Advanced Game Design
Essential References for Game Designers and Developers

These practical guides, written by distinguished professors and industry gurus, cover basic tenets of game design and development using a straightforward, common-sense approach. The books encourage readers to try things on their own and think for themselves, making it easier for anyone to learn how to design and develop digital games for both computers and mobile devices.

Visit informit.com/series/gamedesign for a complete list of available publications.
Advanced Game Design

A Systems Approach

Michael Sellers
Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact inttls@pearsoned.com.

Visit us on the Web: informit.com/aw

Library of Congress Control Number: 2017953924

Copyright © 2018 Pearson Education, Inc.

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearsoned.com/permissions/.

Dedicated to all those creating and becoming the next generation of game designers.
This page intentionally left blank
## Contents at a Glance

Acknowledgments ............................................................ xiii
About the Author .............................................................. xiv
Introduction ................................................................. 1

**Part I** Foundations ....................................................... 11
1 Foundations of Systems ................................................. 13
2 Defining Systems .......................................................... 49
3 Foundations of Games and Game Design ....................... 89
4 Interactivity and Fun ..................................................... 121

**Part II** Principles .......................................................... 169
5 Working as a Systemic Game Designer ......................... 171
6 Designing the Whole Experience ..................................... 187
7 Creating Game Loops .................................................... 219
8 Defining Game Parts ..................................................... 267

**Part III** Practice ............................................................ 293
9 Game Balance Methods ............................................... 295
10 Game Balance Practice ................................................ 319
11 Working as a Team ....................................................... 355
12 Making Your Game Real ................................................. 375

Bibliography ............................................................... 409
Index ................................................................. 417
This page intentionally left blank
# Contents

Acknowledgments ........................................... xiii
About the Author ........................................... xiv

Introduction .................................................... 1
A Combined Approach to Game Design ..................... 2
Where This Book Came From ................................ 2
What This Book Is and Isn’t About ........................ 3
Goals of This Book ........................................... 3
How to Read This Book ....................................... 6
Summary ....................................................... 10

## Part I Foundations ........................................ 11

### 1 Foundations of Systems ................................ 13
   Ways of Seeing and Thinking ............................ 14
   A Quick History of Systems Thinking .................. 28
   Systems as the Process of the World .................... 33
   Summary .................................................... 48

### 2 Defining Systems ....................................... 49
   What We Mean by Systems ................................ 50
   A Brief Definition ....................................... 50
   Defining Parts ............................................ 51
   Loops ....................................................... 60
   Wholes .................................................... 86
   Summary .................................................... 86

### 3 Foundations of Games and Game Design ............... 89
   What’s a Game? .......................................... 90
   Game Frameworks ........................................ 92
   Summing Up Game Definitions ........................... 95
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Systemic Model of Games</td>
<td>96</td>
</tr>
<tr>
<td>The Evolution of Game Design</td>
<td>117</td>
</tr>
<tr>
<td>Summary</td>
<td>119</td>
</tr>
<tr>
<td>4 Interactivity and Fun</td>
<td>121</td>
</tr>
<tr>
<td>The Player’s Part of the Game as a System</td>
<td>122</td>
</tr>
<tr>
<td>A Systemic Approach to Interactivity</td>
<td>122</td>
</tr>
<tr>
<td>Mental Models, Arousal, and Engagement</td>
<td>129</td>
</tr>
<tr>
<td>Interactive Loops</td>
<td>137</td>
</tr>
<tr>
<td>Recognizing, Defining, and Creating &quot;Fun&quot;</td>
<td>163</td>
</tr>
<tr>
<td>Summary</td>
<td>168</td>
</tr>
<tr>
<td>Part II Principles</td>
<td>169</td>
</tr>
<tr>
<td>5 Working as a Systemic Game Designer</td>
<td>171</td>
</tr>
<tr>
<td>How Do You Even Start?</td>
<td>172</td>
</tr>
<tr>
<td>Designing Systemic Games</td>
<td>175</td>
</tr>
<tr>
<td>Analyzing Games from a Systems View</td>
<td>182</td>
</tr>
<tr>
<td>Prototyping and Playtesting</td>
<td>184</td>
</tr>
<tr>
<td>Summary</td>
<td>185</td>
</tr>
<tr>
<td>6 Designing the Whole Experience</td>
<td>187</td>
</tr>
<tr>
<td>What’s the Big Idea?</td>
<td>188</td>
</tr>
<tr>
<td>The Concept Document</td>
<td>193</td>
</tr>
<tr>
<td>Designing the Game+Player System</td>
<td>214</td>
</tr>
<tr>
<td>Questions to Consider About Your Design Vision</td>
<td>216</td>
</tr>
<tr>
<td>Summary</td>
<td>217</td>
</tr>
<tr>
<td>7 Creating Game Loops</td>
<td>219</td>
</tr>
<tr>
<td>More Than the Sum of the Parts</td>
<td>220</td>
</tr>
<tr>
<td>A Brief Review of Loops</td>
<td>220</td>
</tr>
<tr>
<td>The Four Principal Loops</td>
<td>224</td>
</tr>
<tr>
<td>Three Kinds of Gameplay Loops</td>
<td>235</td>
</tr>
<tr>
<td>Defining a System’s Loops—And Goals</td>
<td>258</td>
</tr>
</tbody>
</table>
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team Roles</td>
<td>364</td>
</tr>
<tr>
<td>Summary</td>
<td>373</td>
</tr>
<tr>
<td>12 Making Your Game Real</td>
<td>375</td>
</tr>
<tr>
<td>Getting Started</td>
<td>376</td>
</tr>
<tr>
<td>Making the Pitch</td>
<td>376</td>
</tr>
<tr>
<td>Building the Game</td>
<td>385</td>
</tr>
<tr>
<td>Designing, Building, and Testing</td>
<td>385</td>
</tr>
<tr>
<td>Finding the Fun Fast</td>
<td>386</td>
</tr>
<tr>
<td>Effective Game Prototyping</td>
<td>386</td>
</tr>
<tr>
<td>Effective Playtesting</td>
<td>391</td>
</tr>
<tr>
<td>Phases of Production</td>
<td>399</td>
</tr>
<tr>
<td>Finishing Your Game</td>
<td>406</td>
</tr>
<tr>
<td>Summary</td>
<td>408</td>
</tr>
<tr>
<td>Bibliography</td>
<td>409</td>
</tr>
<tr>
<td>Index</td>
<td>417</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

Any book is a journey in the writing. I would like to thank all of my family members, friends, and colleagues who have over the years helped me sharpen my thoughts on game design and who urged me, sometimes forcefully, to take this journey. In particular, I’d like to thank Ted Castronova and Jeremy Gibson Bond for their continuous support; my students in the Game Design program at Indiana University for playtesting this book with me; and most of all my wife, Jo Anna, for her unwavering love, support, and inspiration across many years and adventures.

I’d also like to thank Kees Luyendijk for taking on the role of illustrator and early reader for this book, while also being a graduate student! I am grateful, too, to Laura Lewin, Chris Zahn, and the rest of the editing team at Pearson Education for their guidance and support in making this book a reality, and to Daniel Cook and Ellen Guon Beeman for being such generous, thoughtful, and incisive technical reviewers. This book could not have happened without the hard work of each of these individuals.
ABOUT THE AUTHOR

Michael Sellers is Director of the Game Design program and a Professor of Practice at Indiana University in Bloomington, Indiana.

