditioning and actually represent two starkly contrasting implementations of this area of behaviorism.

The In-Game Slot Machine

If you self-identify as a gamer or you're in any way exposed to video game press, then you may be aware of the concept of Loot Crates. Loot Crates, such as those found in Blizzard's *Overwatch* (Figure 5-1, called "Loot Boxes" in the *Overwatch* world) are almost a perfect implementation of the variable ratio reward model.

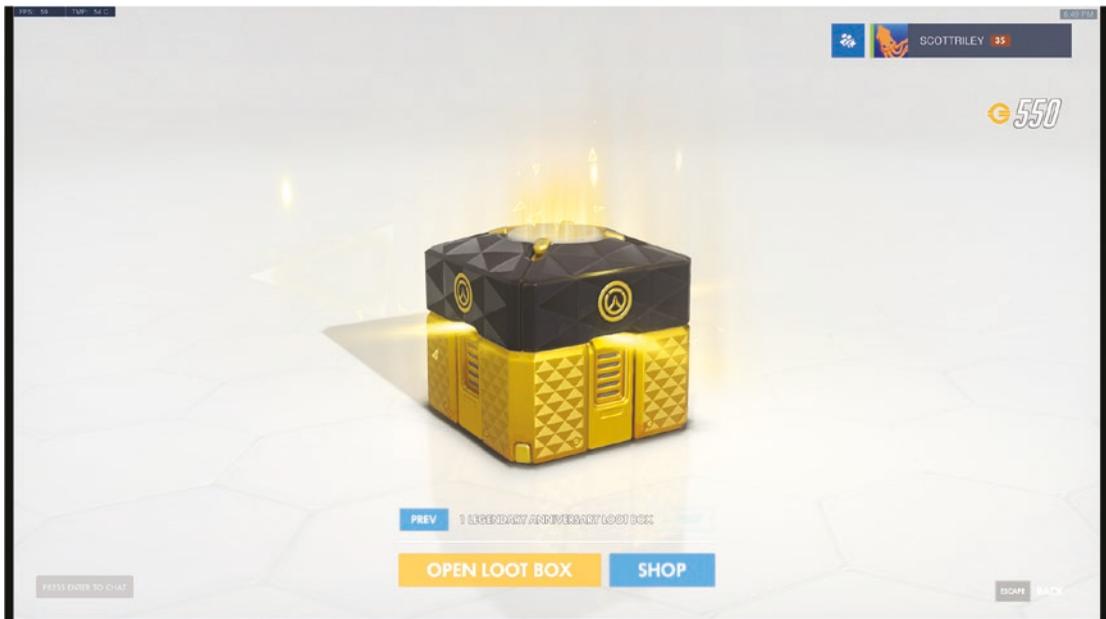


Figure 5-1. *A Loot Crate in Overwatch (Overwatch Blizzard Entertainment)*

The premise of a Loot Box is as follows: you purchase (usually with real-world money) some form of in-game “container.” You “open” that container (usually with a great deal of suspense and to some form of fanfare). You’re presented with random in-game rewards, such as new ways to customize your character, new weapons, and myriad other purportedly game-improving items. Usually, the rewards you receive are seen as “common”—less desirable and impactful rewards than the enticing ones available at a worse probability. This creates a dangerous paradigm where, often having already paid for a game outright, one can quote easily spend hundreds of dollars a *month* on chasing these super-rare rewards. I have my own history with the addictive nature of Loot Crates. Eventually, and rather embarrassingly, I had to apply cognitive behavioral therapy techniques to stop spending money on them in multiple games. Apparently, I’m not the only one who has had to take drastic measures to escape this trap, though. At the time of writing, there are serious and progressive discussions around the video game industry’s governing bodies about regulating (or even banning) this approach to further monetizing games.

Loot Crates are, quite simply, another form of gambling. Their implementation is taken straight from the handbook of addictive design, almost directly mimicking the slot machine model of reward and reward schedule variation. The fact that there is a consistent and severe backlash from gamers (EA's *Star Wars: Battlefront 2* caused a huge furor among fans and critics over its implementation of Loot Crate mechanics) and that the practice is under extreme scrutiny from governing bodies within the industry shows us precisely why such controlling, compulsion-focused practices are considered by many to be ethically dubious. This is the kind of company that “persuasive” design is keeping through its steadfast application of surface-level behaviorism.

Beyond the surface-level gambling of Loot Crates, video games do present us with a somewhat positive implementation of reward scheduling. With this topic, I feel we're most able to explore a more humanistic approach to reward-motivated behavior.

Intrinsic Rewards

On the more positive end of the spectrum, the whole gamut of reward scheduling can be seen at play during many a video game session. Video games are, inherently, *fun* things to engage with. They often have engaging stories, offer engrossing escapism, and provide challenges that play to our desire for mastery. In the environment of video games, the reward economy is traditionally discussed in a way that encourages progression through a game. Role-playing games (RPGs) do this exceptionally well. Generally, RPGs have various statistics that represent the skills of a specific character. Perhaps strength determines how much damage your little anime man can do with his oversized sword, or speed determines how quickly a turn can occur during battle. Figure 5-2 shows the classic RPG Chrono Trigger's rather simple stat list.



Figure 5-2. Chrono Trigger’s character stats determine the utility and effectiveness of various characters in various situations

Here we’re exposed to some of the raw numbers used in the game’s calculations for damage dealing (how hard you hit enemies) and reduction (how well you “soak up” damage from an enemy). While these stats start off at a specific number, they can be increased through *leveling up*, which involves battling (the game analogue to the “work” part of our reward flow) enemies and gaining experience points (EXP, one example of in-game rewards). After gaining a certain amount of EXP, a character levels up (Figure 5-3) and their stats are increased. Moreover, as a character levels up, there’s a *chance* that they can learn a skill, which can be used in battle. Generally these skill-learning moments are kept as a surprise.



Figure 5-3. Leveling up in Chrono Trigger

There's a *lot* going on here, so let's try and break this down. Once we have established that battling enemies can give us EXP, we have our first reward association. Yet, EXP is a pretty abstract reward. Pretty soon we'd start getting tired of it, first, because it's a known reward—literally every battle won garners EXP—and, second, because it's really just a kind of *sub-reward* of the real goal of leveling up. In this sense, we can see EXP as a fixed-ratio (it happens every time you win a battle) reward of low saliency (it's hardly life-changing).

After enough EXP, our leveling up moment happens—the *real* reward! Except, this is still an abstraction, what we're really chasing is the stat boost and the *maybe, hopefully, please* moment of learning a new skill. In this regard, leveling up can be seen as a variable-ratio reward (the amount of EXP required to get from Level 1 to Level 2 is different from that required for going to 2 to 3) of relatively low saliency (it's still an abstraction). The stat increases are a byproduct of leveling up, so we *know* every level up will give us a stat increase—we just don't know *what* stats or by *how much*. In this case, the reward itself is variable, while the schedule is not. We might say this is a fixed-ratio *variable quantity* reward of medium saliency. Finally, the new skill learning (Figure 5-4) is the "classic" example of variable-ratio rewards.



Figure 5-4. Skills in Chrono Trigger

New skills allow you to approach battles in a new way, to dispatch previous enemies with ease, and to implement their usage into your strategies for tougher enemies. We could stop there and say our new skills represent the upper limit of our rewards—they’re variable ratio rewards of high salience. Except, the *real* rewards from video games don’t actually exist within the games themselves but in the mind and world of the gamer. Just as the monetary reward of a slot machine represents an abstraction in that it allows us to purchase the things we enjoy, all this in-game goodness is an abstraction of the reasons *why* we play games in the first place. Getting stronger and progressing through a game lets us know that we’re further down that path toward mastery. It lets us express our creativity in the form of opening up more unique strategies for battles and enables self-expression through the choices we can make. These are video games’ real rewards, and they represent a rather interesting context in which to discuss reward-motivated behavior.

One of the primary notions to take away from this approach to rewards is that—even though the more salient in-game rewards might feel like real-world rewards (and, to an extent, they are)—they serve a very different primary purpose: they are a feedback mechanism.

When you provide someone with an idea of how much they've progressed through a game, you're giving them a heuristic for their mastery. Someone can take a look at their character stats and instantly get at least a rough idea of how far along that journey they are. They can view the skills they've amassed over the course of their play through and formulate strategies, or they get a broad idea of how their own preferences and skills can combine with the in-game skills of their characters in unique and fun ways. In-game rewards become milestones along an individual's path through a video game—mementoes of their journey so far.

Rewards as Feedback

We can learn a lot from this aspect of video game design. By understanding the true goals of their audience, video game designers take a holistic approach to their implementation of rewards. As we explored in Chapter 3, it's very easy to fall into the trap of confusing features or interactions with goals. Great game designers understand that “getting to level 99” is not necessarily a goal—it's an abstraction for something along the lines of “portray total mastery of this game.” Similarly, “learn the Omnislash skill” isn't necessarily the goal for skill acquisition. It's something like “learn an important skill that allows me to be more creative in my strategies.” The in-game rewards are always secondary and very often play the role of progress indicators rather than acting as manufactured rewards.

This takes us to the first and third mistakes mentioned earlier in this chapter. The idea that you can, or should, attempt to manufacture rewards in an application is flawed since it assumes that any manufactured reward will somehow be salient enough for someone to desire. Similarly, forgetting about the *real-world* goals people bring into any conversation with your interface is a surefire path to shallow implementation of rewards.

A good example of both sides of this approach can be seen in the language-learning app Duolingo. Duolingo is *gamified* in the sense that its mechanics are lifted directly from the rewards-based practices popularized by video games. When you first start learning a language in Duolingo (Figure 5-5), you're dropped in with 0 experience points and a “Crown Level” (which, honestly, doesn't exactly get explained very well) of 0.

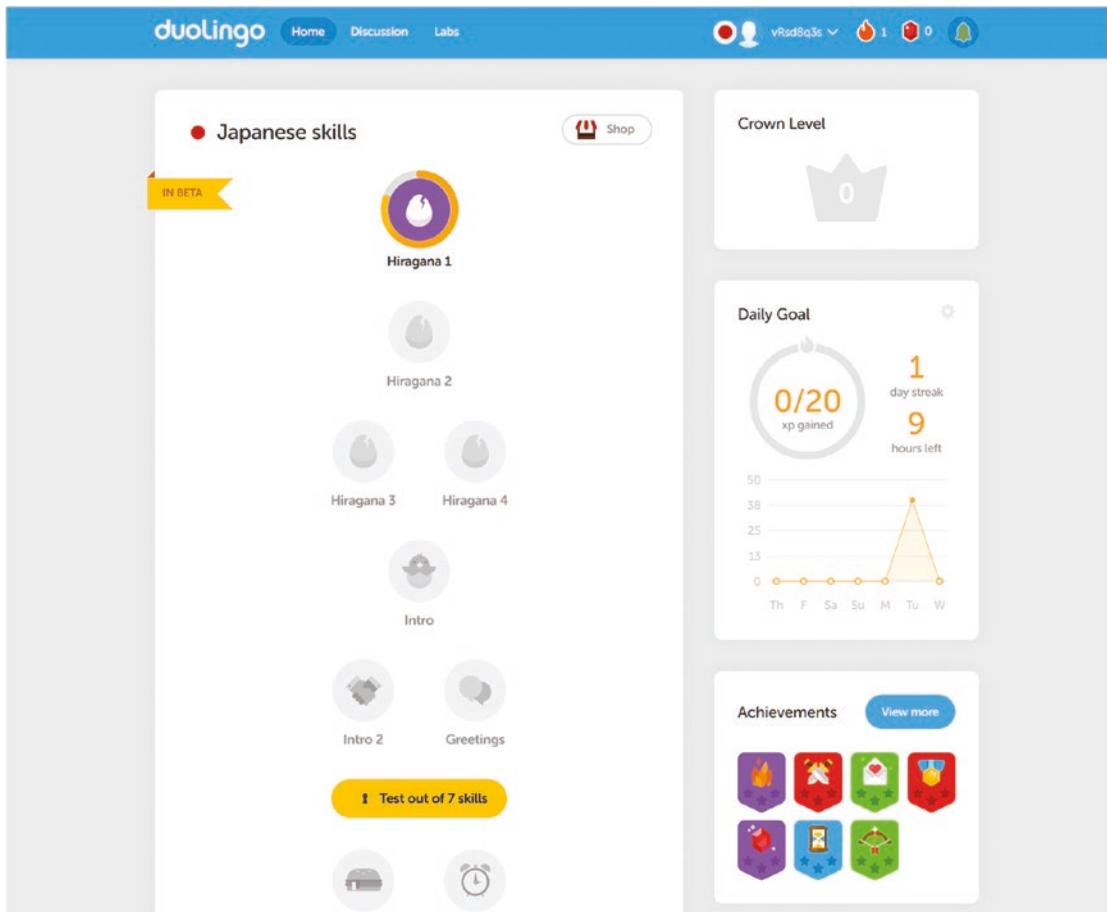


Figure 5-5. A (somewhat) fresh and untouched Duolingo dashboard

You're also given a clear path to mastery for your chosen language in the form of a vertical skills list. The Daily Goal section gives you a tangible reference point for the shorter feedback loop of daily learning, and the day streak gives you a reference point for how many days you've logged in and practiced. Now, to caveat this, the trend of daily streaks in digital products is quite often a poor manifestation of the behavior manipulation I've spent a good portion of this chapter attempting to discourage. I don't wish to gloss over the fact that I'm not personally convinced this is a positive implementation of the feature—or that a positive implementation of it even exists. Duolingo's approach is far from perfect.

Completing lessons in Duolingo, similar to winning battles in an RPG, gives us EXP, and enough EXP can increase our Crown Level (Figure 5-6), which we're shown contributes toward our daily goal.

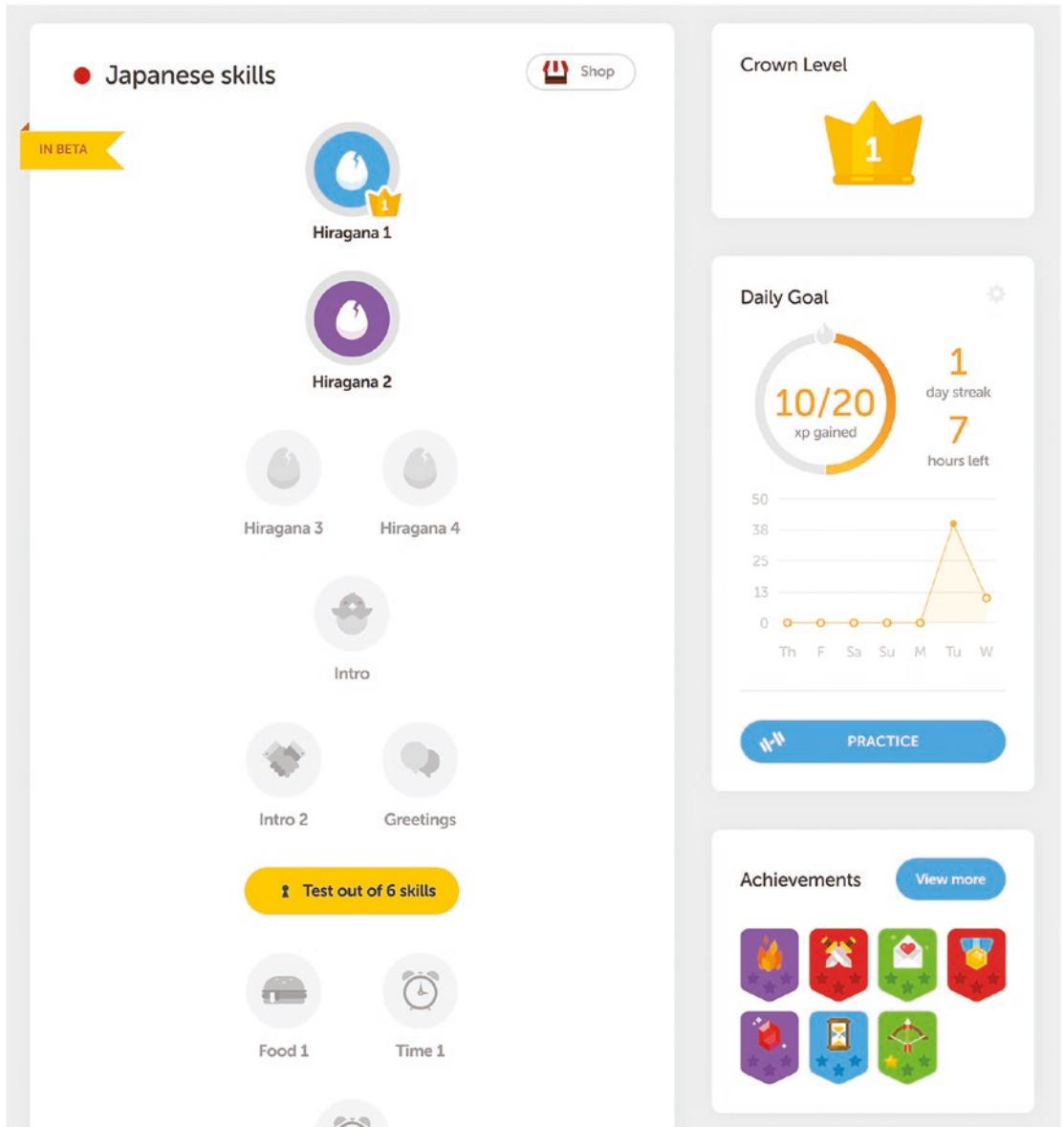


Figure 5-6. Crown Levels and EXP increases as we learn and practice a language in Duolingo

Aside from the dubious “streak” red flag, however, the most interesting aspect of this interface is how it manages to be reward-centric without confusing in-app milestones with explicit rewards. Duolingo’s designers understand that “increase my Crown Level for Japanese” is not someone’s real-world goal. It’s far more likely to be “learn the basics of Hiragana” or, pushing the envelope a bit further, “learn enough Japanese to survive in Tokyo for a month.” The crowns and EXP are *not* the rewards in Duolingo’s interface. The real world mastery of a brand new language and the social acceptance we may find from speaking our friends’ native languages are the true rewards of this. Crowns are the rather arbitrary abstraction for our progression toward mastery, and our daily EXP counter plays the role of allowing us to see our progress toward our daily goals. The daily streak is there, I assume, to make us feel anxious and compelled to return or to have us watch in dismay as it resets to 0 after daring to take a break from intensive language learning.

Rather curiously, the part of Duolingo’s interface that one might most associate with “typical” implementations of reward flows represents one of its most intrinsically irrelevant features. The Achievements section riffs on the common gaming approach of rewarding very specific in-game behaviors with achievements or trophies. These trophies are generally attached to a gamer’s social profile and represent a kind of ‘status’ indicator in the social realm of gaming. (Figure 5-7)

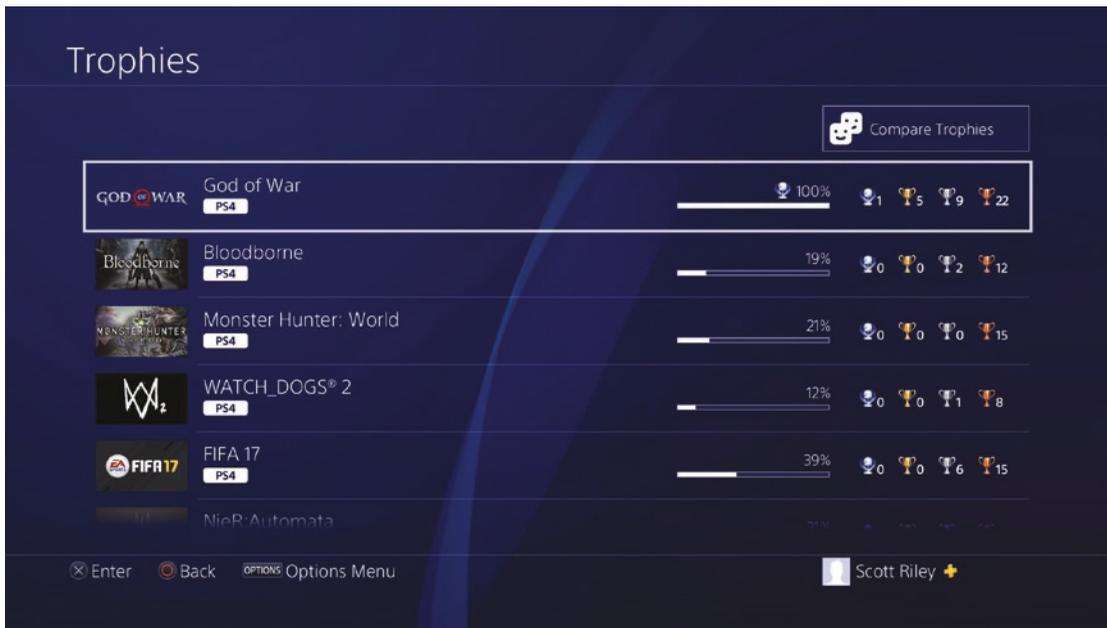


Figure 5-7. A trophy overview for a person on Sony’s PlayStation 4 console

That's not to say achievements unto themselves cannot be tied to intrinsic rewards trophies, and achievements are often tiered, with the *Platinum Trophy* being the primary indicator of mastery for many games. But, they are often attached to arbitrary actions that many gamers feel like they need to “grind” (repeat the same task over and over again for reasons beyond enjoyment) to achieve. Many of Duolingo's achievements are also somewhat arbitrary and superficial, such as the “Friendly” achievement. (Figure 5-8)

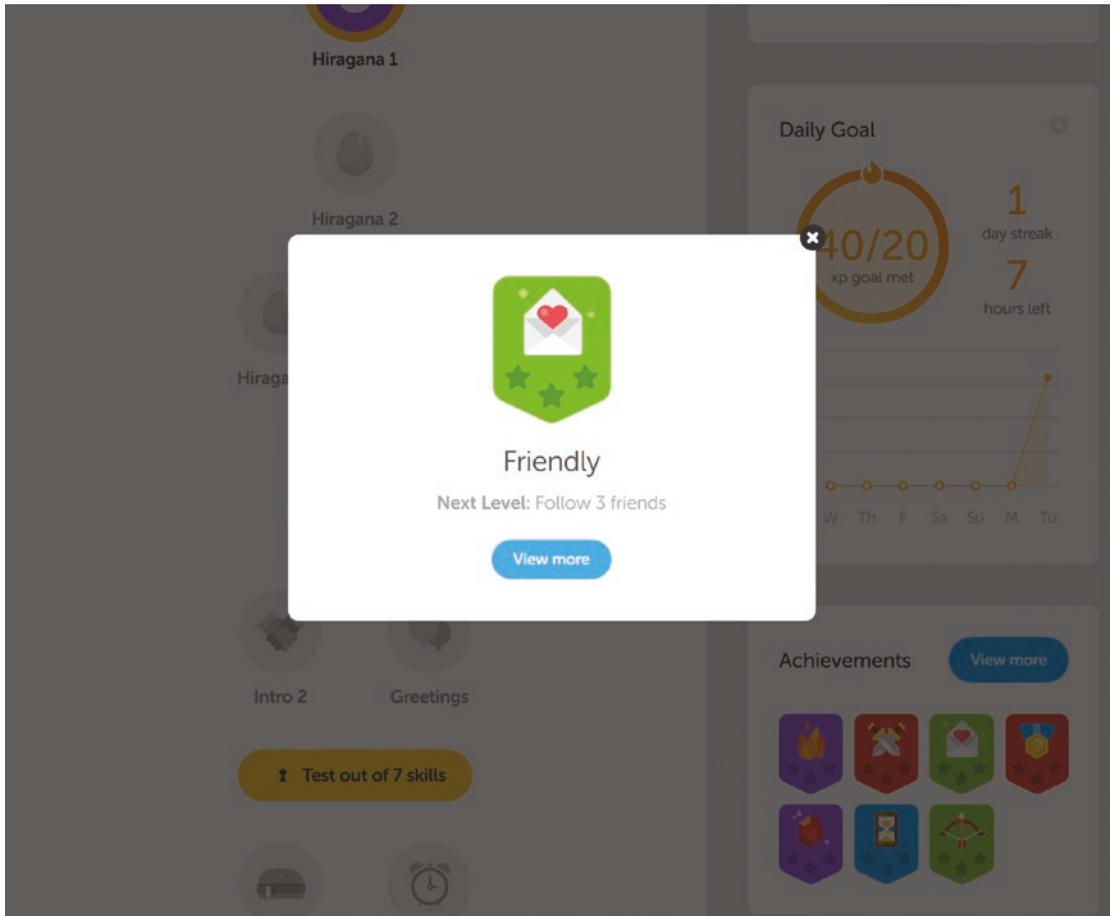


Figure 5-8. Duolingo's “Friendly” achievement, awarded for following three friends

This kind of achievement, alongside a few more seemingly arbitrary ones (“practice before 8 p.m.,” for example) seems to be positioned to change or manipulate behaviors for reasons that are, at best, overly abstracted from real-life goals. Learners are motivated by, we can assume, some salient factor that drives them to do the work required to master a new language. It could be argued that there’s a benefit, either through collaboration or competition, to building up a network on Duolingo—for example, wanting to beat your friend to the next level of learning—but for many people, this is either completely irrelevant or never explained. The idea of attempting to compel someone to practice in the early hours of the morning feels even more arbitrary. Again, this isn’t necessarily lacking in justification. Perhaps the idea is that implementing your daily language practice into your routine is an effective habit to form—but why 8 a.m.? And why is this a goal that’s incentivized to us right off the bat?

For the vast majority of Duolingo’s interface, we can use the game-like mechanisms as good indicators of progress and as milestones throughout our learning journey. The skill-tree approach to progressing through the different lessons is genuinely wonderful, and the daily EXP goal is a useful hint to get that all-important daily practice in. The Crown Level, while being something of a mystery when you first sign up, is an overall indicator of your progress toward mastery. It’s quite a shame, then, to see the standard approach of manufactured reward rear its head in other areas. Looking at the behaviors Duolingo tries to encourage with these shallow rewards tells us an interesting story on how Duolingo (probably) makes its money. It stands to reason that daily, continued use of the Duolingo service is important, which is standard for any product that generates ad-based revenue. Consequently, one can assume that the daily streak and the “practice before 8 a.m.” achievement are attempts at encouraging this. Friend following would likely involve us having to first invite friends, which plays into our desire for social belonging and social status, but also gets more potential eyes on Duolingo’s adverts.

Now, I am not suggesting that this approach to rewarding is *bad*—there’s a very tangible and direct benefit associated with continued use of services like Duolingo, and their implementation of advertisements does little to exploit a learner’s attention. Hence, it’s hard to begrudge them at least the room to experiment with manufactured reward. And there are a number of achievements that do actually refer back to intrinsic goals—the “Sharpshooter” achievement, for example, is awarded for progressing through a lesson without making a single mistake. Getting this achievement can easily be associated with progression and mastery. There is, however, a rather interesting divide between the rewards that can be easily traced to real life motivators and the rewards

that are quite obviously there to serve business needs. Continuing our discussion on mental models from Chapter 3, the skill tree, EXP, and Crown Levels represent an effective translation of system-to-human models, whereas daily streaks and arbitrary achievements feel more obviously like a representation of the system model. Duolingo is wonderfully designed and is a fantastic reference for any designer who might need to consider implementing a game-like experience or a rewards-based model of some kind—especially if your product involves fostering a prolonged and consistent learning environment.

The takeaway from this is to try, wherever possible, to recognize the value that our products provide. Only when we truly understand the value of our products can we hope to determine what people may or may not find intrinsically rewarding or enjoyable. From this viewpoint, we can start to explore potential goals related to the value our product provides. In Duolingo’s case, the value it provides is obvious—it’s an extremely well-made learning tool that gives people access to learning languages. This is an incredible democratization of technology, allowing learners to improve the skills that allow them to understand and contribute to the sharing of cultural norms and ideas and to better communicate with friends and family members around the world. The *real* goals of Duolingo’s learners are so exciting to break down and discuss—the couple whose parents speak different languages learning to communicate with, and become part of, their partners’ families; the socially conscious entrepreneur learning the appropriate language to communicate with their customers and business partners across the world; the anxiety-ridden worrier desperate to commit a few important phrases to memory lest they face embarrassment in a foreign country (this is me, hi)—all of these scenarios, plus the thousands more that language learners express every day, are saying something that we should never take for granted: “I want to use your product to achieve goals that mean something to me.”

When we understand this statement, we can see rewards in a different light—yes, the digital representation and abstraction of *a good thing* is still enjoyable, just as money as a path to enjoyable things is still a rewarding thing to receive, but the true emotional moment occurs far beyond the frame of our screens. It occurs in the actions, abilities, gestures, and thoughts of the humans who found value in our work. In these cases, what we traditionally view as “rewards” fill the role of emotional signposts and of progress indicators. They’re small, digital abstractions of skills and moments. These are things that do not exist without the underlying delivery of value—intrinsic and self-deterministic responses that can never be manufactured.

Humanistic Motivation

Humanistic psychology emerged out of, in part, a rejection of the behaviorist school of psychology we've discussed throughout this chapter. Humanism is centered around notions of self-exploration, existentialism, and self-determination. Similar to proponents of Gestalt School, humanists believed in seeing a person as a whole, greater than and uniquely different from the sum of their parts. This application of holism to the subjects of mind, consciousness, and our notions of self represented a radical shift from the deterministic, "mind-as-a-black-box" philosophies of behaviorism.

The principles of humanistic psychology give us a perspective from which to view motivation that is refreshingly antidotal to the crude and reductionist ideas we've borrowed from behaviorism. While the humanistic perspective has a rather narrow reach in interpreting the mind—its focus on qualitative and mandated non-scientific methods of study have seen it applied only to a small subset of the discipline—one of the areas it has had a notable influence on is motivation.

One of the most famous artifacts of the humanistic approach to motivation is Maslow's hierarchy of needs. Maslow's hierarchy focused on the idea that humans have a universal set of needs, the most-evolved model being, in order, as follows: physiological needs (water, food, etc.), safety needs (law, order, protection from the elements, etc.), love and belongingness (affection, intimacy, friendship, etc.), esteem (status, mastery, dignity, etc.), cognitive (curiosity, knowledge, etc.), aesthetic (beauty, form, etc.), self-actualization (personal growth and self-fulfillment), and transcendence. Maslow's hierarchy has been criticized from many angles. As a humanistic study, it relied on non-scientific means and suffers the same criticism as all humanism. Furthermore, Maslow's dataset was infamously limited, focusing almost completely on privileged white men such as Einstein and Beethoven. Add in the fact that we can plainly and routinely discredit the notion that these needs are hierarchical (for example, those who live in extreme poverty and in exposure to the elements are observably capable of expressing and being motivated by love), and Maslow's hierarchy is quite clearly unfit for means as a motivational theory. However, the importance of this theory as part of a humanistic perspective should not be overlooked—especially its role in bringing the notion of self into the discussion around needs.

The Self-Determination Theory

The Self-Determination Theory (SDT) is a relatively modern theory of social psychology that presents an extremely elegant model of intrinsic motivation. Seemingly, a spiritual successor to the humanistic theories of self-actualization, SDT suggests that humans universally possess three main psychological needs in order to feel a sense of mental well-being:

- **Competence** refers to our need to attain and display mastery, especially as a means of controlling situations and outcomes. When we display competence through an activity, we find ourselves intrinsically motivated to perform that activity. In cases where competence can be seen as a motivating factor (our Duolingo example above), creating environments that are conducive to developing and demonstrating mastery is crucial.
- **Relatedness** is our innate desire to interact with, feel connected to, and care for others.
- **Autonomy** is our intrinsic need to remain in control of our own lives, to feel a sense of causality, and to act within the expectations of our self-identity.

In the early experiments that led to the formation of SDT, a crucial observation on the role of extrinsic motivation in intrinsically rewarding tasks was made. When testing the results of introducing two different types of extrinsic rewards (money and praise), Deci (1971) found that, when performing tasks that subjects found intrinsically enjoyable, subjects that were given money were less likely to perform the tasks of their own volition at a later point after monetary reward was removed. Subjects rewarded with positive communication in the form of verbal praise, however, showed an increased likelihood to perform the task. These results were verified in a later study (Pritchard, Campbell, and Campbell, 1979) where again monetary rewards were shown to reduce intrinsic motivation to perform a task.

These findings suggest a concept that should be of the utmost importance to us when we consider motivation in our work: poorly chosen extrinsic rewards are *worse* than no reward at all for tasks that are intrinsically motivating, and the most effective extrinsic rewards are positive reinforcements. This component allows SDT to inform our entire earlier analysis of Duolingo and the broader idea of rewards as a feedback

mechanism. When something is intrinsically motivating, we should look to support the very reasons behind this motivation. When competence forms part of our motivation to perform something (e.g., learning a language), then trying to manufacture extrinsic rewards that fail to provide sufficient feedback (e.g., arbitrary achievements) can actually *reduce* intrinsic motivation, meaning people are less likely to return. SDT also shows that *negative* reinforcement (for example, attempting to “punish” failures within intrinsically motivated activities) serves only as a means of reducing intrinsic motivation. This is why the daily streak approach is something to be wary of. While initially it can be seen as a positive motivator to practice on a daily basis, the resetting of a streak back to 0 is a clear example of negative reinforcement. Imagine getting your daily streak up to 100+ days, only to have a major life event drag you away from your routine for a few days, and then coming back to your learning tool to find your streak completely reset.

One of the key underlying points of SDT is the idea of autonomy and control, specifically, the notion that feeling controlled is a demotivational and negative experience for us. When we see autonomy and competence as integral psychological goals, it further leads us down the line of nonlinear, environmental designs as opposed to strict, linear paths. In order for us to not reduce intrinsic motivation, we must foster environments that encourage and allow for displays of mastery and autonomy. This allows us to present a hypothesis that not only frees us from the narrow and myopic view of controlled behavior through compulsive dopamine loops, but also allows design to play a role in a much higher form of mental experience. By creating environments where intrinsically motivated behaviors of creativity, mastery, autonomy, and experimentation are applauded through positive feedback and interaction cues, we not only empower psychological needs, we can also, in the tiniest of ways, play a role in a journey of self-discovery and self-actualization.

Finally, in the spirit of the humanistic perspective and Gestaltism, if we are honest with ourselves and see that our products can neither meet nor reinforce the psychological needs presented by SDT, then perhaps the best contribution we can make is to simply be invisible. The attentional focus required to perform many intrinsically rewarding tasks is often intense, and many times the best design decisions we can make are the ones that actively reduce the presence of our products. Our work, then, as a part of a unified whole, takes the form of an enabler through invisibility and efficiency. Through refusing to distract and take up mental efforts, we allow intrinsically motivating activities to occur. This leads to the final mistake listed at the start of this chapter: the dangerous assumption that our work should be emotionally engaging at all. When we make delight a design goal, a reality that the design industry tends to insist on

almost by default, we can easily forget that beneath that lies an incredibly assumptive premise—namely, that the value brought by our product is deserving of the energy that emotional engagement costs. The underlying value of the products and services we design and the goals that people bring along with them should ultimately inform the “style” of the system we implement. For some, we can feasibly look to cater to truly intrinsically motivating environments—learning important skills, creating beautiful artwork, and simply playing a fantastic game are some that spring to mind here. Other times, however, we may wish to simply become invisible, allowing for an intrinsically rewarding experience through our absence. iA Writer, discussed way back in Chapter 1, a great example of this. Aside from an optional word count indicator, the interface fades to nothing, leaving just yourself and words. Control is maintained through simply getting out of the way. Quite likely though, you’ll be operating somewhere between truly intrinsically rewarding and utilitarian invisibility—and it’s important here to be honest about which parts of your system can feasibly feel rewarding and which parts simply do not require emotional engagement. Furthermore, it’s important to keep in mind that “do not assume your product deserves emotional engagement” is not the same as “make your product dull and lifeless.” There’s a balance! Helping someone go about a dull process (such as signing up for insurance) in a colorful and friendly tone can be fun, nice, and even pleasantly surprising—just don’t mistake that for delight. It doesn’t need to be.

The Ethics of Rewards

I want to talk more about dopamine’s involvement in all of this. It would be an exercise in referencing and repetition to reel off the different functions and roles dopamine plays in the body and the brain during our daily phone-checking and, quite frankly, for this level of discussion, it really doesn’t matter. Yes, dopamine is responsible for our reward-seeking behavior. Yes, an overabundance of dopamine is associated with a variety of mental illnesses. Yes, various addictive drugs can instigate euphoria by stimulating dopamine release throughout the brain. But *consciously* this doesn’t matter to us. We can’t consciously *feel* dopamine, insomuch as it would take literal superpowers to consciously differentiate which neurotransmitters are affecting your mood or actions at any given time. When people say things like “I enjoy the dopamine high of getting a whole bunch of retweets,” they can’t *literally* feel it at a neurological level. Neurotransmitters are not stimuli, nor are they emotions or anything else we could feel, touch, or describe.

Dopamine is not sentient. It is not some malevolent villain that sets out to get us addicted to anything and everything that might result in our being rewarded to some variable schedule. As we've seen, a genuine result of a lack of dopamine is actual starvation. The celebrity nature of dopamine has provided some strange, chemical scapegoat for the very real, very damaging decisions of *people*. Whether this is by accident, malpractice, or purposeful manipulation, we must acknowledge that we are suffering at the hands of human negligence, incompetence, or greed—through one of many organic chemicals that makes us go and find food, or lets us reach out a hand, or helps us lactate.

Fortunately, just as operant conditioning represents a tiny slither of psychology, so too can the pursuit of extrinsic rewards represent a tiny slither of our application of psychology to design. A humanistic approach to research and an understanding of self-determination offers one of many alternative routes to creating environments where reward and motivation can be discovered and applied. Furthermore, we can and should embrace the *avoidance* of attempted forced motivation. Great design can lie simply in the acceptance that many times our products plainly do not need any explicit plan for reward-motivated behavior—whether that's because they're autotelic and provide their own intrinsic fun or value, or simply because the value they *do* provide is in their invisibility and essentialism.

Summary

This chapter delved into the rather murky waters of reward cognition and motivation. We observed how the tech industry, through concepts such as persuasive design, attempt to utilize the findings of behaviorism to create slot-machine-like experiences. In using variable-ratio rewards, many applications attempt to instigate a dopamine feedback loop of sorts.

We learned that dopamine is responsible for reward-seeking behavior and the notion that it is a “pleasure chemical” is deeply flawed.

Video games provide a fascinating insight into the good and the bad of reward cognition—with Loot Boxes adopting the same slot-machine, variable-ratio reward mechanics as many social media applications. Inversely, they also offer a fascinating insight into a more humanistic approach to reward by using rewards as a feedback mechanism and an indication of progress toward mastery. By feeding back on innate

desires for self-expression, mastery, and autonomy, we see that the true rewards of video games exist in the minds of the people who play them.

We further explored this idea by delving into the humanist school of thinking, where references to self-expression and self-actualization lead us further away from seeing the brain and mind as pesky black boxes in a simple behaviorist cycle of environmental influence leading to action.

The Self-Determination Theory (SDT) presents a modern, humanist-influenced theory of intrinsic motivation, suggesting that we inherently require competence, relatedness, and autonomy as basic psychological needs. By providing a well-researched counterpoint to the reductionist, behaviorist approach to motivation, SDT offers us a much more design-friendly framework within which to operate. By creating nonlinear environments and actively eschewing the need to even attempt to control behavior, we can create interfaces that help people satisfy their innate desires for competence and autonomy.

Furthermore, multiple experiments relating to the SDT theory show that poorly managed extrinsic rewards can actively damage how intrinsically rewarding someone finds a particular activity.

The single most important idea in this chapter (and I would argue, this entire book) is that attempting to control behavior—especially through reductionist, behaviorist motivation concepts—is a practice that should be met with the utmost scrutiny. That, as supposedly human-focused designers, we have a responsibility to the people who find themselves using our creations—a responsibility to contribute positively to their lives, to do no harm, and to protect them from practices we believe to be manipulative, especially in the name of sheer profit.

In SDT, we have an alternative theory of motivation that is inherently more compatible with a human-centered approach to design than the reductionist notion that dopamine loops are the golden geese of design psychology. In embracing humans as wholes, in all our fallibility, we're able to focus our craft on understanding, maintaining, and enhancing our notions of self. To further propagate dangerous behaviorism in the name of profit and habituation is tantamount to accepting that the very basis of our work, creating tools for humanity, is a lie. Profit can exist without addiction and manipulation, and we have the tools, and the science, to prove that.

CHAPTER 6

Designing Environments

I've never been particularly fond of the word “user.” Despite its permeation in the digital world, there's something inherently cold about referring to human beings as “users.” I'm even less fond of the term “User Experience (UX) Design.” While I don't disagree with the core concepts of UX Design, and I feel that the artifacts produced and the research conducted in these practices are extraordinarily important, I can't shake the idea that “user” is reductive and “experiences” cannot, really, be designed.

That first point—that “user” is far from a flattering term for humans—is hardly controversial. The idea, though, that UX Design isn't really “a thing” is something that is likely to produce more than a touch of ire. Allow me, if you will, the indulgence of elaborating on this point and suggesting an alternative approach that I feel is far more applicable when implementing the information from the first part of this book and, thus, far more integral to the idea of mindful design. First, though, I believe it's important to look into the history and develop a working definition of UX to use as a platform for this chapter's discussions.

What Is UX?

User Experience, as part of a job title at least, is commonly traced back to Don Norman (many of whose ideas and phrases have been, and will be, discussed throughout this book) who, in 1993, became Apple's first “User Experience Architect.” Norman's book, *The Design of Everyday Things* (1988) is seen as a seminal book on the psychology of design and is likely one of the most-cited design books ever written.

UX, broadly speaking, is a design methodology: a series of activities and tasks as well as an ethos that is designed to bring emotion and cognition to the forefront of a design process. A UX process will often involve research, planning, sketching, prototyping, and iterating. These general activities will almost always involve many more granular activities, such as interviews, persona creation, wireframing, and paper prototyping.

UX design is an incredibly broad practice. Finding one solid definition shared among individuals, companies, or even industries as a whole is incredibly difficult. As a UX practitioner, depending on the granularity of your role, you might find yourself conducting and facilitating research one day and designing the minutiae of an app's interactions in a motion design tool the next. Like many of the supposed catchall terms for early-stage design methodologies, "UX" seems to suffer from a crippling vagueness. To some, it's a very prescriptive set of tasks and deliverables that essentially amounts to "everything up until the wireframes are done." To others, it's an amorphous, elusive blob of a concept that "describes every interaction a person has with a product, application, or company."

As the design field has evolved to incorporate more "scientific" concepts, such as phenomenology, research methods, anthropology, and experimentation, so too has the number of acronyms and proposed sub-disciplines that fall under the broad umbrella of "design that involves more than how things look and move." A brief look at design disciplines right now reveals the following practices, most of which occupy *extremely* similar intellectual spaces:

- User Experience Design (UX)
- User-Centered Design (UCD)
- Human-Centered Design (HCD)
- Activity-Centered Design (ACD)
- Customer Experience (CX)
- Service Design
- Design Thinking

Similar to UX, we have User Centered Design (UCD), Activity Centered Design (ACD), and Human Centered Design (HCD). While UX is popularly seen as a *methodology*, UCD, ACD, and HCD are viewed as overarching *frameworks*. The difference is subtle and arbitrary, but still somewhat important. UX, by its most accepted definition, is predominantly task- and deliverable-based. Wireframes, interactive prototypes, research notes and competitor reports are all potential artifacts of a UX process's activities. The aforementioned frameworks, however, present themselves more as *mindsets*. They strive to guide the focus and thinking we apply to our approach, without dictating the tools and artifacts of our approach.

This list merely scratches the surface of potential design disciplines at the time of writing this book and doesn't include the more easily defined disciplines such as graphic design, interaction design, and motion design. If you're new to the design industry, this can all feel *extremely* overwhelming. While the proponents of each discipline might feel passionately that theirs is the One True Method, the reality is that the differences between these practices are often arbitrary and rarely well documented. You can forge a successful career in design without ever really needing to define these concepts. While they might provide us with a slightly clearer direction or a manifesto to get behind, not knowing the ins and outs and minute differences between such amorphous concepts isn't going to hinder your progress!

Note UX is often restricted to the *digital* areas of a design process, while UCD and HCD are often more holistic and involve buying into a specific mindset, approach, and underlying goal. As IDEO's excellent *Field Guide to Human-Centered Design* suggests,

Embracing human-centred design means believing that all problems, even the seemingly intractable ones like poverty, gender equality, and clean water, are solvable. Moreover, it means believing that the people who face those problems every day are the ones who hold the key to their answer.

The key thing to keep in mind is that UX, ACD, UCD, and HCD served to bring qualitative and quantitative research into design processes, popularizing activities such as user research, storyboarding, user persona creation, and journey mapping. Essentially, the goal of all of these concepts is to bring a focus on humans and human problems into the forefront of our design and development processes

Design Intangibles

The last two on our list, Service Design and Design Thinking, are, at the time of writing, extremely popular concepts. Defining these two terms has spawned hundreds of medium articles and blog posts. They're also a lot more theoretical, intangible, and broad practices, generally involving the whole spectrum of a human's interaction with an entire system.

A good example of this would be if we were to consider the design of a museum. The whole end-to-end flow of ticket purchase, finding and accessing the venue, knowing which way to enter, finding a path through the various exhibitions, exiting through the gift shop, buying an overpriced scale model of a prehistoric ground sloth, finding the bathroom, exiting the building, and finding your car could be overseen by a Service Designer. At various points, UX would play a key role in the museum's digital offerings (if we accept the common notion that UX is restricted to digital "experiences"), perhaps in a companion app that allows for interaction with the museums exhibits that could include audio commentary and maybe even augmented-reality additions to the real-world experience. HCD might come in to play when planning the ideal order that the exhibits would be visited in, either by focusing on a path that brings the most interactive, most exciting exhibitions in at a "sweet spot" along the journey, or by structuring them in a way that allows the knowledge from one exhibit to inform the experience of another. CX, then, could be arrows on the floor or a natural, followable path through the exhibits.

It's highly likely that throughout our museum visit, we'll encounter other, more traditional design disciplines such as the graphic design of the exhibition summaries and a printed museum map. Furthermore, we'd also need to consider accessibility—both physical and digital—by ensuring we have things like wheelchair access, Braille translations, and audio descriptions for exhibits in multiple languages. Additionally, there's likely a whole bunch of business- or industry-specific red tape to design around. A beautiful museum is useless if it's a death trap that defies multiple health and safety rules. And we'll almost definitely need to consider how we're going to place security cameras and alarms to keep our bosses and insurers happy.

While this book is *probably* a UX book or perhaps (and if you ask me) an HCD book, a lot of the information from Part 1 *can* be applied elsewhere. I do, however, want to discuss the implications from Part 1 of this book mostly in relation to digital products—which leads us to focus primarily on UX and HCD. This book revolves around making decisions that affect the workings of digital products, with a focus on the real-world impact of those decisions. However, I do believe the discussions and ideas around Service Design, in particular, are worth exploring. Accepting that design has never started or finished on a screen is an important idea to keep in mind throughout our careers.

Another important reason for this limitation is that I'm trying extremely hard not to propose a new discipline or a new job title into this mix. Everything I'll propose in the remainder of this book is designed to piggyback on, or bridge the gap between,

existing stages of common UX approaches. When I talk about “environmental design,” for example, I’m not suggesting we adopt an entirely new design framework that we have to sell to our bosses or our clients. I absolutely do not want to spark an “Environmental Design” movement. I want to offer a few, slight mental shifts that I feel can enhance and re-frame many of our existing design practices in order to greater assist and respect the idiosyncrasies and fallibility of the human mind.

With all that in mind—let’s try to broadly define “UX” as a methodology of design, with a focus on psychology, research, usability, accessibility, testing, experimentation, and the human impact of our work. Some UX artifacts we’ll likely encounter in our day-to-day work might include the following:

- User research documentation
- User personas and user scenarios
- Wireframes and interactive prototypes
- User journey diagrams and sitemaps
- Usability reports
- User testing documentation

And while this list is *far* from exhaustive and the boundaries of a typical UX job (especially on smaller teams) are always blurred—information architecture, accessibility audits, UI, interaction design, and sometimes even copywriting and front-end development can fall into the gamut of common UX designer tasks—to discuss even this short list of deliverables in depth would be an undertaking requiring an entire book unto itself. Fortunately for us, there exist a plethora of UX-specific books that do just that, including Leah Buley’s *The User Experience Team of One* (Rosenfeld Media, 2013) and *UX For Lean Startups: Faster, Smarter User Experience Research and Design* (O’Reilly, 2013) by Laura Klein, to name but two. Furthermore, formal and informal UX courses and certifications exist in abundance—with a seemingly endless list of online courses and formal academic syllabi to choose from.

All of this is to say, while this book intends to present ideas that attach themselves to existing UX practices and processes, that presenting a deep-dive, exhaustive introduction to such is beyond its scope. Now, you don’t need to run off right away and clue yourself up on all and sundry related to UX—we’ll touch on many of its key ideas and artifacts as we progress through this book—but I’d thoroughly recommend that after—or even during—reading this book, you jump into some short beginner or

refresher content on the follies of UX design if you're feeling a little lost or forgetful when things such as wireframes, user testing, personas, and journeys are mentioned.

With a hopefully less-murky working definition of UX design, I'd like to propose what I believe to be an integral problem, both with the terminology and with the underlying practices of the discipline.

The Folly of UX: Control and Linearity

I discussed at the start of this chapter my general distaste for the word “user”—it's cold and assumptive and just, well, a crappy way to talk about *people*. It also serves to create some cognitive and emotional distance from the actual human with their actual human brain and their actual human feelings, beliefs, talents, flaws, and heuristics. For some reason, at some point, someone arbitrarily decided that the correct way to talk about the people who interact with the things we make was to refer to them as “users”—and it's stuck, and fighting it is fighting a losing battle.

Just to add to my cynical gripe list, I also really dislike the word “experience”—at least in this context. “User Experience Design” implies, intentionally or not, that experiences themselves can somehow be designed. This is a concept with which I take great umbrage—an experience is a combination of infinitesimal emotional responses, a conscious and subconscious self-deterministic processing of one's self and one's place in time, in space, in life. An experience happens both in the minds of the experiencing humans and the environment in which it is facilitated, and it impacts these actors in unpredictable and immeasurable ways. The outright hubris it requires to suggest that we are, to any degree, in control of the actual experiences people may have throughout even a single interaction is, to me, astounding.

Words are important, but there comes a point when one has to accept that an industry has its standards, and a big part of such standards are the arbitrary terms, job titles, and never-ending lists of acronyms that attempt to describe its machinations. UX is *a thing*, it has been accepted and embraced, and its impact on the design world has been, on the whole, extraordinarily positive. I do, however, feel like in our pursuit for control, and in the potential hubris that has seeped from the term to its practitioners, we've taken the discipline to a point where best practice, process efficiency, and ease of education has led us to an intellectual space that is a little too preoccupied with *control* and *linearity*.

This, really, is what I believe to be the root problem with the UX methodology at the time of writing. In trying to control as many aspects of an experience as possible, we've maneuvered ourselves into creating products and interfaces that are just far too linear. Without realizing it, the artifacts of our profession play directly into habitual restriction and attempted control. User journeys, for example, tend to be incredibly linear, as do the various "flow diagrams" we're wont to produce. Looking more deeply, even the atomic forays into wireframing and prototyping often tend to focus on key "paths" through an interface, and user testing sessions often focus on the speed at, and efficiency with, people can complete these atomic, de facto paths. When people get confused, we add more wayfinding aides, we limit auxiliary information and tasks and we double-down on focusing on these key paths or points along them.

One of the key ideas that I hope Part 1 communicated is the notion that self-expression, displays of mastery, and perceived autonomy are key factors of intrinsic motivation. Oppressive linearity makes such activities and emotions impossible to achieve and experience. While clear wayfinding and constrained flows are critical moments of any interface, they are, quite often, far from all that is required. As we discussed in Chapter 3, when we spoke about different methods of creating learnable interfaces, linear paths and constraining of possible actions can be fantastic *tools*, but they should be utilized conservatively, and only when genuinely required. For many of the complex, interaction-rich digital products that we presently create, linearity should not be our default stance.

Quite possibly the loftiest goal of this book is to present a solution to this problem that *doesn't* brand itself as a whole new acronymized methodology with its own manifesto and a thousand arbitrary deliverables that you have to sell upward to your bosses and clients. I believe that by focusing on creating digital places where interactions can simply happen; by eschewing linearity and pathfinding as unquestioned defaults; by designing moments and by seeing our work in terms of states, transitions, and models, we can start to hone a mindset that allows us to view our current processes and artifacts in a different light. We can create much more open interfaces where self-expression and mastery can occur, where edge-cases melt away, and where personas become a challenge to the status quo, rather than yet another tool of limitation and linearity. In my head, I call this "digital environment design"—but you don't have to.

What Is a Digital Environment?

When I talk about designing an environment, I'm referring to—in the simplest terms—a place where *stuff can happen*. Crudely speaking, an environment acts as a container for actions and interactions, and it updates itself to reflect the results of such. Environments, unlike experiences, *can* be designed. They can have explicit and implicit rules, just as our world employs rules of physics like gravity or our society has concepts like manners and cultural convention. Environments allow moments to occur over time, creating a rhythm based around human activity and response. They allow for expression, introspection, and creativity where applicable. Conversely, they contract into linear, obvious paths when necessary. Environments have a breath and a cadence that is inextricably tied to the actions and desires of the people exploring them, and I feel they represent the culmination of everything we've learned so far in this book. The remainder of this chapter will focus on the slight shifts in mindset required to make a dramatic change in the way you approach design.

Broadly speaking, in addition to our existing tasks and methodologies, this approach to design involves balancing linearity and nonlinearity; designing rules; treating design as a function of state; reconciling system, conceptual, and mental models; and designing *moments*. This may seem like a lot of extra work, especially when we consider how involved a typical UX process is already, but the vast majority of it can be achieved using the tools and approaches that you've either already learned (if you're an experienced designer) or that you're currently studying right now (if you're still coming to grips with the industry). Most of these ideas can manifest themselves as sketched diagrams or wireframes as part of a standard prototyping process or as a form of design and decision documentation.

Balancing Linearity and Nonlinearity

As discussed earlier, the linearity of an interface plays a substantial role in its suitability to the self-deterministic motivators of control, autonomy, and mastery. One of the key considerations when planning your work will be its *cadence*. Rather than focusing on neat, end-to-end flows or journeys, think about broad moment-to-moment *acts*. The key to this approach to design lies in harnessing time and progression to create an environment that feels authentic, no matter what order things happen in and no matter how someone prefers to interact with its constituent elements.

The closest analogy to this approach can be found in video games' level and world design. You might have noticed by now that this book leans heavily on the teachings and examples of our colleagues in the games industry. As creators of open, interactive experiences, their influence on our work, as technology and expectations grow interminably, is becoming more and more unavoidable. When a level design team sets out to create an environment in their game, they rarely start thinking in terms of linear paths. Video game worlds are designed to be explored. They're created with a set of rules that determine what, and in what way, one actor can interact with another and how the environment may shift in response to that action. Progression through a world in a video game doesn't necessarily occur over time, as we'd expect with a movie or a piece of music; it occurs based on interaction—one action results in a change to the state of an environment, and the environment shifts to reflect the new state. Rinse and repeat. The parallels to world design in video games and my proposed digital environment design are stark.

Linearity

When it comes to high degrees of linearity in our work, we've fortunately got it (mostly) figured out already. Common design practices and processes lend themselves very nicely to planning and creating linear flows and paths. Linearity is often a result of subtraction—an act that many people believe to be the core of design itself. By limiting options and removing an apparent overabundance of choice, we've found ourselves remarkably adept at constructing interfaces of singularity—designs that allow for the extremely efficient performance of single-to-a-few actions.

There are a number of downsides and dangers, however, to this approach. To quote Tristan Harris, "If you control the menu, you control the choices." Our approach to manufacturing linearity in our interfaces is often one of restriction. If an action that our products are quite capable of allowing to occur is not seen as "important" (which often is simply a replacement for "profitable"), then we can just throw it in the bin. Rather than creating alternative, less-invasive ways to allow that action to take place, we've formed a habit of ruthless subtraction. This isn't a negative habit at all—I'd be a hypocrite of the highest order given the points raised in the first part of this book if I were to suggest that removal-by-default is a poor design ethos—but this ruthless removal isn't always the *right* tool to throw at a situation.

Harris’s menu metaphor—the notion that our interfaces, at any one time, can resemble a menu of specific options in the form of actions a person can perform—gives us an interesting heuristic for measuring the linearity of our interface at any given moment: *linearity is a function of all available options*. At any point, we want to be able to look at the options for action that our interface allows, regardless of how obvious they are or how they are performed, and use this number as a proxy for linearity. A small number of possible actions represents a limited menu and thus a high level of linearity. An abundance of options represents a more open, less linear state.

Simply adding or removing potential actions depending on application state, while a good start, is not going to get us far if we want to elegantly manage the linearity of our environment states. Clearly, too, some actions—regardless of state or linearity—will *always* be more important than others. We’re not about to start sacrificing hierarchy and categorization in some naive pursuit of openness, and it remains critical that we continue to apply our understanding of attention, information overload, and decision overload to our work. As an extremely broad metric, though, option abundance can provide a useful initial insight into how linear our application states may be.

Designing Linearity

Most of what we explored in Chapter 3 can provide us with the tools required to manipulate the linearity of an interface. Concepts like setup wizards, constrained onboarding flows, and streamlined checkout processes all utilize linearity to limit the available options and, really, that’s what most of the linearity of our environments boil down to: how many actions do they *allow* someone to perform?

In the case of an extremely constrained onboarding flow, such as Google’s Hangouts for iPad onboarding that we looked at in Chapter 3 (shown again in Figure 6-1), linearity is created through strict reduction.

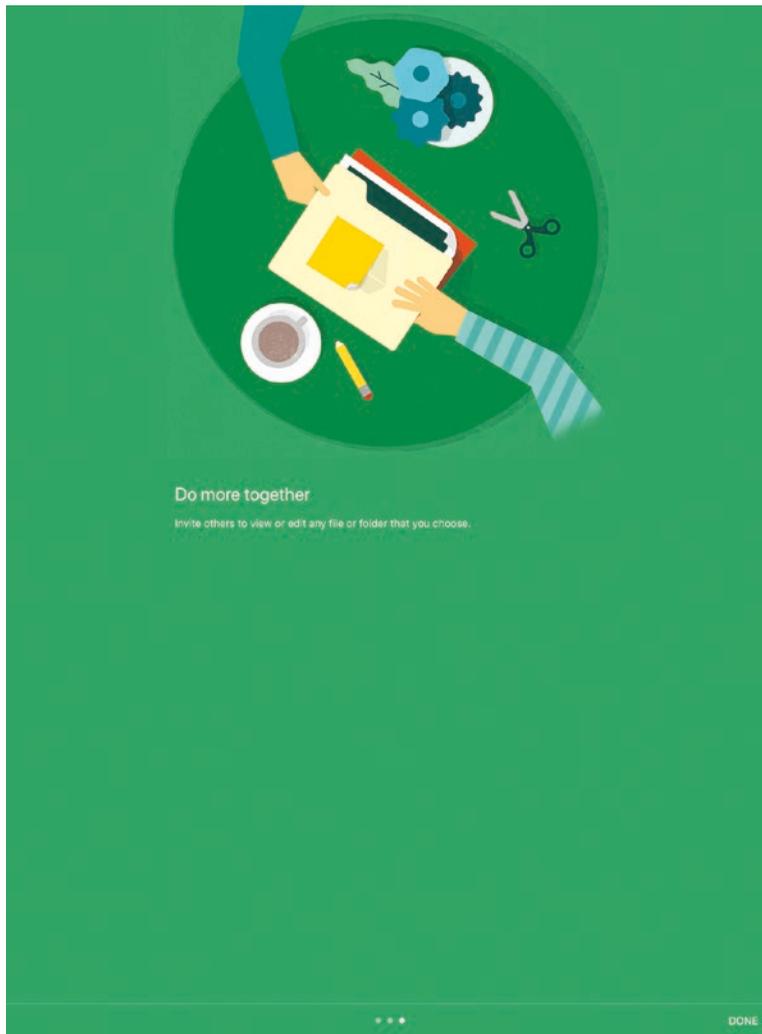


Figure 6-1. Google Hangouts' iPad app onboarding

In this flow, we're constrained to a relative paucity of actions and information. The copy is extremely terse; there's a single, simple illustration; and we can either click Next, or swipe left or right to navigate between sections, or click Skip to leave the onboarding flow at any time. By limiting options in this way, Google is "controlling the menu." As we discussed in Chapter 3, onboarding can be an extremely valuable learning tool, but with Google's example, we see it used almost as a piece of marketing design. We're essentially being invited to scroll through a feature list one last time before we start using the application.

This has its upsides, for sure. For one, it definitely sets expectations as to what the perceived key features of the application are. It could also act as a sort of *primer* for the app’s audience, essentially a way of saying “These are the actions you can expect to perform.” However, as we explored, by removing so much context and not allowing concepts to be interacted with in a way that allows for deeper processing of information, its effectiveness as a learning tool—which, really, all good onboarding *should* be—is limited.

Similarly, look at Amazon or any other shopping site, and you’ll see their checkout processes greatly limit the number of potential options we can perform. During normal shopping, we’re inundated with navigation, categories, related items, sales, and deals. Yet, when it comes time to check out, we’re mostly limited to a few basket-related options, such as changing item quantities, removing items, and completing the checkout process. In fact, for all the talk in this book—and in the industry—about innovative, interactive applications, online stores are one of the best examples we have of linearity being manipulated at various points throughout their usage.

By allowing for exploration and an abundance of (and often far too much) choice, online stores like Amazon provide an open and almost unlimited initial state. At various points, and mainly at the checkout stages, that openness transforms into a constrained, controlled, and limited linear flow. Once the purchase is complete and confirmation is shown, the environment opens up again and more choices—more potential actions—are presented. Shopping apps tend to have a very specific cadence, or rhythm, to their linearity.

A linear flow, then, is one of constraint and rescinded control. At any given point, should we wish for our environments to enter such a state, we need to be sure of a few things. First, that there is indeed enough implicit intent to justify this rescinding of control. Second, that the transitions between any preceding, open states are managed well. We’ll discuss the latter point later on when we start looking at dissecting interactions, but understanding intent is something that can very quickly be assumed, tested, and proven. Essentially, we’re looking to deduce, based on the statement behind previous actions, what someone wants to get from our interface at that moment in time.

For a shopping app, this is quite simple. We’re often presented with a Checkout button either directly on a product page, in our basket view, and usually in both. The clicking of this button communicates very clear intent, essentially stating, “I’m done with my browsing and now I’d like to pay for this stuff.” For other apps, this can be a

little more difficult to infer, but for every call to action, link, or button label you have, try to write a succinct, one-sentence summary of the intent behind the interaction. This is also a handy method of finding any vague, ambiguous action items. If you can't clearly deduce the intent behind a specific action, try to analyze why. Oftentimes it'll be poorly worded button text, such as Submit or "Go.," Sometimes it'll be the elements or copy that surrounds the action point, such as the alert mockup in Figure 6-2.

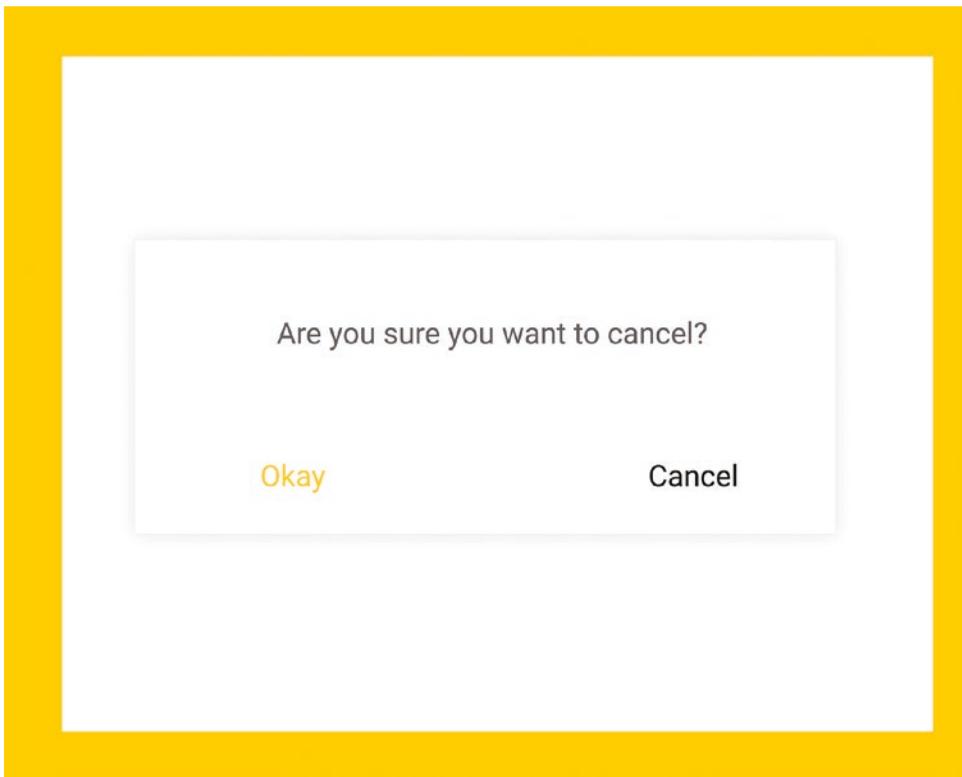


Figure 6-2. *Are you sure you want to cancel?*

Alerts like this are unfortunately common and could be prevented rather easily with a simple audit of the copy, assumed intent, and actual action.

Gauging intent is globally important (and will be discussed throughout the second part of this book), but especially when ushering someone into a linear flow from a more open, explorative environment. This comes purely down to the parallel between increased linearity and reduced control. By limiting options and directing someone

through a constrained path, you're essentially taking control away from them. Do this for long enough, and you run the risk of people becoming disinterested and losing motivation. Once you're sure that a linear process is necessary and you've determined the essential elements and features for this linear flow, keep the following optimizations in mind:

- *Limit Options.* The first step to linearity is to limit the available options and actions to the bare minimum required. When entering into a linear flow, we can reasonably assume that we must cater for a period of focused attention. That means being ruthless with the options we present and removing any distractions, clearly grouping and spacing related elements, and ensuring that copy is terse and clear.
- *Understand the Path to Completion.* By their nature, linear paths have an ideal endpoint. Keep this endpoint in mind and constantly question how any decision you make allows for progression toward this.
- *Display Progression.* Similarly, find ways to communicate someone's progression through this process. "Linear" doesn't always mean *short* and doesn't ever have to mean *compressed*. In many cases, our linear flows might be broken down into multiple smaller steps (similar to the Google Hangouts onboarding discussed earlier). Whether our linear flows are explicitly stepped through or progress is a little more ambiguous, we should look to communicate *how far* toward this aforementioned end goal someone is at any point. This can be done through explicit progress indicators, such as numbered steps or progress bars, or through implicit indicators, such as Almost there! prompts.

- *Be Explicit and Timely.* Linear moments in interfaces suffer immensely if ambiguity is introduced into the equation. Furthermore, delayed responses—such as waiting until the end state of a checkout flow to show an error in the first section of the address form—can be equally as catastrophic. Shallow processing and linear flows go hand-in-hand, so ensure that any signifiers and mental models used in communicating concepts are as universally recognizable as possible. A truly linear interface, although rarely encountered in real life, would rely exclusively on recognition and not require any learning or memorizing at all.
- *Design the End State.* One of the most important aspects of any flow, but specifically a linear one, is how we present the success or completion state. If the flow involves some form of deletion, then take the learnings of empty states and apply them here. If it involves the creation of data, such as a new Tweet, then present an 'All done!' message of some kind and, wherever possible, *show their creation* in context. If someone has given up control to follow a constrained and linear path to creating, removing, or editing something, the least we can do is let them see clearly that their changes were successful.

Nonlinearity

A less-ventured concept, in digital design at least, is that of nonlinearity. If our understanding of a linear interface state is that of limited options and constrained paths, nonlinearity may take the form of an abundance of potential actions, or a central “hub” of sorts, from which many paths can be explored. Good examples of openness and nonlinearity in interfaces can be found in many products that revolve around creativity and self-expression. Photoshop, which we analyzed in depth in Chapter 3, is a good example. So too is Reaper (Figure 6-3)—a music recording and production application from Cockos.

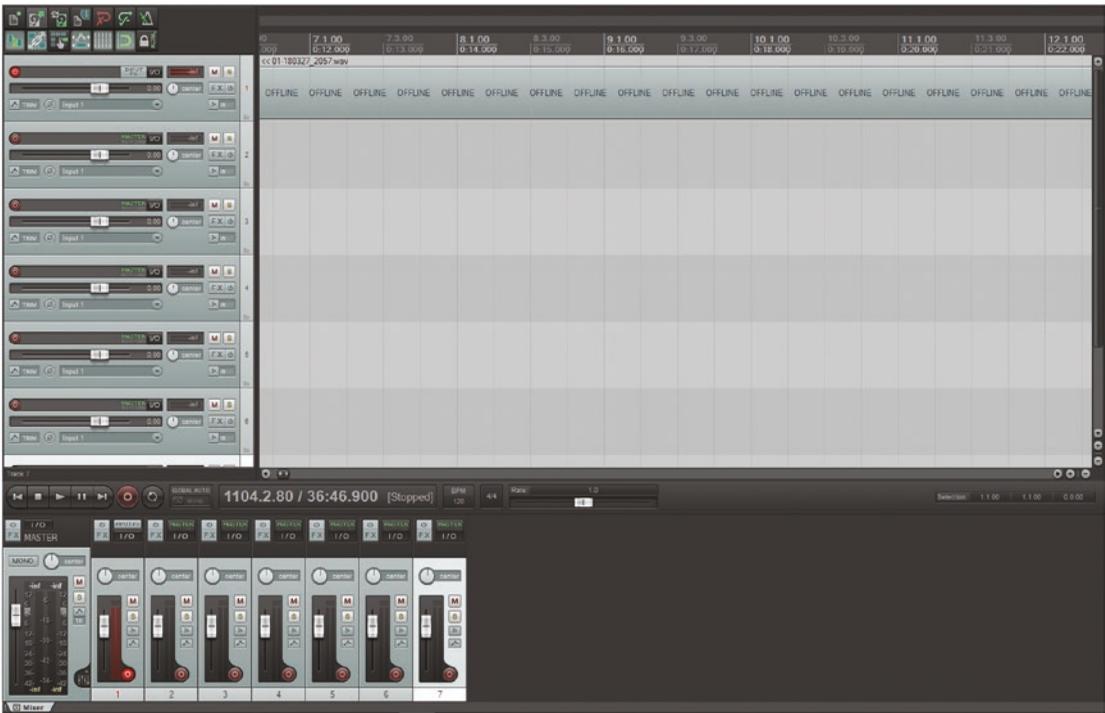


Figure 6-3. An example project open in Cockos Reaper

At first glance, Reaper seems like a textbook example of information overload. There are literally *hundreds* of potential actions able to be performed from this single state of the interface. To many, this is likely an incoherent mess of non-standard interface elements, yet professional music producers and audio engineers use software like this for hours on end to create, produce, mix, and master a huge percentage of the music we listen to every day.

Reaper, Photoshop and dozens of other apps that we might broadly define as “creativity tools” share a few traits that, were we to apply a reduction-first UX gaze, could convince us they were overbearing, or at least in need of simplification. While there’s no denying that many of these tools *could* benefit from a solid action audit and some ruthless simplification, part of their nature as creative tools means they’re playing by slightly different rules. What we initially see as complexity and a questionable density of features and information can alternatively be seen as an environment that refuses to be overly constrained or prescriptive in the name of encouraging creativity and self-expression.

Nonlinear applications often require at least some degree of learning and will almost definitely require moments of prolonged focus and deeper levels of processing. The positive side to this lies in how they allow for displays of mastery, autonomy, and self-expression. Many of the concepts explored in Chapter 3 apply directly to our nonlinear application states. We must allow for—and forgive—mistakes, focus on exploration, and teach our difficult concepts well. For nonlinear states in our applications, we lack the nice, sequential snugness of user journeys and simple flow diagrams that often work perfectly for documenting our linear flows. Attempting to document a nonlinear process can actually be a huge pain—user journeys become almost impossible in their traditional format, and flow diagrams become a tangled web of circling back and incomprehensible loops. One of my main goals for this chapter was to present a groundbreaking new method of early-stage documentation that would translate perfectly to open, nonlinear environments. On that point, I completely failed. Nonlinear environments, by their nature, don't fit nicely into diagrams or documentation. They're a container for elements and interactions that consistently change based on state and input. As we discussed earlier, they're much more like the worlds and levels of video games. Without turning this chapter into a deep dive on video game design, a concept that I think directly applies to our work is that of creating an environment that has overarching rules and purposefully arranging objects within it.

In video games, rules are often a mixture of global physics and internal, codified ideas. Much of the *feel* of a video game lies in its use of the rules of physics, such as gravity and friction. Video game designers can essentially place elements of varying mass into their worlds and have them governed by these global rules. Gravity, for example, might dictate how high and fast a character can jump (Mario's trademark jump went through numerous iterations, all based around the peak height he reaches and the speed at which he rises and lands), how fast they move, how instantly they stop, and so on. The purpose of these rules in a video game is not necessarily to create a highly realistic world—although, for many games, that might be integral—but to provide an engaging and enjoyable experience.

We can use a similar, albeit to a much less integral degree, approach when designing how our interactive elements behave—are our buttons heavy and stiff, or squidgy and bouncy?—and how they animate—do we have a high-gravity, low-friction environment? Is this one element “heavier” than another? Just like games designers, our environments don't have to mimic the physical qualities of the real world (we tried that, it didn't go

well), but having a consistent and predictable rule set can make the difference between an environment that feels united and one that feels like a messy interlinking of discrete elements.

Finally, as we explored in Chapter 2, video games designers are experts at subtly communicating information that we codify within these environments. Horizon: Zero Dawn’s use of yellow as a signifier for climbable elements provides the perfect example for this. Don’t underestimate the power of allowing people to discover new ways of executing complex tasks (for example, through keyboard shortcuts or gestures) on their journey to mastery.

A rather interesting example of this is the text editor Vim. Vim (Figure 6-4) is an astoundingly idiosyncratic text editor, mostly famous for no one ever knowing how to exit it.

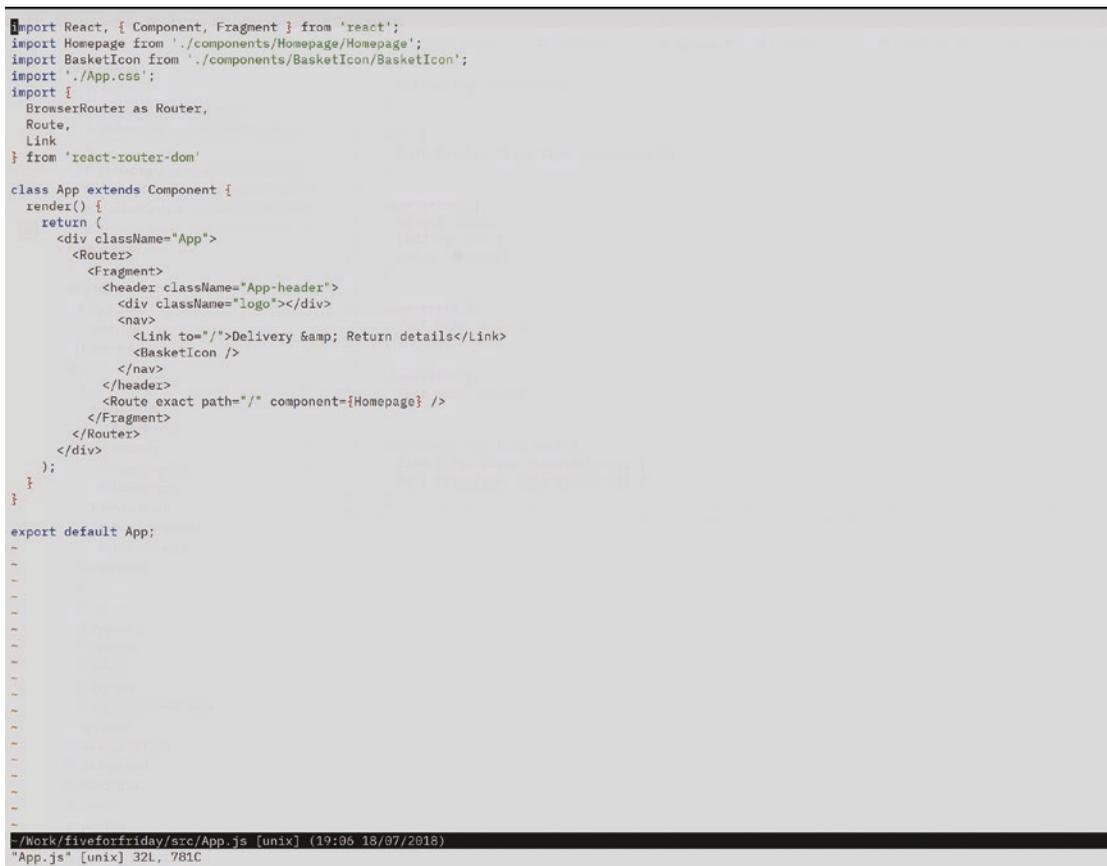


Figure 6-4. Vim in action

To the unsuspecting first-time user of Vim, it's a complete enigma to figure out. There's no mouse control, highlighting text is a chore, it has "modes" that completely change the function of most keys on the keyboard, and it relies heavily on people learning its own, specific way of operating. On the surface and through the gaze of ruthless simplification, Vim is a decades-old, archaic tool that should be long dead. Yet, thousands of people swear by Vim and its unabashed weirdness. A text editor that has hundreds of tutorial playlists on YouTube should not be loved by so many humans, yet it's what Vim offers in return for learning its ways that make it so tantalizing to use.

First, Vim is set up to be 100-percent keyboard usable. While there are plug-ins that enable mouse interaction, the "pure" way of using Vim is by keeping your fingers over the keyboard at all times. The speed, so people say, of using the keyboard to navigate and select blocks of text far outweighs the initial time taken to get used to such an idiosyncratic way. Vim's focus on keyboard control doesn't end there. The H, J, K, and L keys replace the Left, Down, Up, and Right arrow keys. The argument here is that the "home row" (the middle row of the keyboard) is where most advanced typists want to rest their fingers for the majority of their usage and reaching those centimeters to the arrow keys is far too imposing a task.

The core reason behind Vim's widespread usage, in spite of its inaccessibility and idiosyncrasies, is that it allows for an extremely efficient writing or coding process for those who have taken the time to learn it. By being hyper-focused on efficiency, to the point where around a centimeter of space is seen as too far to move one's fingers, Vim has sacrificed simplicity and linearity to focus on an end goal of mastery.

One of the most intriguing aspects of Vim is just how many different ways there are to use it. Although its initial appearance is hardly delectable to the visual palate and it remains a notoriously difficult editor to come to grips with, it actually has *many* more layers of complexity beyond its basic usage. Through combinations of keystrokes, well-practiced Vim users can select individual words, sentences, lines, and paragraphs in a fraction of the time it would take to use a mouse or trackpad to highlight them. Similarly, one can very easily perform complex tasks such as toggling the capitalization of various words or letters or changing the text inside quotes, parentheses, or brackets in a similarly speedy combination of keystrokes. Watching someone experienced with Vim fly about their text or code at ridiculous speeds is itself a lesson in the untapped power of complexity.

While Vim looks bare and barren, it's actually an extraordinarily powerful editor and has survived for this long not through consistent innovation, but through simply being

the perfect tool of choice for people who *really* care about the speed at which they can code and write. Compared to Reaper, which is as visually intimidating as it is in concept, Vim relies almost purely on a willingness to learn and embrace its weirdness.

The key thing, I believe, we as designers can take from Vim is that multiple methods of execution, especially of complex tasks, can make an environment feel incredibly enjoyable to work in. I believe this boils down to mastery. While it's extremely common to hear things like “our audience isn't experts,” “that's more of a power user thing,” and “we're designing for the mainstream,” I believe that perceived mastery, *especially* in creative applications that are used for hours daily, is a fine goal to work toward.

Trello is another fantastic example of this. While its card-based, drag-and-drop interface (Figure 6-5) is clear, succinct, and intuitive, there's a great deal of complexity hidden in its keyboard shortcuts.

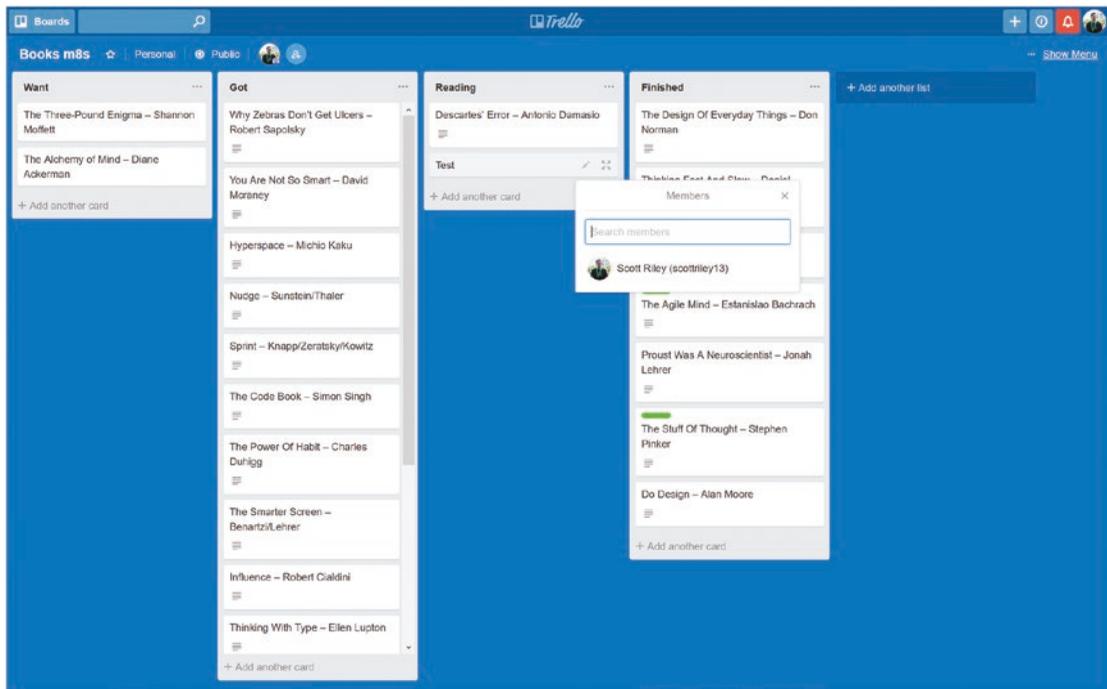


Figure 6-5. Trello's interface is clear and precise, with a host of complexity living in its keyboard shortcuts

Hover over, or keyboard focus, a card in Trello and press a key and you're able to, for example, quickly assign a member to a card (with the M key), change the due date (D), archive it (C), or move it around and between lists (',', '<', and '>'). While Trello is fully usable as a "traditional" point, click, and type web app, watching someone extremely comfortable with its shortcuts and specifics whiz around a complicated board is a genuinely insightful experience.

Vim and Trello offer a different example of nonlinearity—multiple methods of interaction. Both applications have their own learning curve, but they allow us to explore their environments in different ways. By offering multiple paths to the *same* result, Vim and Trello—in their own ways—create a sense of openness and self-expression that simply doesn't exist in the constrained, linear flows we discussed previously.

By thinking beyond the obvious visuals and offering multiple paths to the same solution, encouraging exploration and learning, and consistently providing timely feedback, we're able to create environments that are open enough for self-expression and perceived mastery, but not so opaque or idiosyncratic that they're too dense to be usable.

Balancing the Two

Balancing linearity ultimately comes down to how much we might limit control at any given point. A limited, constrained path has the advantages of allowing for an explicit showing of progress and a clear indication of intent, and it is relatively simple to document and design around. However, constantly seeing an interface as a sequence of interlinked, extremely linear flows can quickly lead to uninteresting, unmotivating experiences. Conversely, nonlinear interfaces allow for much greater self-expression, allow people to problem-solve within the constraints of the environment, and can play directly into the notion of self-determination and intrinsic motivation. Conversely, nonlinear interfaces require much deeper information processing, take time to master, and can often appear daunting at first glance.

One of the first considerations we should make when starting a design is the level of linearity that makes the most sense for our various application states. Many times, especially for mainstream or single-purpose apps, we'll be heavily operating on the constrained side of the spectrum, but that decision should be informed by correct research and testing. If we don't understand our audience, it's impossible to deduce just how—and when—our interfaces should expand and contract. Once we *do* have a solid idea of what people expect from our work, we can look into crafting our moments and designing the various states that make them up.

Design Your Moments

This approach to design might seem daunting at first. It essentially revolves around feeling comfortable ditching the tried and tested documentation of journeys and flows and embracing the chaos of “just throw some interactive elements into a room and let them do their thing.” Interfaces, though, aren’t video games. They’re rarely, if ever, autotelic, and there will always be a need for linearity and nudging, even in the most abstract of interfaces.

The key to this approach of design is to learn and practice when it makes sense to think in terms of global rules and properties and when it makes sense to zoom in and focus on linear flows. Essentially, this boils down to designing *moments*.

A moment in this context is a point in our environment that holds a level of importance, usually resulting in some sort of contraction or expansion of linearity. In the Amazon example earlier in this chapter, we might list the following moments:

- Seeing your search results
- Spotting an item you like
- Adding that item to your basket
- Checking out and paying
- Seeing payment confirmation

Now, this might look extraordinarily like a user flow, but the important consideration is to, wherever possible, treat these moments not as linear, interdependent actions, but as discrete events that can potentially occur at any given time. “Spotting an item you like” doesn’t necessarily have to come after seeing your search results, and people don’t always go from adding an item to their basket to checking out. Consider, too, how stored baskets allow someone to leave with a full basket and return at a much later time with all their items saved and still intact. I’m willing to bet that an early user journey in at least one of your recent projects contained such moments that were unnecessarily forced into a linear, interdependent flow diagram. That’s not me being all high and mighty, either. This is the most common “mistake” I continue to make. The temptation to apply order to, and demand control over, this part of the design process is a natural and healthy instinct, but that doesn’t mean it’s never problematic.

By focusing on these key moments and letting go of any need to fully control *when* they might happen, we start to build up an idea of our key interactions and the elements

that might make them up. We can also start thinking about how the various states of our application might shift to allow for, or respond to, these events.

Moments like this represent the key events that our environments should allow to occur. It's highly likely that we'll be able to break these down further, either into smaller "sub-moments" or into their own constituent linear flows. Remember that design is a *progressive* discipline. The media we produce takes place both over time and in response to interactions. In a movie, these moments might be our key plot twists or story scenes; in a musical composition they might be conceptually relevant parts of an opera; and in a video game, they might be boss encounters or conversations with important characters in the world. Because of the similarities between other forms of progressive media, I like to think of combinations of these moments as "acts."

Essentially, an "act" represents two interlinked concepts: the moments that can occur during that act and, subsequently, the degree of linearity presented during this act. This might sound like a somewhat convoluted way of thinking, but it essentially boils down to understanding and documenting your key interactions and, just as importantly, when they might occur. Going back to video games again, From Software's approach to world design in *Bloodborne* is a huge inspiration for this way of thinking.

Bloodborne revolves around traversing various, usually terrifying areas of a larger world, opening up shortcuts and hidden passages, defeating a host of monstrous enemies, and eventually finding and defeating an area's main boss. And dying. Lots. While many games like this adopt a very linear approach by dictating the order in which you encounter bosses (an adage of the games design world is that a boss should act like a "test" for the skills you've picked up and developed in the preceding areas) to try and control and dictate your path through the game, *Bloodborne* eschews this common approach in many stages, allowing the player to pick and choose (or stumble upon, in my case) the order in which they face the available bosses at a given time—even going so far as to making many bosses completely avoidable, should the player wish.

By choosing the best times to open the world up and provide the player with freedom and control or to constrain their options and nudge them along a more-linear path, *Bloodborne* provides a perfect example of this kind of "rhythm of linearity" that I feel is integral to this approach to design. *Bloodborne*'s acts revolve around how open the game is to this kind of exploration. It starts relatively linear, forcing you to make your way past one unskippable boss encounter, with an optional boss presented early on should the player wish to test themselves. After that, it opens up greatly, and the game doesn't progress until a number of bosses are defeated, yet it leaves the order in the player's

control. This trade-off between controlled linearity and chaotic openness continues as a feature of Bloodborne throughout the game.

While we're admittedly not designing games, FromSoftware's approach to Bloodborne is rooted in mastering the balance between control and exploration, and it provides us with an ideal case study in such. FromSoftware understands that there's a give-and-take to gaming—that there are times when the gamer must sacrifice control for a storyline or focused learning, and there are times when the developer must sacrifice their control of the player's destiny and how their game is played—and they transition between various levels of linearity expertly. We, too, can learn to integrate this approach into our work. By understanding that our key moments might happen at any given time, we can start thinking in terms of limiting options. *What moments should the current state allow?* is a lot more open-ended than *What happens next?* Often that open-mindedness is exactly what we need.

Design as a Function of State

The final ingredient to this approach to design is borrowed directly from a front-end development mindset. If you've delved into the world of JavaScript frameworks, it's quite likely you've encountered Facebook's React. Part of what popularized React is the oft-repeated idea that UI is a "function of state." This might sound somewhat obtuse, but the general idea is something that completely changed my outlook on design.

Applied to design, this concept mandates that an interface, at any point in time, is a product of all the underlying variables of its system. If you're wondering why throughout this book I've referenced "application states" and "interface states," this should hopefully clear things up. "State," here, is essentially a snapshot of all these underlying variables. We've explored (and will continue to do so) the idea that design is a conceptual model of a system that is then used to inform the mental models of those who perceive it. Communicating state is an integral component of these models.

Outside of its rules and concepts, an environment can simply be seen as a representation of a system's current state, presenting all the variables and interaction possibilities at any given time. When a certain element within an environment is interacted with, we might induce a change in the system's state, and the environment must shift accordingly.

As an example, let's look at a small subset of the state variables we might have for a simple project management application:

- Current logged-in user (if any)
- List of projects
- Currently selected project
- Project tasks
- Currently selected task
- Can I create a new task?
- Is a new task being created?
- Can I create a new project?
- Is a new project being created?
- Can I edit a task?
- Am I currently editing a task?
- Am I moving a task?

The premise of treating our interfaces as functions of their system's underlying state relies on assuming that any one of these variables can, and will, change in many different ways throughout any one single session with our application. Interaction Design relies on the premise of modulating state. An environment exists in one state, and through action and response, modulates to another. By deciding how our interfaces should represent various components to reflect the underlying state, we're able to craft an environment that feels consistent and compelling.

When looking at our designs in this way, our environments have two main concerns: how should the current state affect what is communicated and how it is presented? And how should the actions that perform changes in state be presented and communicated? Again, while this mindset might be new to you, the concept is rooted in tradition UX design. Instead of creating single screens as part of larger flows, though, we'll be creating discrete components and component groups that respond to and/or control various parts of our underlying state. The easiest way I can describe this is by considering the state as a central *conductor* for our interface. Rather than thinking in terms of step-by-step flows, think in terms of what any action might communicate to this central state and then how every component you feel necessary to include in each moment of state might respond to reflect that.