Sellers has worked as a professional game designer since 1994, with a focus on designing social, mobile, and massively multiplayer online games (MMOs). He has started and run three successful game studios and has also worked for notable game developers such as 3DO, Electronic Arts, Kabam, and Rumble Entertainment as a lead designer, executive producer, general manager, and creative director.

His first commercial game was the award-winning Meridian 59, the first 3D MMO, released in 1996. He was also the lead designer on The Sims 2, Ultima Online, Holiday Village, Blastron, and Realm of the Mad God, among other games.

In addition to his work in games, Sellers has conducted and published original research in artificial intelligence. His AI research, partly funded by the U.S. Defense Advanced Research Projects Agency (DARPA), focuses on “social artificial intelligence”—creating agents that behave plausibly in social situations. As part of this effort, Sellers has published groundbreaking work on enabling artificially intelligent agents to learn, form social relationships, and have and express emotions based on a unifying psychological architecture.

Sellers has a BS in cognitive science. In addition to working on games and AI, he has worked as a software engineer, user interface designer, RPG miniatures sculptor, and briefly as a circus roustabout and movie extra.

He has a Bacon number of 2 and hopes someday to have an Erdos number.
In this chapter, we move from the foundational theory to the practice of designing games. Here we look at different aspects of the game design process and how to get started in each as a systemic game designer.

This is an overview that will be supplemented by Chapters 6, 7, and 8, where we go into more depth on designing the game as a unified whole, then its loops, and finally its parts.
How Do You Even Start?

Lots of people want to design games. They dream about it and talk about it but somehow never manage to actually get started. This is common, and most people who say they have a burning desire to design games never actually do it. Few manage to gather their courage and begin the journey of wading into the dark waters of game design. Rarer still are those who emerge on the far side, dragging their game kicking and screaming from the inchoate sea of design ideas. (That may seem like an overwrought metaphor, but when you complete your first game, you may no longer think so.)

One of the first questions people commonly ask when contemplating doing game design as more than a hobby, more than a “wouldn’t it be cool if” activity, is along the lines of “How do I even start?” Designing a game can seem like an impossible problem with no easy handles, no obvious way in. The sheer complexity and impenetrability of the problem can make it seem like the best you can do is leap in with both feet and hope for the best. That is, in fact, what generations of game designers up to now have done. At some point, those of us who have been designing games for decades just sort of made that first leap. For many the first few attempts are utter failures. Rovio went through 51 attempts before hitting it big with Angry Birds—and even this attempt looked like a flop at first (Cheshire 2011).

Failure itself isn’t a bad thing; anytime you try something new (which is most of the time in game design), you are going to fail a lot. However, you can reduce the amount and duration of failure by approaching game design systemically. Seeing a game as a system (containing other systems) is a good way to crack the problem of where to start in the otherwise overwhelming process.

From Wholes to Parts or Parts to Wholes

One key to knowing how to start is figuring out whether to begin with the parts, the loops, or the whole of your design. Opinions run high on this question. Many designers are firmly in one camp or another, and what they do works for them. Some designers will declare that any game design must start with “the nouns and verbs”—that is, the parts that will form the systems—while others begin with a more intuitive feeling of the kind of experience they want to create. Occasionally some will even start with Ellenor’s (2014) idea of “a machine that does x” and then work out what parts make it go and what sort of gameplay experience emerges from it. Differences of opinion on the “right” way to approach game design can make for miscommunication and talking past each other.1

---

1. I had this experience while working with Will Wright of SimCity fame. He is firmly a “nouns and verbs” kind of guy, while I often approach designs from a more holistic-experiential point of view. It took a while before we were able to understand each other’s perspectives.
Despite strong opinions from some designers, there is no single “right” way to approach game design. Our systemic view should make this clear: in designing a game, you need to get to the point where you have fully defined the parts, the loops, and the whole of your design. As a game designer, you need to be able to move up and down the organizational levels with ease, shifting your focus between the parts, the loops, and the whole as needed. As a result, you can start the design process with whichever of these makes the most sense and bounce between them as needed.

**Know Your Strengths, Work to Your Weaknesses**

When you begin thinking about making a game, where do your thoughts lead you? Do you think about things like having a game where players are sharks or superheroes, or where each is a kite in the sky? Or are you more likely to approach a game as a simulation or modeling problem? If it’s a game about a little one-celled organism, do you start by listing all the parts of the cell? Or do you maybe start thinking about a game where the player is the manager of a remote trading post by jotting down how buying and selling would work?

Every game designer has their strengths; everyone has their “home place” where they start—and then retreat to when making the design becomes difficult. You need to find out where your game design home is and then work out ways to not give in to the temptation to stay there; you also need to figure out how to work with others who approach game design differently from you.

The *doing* of game design is the best way to figure out which parts of the process come most naturally to you. Still, it is worth considering where you think it should start and working from there.

**Storytellers**

Game designers who tend to start with the whole experience often paint an evocative picture of the player’s journey through a game: how the player feels, what they encounter, and what sort of changes they go through. Game designers like these can sometimes seem like expert storytellers. They’re able to give you the grand sweep of the world…but they can run into trouble. Games aren’t stories. “Telling” a game like a story can be a satisfying first pass at building the world that the players inhabit, but ultimately the game has to be much more than that.

A storyteller needs to hang on to their talent for painting a mental picture of the experience of a world but not get stuck there. If you are a storyteller, you need to build your talents for creating working systems that have their own tokens, rules, and dynamic elements. You likely have the thematic part in hand, but you need to support it with the structure of the underlying game—and work with others who can help you do so.
Inventors
Many game designers are enamored of inventing complex mechanisms—things like clocks with lots of gears, marble-run sculptures, and so on. These can be mesmerizing displays of systems in action. Similarly, sometimes game designers come up with ideas for new kinds of ecological or economic mechanisms and spend time playing with them. For example, the early prototypes for the game *Spore* included lots of different simulation mechanisms, including one that (with a bit of help from the player) simulated the formation of a star system from an interstellar cloud of gas and dust.

But as fascinating as these inventions can be, they aren’t games. As with telling a story about a game, designers will sometimes build a mechanism that scratches the “watch it go” itch, only to realize that they left out the need for a human player. The designer may toss the player a few scraps of things to do, but it’s clear that the mechanism or simulation remains in the spotlight. If you are an inventor, you can do a lot to build fascinating dynamic systems—but don’t forget that games must have human involvement as an integral part of the system and that players need to have long-term goals and reasons to play the game (the whole of the game), or it will be uninteresting to them.

Toymakers
Finally, some game designers are first and foremost toymakers. They love to make little pieces or mechanisms that don’t really do anything but are still attractive and engaging, at least for a minute or so. Or they might be among those with highly specific domain knowledge—things like the climbing rate and ammunition capacity for a Sopwith Camel or the relative merits of different sorts of swords in medieval (or at least fantasy) combat, or the types of coral on a typical reef—or may just love digging in to find this kind of information.

Many game designers who start with the “nouns and verbs” of their design fit into the toymaker category. Maybe you want to make a game about cells in the immune system attacking invading viruses, and so you start with what you know (or anything you can find) about how a T-cell works. What the player does and why this is engaging or fun are questions that you may not think about right away or that you may have difficulty finding answers for. Having the ability to ground your design in specific parts and behaviors—tokens and rules, nouns and verbs—helps you create prototypes quickly. However, to make it into a game, you need to find ways to build interactive systems and find some goals for the player to pursue and experience.