An extremely basic example of this would be a web form in an errored state. Our state might have the following variables:

- Name text
- E-mail text
- Submitting state
- Errors
 - What field is errored
 - An error message for that field

And our form might respond to the errors variable by:

- Showing or hiding an error message next to the fields that are invalid
- Disabling the submit button

In this case—let’s say our e-mail text isn’t valid—any change to the e-mail text should be communicated to our central state. The state may then analyze if this is valid or not, and update itself to add or clear any errors. Everything else on this screen should then update itself accordingly. For example, our button “knows” that if the form state is errored, it should be disabled. Our input field “knows” that if the error applies to it, it should show with a red border and with error message close to the input.

This practice essentially has us going back-and-forth between the low-level notion of state and the high-level notion of component-specific reflection of state. While, at first, this might seem strange, dizzying, or counterproductive, over time, thinking in terms of systems and state is one of the key tenets of this approach to designing environments. As we’ve discussed already, if design acts as the conceptual model of a system, a huge part of its role can be boiled down to accurately representing the current state of the system.

Of course, we’re rarely working on interfaces that are just a single web form. performing an audit of *all* the underlying variables and *all* the discrete states your system might exist in can be a huge undertaking, but if we take this approach from the start, our work will grow alongside our understanding of our system’s underlying concepts and data.

When we look at much more complex environments, state becomes a lot more integral. It not only dictates how all the components in the interface should be presented, but it also has a great impact on the potential actions we allow at various manifestations of state. If our project-management application from earlier were being described as manifestations of state, a core part of that would be determining and

communicating the *allowed* actions based on these manifestations. So if someone is creating a new project, for example, we'll likely want to contract into a much more linear flow than if someone was browsing a complex board of tasks. The entire "add project" flow could be extremely linear, taking the form of a kind of wizard, or even as the onboarding process for the entire application itself—but the key consideration is that we've been shown clear intent. Someone has clicked a New Project button (or used a keyboard shortcut, gesture, voice command, or whatever other alternative input methods we might be catering for) and clearly communicated their intentions. Were we to allow for tons of open exploration during the new project moment, we'd likely just end up detracting from the clearly communicated need for a focused flow.

So, our "is creating a new project?" variable from our state is true. Our environment then contracts into a linear flow, greatly reducing the potential actions and thus removing a lot of control away from the project creator. This is still a function of state! The actions that we allow people to perform are informed by the underlying state of the system at any one time. And this is where the idea of balancing linearity comes to the forefront. Just as our visual design—our colors, signifiers, typography, grouping, and animations—is dependent on state, so too is the openness of our environment. By constantly interrogating the data-level building blocks of our system's state, we can make intelligent decisions around how and when we expand and contract our environments, using state as a representation of our system *and* as a heuristic for intent.

This approach, fortunately for us, also doesn't necessarily require the explicit production of wireframes and prototypes for *every* permutation of state. While the key moments we discussed earlier in this chapter will likely require focused attention and detailed mockups, often it's *just fine* to document how state affects your individual components and just *let it happen*. This, again, is pretty difficult to swallow at first. Rescinding control of all the minutiae of our applications in order to present designs that are responsive and malleable based on the underlying state can feel like giving up part of what makes us designers. But, we absolutely don't *stop* controlling things altogether, and there will always be a need for linear wireframes and flows. It just so happens that these form only *part* of our work, and we need to accede control here and there, usually in the form of opening up our environments and allowing people to express themselves within them, to not be ushered through a linear flow by the digital equivalent of a pushy salesperson.

Take a creativity/design tool like Photoshop as an example. There are *millions* of potential discrete combinations of state variables in Photoshop's underlying system, and it would be impossible to design linear flows for every one of them. However, Photoshop,

as we've discussed, is an incredibly open and nonlinear environment. It exists to facilitate creativity, and even its most-constricted moments—such as its lovely in-app guidance we dissected in Chapter 3—still allow for almost untethered exploration. Instead, Photoshop's interface adapts to state inconsistent and learnable ways. Take tool selection as a basic example. The important state variable to us is “currently selected tool,” and the interface's discrete components update to accommodate that. Figure 6-6 shows how Photoshop's interface changes to reflect its underlying state.

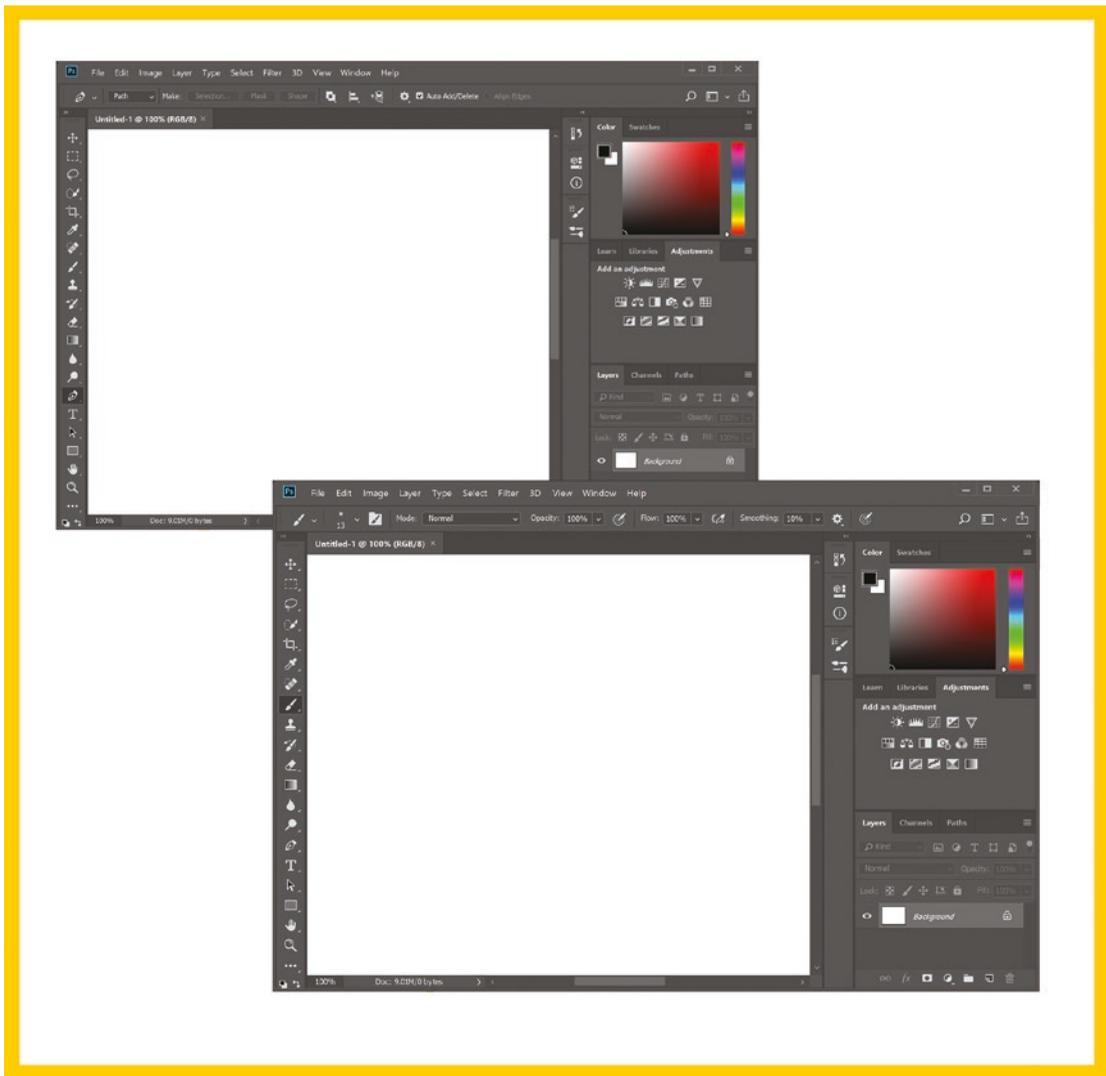


Figure 6-6. Photoshop's interface when the pen (left) and brush (right) tools are selected

When switching from the pen tool to the brush tool (both of which can be selected either by clicking or by using a keyboard shortcut), the Photoshop interface shifts in ways that let us further control the tool we've selected. When the pen tool is selected, the following changes occur:

- The pen tool icon in the left toolbar is styled differently to denote it as the “active” or selected tool.
- The top toolbar, where most tool-specific settings are changed, allows us to determine whether we're drawing paths or shapes. The settings perform various path operations, including adjusting path alignment and arrangement.
- The cursor, when we hover over the canvas, changes to a pen nib-like icon.

When we select the brush icon, the toolbar and cursor update visually to let us know we've selected a new tool, and the top toolbar is full of settings that affect the style and performance of the brush we've selected. This approach is far removed from the usual UX processes we may be used to when creating web and mobile applications. It leans heavily into the notion of letting state and intent shape the environment, rather than prescribing set paths, flows, and journeys for every possible permutation of state. This is the key difference between designing linear flows and designing open environments—one is a representation of user intent, the other is a representation of the intent of the designer, or the system, or the business.

If creating global rules, designing individual, self-contained components that communicate with and reflect underlying state, and rescinding control over the minutiae in order to focus on these sounds familiar, it's because it essentially forms the basis of a *Design System*. Design Systems are a relatively new introduction to our design lexicon at the time of writing, but they're fast becoming a standardized, well-documented practice for documenting large, interaction-rich interfaces. We'll discuss Design Systems and their role in the design stack in detail throughout the next chapters of this book because they act as the perfect foil for documenting the more open, nonlinear aspects of our work.

Summary

This is a highly theoretical chapter, and many of the concepts we've explored might seem at least a little bit fuzzy and perhaps even impractical. I want to stress that you absolutely don't need to use words such as "environment design," "acts and moments," or any other obtuse terminology you may have turned your nose up at throughout this chapter. These concepts are simply metaphors that attempt to summarize an approach and a mindset that I feel is often lost when we strive for control.

The key facets of this approach boil down to understanding and reflecting the underlying state of an application at any given time, designing how your interface will expand and contract based on state (including any currently performable actions), and embracing the letting go of control in the name of presenting open, explorable, and progressive compositions.

The remaining chapters will inspect and analyze the traditional design practices where I feel this approach can be best implemented, without stepping on the toes (too much, at least) of accepted practice and industry standards. We'll first explore the broad areas of the modern design stack and look at the artifacts and processes of each area that I feel can most benefit from slight tweaks or changed mindsets. Then we'll take a look at a number of less common approaches to planning, documenting, reviewing, and testing designs specifically to focus on the theories and concepts discussed throughout this book.

I truly believe that the best possible design decisions can be made by using the understandings from part one to inform our environments, specifically around how they respond to and reflect state, changes in state, and communicating conceptual models of an underlying system. I hope by the end of this book you'll agree.

CHAPTER 7

The Design Stack

After a highly theoretical chapter, let's jump right into some practical discussion. Chapters 8 and 9 will explore the early-stage design process from two different angles with the end goal of providing a unified, practical approach to incorporating many of the theories and practices we've discussed so far in this book. In Chapter 9, we'll take a look at several underused and new concepts that I feel we can weave into our design process. However, in this chapter, I'd like to explore the practices—and their resulting artifacts—that are, first, well set up to incorporate these findings with almost no disruption and, second, already part of myriad established design processes.

I'd like to stress that this chapter will potentially include tasks and activities that you currently don't do and will almost definitely not include everything in your current process. There are so many design disciplines, as we've seen, and exponentially more ways of combining these disciplines into a single designer's role and skill set. Many designers specialize in a single discipline of the industry, such as "strict" UX, Visual, Motion, or Service Design—just as many of us are generalists in the field and blur the lines between many different disciplines. The goal of this chapter is to not be prescriptive or overwhelming, but to provide an accessible and streamlined reference as well as suggestions for tweaking them to accommodate the ideas presented throughout this book.

This chapter will be split into some broad categories of design tasks: research and planning, scribbling and sketching, and testing and iterating. Within each of these categories, we'll delve into a number of smaller tasks, such as customer research, wireframing, and user testing, and see where and how we might work in some of the theories and knowledge we've been building upon throughout this book. If you're new to the industry, this will hopefully be a good primer for dipping your toes into the various areas of the design process. If you're further along in your design career, feel free to skip to the sections that fit your role!

Research and Planning

The research and planning stages of a design process are where we set our foundations for our projects. Just like a house, if the foundations are weak, the end product will suffer. Of all the projects that I've worked on, the ones that have failed have done so almost exclusively because of poor research or planning by designing around the wrong problems, designing for the wrong audience, or simply just making far too many unverified assumptions.

These two stages are all about getting yourself clued in on your audience, your problem definitions and the underlying system you'll be building. Then you can define your potential solutions. I've grouped research and planning together here because they're almost inseparably cyclical in nature. The more we research people, problems, markets, cultures, and competitors, the more likely our plans are to change and adapt to accommodate them.

The depth and reach of your research will depend on a rather large list of variables, specifically your research budget, team size, any deadlines you may have, and the level of access you have to the people you're designing for and the tools they use. While *the dream* might often be to have a bottomless pit of money and as much time as you'll ever need to get the perfect amount of work done, in reality, money is always an issue, and time, at the risk of sounding clichéd, is money. Furthermore, even with this mythical "as much time as necessary" at our disposal, research is a task of diminishing returns. After a certain point of information gathering, dependent on the depth and ubiquity of the problems we're trying to define, we all hit a point where the value of the information we're gathering drops to the point where we've learned enough to jump in and start our project.

Before we delve into some of the common approaches to design research, I want to discuss *why* this stage is so important, especially in relation to the content of this book. When we embark on a research phase of a project, our goal is to collect, collate, and explain *understandings*. From the moment we start delving into what makes our audience tick, we're on a mission to build upon knowledge, insight, and data that will be constantly referred back to throughout the project. Research is what takes the amorphous, fuzzy blob that is the idea we have of the people who we're designing for and turns it into a sharpened, well-documented set of findings, examples, hypotheses, and problem definitions. Most importantly though, everything we do at this stage is bubbling with potential sources of empathy and compassion. This is our chance to truly learn about the humans for whom we're creating our products. To observe our audience,

in all their fallibility and idiosyncrasy, while our project is still comparatively free of bias and guesswork is the best chance we get to truly understand *why* the thing we're creating needs to exist.

If research is all about gathering this understanding, then planning is a statement as to how this research will be used. Design planning is a lot different from planning a product or a project. While the findings of design research form an integral part of product and project management, we're veering into "not-my-job" territory. It would be remiss of me to try to explain the intricacies of a product owner's, product manager's, or project manager's job on their behalf, and this book is far from a design strategy book. There's already more than enough information out there on the subject, and it is *so* dependent upon internal practices. For our purposes, design planning lies in problem definition and in distilling our research findings—especially the human qualities of such—into artifacts and assets that can be referenced throughout our project by colleagues of many different disciplines.

Finally, research and planning act as the first link in a chain of accountability and reasoning. Any time a decision is questioned or debated or when we're simply unsure of its importance to our core goals, our research findings and project plans become our point of reference to determine if this decision is true to our original intentions. This is critically important. As designers, we'll either be working directly with researchers or be conducting some, most, or *all* of this research ourselves. One of the best things we can do is instill this empathy and compassion into our project at the ground floor. Like our design work, our research findings and documentation have an audience: our clients, our colleagues, and the project's stakeholders. This is our best chance to ensure that the emotional, irascible, and human sides of the people we interview, observe, and analyze permeate our entire project.

With that in mind, I'd like to focus on just two of the industry standard areas of research and planning that I feel are most open to providing this platform: customer research and user personas.

Customer Research

This stage of the research phase is all about understanding the makeup of the people who might want to use our stuff. While this method of research often takes a quantitative form with many stoic reports being the only artifacts of such a phase, providing data and averages and charts is not enough. Getting to know people on an emotional

and motivational level is absolutely critical at this phase. While data is an essential part of presenting research findings, it'll never be more important than the intrinsic understanding we can get from delving into the emotional underpinnings of people's problems, frustrations, excitement, and motivation.

The intangibles of this phase should light a fire in your belly. *This* is where you observe the behaviors of the real people whose real lives you want to improve, where you witness their fallibility and frustrations, their routines, and their rituals. Every moment of this kind of research should be seen as a privileged insight into the minds of the humans on the other side of your products. Do not let that pass you by in the name of producing orderly, assumption-affirming research reports for your bosses, colleagues, or clients. In fact, if—for whatever reason—you have to choose between spending more time with people or documenting your findings, get out there and spend time with people. Regardless of how deep into a project you'll be involved, your understanding of the human aspects of the problem it intends to solve will be of much more important than an accessible and well-documented market analysis.

Customer research is one of the most expansive areas of design, and every design methodology and framework seems to favor a certain subset of research practices. I'd like to focus on just one area of customer research for this section, as I believe it's the single most important: observation research.

Observation Research

Design happens in context. And research is simply understanding that context.

Erika Hall, "Just Enough Research" (*A Book Apart*, 2013)

Observation research follows a similar process to user testing in that you present people with tasks or scenarios and watch on as they unfold. Similarly to user testing, observation research is often best conducted with as little interference from the designer as possible. One of the key things with observation research is getting out into "the field" and observing people in their everyday environments. This is commonly known as field observation. Far too often, interviews and testing sessions are conducted in manufactured environments that reveal little beyond the most obvious stumbling blocks and usability issues that people face. By getting into the places where work happens and where problems are encountered, you're not only observing individual people, but

you're also observing their environment. This is a critical consideration, especially in the early stages of your research. In fact, I'd go so far as to suggest that in-field customer observation is the first act of research you perform.

One of the key benefits to early observation is that, in the embryonic stages, we are as unaffected by project-internal bias and assumption as we'll ever be. Every interaction we have with a colleague who has a preconceived notion of the problem space and customer base within which you're operating is, at this point in a project, nothing more than a well-meaning distraction. You absolutely *will* want to quiz the experts of the domain you're designing within, and often your clients and your team members are the perfect people from whom to acquire that knowledge. However, at such an embryonic stage, I feel it's incredibly important to visit your potential audience and just *watch them*.

How It's Done

Observation research is an extremely broad methodology and often takes multiple forms, depending on which stage a project is at. Anything that involves a designer or researcher watching people *do stuff* can be classed as such, and almost any kind of observation can be beneficial. There are, though, a few things to consider to make sure our observations are as efficient and as usable as possible.

To best accommodate the diminishing returns of research, I'd suggest scheduling between two and four sessions of observation research, with the first being scheduled as close as possible to the project start date. This first foray into observation is your chance to soak in everything you can about the way your potential audience approaches the problems you're trying to solve for them. It's your chance to start compiling an initial list of the things that make your people tick, to see what frustrates them, what hacks they use to get around technical or environmental obstacles, what appears to excite them, and what you feel motivates them. Further sessions can become progressively more interactive, eventually becoming a hybrid between observation and interview.

These ideas are super fuzzy without an example project, so let's say we're working on a new tool that allows designers to animate their static design concepts. We want to get a grasp on how our potential customers currently go about this. Our *problem space*—the gamut of problems we might potentially wish to solve—is somewhere between a static design concept being completed and a final prototype being shared with a client or a colleague.

Our core goal with research is to understand the *people* who operate in this problem space, the environments in which they operate, the *distractions* they might face, and the *motivators* that keep them ticking along. In my opinion, the absolute best way to discover this is through going out and watching people operate within this problem space. I'll discuss how we can work around limitations shortly, but for now, let's assume that we're at least able to get out and visit the people we wish to learn about.

The first things we'll need are participants and consent. Many of us are horribly anxious and self-aware when we're being observed—unpermitted observation is simply an invasion of privacy. If we have plans to watch designers work, we have plenty of options. Social media is a big help, and many of my less-formal research projects have solely involved visiting people from Twitter and watching them work. A simple “Designers! I'd love to observe you for an hour or two as you work on a problem. Coffee's on me.” tweet can provide you with more than enough willing participants. Other options would be to visit local coworking spaces, or even the office of a friendly agency or startup, and ask for some willing volunteers. Almost all recruitment can be done online, though, and it's quite rare that you'll want or need to venture out and start soliciting subjects on the street. Finally, remember to pay people for their time. A common approach is to offer Amazon gift vouchers, but in the past I've had success offering free coffee, buying participants brunch, or simply offering cold, hard cash.

This stage is commonly known as *recruitment*. A large part of it consists of ensuring that you're working with a diverse range of participants while still making sure that they all work in your problem space and would make good subjects. (Filtering potential participants is known as *screening*.) Good recruitment and screening are integral to good observation. If you notice that your entire shortlist *just happens* to be white dudes in tech startups, then work on diversifying your network and your recruitment, or outsource your recruitment to professional researchers who can provide you with an expansive array of willing participants. A lack of diversity—of background, race, sexuality, gender identity, age, financial well-being, physical and mental health, and so much more—in your recruitment should set off the loudest, flashiest alarm bells in your head. The whole reason of investing in research, especially in recruitment, is to ensure that our understanding is wide reaching and expansive. If you're interviewing people that don't challenge your assumptions and your status quo, then you might as well just interview yourself and throw your research budget in the sea.

Once we have a diverse range of participants, have their permission to study them, and have made our way to our new friends' place of work—whether that's an office, a

coffee shop, a coworking space, or even their own home—we can prepare our subjects and our station. Here, we have a few options, with differing levels of practicality and personal space invasion. We’re going to want to record our session in some way—so let’s tackle that first.

One option—and this will massively depend on the environment—is to set up a camera on a tripod and have it record the general area where your subject will be working. This is more practical in offices and coworking spaces, where this kind of setup might be a little more commonplace than it is in a coffee shop, where you’re likely to just get in the way. Another option is to wear some kind of mounted camera yourself. I’ve recorded sessions with a GoPro and a chest strap in a pinch. While you’re likely to get some weird looks, it’s a lot less invasive and obvious than an impromptu film set. Finally, you can ask the participant themselves to start a screen-recording application before the session starts and use that as your reference. Personally, I think the ideal scenario is a combination of all three: a camera with a wide-angle lens set up to capture most of the environment, a smartphone or smaller camera on your person, and screen-recording software running on the subject’s machine (ask them kindly to install this beforehand).

It’s important to note that the more you “tamper” with the environment in this way, the more you dilute the experience. If you feel you’ve impacted your subjects’ environment too much, scale things back. The more aware someone is of their surroundings, the less likely they are to act like themselves. You may find that when observing people who are hyperaware of the fact, they’ll—consciously or otherwise—embellish their process or practices. Alas, this embellishment is unavoidable, and all we can do is limit our interference to the point where it’s at an acceptable level. If we inspect our own creative process, I’m sure there’s a whole host of things we do that we’d like to very much keep to ourselves! Use your intuition and best judgment. Some people are simply more confident and more open than others, but there’s absolutely no one-size-fits-all approach to setting up your recording methods.

In a case where you feel like you would compromise the environment or process too much even with the subtlest of recording setups, I believe it’s much better to go without and document by other means, such as taking detailed notes. You don’t want to spoil your data before you even start.

Once we have our means of recording set up, we need to position ourselves. This can be trickier than it seems because we’ll need to be close enough to our subjects to see the broad stroke details of their actions, but far enough away to not impact their work and to observe enough of the external environment. There are no hard and fast guidelines for

this, so you'll have to use your intuition again. Err on the side of caution. If you have to choose between being a distraction or potentially missing the action, get into a position where you're passive and out of the way. Seeing the minutiae of the actions being performed is almost meaningless if you're impeding your participants. Between your own positioning and your recording set up, you should be able to see the following (with my preferred method of recording/viewing in parentheses):

- The participants' screen(s) (screen-recording software)
- The participants' input devices (camera, your eyes)
- The immediate environment (camera, your eyes)
- The whole environment (wide-angle camera)

At a minimum, we should be recording participants' screens and be able to see the immediate environment. This means we can use screen recordings as a documentary of the entire session and take notes about the other aspects we observe. In many cases, this is all it takes to get a solid observation session underway. Once we're all set up and positioned, we observe.

When we observe a session, we're looking for answers to a pretty extensive list of questions:

- What environmental distractions slow progress?
- Can you spot what frustrates or hampers?
- Are there any points where participants seem particularly enthused?
- What on-device distractions interrupt their flow?

Let's say for our first testing session, we're in a small, open-plan office and our subject is able to work away without an overbearing awareness of our prying eyes. During this session, we'll want to look out for the four broad categories mentioned earlier. I like to prepare a notebook with plenty of pages split into quadrants (Figure 7-1), with space below for general notes before I start this process.

The figure shows a template for documenting an observation session. It consists of a large yellow rectangular border. Inside this border, there is a white area divided into four quadrants by a vertical and a horizontal line. The top-left quadrant is labeled *Distraction*, the top-right is *Frustration*, the bottom-left is *Flow*, and the bottom-right is *Enjoyment*. Below the bottom-left quadrant, there is a horizontal line followed by the text *General notes...*

Figure 7-1. Preparing to document our observation session

If we're looking out for dozens of things at once, the last thing we need is to have to bring out the pencil and ruler, so make sure you've got ample room to take notes. Any time you spot something that might fit into these categories, jot it down into one the correct quadrants. Then start looking out for the next.

Let's say, in our first observation stage, we manage to interview five designers of varying expertise, working environments, and backgrounds. Perhaps our only "designer-at-a-busy-agency" participant gets frustrated at the lack of time they have to complete their tasks, is mostly distracted by Slack DMs from their project managers, and feels motivated by positive client feedback. Maybe our two "interaction-designers-collaborating-in-an-open-plan-startup-office" are motivated by being able to quickly show each other their ideas, are distracted by the constant noise and chatter in their office, and are frustrated by interruptions from the patrolling, over-eager CEO. Already, with such a tiny subset of notes, we're able to start seeing the problem space from a number of perspectives. Perhaps our tool should offer real-time collaboration or an extremely simple way of sharing prototypes with colleagues and clients. We also know what distracts and frustrates our participants. Let's say our tool is going to be web-based. We're now competing with team chat tools, the allure of social media, the patrolling CEO, nagging colleagues, and the bustle of an open-plan office.

These findings, a hypothetical subset from two hypothetical observation sessions, might already start shaping your product feature set and will *definitely* start sculpting your understanding of the problem space. Even if you have a broad idea of the problem you're trying to solve, say, "Designers struggle to take their static screen designs into web-friendly, animated prototypes," you *still* need to explore this space. The act of putting these prototypes together as well as the gaps in the market is extraordinarily broad problems that, frankly, most people have the vision to see and define. These observation stages allow you to come to grips with the reality of the problem space, spotting the *sub-problems*—the tiny frustrations, the daily distractions, and the limitations of tools—that provide depth and diversity to your solutions. *This* is where problem definition starts taking shape, where so-called blue-sky thinking becomes actionable, and informative findings become possible. Our core offering (an animation and prototyping tool) might not change much, but the features, qualities, and mental models that make it up will be greatly informed by the work done in this phase.

I mentioned earlier that I believe it's best to have between two and four stages of observation research. While I believe the first stage should be as untainted by bias and other research areas as possible, I think it's essential to conduct at least one more stage of field observation toward the end of your research phase. In these latter stages,

increase your interaction with your participants—even consider turning it into a combination of observation and interview—and try to figure out what makes them tick. In these latter stages, when you understand the problem space and your audience much more, you can start trying to answer some of the countless questions you’ve amassed throughout the research phase.

Compassionate Interview Tips

Interviewing customers, either as part of a hybrid field observation and interview approach or as its own discrete task, is an art form. Our goal is to encourage participants to talk about their experiences and recall moments when they felt certain ways about their work, their tools, or their life. So, rather than asking someone, “What do you like about your current prototyping tool?,” consider asking them to tell you about the last time their current tool made them feel like they were a superhero or about the last time they got frustrated with their current prototyping tool.

The first question, “What do you like...” will likely just get you a big list of features, and, if the tool’s marketing team has done a good job, you’ll probably find most of these on the company’s web site anyway. By asking someone to recall and explain *moments* in their life, you’re going to get a more holistic recollection of their experience. Encourage your participants to become storytellers. Listening to them as they regale you with tales of their triumphs and failures is a humbling and humanizing experience. Just like with field observation, try to stay out of the way as possible, only intervening when the participant appears to need a nudge in the right direction: “Oh, tell me more about how broken link-copying code nearly resulted in your boss getting a swearsy Winnie-the-Pooh GIF instead of the link to your prototype” (which may or may not have happened to me. Twice.). You’re getting all the information you need—what frustrated them, what feature in particular allowed this, and how they reacted to the situation—but in a way that has a valuable emotional context.

Keep in mind that when you’re trying to eke out details of someone’s frustrations or negative experiences with a process or a product, these could be traumatic and uncomfortable memories. An interview is an invitation into someone’s personal headspace, and as guests there, we are entitled to nothing. If you notice that your line of questioning makes a participant uncomfortable in any perceivable way, apologize, offer to take a break, or even end the discussion. Interpersonal skills and emotional intelligence are prerequisites of an acceptable research skill set. If you struggle to recognize when others are uncomfortable, you should be wary about entering their

personal space altogether. Freelance researchers and agencies exist for this reason. If your skills lie outside the realm of conducting and directing compassionate face-to-face conversations, outsource this work to someone else and observe the interview, or watch the recording afterward. Again, when we conduct interviews like this, we are *not* entitled to make anyone feel uncomfortable, to press them for details when they appear unwilling, or to take their emotions and mental health for granted.

Following are some useful prompts to get the conversation flowing:

- Tell me about a time that *X* made you feel like a superhero.
- Tell me about your favorite moment with *X*.
- Tell me about a time that *X* frustrated you.
- Tell me about a time where you used *X* to do something it wasn't intended for.

These starting points will give you insight into the areas we often most care about: what the tools in our problem space do well, what they do badly, and how people wrangle limited technology to do things that might not have been considered by its designers (this, incidentally, is a byproduct of rescinding control and creating explorable environments). This information, combined with your observation notes and recordings, can quite easily form the backbone of your entire project.

With a diverse range of participants and environments, we can quickly build up an extensive library of research and documentation that can be used to inform the subsequent stages of our projects. I believe that this approach to research has the biggest return on investment of any of the popular methods. In addition to customer interviews and competitor research, it's quite possible to amass an enthralling and diverse range of research materials in a very short space of time.

Competitor Research

While many research practices treat customer and competitor research as two very different disciplines (for practices like market research, this is essential), for design research, they're often both results of the same kind of approach. If we take into consideration that a huge part of design research consists of understanding the problem space, then the products and services that attempt to solve problems within that space are an integral part of it. When defining our problems, it's important to explore these potential solutions and get a read on our competitors.

I won't dwell too much on competitor research, but I will say that you'll likely be conducting it if you're observing and interviewing people. Make a conscious effort to find participants that use a wide range of your competitors' tools when you're recruiting for your field observation and interview participants.

For example, part of our screening and recruitment process for our prototyping app participants could be to ensure we have a range of people who use After Effects, Principle, Figma, InVision, Keynote, and Framer. This lets us double-up our observation research with competitor research.

An analysis of competitors can also reveal some helpful information on the mental models that customers might take when using products within the same genre and problem space. In the case of our prototyping and animation application, we might notice that all the animation apps our participants use make use of timelines for editing transitions. If we were making a video-editing application, we might notice that all our competitors use the razor blade icon to indicate a tool that splits clips.

One simple method of conducting competitor research as part of a design research phase is to actively user test the competitor's products. This might appear to be a strange suggestion, but if we consider the notion that user testing our own interfaces provides us with considerations as to how we can improve our own offerings, then doing so with our competitors is essentially a fast-tracked market analysis. If this is something you're interested in trying, the user testing section later in this chapter will be of particular interest!

Customer Service Is Your Friend

If you happen to be working on a new feature or product for an existing company, *make friends with customer service*. Quickly. Customer service should be your go-to any time you're struggling to pinpoint the core problems that people are facing. Get to know the company's customer service team, get access to their support system, and shadow them for a day or two. Get to know every possible customer-service employee on a first-name basis!

User Personas

I have, throughout this book, aimed a few subtle digs at user personas, for various reasons—all of which I feel are relevant, and most of which I will discuss. However, I must stress that they stand alone as the single artifact of the research and planning phase that can best present and communicate compassion and empathy throughout our project's lifecycle.

User personas are simple, digestible *archetypes* of human beings. In Chapter 4, we spoke about Prototype Theory—how we form underlying “feature sets” and rules that underpin a category and how the things we perceive in life can be more, or less, applicable members of that category (Rosch, 1988). User personas operate on a similar level; they act as statistical averages for human beings, and they’re often used to present very important stakeholders with a semblance of understanding of the people our products are going to impact and effect. A sample user persona is shown in Figure 7-2.

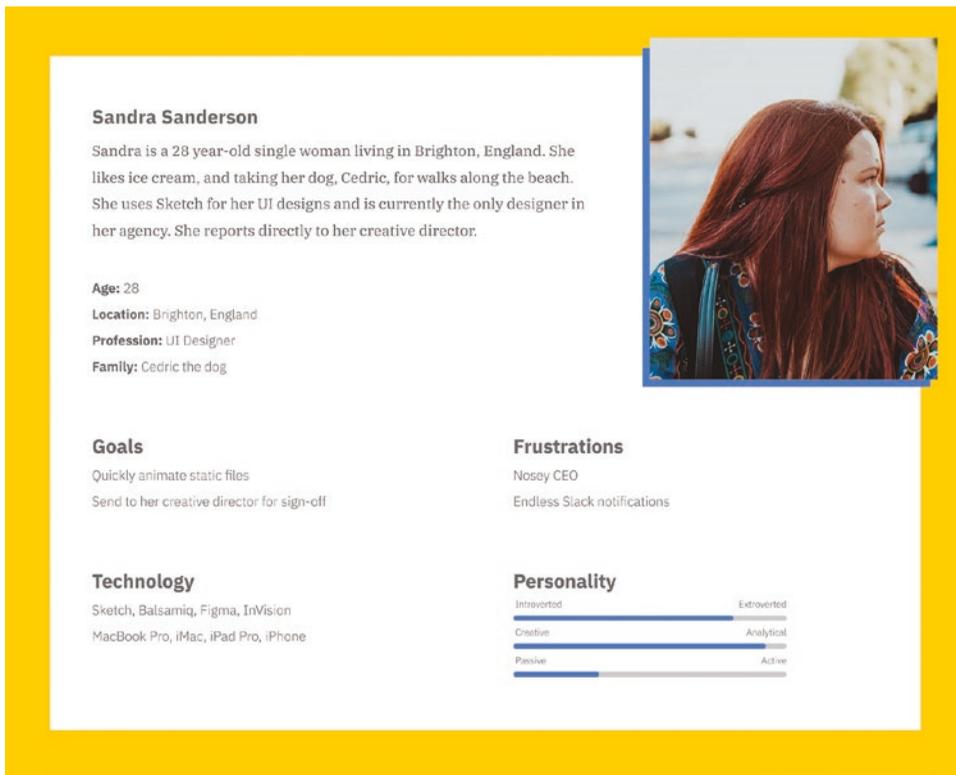


Figure 7-2. A common format for a user persona

Now, deep breaths because here are my problems with user personas. First and foremost, they are manufactured empathy, an abstraction of a human being often distilled into a banal and mundane list of life experiences, skills, abilities, and preferences. Human beings do not fit into such a box. All of us have idiosyncrasies, bad days, things we enjoy, secrets, desires, dreams, and ambitions. It’s impossible to cram all of this into an entire book, never mind the single PowerPoint slide format that most personas occupy.

Second, personas can suffer greatly from a cascade of bias and assumption, depending on how many people there are between the people who've been observed and the weird, digital protohuman that sits in your research slide deck. It's impossible, when creating these fantastical fictional characters, to avoid project-level, colleague-level, and personal-level biases and assumptions. So we often end up with personas that not only fail to represent even a fraction of the humanity that we're *really* designing for, but that are essentially pre-judged and pre-victimized by the unconscious and systemic biases that live within our teams and ourselves. The overwhelming majority of manifestations of user personas I have witnessed in my career have served both as placation for executives that we feel are too busy, too important, or too removed from the project to display true empathy and as a convenient dictionary of the biases of our teams and companies.

Once we've traded away people's frustrations, eccentricities, and self-expression for nicely formatted human-shaped pigeon holes, these artifacts often become the single point of reference throughout our work until we finally get some working prototypes and ideas in front of real people. This is a dangerous and insidious precedent in an industry that is already insufficient in the empathy department.

Phew.

Now, why then, if I dislike personas so much, am I treating them with such importance? Simply put, *they work*. Personas' ability to unify a team around a core set of principles and supposed human qualities are unrivaled in the world of design artifacts. We just need to make sure that our personas communicate the correct principles.

Here's my suggestion: every user persona should challenge an ingrained assumption or bias that you feel exists within your team. Maybe that assumption is that only able-bodied people will use your product. Or perhaps the assumption is that only affluent, white millennials care about your solution. Every time someone on your team makes an assumption, write it down and figure out how to break that assumption down in your persona work. Nominate someone on your team, preferably someone who will be involved in early-stage planning, to call you out on your own assumptions and vice versa. Personally, I've had the good fortune to work with amazing and compassionate product owners who make the perfect foil for BS-calling-out—they're usually involved from the very beginning, and the artifacts of your work massively impact their job.

You don't have to share with your team that you're doing this. In fact, doing so is probably the fastest way to put up barriers between you and your colleagues (being *that person* who is constantly taking notes about other people is *not* a great shtick) and

potentially cause some pretty tough “political” situations. My advice here is simple: the impact your work has on the people it is intended to help is more important than some friction and discomfort. The other side to this, of course, is that being taken seriously in the workplace is something that is often afforded disproportionately to the privileged among us, and this kind of assumption-challenging can often grate on the egos of our colleagues. If you are a design manager or if you work with a design manager you trust, this kind of approach is one that’s best initiated from a position of leadership.

When we treat personas as a means of challenging assumptions, we start to infuse a form of BS-detection into our early-stage documentation. If you work on a team that already uses “standard’ personas, you might have witnessed the power a well-placed “but how does this work for Steve the Persona?” possesses when diffusing a situation or settling a debate. If we strive to present a truly challenging and diverse set of personas, we provide ourselves with a safety net should our discussions lead away from compassionate design decisions and veer toward the distracting, the manipulative, or the just plain exploitative.

Consider the following “traditional” persona for an imaginary flight-booking application.

Sandra is a 28-year-old single woman living in Brighton, England. She likes ice cream and taking her dog, Cedric, for walks along the beach. She travels to her company’s head office in Seattle on a quarterly basis and rarely travels for leisure. She uses her smartphone to book her flights and is more concerned about getting things done quickly than in finding a bargain.

This all sounds pretty solid. We can actually start to imagine what Sandra’s preferred relationship with our application might be, understand her motivations, and probably draw out a nice, convenient use case for her time with the app. Sandra is, however, a product of our imagination. She is a work of fiction based on some form of statistical analysis and observation research. We’ve stripped this prototypical person down into the chunks that are convenient for us to work with. In many cases, Sandra will simply be a byproduct of our assumptions. If we assume that everyone who travels for work doesn’t care about bargains and would rarely use our app to book leisure flights, then we’ve created Sandra to appease these assumptions. Consider the alternative.

A is 24 years old and suffers from anxiety. They work from home and enjoy taking hot vacations in the winter. A relies on their phone’s calendar and schedule to provide an in-depth itinerary of their day. A struggles to cope with chaotic or unexpected circumstances. A is easily distracted by notifications and ads, and their anxiety is often triggered when their control over the particulars of an event is taken away.

This persona challenges a few assumptions that we might encounter when kicking off such a project: the people using our app will be neurotypical, something as simple as a travel booking app couldn't possibly trigger an anxious episode, and our app will be used in a serene/distraction-free environment. With these assumptions challenged, it forces us to accept the fallibility of the humans we're designing for. We suddenly have to consider how our work might negatively impact someone's life (an anxious episode really is a heavy experience) and how we might take this consideration into the later stages of our project. Let's take a look at a third persona.

J is a single parent living below the poverty line. They save enough for a vacation every couple of years and like to let their children choose among a small number of potential options. They often have a limited budget to work with and try to avoid showing the kids the more expensive vacation options. J is sight-impaired and struggles to make out smaller forms or words.

In this case, we have a slightly different set of challenged assumptions. From earlier in this book, we understand that there's a cognitive load associated with poverty. We can also infer that, as a single parent, a certain degree of chaos will be the norm in this person's life and start thinking about how to accommodate that. We're also presented with a concept that challenges what we might show this person. If our app makes money through upsells or advertising, we run the risk of showing J's children an ad for a trip they simply can't afford. If you're on a limited budget and can only afford to get your children to a place that has a beach and a water park once every couple of years, how would you feel if they suddenly got bombarded with a full-screen takeover ad for an all-inclusive Disney adventure?

While these are far from extensive manifestations of user personas (we'd also want to include something that hints at their technical proficiency, the devices they might use, what brands they like, and so on), I hope they offer an insight into how we can use personas to challenge assumptions. For every persona you create, in addition to ensuring they're representative of your research findings try to also work in these challenges to assumptions. The end result is something that transcends the basic goals of your project and forces you and your team to consider the distinctive environmental and personal circumstances people find themselves in every day. If your personas aren't impaired, frustrated, or cognitively affected in any way, then they aren't real enough to be of any use. And while randomly bestowing our imaginary friends with unfortunate situations or impairments might seem somewhat vulgar at first, it's one step on a path toward imbuing our projects with the empathy and compassion they require.

The Important Questions

Research and planning should provide you with answers to the following key questions:

- How do I succinctly define the problem we are solving?
- Who are we solving this problem for and what makes them tick as human beings?
- What frustrates these people?
- What motivates them?
- What distracts them?
- What kind of environments are these problems encountered in?
- What mental models already exist in this problem space?
- How could our product hurt, or otherwise negatively impact, someone who uses it?

The Common Deliverables

Research and planning usually result in some of the following artifacts:

- Competitor and customer research reports
- Stakeholder interviews
- User personas
- Empathy maps
- Problem definitions
- Research videos and documentation
- Usability analysis (for existing products or for your competitors)

Scribbling and Sketching

Scribbling is one of the most important activities of a designer. We all do it, some of us a lot more formally and in a lot more in detail than others. This is the stage where we start taking our high-level plans and turning them into proposed solutions. Through activities

like thumbnailing, wireframing, and paper prototyping, we're able to very quickly get ideas from our head into the world. Everything at this stage is disposable by necessity—we're not here to be precious or protective of our work.

I believe this phase of a project is probably the one with the most widely distributed and disparate methods. Some of us like to keep our sketches and scribbles confined completely to “offline” media—paper, pens, pencils, whiteboards, and Post-its. Some prefer the power and flexibility of digital media—drawing by hand on a tablet, in a purpose-built wireframing tool such as Balsamiq, or in the design tool that they'll later use to put together their high fidelity work.

There is no *correct* way, or place, to perform this work. Whatever your preference might be, it's the *mindset* that you take into it that matters. We need to make sure that our ideas and expectations aren't limited by the tools we're currently working with—hence, why many prefer to get away from screens and do all their early work in notebooks, on the back of napkins, or on whiteboards. However, the flexibility offered by digital tools is, to many others, an integral part of their process. My advice, if you don't have a preference right now, is to try as many different methods of sketching that you can. If you're able to get ideas down on a tablet or computer without being restricted by the medium or distracted by the temptations that are wont to live on such devices, then why not do all your sketching there? If you're like me, and as soon as you look at a screen you're already typing the first part of “twitter.com” into a browser, then perhaps sticking to pen and paper—or a big whiteboard—is the best solution.

Let's go back to that elusive mindset that is so important. What we should be considering at this stage is twofold: our problem definition, and our broad-strokes, proposed solutions. The first consideration, our problem definition, strives to answer the question “Is my problem defined enough to begin solving it?” While it's arguable that the problem definition derived from our research should already have answered this, there are myriad reasons as to why we might enter into our sketching phase with a still-fuzzy definition. Jumping into this stage is a fast-track method of figuring out if your problem definition needs work. As soon as you start mind-mapping or thumbnail sketching, you'll be able to internally query your understanding of your problems and how technology might start solving them.

Sketching is a form of iterative refinement in the sense that we'll be transitioning from a starting point of comparative chaos and slowly refining, embellishing, and redefining our ideas and understandings. This is why a solid research and planning session is so important, and why resisting the urge to prematurely jump into

solution-izing will serve you well. The more data we have to feed into this phase, the more patterns we've noticed and analyzed from our research phase, and the more complete grasp we have of our problem space, the more painless this phase will be.

If the results of our research and planning phase act as a problem definition, then the results of our scribbling and sketching phases act as *solution* definitions. Through documentation like design briefs and feature lists, sketched ideas such as thumbnails and wireframes, and more holistic representations of our products such as storyboards, we're able to very quickly iterate through potential solutions without getting into the nitty-gritty world of visual design and prototyping.

Design Briefs and Feature Lists

A good design brief can often be the difference between a successful project and a comparative failure. I debated whether a design brief is a product of research or that of sketching, and my conclusion was that it bridges the gap between the two. While our research phases will produce many reports and artifacts that explain our products, our customers, their problems, and their cognitive and emotional qualities, the design brief will explain, as succinctly as possible, *what* we'll do to achieve that.

A design brief can take many forms, from a scribbled mind map that lists the features and the problems they solve to a formalized document that both abstracts the research findings and explains how you plan to go about solving the problems you've defined. The format will depend on the size of your team and the layers of approval you might need. Smaller, autonomous teams can get away with their design brief being rough and utilitarian; larger teams or teams with more politics and ego-stroking will likely need a more formalized, persuasive task.

Your design brief, regardless of the format it's in, should answer the questions in the following sections.

Who Are We Designing For?

Who is our target audience? List the challenges they face in their day-to-day life. Reference your personas. Discuss what their common goals and ambitions are. Make sure to present a diverse range of people, with fallibilities, frustrations, and intrinsic motivators that match those of the customers you've interviewed.

What Is the Proposed Solution?

What are we going to build? This can be anything from a quick couple of sentences to a multi-paragraph essay—again, your preferences and those of your team will determine this. Personally, I prefer a couple of sentences that present a broad and open-ended solution concept. For example, “We will build an animation tool that can automatically import static Sketch, Photoshop, and Figma files. It will allow designers to link art boards together, decide how transitions should occur, and finally send a prototype preview link to their team members and their clients.” This is a *broad-stroke* definition, and it shies away from being prescriptive.