Working Together to Find the Fun
The good news about these different views of game design is that once you find your starting point as a designer, you can extend your abilities into the other areas. Any one of these is great as a starting point, as long as you don’t end there, too. The better news is that you can also find others who have different game design talents and work with them. It can be difficult and even frustrating for game designers with different design styles to work together, but the result is almost always far better and more engaging for the player as a result.
No matter which part of the game design process you prefer, you will need to extend yourself into the other areas and learn to listen to and work with those who see the game design process differently from you. A lot of game design comes down to being able to communicate your ideas, hear other people’s ideas, and generally work together with those who have strengths that are different from yours. Understanding game design as systemic design helps illuminate these different views on games as systems and on game designers as system designers. That understanding should help you refine your skills and look for others who complement them.

A large part of doing game design is in the oft-repeated phrase “find the fun.” You may start with a cool toy, an intriguing mechanisms, or a compelling experience—the parts, loops, and whole of a game—but you will need all three elements plus engaging interactivity to build a fun game. To do that, you need to apply your knowledge of systems to creating game systems and games as systems.

Designing Systemic Games

As a way to approach designing games as systems, we can look at the properties of effective systems in games and how they affect the process of game design.

Qualities of Game Systems

Achterman (2011) has provided helpful guidelines for building game systems. In his view, five qualities are the hallmarks of effective game systems:

- **Comprehensible:** As a designer, you have to understand your game as a system and the systems within it. Of course, your players have to be able to comprehend it, too. This is why both design documentation (for you) and presenting the game in such a way that players can build a mental model of it are so important.

- **Consistent:** Achterman points out the importance of having “rules and content [that] function the same in all areas of your game.” It can be tempting to add an exception or a special case to fix a problem, but doing so tends to decrease the resilience of the system (which sets up the game for later problems) and makes it more difficult to learn. (This is similar to the discussion in Chapter 3, “Foundations of Games and Game Design,” on elegance.)

- **Predictable:** Game systems should have predictable outputs for given inputs. While making games predictable helps players build mental models of the games, it can also be somewhat at odds with designing systems for emergence. Being predictable should not be taken as meaning that game systems should be obviously or boringly mechanistic. However, neither should your systems produce wildly different results for similar inputs, much less become brittle and break down due to unforeseen circumstances. You should at
least be able to know that you have accounted for any edge cases that might hurt a player’s experience or provide them with a gap in the system to exploit to their advantage.

■ **Extensible:** Building games systemically typically makes them highly extensible. Rather than depend on custom-created content “set pieces” (e.g., expensive hand-created levels), as much as possible you should create game systems such that content can be reused in new ways or created procedurally. You want to create parts and loops that can be used in multiple ways, not a single-use arc that makes for a complicated rather than complex set of relationships. While in a loop the parts affect each other cyclically, as veteran game designer Daniel Cook said, “An arc is a broken loop that you exit immediately” (Cook 2012). Designing in terms of loops rather than arcs also makes it easier to take a system and add it to a new game or put it in a new context, where it acts as a part in a new larger system. For example, you may decide that you want to add a whole new class of buildings for players to construct; if you have a general “building construction” system in the game, this is much easier to do than if you have to hand-craft another one. By designing game systems carefully, with only the needed parts and sufficient loops between them, you will be able to extend the systems internally or extend their use externally far more easily than if you rely on more static content or fractured, separated systems in the game.

■ **Elegant:** As discussed in earlier chapters, elegance is often a hallmark of systems. This quality sums up the ones above. It goes beyond but is related to the quality of consistency discussed above. The following are some examples of elegance:
  - Creating a diverse space for players to explore based on only a few rules (Again, Go is the archetypal example of this.)
  - Having systemic rules with few exceptions that are easy to learn, where both predictable and emergent behaviors are possible
  - Enabling the system to be used within multiple contexts or to have new parts added within it

**Tabletop and Digital Games**

This book uses examples from both tabletop games—also called analog games, board games, physical games, and so on—and digital games—those played on a computer, console, tablet, or phone. From a game design point of view, there is a great deal of commonality between these types of games, no matter their genre or other differentiating attributes.

There is a great deal to be learned from studying tabletop games, even if you never plan to design one. Designing for situations in which the only “computing power” is in the players’ heads and where all interaction must happen using tokens the players can physically manipulate presents a significant challenge. It constrains what you as a designer can do to bring a game concept to life and highlights the relationships between the game’s tokens
and rules, loops, and overall experience. Digital games can hide a lot of game-designer laziness behind flashy graphics and narrative cut-scenes; tabletop games do not have that luxury.

In speaking to university theatre students, actor Terrence Mann said, “Movies make you famous, television will make you rich; but theatre will make you good” (Gilbert 2017). There is an analogy here to game design (not that any particular type of game design will necessarily make you rich or famous): designing tabletop games has the same sort of relationship to designing digital games that acting in theatre does to acting in movies. Like theatre, tabletop games are closer to the audience; you as a game designer can hide less, and must hone your craft in designing for this environment.

This is not to say that all game designers must design board or tabletop games, though it is good practice. But if at times you wonder why so many board games are used as examples when “modern” games are typically played on computer, this is the reason. Tabletop games have undergone every bit as much of a renaissance in the early 21st century as have digital games. As a systemic game designer, you can learn from both, and you may well find that designing tabletop games challenges your skills in ways that designing for games run on the computer does not.

The Process of Designing Games as Systems

Stepping down a bit from the abstract qualities we hope to find in game systems, we can look at the overall design process common to systemic game design (whether tabletop or digital).

This is necessarily an iterative process between designing the parts, the loops, and the whole. At first, this process may be iterative in your head, on a whiteboard, and on scraps of paper and then in documents and spreadsheets. Once the game begins to take shape, the iterative cycle of prototyping and playtesting discussed briefly below (and in more detail in Chapter 12, “Making Your Game Real”) becomes important: it is far better to prototype fast and playtest early than to hope the idea you have in your head will spring forth fully formed like Athena from Zeus’s skull. (They never do.) This process is the game designer’s loop shown in Figure 5.1 (which is the same as Figure 4.3).

As stated earlier, it is possible to begin at any point in the systemic structure: with parts, loops, or the whole experience—as long as, having started with one, you move to the others so that they mutually support each other. With that reminder, for convenience here we will start with the whole, the architectural and thematic elements, and then move to the functional looping aspects, and finally move to the parts.
The Whole Experience: Thematic Architecture

As discussed in Chapter 3, the high-level design of a game has to do with the player’s overall experience. We can separate this into architectural and thematic elements—the technical aspects of the user experience (how the game looks and feels) and the more ethereal, sometimes tacit qualities that define what the game is about. Understanding the whole of the game answers the question What is the point of the game (or a system within the game)?

As one example, in a recent conversation, Jason VandenBerghe, creative director on the game *For Honor*, said, “I believe that combat is an art form. The game sprung from that belief” (personal communication, December 2016). His desire was for the player to experience hand-to-hand combat as a lethal, dance-like form of art. While that desire is not enough on its own to support the game design, it is a compelling vision, a star to guide the game’s developers and from which all the interactions and details of the game eventually arise.

Many times, game designers or entire development teams will launch themselves into the game development process without stopping to entirely clarify what the “whole experience” is that they want in their game. Questions of theme and vision seem frivolous; the team wants to get to making the game! However, as you will see in Chapter 11, “Working as a Team,” having a shared, coherent vision of the game your team is making is the single most important indicator of success.

There are multiple aspects of any overarching vision, as discussed in the following sections. These aspects represent and point to more detailed elements that have to be articulated to get an idea of what the game will be.
The Game’s World and History
To begin with, what is the world, and what is the player’s point of view within it? You may be thinking of a gritty, cold-hearted world of spies and double-dealing—but is the player a spy working their way up in this world? A spy-master overseeing and pulling the strings on a sometimes wayward team of spies? Or possibly an old spy coming out of retirement for one last vengeful mission? Each of these paints a different picture and will take your game design in a different direction.

To fill in the world somewhat, what are the major events in its history—those that are applicable to the players? If you’re a storyteller, you may have to resist the urge to write 100 pages of world lore. If you have the time and money, and especially the experience to know what’s useful and what’s not, then you can indulge yourself in this; you will likely add important details to the game world that make it come to life all the more vividly. But if you have any time or budget constraints, or if you’re just starting out, you should avoid the siren song of diving too deeply into the backstory. You need to know what the world is and what it’s about, but to start with, you can do this in a page or two of text. You shouldn’t write any more than you need to support the rest of the design. Later, as the game is beginning to come together, you can flesh out the deep, tragic history of the city where the streets hold a million secrets.

Narrative, Progression, and Key Moments
The game world’s history is its past. Its present and future are contained in the game narrative. Does your game have a predefined story the player has to work within? Are there larger events happening around the player that grow out of the large-scale history but that leave room for the player to make their own decisions? Or is the game’s history a jumping-off point for the player, where what’s past is prologue, and there is little in the way of continuing narrative to guide the player’s actions?

Understanding your game’s world and (some) of its history will also help you begin to define major events that happen in the game, the player’s goals and progression through it, and “key moments”—short moments or stories that you can tell that help communicate meaningful, climactic points for the player.

Art, Monetization, and Other Whole-Experience Concerns
There are a variety of questions to work through at the level of the whole-game experience: Will the game’s art style be 2D or 3D? Painterly, cel-shaded, or super-realistic? How does your choice reflect the game’s heart and theme to the player? Closely aligned with this is the way the player interacts with the game—the user interface and user experience, often referred to as UI/UX. Even monetization design—how your game makes money—is something you have to consider at this stage.

In Chapter 6 we will look in more detail at the process of designing and documenting the gameplay experience as a whole. For now, keep in mind that it doesn’t matter so much whether you start with a high-level, blue-sky creative vision that you then support with underlying
loops and parts or whether you arrive here after first nailing down those dynamic and specific aspects; either way, you will iterate back and forth between them as you refine your ideas. What matters is that before you begin developing your game—before you assure yourself that you know what the game is—you have this theme and vision, the whole of the player’s experience, clearly articulated and shared by your team.

**Systemic Loops and Creating a Space for Play**

Chapters 3 and 4, “Interactivity and Fun,” discuss the game’s loops: the game’s dynamic model of its world, the player’s mental model of the game, and the interactions that happen between the player and the game. Designing and building these loops and the structures that support them is the heart of being what is often loosely referred to as a “systems designer.” In addition to the overview here, this topic is explored in detail in Chapter 7.

In creating a space for the player to explore and inhabit—rather than a singular path for them to follow through the game—you need to define the game’s systems. These systems need to support the theme and desired player experience, and they must work interactively between the game and the player. You need to specify and create (via iterative prototyping and playtesting) the player’s core loops, explicit goals, and the way they progress through the game.

Creating systems like this may be the most difficult part of game design: it requires that you envision the system as it uses the game’s tokens and rules to create an experience that is hard to see clearly in advance. Of course, you don’t have to do this all at once—which is why prototyping and playtesting are so important—but being able to imagine multiple looping systems well enough to record their designs and implement them is nevertheless a daunting task. For example, in many games, the systems controlling resource production, crafting, wealth production, and combat all have their own internal workings, and all interact with each other and the player to create the player’s experience. Getting all these to work on their own and contribute to a systemic whole requires skill, patience, and resilience in the face of repeated attempts when something just doesn’t quite work.

**Balancing Game Systems**

Part of making game systems is ensuring that all parts defined by the game are used and balanced against each other and that every system in the game has a clear purpose. If you add a quest system to your game and players ignore it, you need to understand why it isn’t contributing to their experience and determine whether to remove it or fix it so that it does. Chapter 9, “Game Balance Methods,” and Chapter 10, “Game Balance Practice,” go into this process in detail.

**The Structural Parts: Tokens, Values, and Rules**

It may be that you started the game design process with an idea for a fun looping mechanism or interaction. Or maybe you started with the kind of experience and feeling you want the players to have, and so you’re defining the game and interactive loops. Or in some cases you may start with an idea for the building blocks out of which you want to construct your game. In any
case, before the game is really a game, you need to situate the game’s functional loops into the context of the whole—the game experience—and also create the structural parts of the game’s systems.

You first read about the tokens, values, and rules in a game in Chapter 3. You will see them again in detail in Chapter 8. For now, in terms of working as a systemic designer, you should understand that the process of nailing down exactly what is going on in a game—getting past the hand-waving descriptive stage and being able to implement the game—is vital. You don’t have a game without it.

This aspect of game design is sometimes called “detailed design,” and it is where the game design becomes entirely specific. Does that sword have a weight of 3 or 4? A cost of 10 or 12? How many types of troops, or horses, or flower petals are there in the game, and what differences do these numbers make to the overall gameplay? Tracking and specifying these structural parts of the game has been called the “spreadsheet-specific” part of game design. This is a crucial part of systemic design; it is in many ways how the game becomes real. Such specific design is needed for balancing the different tokenized parts against each other to make the game a cohesive whole rather than allow it to become separate systems that can fly apart.

The issues you need to think about here are how to specify tokens that represent the objects in the game—the player, other people, nations, creatures, spaceships, or whatever the operative units are within your game—and give each of them sufficient attributes, values, and behaviors to define them. One way to think of this is to answer the question What is the smallest number of attributes, states, and behaviors you can use to support the game’s systems and provide the overall gameplay experience you want?

Related to this are the issues of how to make obvious to the player what the tokens in the game are, what they do, and how the player can affect them. This in turn feeds into the game’s UI/UX—how the board or screen is laid out to present the necessary information about the game to the player. This cannot be specified until you know what the necessary information is. At the same time, approaching this issue by asking what sorts of information you think the player needs to know to play the game can itself help clarify the tokenizing process.

Chapter 8 talks about this process in more detail, including how to create complex objects, game pieces, or tokens by having a small number of general attributes that interact with each other to create their own subsystems within the larger game systems. Chapter 8 also discusses the importance of inter-object behaviors and how to avoid “easy win” or other gameplay-killing tokens in your games.