What Features Are We Going to Include?

What will our product do? What problems will our features solve, or what motivations will they satisfy? A short table of features for your proposed solution is enough for now, and you don’t need to be married to the list. Think of it as an alternative means of documenting problems or features you feel will make your customers happy or more efficient. Figure 7-3 shows what our hypothetical feature table might look like.

Feature	Why?	Notes
1 Import Sketch Files	Designers struggle to get their static files into an animate-able format easily	
2 Load Figma Project via API	*	Use Figma API?
3 Use any layer as a clickable/tap-able hotspot	Designers fumble through adding hotspots to static files (old InVision etc.)	
4 Animations start on artboard transition	No current easy way of starting an animation chain when transitioning between states	
5 Timeline editing for complex animations	Lack of control over how individual properties are animated	
6 Chain animations (start B when A ends, etc)	No tool in our problem space allows for this	
7 Realtime Collaboration	Designers struggle to share animations and prototypes without conflict	
8 Component Library for animations	Lack of 1:1 representation between design system docs and animated components	Animated components are diff to animated states/artbo...
9 Generate sharing link	Sharing finished prototypes with clients or colleagues	
10 Auto-animate before/after artboard layers	Quickly get basic animation flows prototyped	
11 Switch Between Spring + Easing-based styles	Spring physics don't allow for set-time anims + Easing-based feels unnatural = Use Both	

Figure 7-3. A feature table for part of an example design brief

A Note on Feature Lists

Feature lists aren’t something we can magically put out of thin air! I *highly* recommend stepping back away from the design brief and exploring your feature list as a stand-alone deliverable first. Get away from a screen and throw Post-its on a whiteboard. If you’re

remote, use something such as RealtimeBoard (<http://realtimeboard.com>) to collaboratively throw Post-its around a digital whiteboard.

Feature lists are often the real heroes of design briefs since they comprise the first stage when we actually start presenting some granular potential solutions to our customers' problems. Here's my advice on how best to generate these. Get yourself in a (physical or digital) room with as many people involved in the project as you can muster. If you're on-site, pick the room with the biggest whiteboard. If that room is the CEO's room, kick them out and wipe their motivational Gary Vaynerchuck quote off, and do some real work on it. If you're doing this remotely, get on a other digital whiteboarding tool and share your screen. Have all the other participants log in to the same tool beforehand.

First, make sure you have all your research reports handy (and that everyone who is coming has at least been given the Cliffs Notes of the results of your research). If you're conducting this exercise in person, pull your personas up on a large screen and try to get your user videos up on another. If you don't have access to big screens or projectors, have that one PM who refuses to work on anything but a 17" Thinkpad get them up on their widescreen hunk of plastic. Just make sure that your research deliverables are accessible and in everyone's face. If you're facilitating this as a remote session, share your screen and keep them open in a new tab, or dial in as a separate user solely to show the deliverables.

Divide your board into broad-level problems that you know you need to solve. In the case of our animation tool, our problems might be the following: people struggle to collaborate on prototyping, people can't easily share prototypes, and people can't animate in great detail. I'm purposefully limiting this to three for this example, but you'll want to keep dividing the board up into as many problems as you feel can be handled in this session. You can always attack a new set of problems the next day, so don't worry about being succinct in the name of exploration.

Once you've split your board up into problems, have everyone brainstorm potential solutions. No idea at this stage is a bad idea, so keep the Post-its flowing and take responsibility for morale in the room. This isn't the time or place to shoot down ideas or to assert your problem-solving dominance as the World's Least Awful Designer; it's a place to get a good portion of your team exploring the problem space in an interactive way. Set aside at least five minutes per problem, but don't be scared to move on sooner if ideas are running low. During this time, if people aren't sure what a problem is or why we're trying to solve it, *show them* in your research. At this point, you should be the de facto expert on your research findings, so if someone asks, "Why do we need to even

bother with collaboration?,” you can show them the exact interviews, observations, or other items of research that show this.

Once you’ve filled your board—which could take anywhere from 15 to 20 minutes to several hours—offer the ideas up for discussion. If you’ve exhausted your team and yourself, don’t be scared to take a break and come back and discuss the next day, but be prepared for people to come in with new ideas and to have forgotten all about many of the ones they initially had. This is why I like to split these exercises up into smaller ones—it allows us to get the brainstorming and the discussion done in one sitting. This discussion process should be as streamlined as possible. As the facilitator, start with one problem, pick out a Post-it, and ask the person who made it for a quick, one-sentence explanation. Continue this for every card and let the conversation flow naturally around the more interesting or exciting ideas, but don’t start fully solutionizing yet. Ideas such as *built-in version control* are great—vague and non-prescriptive—but ideas such as *a git-style version control tree in the left-hand sidebar, with a purple-pink gradient timeline indicator and a 2px drop shadow* are terrible. Keep the details out of the discussion and absolutely do not let the meeting devolve into design by committee.

Once you’ve gathered the reasoning and ideas behind these cards, end the session. Make no promises as to whose ideas will “make it” into the feature list and whose ideas will be scrapped. Just thank everyone for their time, end the session, and document the board. Figure 7-4 shows an example of a “completed” board for our app.

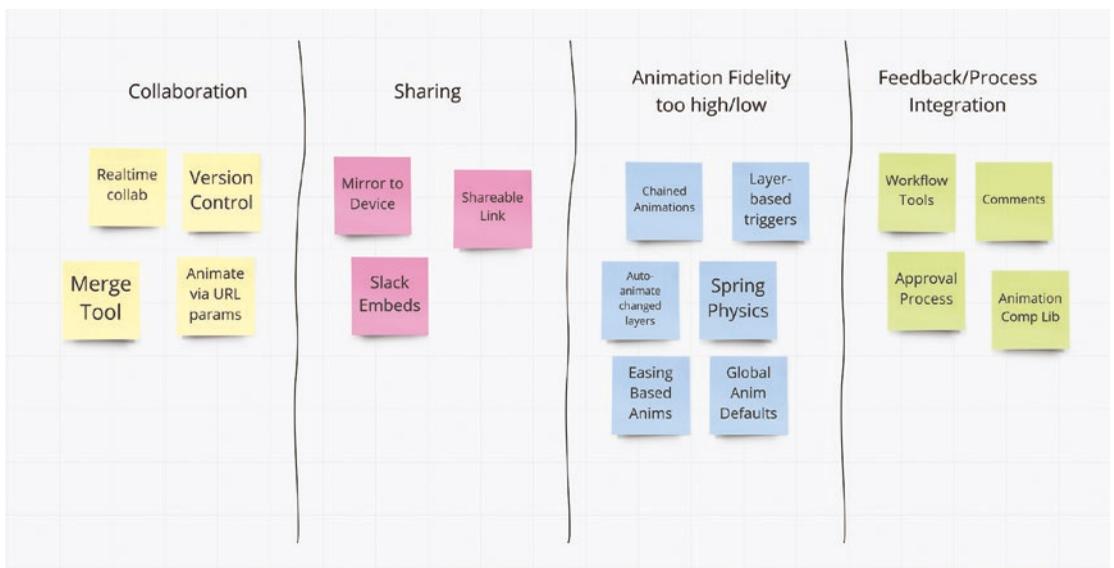


Figure 7-4. Our board full of feature ideas

After that, take the board into a refinement session with your design and product team alone. Your job now is to filter these ideas and start forming the *real* feature list for your brief. This might seem to relegate the previous session to smoke and mirrors, nothing more than a surface-level game of Happy Families, but the reality is that you and your design and product colleagues have been hired for your ability to drag signal from noise. You and your team will understand the problem space better than anyone else involved in the project and, thus, any potential feature set should be filtered first and foremost through this lens.

Many times, the opportunity to bring colleagues into this process won't exist. This is especially likely if you're a freelancer hired into a project as the sole "full-stack" designer. In this case, I'd suggest still performing this exercise—just do it on your own. Find a room or a corner, grab your Post-its, throw some Enya on, and enjoy the peace and quiet.

Success Metrics and Goals

One of the key aspects we'll need to consider is how we'll judge the success of our product. Again, this book is not a design strategy book, so I'll keep this brief. We've already discussed what we're building and the problem space we're operating in, so this section will explain how we know we've succeeded with our work. This could be through quantitative improvements such as a reduction in customer support tickets and a conversion rate increase of at least 16%, or it could be through qualitative improvements, such as noticeably saving our customers time or improving their lives. If you're not sure how best to suggest these metrics, speak to your client, manager, or a key stakeholder and discuss what their ideas for success are. You could even collaboratively build a list of clear markers of success.

I recommend this approach to a design brief for any situation where your team or client is comfortable with terse and open-ended solution descriptions. If you go with a more open-ended design brief, it can act as a great primer to your wireframing and prototyping work. However, if you're dealing with particularly difficult stakeholders who want more fully-fledged documentation, you might need to spend a little more time fleshing your ideas out. My advice here is to, wherever possible, invite these stakeholders into the process. Explain the research findings to them, invite them into your thought process, show them your customer interview videos or your competitor user testing, and explain to them why the problems and features in the brief exist. Often, our office

politics simply boil down to people feeling like they've been left out or don't have a seat at the table. Fix this by inviting them in early and working their contributions into the design brief.

Finally, consider turning your design brief into a presentation. As part of the team (or the sole designer) who defined the problem space *and* the proposed solutions, you're going to need to get practice in explaining the ins and outs as well as the whys and the why nots of this proposed solution. Treating the design brief almost as a pitch to stakeholders can help you polish your definitions, practice answering on-the-spot questions, and clue you in on any early-stage objections or potential conflicts.

Wireframes

Wireframing is, in my opinion, where any early-stage or full-stack designer proves their worth. While design research is extraordinarily important and, coupled with a solid brief, will form the foundations of our projects, wireframing is our first foray into domain-specific solutions. Wireframes are where we take all the theory surrounding a project and start down the path of practical, presentable, and testable solutions.

Having said that, it's often highly tempting to jump into the wireframing stage prematurely, so make sure you've got some solid research, a good brief, and an actionable understanding of the people and problems that your designs will attempt to serve and solve.

I see the purpose of wireframes as twofold. First, the traditional definition of wireframes as the structural representation of a potential interface is absolutely true. They're our blueprints for structure and form the basis of our hierarchy, information architecture, and logical groupings. Second, wireframes act as our first opportunity to start testing our conceptual models against people's mental models. This has been a running theme throughout this book, and I think it's something that many designers (and I absolutely include myself here) can easily forget about at this stage. The excitement of your solutions finally taking shape and the inevitable gun-jumping we all do when we feel as though we've touched on an elegant solution to a complex problem both conspire to have us double-down our thinking further toward solution intricacies and further away from the system, conceptual, and mental models that we're attempting to reconcile.

As I touched on in Chapter 7, when thinking in terms of environments, moments, and models, we'll be doing a lot of metaphorical zooming in and out in terms of our processes. This is just as true for our wireframing as it is for our visual and interaction design phases. As you progress with your wireframes, you'll want to make sure that you're questioning your solutions against existing mental models that you've (hopefully) managed to garner from your customer and competitor research.

Thumbnail Sketches

Usually, one of the first ports of call when wireframing is to perform extremely basic thumbnail sketches. First, you need to have roughly broken your content and/or features down into broad structural categories. Let's say, for our prototyping app, we have the following:

- A list of layers
- The filename
- A profile menu
- A timeline for tweaking animations
- A toolbar for setting animation variables such as easing and timing

Divide a page of a notebook or your digital canvas into a number of small rectangles, like those shown in Figure 7-5.

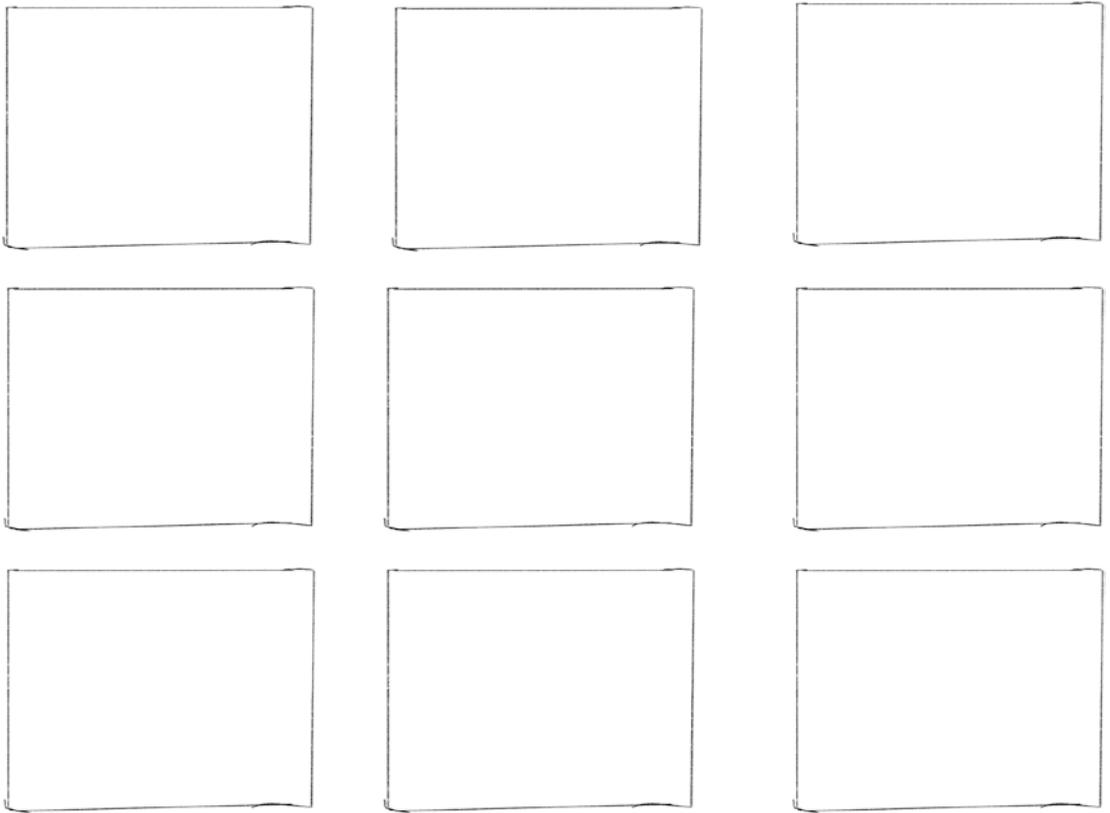


Figure 7-5. *A page prepared for thumbnail sketches*

Now, try to divide these boxes into those broad, structural areas of your interface. Work quickly and try not to overthink or overdesign things. If you're working on an analogue notebook, try and keep your boxes no bigger than a few square inches. If you're working in a large notebook, use thick-tipped sharpies or brush pens. If you're working on a digital device like a tablet or touch-screen monitor, *do not zoom in*. The goal with thumbnailing is to consider the different ways you can lay out your information and features, so work in rectangles and lines. My favorite tools for thumbnailing are a small whiteboard and the most offensively thick marker I can find.

Move fast and set yourself a goal of filling 6-12 thumbnails in one sitting. You'll notice that the first few thumbnails come readily and easily—namely because you'll almost always have an idea in your head of your interface before you even start sketching and because you'll have seen examples in your competitor research. Things get harder and harder as you exhaust your initial, most obvious ideas. Don't stop until you hit your thumbnail goal and don't worry if your ideas are outrageous or silly at this stage. Figure 7-6 shows the results of a thumbnailing session I did for our hypothetical animation app.

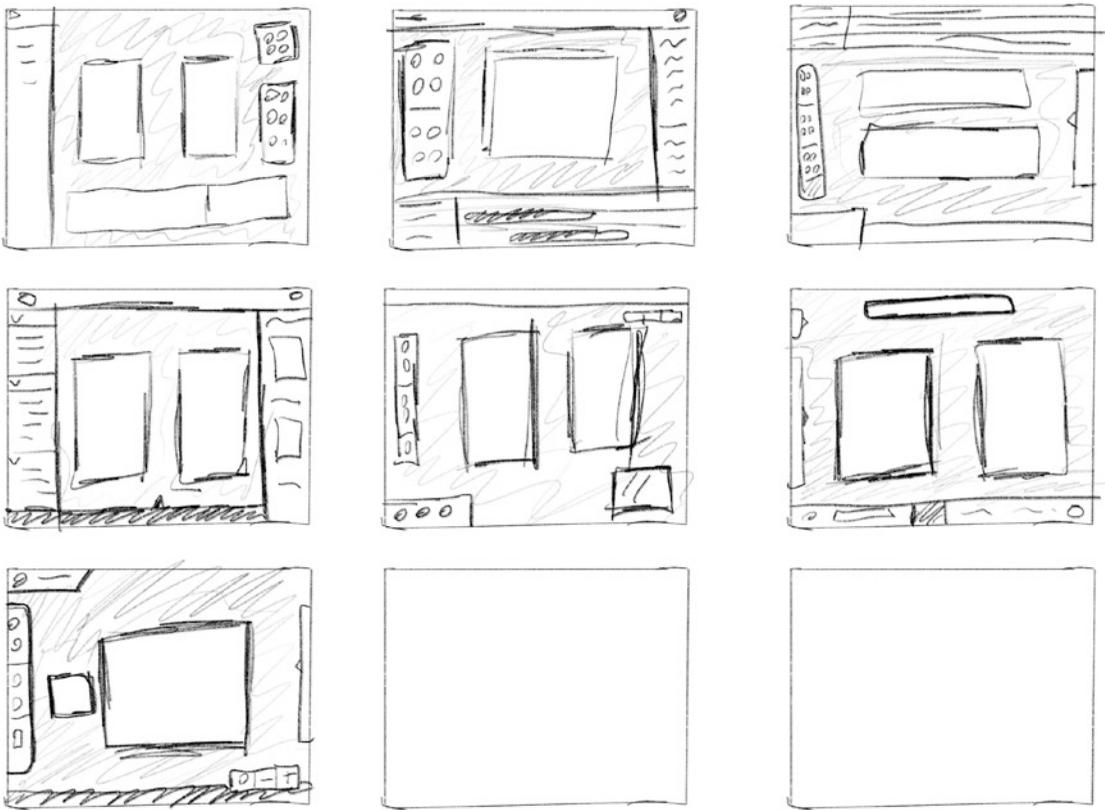


Figure 7-6. *The thumbnail boxes filled in*

The point of thumbnailing is to explore solutions for visually arranging your broadest application areas. Think in a suitably broad mindset when doing this exercise. Even at this point, however, consider convention and mental models. There's a reason that apps in the same problem space are often structurally similar; it rarely suggests a lack of creativity on the design team's part. Even at this early stage of wireframing, we need to be thinking about how we balance what we believe to be an elegant or innovative solution with the mental models and convention that we *know* our customers will expect. Now, we don't have to think this critically *while* we thumbnail; this is our unconstrained and expressive flow of creativity after all. However, when it comes to *analyzing* them, our most creative solutions are often our least practical.

Interface Sketches

Once we've analyzed your thumbnails, we can move on to turning them into more detailed sketches. Depending on how confident you are with your thumbnails or how experienced you are with the wireframing process in general, you'll either want to do some more-detailed thumbnailing work or just go straight into "full" wireframes.

If you're not super confident with your thumbnails and you have several options you think might work, try upping the fidelity a little and working at a larger size. Still, stick to mostly rectangles and lines, but don't be scared to add a little shading or highlighting here and there. Alternatively, perform even more rounds of thumbnailing. The beauty of these stages of our process is that everything is so disposable and so quick to create.

Once you have two or three potentially workable thumbnails of whatever fidelity makes you happy and confident, you can start putting together some basic interface sketches. Think of an interface sketch as an iteration on your thumbnails. Here, we'll start introducing actual words, at least for our main actions and headings. We'll also start separating our interface elements more granularly, through shading, highlights and hairline borders as we break away from simple rectangles and lines. We may even get so fancy as to consider a drop shadow here and there. Figure 7-7 shows a progression from the previous thumbnails to a very slightly more-detailed interface sketch.

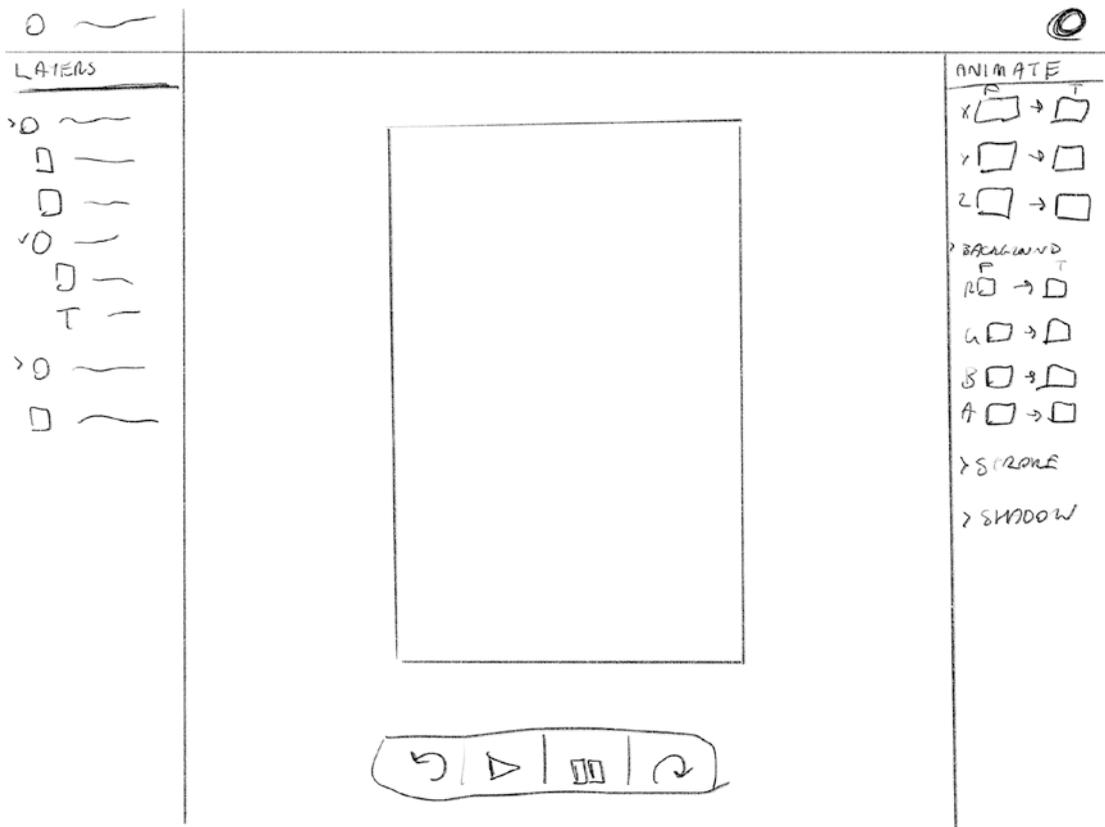


Figure 7-7. A slightly more-detailed sketch, based on our previous thumbnailing example

At this point, we want to think back to Gestalt. One of the most integral aspects of structural representation is perceptual grouping. If you need a refresher (I always do), try wireframing with a Gestalt reference (flip back to Chapter 2!) in front of you and consult it if you get stuck.

These more-detailed wireframes are also an excellent place to start reviewing your content in context. While content strategy and information architecture are separate disciplines to design, with their own processes and methodologies, wireframing is often the point in a project where these processes first truly converge. Showing your content team their work in a structural and more contextual blueprint can often spark some great collaboration. Remember, too, that when we test our wireframes, we’re also testing our content. Banish *Lorem Ipsum* wherever you can, and get real words and real data into your sketches wherever possible.

Storyboarding and State

If you're designing an application that is going to change drastically based on state or a web site that you know is going to have multiple templates or page designs to meet varying goals, you'll want to repeat this thumbnailing-sketching-iterating process for your key pages or permutations of state. A great companion exercise for this is storyboarding.

My preferred approach to storyboarding will be discussed in explicit detail in Chapter 9, so we won't dwell on it for too long right now. Just keep in mind that, as well as presenting ideas for main structural areas, wireframing is as much about laying down the foundation for the moments and exploration that our interfaces will allow.

The Important Questions

Scribbling and sketching should provide you with answers to the following key questions:

- Have I defined our problem in a way that opens it up to solutions?
- Have I presented a clear and concise brief detailing the reasons this product should exist?
- What functionality (and thus system models) does our feature list dictate?
- How is our proposed solution compatible with convention and existing mental models?
- How do our proposed components and structures update to reflect state?
- What are our key moments?

The Common Deliverables

Scribbling and sketching will usually result in the following:

- Design briefs
- Thumbnail sketches
- Mind maps and flow diagrams

- Site maps and feature lists
- Wireframes
- Paper or digital prototypes
- User journeys, stories, and scenarios

Testing and Iterating

While I've included testing and iterating as its own phase, and included it last in this chapter, it's imperative to stress that this is neither a discrete phase nor is it something that should be restricted to the end of a design process.

In reality, we should test and iterate during every phase of our work. Often we'll go through dozens of rounds of testing for a single phase, *especially* while we're wireframing and prototyping.

Testing is our means of discovering and understanding the validity and efficacy of our potential solutions. It's how we prove our hypotheses. It's how we reveal the underlying biases and assumptions that are holding our work back and the happy accidents and small touches that highlight the unexpected moments of enjoyment and expression that people may encounter.

Given that the design process is often cyclical, it can be difficult to separate testing from research. For the purposes of this chapter, both testing and research are almost functionally identical. The rules are the same: we need diverse recruitment, informed consent to record and annotate our sessions, and respect for our participants. The only real difference is that testing is used to prove, measure, and observe our current work, while research is focused on understanding our problem space and our competitor's tools.

User Testing

User testing is to the testing and iterating stages of a design process what customer research is to the design research phase. It involves getting our work in front of real people in their real environments, where we can observe them using our products and interview them about their experiences with them.

While customer research undeniably works best as a field observation exercise, user testing can—depending on the phase you're testing and the assumptions or hypotheses

you're validating—be conducted in slightly less realistic environments, including a remote session. Before deciding how best to conduct user testing, consider the phase you're in and the assumptions you need to validate. If you're in the wireframing phase and you're trying to discover if your proposed conceptual model fits in with your customers' mental models, you'll likely want to conduct a more guided, interview-based testing session. Given that wireframes are often still just representations of structure, they almost always require a form of priming or explanation. If we're mostly considered with mental models, we're actively going to have to *ask* our participants about their expectations and past experiences.

Contrast this with the kind of testing session you'd conduct on a completed product or a high-fidelity, fully animated prototype. In this case, we'd likely want to conduct a number of field observation tests in the best facsimile of our customers' environment as possible. Refer back to the customer research section of this chapter for a primer on conducting such sessions.

The kind of information we wish to garner from these sessions is also just as dependent on the phase we're testing. In our wireframe example, we're looking to see if the basic structural, conceptual notions we're presenting match the mental models and expectations of our participant. Slight confusion over the shape of a button is negligible, but shutting down at what we believed to be a super-innovative and elegant solution because we are unable to reconcile it against our users' expectations or model is a huge issue. With regards to our finished product or hi-fi prototype test session, we want to explore how well our product holds up against all the intangibles and unpredictable occurrences that daily life throws at our participants. Are they able to get back to their task if they're distracted at any point? Are we operating at a low enough cognitive load that work can be done among the bustle of a busy coffee shop? These contextual and environmental considerations are absolutely paramount.

The best testing sessions are designed to throw curveballs at us and force us to discover the areas of our products that are blighted by assumption and bias.

High-Tech Testing

I've had the opportunity to conduct and observe testing sessions in fancy usability labs with intricate eye-tracking rigs and a whole host of biometric monitoring. Honestly, though, absolutely none of them were anywhere close to beneficial to a project, at least compared to well-conducted field observation and interviewing.

As an obvious proponent of neuroscience, cognitive psychology, and scientific methodologies, it might seem strange that I would decry lab-based testing and experimentation, but here's the thing: design testing is not a precise scientific field. It benefits from a lot of the core tenets of scientific methodologies, especially in recruitment and data collection and analysis, but when we're looking at our testing and research methods and results, we're far closer to the realms of ethnography than we are of rigid scientific experimentation.

A well-conducted field observation will give us qualitative data on our subjects, such as their habits, their frustrations, what distracts them, the particulars of their routine, and what motivates them. This is actionable, humanized data that almost anyone can look at and *at least* empathize with to a certain extent. Contrast this to a typical usability lab result set, where we'll be collecting quantitative data such as eye-tracking heat maps, facial expression data, and sometimes even skin sweat analysis.

Frankly, the best use I've ever seen for such a UX lab is as a marketing tool for design agencies.

Now, I don't believe eye-tracking data to be completely irrelevant—that is categorically not the case—but it doesn't *tell a story*. You cannot, from how someone lingers or skips merrily through the various parts of our interface, accurately deduce that their held gaze was that of confusion, sheer awe of our visual design skills, or just a case of the mind-wandering absentmindedness that is wont to skip off hand-in-hand with our focused attention at any given moment. Eye-tracking, *alongside* our more qualitative ethnographic research and testing, can be a useful tool. Remember that our goal with testing is to reveal the assumption and biases inherent in our work. If our assumptions are “this part of the interface will demand the full gaze of someone when they are asked to animate from one art board to another,” then absolutely break out the eye-tracking and go for it. However, I vehemently believe that design testing doesn't revolve around these kinds of questions. Heat maps can be useful auxiliary data, but often stakeholders get caught up in the “highly scientific” realm of tracking and biometrics. Your carefully crafted field observations and emotional connection with your customers is suddenly open to debate because “the science disagrees with you.”

Facial expression coding and galvanic skin response (GSR) testing are the last two methods. They're both, in my humblest of opinions, absolutely useless to any form of design research or testing. GSR involves attaching electrodes to peoples' hands. The electrodes measure sweat gland activity and then use this activity to determine a level of emotional arousal. This is often combined with facial expression coding techniques

to determine whether the supposed strong emotional responses are negative or positive. First, the idea that any one emotion has a discrete and measurable biological “fingerprint: is devoid of consensus and evidence (Barrett, L. F., 2017) and, second, the amount of interference and abstraction these laboratory conditions enforce makes their results questionable at best, and complete garbage most of the time. Here’s the thing—if you start attaching stuff to people, they sometimes, kind of, *maybe* don’t enjoy it. If you invite them to a laboratory, put a glove on their hand, say, “This is going to measure your sweat, enjoy!” and then pop off to your terminal, you are putting your subjects in an environment that is so far removed from their day-to-day encounters that your results are going to border on meaningless.

This, for me, is the crux of testing: we need to test in context, in the field, and around the distractions and motivators that our subjects experience. While psychology and neuroscience both benefit immeasurably from controlled laboratory conditions for many of their studies, design does not. We sacrifice our controlled conditions for the real-life chaos that our work will be viewed in. Our research and testing is not an academic pursuit; we’re trying to improve people’s lives and solve people’s problems. We get that from understanding them at a human level.

This brings me to my main concern with user testing as a whole, at least in relation to this book: it should be done *in context* as much as possible. Through our research, we should have a list of all the environmental and on-device distractions we can expect our work to do battle against in its day-to-day usage. Through our grasp of the relevant brain science, we know the kind of mental processes and potential issues that our work must accommodate. When it comes to testing, we have to make sure that we allow for these intangibles to unfold. Creating a completely controlled laboratory environment is not how we achieve this. If you *do* wish to conduct controlled environment experiments, bring some distractions into the mix. If you’re testing a mobile application, don’t put your subject in a closed, empty room or corner and hand them an empty iPhone on Do Not Disturb mode. Fill the room with distractions, put a movie on, have someone offer them a glass of water at a random point in the experiment, ping the test phone with notifications and messages at varying intervals. Like I said earlier, if your subject has children—or a particularly needy pet—have them bring them along. Let the overfriendly office dog loose. You can even let your CEO come up to them and absolutely thrill them with a tale of how their latest VC pitch went.

Recreating the chaos of the real world in a controlled environment might seem counterproductive to a data-gathering exercise, but we’re not here for pristine data or

to produce a whitepaper that will stand up to continuous academic scrutiny. We're here to learn about how suitable our work is *for the real world*. If, for whatever reason, you can't get out and test in your subjects' common environments, bring the chaos of the real world to them. A huge part of design testing and research is learning to spot the intangibles in an environment or a person that either informs the problem space we're exploring (in the case of research) or qualifies the efficacy of a decision we have made (in the case of testing). If you feel like you could use some work on this, take a course on ethnography or grab a book on the subject. Resist the urge to equate design and usability testing with fancy labs and ridiculous sweat gloves. Your mental image of a design researcher should be you, sitting in a room and watching someone try to simultaneously use your product while they juggle their toddler, a cat, their nosy neighbor, a TV they can't find the remote for, and the undulating stress, distraction, and daydreaming of everyday life.

Iterating after Testing

Iteration is the act of adjusting, changing, or embellishing your product away from a previous manifestation toward another one. Changing a button from blue to a slightly different blue is an iteration. Pivoting entirely from a flight-booking application to an electric-scooter sharing company because your main VC read a *Wired* article is an iteration. One requires a lot more work than another.

It's exceptionally hard to quantify iteration—it's simply *doing the same work* you've already done, but under different constraints or within an adjusted problem space. When we first start scribbling, we do so based on our *current* understanding of these constraints and spaces. Every time we test our work, we're essentially opening ourselves up to further changes to its foundation. If, for example, our research suggested that collaboration was a key feature of our prototyping application and we assumed that the problem we were solving through collaboration was that of version control—understanding who made what changes, the ability to revert changes, etc.—then we might propose a fully featured Git-style timeline as a feature. If we then tested this assumption and it turned out that the underlying problem was an inability for two or more people to work on the same prototype simultaneously without causing conflict, our problem space is suddenly very different. Instead of version control, we might want to consider real-time collaboration functionality.

This is what testing gives us. Through a slight misdefinition of our problem, we've incorporated a feature that no one needs—and we've discovered this by getting it in front of people. If we learned this by testing our wireframes, we've just managed to save *a lot* of time and effort, our super-complicated version control feature doesn't make it beyond a quick scribble or a structural blueprint, and our problem definition gets updated and we resketch and rethink based on that. If we learned this by testing our final product, erm ,, , *oops*.

Every stage of the design process should be open to iteration. The further toward “finished” we get, the more expensive it becomes to fix issues with core problem definitions or challenged assumptions. So, test your wireframes. Heck, test your *personas*. Show them to customers and ask them if they can relate to them in any way. The earlier you can have you and your team's incorrect assumptions and biases laid out on the table, the less expensive it becomes to redefine your problem and rethink your solution. Iteration is not simply the procedural improvement of an area of your interface—although this is a form of iteration—it's also a systematic reviewing, editing, and realization of your problem definitions and a cyclical rethinking of your solution. Test early, test in context, and test often.

The Important Questions

Design testing should answer the following questions:

- What false assumptions have I made?
- What biases have I introduced in my work?
- Does our product “work” in real-world situations?
- Am I still solving the correct problems?

The Common Deliverables

Design testing will usually result in the following:

- User testing reports and videos
- Usability analyses
- Accessibility analyses

- UX reviews
- A/B test reports
- Snap-judgment test reports
- Revised versions of other deliverables (new wireframes, prototypes, or designs to accommodate test findings, for example)

Summary

With such a diverse range of skill sets and disciplines in modern design jobs, and with the responsibilities, frameworks, and methodologies seemingly continuing to grow apace, documenting the design process is a thankless and arduous task. I've tried to limit this chapter to the areas of the design process where I feel like we—as a union of design workers—can make the most impact in humanizing our process. I've also discussed areas where I think it's very easy to lose track or deviate from our goals of creating problems that operate in a space of compassion and empathy.

The phases and deliverables discussed here are far from a complete overview of any atomic design process—partly because such a thing no longer exists—but I do feel that the key talking points can permeate throughout the rest of *your* process, whatever that may be. Remember that our research—including our design, usability, and accessibility testing—should provide us with insight not only into the high-level preferences of people, but to their habits, their environments, their distractions, frustrations, and the *stuff that makes them happy and motivated*. When distilling this information down into user personas, make sure that your user personas challenge the assumptions and biases evident in your own thinking and that of your team members. Personas should not serve already existing assumptions. A persona that doesn't provide a challenge is one that exists simply to placate.

When sketching and wireframing ideas, think both in terms of the “traditional” wireframing goals of correct structural representation and early-stage exploration *and* in terms of reconciling mental models. Use your understanding of your competitors and similar tools that exist in your problem space—especially if you're trying to bring a traditionally manual process into a digital environment—as well as customer interviews and observations to map out the commonalities in language, structure, and process. If you understand these mental models well enough, you'll be able to understand what can stay in as convention (the razor, for cutting a video clip) and what you might want

to do differently, deviating from a mental model to solve a problem in a way that you're confident people will learn. Think also in terms of state and state transitions—exploring moments and movements as well as static representations and linear flows. More on this one in Chapter 8!

Remember that testing is as much a means of further revealing assumptions and biases that slipped through the cracks as it is a means of validating the efficacy of certain interface features or components. Design testing and research are intertwined. Just as our best research manifests itself in ethnography and qualitative study, so too do our best testing sessions. High-tech design testing should never be our *only* form of testing, but eye-tracking, in particular, can provide us with a degree more of context to our qualitative explorations. Just remember that the qualitative “cold, hard facts” of such studies can easily be overhyped. Remember that iteration is as much about iterating on your *definitions* as it is on your product. Redefining your problem at an early stage is a lot less expensive than releasing an untested product full of assumptions and biases that solves problems no one asked for.

I realize that I've almost completely skipped over visual, motion, and interaction design in this chapter, but that's not because I think they're unimportant! Quite the contrary. I'll be devoting a large part of the remainder of this book to discussing how we might marry these practices with some of the early-stage and theoretical concepts we've explored so far. These areas of design rely heavily on problem definitions being validated and scribbles being representative of a solution that we're confident in. The practices listed in this chapter are our vehicle for reaching that point. Before we get into these areas, I want to take one last chapter to discuss process—namely, the new and underused concepts that I think we can weave into our design process to further imbue it with compassion, empathy, and a desire to serve the human mind.

CHAPTER 8

Emotion Mapping

As an interlude between our discussions on the design stack, I want to present what I've affectionately been referring to in my head as "the Post-it Chapter"—namely, because everything in here involves a *heck of a lot* of Post-its.

"Design documentation" is, conceptually, an extremely nuanced subject to address. The form our documentation takes will depend on what stage of design we find ourselves at, how "design-savvy" our clients or stakeholders are, what our relationships with our colleagues (especially those in research, product, and engineering) are like, and much, much more.

Some design documentation, like design specs and design briefs, are simply just words or lists, maybe with some links or references to external sources like research findings or wireframes. Other forms (often the best forms) of documentation are often photos of jumbled and chaotic whiteboard brainstorming sessions. At the tail end of our projects, design documentation usually takes the form of pattern libraries, either as stand-alone documents or as a living, breathing part of a design system.

Furthermore, and rather unintuitively, there's a relatively stark, but often overlooked difference between documentation *of your designs* and documentation that *assists with your design work*. The first is used to sell and to communicate (as designers, we need to sell our work; it's part of the job) your work, whether that's a write-up of your thinking behind a certain feature or a complete Keynote presentation of your final prototype. The second is used as a means of informing *your* process (this is what this chapter covers). If you're interested in how to sell your work or in any way worried about communicating your decisions, Mike Monteiro wrote *Design Is a Job* for you.

One could quite easily present a case for design itself to be considered a form of documentation. Design acts as an abstraction between a system and a human. It's not much of a reach to consider that abstraction as just another form of documentation. While I find this rather reductionist (and I'm also not about to argue with my own straw man), I think there's a level of truth behind the notion. Throughout our design

process, we're producing work that blurs the boundaries between idea, execution, and documentation. Take wireframes as an example. They're a representation of an idea in a way that communicates our vision, without the extra layers of design required to call them an *execution* of sorts. We can quite easily turn wireframes into interactive prototypes, whether that's through a digital tool like Figma or InVision or through good, old-fashioned, paper prototyping. However, no matter what we do with our wireframes, they'll never be the finished *thing*. In this sense, they're a means of progression from one level of granularity to another.

I see wireframes as documentation, and I think this is important to state—not to denigrate wireframes (or any early stage design work), but to elevate documentation.

In the world of tech, documentation is often seen as a chore—an afterthought that we believe our process mandates in the name of smoother collaboration or to convince our bosses that we know what we're doing. When I write code, I loathe documentation. It usually means going through and writing tests, or adding comments, or endlessly updating a README file with every change I make. Equally, when I design, I loathe documentation. It usually means explaining what I see as “obvious” design decisions or looking around for a useful link that justifies my intuition to make a success message bounce using over-the-top spring physics. Because I'm a professional.

Yet, documentation doesn't *need* to be a boring, box-checking exercise—*especially* as part of the design process. It's my goal in this chapter to convince you, as I've convinced myself, that the documentation we proactively produce throughout our process can invariably improve our work and break down some of our more complex lines of enquiry down into digestible and actionable insights. Documentation bridges the gap between one phase of design and the other. Treating personas as a document of our research results lets us confidently go into storyboarding, or thumbnailing and wireframing, or whatever else our process might involve. Similarly, seeing wireframes as a means of documenting ideas or pattern libraries as means of documenting design principles and visual language should raise the profile of documentation in our process.

Documentation is a way of removing the mystery from our work—a means of distilling ideas down into digestible and ascribable forms. We won't be producing reports or retroactively justifying our decisions. We'll be creating a diverse landscape of reference that demystifies our process and gives definition to the fuzzy intangibles that we often struggle to work with. And we'll be doing it by using *lots* of Post-its.

Thinking in Systems and Moments

One of the fuzziest concepts I feel I've presented in this book, that we're yet to truly deep dive into, is the concept of 'system' thinking and 'moment' thinking. In chapter 7, I presented the arguments that digital environments should be the focus of our work, that design is both a communicator and a function of system state, and that we will often need to shift our thinking quickly between high-level system thinking, and lower-level, moment-to-moment thinking. I think the best way to discuss these concepts is to simply tell you how I document them, with some explainers along the way. Spoiler alert: it's probably stuff you already do, without realizing it, because you're smart and great.

System Thinking

Let's limit the scope of our definition of "system" for a while. Until now, I've been purposefully vague with my use of the term because it can encompass any part of the product we're making or the company we're making it for. For now, let's say that when we talk about our system, we're simply talking about our digital product (forget about our infrastructure, support teams, sales process, and marketing material, even though it's important to stress that they *are* constituent parts of a broader system, alongside our product). In this sense, our system is the sum of the constituent underlying workings of this product, the features we have to present, and the state that we have to design around.

If our product is a photo-sharing and editing app for smartphones, then our system is everything behind the scenes that make this product work. As a list, we'd start with the following:

- Accessing the phone's camera and photo feed
- Taking a new photograph
- Editing a photograph
- Saving a copy
- Uploading a the photograph to our server
- Viewing other people's photograph
- Following and unfollowing people
- "Liking" someone's photograph
- Seeing who "liked" yours

Our work, then, as designers, is to communicate this system in a way that allows people to achieve their goals. We take a potential system, we learn about the problem space in which it's going to exist, and we turn it into something useable, perhaps even delightful. Simple, right?

Except, we're dealing with humans, and our system exists as a series of 0s and 1s, as a complex flow of data from one point to another, and as a concoction of API calls and permission requests and package loss and compression algorithms. People don't *get* systems. They might get one *part* of a system, but the rest may be lost on them. A professional editor might understand things like HSL and exposure sliders, but be completely lost if we expected them to "just get" our compression algorithm. Furthermore, what about someone who has no idea about compression *or* photography? How do they interact with something that simply presents all the features of a system with no abstraction?

The answer is, well, *they don't*. This is why we have jobs. As designers, we translate and abstract away our system language and functionality into people language (or brain language, if you want to be super cool). Does someone who wants to share a few photos of their kids at the zoo want to directly manipulate a RAW file with complex sliders like HSL and calibration? Or do they just need a pleasant filter and a quick auto-enhance? But what of the professional who *does* want those options? How do we keep them both happy? Does *anyone*, professional or otherwise, care about our compression algorithm? No? Okay, so we don't talk about it—we let it do its thing in the background.

This methodology puts design forward as a translation of sorts. Our system has a certain way of receiving input and getting things done, but that method is not intuitive or compatible with the mental models or cultural conventions that exist in the minds of people. "Send the text entered into this textbox to our server and attempt to store it in the comments database" becomes "Send." "Access the phone's camera but make sure it always inconveniently defaults to the front-facing camera so they see their own delightful thumb head" becomes a camera icon.

This translation of our system language into a human language (and we're including iconography, implicit grouping, and visual design as "language") is, for me, the core of our work as designers. We take our system, we learn about the people and processes that exist in our problem space, and we make sure *they* can achieve their goals with *it*.

Don Norman consistently uses the concepts of *system image* and *conceptual model* to discuss these realms. Norman's way of abstracting this is to suggest that designers take their own understanding of the system (the designer's conceptual model) and create a system image that another person then communicates with. With this approach, the success

of the system image (the designed product) is dependent on the designer’s conceptual model being understood—that is, there are no gaps between the designer’s models and the audience’s. This is Norman’s attempt at conveying design as a conversation, and I think it’s a wonderful starting point for discussions around system thinking.

Documenting Systems

It should come as no surprise that in order to effectively translate our system, we need to understand it. We also, by proxy, need to understand the people who will be interacting with it and the environments in which they’re likely to do so. When we talk about documenting a system, I feel the best abstraction we have is the concept of the system’s *state*—another discussion point from Chapter 7. When it comes to understanding people and their environments, however, much more interactive and interesting forms of documentation arise.

We’ve already discussed user personas, which I believe are likely the most ubiquitous form of design documentation. (User personas, in essence, are a documentation of research results, an abstraction of our audience.). Turn back to Chapter 7 if you need a refresher on them. Aside from personas, empathy maps and mental models are incredibly useful tools for documenting the emotional, cultural, and environmental factors that permeate our design process.