**Revisiting the Systemic Design Process**

As a systemic game designer, your loop—the designer’s loop—involves cycling between seeing the game as a whole, as systems, or as individuated parts (see Figure 5.2). You need to be able to see them all at the same time and how they affect each other. You also need to
be able to dive into any one in detail, depending on what’s needed by the game design. It’s important that you not focus on any one level to the exclusion of the others; you also don’t want to continue to work ineffectively on any one of them. When you find yourself pushing on one level without any real effect, it can often help to switch and work from the point of view of the other levels to help reveal what you need in another. If you can’t quite get the experience down, explore the tokens and how they work; see how they inform the experience. Or if you have the experience clearly in mind but can’t quite specify the tokens, see what the systems tell you about how those have to work. At the same time, don’t let yourself avoid tokenizing your systems, making sure there are interesting interactions, or ensuring a cohesive theme because one or more of these aren’t in your comfort zone as a game designer. All of these are necessary for any working game, and all are necessary activities for a systemic game designer.

Figure 5.2 As a game designer, you need to be able to see the parts, the loops, and the game’s whole experience all at the same time and zoom in on any one of them as needed

Analyzing Games from a Systems View

Working as a game designer doesn’t just mean designing your own games; it also means playing and analyzing a lot of other people’s games. It’s important to be able to understand what makes other games work—or not work in particular areas.

You can follow the same systemic structure for analysis as for design. It involves looking at the whole experience, including how you build your mental model of the game; the game’s internal and interactive loops; and the rules and tokens that make those up. By carefully identifying and separating these, you can gain insight into the decisions made by the game’s designer and improve your own designs as a result.

When beginning to play a game for the first time, examine how you go about building your own mental model of it: Do you understand the setting and theme? What surprises you about it? What concepts about the game did you find to be important, incomplete, or hard to understand as you learned the game? How might the game have increased your engagement early on?
While playing and after playing, think about the whole of the experience you had. What kind of experience and feelings do you think the game designer was trying to elicit in you as a player? Were there particular aspects of the game that supported or detracted from your experience?

What visual and interactive elements of the game support its theme and the desired player experience? What can you infer about the game designer’s intent for the game, based on the art style and interactive aspects?

What specific game systems can you identify in the game? Are there systems that operate independently of the players, or do they all rely on the players doing something first? The board game Power Grid is a great example of a (nondigital) game that has systems that operate mostly outside player control. For example, in this game there is a simple but highly effective depiction of supply-and-demand economics: as players buy more of any one kind of fuel, the price for it goes up until its supply is replenished on the next turn (see Figure 5.3).

**Figure 5.3** The board game Power Grid, showing the track representing prices for the resources coal, oil, trash, and nuclear fuel. As players purchase each and supply decreases, its price rises. Supply is replenished each turn, driving prices lower if the fuel is not used.

Continuing with the analysis overview, as a player in a game, how do you progress, and what reinforcing loops can you identify? What balancing loops are there that push back against player advancement or that keep one player who outstrips others early on from simply winning the game?

What are the primary forms of interactivity in the game? How does the game allocate its interactivity budget? Is this a game of strategy and socializing, or one of quick thinking and fast action? Do the ways you as a player interact with the game help establish the game’s theme, or do they work against it?
Finally, what are the particular tokens and rules—the atomic parts of the game with their values and behaviors? Do they support the desired gameplay experience or get in its way? Having learned one system in the game, can you transfer how that works to another part of the game, or are there lots of rules to learn, each with its own exceptions—so that you have to spend a lot of time thinking about how to play the game?

Often the art style of a game is expressed in its individual tokens, sometimes in surprising ways. For example, the tabletop game *Splendor* is about building up your business as a gem merchant, starting with individual mines and ending with courting the favor of various nobles. The physical pieces in the game are like poker chips. They represent individual gems, and each has an unusual amount of heft. Their weight subtly adds to the desired experience of the game, even though, like the rest of the art (and most art in games), it is nonfunctional.

As you analyze games by examining their parts, loops, and wholes, you will begin to see commonalities across them, as well as how each is unique. Understanding the similarities and differences will help you improve your own designs—avoiding the mistakes of others, springboarding off their good ideas, and keeping your game design fresh and engaging.

**Prototyping and Playtesting**

A final important part of working as a systemic game designer is iteratively getting feedback. Game design is necessarily a process of repeatedly testing and refining game design ideas in the service of an overall vision for the game. Game ideas will not make it from your mind to their final form in front of the player without having gone through many changes first. It’s common for almost everything about a game except for its single unifying vision to change multiple times during development.

As an example from a related creative field, making movies, Ed Catmull, president of Pixar, has been open about the many gyrations that films at his studios go through. “All of our movies suck at first,” he said when speaking to aspiring movie animators. He clarified that statement by adding, “A lot of people don’t believe me when I say that. They think I’m being self-effacing or modest, but I don’t mean it in that sense. I mean it in the way that the film sucks.” He went on to discuss the many story changes that the movie *Up* went through during its development: it started with a story about a kingdom in the sky with two princes who didn’t like each other, who fall to earth and end up meeting a giant bird named Kevin. That version went through a huge number of changes. By the time they completed the movie, he said, “All that was left was the bird and the word ‘up’” (Lane 2015).

The same sort of thing happens in games. While your game may not change as drastically as a movie like *Up*, you must be prepared for many iterations—many cycles through the creative process. This means you have to be willing to test your ideas over and over again, learning and changing them as you go. And it means you have to be humble enough to change an idea or throw it out if it isn’t working. Iterating and “finding the fun” inevitably means throwing away...
a lot of work—drawings, animations, programming, design documents, and so on. You cannot cling to something you have worked on just because you put a lot of time into it. If you do, you will be settling for an idea that is okay (or mediocre) when with a little more work and polish it could have been great.

To iterate effectively on game designs, you need to make them real. The only way to do this is to make early versions—prototypes—and test them. You may start with drawings on a whiteboard or pieces of paper and coins being pushed around on a table—anything to start actually playing with the idea you have. Most of your prototypes will be varying degrees of ugly or unfinished, converging on the full, finished, and polished product at the end. The point is to take your game design out of the realm of ideas and into real implementations that can be played and tested—and to do so as quickly and often as possible.

Playtesting is how you validate your prototypes—or, more often, how you find out where your game design is broken. Developing a game designer’s intuition for what will work or not is important, but even for the most experienced designers, it is never a substitute for testing the gameplay on players who have never seen it before. As Daniel Cook has said, without implementation and playtesting a game design remains an “ineffectual paper fantasy” (Cook, 2011b). You will need to test your design ideas with other people early and often to keep your game on track.

We will return to the topics of prototyping and playtesting often in the following chapters, particularly in Chapter 12. For now, understand that a core aspect of working as a game designer is having the humility and creative flexibility to test and refine your game design ideas based on what others think of them. You will need to make fast, often ugly prototypes, and you will need to test them with potential players repeatedly during design and development. The bright shining idea you have in your head will never survive contact with reality without change—most likely a lot of change.

**Summary**

This brief chapter provides an overview of what it means to work as a systemic game designer. While getting started on a new game design can be truly daunting, by breaking down the game into its parts, loops, and wholes—not necessarily in that order—you can begin to get a handle on defining the game at each of those levels.