State Diagrams

A state diagram is essentially a big list of the underlying variables that constitute our system. It’s essentially a parallel of what a developer would consider if they were creating a front-end store or a database schema.

(Okay, so hear me out. I know I promised that we were going to explore the fun side of documentation. And now here I am, talking about databases and variable lists. We’re getting the boring one out of the way first. I promise.)

As dull as state diagrams sound, they represent an integral insight into the underlying function of a system or product. In Chapter 7, we discussed design acting as a function of state, where the current state of our system determines what we display and how we display it. Understanding what makes a system tick is the first step in deciding how we translate it.

The granularity of our state documentation will depend massively on the phase of the design process we’re at and whether we’re working on an existing product or kicking

off a new idea from scratch. If we decide to mock up a state diagram early on, we must be prepared for it to change dramatically as we remove and reconfigure our product. If we're mocking one up later in the process, we'll likely be tempted to work at a more granular level. Just remember that you don't need to be writing pseudo-code here. You just need to explain in broad terms the data that underpins your work.

Let's look at a finished example (Figure 8-1) and break our state diagram down from there.



Figure 8-1. An early stage representation of state

This state diagram is likely a far cry from a complex database schema (if you would prefer to visualize state that way, go for it!), but it has all the hallmarks of a good conceptual model. Breaking a system down into its constituent parts and logic is the name of the game here, and this is why I'd suggest doing this kind of exercise with a developer whenever possible. While I'm loathe to play into the stereotype that developers aren't creative (they literally create something out of nothing—it's creativity) and that designers aren't technical (we sometimes use ProMarkers for coloring—they're technical), there is no escaping that part of a developer's job description is this very approach: breaking stuff down into variables and logic.

If you're part of a team that has access to developers, either as an in-house team or an agency, I'd love to assume your process involves getting developers involved with the problem definition early on. If not, chat to your product team and find out why. If you're a freelancer and you're not sure if the client has sorted development yet, ask them! And if you build your own designs, then awesome—you already have the ability to pick your ideas apart like this. Got a developer willing to help (or got yourself warmed up for a data-heavy exercise)? Then break out the Post-its and get to it.

First things first: define the scope of your documentation. You're not always going to be doing this for the full representation of your system. If you've got a pretty complex system, it'll make a lot more sense to focus on one or two main "chunks" of state first. You might even be working backward from a "moment" you've storyboarded out (more on that a little later) and creating a state for a specific linear flow.

If your system is going to be complex (if you've got more than a few chunks of state in your mind at this point, you can bet it is), focus on your key features first.

Say we were working on our animation application from Chapter 7. We might want to forget about account creation, onboarding, file imports, and integrations for now, and just focus on nailing our understanding of the animation creation and editing interactions. Perhaps we've got a quick wireframe of a potential animation edit panel together and a rough understanding of what information underpins that. It's time to document it.

Without going full *Programming 101*, we're going to be dealing with a few variable types, and it's important to have a functional grasp of what these might be:

- *Booleans* can be true or false. They answer yes/no questions like "Is this person logged in?" They represent the building blocks of your system logic. Most of your state diagrams will revolve around booleans.

- *Numbers* are, well, numbers. They can be integers, doubles, floats, or all kinds of other types. But we don't really care at this level—numbers will answer questions like “What value is the saturation slider currently at?”
- *Strings* are text characters. The name of an animation will be a string, as will a customer's e-mail address.
- *Objects* are a data structure that is represented by keys and values. We could have a “customer” object that has a name, an e-mail, a list of hobbies, and their age, for example. They're basically a means of saying, “All these different variables represent a specific *thing*.”
- *Lists* are a collection of variables or objects. They're usually called arrays. Lists are ways of grouping discrete bits of information together. Your shopping list could be represented as an array of strings, that looked something like this:

```
["apples", "mountain dew", "cream for weird foot thing",  
"doritos", "Mindful Design by Scott Riley", "bath bombs"].
```

It's a way of keeping relevant information together.

Don't worry too much about learning these items in detail. You're not being tested on your knowledge of variables, and you are *absolutely not* creating a spec or a schema for a developer to follow. That's not our job. Even if we code our own stuff, that comes later. Just work these general ideas into your vocabulary and don't be worried about using a reference or asking for feedback or help while you put your diagrams together.

So, now that we know roughly the kind of data we'll be working with, let's start throwing some Post-its at a wall. Start broadly, thinking in terms of the large objects or considerations of the system. If you've limited our focus to creating an animation, then ask yourself, “What does the system need to know to allow for this?” For us, the big concerns will be our document structure, the currently selected layer, the start values of this layer (X, Y, width, height, etc.), the end values, and the timing function for the animation. These are all mapped out in Figure 8-2.

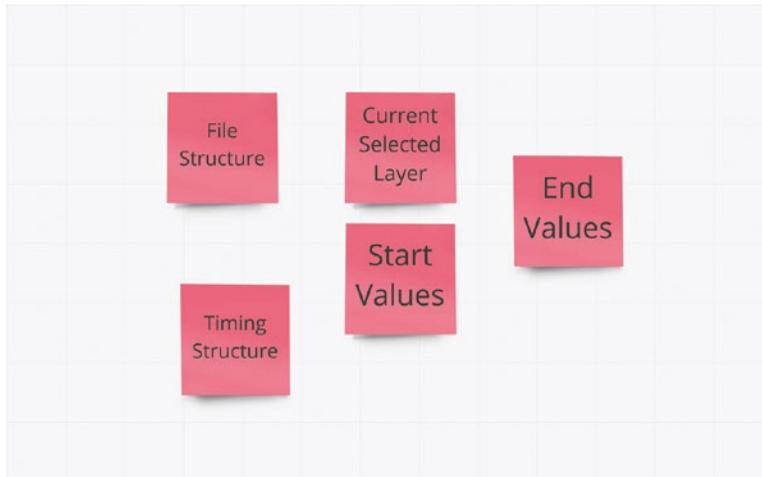


Figure 8-2. *The broad data that our animation tool might need when creating an animation*

Once you're happy that your broad level data is plotted, you can start breaking each one down—if you feel it necessary—into a more granular structure. Again, try not to get too bogged down by getting things “right.” Just focus on getting things *down*. This exercise is as much about *you* understanding the system as it is about the end result. Figure 8-3 shows this broken down in more detail.

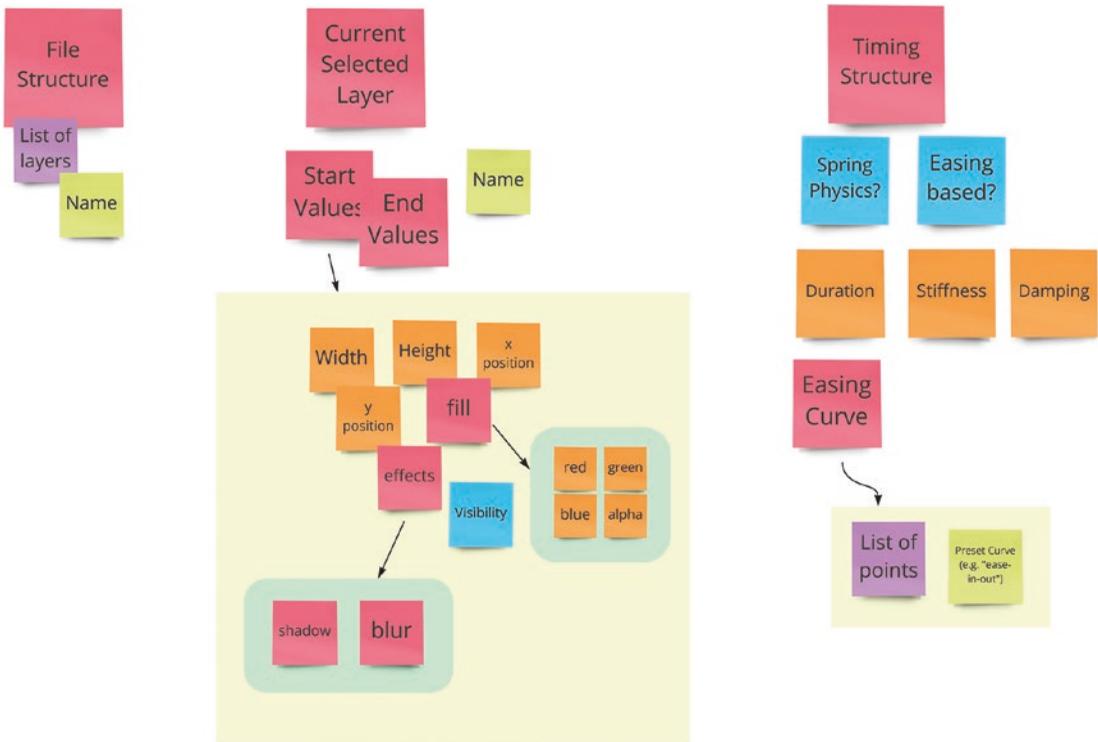


Figure 8-3. A more-detailed state diagram

If you're eagle-eyed, you've probably noticed that I've used *different-colored* Post-its. This is serious design work now. These represent different variable types; pink is an *object*, purple a *list*, green a *string*, orange a *number*, and blue a *boolean*. If you're reading this on an e-reader with a black-and-white display, you can access these diagrams at any time in this chapter's folder in the Resources repository on a color device. If you're color-blind or work with people who are, try to avoid using problematic colors or consider getting sticky notes of differing shapes.

As well as codifying the variable type, there's a level of hierarchy. Our "fill" variable, for example, is made up of four numbers (red, green, blue, and alpha).

I must stress, it's okay to be messy here, and it's *totally fine* to be a little wrong. Think of this almost as a mind map for a non-sentient *thing*.

This is a state diagram I would be happy to work with, and here's why: it lets me know the kind of data I need to design for and how it will likely need to be grouped. If I know that a layer's start width, for example, is a number, then I'm already able to run

some sort of mental filter on it. Experience tells me that numbers are often controlled by sliders, but also by number inputs with increment and decrement controls. Observation research showed that people expect to be able to type directly into an input or to change a number input in multiples of 10 by holding down shift and pressing an arrow key. Seeing state mapped out in this way lets us apply the lenses of mental models and convention to how we might want to display and manipulate this data.

This leads us to the real utility of state diagrams. They expose us quickly to the types of interactions and display patterns we might need to account for. As we discussed in Part 1 of this book, convention plays a huge part in design. State diagrams let us know not only the type of data we'll be exposing, but also the *context* we'll be exposing it in. "Someone is editing a number field" is useful in that it gives us the type of data, but the notion of "someone is editing a number field that represents the red value in an RGBA color format" gives us the context and relationship that we *really* need.

A state diagram gives us a handy reference point for the type of data we'll be working with and it goes hand-in-hand with the next piece of documentation: mental models.

Mental Models

Mental models have been a recurring theme throughout this book and with valid reason. They're the lens through which a person will perceive our work. Mental models tell us what someone *believes* about a system, not necessarily the system's "real" workings. Mental models distort reality. They're rife with heuristics and prediction errors, loaded with biases, and subject to random shifts in logic and structure at any given moment. Just like the human minds of which they are products.

The search for a conceptual model that accounts for all the idiosyncrasies and shifting emotional frames of reference of every mental model we encounter is the alluring folly of design. Just as it's impossible to create a logo that is universally appealing to everyone who encounters it, so too is it impossible to produce an interface that perfectly resonates with the brainwaves of every human that interacts with it. So, just like the mind, we deal in statistical averages. Prototypical mental models that give us insight into the expectations and motivations of the people who form them are similar to user personas. They're a usable abstraction of "real" mental models. So, we create an abstraction of the common mental models we see when we investigate our problem space. Mental model models. Which is definitely not confusing.

Abstracting Mental Models

Mental models should tell us two broad things; first, what people expect to be able to do with the kind of product we're making, and second, what conventions and habits exist that frame these expectations. We can then use these models to decide which patterns and conventions are compatible with our system and which we might want to deviate from.

As we work with mental models, we'll invariably encounter a point where we need to assess the scope of our work. A person's mental model of a given scenario is affected in unpredictable and infinitesimal ways by a never-ending list of external and internal forces. Documenting such a thing is frankly impossible. So we abstract up a little. For our practical reasons, a mental model becomes, essentially, a list of heuristics, hypotheses, and expectations that someone brings into any session with our products.

There are currently a number of common practices for documenting mental models. Indi Young's 2008 book, *Mental Models: Aligning Design Strategy with Human Behavior*, is a great starting point that discusses how mental models have historically permeated through business and design decisions. However, I'm going to go out on a limb and say that I think the vast majority of current implementations just aren't compatible with the approach to design I'm proposing in this book. Traditional mental models are commonly broken down into "steps" and—I think you know where I'm going with this—generally only accompany a linear path through decision-making.

With our focus on systemic thinking and eschewing the idea of linear-by-default, I think it's much better to approach mental models in a more holistic way. The mental models we produce should be a somewhat thorough representation of what people *bring* to our interface. This includes their preconceived notions of what they *think* a product might help them achieve, their past experiences with products in similar problem spaces, their real-world environments, and their potential motivations. A good mental model (and we'll want to create several for each project—one for each persona) will work alongside personas and empathy maps to help inform *how* and *why* we translate our system the way we do. If design is a conversation, mental models tell us the metaphors we can use, the frames of reference for our discussions, and the cultural conventions that we need to be aware of, lest we offend, confuse, or misrepresent.

To draw up a mental model, we need solid research results. Again, observation research and ethnographic interviews are key in this regard. If we're able to watch and potentially interview people in their usual working environments, we're able to form well-rounded mental models that cover these bases.

When we create our mental models, we're looking for recurring themes that inform or display any of the following:

- Expectations and beliefs
- Approaches and habits
- Conventions

These three categories, similar to the 2x2 grid for taking notes from Chapter 7, will provide invaluable insight into every part of the design process. *Expectations* and *beliefs* tell us any pre-existing ideas that people have about the tools and processes in our problem space. *Approaches* and *habits* let us know how people currently get stuff done, any rituals or idiosyncrasies they might perform, and the ingrained processes they might follow. *Conventions* tell us what the various tools in our problem space have in common.

As you observe or watch recorded observations, make a note any time you spot one of these things. At first, you might struggle to recognize them, but over time you'll be able to document this data in close to real time. Like before, let's take a look at a complete example and dissect from there. Figure 8-4 shows a completed mental model of what a designer might bring in using our animation tool.

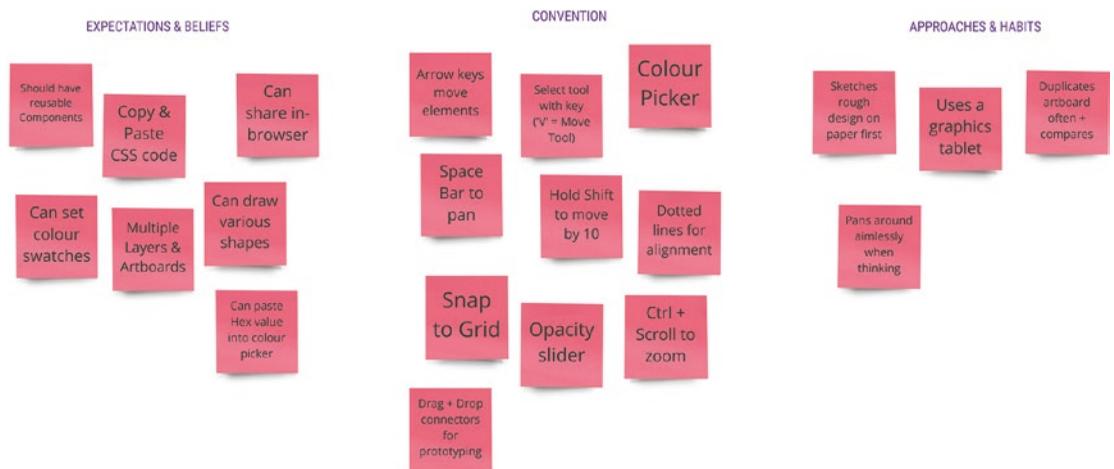


Figure 8-4. An example mental model

This data will likely come from different sources. It's difficult to understand what people's expectations of *your* product are if you're asking them off the back of a competitor research session, for example. Given this, it's quite likely that you'll be filling in these mental models throughout the early stages of design.

Approaches, habits, and hacks can all be documented early. Again,, observation research and interviews are your go-to here.. Ensuring you collect this data in as “real” an environment as possible is key.

Conventions are often the result of competitor analysis, and research and can be performed at any time. Simply get hold of as many products in the same problem space as yours and jot down their similarities. Watching others use these tools can often be extremely eye-opening, so make sure you’re not just limiting your research to your own use of the products.

Finally, expectations and beliefs will often get filled in later in our process. It’s hard to decipher someone’s expectations for a product that doesn’t exist, and their beliefs are something that we’ll have to delicately tease out when we conduct more in-depth interviews. This will be useful once you start testing prototypes. Many usability issues arise from expectations not being met, so any time you see someone struggling with a specific part of an interface, try to find the underlying conflict between expectation and reality. Is someone trying to press a disabled button? Trying to click-and-drag to draw with a “pen” tool? Hitting a submit button multiple times because they’re not given any feedback? Explore how these might be manifestations of mental model gaps—even the obvious fixes (for example, our disabled state doesn’t look “disabled enough”) will be rooted in some form of convention or expectation.

Mental models give us a contextual reference point for *how* we might communicate our system’s state and possibilities. If design, in part, acts as a translation of system language into a form of visual (or otherwise representative) language, our mental models let us know how best to pick our “words.” In our case, our state diagram from earlier had a “fill” variable that was an object that consisted of “red,” “green,” “blue,” and “alpha” number variables. Our mental model shows us that a color picker is a convention found across every other design tool we tested. Our competitor testing shows that people used this kind of tool often. With all of that information in hand, we’d very likely want to explore the idea of having a color picker to allow for the changing of these variables.

While this may seem a rather obvious conclusion to draw, it serves as an example of the kind of thinking that reconciling the expectations and conventions of our mental models against the cold data of state diagrams.

The Ubiquity of Mental Models

It's worth pointing out that while we're actively concerned with the mental models people may form of *our* products, people form mental models for pretty much *everything* in their lives. They have mental models for how they get dressed, for checking social media, for how their TV works, for playing the guitar, for crossing a busy road, and for riding a bike. Mental models are abstractions and heuristics that inform what we *believe* about a product or system.

In Part 1, I gave the example of an operating system using folders and files as a metaphor for a system storage. This is a long-standing mental model, to the point where a folder icon is almost as ingrained in our cultural conventions as “red means stop.” This abstraction frees people of the weight of understanding a technology or an idea. By presenting people with a nested folder structure as opposed to burdening them with somehow understanding of the storage idiosyncrasies of computers or servers, we're able to abstract away the complexities of a system.

It's important, then, to understand some of the common mental models people have, even those that you wouldn't necessarily see as immediately useful for your specific project. For example, let's say that we wanted to let designers save their animations for later use. By understanding the mental model of folders, as an abstraction of *showing a potentially nested and navigable item structure*, we're able to discuss or validate it as a potential solution for displaying stored animations.

Once you understand that mental models exist, you'll notice them everywhere. Any time you see a complex system abstracted out into metaphor or simplified through removal, you're witnessing a mental model in action. If you've ever presented someone with a prototype and watched on as they've ham-fistedly tried to tap on 14 things they think are interactive, but are just dummy content or, frankly, poorly designed static elements, then you've witnessed a disparity in mental models.

Ever see a toddler raised on tablets and smartphones try to tap and swipe a physical book? That's a clash of mental models. The product does not match the model that our high-tech infant has brought to the experience. To those of us who were raised on good old-fashioned real paper books that smelled yellow and had genitals drawn in the margins, we might see this interaction as a harbinger of doom, of technology's tightening grasp on the minds of our young. This is, however, a simple prediction error of sorts. Very quickly, our toddler friend might start thumbing through the pages, responding to the images, and making their own stories and inferences from the pictures found within. The realization that the book did not respond to gestural input (I swipe my iPad, why can't I

swipe this?) is usually a pretty quick one. Often, there's an adult there to show them how books work. Suddenly, a new model of *how a physical book works* starts to form.

What I want to stress, is that, just like how our brains shape our realities—how we're able to see the river, the bridge and the birds in the *nonfinito* style of Monet's *Impression, Soleil Levant*—so do our mental models shape the reality of a system. If we *think* we can do something a certain way or if we *expect* that a certain element will respond in a certain way to a certain interaction, then until we discover otherwise, that *is* the reality of a system to us. When our mental models clash with a designer's conceptual model, we're experiencing something new. Whether this *newness* is novel, confusing, disappointing, or just an outright inconvenience depends on the gap between our models and, fundamentally, the efficacy of the new interaction.

Gestural interactions and touch devices were quickly adopted once people started realizing the benefits that gestural interaction had over analogue input devices. While these devices challenged mental models and many people were left confused, they survived through the early days of cumbersome interactions and unrefined models to the point where these devices are ubiquitous.

As we discussed in Chapter 3, when talking about learning, we can treat a gap in models as a chance to educate and encourage adoption of the concept we're presenting. The key thing here is *knowing* that what we're presenting might not be compatible with the models that others will bring to it and figuring out how we bridge that gap. While our work relies heavily on the conventions and metaphors that have proliferated through digital interactions, if we believe our conceptual model is better, we can teach it and provide a path for people to integrate *our* concept into their mental model going forward.

The key idea with mental models is that they document *current* expectations. While the need to have these plainly stated is important, we should never feel pressured to follow them. But having them documented gives us the confidence to challenge them. Similarly, it rules out the dreaded *just because* moments in design. If we want to present an alternative to a traditional color picker, for example, then we should be able to discuss *why* we're deviating from a common paradigm. A huge part of this reasoning will come from an analysis of the current models and conventions.

By documenting mental models, you're not just creating a big list of things that your product should do. You're creating a list of things that you have to *justify* moving away from if you decide to do so.

On Flipping Mental Models

As well as understanding the common mental models people may bring into our products, we should also be aware of the potential that lies in “flipping” mental models—that is, re-framing an established concept within a new mental model. One of the best examples I’ve seen of this approach is Google Inbox (Figure 8-5) taking the existing concept of an e-mail inbox and designing around the mental models of a to-do application.

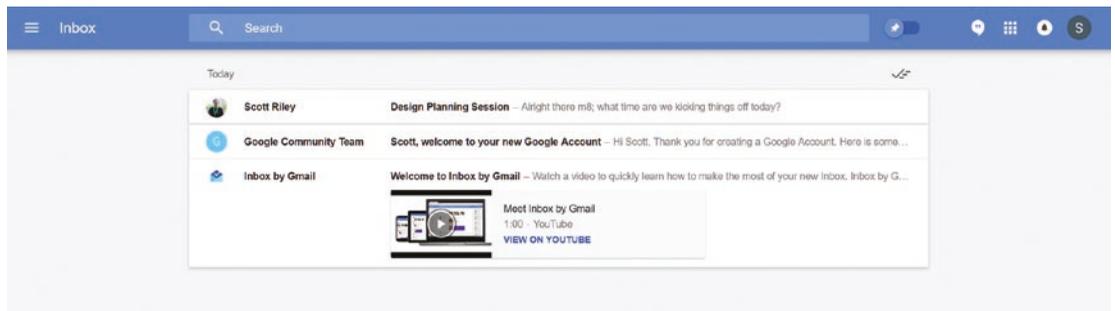


Figure 8-5. Google Inbox’s “flipping” of mental models

Inbox takes the tried and tested mental model of an e-mail inbox and, well, throws it in the bin. Back in the halcyon days of early e-mail, your inbox behaved like a physical mailbox. You would receive the occasional bit of jovial correspondence from someone in your contacts, mull it over with a cup of coffee, perhaps have a little chuckle at the typo in the sixth paragraph, make a decision as to whether to reply or not, and then go about your day, excited to return home and check if your good friend had replied to your hilarious retort or if your great-aunt had sent you another hilarious picture of a squirrel in mittens. E-mail was personal. The visual and conceptual language played to that. Inboxes, for the longest of times, were predominantly made up of a list of e-mails, with unread e-mails highlighted and a separate panel for the content of your currently selected email.

Then social media happened. Smartphones happened. Company mailing lists happened. Online shopping, online dating, online reading, online *everything* happened. People had accounts for *everything*, and *everything* was sending e-mails. Fast-forward to the modern day and e-mail is far removed from its initial incarnation. E-mails are no longer these happy pings you got once or twice a day. They’re an avalanche of spam

and notifications and *noise*, and they represent an ever-growing list of anxiety-inducing reasons to throw your phone in the sea.

E-mail is no longer a reading exercise. It's a painful chore and a reminder that every single thing on the Internet wants all of our attention immediately. What do we do with painful chores? Well, if you're me, you let them pile up continuously for weeks on end before you have a mental breakdown and clean your entire house while listening to Enya and sobbing. But if you're like many people, you'll make a to-do list and push through them, marking items off as you go. Inbox decided, on our behalf, that e-mails are now better seen as to-do items, and it's played out beautifully.

The to-do mental model is evident throughout Inbox's approach. Gone is the endless, uncategorized list of e-mails; gone is the focus on reading full content and replying directly. Inbox presents us with a bundled, categorized list of e-mails, and the primary action is that of "marking as done." The visual language—"check" icons for marking as done, "clock" icons for snoozing till later, and an "all done" empty state that celebrates getting through your list—is that of a to-do application.

Inbox's underlying system is still that of a modern-day e-mail provider app—one can still view their e-mails in Google's "standard" Gmail app—but the features presented and the language used to present them are that of an entirely different model.

The design brief for Inbox might have revolved around, as a crude example, a mission to *make e-mails easier to get through*. The problem space of "it's difficult to get through a big list of tasks" is one that is going to naturally be littered with to-do and "get-things-done" applications, to which people will naturally bring their own models, conventions, metaphors, and approaches. By dissecting the mental models people bring to these products instead of focusing on the tired mailbox metaphor, the Inbox team was able to ship an e-mail concept far more suited to the demands and load modern e-mail paradigms place on the human mind.

Try and jot down a mental model for some common experiences that aren't really related to your product. Education is a good one, as people of varying ages, backgrounds, and cultures have very different beliefs on what a learning experience will, or should, be. Typical mental models for learning might be made up of things like "grades that show how well I've done," "centralized syllabi for what I should learn," "semi-frequent exams as a means of proving knowledge," and "a strict teacher who takes no crap."

As an exercise, take one of these concepts and modify or remove it. Let's say we wanted to remove grades from education. How would we do that? If we simply said, "Okay, no grades," then we remove the classic means of understanding our progress and

portraying our mastery. So perhaps we'd want to swap it with something else. We could take a cue from Duolingo, where grades are replaced with EXP points.

In fact, Duolingo, which we explored in Chapter 5, is a great example of a modulated mental model. While fundamentally existing as an educational product and relying quite heavily on the ideas of mostly linear syllabi and frequent demonstrations of learned concepts, Duolingo asks us to apply the mental model of a game to its progression model. I can think of many reasons to want to remove grades from education, but Duolingo realized that if it were going to do that, it'd need *something* in place to track progress.

In fact, *gamification* is, at least in part, the act of replacing one mental model—the traditional model of what you're trying to gamify—with that of games. The EXP metaphor is straight out of tabletop RPGs and has carried over into tabletop and video games of all permutations. By exploring the models that make up modern-day gaming and figuring out how to replace or augment an existing model with these findings, we can start to see gamification as less of a buzz-wordy catch-all for trophies and badges, and more as an intrinsic model that brings its own convention, hypotheses, and biases with it. It also makes it a *lot* easier to spot areas where gamification is being attempted *just because*.

All of this is to say that—even if you're dealing in a problem space that has truly ingrained and obvious metaphors and mental models—defying expectations, reframing your problems, and applying the conventions of a different model to areas of your work can be a genuinely impactful practice. The exercise of writing down as many variations of “what if [e-mail] was [a to-do list]” as you can think of is, unto itself, a beneficial creative exercise. And it just might end with an innovation you never expected.

Empathy Maps

Empathy maps are, in addition to user personas and mental models, the final piece in the rather fuzzy puzzle of *things that distill people down into useful Post-it diagrams*. Personas provide us with a brief insight into a fake person's life, interests, and technology preferences; mental models give us an overview of a fake person's thought processes, heuristics, and hypotheses; and, finally, empathy maps let us take our fake people, put them into fake real situations, and draw some conclusions about what they might say, think, hear, and do.

Just like personas, empathy maps are a means of documenting the ever-important ethnographic research we do. My personal rule is to create at least one empathy map for each persona created, making sure that the situations we put them in are as much of an aggregate of our findings as the ethnographic and pseudo-personal information we use. If a bunch of our research participants worked from busy coworking spaces, for example, then we'd make sure some of our empathy maps "happen" there.

To me, an empathy map is all about simulating a persona's response to a specific concept in a specific environment. So, take one of your personas, think of a common environment they might work in, and start mapping.

A traditional empathy map will be split into four quadrants: what our protohuman *says, thinks, does, and feels*. Over time, new formats for empathy maps have emerged, with *think and feel* and *say and do* being merged into single categories, and *hear* and *see* being added. The secret here is this categorization almost doesn't matter. The quadrant approach is an extremely useful visualization technique, and the categorization it offers is equally as helpful, but a little pragmatism goes a long way in design documentation. The main goal when preparing for your mapping session is to think of some categories that you feel are most useful to your current project.

Personally, I really like using *say, think and feel, tasks and actions, hear and see* as starting points because they represent the building blocks of how our story might unfold. Importantly, though, I'll add three extra sections to a map: *motivators, distractions, and pain points*, as well as an area to jot down the high-level goals this person might have. Other maps might add *pain* and *gain* sections while others might break out *habit and routine* into its own section—especially if the designers are mapping out personas for a product that digitizes a previously analogue process.

When I look at an empathy map, I want to be able to imagine a story unfold from it. Regardless of what sections and categories are used, regardless of whether the map is made by one designer or a 16-person, half-day Post-it bonanza, the key factor is that it sets up the parameters for a story: our imaginary person finds themselves in an imaginary place and has an imaginary conversation with our product.

Let's take a look at an empathy map and work backward from there. To start with, we'll need a persona to work with, so, for brevity sake, let's say our persona is Alex, a designer who mostly works from coffee shops on a MacBook. A full persona of Alex would show a much more in-depth insight into their preferences and the challenges they face, so flip back a chapter if you need a refresher. Let's take Alex, plunk them in a busy coffee shop, and draw up an empathy map (Figure 8-6).

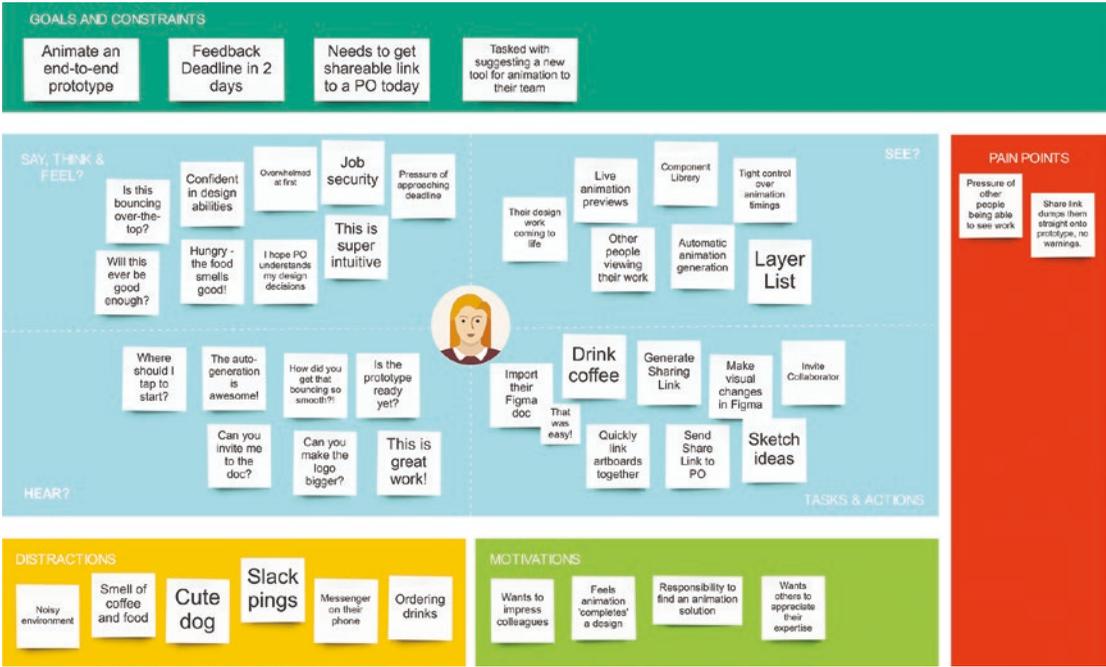


Figure 8-6. An empathy map for an example persona, showing their various perceptual responses, distractions, motivators, and pain points

This approach to mapping does a number of things. First, it sets the scene: Alex is in their local coffee shop, they have a tight deadline, and they need to get something over to a PO today. They’ve also been tasked with proposing an animation solution to the rest of their team.

Our main four quadrants give us insight into much of the surrounding events and stimuli that accompany this situation. Actions such as “drink coffee” sit alongside “send share link to PO”; feelings like hunger or coffee cravings are in the same space as their feelings on our product, as well as their intrinsic, self-deterministic thoughts. Empathy maps should paint a wide-reaching picture of the stimuli, mental processes, actions, and reflections that might occur when someone sets out to achieve a goal within our system.

It’s especially important here to consider environmental factors alongside internal thought processes and product-specific features. Empathy maps should be as much about what you’re competing with—and what expectations you need to live up to—as they are about your product’s appeal or the efficacy of its features. Nevertheless, it’s totally fine—and encouraged—to be optimistic with some of your Post-its. If you believe your product is going to make someone say “Wow, this is so fast!” then by all means, add

it in there! An empathy map should present a balanced range of responses, including what we think will excite people.

Distractions, in my opinion, are a must for any empathy map. They show, at a glance, the various factors of an environment that our product is competing against for attention. This lets us, hopefully, see our work as something that exists as part of an extensive and diverse ecosystem of priorities and desires. In terms of a busy coffee shop, an open-plan office, or a busy coworking space, we're going to be faced with myriad distractions. From on-screen notification pop-ups, to someone else's phone with the same notification sound as ours, to the smell of that one dude who refuses to stop microwaving fish—these all have the potential to drag us away from what we're doing. And I don't care how great you or your boss or your client *thinks* your product is going to be, I guarantee it won't stand up to the momentous occasion of meeting an adorable new office dog. Distractions, perhaps most importantly, also let us get a feel for how much cognitive load we'll *really* have to work with. Going back to Chapter 1, you'll recall that so much of our thought processes and day-to-day interactions are rife with distraction and daydreaming. In ensuring our map's environments account for this day-to-day attentional undulation, we're able to keep our ideas rooted in the chaos and unpredictability of the real world.

Finally, in making room to discuss motivations, we're able to see a more rounded representation of the intrinsic notions people might bring to our product. While goals tend to focus on the utility and final outcome of our product, motivations are self-deterministic. A *goal* of Instagram, for example, might be "upload my latest travel photo," but motivations speak for the emotions and desires *behind* this goal. Perhaps we upload it to impress our friends with how much we travel, or we want to keep it in an easily accessible place for reminiscing. Maybe we're more concerned with the artistic qualities of the photograph and we're uploading it to show off our photography and editing skills. Or perhaps we just think it should exist in the world and we're excited for other people to look at it. Conflating intrinsic motivations with task-centric goals often leads to a limited and insular understanding of an audience, where their supposed motivations read like a salesperson's checklist rather than the complex and diverse desires, motivators, and notions of self we harbor as human beings.

Remember, too, that you're not here to sugarcoat any potentially problematic or toxic attitudes that might be prevalent in your audience. There's no reason why your empathy maps shouldn't include selfishness, neuroticism, or narcissism. In our case,

our audience is all designers, so, *of course*, there's none of that. However, most real people have flaws, heuristics, and biases. If your research shows that certain parts of your demographic are prone to shouting, "I am a design god," at spontaneous moments during the workday, then throw it in the "say, think, and feel" quadrant. The same goes for displays of self-deprecation, impostor syndrome, workplace anxiety, emotional triggers and the myriad other negative mental phenomena we navigate on a daily basis.

Putting It All Together

Documenting our systems involves documenting our audience. For me, the combination of personas, mental models, and empathy maps serve this purpose without being overly intensive to create or overly obtuse to share and understand at a glance. Coupling this ethnographic documentation with a core state diagram lets us see the cold hard data of our system's inner workings against a diverse backdrop of convention, emotion, distraction, motivation, expectation, and belief.

If you're strapped for time or money when it comes to documenting and you only have time for one of these tasks, go with empathy maps. There's often a lot of overlap between mental models and empathy maps—especially with regards to habits and biases—and they can often completely replace personas in the right circumstances and with the right team structure. If you *do* skip on persona creation and mental models, I would suggest taking at least an hour to dissect the conventions in your problem space. It's probably the most valuable bit of mental model documentation and can be done in a pinch by simply exploring every tool in your problem space you can get your hands on. You'll never beat watching people with various levels of expertise use their products of choice, but you'll at least be able to pick up the primary similarities (a sure sign of convention) between these products.

The goal of documenting a system in this way is to understand the raw information and functionality that you'll be working with from as many angles as possible. Mental models let us know what conventions, expectations, and beliefs might affect how our system is translated. Personas give us some initial demographic separation and provide a challenge to our initial assumptions of how our system might be communicated. Finally, empathy maps let us theorize how our product might be perceived and used—and what the underlying goals and motivators for this use might be—in a real-world situation, as well as the potential distractions of competing stimuli.

Later in our projects (and later in this book), we'll want to come back to system documentation in the form of artifacts like pattern libraries, design systems, and design principles ,but before then we'll "zoom in" a little and start thinking about our moment-to-moment flows and granular interactions. In Chapter 9, we'll start looking at how we can take this understanding of our systems and audiences and put together flows and interactions that are just as rooted in the idiosyncrasies and diversity of the human mind.

CHAPTER 9

The Anatomy of an Interaction

We have spent a good while extoling the virtues of systematic approaches to design by diving into an exploration of the more granular, moment-to-moment interactions that form the building blocks of our interfaces feels opportune.

While I believe that adopting a systematic mindset and that finding ways to communicate that are referential to overarching systems are increasingly important skills for designers to practice, there's no denying that a huge part of our work—and our value—lies in the finer details. Even through the definitely cynical lens that a comparatively long and eclectic career in this industry has gifted me, there's still very little I enjoy more than applying the finishing touches to a tiny interaction.

As the influence of technology—and by proxy design—grows and pervades into more areas of our lives, so too does the weight of the responsibility that we assume. Being responsible for large, impactful decisions, especially those that indubitably affect other humans, can be a source of malaise and anxiety for all of us. And, while this book espouses heavily the need to be cognizant and proactive about the impact of our work, ours is also a craft rife with intrinsic reward and avenues of escapism. For me, this is never more evident than when I'm designing these smaller moments, attempting to imbue taps and clicks and swipes with the character and responsiveness that can often be the infinitesimal differentiator between a moment that *feels right* and one that leaves us nonplussed, confused, or just outright frustrated.

This chapter will focus on the more granular side of the design stack—the processes that usually come after we've done enough research, documentation, wireframing, and prototyping to progress on with implementation. Of course, no design process is a truly linear activity; discoveries in the latter stages can *always* force us to revisit the phases we previously deem “done.” There *does*, however, appear to be—in every process I've tried, observed, and implemented—a moment.

More specifically, *that moment*.

That moment is when we, our team and our stakeholders, realize that we've discovered our broad solution. We've defined our problems, explored the problem space, and researched our audience and competitors. We've produced countless Post-it-based diagrams, scribbled dozens of potential wireframes, prototyped our lives away, revisited our problem definition 14 times, played with Post-its again, created a CSI-style wall of incomprehensible linked-together notes and personas and brainstormers, and, suddenly, we take a step back, and realize we've *got it*. What *it* is, at least in a tangible form, is as varied as our processes, projects, and people—but what *it* represents is the shift from *problem definition* to *solution implementation*.

This shift is probably the most obvious change in mentality and approach that we'll see in a project. The zoomed-out, disposable, and low-fidelity work we've been producing is catalogued and documented, and we switch into *implementation mode*. This is where we get to do what visual and UI designers will tell you is the *real design work* and what UX-only designers, front-end developers, your CEO, and most of your family will tell you is *coloring in*. (I say this mostly in jest, but the amount of times this moment is seen as a strict baton pass from one team to another, especially in large companies, is a genuine concern—and partly why this book assumes you're at least interested in most sides of a design role.)

So, we're ready to go. We've got *it*. We know what we're going to build and the problems it's going to solve. We have rough wireframes and a bunch of research data to back the project. It's time to zoom in to 1,200% and start tweaking button gradients, right? Not quite. Naturally, being the spoilsport that I am, I want to take one last step back, and—I hope you're prepared for this—break out more Post- its.

Thinking in Moments

As Chapter 7 alluded to, the zoomed-in companion to our zoomed-out system thinking lies in considering the *moments* that might arise in our interfaces. If our “environments” represent the overarching qualities of our system, communicate what is possible, frame interactions, and generally act as an open and explorative means of discovery or expression, then our *moments* represent the interactions themselves. They act as the set pieces of our work.

It's worth stating outright that your interface might quite easily just be seen as *one* moment. A good portion of this book has been spent encouraging and exploring the

value in open and copious digital environments, but forced ambiguity and contrived expansiveness are as damaging to a limited-scope product as peremptory defaulting to constrained, linear flows is to a wider-ranged product. If you're working on a product that focuses on making one single action as efficient or enjoyable as possible, you might see your work as one, tightly defined moment. For most interfaces, however, we'll be dealing with some kind of balance and undulation between system focus and moment-to-moment focus.

Before we progress into documenting and planning these moments, I want to attempt to elucidate what I mean when I refer to a moment of an interface.

If we view our *environment*—the “default” scope of our interface—as an explorable area that communicates many (or all) of the actions that can possibly be performed at any given time, a *moment* is the shift from communicating the possibility of action to the purposeful performing of it. A moment, as a response to implied intent, shifts the focus of our interface from one of communication and exploration to that of task positivity and focus.

Finally, moments can themselves be treated as systematic, essentially being made up of smaller “sub-moments.” In a sense, they're the components of interaction design. Just as in a pattern library, a larger “modal” component might be made up of an “overlay,” a “panel,” and a “button” component, so too might an “onboarding” moment be made up of “check username availability” and “avatar upload” moments. If you're reading this and it sounds like I've just given “user flows” an unintuitive new name, then, well, you're almost right!

For me, moments exist as discrete, end-to-end combinations of interactions. You could quite easily take a standard user flow approach and document a moment with it, and, in many instances I would encourage doing so. A moment is a concept—it's a way of saying “this particular combination of interactions represents an important point in the life cycle of our product's use.” User flows are documentation, and they represent only one way we can document our moments.

Conducting an inventory of sorts for the moments you expect to appear throughout your interface can be an extremely valuable task, especially as a means of bridging the gap between the more explorative phases of your process and the more practical implementation-focused phases. Your moment list will provide a high-level overview of all the key set pieces in your product and can go on to inform many of your most important design decisions.

Finding Your Moments

Moments are the set pieces of a system. Think of them like the key plot points or action scenes of a story. Forming a list of these moments is, generally, a pretty straightforward task. Think of the places where you feel as though your product will stand out. Your fundamental calls to action, times where a feature solves a particularly widespread or painful problem, and actions that guide people through a tricky path are all candidates for the “moment treatment.”

If we take Instagram as an example, some of the key moments might be as follows:

- Creating your account
- Following your first X accounts
- Finding your friends
- Landing on your timeline for the first time
- Uploading a post
- Uploading your *first* post
- Creating a story
- Creating your *first* story

While Instagram is generally a somewhat explorative application (in that, most people browse Instagram more than they post to it), its potential moment list is still pretty long.

You’ll notice that performing actions for the *first* time are listed as separate moments to the standard moments. It’s quite common to see apps hand-hold us through our first forays into key moments—something that we explored in some detail in Chapter 3 when we discussed in-app guidance and onboarding. I’d recommend doing similarly wherever possible—especially for your complex moments. Treating the first time someone enters a particularly key moment separately from its subsequent occurrences lets us consider how we might want to guide someone through a learning experience, for example, or how we might want to go a little overboard celebrating the success of the action. While uploading an image to Instagram, for most of us, is likely not much of a learning experience, the *first time* we do so absolutely is.

As we also discussed in Chapter 3, in-app guidance or constrained onboarding flows often serve to overly limit a moment. While we’d eventually want people to begin expressing themselves through editing their photographs or telling their stories—to ditch

the training wheels, so to speak—until they’re comfortable doing so, the leading hand of in-app guidance can be a valuable ally. By separating first-time moments from their repetitive counterparts, we’re not only able to see the moments that might benefit from such guidance more easily, but we might also find ourselves designing much more open, expressive ways of performing the non-first moments.

Your idea of what constitutes a moment in your product is absolutely open to interpretation. Your key set pieces are going to vary depending on your problem space, your audience, and how your company or client wishes to monetize their service. However, most moments display some—and often all—of the following traits.

Moments Tell a Story

By deviating from a default explorative interface to a more specific or more constrained set of actions, moments essentially have their own start and end, points of friction, moments of resolution, and consistent thread throughout. While our default interface views may well be sprawling, open, and full of possibilities, the moments within them are where these possibilities come to fruition.

Moments Make Us Money

Whether literally through being a checkout flow or payment process, or figuratively due to their role as a key feature and selling point of our systems, moments often generate revenue. If a flow ends in money in a bank account or the achievement of a feature that you list on your marketing page, it’s probably a key moment.

Moments Constrain Our Interface

Moments will usually limit our interface to a subset of possible actions or contort it temporarily into an entirely new set of possible actions.