The coming chapters add more detail to the topics discussed here. Chapter 6 examines the whole of the game experience in more detail—how you discover it, document it, and set up for creating the underlying systems. Chapter 7 revisits the game’s functional loops, this time using the knowledge of systems thinking and game loops to specify the particular loops for your game. Then Chapter 8 looks again at the game’s parts and how to create these “spreadsheet-specific” tokens, values, and rules.
INDEX

Numbers
3-door problem, 307–309
5G networking, 34

A
absorption, 134
acquisition funnel, 350–351
acquisition of players, 350–351
action games, 197
action/feedback interactivity, 138–139
moment-to-moment gameplay, 141
present-tense action, 139
reflexive attention, 139
stress and reward of fast action, 139–141
action-social motivations, 201
active opponents, 109
adaptability, 51, 82
ad-supported games, 211
Adventure Capitalist, 247, 325
adventure games, 197
aesthetics, MDA (Mechanics-Dynamics-Aesthetics) framework, 92–93
affordance, 131
agency (player), 100
Agile, 400–401
agon games, 91
agreeableness, 202
Albion Online, 344–345
alea games, 91
Alexander, Christopher, 32, 46–47, 60, 81
alpha milestone, 405–406
analog prototypes, 387
analytical balance, 299, 348
analytical balance, 299, 348
analytic driven design, 301
analytic-informed design, 301
cautions about, 301
player behavior data
acquisition and first experience, 350–351
community, 354
conversion, 352–353
user, 353–354
player cohorts, 349–350
player information, collecting, 349
sample size and information distortion, 301–302
Tumbleseed example, 299–301
analytics
analysis from systems view, 182–184
analytical balance, 299, 348
analytics-driven design, 301
analytics-informed design, 301
cautions about, 301
player behavior data, 350–354
player cohorts, 349–350
player information, collecting, 349
sample size and information distortion, 301–302
Tumbleseed example, 299–301
hypothesis-driven analysis, 17
production analytics, 404–405
analytics-driven design, 301
analytics-informed design, 301
Angry Birds, 172, 188
Antichamber, 100
Apache OpenOffice, 261
APs (associate producers), 366–367
arbitrage, 342–344
architectural game elements, 111–112
autotelic experience, 114–115
content and systems, 112
balancing, 114
content-driven games, 112–113
systemic games, 113–114
meaning, 116–117
narrative, 115–116
themes, 116–117
architecture of companies, 364–365
Aristotle, 22, 46, 60, 96
arousal, 129–130, 132–133, 148
ARPDAU (average revenue per daily active user), 352
ARPU (average revenue per user), 352
art, designing, 179–180
The Art of Computer Game Design (Crawford), 118
articulatory distance, 125
artists, 371–372
associate game designers, 368
associate producers (APs), 366–367
atomic parts, 268. See also game parts
atoms, structure of, 39–42
attention
balancing progression with, 340–341
evacutive, 141
reflective attention, 147
reflexive, 139
attributes
defining, 270–273
first-order attributes, 271–272
nautical game example, 275
ranges, 272–273
second-order attributes, 271–272
third-order attributes, 271–272
values, 335–336
weights, 334–335
audience, target, 200–204. See also players
demographics, 202–203
environmental context, 203
identifying, 378, 389–390
motivations, 200–202
psychographics, 200–202
audio style, 209–210
auditory feedback, 281–282
Austin, Thomas, 25–26
Australia, introduction of rabbits into, 25–26
autopoiesis, 32–33
autotelic experience, 114–115
avatars, 97
average revenue per daily active user (ARPDAU), 352
average revenue per user (ARPU), 352

B
balance. See game balance
Balileo, 28
basic emotions, 147
basic progression ratio, 324
behaviors, 276–277
behavior extinction, 130
definition of, 94
emergence, 278–280
feedback, 280–281
amount of, 282
comprehension and, 282–283
kinds of, 281–282
nautical game example, 283–285
player expectations and, 281
timing of, 282
generic, modular behaviors, 277–278
interactivity, 124–125
game behaviors and feedback, 125–126
intentional choices, 127
player behaviors and cognitive load, 125–126, 159
local action, 277
looping systems, 285–286
player behavior data
acquisition and first experience, 350–351
community, 354
conversion, 352–353
player information, collecting, 349
retention, 351–352
usage, 353–354
bell curve, 306
belonging (Maslow’s hierarchy of needs), 149
benefits
cost-benefit curves
exponential curves, 324–327
linear curves, 323
logistic curves, 327–328
NAP (near-arithmetic progression) curves, 331
piecewise linear curves, 329–331
polynomial curves, 324
cost/benefit definitions, 320–321
core resources, 321–322
special cases, 322–323
subsidiary resources, 322
Bernoulli process, 309
Bertalanffy, Karl von, 30
beta/first release milestone, 406
biases, cognitive, 307–309
Bioshock, 153
Blake, William, 30
“blind boxes,” 307
blue-sky design
  cautions, 192
  constraints, 191
  curation, 190
  definition of, 188
  methods, 188–190

Boosting engines
  definition of, 236–237
  engine problems, 238–239
  examples of, 237–238

Boomshine, 231

Brainstorming, 188–189

Braking engines
  definition of, 239–240

Breadth (concept document), 215

Bridges, collapses of, 68–69

Bushnell, Nolan, 84, 130

Bushnell’s Law, 84, 130

Caillois, Roger, 90–91

call to action, 131, 384

Candy Crush, 143, 161–162

Capra, Fritjof, 32

CAS (complex adaptive systems), 31–32

Case, Nicky, 260, 279

casual games, 197–198

Catmull, Ed, 184

causality, upward/downward, 80–81
causation versus correlation, 18–19

CCOs (chief creative officers), 365

CDs (creative directors), 365

CEOs (chief executive officers), 364

CFOs (chief financial officers), 364

Chabin, Michael, 50

Chambers, John, 34

changing probabilities, 307

channels, market, 346–347

chaotic effects, 68–70

Chess, 98, 99

Chief Creative Officers (CCOs), 365

Chief Executive Officers (CEOs), 364

Chief Financial Officers (CFOs), 364

Chief Operating Officers (COOs), 364

Chief Technical Officers (CTOs), 365

“Choose your own adventure* books, 105

Cisco Systems, 34

Civilization, 114

Clash of Clans, 227–229

“cobra effect,” 19

cognitive biases, 307–309

cognitive interactivity, 141–143

cognitive load, player behaviors and, 125–126, 159

cognitive threshold diagram, 159–161

cohorts, 349–350

collapsing bridges, 68–69

collecting player information, 349

collections, simple, 58–59

combat systems, 256

combined effects, 63–65

combining loops, 251–253

commercial release, 406

commons, tragedy of, 76–77

communication, team, 361–362

community analytics, 354

company architecture, 364–365

competition, 145

complementary roles, 146

complex adaptive systems (CAS), 31–32

complex resources, 222

complex systems, 59–60

complicated processes, 58–59

comprehensible game systems, 175

Computer Game Developer’s Conference, 118

concept
  blue-sky design
    cautions, 192
    constraints, 191
    curation, 190
    definition of, 188
    methods, 188–190