Now, this list could describe almost any interaction in any interface. A single button click could easily meet all these requirements and, realistically, micro-interactions *are* moments—they just might not be the moments you want to prioritize. You’ll have to apply your own scope and granularity to forming this list. Larger teams and limited products will likely have more granular moment lists; smaller teams and more expansive products will likely result in broader, zoomed-out lists.

Once you’ve defined the parameters for what constitutes a moment in your interface, you can list them. Whether you type this list in a notes app, throw it on a digitized Post-it

board, or write it in six-foot letters on a wall is totally up to you. The only thing to avoid is projecting linearity into your list.

It's *very* easy to get caught in the trap of assuming certain moments will be performed conveniently in sequence—someone will create their account, *then* add their profile image, *then* follow their friends, *then* view their timeline, and so on. Unfortunately, this happens far less often than we would hope. For this reason alone, I like to write my moments down on Post-its and jumble them up occasionally.

This process should not take too long—I've never had it take longer than 30 minutes and most sessions are done within 10—and it's often a case of documenting what's already in a design brief or a client's head. For products that encroach more into service design, where you'll generally be working with multiple interlinked products and sub-services, it might make more sense to draw up a list of moments for each of these and then consider moments of transition as well.

Your moment list should be a simple reminder of your product's key set pieces and should complement some of the broader, system-level documentation we'll amass through our process. Figure 9-1 shows a simple example of a Post-it moment list.

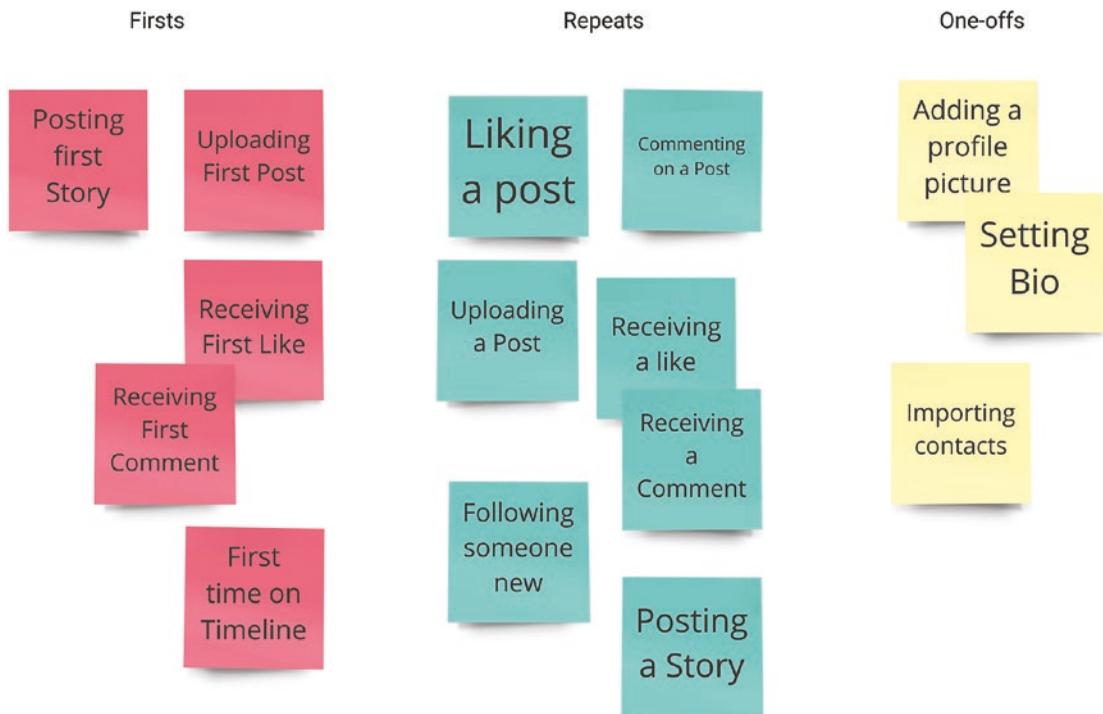


Figure 9-1. A subset of what we might see as Instagram's key moments

Finally, I'd highly recommend finding criteria against which to prioritize your moments. Again, as every project and process has its own goals and idiosyncrasies, it would be remiss of me to mandate these parameters, but the following questions are good starting points:

- *How integral to the core product is this moment?* Uploading an image on Instagram, for example, is critical for the product. No images means no product. Adding a bio or a profile picture would be lower on the list.
- *How integral to your audience's motivations is this moment?* Creating something new in a creative application and displays of mastery or skill while learning are obvious candidates for key moments. If you understand these desires and motivations through ethnographic research—and they're not used as a crutch or with malicious intent—then moments that allow for, and celebrate, the realization of these can be extremely powerful.
- *How much does this moment contribute to the overall expectations of our product?* Through your marketing site, word of mouth, social networking, sales pitches, and any other medium that revolves around selling your product and its features, people will start to build up a whole host of expectations. Moments that might seem somewhat innocuous at first can, in hindsight, be critical moments in realizing these expectations.
- *How much does this moment deviate from common convention or your audience's mental models?* Moments that involve high degrees of learnability—or esoterica—are likely to be key moments by virtue of them still existing in your interface. In Chapter 3, we discussed drawing a feature complexity diagram like the one shown in Figure 9-2. Through the process of drawing this diagram, we should already have a solid idea of the unintuitive features of our interfaces that we need to teach.



Figure 9-2. A filled-in feature complexity map. Features that are important but complex will likely represent key moments.

The purpose of listing our moments is not to provide an exhaustive feature list or a widespread justification for our product—it’s important not to overthink things. Many times I’ve spent too long on such an exercise and the whole point—to produce a succinct list of the most impactful moments—is lost, replaced instead by a tangled mess of smaller features that slowly become impossible to prioritize. When in doubt, think in broad strokes.

Once we’ve gotten our Goldilocks-style list of not-too-vague-not-too-granular moments, we can look to categorizing the items.

If you look back at Figure 9-1, you’ll notice I’ve used the categories “Firsts”, “Repeats” and “One-offs.” “Firsts” represent the moments that we want to use to highlight

someone’s first performance—“Firsts” are especially important for features that might need a certain degree of guidance or repeated features that are integral to our core offering that we’d want to celebrate and offer positive reinforcement for. “Repeats” are actions that we’d expect many people to perform many times. In the case of our rough Instagram-style moment list, this category will include actions such as liking other people’s posts, posting to our story, and posting for ourselves. Generally, these recurring actions will often represent the core loop of system usage. Finally, “One-offs” represent actions that will likely only be performed at certain points, or at least very rarely. While many of these actions might well be key moments, they’re simply occurrences that just don’t happen regularly enough to be seen as part of a core loop or common usage. Actions such as adding a profile image, updating payment details, changing your bio, and updating an e-mail address are all candidates for important moments that don’t occur regularly.

Once we’ve got our moment list together, prioritized, and categorized them, we can finally document and design them.

Documenting Your Moments

If you’re a seasoned designer—or if you’re studying design from an angle that involves UX, CX, or any other acronym the industry has invented that ends in an “X” since the time of writing, you’ve probably encountered dozens of methods of documenting a flow. Some people swear by journey maps while others prefer a good old-fashioned flowchart. Others, myself included, like to take a story-based approach—using storyboards and scenarios to map our moments out. If you’re particularly fond of an approach for plotting out flows or journeys, I’m not here to change your mind!

I did, however, mention earlier that moments weren’t simply an unintuitive new name for user flows. While the tried and tested methods of documenting flows are almost analogous to mapping out moments, there are three key considerations that I feel have the potential to impact how we document our moments.

Moments Rely on Intent

Whether we derive intent from the performing of a specific action (clicking a “Pay Now” button should, we’d hope, indicate intent to pay for some items) or through a manifestation of state (which I’ll discuss in detail shortly), we need to be able to

somewhat accurately infer intent. It seems almost completely obvious, but ensuring we have sufficient input to suggest intent is an often overlooked task. It's a surefire way to make sure you question if any of your calls to action or framing scenarios are vague.

Moments Are (Mostly) Stand-alone

Having a moment that relies on a whole host of preceding moments is one of the easiest indicators that we're attempting to turn an explorative process into an overly linear flow. By documenting our moments in a way that makes them initiatable from *almost* any scenario, we're able to provide a clear path—or story—through our individual moments without requiring a sequential chain reaction of interactions. There's absolutely nothing wrong with actions that are only performable given specific conditions, and progressive disclosure is often an integral aspect of our work, but there's a big difference between saying, "Someone can only add a bio if they've set up their username," and saying, "Someone must add a bio after they've set up their username."

Moments Change State

Once a moment has played out, our system's underlying state will be changed in some way. By its very nature, interaction design revolves around undulations in state, and our moments often build up climactically to these changes. Take an image upload and edit process as an example. Our starting state might be a timeline with a list of posts. We initiate the "post-new-image" moment by clicking a camera icon. We go through the process of editing, tagging, and describing our image, and the upload succeeds. While it's possible we'll be returned to our timeline once the upload succeeds, our system state has changed, we've posted an image, and our timeline should update to reflect that. We might also wish to show a success message, or, if it's someone's first time uploading, highlight any actions that can be performed on their *own* image that they can't on the images of others, such as editing their post, or tagging other people.

Understanding these final changes in state might seem somewhat obvious, but I'm consistently surprised at how often I see designers—including myself—prematurely deem a moment or flow as over without considering the *true* final impact of the actions performed within.

In essence, moments tell stories. They have a beginning—one that is usually borne out of a need to achieve a goal or solve a problem, often with moments of friction,

conflict, adversity, and resolution—and they have an ending—the effects of which ripple through the fabric of our interface. While the cliché of “design is storytelling” has permeated the industry and, frankly, been discussed half to death, it’s hard to escape the structural underpinnings shared by interaction design and storytelling.

To me, interaction design’s core lies somewhere in this parallelism. Conducting not only what someone encounters from the start to the end of a moment, but how the reverberations of its resulting actions permeate through the rest of an interface is the art of interaction design. Done right, even the simplest moment can become an integral plot point as its elements and components shift in tune to action and manipulation.

It’s for these reasons that I believe communicating and documenting our moments in a story-based medium to often be the most enriching part of a design process. This stage is where we take everything from our feature list to our empathy maps and start piecing together how our work might be used in real-world contexts. In communicating how the key moments of our work fit in and around the day-to-day lives of people in a realistic and granular way, we create a bridge between researching, planning, and making.

Moment Stories

While storyboarding (and story mapping, journey mapping, and a whole host of other mappings) is nothing new in design, there’s a nuance to the approach that I feel many of us can easily overlook. It’s extremely easy to view any moment as a series of simple, matter-of-fact steps—to communicate it through the lowest common denominator by simply annotating these steps and then rushing straight into wireframing and prototyping. There’s very little point to a story-based approach to documentation that doesn’t account for emotion.

To view a moment as a story, we need to understand what *makes* it a story. This is more than having a beginning, middle, and an ending—although we will need those—and is more rooted in the fact that there is emotional fluctuation, creating interest or impetus. There are moments of friction or adversity that need to be overcome. There are points of dissonance that make us feel tension and resolutions that make everything feel right again.

A Moment's Beginning

Before a moment *really* begins, it has to be presented and framed. Framing an interaction involves purposeful representation of the possibilities of your interface at its current permutation of state. This is where the signifiers, cultural conventions, and affordances from Chapter 2 come in to play. Essentially, our interface needs to communicate the possibility of action, whether that's through a button designed to look clickable; an underlined, differently-colored link in a body of text; or a toggle switch in a form.

When we frame an interaction, we're looking for *intent*. As we explored earlier, the more certain we can be that someone wishes for a moment to unfold, the more confident we can be in shifting our interface to accommodate this. Intent comes from framing. While a clear visual language can dramatically help with this, it is often our interfaces' microcopy that most affects how confident we can be in translating action to intent. A Let's Go button, for example, does not communicate intent as well as an Upload button. Getting the right balance between the amount of context and the visual or cognitive weighting of a specific action trigger is key to framing interactions.

We'll discuss framing in more detail a little further into this chapter, but for now, let's just acknowledge that without solid framing, we limit how confident we can be that someone actually intended to instigate our moments.

Once we're confident our moment is framed well, we can think about the "real" beginning.

Moments begin as goals. Just as a movie might begin by introducing us to the goals and motivators behind our protagonist's journey, so too should we be aware of a person's goals when they begin a moment within our interface. From this point on, the thread we weave throughout our moment is there to serve this goal. In addition to the explicit goals that might be evident for our moments, we'll also want to consider the motivators that might affect it. We discussed in Chapter 8 the utility for empathy maps to accommodate both overarching goals and the underlying intrinsic motivations. Our moment documentation is no different. Goals are the *what* ("editing and uploading a photograph" is a goal; motivations are the *why*)— "feeling validated," "showing people how much I travel," "for the sake of artistic expression," and so on. Our moments, then, represent *how we get there*.

Just as any good novel presents a protagonist with obstacles and conflicts, so too should the stories of our moments. The "middle" of our moments, inasmuch as they can be wedged into a typically linear story flow, represent the progression through and ultimate culmination of the obstacles and problems that stand between someone and their goal.

Points of Friction

As we explored in Chapter 4, tension, resolution, surprise, and familiarity are all key factors that should be balanced throughout any form of progressive media. Interaction design is no different. Interaction design stands out most when we consider that the *source* of tension—those obstacles, problems, or schematic violations—are usually external forces; they’re the momentary distractions, the duress of cognitive load, and the day-to-day stresses of simply being human in this age of information. Music, movies, and novels get away with manufacturing their points of tension themselves. In the name of entertainment and engrossment, as designers we’re mostly left with how people’s environments, mental models, and emotions find ways to disrupt their moments of flow.

That’s not to say that there will never be points of friction intrinsic to our designs—there will almost always be awkwardly placed requests for account creation or the need to trawl through a relatively dull form to get to where we need to go—but that *intentionally* designing for tension is something that would require a great deal of thought and consternation.

As well as external sources of friction, technological constraints also present clear examples of unwanted tension. Network speeds largely dictate how long an interaction that relies on communicating with a server or another device will take. Hardware limitations can lead to certain interactions stuttering or generally feeling slow and clunky. Some moments of technology-based friction are simply unavoidable, and the best we can do is ensure that our products function well on older hardware or continue to function to the best of our abilities when a network connection drops. Catering to these unavoidable points of friction with the same dedication to communication we put into our “happy paths” is also key.

A request to a server failing—while often a clear source of drop-offs and poor conversion rates—doesn’t always have to result in the instantaneous end of a moment. Well-formatted and clearly communicated error handling, with clear instructions as to how to resolve a situation, could be seen as a moment unto itself.

While an understanding of intent, goals, and motivations is key to any approach to interaction design, it is the reality that so many points of friction will invariably be encountered that makes story-based documentation so applicable for modern design. By considering these various points, we’re accepting that both our own requirements and the real-world situational, environmental, emotional, and cognitive factors in someone’s life can be sources of tension, friction, and affliction.

The Moment of Truth

While the climactic rise from goal to completion might not be as palpable or as epic in an interface as it is in a movie, video game, comic book, or novel, it still exists. Now, it's true that very few people would describe their day-to-day interactions with products in the same ways they would describe the sheer exhilaration of the Battle of Helms Deep (or Happy Gilmore's final showdown with Shooter McGavin if you're me)—there's still a genuine point of climax in interface moments.

If we consider the “middle” of our interaction story as the “work” stage of a “signal, work, reflection” cycle, it's clear that, upon reaching a climactic point of a flow, someone has put effort into getting there. While hardly a trip to Mordor (although I'm sure we've all used interfaces that have felt less taxing), this work and effort still culminate in one final plot point.

If we stick with posting on Instagram as an example for this framework, the point of culmination is when we've uploaded, edited, and described our photograph. We've started with a goal (post a photo) and intrinsic motivators (I want people to see what I'm up to or I want to share a photo because I think it shows my skills as a photographer), we've done the work (picked a filter, edited the finer details, given our photo a short description), and we've done so through moments of friction (other apps' notifications begging for our attention, a crying kid who is tired of being neglected for social media, the creative anxiety of sharing your work with the world) along the way. We're finally able to hit that Share button.

In essence, this point of climax can be seen as a moment unto itself. It represents all the hallmarks of our larger moments in microcosm. The action to complete our moment needs to be framed well, and we need to be confident that it's what someone intended to do. There are potential points of friction, network issues, server errors, and even just a too-long wait, and there's an ending that allows for reflection, further action, or both—which brings us nicely to our moments' endings.

A Reflective Ending

Good endings show the value of the actions someone has just performed. When presented with tasks, work, conflict, or friction, we can quite quickly lose track of *why* we started this task in the first place. What was the point of it all?

By guiding people through these moments, like a skilled conversationalist telling a story, you're able to build toward a crescendo point, where the “main action” occurs,

that Post button is hit, our glorious photo is uploaded and ready for people to view... but then what?

A moment isn't over at its crescendo, and abruptly ending a moment at this point can become an insidious habit. All forms of progressive media are heightened by a *denouement* of some sorts—a narrative structure that involves tying up the loose ends of discrete plot points, releasing the tension of an enthralling crescendo, allowing for moments of respite, and reflection to ease out of the narrative.

Interaction design should be no different. The tiny stories that we weave into our moments—as well as how they contribute to longer, overarching story arcs that we might want to tell through continued interaction with all parts of our system—need good, reflective endings.

This denouement of sorts is an important stage. It's where we're able to show value and causality, to present someone with the results of their work and say, "Hey, you did this?" It is here where the reverberations of an action can be felt, a short few moments where someone is able to watch the effect of their work permeate through an interface. And while this short moment of reflection or celebration may seem ostentatious, without it we're depriving people of satisfactory endings, of being able to see that what they've just done—whether it's sending a form full of data or creating a masterpiece in a digital painting app—has results, an impact, a real ending.

In the world of interaction design, we often talk about "delight"—to the point, frankly, that the word has almost lost all meaning. The closest we get, in my opinion, to instigating this delight is by designing these moments of reflection. Instagram does this by showing us our photograph in our timeline after we've completed the posting process. This might seem like a matter-of-fact, obvious thing to do, but that doesn't detract from the fact that we're presented with a satisfactory ending. Through browsing Instagram as a "consumer" (a word which will never sit right with me) of its content, we've seen other people's photographs in this format, we understand that our timeline is the stage for these captured moments (and ads for weird clothes that no one wears and detox tea that no one drinks, but we'll forget about ads in the name of romanticizing interactions), and now we see ourselves on our timeline, knowing that others will see our same post, in the same format, on theirs.

If our timeline was left untouched, or for some reason, Instagram decided not to show us our own posts on our own timelines, we wouldn't have this moment of reflection. We might have a simple success message, an assurance that, yes, our photo was uploaded successfully, but we don't get to *see* that—we're not allowed to reflect or revel for a short moment at what we've achieved (Figure 9-3).

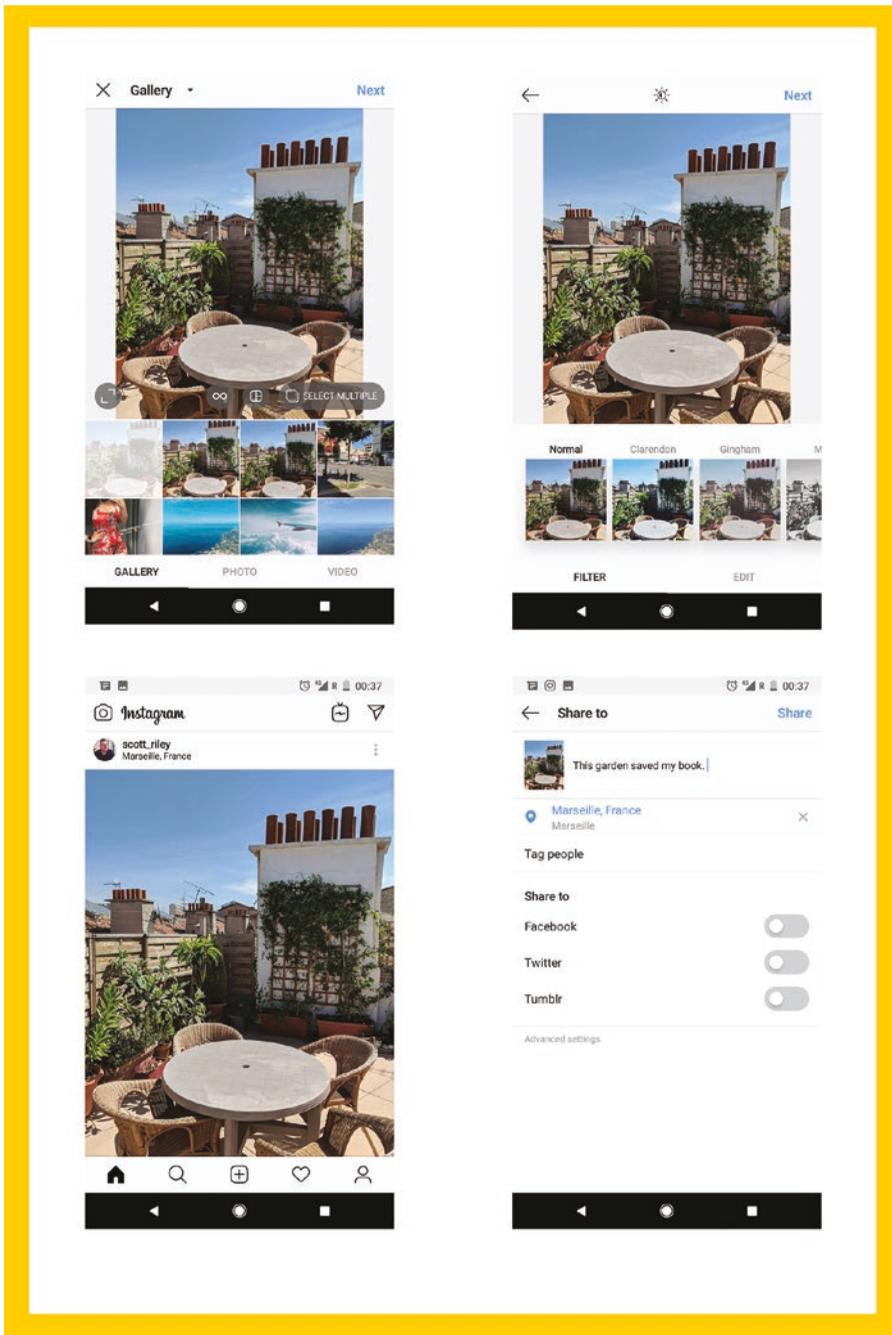


Figure 9-3. *Uploading a post to Instagram, clockwise from top left: choosing and adding a photograph, editing the photograph, adding the final post details, seeing the post presented in your timeline*

These four steps are a distillation of the entire story behind posting an image. It took me less than a minute to complete, but many millions of people every day enact their own versions of this out.

Posting a photo might seem, initially, to be a trivial and ubiquitous interaction. A simple checklist item on your never-ending task list. You've probably used, or designed, something that involves some form of uploading—adding a profile picture on a social network, uploading a screenshot to Slack, attaching documents to a bug tracker or task manager, and so forth. However, if file uploading or image posting represents a core moment in our interface, we'd do well to remember that stories lie behind the seemingly obvious stages of the interaction. Stories that start with goals and motivations encounter points of friction along the way, build up to a climactic moment, and finally ease out allowing moments of reflection and resolution.

Storyboarding Moments

As far as early-stage communication of the intrinsic emotion behind moments and interactions, storyboarding is—in my opinion—a priceless method of documentation. Storyboarding allows us to present a condensed version of the moment story we're trying to tell while still communicating the fundamental goals, motivations, points of friction, climax, and ending that constitute our moments.

Storyboarding is a technique lifted straight from the movie industry, where scenes are storyboarded in comic-book-style (Figure 9-4) panels to quickly convey the main plot points.

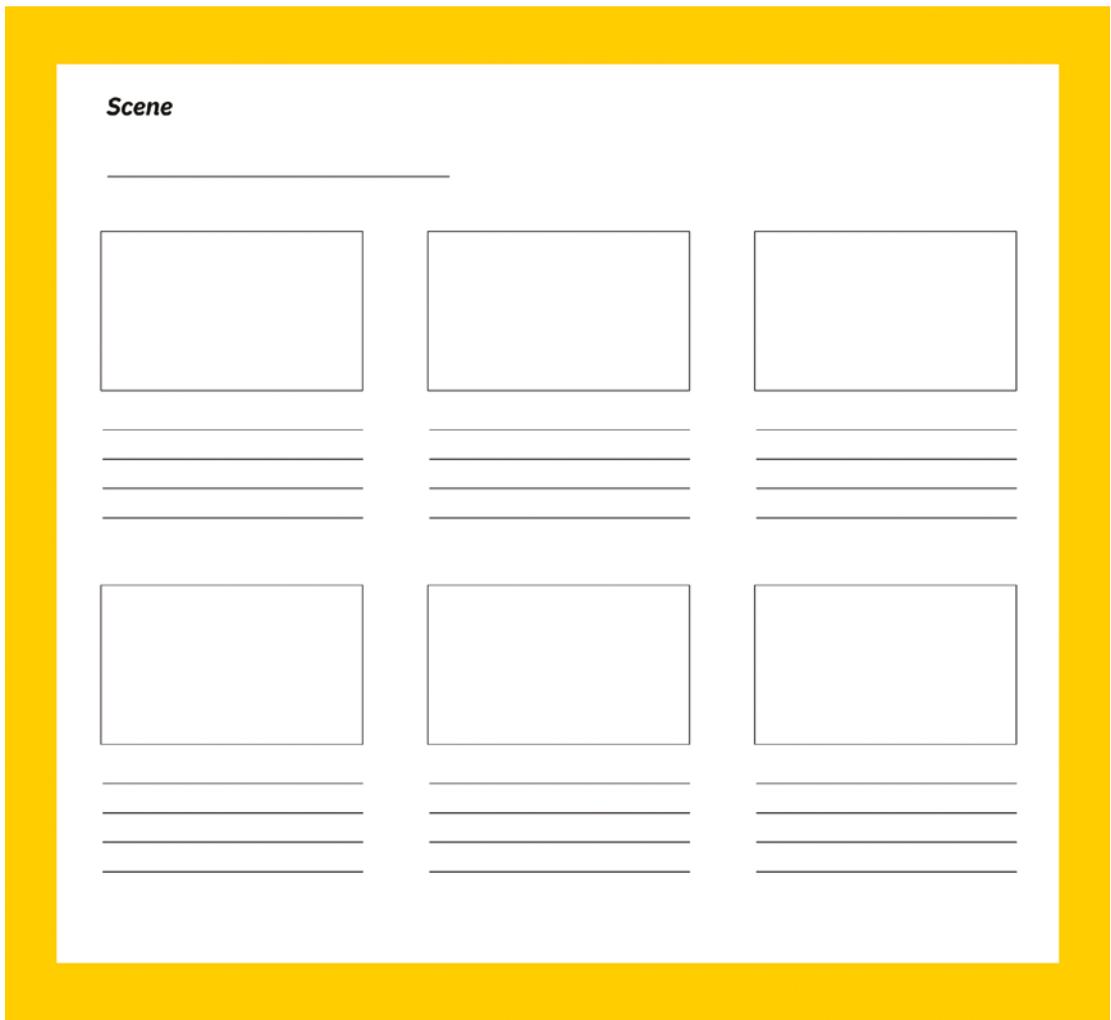


Figure 9-4. A storyboard

In product design, storyboards work exactly the same. Often compiled to show a more abstracted story of someone’s general end-to-end experience with a product, they’re just as applicable in our communicating of smaller, discrete moments. When we consider the push toward more open and explorative defaults, with story-driven, linear moments forming the crux of performable actions, it often makes sense to *only* storyboard our more concrete modes and to fall back on state documentation and design systems to communicate our system’s broader qualities.

A common barrier to trying storyboards is the idea that you need to be somewhat artistically skilled to produce a usable storyboard. While it's pretty daunting at first to imagine clients and colleagues pouring over your artwork, it's important to acknowledge that storyboards made entirely of stick figures and poorly drawn environments are far more commonplace than the overdesigned examples you might be used to seeing. Get over your fear of drawing and remember that we're not here to produce a best-selling graphic novel. We're here to communicate a story in a rough, visual way.

A simple approach to storyboarding is to stick to nine panels, have your beginning panel set the scene—preferably with goals and motivations—spend the next six panels incorporating your potential app steps, and finally have panel seven as the moment of climax, panel eight as your ending, and panel nine as a moment of reflection. You can (and should) produce multiple storyboards per moment if you need to communicate more points of friction, different scenarios, and contrasting or conflicting goals and motivations. My advice for a storyboard is to follow a simple framework, starting with a template and following some set steps (Figure 9-5).

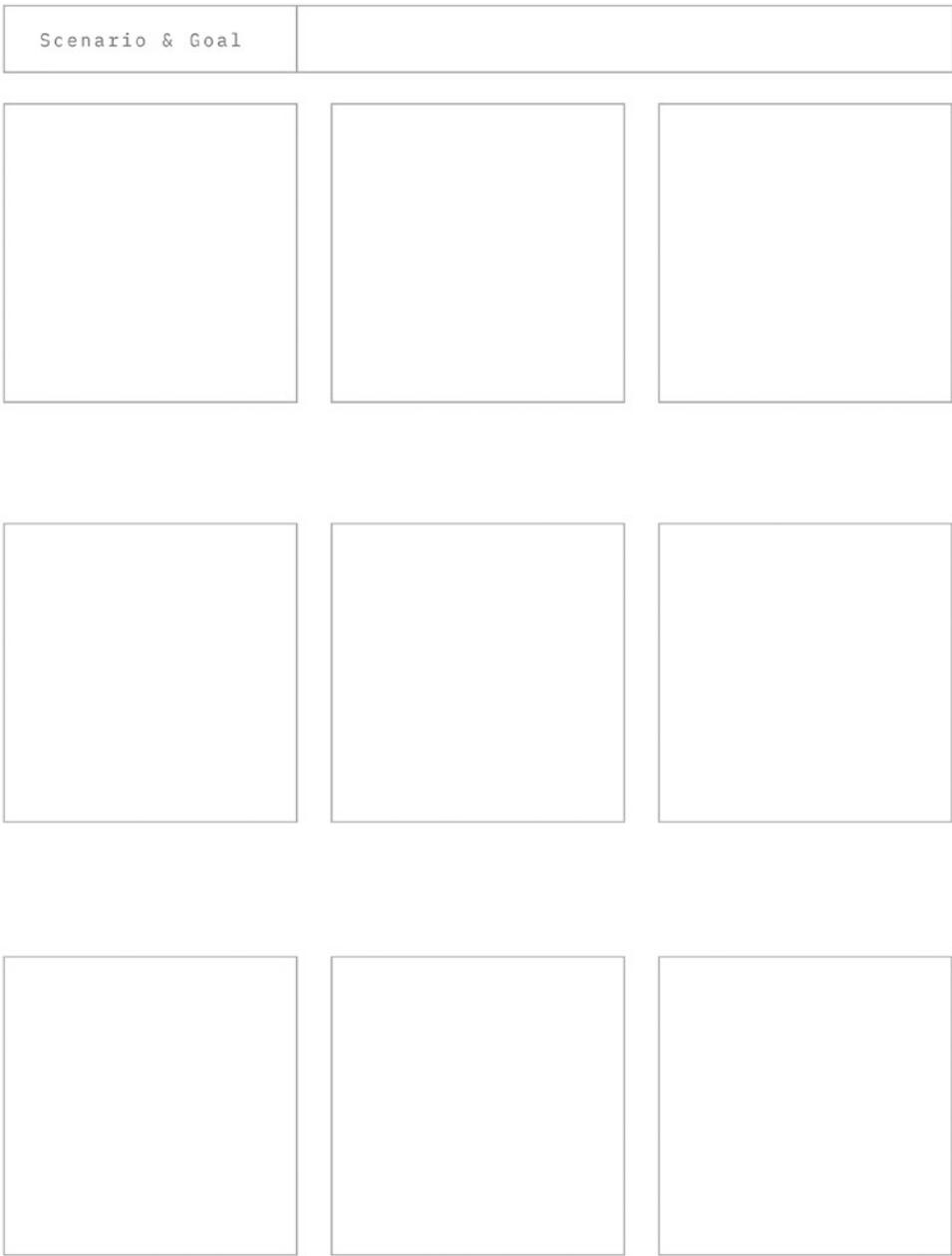


Figure 9-5. *A blank storyboard template*

Once we have our template drawn, printed, or loaded into a digital drawing app (you can plot your storyboards out on a whiteboard and use full-sized sheets of paper per panel if you'd prefer a bigger canvas), we can get started with our storyboarding.

Pick a Moment and an Empathy Map

Given that storyboards should take environmental, emotional, and motivational contexts into consideration, empathy maps are a great tool to bring in to a session. Write down the goals and motivations behind the moment that you're storyboarding. For example, when storyboarding our "new post" moment, we might list "upload and edit a photograph from our vacation" as a goal, and "I want people to appreciate my photography," and "I want people to see me as a well traveled and cultured" as intrinsic motivators. Illustrate this as your first panel and jot down some context outside the panel's frame.

Plot Out the Story Flow

It often helps to plot out a simple flow first. Break out the Post-its and try and describe your moment in simple, conversational steps (Figure 9-6).

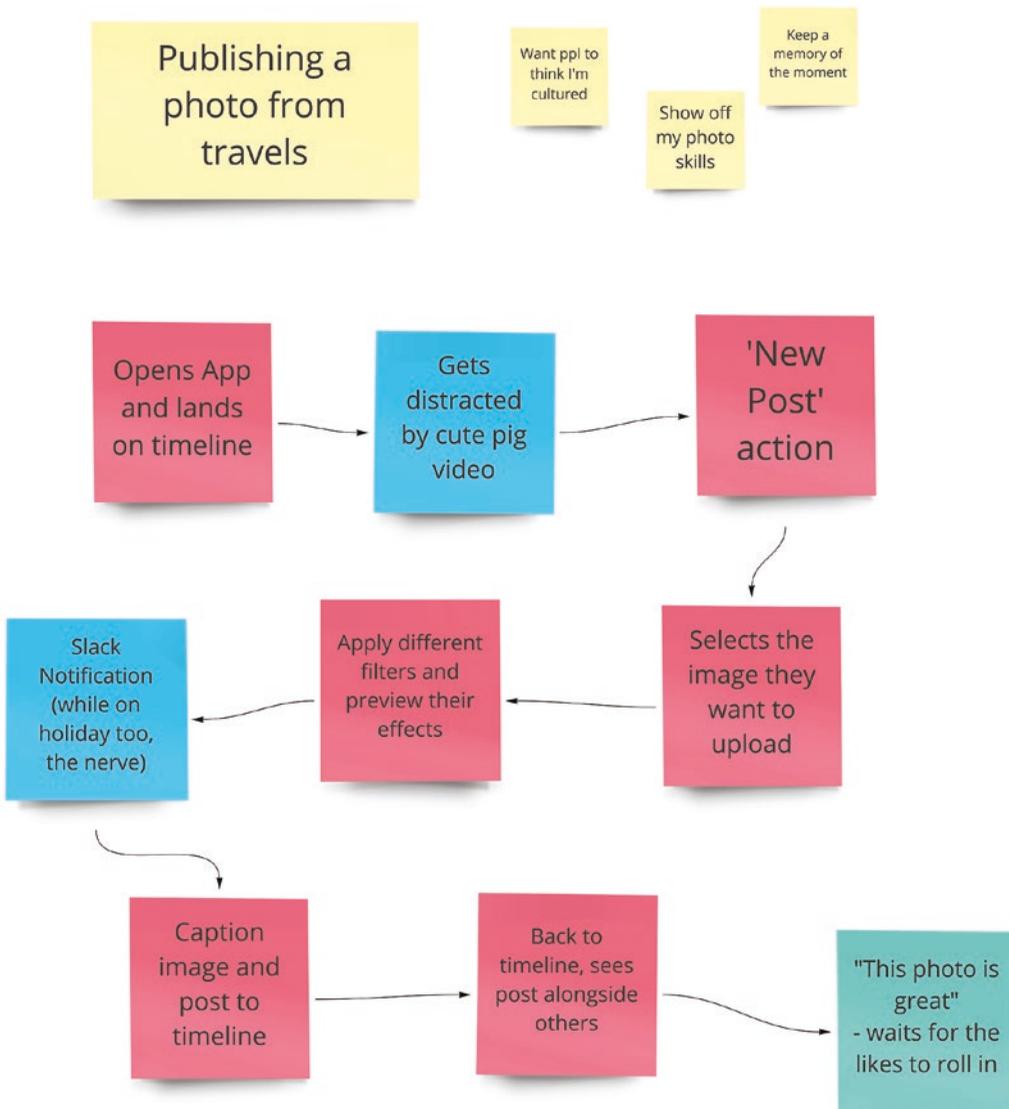


Figure 9-6. A basic text-based story plan

Tell the Story

Take your rough flow and turn it into a sequential, comic-like storyboard. Pay particular attention to the points of friction and the climax of your story. If a particular feature along the way might represent an obstacle or stumbling block, make sure it gets its own panel. As we explored earlier in this chapter, points where mental models are deviated

from, long processing, or upload times and generally demanding tasks are all prime examples of potential moments of friction.

Consider how your character (the persona associated with your empathy map) might respond to these moments of friction based on their goals, motivations, and environmental issues. Again, good ethnographic research and well-rounded, challenging personas and empathy maps are vital here. The more confident we are in these, the more confident we can be in our storyboards representing atomic moments of use.

When telling your story, use speech bubbles and thought bubbles to show what your character might be thinking or saying. We can often lift these directly from our empathy maps. Other times, we'll have to hypothesize when it comes to more specific, context-dependent dialogue. Again, we're not trying to be prodigious scrip writers here; simple, condensed thoughts and phrases are totally fine. Consider, too, working in any verbatim quotes you've acquired from your research interviews. Any opportunity we find to bring a real human being's wording or feelings in as a reference point is one that should be snapped up without deliberation.

End the Story Properly

Make sure you consider beyond the climactic point of your moment and have at least one panel reserved for the point of reflection and reverberation discussed earlier in this chapter. Furthermore, try not to lose track of the fact that the *true* endings of our moments often manifest themselves in the real world.

While our goals might be specific to the app or service with which we hope to accomplish them, our motivations often exist far beyond the borders of our applications and the devices on which they live. It's extremely easy to fall into the trap of limiting design thinking to purely within the constraints of the product or medium for which we're designing. While this degree of selective focus is a genuinely essential ability—being able to identify, implement, and manipulate the conventions, common signifiers, and implications of the medium we're designing for is a core design skill—it can often lead to harmful presuppositions if we adopt it too early in our process.

An ending that shows not only the value of the tasks that have been performed but also the changes caused and the motivations realized in their performance is often key to a satisfactory storyboard. One that takes the time to consider the real-world implications of a particular interaction is paramount.

Figure 9-7 shows a completed storyboard.



Figure 9-7. A completed storyboard that might be representative of posting to an image-sharing social platform

While extremely rough, this storyboard communicates the plot points from the previous step and ensures that potential distractions are accounted for. Note that, while nine panels are a good starting point, many stories will span more than one sheet and many more still won't fill a single page.

Annotate

Once your storyboard is laid out and you're happy that it's appropriate and representative, it's time to annotate. Some key questions to jot down include the following:

- *How might people be feeling, thinking, or saying right now?* Remember your points of friction, goals, and motivations and try to ensure that the emotions you jot down are representative of your research.
- *How might interruptions and distractions occur?* For each frame, try to annotate some of the potential distractions and interruptions you uncovered in your research or testing phases. Annotating potential distractions lets you weave the real world into your stories and, more importantly, keeps relevant the need to consider that your moments need to be continuable after a distraction. Relying on the context of a previous action or some copy earlier in the flow can often lead to interactions that are extremely well designed when performed in one sitting, but essentially crumble if someone is distracted part-way through.
- *Why might someone drop off here?* Every point in a moment is a potential drop-off point—as I'm sure anyone who has worked on funnel analytics could attest to. While many of your drop-off points will be inherently tied to your interruptions (getting back into the flow of a moment after being dragged out of it is often just too much to ask), we should also consider the reasons intrinsic to our moment that might cause someone to abandon their goals. More obvious examples for this include sign-up forms (asking for meaningful data takes a conversation beyond causal), payment flows (asking for money *absolutely* changes the tone), and long waits and concepts

that are too complex or obtuse to understand. Annotating potential drop-off points allows us to take our moments into wireframing and prototyping with an established idea of the problems we might face. Might something be too far removed from our mental model? Make sure we take the time to describe it well. Is asking for payment information wont to deter someone from continuing through a moment? Make sure we frame the value of what they're paying for and try to ensure the payment form isn't just dropped on someone halfway through an otherwise casual conversation, like a cartoon anvil on the head.

Moment Mapping

While annotating directly on your storyboard is totally fine, you might find that there's just not enough room in the margins to fit your notes in. Additionally, while annotating your storyboards, as a solo exercise, is a valuable use of your time, it's often even better to use this as a chance for collaboration. A diverse group of people contributing to this annotation is—as is the case with any part of our process that involves speculating about emotion, mindset, problems, and distractions—always going to serve us well.

My preferred approach to annotating is—wait for it—whiteboards and Post-its. If you've drawn out your storyboards digitally, print each frame off at A4 size and stick them up on a whiteboard. If you've drawn them out by hand, stick the page up and number each frame. If you've drawn them out by hand on full-size A4 pages because you knew you were going to annotate them on a whiteboard—good job!

Once we have our panels arranged on our whiteboard (or canvas—this approach, like every other I've mentioned, is just as doable in a digital environment as it is a physical one), we can arrange some rows beneath each one. You might already have experience with this kind of mapping. It's a common approach to journey mapping that many UX professionals will advocate. Personally, I like to split my map into four rows—one for how someone *might* be feeling, one for *how we want them to feel*, one for distractions/interruptions, and one for potential drop-off points (Figure 9-8).

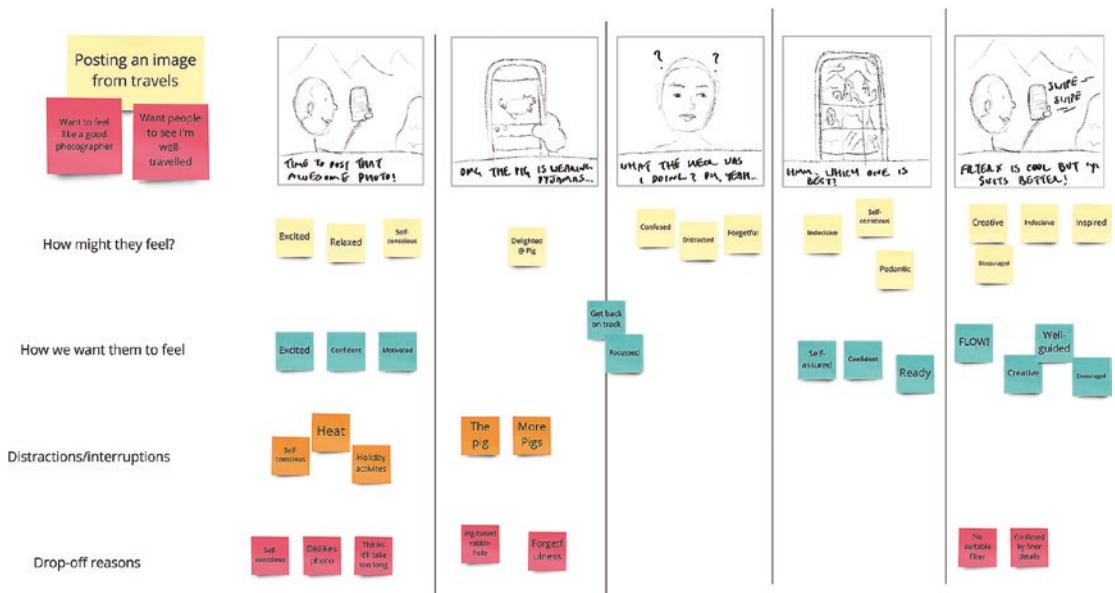


Figure 9-8. One possible way to set your storyboards up for annotation

This approach lends itself nicely to a group brainstormer. While I believe that storyboarding itself should be conducted by a limited group (especially for our moment stories), annotating storyboards is a fantastic way of getting larger teams invested in these stories. Start by briefly running through the overall stories you'll be annotating—remember, we're thinking key moments here. If you have many stories (multiple key moments and multiple empathy maps can see your story list stack up *fast*), you might want to pick just a couple that you feel would truly benefit from this workshop-style approach to annotation.

After regaling our colleagues with a brief run-through of our moment story, we can start annotating. Work from the first panel and try to keep a brisk pace while still being casual and not bulldozing through your stories. A minute or two per panel is often enough, but absolutely spend more time on the juicier points if discussion opens up. (A panel that essentially represents a yes/no answer is going to be extremely light in the way of notes compared to, say, a panel where we ask someone for their payment information.)

It might take a while for the room to warm up to this approach, so treat this session like a story itself. Start by stating the goals and *always* bring your simplest story in first. While you may be chomping at the bit to get everyone annotating your nine-panel epic onboarding story, remember that you've spent a great deal of time with your head in

these stories and many of your colleagues or clients might be coming in cold. Warming up with a simpler story to get conversation flowing and to allow for mistakes and misinterpretation is often a shrewd move. End by explaining how the work done that day will impact the rest of the project, how you hope everyone involved now has a better understanding of *why* we're focusing on the moments we are, and that you're *super excited* to get on using these stories to inform your prototypes.

Like any group brainstorming session, you'll want to go away after the meeting and prune the ideas and suggestions that arose. While nobody wants to see their ideas discarded (and this is why we do it after the meeting, as opposed to looking a developer straight in the eye and tearing their Post-it up in front of them), an integral part of our jobs is ensuring that our work is representative and grounded. If the CEO thinks that someone will feel "delighted and in complete awe" at the "seeing-pricing-for-the-first-time" panel in a story, we'll need to find a way to ease that out in favor of the more realistic "concerned-about-financial-impact" and "wondering-if-this-product-is-actually-worth-it" suggestions we might have. (In this case, feel free to do the whole staring-while-tearing-a-Post-it-up power play—it's only the CEO.)

Leave the Post-its on the whiteboard (physical or otherwise) and thank everyone for their time. End the meeting, grab a coffee and a water (please hydrate, meetings are hard), and gather your design and product colleagues for phase two: pruning.

Pruning is something we should do after every brainstorming session, and I recommend doing it straight after. It makes for a pretty draining day, but you'll be pruning a brainstorm while the context behind everyone's suggestions is still somewhat fresh in your head. The purpose of pruning is to take any unrealistic suggestions and either get rid of them entirely or turn them into more realistic ones. As the team (or person) with the deepest understanding of your research (and if that doesn't describe you, try to make sure it does next time!), you're in the perfect position to make a judgment call on what is signal and what is noise.

The more you prune, the better you'll get at spotting red flags—small traits that set alarm bells off in your head. An obvious red flag is the kind of unbridled optimism that anyone who might describe a company as "my baby" (founders, CEOs, investors who like to overstep boundaries, etc.) might display. Being overly attached to a product's success to the point where one believes it to be a flawless solution, devoid of any potential malaise, represents a special kind of myopia. In fact, one of the most important skills you can learn as a designer (that, unfortunately, tends to only come with experience) is the ability to quell this kind of optimism-ad-nauseum without appearing as if you're a prodigal spoilsport who only exists to ruin dreams and scare children.

Other red flags include opinions from people who have lived a life of sheltered, abundant privilege, who have yet to prove capable of genuine empathy (apologies, but unless you're working on an app to dodge mansion tax or a croquet-court finder, then coddling the opinions of ever-comfortable people can be one of the fastest ways to sink a product), suggestions that revolve completely around revenue, ideas that rely on manipulation, and proposals that appear completely rooted in “competitor X does this.” (Tread lightly with that last one, though, as competitor research is a *hugely* valuable insight into convention and mental model consistency.)

The result, post-pruning, of our storyboard and story-mapping exercises should be a set of succinct maps that not only communicate potential scenarios and stories but are also imbued with many of the environmental, emotional, and motivational details we uncovered in our research phase. While a story for each combination of empathy map and key moment might seem daunting at first, it's important to stress that these activities can be performed surprisingly quickly. It's also feasible, should you be strapped for time, to replace a number of common UX deliverables—such as journey maps, user scenarios, and flow diagrams—completely with moment-specific, annotated storyboards.

Storyboarding and annotating, while best suited to an early-stage, pre-wireframe phase of our process, can also act as a great means of evaluation. If you have a particular feature that appears to be underperforming—either through high drop off rates from analytics or through poor performance in testing—storyboarding its *current* incarnation can reveal plot points that would otherwise be difficult to spot or elucidate. Using that to then create a more representative storyboard to lead into a feature redesign can be a surprisingly efficient means of improving existing products—or at least discovering areas where a redesign could be beneficial.

The Anatomy of an Interaction

One of the many benefits of this moment-based approach to interaction design is that the framework scales. Whether that's scaling upward to think about how the discrete, broad components of a wide-reaching system interact or scaling downward to the finer details of micro-interactions, describing, drawing and annotating the stories that accompany interactions is—in my experience at least—an extraordinarily valuable habit to introduce to a design process.

While there is absolutely a “sweet spot,” of sorts, for moment story documentation—lying somewhere between a discrete usage session and the story of a single key moment—the principles and theories can be applied at surprisingly varied levels of granularity.

Given this, it would be easy (and admittedly tempting) to ask you to apply the above logic at a different level of granularity for some just-add-water interaction design framework. However, while moment stories are a valid starting point for smaller interactions, I’d like to explore in a little more detail how these single, smaller interactions fit into our work.

Interaction design—that is, the *specific* design of smaller interactions, not the overarching industry term for “the arbitrary areas of design that an interaction-focused designer is responsible for implementing”—is increasingly becoming more and more important. As technology advances and operating systems and browsers become more capable of supporting rich interactions and myriad input and output methods, the central role that interaction design plays in any product is becoming more evident.

Good interactions make use of a wide array of design disciplines. Visually, they need to be on-brand and consistent, and they should adequately signify possibility. From a motion-design standpoint, easing harmoniously between interaction states is integral to ensuring a smooth interactive flow. Everything from copy and microcopy to cognitive load to the easing curve or spring physics used when animating an element needs to be constantly balanced throughout interactions—and that’s before we even consider how this often has to be done with limited space and.

But what *is* an interaction? It’s not the element itself—a button, of course, is not an interaction. It’s not *quite* the words in or around the element, although describing an action is important to causality and communicating possibility. Nor is an interaction the specific animation or string of animations that occur after an action is attempted. Interactions are abstract, undulating entities. They simultaneously exist as a communication of action possibility, as a manipulator of state, and as a representation of current state. However, this eulogizing essentially leaves us with *Schrodinger’s Interaction*—something that can only be defined based on its current observable state, which is just a little too pretentious, even for this author.

Instead, interactions can be seen as micro-moments. They’re *things* that play out progressively dependent on human action and our system’s state. Interactions push our stories along, one-by-one—every button click, every swipe, hover, or scroll helping us further toward our goal. If we see our moments as the stories of our work, then

interactions act simultaneously as plot points and segues, providing the key action points for progression as well as controlling the transitions between different states as this progression occurs.

So, what makes a good interaction? Given that we know interactions are somewhat abstract things, liquid in their definition, we can be sure that context will play a deciding factor in the efficacy of any interaction we design. If something is supposed to be extraordinarily intuitive, then an interaction that has us stopping to think is likely a poor choice. Likewise, if an interaction is supposed to slow things down and have us purposefully interact with a feature at a deeper level (as we discussed in Chapter 3, this is an essential characteristic of learnable concepts) then an interaction that essentially involves clicking a Continue button is likely to be just as ill-suited.

Aside from ensuring that tools remain contextually appropriate, there are several common factors we can consider that should at least equip us with the ones necessary to get started planning and evaluating the interactions we design. A good framework for discussing these factors is to treat interactions as a form of conversation—which, if you agree with the earlier chapters of Part 2 of this book, they essentially *are*. Furthermore, by adopting a story-based approach to broader moments and a conversational approach to smaller interactions, we're able to imbue a pretty consistent context into our early-stage design work. If you'd rather steer clear of the story and conversation metaphors entirely, though, that's completely fine—just abstract the ideas in this chapter up a level and apply them to your preferred methods of communicating your design decisions.

A good interaction, then, relies on the next areas we will discuss.

Framing

Just as we discussed earlier in this chapter, a moment, plot point, interaction, or *any* narrative or conversational tool is nothing without effective framing. To repeat the sentiment, the more confident we are that someone has communicated intent, the more confident we can be throughout the rest of our interaction.

Framing allows someone to preemptively envision what a certain interaction might do. The obvious example here is button text—with one of the key (albeit most-ignored) tenets of interaction design being that button text *should describe the action it performs*. Poor framing is, in my experience, one of the most avoidable, yet predictable errors in interaction design. However, it's not always as simple an issue as taking an inventory of your microcopy. After all, plenty of wordless interactions (I'm willing to bet that clicking

a camera icon and subsequently a wordless, circular button to take a glorious selfie is one of the most practiced and ingrained interactions modern society has with their devices) are completely satisfactory.

In order to frame interactions well, we must rely on many of the concepts already explored in this book. Gestalt rears its wonderfully practical head again here—perceptual grouping and consistency between elements is incredibly important to communicating action possibility and causality. Signifiers and convention also play hugely important roles in interaction framing. Buttons that are obviously usable, separation of decorative icons and clickable icons (especially common on smaller screen interfaces where space is precious), obvious states for items like check boxes or toggle switches, and myriad other considerations all have a potential impact on how well-framed an action is.

The key phrase here is *action possibility*. While entire interface states can often be devoted to convincing, pushing, or nudging (or whatever over manipulative verb we're controversially comfortable using this week) people into performing an action, seeing an interface primarily as an open venue and a communicator of action possibilities offers respite from constant persuasion and illusions of control. Now, I'm not suggesting we shy away from a hierarchy of actions—that is, in fact, often a recipe for disaster—just that if we frame actions well enough (part of which involves communicating their hierarchy based on the current state), we can present an explorative interface when we feel its applicable to do so.

Acknowledgment

During a conversation, when someone provides their input, they usually expect it to be recognized. Even if we have nothing to offer to a particular conversation point or we need to pause and think, we'll generally at least *acknowledge* that something has been said. Interaction design is no different. When we do something—whether that's click a Submit button at the end of a long form or a simple swipe to change an image's filter—we want to be told that our message has been received.

Now, in my experience, it's extremely rare to find someone who expects digital interactions to be instant. Most of us are all too familiar with loading spinners, progress bars, and general slow responses and, even as network speeds and average device specifications continue to improve year after year, the very nature of Internet communication means there will almost always be *some* kind of delay to interactions that rely on network communication. Fast acknowledgement does not have to mean fast server response times (although that invariably helps) or instant app loading.

What many of us *do* expect to be close to instant, however, is this acknowledgement. Think about a conversation where you've been asked something that's required you to think a little. Maybe you've been asked a tough question or you're trying to remember something that's relevant to the conversation. In such a situation, we can find ourselves doing all kinds of strange gestures and facial expressions to signify we've acknowledged someone's input. The old trope of people looking "off and to the right" when trying to remember something is a classic example, but everyone is different. Some people will nod along to show (or feign—you know who you are) interest and presence while others will verbally "mm-hmmm." When we need a moment to think, some of us might pull that strange scrunched-up pouty face or stare up at the sky. The fancy among us might even prod our chin or press a finger to pursed lips to really hammer home our consternation.

In fact, we rely so much on these verbal and visual cues, that we specifically exaggerate them. As someone who personally suffers from social anxiety, my brain constantly looks for rejection and disinterest in the behavior and expression of others. While I struggle to remember plenty of conversations (especially ones about deadlines, sorry), I can vividly remember the few rare occasions where something I've said has been met with an emotionless blank stare. As we discussed in Chapter 1, ruminations can often associate themselves with what, at the surface level at least, may seem to be banal experiences. Yet, a propensity to panic in the face of a blank response or lack of acknowledgement is not a trait exclusive to those of us who live with anxiety. Inherently, we all have a need for conversation and, with that, comes a need for acknowledgement.

Interactions should work no differently. If our system has to speak to another (our front end to an API, for example), one of the worst things we can do while that time passes is *nothing*. Simple loading spinners is ubiquitous in modern applications for this very reason. Perform the digital equivalent of a typical "I'm thinking..." gesture lets someone know, pretty much instantly, that their input has at least been acknowledged.

Lack of acknowledgement is yet another subtle but insidious error that often goes unchecked in design work. The phenomenon of "rage clicking" is inherently tied to this.

Rage clicking, as the phrase might imply, involves clicking (or tapping) on one, more, or all areas of an interface in the hopes that *something* might happen. Seeing someone rage clicking around your interface is probably as close to a personally heartbreaking moment as design can produce. It's when you know you've screwed up to the point where someone would literally rather apoplectically wear their finger down on a keypad in the hopes that *anything* will happen rather than think rationally about what is (or isn't) happening. Rage clicking, essentially, is a total confirmation that we've done

our job wrong. Yet, while single-handedly justifying the need for emotional support animals in design studios across the country, it does present us with clear evidence that our framing is poor (someone is clicking something that isn't clickable) or that our acknowledgement is worse (our interface is basically staring blankly into someone's eyes and hoping the ground swallows it hole).

Acknowledgement also serves a secondary goal in our interface and that is communicating a lack of action possibility in its absence. If we consistently acknowledge someone's input from their first moments of interaction and if they try to interact with something that is purely decorative or descriptive (e.g., clicking a decorative icon believing it to be an icon button), then the lack of response—within an otherwise responsive system—can provide a last-ditch anti-signifier: “You can't do anything with this.” This, of course, relies on “any action will be acknowledged” being part of someone's schema for our application, which means we need to be responsive early and consistently. It does, however, represent an interesting knock-on effect of consistently acknowledging actions as soon as they're attempted.

How you acknowledge an action is going to depend heavily on context, constraints, and the personality you're trying to convey. A ubiquitous example of acknowledgement is turning button text into a loading spinner—which works extremely well if you know your wait time will be relatively short (Figure 9-9).

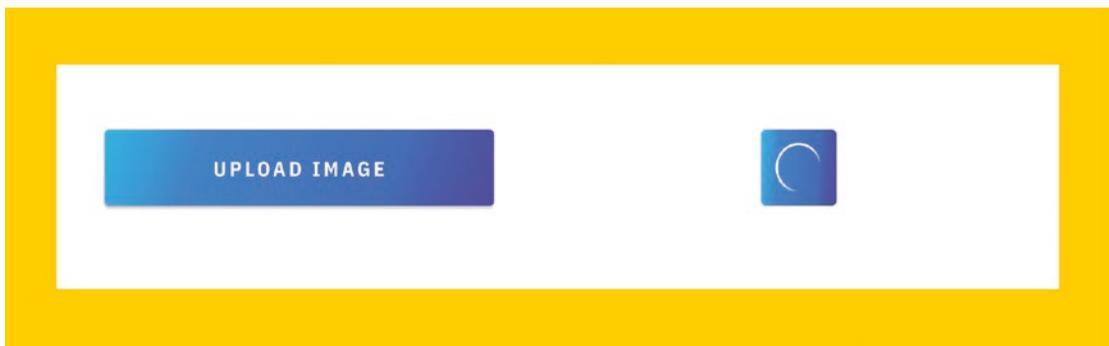


Figure 9-9. *The before (left) and during (right) states of a button that acknowledges input*

This approach is effective for a number of reasons. First, a loading spinner is an extremely common convention, found in the vast majority of mainstream apps, and it communicates a clear message. Second, by replacing the button text, we're acknowledging an action in the same context that it occurred—yet another point for

perceptual grouping. Finally, by replacing the action's text, we're subtly and temporarily removing the action possibility from our interface.

As I mentioned, our approach to designing these in-between states will change drastically. In some cases, we might want to go all-in and have a full-screen overlay with a spinner to be completely sure someone's action has been acknowledged. Other times, a subtle progress bar at the very top of an interface can do just fine. Probably the most integral factor to this will be the time we expect someone will be kept waiting.

Waiting

Waiting is the undesired inevitability of interaction design. As anyone who has worked in the service industry can attest to, most people have a rather ingrained disdain for being kept waiting—yet, as interaction designers, we *do* have the somewhat fortunate safety net of expectation in our corner. Indicators show that a specific action is taking its sweet time to process (our good friend the loading spinner, progress bars, etc.). They are some of the most ubiquitous design elements around, and they should tell us how common waits are in digital interfaces.

Yet, these expected waiting periods are also ripe for subversion—an interface that we perceive to be faster than others is one that stands out more often than not. Exclamations of “Wow, that's fast!” and similar displays of shock and awe that an interface takes less than a perceived age to communicate with a back-end server are still some of the most common forms of praise that people bestow on interfaces. While often outside of our control, maximizing the speed at which various nodes of our systems communicate with one another is an incredibly important task.

However, simply shaving off precious milliseconds of loading time isn't the only way we can subvert the waiting process. For some categories of interaction (video sharing, uploading high-quality images, processing extremely large files), long waits are an inevitability. By eschewing the sadly common practice of disabling interaction with the wider interface and allowing other areas of our interface to be explored, we're able to essentially negate the impact of long waits.

Say, for example, we were working on a browser-based, UI design tool. A key feature of this will likely be uploading images. After all, is a marketing site without a full-page header image *really* a marketing site? Traditionally, image uploading has been a painfully drawn-out interaction in web sites and web applications, often forcing us to simply sit and twiddle our thumbs while waiting for an upload to succeed. However,

with the advancements that browsers and operating systems continue to make, this really doesn't have to be the case. Let's say that, on an average connection, the average image that people upload for use in their designs takes one minute to upload. Not totally implausible given that designers often work with extremely high-resolution images while mocking up their work

Option 1—the easy option—is to fall back on the tried and tested common practices for image uploading. Click a button, select an image, and ... well ... *wait*.

Option 2—which is thankfully becoming more and more popular—is to process this upload in the background while keeping the entirety of the interface usable for the duration. This would let us drop our image in and spend that precious minute deciding which generic geometric sans serif from Google fonts we should use this time. We could even take this a step further and load in a placeholder or low-res preview of the selected image while the full image continues to upload—allowing designers to move and size the image into place without worrying about progress percentages or watching a loading spinner for a minute straight.

Finally, consider the concept of *perceived wait*. Perceived wait is, far above and beyond the *actual* waiting time, our major concern when designing interactions that might be a lot more involved than a basic “send-this-10kb-of-data-to-an-API-server-and-process-the-response” model. Many studies have been conducted on perceived wait in environments as diverse as hospitals, airports, and theme parks.

In a 1996 study of emergency-room waiting times, David A Thompson, MD, and colleagues concluded that “perceptions regarding waiting time, information delivery, and expressive quality predict overall patient satisfaction, but actual waiting times do not.” (Thompson, Yarnold, Williams, Adams, 1996). Thompson and colleagues, in a survey-based study, show that, while perceived wait had an impact on satisfaction, actual waiting time did not.

More anecdotally, we can look at how Disney manages their theme-park lines to account for the importance of perceived wait.

In Disney theme parks across the world, people wait in line—often for times approaching an hour (and sometimes longer still)—to ride a rollercoaster for 20 seconds. And while the psychology behind thrill-seeking (and rollercoaster riding in general) is genuinely interesting, I'm afraid it's the lining-up part of things we'll be examining right now.

While there are many logistical and systemic optimizations that can be performed to reduce actual waiting times in theme-park lines, a wait that *feels* shorter than it really is can often be the deciding factor in satisfaction—and Disney is a master of this. Our problem, it would seem, with waiting is that we’re just so easily bored. The time between formulating a plan (get on this ride) to achieve a goal of ours (feel like our stomach is turning inside out because we’re awfully strange animals) and having that plan actually unfold can, understandably, be excruciating. Disney—and to be fair, almost every other theme park in the world—understands that by filling this “empty” waiting time with smaller attractions and distractions, many people may be distracted enough to forget that they’re actually even waiting in line.

The lines for some of Disney’s larger attractions unfold through winding paths, littered with greeters and performers, food and drink stalls, and even video-game stations. By presenting people with these distractions, Disney benefits from people’s thoughts being diverted from “I wonder how long this is going to take?” to “I wonder if those are Gaston’s real muscles or if it’s just part of the suit?”

Closer to home (unless you work at Disneyland, of course), video games also handle perceived wait extremely well. As video games grow larger in every conceivable way, the time it takes to load certain areas can reach levels that are often worrisome for games designers. By managing our perceived wait, games designers can often mitigate the potentially drastic effects of loading times. The popular Ubisoft series *Assassin’s Creed* allows the player to roam around in an extremely limited environment while the main map and assets load—giving us a chance to practice our skills or reacquaint ourselves with any controls we may have forgotten. Bethesda Softwork’s *Fallout: New Vegas* manages perceived wait by presenting short facts or tips about the game’s world, lore, or mechanics (Figure 9-10).



Figure 9-10. *Fallout: New Vegas* (Bethesda Softworks) helps to improve perceived wait by presenting facts and tips to the player as the game world loads

For longer waiting times that cannot be relegated to background tasks, perceived wait is something that we can actively look to manage. By giving people something to do, or at least to read, while a longer process plays out, we can mitigate the negative impact of long waits. Of course, while shorter waiting times are always valuable, sometimes we have to throw our hands up and admit that we’re constrained by the technology we design for. When reducing actual wait times is either not possible or past a point of diminishing returns, look to managing perceived wait.

Respond

Finally, once our exhaustive-yet-exceptionally-well-designed waiting period is over, our interfaces must respond to the action that’s just been performed.

There’s not much to this section that hasn’t already been covered in the earlier exploration of moment endings. An interaction’s response is how we let someone know whether their intended action worked or not—which is an abstraction of telling them whether or not (or to what degree) their goal was achieved.

Responses, like acknowledgments, can take many forms, depending on context and relevance. Success and error flash messages are extremely common (Figure 9-11).

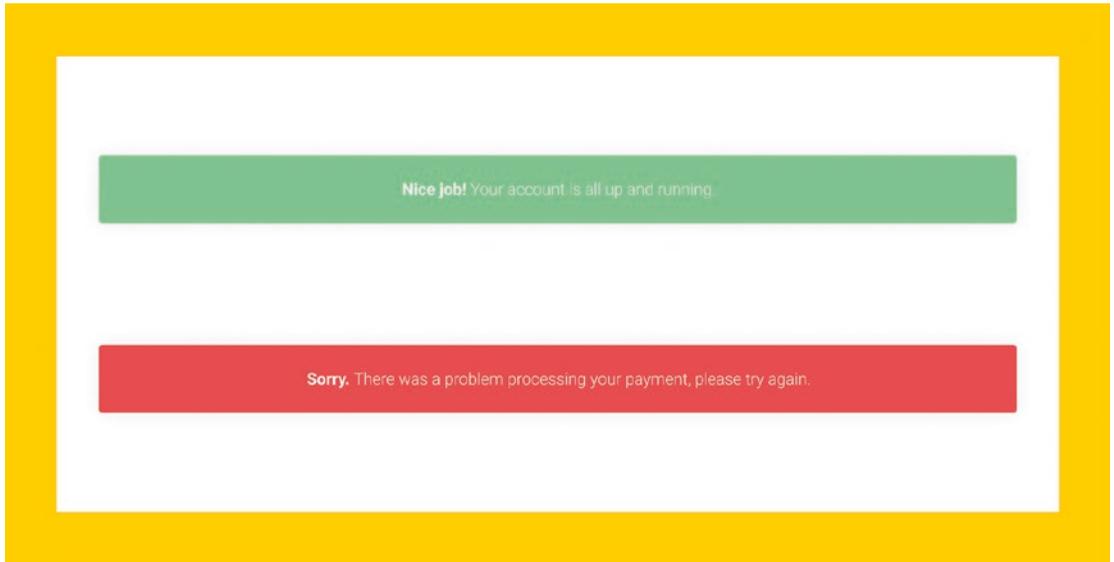


Figure 9-11. *Examples of different messages that we might display in response to an attempted action*

Other, more subtle, forms of response can be beneficial too, especially when space is limited or when the interactions we’re designing are themselves small and subtle. The spinner-inside-the-button from Figure 9-9 could, for example, present a brief success message in the button itself—essentially making for a single-element micro-interaction (Figure 9-12).

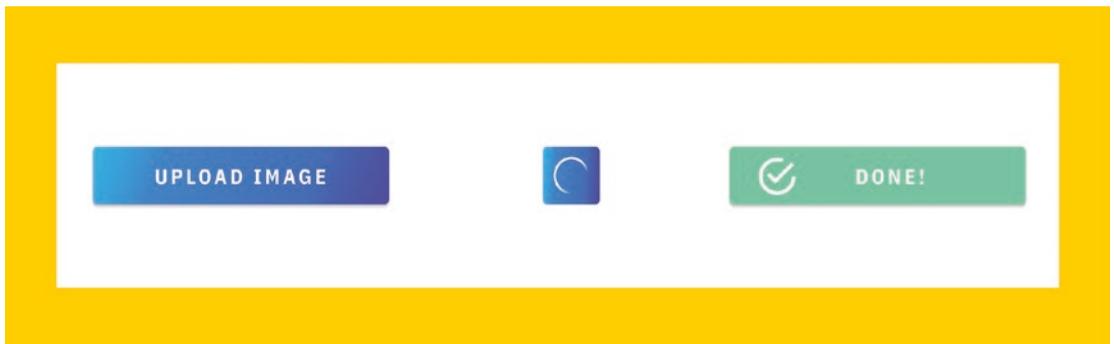


Figure 9-12. *A sequential flow that could quite easily be an entire microinteraction contained in a single element*

Just as a moment's ending should offer some form of reflection, so too should an interaction's response. If the interaction has fundamentally changed state in any way, then finding a way to reflect this will be a major factor in our response. In Chapter 2, we explored how animation can assist greatly in indicating state changes as the result of an interaction. I'd recommend a quick re-skim of that chapter if some of these concepts doesn't sound too familiar!

Finally, action responses will, in many scenarios, need to communicate *new* action possibilities that might have arisen from the performing of the previous action. For example, the response state of adding a new to-do item might need to present new actions for marking that to-do as done, archiving it, or placing it somewhere different in our list. However, I'd like to use the remainder of this section to discuss what I believe to be the most important action, across the board, that should be communicated in any interaction's response state: *undo*.

Throughout this book, I've tried to be as open as possible to different approaches to design, workflow, process, and implementation. As a field, digital design is, comparatively, in its infancy, and I truly believe there is no place for dogma or bullheaded absolutism in such a fast-growing industry backed by an ever-changing landscape. *However*, this is where I break my own rule and espouse an absolute: *Always let people undo their major actions*.

No matter how well-framed, how wonderfully acknowledged, and how extensively optimized our interactions are, nothing can truly prevent mistakes. For too long, digital interfaces laid mistakes squarely at the feet of humans (or, to be less passive, designers of these interfaces deferred blame to the people they're supposed to be making things for) and forced us into constrained and intimidating "Are you absolutely positively sure?" moments to (poorly, I might add) mitigate the erroneous performing of destructive or highly volatile actions.

"Are you sure..." is a framing device. It, in a roundabout way, lets people know that a specific action is not to be taken lightly and that *bad things* can happen if someone performs it. Now, I'm not suggesting that this approach to framing needs to disappear, but I feel it's imperative to acknowledge how much of a roadblock this truly can be to creativity, self-expression, or just simply avoiding a state of sheer panic over a button press. My personal rule is to treat a modal as a lazy design decision unless I know I've explored all over avenues, and to me, "Are-you-sure?" modals are even lazier. No matter how much we convince ourselves that enough defensive measures will stop someone clicking the "Yes-I-want-to-delete-my-entire-digital-history-let's-go" button

accidentally, I promise you that with enough use, someone will click that button. Accidentally.

Preemptive modals are a poor and lazy safety net for *us*—designers and developers. Undo is an empowering safety net for the people we design for.

As we discussed in Chapter 3, an undo feature implies that mistakes are fine and expected. In an interface that allows for or attempts to foster creativity and self-expression, this is absolutely integral, but even in our general, everyday-use applications, it can save us. Since Google Inbox implemented “undo” for message sending, I’ve been saved from probably a dozen truly embarrassing e-mail moments, ranging from calling someone by the completely wrong name to pasting in a link that was absolutely, assuredly *not* the link I was supposed to be sending to a client. While these situations might not have resulted in total catastrophe, written communication on the web is *hard work*, and often even just knowing you’ve sent an e-mail with a pretty major typo in it can weigh heavily on your mind post-sending.

By implementing the undo functionality, we’re implementing a subtle degree of peace of mind. We’re offering a statement that, even when you really do mess up, you can always escape back to the previous state like nothing ever happened.

Zooming In and Out

Jumping from moment-level thinking, where everything revolves around stories and interactions, into system-level thinking, which is much broader and rooted in self-determination, expression, and openness, can initially feel like a strange habit. However, if you’ve ever worked on a design system alongside the actual screens of an interface, you’re already familiar with this approach. And if you haven’t—don’t worry! I’ve found that it’s something that comes naturally with practice.

Essentially, I’m asking you to consider operating at two different levels of granularity simultaneously. At the higher level, we have a more conceptual idea of what our system should be—the various moving parts that might make it up, the personality and character we want it to convey, and the overarching goals and motivations people bring to it. At the lower level, we have a more concrete task of piecing together the discrete moments that make achieving these goals and realizing these motivations are possible. Of course, in between, we have a rather substantial grey area, and many of our tasks won’t fit neatly into “just system” or “just moment” work.

While early-stage documentation of systems and moments can be quite different, as the design process moves along, these notions blur more and more into a single entity: the interface. By the time we start on our prototypes, our system and moments should feel less like two discrete entities and more like one unified whole that is inherently different from the sum of its parts. At certain points, our interface will act as an exploratory conveyor of action possibility. At other points, it will contract and constrain into stories, told through action and conversation, with reverberative endings that lead straight back into a new, modified canvas for exploration.

It would also be remiss of me to not point out that you *absolutely* don't need to start your thinking at a system level. In many cases, starting with your key moments and working outward just makes sense. Key moments and interactions, especially those we expect to be repeated, will represent the core loop of our interface. Getting this core loop right will often be the most important design task we perform.

Similarly, don't overrule starting out at system level because you're worried that you'll sacrifice smaller interactions. For multi-faceted or creativity-focused applications, the overall feel and macro-level state changes will often be the key area of focus before delving into some of the subtleties of moments and micro-interactions.

To return to the movie metaphor—action movies have different priorities to Sundance winners. Horror movies are focused around totally different key points to Wes-Anderson- or David-Lynch-style “artsy” movies. None of these genres is any less representative of the overall art of filmmaking as the other (regardless of what the snobs might yell), but they *are* judged against totally different standards.

Our expectations for an action movie lie in its set pieces. The depth of story and character building are often sacrificed in the name of explosions, gunfights, and car chases. Of course, tone and artistic expression matter—they're just traditionally lower down the list of priorities. Conversely, movies that rely heavily on tone, consistency, and artistic expression will prioritize the more systematic areas of production. Wes Anderson is almost as famous for his stunning color schemes as he is his irreverent plotlines and esoteric dialogue. Many of Anderson's movies lean heavily on creating a consistent and engrossing environment with plotlines that feel more loose and consequential, rather than a deliberate, linear push along a traditional narrative arc through action sequences and climactic moments.

All of this is to say that the scope of our application is going to determine where we place the bulk of our focus. Budgets aren't unlimited and deadlines aren't *completely* liquid concepts. There absolutely will be moments where we compromise one area for

another, and that's *totally fine*. If we have a limited budget and our audience wants *Top Gun*, we'd be well served by starting working on our dogfights and on -liners, not by blowing 90% of our budget on creating *The Grand Budapest Hotel* but with *Planes and Explosions*.

My goal of Part 2 is to present you with some of the tools, metaphors, and abstractions to learn and analyze both from a top-down, system point of view and a bottom-up, moment-to-moment one—it's up to you to decide the balance. Besides, I used up all my dogma with the whole *undo* argument.

CHAPTER 10

Responsible Implementation

Everything we've explored throughout this book has, I hope, presented a counterpoint to the often insidious and exploitative methods of modern tech products. Beyond this, however, I hope the studies, concepts, and suggestions within have—even in the tiniest of ways—contributed to a shift in your mindset or approach to designing and building digital products. I truly believe that by embracing openness and exploration, by creating products that can be used in many different ways, and by treating the workings of the mind as insightful allies rather than obstacles that we must hurdle or manipulate to achieve success, we can lay the foundations for a more sustainable, compassionate approach to design and, ultimately, technology as a whole.

I feel that, in calling ourselves designers (of any kind), we accept a certain implicit responsibility to leave the world a little better off than we found it. Furthermore, I believe that the privileged among us should caveat that responsibility—to leave the world a little better than we found it—for people more vulnerable than us.

The ability of technology to augment human existence is an undoubtedly exciting concept and, with the amount of money and innovation in the technology industry right now, the impact of digital products on the world stands to be monumental. Yet, we still operate in an industry that's rife with naivety, bias, and systemic oppression. As designers, this means that we're often presented with ideas and problem definitions that are rarely harmonious with humans outside of the affluent world of tech entrepreneurship.

To achieve these goals, especially in an industry that seems to operate by its own rules, eschewing ethics in the name of profit under the cover of the mystique of programming and technological innovation, we must ensure that our process serves to elevate those who are far too often underrepresented in our work.

In closing out this book, I'd like to present a few final thoughts on these responsibilities as well as some considerations that didn't quite make sense to include elsewhere. We'll explore how ideas can be challenged through an important workshop and brainstorming technique; how we can tie together much of the previous discussion on open, explorative interfaces by looking at a prime example of choice and openness (spoiler: it's Slack); how homogeneity is unavoidably limiting opportunity; and, finally, tech's role in the advent of a difficult, tumultuous sociopolitical climate (spoiler: it's got a lot to answer for).

“Black Hat” Everything—Purposeful Pessimism

First, I want to explore the idea of using purposeful pessimism and cynicism—as a design tool, not a personality trait.

The concept of a “black hat”—a term I find weird and uncomfortable so will replace with “purposeful pessimism” from now on—comes from *Six Thinking Hats* by Edward DeBono. In DeBono's framework, group participants “wear” a specific hat and are requested to respond according to the hat they're wearing (yellow is the “optimistic” hat, for example). The black hat is the hat of the pessimist. Its wearer's goal is to provide some much-needed cynicism to a brainstorming session or workshop.

While I personally believe DeBono's framework doesn't translate well to design planning, and I don't want to dwell on such a framework, the concept of purposeful pessimism is an intriguing and important notion for modern design work.

First, purposeful pessimism is a point where everyone on a team is given permission to be a little bit negative. In the sunshine and lollipops world of tech startups where being unhappy is a cardinal sin, this can be extremely cathartic.

Second, it's an antidote to hubris.

Most people think their ideas are good. It's just part of who we are. We're more attached to things of our own creation (like the IKEA effect, discussed back in Chapter 1, where we prefer something we've made ourselves, even if it's objectively horrible compared to a pre-made version). This can often lead to some really tricky degrees of bias in a project. By having an agreed-upon session, or an agreed-upon part of every session, for purposeful pessimism, we can go some way to removing this stigma of negativity being bad. It's even better when the person who has the idea is able to think objectively as a result of pessimistic discourse.

Most importantly, however, it forces us to consider vulnerable, divergent, and underrepresented members of our audience—something that technology companies are notoriously bad at doing.

Purposeful pessimism is a simple, but nuanced method to brainstorming. My advice is to work it into every important brainstorm you have—especially anything that involves new feature discussion. The rules are simple: you find ways that a proposed implementation can fail.

Failure, in this sense, is not “missing a KPI.” It’s a fundamental failure at serving a goal or solving a problem. It also denotes allowing means of exploiting or using a feature to cause others harm.

When performing this kind of work, it’s important to make sure that all participants understand that, just as brainstorming is a loose and fast-paced exercise, so too is this pessimistic analysis. Not every suggestion for how something might fail will be valid, and any proposed reasons of failure don’t diminish an idea—they just provide important context that would otherwise be missed.

Once everyone is suitably prepared for slightly hurt feelings, conducting a pessimistic analysis should take the form of a loose conversation with a Post-it backdrop. During this discussion, it’s important to think about actions and attitudes. Actions will be the easier of the two to brainstorm, but attitudes can often be much more valuable, depending on the type of product you’re making.

As an action example, take Twitter’s “report” functionality. Reporting allows someone to report another person’s account, stating a reason for their report, and optionally including a number of tweets to support their claim. Approaching this pessimistically, a potential misuse of the feature could be in the mass reporting of a list of people another person simply dislikes. This potential problem would then feed in to the finer details of implementing reporting. If we were considering automatically putting a reported account on hold, for some strange reason, this would present a glaring potential abuse of our proposed functionality. Anyone could misreport someone and get them instantly suspended.

Thinking of ways in which your ideas could encourage negative attitude or behavior is a slightly trickier approach. A common example would be platforms that include “up-voting”—a feature where people can up- (and sometimes down-) vote a particular story, comment, or post. Let’s say we were working on a product that allowed people to ask a community a question, and they allowed members of that community to up-vote answers to that question to help deem the most applicable answer. The optimist in us

might suggest that rewarding question-answering can create a friendly, helpful attitude. However, it can also create one of snobbery, and holier-than-thou, and one-true-method approaches. In this case, we'd want to greatly consider how we can foster the right kind of attitude, both when asking and answering questions—a difficult concept, for sure, but one that could make the difference between an empowering and collaborative knowledge base and a toxic cave of well, actually trolls.

Purposeful pessimism shouldn't discourage you or your team from going ahead with a feature that you feel is integral to your product, but it should at least contextualize both the feature and your overall problem space. Often, too, it's the seemingly most innocuous features that can go horribly wrong—just ask Facebook.

Facebook's Celebration of Death

Facebook has seen its fair share of tone-deaf features—some, in particular, that would have benefited from *any* form of foresight that wasn't blind optimism such as “On This Day” feature and, more infamously, their “Share a Memory” and “Year in Review” features.

These features involved Facebook algorithmically selecting one or more photographs, videos, or posts from a profile and turning them into an often cute, often quirky shareable image or video. Most of the posts they generated were what many would deem as “fun”—cute selfies, pictures of dogs, all to the tune of typical stock, inoffensive, upbeat music loops. Except, when they failed, they failed *hard*.

One person had a supposedly fun, dancing illustration superimposed on their mother's grave. Another was shown a slideshow of their house burning down. Dozens of people have been prompted to “share” glamorized or decorated photographs of dead relatives, dead pets, and abusive ex-partners. These are clear features where emotional harm was caused through a poorly thought-out feature. They represent prime examples of things a healthy dose of cynicism could have preempted.

These issues might seem almost fine to many people. They're *just algorithms* at the end of the day. They can't be mean, or racist, or misogynistic, or transphobic. They're just 0s and 1s, right? Wrong. In these kinds of cases, no one but the team who designed and built the feature into the product is to blame. Whether a feature messed up due to human error or a poorly optimized algorithm, that feature went through multiple people's mental spaces before it was launched. Either no one saw how it could fail, or the people who did neglected to speak up. Normalizing structured and purposeful

pessimism in the name of providing yourself, your colleagues, and your team with a deeper understanding of the problems you expect to face—as well as the platform to speak out—is a core responsibility of a designer.

I'm sure that when Twitter was founded, its creators didn't expect it would one day become the virtual stomping ground of modern-day neo-Nazis. However, there's a fuzzy chain of events, starting with a failure to clearly define the parameters of acceptable use for their platform, passing quickly by a categorical failure to deal with bot accounts in the build up to the 2016 US presidential election, and ending somewhere right about now, when Twitter CEO, Jack Dorsey, has to field questions about how his platform is being used to mobilize and recruit fascist supporters of a white ethnostate. Hindsight, as they say, is 20/20, but a small part of me wonders how different Twitter would be as a social space today if just a little more foresight went into dissecting how the platform could be exploited—something that the Twitter team should be no stranger to, given the platform's historical role as a hub of exploitation through underhand trolling tactics and long-standing, targeted harassment campaigns.

This kind of pessimism can, and should, extend to your success metrics, too. Product teams that work almost purely in response to data are often the main perpetrators of this lack of foresight. When your product's value is intrinsically tied to your engagement metrics, it's extremely easy to fall into the trap of neglecting to remember that each data point you see represents a real person, engaging in a real interaction with your product. With this mindset, it's no surprise that features that “perform well” (that is, increase numbers that are arbitrarily deemed to be good or decrease those seen as bad), but are often open to exploitation, manage to work their way into products. If your work is going to be judged purely on measurable data, then make sure there's a “...without...” caveat in there—something simple, like “Increase monthly average users without them all being members of an online fascist collective” will do.

High morale and momentum are great tools for a project, but there comes a point when blue-sky thinking needs to meet down-to-earth reality, and the reality is that anything that can empower and encourage an individual or a group can almost always be used to denigrate and oppress another. When you think about this, you start to explore your problem space in a specific way. When you examine issues of cultural sensitivity, you start seeing ways your ideas can be used for nefarious measures, and if you give a damn, you add these problems to your understanding of your underlying system.

This can be hard to swallow for many people, especially founders who believe that their ideas will fundamentally benefit the world, but every major tech company and product has its issues. Facebook has launched tone-deaf features on many occasions. Airbnb hosts have been shown rejecting the patronage of people of color while accepting white visitors for the exact same vacation conditions. Twitter is full of actual Nazis. Almost every major tech product out there, *especially* those that involve elements of social interaction, has failed its vulnerable customers to varying degrees. Our job is to, wherever possible, provide the tools to at least attempt to discuss these potential failings.

We won't make any friends while we do it, but if we, our colleagues, our bosses, or our investors want to claim social consciousness, we have a duty to explore this side of our products.

The Role of Choice

Ultimately, design's role in providing platforms for self expression and creative problem-solving gives us another variable that we must consider: choice.

The discipline of design is that of balance. We're constantly, throughout our process, trying to find the right combination of qualities to suit our goals. Whether that's through something as simple as settling for a darker text color to meet contrast guidelines, or whether it's through something as complex as limiting the moving elements in a virtual-reality experience to combat motion sickness—we're faced with dozens, often hundreds, of decisions every day that all boil down to finding moments of balance.

The very presence of a need to strike balance shows that ours is a practice of compromise. Choice, then, is yet another component of design that needs balancing. On one end of the choice spectrum you have limited, single-purpose products—those that require nothing but a single interaction to serve their purpose, such as Amazon Dash Buttons (Figure 10-1), where a single button press can order a specific product to be delivered.

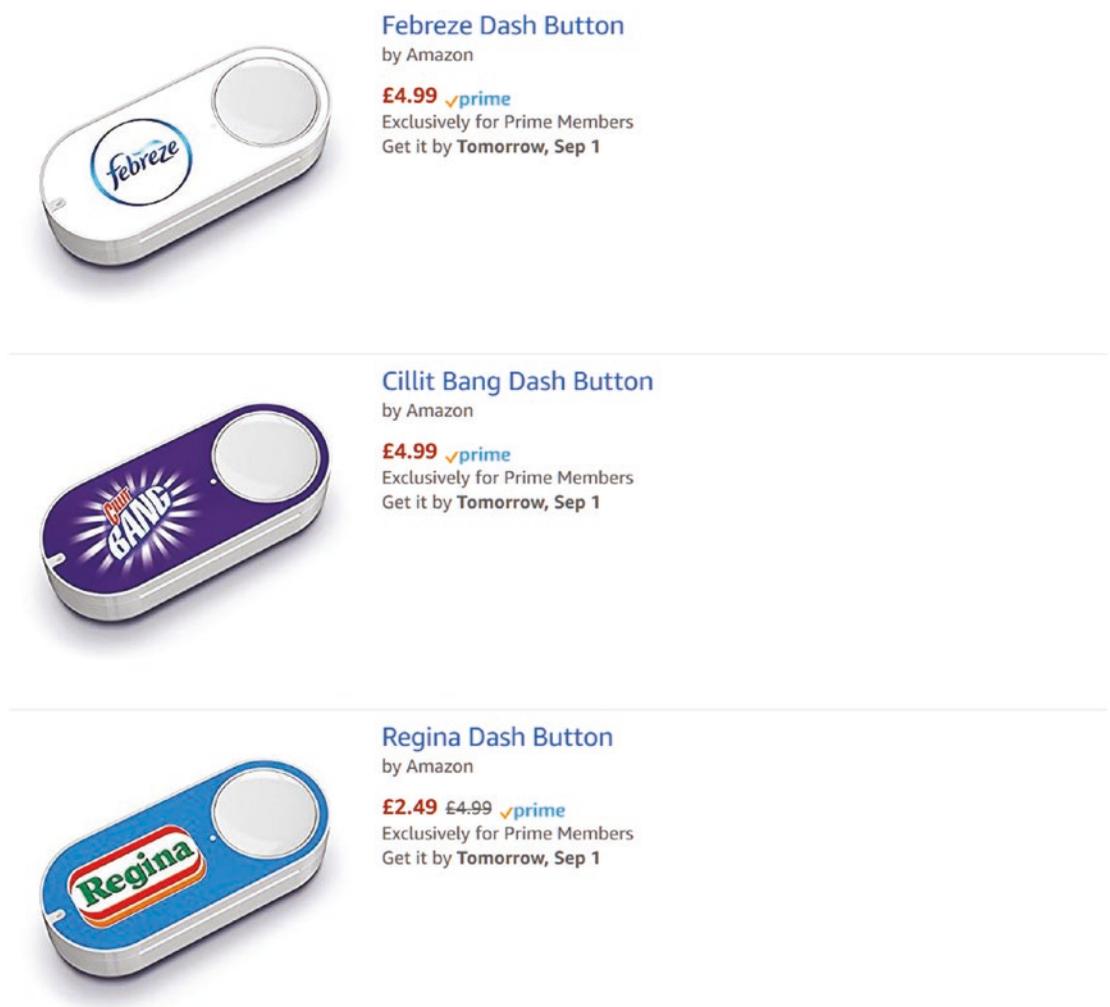


Figure 10-1. Amazon Dash Buttons—a true single-purpose product

I call these kinds of products “spiralizers” after the wonderfully single-purpose kitchen utensil of the same name. The glamorous world of the utensil industry has, in recent years, become rather enamored with products that Alton Brown—the famous chef, author, and television host—has taken to rather disparagingly labeling as “unitaskers.” Unitaskers, as the name suggests, exist to perform one task. Whether the marvel of innovation is the pizza scissors, the wonderfully esoteric avocado slicer, or the bedrock of any good kitchen—the spiralizer, the influx of gadgets that sell themselves on the basis of doing one task really, really well has left us with an abundance of cheap, tacky tools in our kitchens.

Now, there's likely plenty of reasons why these utensils sell well. They're foolproof, they're often cheap, and they're almost always exceptionally well advertised. However, they don't do anything that we couldn't do with a simple, well-made chef's knife. If we spend a little time learning some basic knife skills, we suddenly have one tool that can confidently relegate an entire suite of unitasker utensils to that one junk drawer that everyone has in their kitchen. While I'm loathe to present myself as a culinary snob (I lost the tip of my thumb grating a beetroot, so who am I to talk?), there's something about having a small set of tools that you're comfortable and confident in using for a vast array of tasks.

I believe that while single-purpose apps are still going to be made in droves, one of the true joys of design lies in creating a product that can be used in unexpected ways to create things or solve problems that we—as their designers—never even envisioned.

Part of this comes from a much-needed rescinding of control. This apparent desire to persuade and nudge people down linear paths that I've spent a large part of this book attempting to assuage is something that we, collectively as an industry, need to look to move away from. Too often, we sacrifice choice and openness in the name of perceived surety through constraint. If our goal is to create an environment of possibility and exploration, we need to get comfortable with randomness and idiosyncrasy.