concept banks, 400

concept documents, 193–194

concept statement, 195–196, 381–382

depth and breadth, 215
detailed design, 212–214
elegance, 215
game+player system, 214
genre(s), 196–200
product description, 206–212
questions to consider, 216
target audience, 200–204
themes, 214
USPs (unique selling points), 204–205
working title, 195
definition of, 188
desired experience, 192–193
concept banks, 400
concept documents, 193–194
concept statement, 195–196, 381–382
genre(s), 196–200
product description, 206
game world fiction, 210
monetization, 210–211
player experience, 206–209
scope, 212
technology, tools, and platforms, 211–212
visual and audio style, 209–210
target audience, 200–204
demographics, 202–203
environmental context, 203
motivations, 200–202
psychographics, 200–202
USPs (unique selling points), 204–205
working title, 195
concept phase (game production), 401
concept statement, 195–196, 381–382
conflict, 108–109
connectors, 54
conscientiousness, 202
consistency, 175
constraints, blue-sky design and, 191
construction systems, 257
content, 112
balancing, 114
content-driven games, 112–113
systemic games, 113–114
content-driven games, 112–113
contexts for pitches
elevator pitches, 379–380
pitch meetings, 380–381
contribution (Maslow’s hierarchy of needs), 149
conversion, player, 352–353
converters, 57–58, 223
Cook, Daniel, 114, 207, 400
COOs (chief operating officers), 364
Copernican model of solar system, 28–29
core loops, 127, 156–157, 226–227
detailed design (concept document), 212–213
examples of, 227–231
game mechanics, 220–223
core resources, 321–322
correlation, 18–19
cost-benefit curves, 320–323
exponential curves, 324–327
linear curves, 323
logistic curves, 327–328
NAP (near-arithmetic progression) curves, 331
piecewise linear curves, 329–331
polynomial curves, 324
Costikyan, Greg, 92
costs
cost-benefit curves
exponential curves, 324–327
linear curves, 323
logistic curves, 327–328
NAP (near-arithmetic progression) curves, 331
piecewise linear curves, 329–331
polynomial curves, 324
cost/benefit definitions, 320–321
core resources, 321–322
special cases, 322–323
subsidiary resources, 322
decoupling from value, 336–338
Crawford, Chris, 91–92, 118, 123
creative directors (CDs), 365
Csikszentmihalyi, Mihaly, 153
CTOs (chief technical officers), 365
-cultural interactivity, 152–153
currencies, 222, 243
cycles of engagement, 157–158
daily active users (DAU), 351
Dark Age of Camelot, 249–250, 313
Dark Souls, 297
data-driven design, 290–291
DAU (daily active users), 351
De Mundi Systemate (Newton), 29
Deadlands, 252
deadlock, 238
deriders, 57–58, 223
decisions, meaningful, 107–108
decoupling cost from value, 336–338
dedication, 134
deliverables, 262
demographic profiles, 202–203
depth
  concept document, 215
  interactivity, 167–168
  systemic, 83–86
Descartes, René, 16, 28–29
design documents, 287–289
design process
  designer’s loop, 181–182
  game analysis, 182–184
  for game parts, 286–287
  iterative nature of, 177–178
  structural parts, 180–181
  systemic loops, 180
  thematic architecture of, 178–180
design tools, 260
  rapid prototyping tools, 260–261
  spreadsheets, 261
  whiteboards, 260–261
designer-based balancing, 297–298
designer’s loop, 5, 128, 232–233
detailed design (concept document), 212
  core loops, 212–213
  interactivity, 213
  narrative and main systems, 213
  objectives and progression, 213
deterministic thinking, 16–21
development
development teams
  art and sound, 371–372
  game designers, 368
  organization chart, 366
  other team members, 372
  producers, 366–368
  programmers, 369–370
  QA (quality assurance), 370–371
  UI/UX, 369
  of game design, 117–119
  product development, 359–360
development teams
  art and sound, 371–372
  game designers, 368
  organization chart, 366
  other team members, 372
  producers, 366–368
  programmers, 369–370
  QA (quality assurance), 370–371
  UI/UX, 369
Dewey, John, 91–92
Diablo II, 247
The Dialogue Concerning the Two Chief World
  Systems (Galileo), 28–29
“Diamond Sutra,” 87
digital games, 176
digital prototypes, 387
distance
  articulatory, 125
  semantic, 125
distributions, 305–307
DLC (downloadable content), 211
documentation, 261
  concept documents, 193–194
    concept statement, 195–196
    depth and breadth, 215
    detailed design, 212–214
    elegance, 215
    game+player system, 214
    genre(s), 196–200
    product description, 206–212
    questions to consider, 216
    target audience, 200–204
    themes, 214
    USPs (unique selling points), 204–205
    working title, 195
design documents, 287–289
  mockups, 263–264
  prototyping, 263–264
  spreadsheet documentation, 261, 268–269, 289–291
  system design documents, 262
system technical design documents, 263
updating, 288–289
dominant strategy, 239
dopamine, 135
Dormans, Joris, 223–224, 260
downloadable content (DLC), 211
downward causality, 80–81
drains, 55–57, 223
Dwarf Fortress, 105–106
dynamic engines, 236
dynamics, MDA (Mechanics-Dynamics-Aesthetics) framework, 92–93

e
ecologies, 25–26, 248–249
ecological imbalances, 250–251
types of, 249–250
economic system balance, 341–342
arbitrage, 342–344
challenges of, 347–348
construction of game economy, 344–345
  market channel, 346–347
  price boundaries, 345–346
inflation, 342
economies, 240–241
constructing, 344–345
  market channel, 346–347
  price boundaries, 345–346
currencies, 243
economic issues, 245–246
  arbitrage, 342–344
  inflation, 246–248, 342
  stagnation, 248, 342
economic system balance, 341–342
  arbitrage, 342–344
  challenges of, 347–348
  construction of game economy, 344–347
inflation, 342
stagnation, 342
economies with engines, 243
examples of, 244–245
unfolding complexity in, 242–243
eCPU (effective cost per user), 353
effective cost per user, 353
Einstein, Albert, 41–42
electrons, 39–40
elevator pitches, 379–380
Ellenor, Geoff, 103
emergence, 78–80, 278–280
Emergence (Holland), 31–32
emotional goals, 111
emotional interactivity, 147–152
  challenges of, 151
context, 150
  meaning and, 208–209
  models of emotion, 148–150
  situations and goals, 150–151
encapsulation, 53, 277
endless runners, 140
endogenous meaning, 104
endorphins, 136
engagement, 129–130
  becoming and staying engaged, 134
  cycles of, 157–158
  definition of, 134
  interactive engagement, 158–159
  neurochemical engagement, 135–137
  psychological engagement, 137
engine-building games, 237
engines, 235–236
  boosting engines
    definition of, 236–237
    engine problems, 238–239
    examples of, 237–238
  braking engines, 239–240
economies with engines, 243
  engine-building games, 237
entrances, 207
environmental context, 203
EPs (executive producers), 365
Euclid, 28
EVE Online, 99
evolution of game design, 117–119
Excel, 261
exceptions to rules, 100–101
“excitement” games, 159–161
executive attention, 141
executive producers (EPs), 365
executive teams, 364–365
experience points (XP), 62, 252
experiencing systems, 36–37
explicit goals, 109
exploration in playtesting, 397–398
exponential curves, 324–327
extensibility, 176
extinction of behaviors, 130
extraversion, 202