Once we've broken free of the constraints of linearity—and checked our ego at the door in the process—we can start to look at choice as a variable in our work.

Choice abundance represents the opposite end of the spectrum to the aforementioned unitaskers. At this boundary, the most empowering example I can offer is musical instruments, or to get more abstract, music as a system unto itself. Musical instruments offer us an almost infinite palette of actions with which to express ourselves. The emotional landscape of music is an immense, somniferous blanket that enshrouds us all. From upbeat dance tracks to solemn hymns to exhilarating orchestral movements to angsty emo to visceral hip-hop to whatever the heck dub step is, music's role as an encapsulation of human emotion lies in the abundance of possibility—as a system of loose rules and undulating schemas it provides.

Musical instruments are still built to specifications. They're still *designed* to meet certain goals. A standard piano, at the surface level, is a tool for performance and composition that allows 88 different notes to be sounded by the pressing of their corresponding keys. However, some 300-plus years into its existence, people are still finding millions upon millions of unique ways to use a piano. When we sit down to play, we're not limited by the tool in front of us, but our mastery of it. And we strive to

improve because we *know* that there's more we can achieve. Playing an instrument, or creating music in any way is a journey. We embark on this journey for myriad reasons, but we stick with it because we know that there's still more to come. We're able to look at 88 keys, or six strings and some frets, or a computer program full of soundscapes and, intuitively, know that we'll never exhaust the possibilities that these things present.

Closer to home, we see variations of this every time we do our job. Design tools let us express our ideas—some of which can, and do, change the world—through rectangles, lines, and shadows. While we might have our processes, cheat sheets, and shortcuts, every design we create is something that never existed before we laid it down on digital canvas. We're living proof that a blank canvas and some rudimentary tools can lead to creations and moments that transcend the apparent constraints of the tool or medium we operate in.

Why, then, when we're staring at these tools for hours upon hours every day, do we seem to favor selling people products that sit far from the realms of divergent thought and far closer to whimsical unitaskers? Why do we witness, every day, humanity's willingness to improve often abstruse skills in the name of self-expression and problem-solving yet deem it suitable to leave them with the digital equivalent of an avocado slicer?

I think we do so, first and foremost, because it's *easy*. Single-purpose applications are easy to make, easy to sell, easy to measure, and easy to iterate on. That all adds up to an inherently marketable product. Considering the still far too homogenous makeup of the technology industry, the fact it exists as a self-serving economy, and the fact that venture capital is, at the time of writing, still the de facto means of valuating a company, you find that you have the perfect recipe for an echo chamber.

To flip the discussion on its head, products that limit choice usually arise from narrow problem definitions. Narrow problem definitions arise from a shallow pool of experiences. Shallow pools of experience arise from homogeneity—a lack of diversity that means innovation and potential impact is stifled almost by default. When an industry decides that plucky upstart founders who solve problems and get it done belong on a pedestal, when the distribution of venture capital is based on an almost instinctive appraisal of the usefulness of a product, when a gross misinterpretation of the satire that is meritocracy becomes the prevalent religion to its core demographic of white 20-something Californians, then the problems that get deemed worthy of solving tend to have a very noticeable, very white, *very* upper-middle-class feel to them.

Note While tech’s diversity struggles aren’t the *only* reason we seem so content to disregard self-expression and creativity in the name of control and dominance, I don’t believe it’s a coincidence that a predominantly white, predominantly male, predominantly affluent cohort presents as a hegemony with these apparent values.

Choice is an abdication of power. Any time we present someone with more choices, more ways of interacting with a system, more opportunities to experiment, more problems with which to aim our product at, we’re empowering them. While we should strive to reduce noise, limit pointless features, and be ever cognizant of the attentional impact of our work, choice, openness, and possibility are also important factors.

That’s not to say that every possibility should be presented outwardly as a decision to be made—decision overload is just as problematic as information overload, after all. There’s a huge difference between offering a useful means of customization or expression and passing the buck. Moreover, the message here is simple: design products that can be used in multiple ways for multiple things. Stop being precious about solving one, clear problem and start thinking about the diverse range of problems and people that exist within your problem space.

It’s easy to claim, “This product fixes your problem,” with the caveats that, first, someone needs to be experiencing the exact same problem its creators claim to have solved and, second, the product is used in one set, agreed way. All this does is perpetuate the current climate of narrowly defined products that work for a narrow audience, impacted by the narrow-minded decisions of a homogenous power structure. Diversifying action possibility—adding more choice—maximizes not only the range of problems that your work could be applied to, but also the chance that your work can be used in new, inspired, and unintentional ways.

This diversity of problem definitions doesn't just pop up by accident, nor does it come as a result of learning or knowledge. It comes from including a broad range of experiences, sexualities, ethnicities, gender identities, and cultures throughout our process. The manifold dangers of homogeneity are still all too rarely discussed in technology spheres, yet nothing limits a product faster than a too-narrow notion of the problems it's supposedly solving and the people who supposedly experience them. This, again, is why diverse recruitment and compassionate, empathetic interview methods are so important.

If we buy into the idea that, in fact, we humans are a creative and social species with wide-ranging intrinsic motivators as opposed to black box, behaviorist machines fueled by dopamine, then we know that supporting self-deterministic activities—creativity, autonomy, social belonging—all but *requires* a diverse range of possible actions and applications.

We've explored fragmented examples of smaller parts of this self-deterministic product approach—see Photoshop's excellent onboarding or Duolingo's joyful (yet flawed) displays of progression and mastery—but I've yet to present a product that I feel sums this up. While the perfect example of a self-deterministic product doesn't exist (after all, humans are an erratic and eclectic bunch), I feel the best example, due in part to its sheer dominance of the tech industry, is Slack.

Slack: A Case Study In Open Design

At its core, Slack is a collaborative messaging application reminiscent of IRC (Figure 10-2). It's quite clearly marketed for companies and teams, with its most common use being that of a "virtual office." Dig a little deeper, however, and you'll see that Slack is as much a tool of self-expression as it is a means of keeping your colleagues up-to-date.

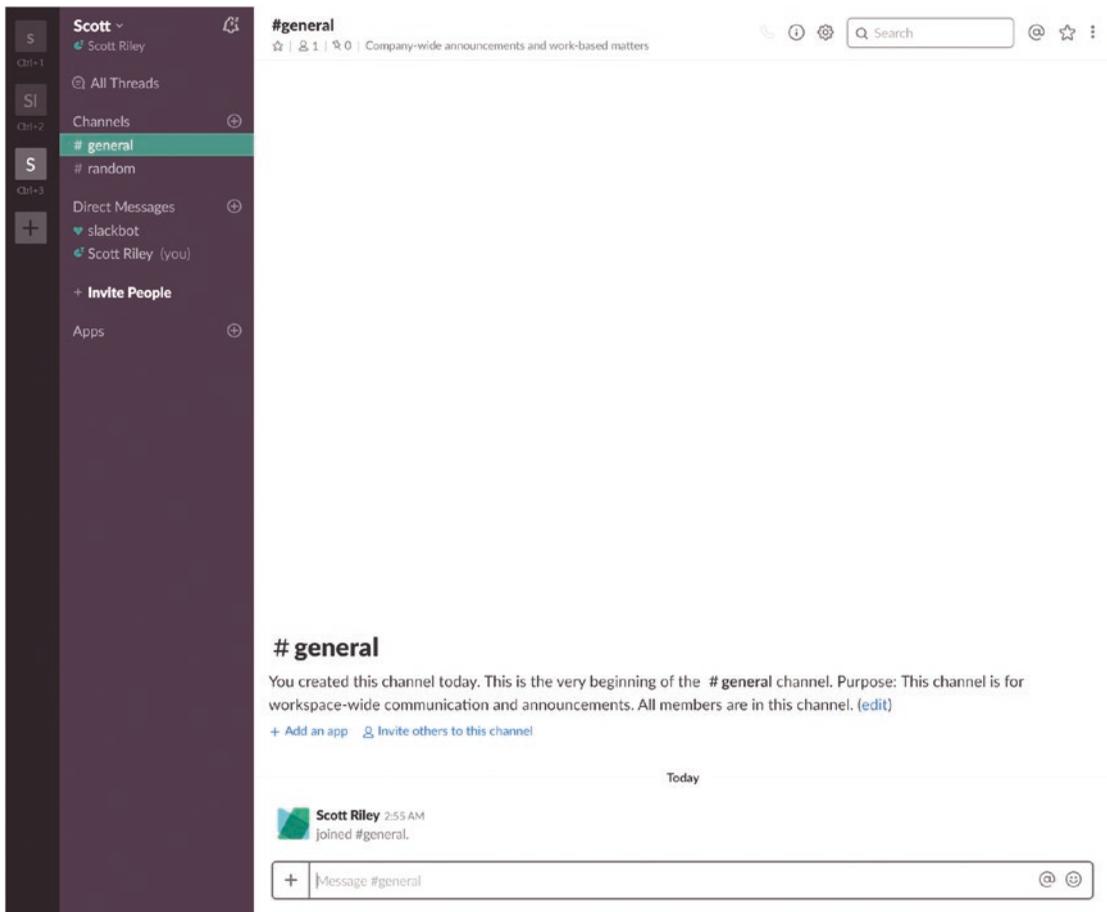


Figure 10-2. Slack

In addition to being a good example of explorative design done right, Slack has more than a few things going for it. First, it’s valued to be worth *a lot* of money—\$7.1 billion at this time of writing. Second, Slack’s CEO, Stewart Butterfield, has a refreshing outlook on the role of technology in people’s lives—stating in a *Thought Economics* interview, “At Slack, we’re building something that’s meant to augment the way in which a group of people, a team, works together. We’re not trying to solve one specific problem for one class of worker, or one domain of expertise, but build a very broad tool that ideally augments the way people communicate” (2018).

While dozens—if not hundreds—of articles, blog posts, and podcast episodes have attempted to dissect what makes Slack so good (or at least so highly valued), I want to focus specifically on the role choice and control play in making Slack an enjoyable daily tool for so many people.

To understand *why* self-deterministic design is so apparent in Slack, we must first explore the core system it's based around: collaborative communication.

Social communication, reciprocity, belonging, and how we string sentences together are all factors personal to us. Language is the most intriguing, expressive, and important system humanity has ever produced and, as more of our lives move into digital realms, so does our communication. With this comes a growing need to find ways to express ourselves through writing. Slack has found itself, alongside social media tools like Facebook and Twitter, at the core of this movement.

The team behind Slack has treated communication—and just as importantly, the people who use Slack to communicate—with the nuance and care that it deserves. When you understand that communication and collaboration are core parts of not only how we express ourselves, but also how we make sense of our self, how we find and foster belonging, and how we build relationships and share knowledge, you realize that attempts to limit communication in order to fit a nice neat problem definition are ill-conceived at best.

While the core proposition of Slack—channel-based, team-centered communication and collaboration—provides a large chunk of its value, Slack's team has struck a goldmine of goodwill and delight that is rarely afforded to digital products with its approach to implementing tools for further expression and customization.

One subtle feature that exemplifies this is Slack's emoji reactions (Figure 10-3). Exploring the linguistic and expressive qualities of emojis could quite easily be its own book (or it's own absolutely shocking animated movie) and, while many people take umbrage with their proliferation, I personally love them. They are a wordless, uniquely expressive means of iconic communication. When someone posts a message—let's say, a quick “good luck” message—in Slack, as a replier, we have a few potential options. We can ignore it, which would be mean; we can type out a thank-you message, which would be nice; or we can react to it with the sparkle heart emoji, which is canonically the most obvious expression of eternal gratitude.

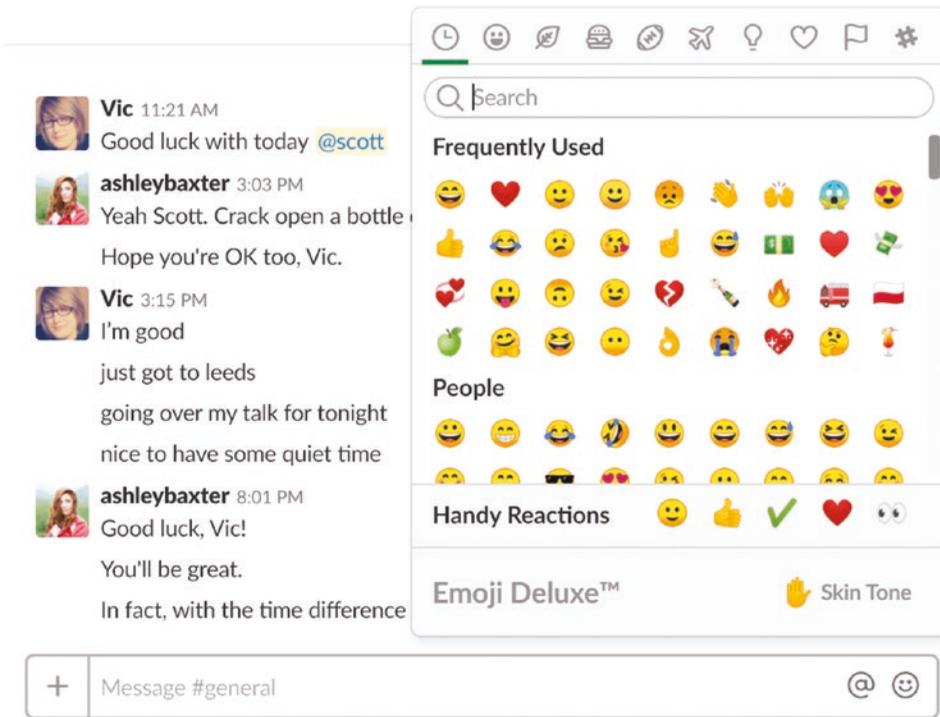


Figure 10-3. *Reacting to a Slack message with an emoji*

While a text box in which to reply “Thank you!” would have been more than enough for most people, being able to attach a reaction to a specific message allows for a more nuanced, alternative means of communication. Slack gives us a facsimile of how, in real life, we learn the ways those around us prefer to communicate and are often cognizant of their preferences, vulnerabilities, triggers, and emotional needs. Perhaps if a conversation has already moved on—after all, Slack is designed for asynchronous communication—we’re able to still let the original poster know that their message is meaningful.

While initially intended as a means of providing a wider range of reply methods, emoji responses have been used in multiple creative ways by Slack users. Some use them as a means of quickly bookmarking posts, for example, with a star emoji (while you can “pin” a message on Slack, it pins it for the entire channel). Slack later implemented a means of auto-sending messages with a specific emoji reaction to a particular channel. Furthermore, as emoji reactions “stack” quick straw polls can be conducted—“Lunch vote! React to this message with taco, sushi, or burger to cast your vote” (Figure 10-4).

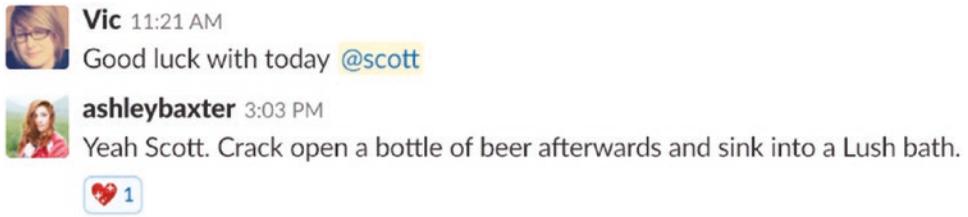


Figure 10-4. Slack’s counter next to emoji reactions

This is a perfect example of providing a feature in an open and explorative environment, only to see it be used creatively for things that we may never have imagined or intended.

On a more holistic level, Slack is also inherently customizable. This ranges from the aesthetic, such as being able to change your sidebar theme, to the important and considerate ability to disable notifications for certain time periods and to mute certain channels. Setting a “Do Not Disturb” schedule in Slack is a painless process and removes one of the largest, most detrimental moments of such an application: unwanted notifications.

Slack’s customization and extensibility doesn’t end there, though. With custom emojis uploading, short statuses to inform people what you’re up to, granular notification control, a huge array of third-party integrations, and much more, Slack goes above and beyond what most products can do so that it can be tweaked to fit around the day-to-day lives of real humans.

This openness ultimately culminates in Slack’s suitability to an extremely broad problem space. I ran almost the entirety of a business through Slack for over a year—everything from client enquiries through to project management went through Slack and its integrations. Many people run Slack communities where the focus is still on communication, but among people who share interests and want to have discourse away from the toxicity that is commonplace on many open social platforms. And that’s just scratching the surface. There are examples of everything from personal journals entire CRM and helpdesk solutions built entirely on Slack, some integrations, some web hooks, and a whole bunch of personal creativity.

This level of openness doesn’t come through happenstance or shots in the dark. Slack is vocal about their diversity initiatives, is leagues ahead of the curve in diversity and representation in technology (there will always be more work to do on this front), and consistently espouses the virtues of operating within a broad problem space and

treating people like, well, people. That a business communication tool has reached the heights Slack has, while consistently putting out a message that still somewhat flies in the face of common tech habits, is the best advertisement I can think of for open and explorative systems.

Finally, even with its multiple layers of customization, legions of integrations, and a constantly evolving feature set, Slack remains inherently usable and remarkably frictionless when coming to grips with it. Clear, concise guidance through key “first” moments gets out of the way when we need it to while it has us actively interacting with elements when we need them (similar to Photoshop’s guidance we saw in Chapter 3), helping us quickly and contextually grasp Slack’s primary features (Figure 10-5).

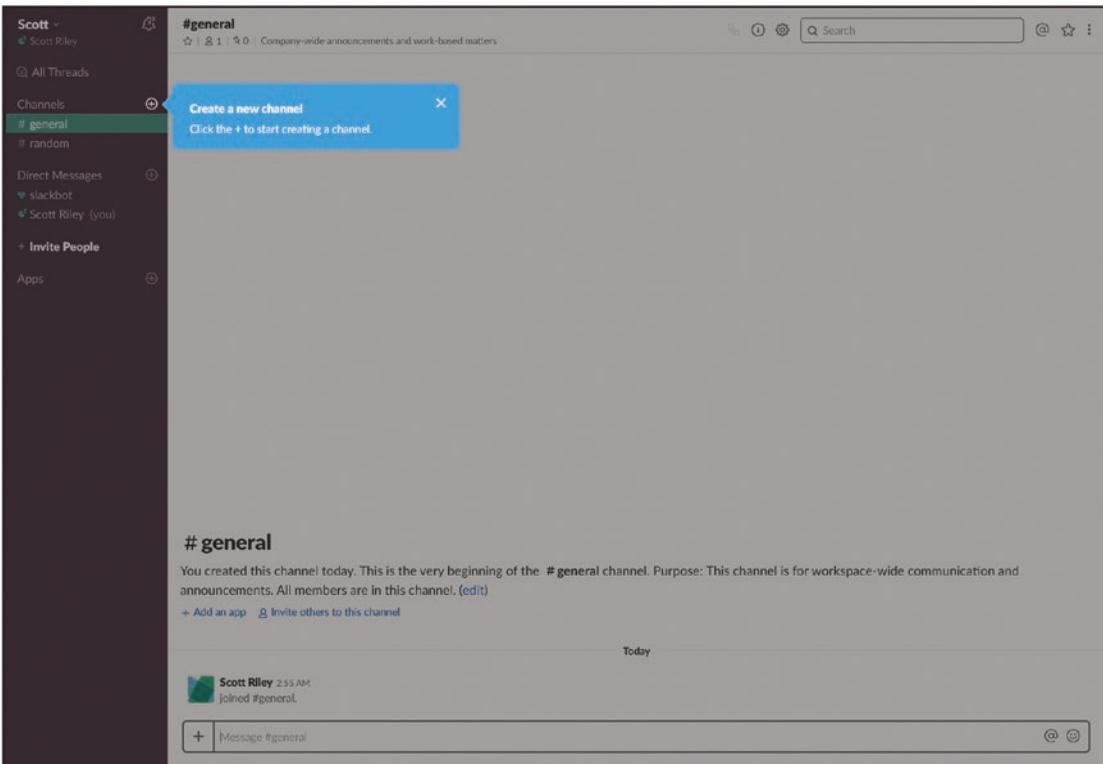


Figure 10-5. Slack’s in-app guidance. Notice how we’re prompted to interact directly with elements in a full interface context

Slack’s riffing on the IRC-style “#channel” structure is a smart way to leverage preexisting mental models (however niche they might be!) and its “text-input-at-the-bottom-message-list-filling-up-the-space-above” layout is reminiscent of almost all

dedicated messaging applications—toeing the line between convention and innovation almost perfectly.

Smart defaults—a channel already created for us, a “Do Not Disturb” schedule pre- set—and genuinely useful empty states make orienting ourselves through our first few minutes with Slack a refreshingly empowering experience (Figure 10-6).

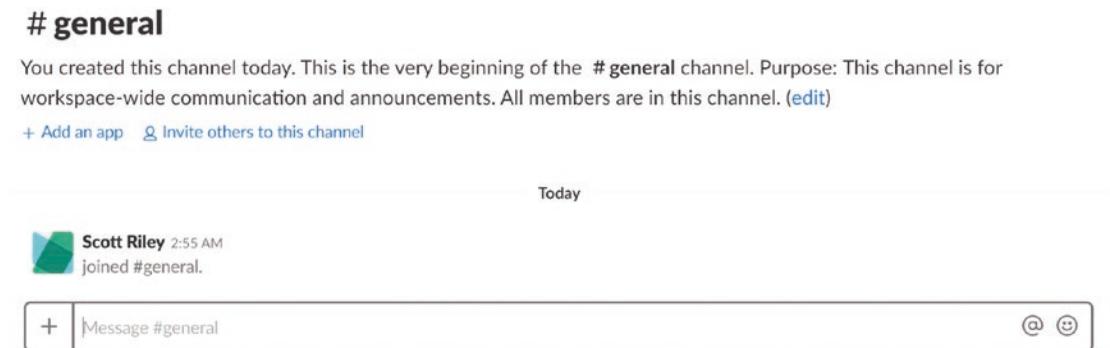


Figure 10-6. An empty channel in Slack provides us with context, lets us know what we can expect from this channel going forward, and frames some key actions we might want to perform

Even an empty Slack seems to invite us to explore it, without ever feeling nagging or persuasive. When you notice the small “slackbot” name in Direct Messages, intrigue alone might lead you to open up a conversation with Slack’s robot mascot, where you can get familiar using Slack by, simply, using Slack (Figure 10-7). As an aside, the language used by Slack’s bot is exceptional, instantly framing our conversation as with a bot, letting us know how best to “talk” to it, and providing an obvious link to a different source of help should we get frustrated with the chatbot format.

Hi, Slackbot here!

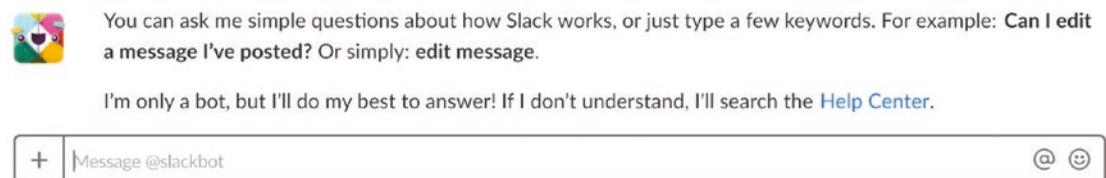


Figure 10-7. Slackbot, Slack’s robot mascot, is here to help us out as we become familiar with the product

Slack offers a valuable insight into what happens when products focus heavily on appealing to and augmenting human nature rather than doing battle with it. By acknowledging that communication is nuanced and fluid and that humans are irascible and unpredictable, and by striving to create tools that embrace these concepts, Slack stands out as a clear example of the success that can be found in relishing imperfect systems and abdicating control.

People Are Not Edge Cases

At the time of writing, in an age of political turmoil and incessant toxicity, when the founder of the world's most-used social network has to dodge questions about his company's role in the viral spreading of false news articles that decided the fate of the most controversial US presidential election ever; when the CEO of another company has yet to find a solution to an open rise of neo-Fascism on his platform; when companies who would otherwise be regulated, unionized entities were it not for their ability to ghost through entire markets under the umbrella of "technology" companies; I'm reminded of our assumed role as designers. We're, supposedly, "the voice of the user." Concepts such as delight, experience, seamlessness, and innovation are losing meaning. Sooner or later, design, like technology, will have to answer for its role in this tumultuous period, and I fear how few of us—myself included—will be able to look back and still hold our heads up high.

Somewhere in the pursuit for unfettered growth and profit, design's perceived role has shifted from democratization and empowerment through technology to invasive and manipulative vehicles of pop science at the behest of founders and investors. At a point in time when democracy is failing its most vulnerable, where racism and jingoism permeate every crack in the facade of social media, where the voices of those calling for a white ethnostate are amplified exponentially on a platform that is grossly incapable of suppressing hate speech, where online tools are used in months-long hate campaigns, I feel it's an important time to revisit our ethical responsibilities. While it would be absurd to lay the blame of such sociopolitical malaise squarely at the feet of design and technology, I feel that, somewhere, somehow, our industry has dehumanized people at a time where we need most to see the vulnerabilities we design for. By breaking away from the common trend of self-serving, narrowly defined problem spaces, by seeing people—all types of people—for who they are, we can look to bring humanity and compassion back into our process.

People are messed up. They do weird things to themselves that fly in the face of survival instinct. Some of us harm ourselves because of chemical imbalances in our brains. Some of us have eating disorders. Some of us have drug addictions. Some of us can't answer the phone or hold a conversation with a stranger due to anxiety and panic. We're human beings and we grieve and we screw up and we get excited about silly things and we're sometimes just as likely to sabotage ourselves as we are to celebrate or reflect positively on our life.

Until we factor this into our work by showing compassion, researching properly, educating ourselves on our privileges and biases, and until we earn our ridiculous salaries, day rates, egregious company perks, and the pedestals our industry so adores by actively designing and building for vulnerable, oppressed, and divergent human beings—how can we possibly say we're making this world a better place? By releasing yet another product that is only usable by those who meet our ridiculously narrow defaults, by those who are capable of the cognitive cost of “normal” attentional faculty—whether through not experiencing poverty, anxiety, sociological impact, or neurodivergence—or by those who simply fit nicely into our abstracted assumption of what makes for a normal human being, what are we *really* contributing?

Homogeneity's grip on the technology industry is something that needs to be tackled from multiple angles—and design is but one of these. Many of the tech darling products we aggrandize to the point of hero worship boil down, in essence, into a category of “things white dudes' parents used to do for them until they moved to silicon valley” (a functional category, if you think all the way back to Chapter 1). These products make it easy and cheap to get rides to work, while capitalizing on systemic oppression and non-unionized labor. They make it simple to get dinner on the table, while heaping more pressure on an already overworked and underpaid industry of service and delivery workers. They make it easier to get your laundry done, easier to find a place for your fourth vacation of the year. Some of them even make that grueling last half mile of your trip to work slightly more bearable by polluting the streets with electric micro-scooters.

There are many trade-offs to this incessant pursuit of lukewarm innovation, though. An astonishing number of products that we hold up as trailblazing innovators (or *unicorns* if you want to be as insufferable as the tech press) exist because they've found technological solutions to exploiting loopholes. Whether that's “technology” companies that are able to drive down prices due to dodging government regulations, CEOs with the charismatic depth of an STD claiming anti-union rhetoric as the key to success, the tone-deaf social media platform that refuses to do anything about its Nazi infestation lest it

damage vanity metrics, or just your average everyday monolithic company paying 1% of the tax it's supposed to—the apparent need for tech to subvert humanity and democracy to succeed is a vile byproduct of untempered attempts at innovation—using nothing but a framework of rhetoric, blame deflection, and a tone-deaf agreement that certain, arbitrary metrics are somehow directly translatable into fiscal value.

Modern-day tech capitalism relies on an onion-skin model of subversion. Everything from where the money comes from to pay salaries to the Rosetta-Stone-needed levels of legalese in terms and conditions to the general public's willingness to gloss over the systemic oppression that is as integral to the bedrock of tech innovations as the server stacks and app stores they exist on (“Well, most of my clothes are made in sweatshops, so why should I care that my taxi driver gets paid an unlivable wage?”) feeds into this implicit, societal permission we give 20-something tech dudes to “go off and innovate.”

All of this leads us to a landscape where social media platforms suddenly become arbiters of perceived truth and conduits of political clout—where the companies that make the devices, operating systems, and browsers on which we live incrementally more of our lives are responsible for everything from waking us up in the morning to helping us maintain our mental health. The more this perpetuates, the more of a responsibility design becomes, and the harder we must work to serve the needs of real, fallible people.

Design, then, leaves us with a choice. As the interface between an underlying system and the model of that system in someone's head, we can either introspect our system through the lens of humanity, or we can attempt to contort and convince and persuade humanity to interact with our system in ways we desire. The former is, simply, our job; but it's hard, and we'll face battles and make enemies, and we'll likely get pushed back from many sides. The latter perpetuates the rather dystopian notion that humanity—including our cognitive faculty and even our very concept of self—is subservient to technology. That being *hooked* on apps, feeling the *dopamine squirt* of notifications and generally being *the good user* is simply the price of admission into a world of possibilities. At least, a world of possibilities if you're a well-off white person in or around a major city with a decent tech hub and own a £1,000 phone.

Yet, as doom-and-gloom as that all sounds, there's still very much a place for design that eschews exploitative nudging and persuading in favor of self-expression and mental empowerment. Slack, one of the most impactful digital products of all time, is testament to this. With even a surface-level understanding of the mind, of cognition, and of intrinsic motivation, we're able to see tech through a mindful, compassionate lens—one

where empathy and exploration exist by default, where inclusivity and a refusal to punch down are as important principles as the aesthetics of an interface or the reliability of an API server.

Design should exist to serve humanity. Real humanity. Not some essence or abstraction of humanity, not the privileged, amalgamated assumptions of four white people sitting down to solve the next mundane problem in their lives. But real, fallible, vulnerable human beings. Humans whose daily struggles and sacrifices in the face of a world and an industry that shows them nothing but apathy and disdain deserve more than to just being a casual footnote on a persona, or a box-checking exercise in our research.

Above all else, this is the essence of this book. Not to make our jobs easier, not to present yet another design framework that gets replaced after a six month trial, but to offer an antidote—however small—to the incessant push for interaction design to be persuasive, or convincing, or manipulative in any way. If this book has whipped you into a stupor and has you wanting to run out, all guns blazing, to try and fix the very fabric of digital design, then awesome! Thanks for the passion. However, if it's taught you even one single concept that you can bring into your next project to make your work a little more mindful, then that's just as good. The monolithic problem of design as a force for profit above all else is one to be chipped away at, from many angles, for a long, long time.

This stuff also takes time and energy, and not everyone has a large enough surplus of both to implement a whole host of changes in one fell swoop. I totally get that. Just as much as I face a task beyond the scope of this book to attempt to let these ideas reverberate through and beyond my little corner of the design world, so do you, dear reader, have a similar task to effect change in yours. The people we work with will be just as fallible and disoriented by new schemas and concepts as the people we design for, and this kind of change will take time. While I've tried to make the ideas behind this book as transferrable to current design processes as possible, there's no avoiding that with lofty goals come uncomfortable changes.

I hope you're on board with that, and if you are, from the bottom of my heart, *thank you*.

Index

A

- Action possibility, 316
- Activity Centered Design (ACD), 192
- Addictive technologies, 164
- Aesthetics, 74–76
- Agile-focused project-management app, 113
- Animation
 - advertisements, 57
 - color and typography choices, 58
 - Common Fate Principle of Gestalt, 58
 - communication, 58, 60
 - design processes, 57
 - emotional, 62
 - existence and sensibility, 62
 - interaction, 62–63
 - interfaces, 62
 - iOS home screen, 60
 - learning and memorization, 60
 - mobile apps, 59
 - simple to-do application, 61
 - tool, 58
 - Web, 57
- Attention and distraction
 - categories and cognitive economy
 - functional equivalence, 23
 - interfaces, 24
 - InVision’s web app, 25
 - mindful design, 27
 - organizations, 22

- person-based actions, 25
- recycling, 22
- sports and politics, 23
- cognitive miser theory, 19
- cost, 9
- decisions (*see* Decision-making)
- default mode network, 6–8
- design, 18–19, 27–28
- digital devices, 17
- economics, 16–17
- emotional distraction, 3–4
- environment, 2
- Internet, 18
- music, 1
- selective attention, 2
- sensory distraction, 2–3
- task-positive mode, 4–6

B

- Behavioral design, 162
- Black hat, DeBono’s framework, 330
- Blackmagicdesign DaVinci Resolve, 90
- Brainstorming technique, 330

C

- Carrot-and-stick approach, 162
- Choice, role
 - Amazon Dash Buttons, 334–335
 - creative problem-solving, 334

INDEX

Choice, role (*cont.*)
definitions, 339
design, 334
orchestral movements, 336
single-purpose applications, 337

Chrono Trigger's
character stats, 174
leveling up, 175
skills, 176

Clippy, 108

Cognitive economy
and categories, 21–24, 27
maximize, 96

Cognitive miser, 19–21

Colour psychology
coding, 52–57
design tools, 50
emotions, 50
red, 50, 51
von Restorff Effect, 51–52

Competitor research, 232–233

Contextual learning, 107

Contrast limitations, 63–65

Cost of distractions, 9

Customer research, 223–224

Customer service, 233

D

Daydreaming network, 7

Decision-making
attention, 10
continual attentional-hopping, 9
GitHub, 13–16
habit-changing, 13
inclusivity and equality, 10–12
social networks, 10

Default mode network
creativity, 7
danger of, 8
neuroscience and cognitive
psychology, 6
stimulus-independent thought, 7
task-focused mode, 6
task negative mode, 7

Design documentation, 261–262

Design framework, 349

Design testing, 254

Design thinking, 160

Digital environment, 198

Digital slot machines, 168

Documenting systems
artifacts, 284
empathy maps, 279–283
ethnographic documentation, 283
goal, 283
mental models (*see* Mental models)
state diagrams
codifying, 270
creation, animation, 268
database schema, 265, 267
developers, 267
interactions and display patterns, 271
observation research, 271
outcomes, 269–270
scope, 267
stage, 266
variable types, 267–268

Dopamine, 159

mesolimbic pathway, 165
motivational behaviors, 165
neurotransmitter, 165
organic chemical, 165
reward system, 166

Dopaminergic pathway, 165
 Dopamine squirt, 348
 Duolingo
 crown Levels and EXP, 179
 dashboard, 178
 designers, 180, 183
 friendly achievement, 181
 game-like mechanisms, 182
 learners, 183
 lessons, 178
 rewards, 180
 shallow rewards, 182
 typical implementations,
 reward flows, 180
 well-made learning tool, 183

E

Economics, 16–17
 Educational applications and learning, 82
 Emotional design, 95
 Emotional distraction, 3–4
 Emotion mapping
 design documentation, 261–262
 documenting systems (*see*
 Documenting systems)
 moment thinking, 263
 system thinking, 263–265
 Empathy maps, 279–282
 Environmental distractions, 3
 Environment, learning, 85, 97
 External teaching, 109–110
 Eye-tracking, 254

F

Feedback, positive, 94–95
 Feynman technique, 97

Figma, 88–89, 113–114
 Fixed interval scheduling, 166
 Fixed ratio scheduling, 166
 Function of state, 214

G

Galvanic skin response (GSR) testing, 254
 Gestalt principles
 closure, 47–48
 common fate, 46–47
 common region, 48–49
 concepts, 50
 continuity, 44–46
 design decisions, 78
 grouped/ungrouped elements, 49
 guidelines, 76
 proximity, 40–41
 relationships, 50
 similarity, 42–44
 visual design, 39
 GitHub, 13–16

H

Habit-changing, 13
 Human Centered Design (HCD), 192
 Humanistic psychology, 184

I, J, K

Iconic abstraction scale, 34–35
 Iconography, 36
 Implementation-focused phases, 287
 In-app guidance
 brush settings, 104
 Clippy, 108
 configuration, 105

INDEX

In-app guidance (*cont.*)
contextual learning, 107
designing, 109
Figma's onboarding, 107
interactive tours, 100
"Learn" screen, 101–102
limiting actions, 109
photoshop's interface, 102–103
photoshop team, 106
process, 108
Spot Healing brush tool, 103–104
surface-level interactions, 101
tool tip instructions, 106

Inclusivity and equality, 10–12

Infinitesimal differentiator, 285

Inherently rewarding, 169

Instagram's key moments, 290

Interaction design, 287
acknowledgment, 316–319
framing, 315
response, 322, 324–325
waiting, 319–321

Interaction-focused designer, 314

Interface sketches, 249–250

Internet, attention and distraction, 18

Intrinsic rewards, 173–177

InVision's web app, 25

Iteration, 256–257

L

Lab-based testing and
experimentation, 254

Learnable *vs.* educational applications, 82

Learning and memory
attentional system, 84
brain's neural activity, 84
cognitive economy, 79

conscious thought, 84

designing learnable
interfaces, 119

vs. educational applications, 82

elements and concepts, 118

environment, 85

framework, 119

goals, 79–81

information transmission, 79

long-term memory storage, 83

native language, 79

neuroscience of, 80

processes, 83, 118

recall, 84

skill, 79

storage and retrieval, files, 86

The Little Mermaid, 146

Limitations, contrast, 63–65

Linearity design

alert mockup, 203

checkout button, 202

elements and features, 204–205

Google Hangouts' iPad app
onboarding, 201

primer, 202

Loot boxes, 171–172

Lukewarm innovation, 347

M

Maslow's hierarchy of needs, 184

Mental models

cognitive economy, 96

communicate state, 95–96

documenting systems
abstracting, 272–274
flipping, 277–279
ubiquity of, 275–276

- exploration and
 - experimentation, 96–97
 - forgive mistakes, 92, 94
 - positive feedback, 94–95
- Mesolimbic pathway, 165
- Modern-day tech capitalism, 348
- Modern tech products, 329
- Moments
 - concept, 287
 - designing, 212–213
 - design work, 286
 - documenting flow, 293
 - change state, 294
 - design process, 295
 - framing scenarios, 294
 - sequential chain reaction, 294
 - environments, 286
 - first time, 288
 - friction points, 289
 - implementation mode, 286
 - interaction (*see* Interaction design)
 - interface parameter
 - filled-in feature, map, 292
 - post-it moment list, 290–291
 - system-level documentation, 290
 - limited-scope product, 287
 - payment process, 289
 - stories (*see* Story-based approach)
 - storyboarding (*see* Storyboarding technique)
 - treatment, 288
- Motivation
 - carrot-and-stick approach, 162
 - cognition, 166
 - definition, 163
 - emotional and intrinsic investments, 162
 - humanistic, 184
 - theories, 162
- Motivation-associated behavior, 166
- Music
 - expectations, 122
 - fourth dimensional design, 122–123
 - happy complexities, 144–148
 - harmony, 130–131
 - melody
 - chromatic scale, 126
 - Da Vinci Code, 129
 - hooks, 124
 - intervals, 126–127
 - intriguing phenomena, 125
 - measure of notes, 125
 - musical rules, 128
 - notes, 126–127
 - octave, 126
 - and phrasing, 124, 130
 - pitch, 124
 - relationships, 124, 126
 - scale, 128
 - sonic distance, 128
 - typographic scales, 129
 - Western music, 127
 - musical interface
 - application, 150
 - attention-grabbing
 - advertisements, 155
 - attention-grabbing phenomena, 155
 - bug-tracking app, 150–151
 - default cynicism, 149
 - degree of tension, 151–152
 - emotional journey, 150
 - empty states, 152
 - expectations and predictions, 150
 - happy state, 154
 - human expectation, 155
 - nudging behavior, 154

INDEX

Music (*cont.*)

- progressive media, 149
- Trello project, 153–154
- vulnerability, 148

rhythm, 131–133

tension (*see* Tension)

Music theory, 126

N

Neurotransmitters, 165

Nonlinearity

- abundance of potential actions, 205
- Cockos Reaper, 206
- creativity tools, 206
- degree of learning, 207
- documenting, linear flows, 207
- Trello's interface, 210
- video games, 207
- Vim in action, 208

O

Observation research

- description, 225
- designer-at-a-busy-agency, 230
- distractions, 226
- document preparation, 229
- field, 224
- in-field customer observation, 225
- interaction-designers, 230
- interview approach, 231
- motivators, 226
- problem space, 225
- recording methods, 227–228
- recruitment, 226
- screening, 226
- screen recordings, 228

side of caution, 228

sub-problems, 230

testing session, 228

P, Q

Perceived wait, 320

Persuasive design, 163

Plastering, 117

Positive feedback, 94–95

Products, habit-changing, 13

Purposeful pessimism

- attitude/behavior, 331
- concept, 330
- cynicism, 332
- Facebook, 332
- parameters, 333
- Post-it backdrop, 331

R

Rage clicking, 317

Research and planning

- accountability and reasoning, 223
- competitor research, 232–233
- customer research, 223–224
- customer service, 233
- deliverables, 238
- key questions, 238
- observation (*see* Observation research)
- user personas, 234–237

Reward-motivated behavior, 165, 167

Rewards

- carrot-and-stick approach, 162
- emotional and intrinsic
 - investments, 162
- ethics, 187–188
- feedback, 177

- mistakes
 - emotions, 162–163
 - manufacture reward, 161
 - overreliance on carrot and stick, 161–162
 - scheduling, 166–167
 - Reward-seeking behavior
 - games, 171
 - in-game slot machine, 171–173
 - intrinsic rewards, 173–177
 - operant conditioning chamber, 166
 - pocket slot machine (*see* Slot machine)
 - rewards as feedback, 177
 - reward scheduling, 166–167
 - Role-playing games (RPGs), 173
- S**
- Schrodinger’s Interaction, 314
 - Scribbling and sketching phases
 - deliverables, 251
 - design brief, 240
 - feature lists, 240–241, 243
 - key questions, 251
 - problem definition, 239
 - solution definitions, 240
 - success metrics and goals, 244
 - wireframes (*see* Wireframing)
 - Self-Determination Theory (SDT), 185–187
 - Sensory distraction, 2–3
 - Silicon valley, 347
 - Sketching, 239
 - Skill-tree approach, 182
 - Slack
 - app guidance, 344
 - emoji reactions, 341–342
 - explorative design, 340
 - granular notification control, 343
 - robot mascot, 345
 - team-centered communication, 341
 - virtual office, 339
 - Slot machine
 - drawn-out time investment, 170
 - money—at best, 169
 - onion-skin-like approach, 170
 - reward-motivating behavior, 168
 - social recognition and involvement, 169
 - tailored dopamine loop, 170
 - Social media, cost of distractions, 9
 - Social media platforms, 348
 - Spot Healing brush tool, 115
 - Storage and retrieval of files/data
 - design, 117
 - digital storage, encoding and decoding, 87
 - folder and file structure, 86
 - media
 - Adobe Lightroom, 89
 - Figma, 88–89
 - film photography, Adobe’s Lightroom, 89
 - interfaces, 90
 - job as designers, 91
 - modern products and interfaces, 91
 - mental models (*see* Mental models)
 - optimal environment, learning, 117
 - physical documents, 87
 - processing levels, 97–98
 - steps, learnability, 110–113, 115–116
 - storing, 87
 - teaching (*see* Teaching)
 - web or mobile application, 87
 - Story-based approach
 - completed storyboard, 307, 309
 - denouement, sort, 299
 - framing interaction, 296

INDEX

Story-based approach (*cont.*)

- friction points, 297
- intrinsic motivators, 298
- reflection moment, 299, 301

Storyboarding technique

- annotate, 309
- barrier, 303
- brainstorming session, 312
- current incarnation, 313
- empathy maps, 305
- ethnographic research, 307
- mapping, 310
- panels, 303
- product design, 302
- pruning, 312
- template, 304
- text-based story plan, 305–306

Surprise, 142

- gateway emotion, 143
- hippocampus, 142
- memory retention, 143
- remedy to habituation, 144
- shock-value moments, 143
- very surprising things, 143
- violation, 142

Synaptic cleft, 165

Systemic oppression, 329

System thinking, 263–265

T

Tactics, 160

Task-positive mode, 4–6

Teaching

- concepts, 98
- external, 109–110

in-app (*see* In-app guidance)

key areas, guidance, 98

onboarding phase, 98–100

Tension

control, 123

schema, 133

and anticipation, 136

Aristotelian Model, 134

challenges, 137

instrumentation, 137

mutability scale, 135

myopia, 138

and prototypes, 135

Prototype Theory, 134

rhythmic structure, 136

sociological events and cultural
convention, 135

transgenderism, 135

as schematic violation, 136–139

Testing, 252

Thumbnail sketches, 246–248

Tool tip instructions, 106

U

UI design tool, 319

User Centered Design (UCD), 192

User experience (UX) design

activities and tasks, 191

control and linearity, 196–197

description, 192

design disciplines, 193

design intangibles, 193–195

process, 191

User personas, 234–237

User testing, 252–256

V

Variable interval scheduling, [167](#)

Variable ratio scheduling, [167](#)

Vision, perception and aesthetics

animation and digital interfaces

(*see* Animation)

colour (*see* Colour psychology)

conventions, [66–67](#), [73](#)

digital e-readers, [71](#)

digital interfaces, [72](#)

Gestalt principles (*see* Gestalt principles)

iconic abstraction scale, [34–35](#)

impressionist painting, [33](#)

limitations of contrast, [63–65](#)

metaphor, [68–69](#)

painting, [31](#)

personality, [73](#)

photography, [36–38](#)

sensory process, [33](#)

signifiers, [66](#)

Super Team Deluxe web site, [75](#)

technology, [75](#)

W, X, Y

Wanting, [163–164](#)

Wireframing

conceptual models, [245](#)

interface sketches, [249–250](#)

storyboarding and state, [251](#)

thumbnail sketches, [246–248](#)

traditional definition, [245](#)

visual and interaction design

phases, [246](#)

Z

Zooming in and out

design process, [326](#)

scope, [326](#)

system-level thinking, [325](#)