F
F2P (free to play), 210–211
fairness, 309–310
fantasy, 206–207
Farmer, Randy, 342
Farmville, 146
faucet/drain economies, 55–57, 223
FBS (Function-Behavior-Structure) framework, 94
feedback, 280–281
    action/feedback interactivity, 138–139
    moment-to-moment gameplay, 141
    present-tense action, 139
    reflexive attention, 139
    stress and reward of fast action, 139–141
amount of, 282
comprehension and, 282–283
game behaviors and, 125–126
kinds of, 281–282
nautical game example, 283–285
player expectations and, 281
playtesting feedback, analyzing, 398–399
timing of, 282
The Fifth Discipline (Senge), 31
fifth generation (5G) networking, 34
final checks, 396
“finding the fun,” 174–175, 386
finishing games, 406–407
playtests, 397
fireflies, as chaotic system, 69–70
Firewatch, 144
first-order attributes, 271–272
first-time user experience (FTUE), 350–351
Five Factor Model, 202
five processes (Wu Xing), 312
flocking algorithms, 278–279
flow in interactive loops, 153–155
between parts, 54
of resources, 222–223
follow-up on pitches, 384–385
For Honor, 178
Forrester, John, 31
frameworks
    FBS (Function-Behavior-Structure), 94
    MDA (Mechanics-Dynamics-Aesthetics), 92–93
    other frameworks, 95
    SBF (Structure-Behavior-Function), 94
free to play (F2P), 210–211
friction, 239
FTL, 90, 114
FTUE (first-time user experience), 350–351
Fugitive, 147–148
Fuller, Buckminster, 43
full-stack programmers, 370
fun, 163
    characteristics of, 164
    definition of, 165–166
    “finding the fun,” 174–175, 386
    negative affect in gameplay, 166–167
function, definition of, 94
functional aspects of games, 102
    definition of function, 94
    functional elements as machines, 103
    internal model of reality, 103–106
    Dwarf Fortress example, 105–106
    endogenous meaning, 104
    second-order design, 104–105
    meaningful decisions, 107–108
    opposition and conflict, 108–109
    player’s mental model, 107
    possibilities for play, 102–103
    randomness, 106–107
    uncertainty, 106–107
Function-Behavior-Structure (FBS) framework, 94

G
game analysis, systems view of, 182–184
game balance, 180, 295
analytical balance, 348
analytical methods, 298, 299
    analytics-driven design, 301
    analytics-informed design, 301
cautions about, 301
    sample size and information distortion, 301–302
Tumbleseed example, 299–301
<table>
<thead>
<tr>
<th>Game Balance</th>
<th>Defining Principles</th>
<th>Design and Development</th>
<th>Prototyping and Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Balancing Loops</strong></td>
<td>61–62, 220–221</td>
<td><strong>Content and Systems</strong></td>
<td>114</td>
</tr>
<tr>
<td><strong>Economic System Balance</strong></td>
<td>341–342</td>
<td><strong>Defining the System</strong></td>
<td>296</td>
</tr>
<tr>
<td><strong>Arbitrage</strong></td>
<td>342–344</td>
<td><strong>Designer-Based Balancing</strong></td>
<td>297–298</td>
</tr>
<tr>
<td><strong>Challenges of</strong></td>
<td>347–348</td>
<td><strong>Economic System Balance</strong></td>
<td>344–347</td>
</tr>
<tr>
<td><strong>Construction of Game Economy</strong></td>
<td>344–347</td>
<td><strong>Inflation</strong></td>
<td>342</td>
</tr>
<tr>
<td><strong>Stagnation</strong></td>
<td>342</td>
<td><strong>Importance of</strong></td>
<td>296–297</td>
</tr>
<tr>
<td><strong>Intransitive Balance</strong></td>
<td>316</td>
<td><strong>Mathematical Methods</strong></td>
<td>303–304</td>
</tr>
<tr>
<td><strong>Parts</strong></td>
<td>332–334</td>
<td><strong>Attribute Values</strong></td>
<td>335–336</td>
</tr>
<tr>
<td><strong>Attribute Weights</strong></td>
<td>334–335</td>
<td><strong>Decoupling Cost from Value</strong></td>
<td>336–338</td>
</tr>
<tr>
<td><strong>Player Behavior Data</strong></td>
<td>350–351</td>
<td><strong>Player-Based Balancing</strong></td>
<td>298</td>
</tr>
<tr>
<td><strong>Acquisition and First Experience</strong></td>
<td>350–351</td>
<td><strong>Probability</strong></td>
<td>304–305</td>
</tr>
<tr>
<td><strong>Community</strong></td>
<td>354</td>
<td><strong>Changing Probabilities</strong></td>
<td>307</td>
</tr>
<tr>
<td><strong>Conversion</strong></td>
<td>352–353</td>
<td><strong>Cognitive Biases</strong></td>
<td>307–309</td>
</tr>
<tr>
<td><strong>Player Cohorts</strong></td>
<td>349–350</td>
<td><strong>Definition of</strong></td>
<td>304</td>
</tr>
<tr>
<td><strong>Player Information, Collecting</strong></td>
<td>349</td>
<td><strong>Fairness</strong></td>
<td>309–310</td>
</tr>
<tr>
<td><strong>Retention</strong></td>
<td>351–352</td>
<td><strong>Likely Occurrence of Unlikely Events</strong></td>
<td>310–311</td>
</tr>
<tr>
<td><strong>Usage</strong></td>
<td>353–354</td>
<td><strong>Probability Distributions</strong></td>
<td>305–307</td>
</tr>
<tr>
<td><strong>Probability</strong></td>
<td>304–305</td>
<td><strong>Randomization</strong></td>
<td>304–305</td>
</tr>
<tr>
<td><strong>Separate and Linked Events</strong></td>
<td>305</td>
<td><strong>Progression and Power Curves</strong></td>
<td>320</td>
</tr>
<tr>
<td><strong>Cost/Benefit Definitions</strong></td>
<td>320–323</td>
<td><strong>Exponential Curves</strong></td>
<td>324–327</td>
</tr>
<tr>
<td><strong>Linear Curves</strong></td>
<td>323</td>
<td><strong>Logistic Curves</strong></td>
<td>327–328</td>
</tr>
<tr>
<td><strong>NAP (Near-Arithmetic Progression)</strong></td>
<td>331</td>
<td><strong>Piecewise Linear Curves</strong></td>
<td>329–331</td>
</tr>
<tr>
<td><strong>Polynomial Curves</strong></td>
<td>324</td>
<td><strong>Progression Balancing</strong></td>
<td>338–339</td>
</tr>
<tr>
<td><strong>Pacing</strong></td>
<td>339–340</td>
<td><strong>Secondary Progression</strong></td>
<td>341</td>
</tr>
<tr>
<td><strong>Time and Attention</strong></td>
<td>340–341</td>
<td><strong>Transitive Balance</strong></td>
<td>315–316</td>
</tr>
<tr>
<td><strong>Achieving</strong></td>
<td>315–316</td>
<td><strong>Examples of</strong></td>
<td>311–314</td>
</tr>
<tr>
<td><strong>Requirements for</strong></td>
<td>314–315</td>
<td><strong>Game Concept</strong></td>
<td>188–190</td>
</tr>
<tr>
<td><strong>Blue-Sky Design</strong></td>
<td>194</td>
<td><strong>Concept Banks</strong></td>
<td>400</td>
</tr>
<tr>
<td><strong>Constraints</strong></td>
<td>191</td>
<td><strong>Concept Documents</strong></td>
<td>193–194</td>
</tr>
<tr>
<td><strong>Curation</strong></td>
<td>190</td>
<td><strong>Concept Statement</strong></td>
<td>195–196, 381–382</td>
</tr>
<tr>
<td><strong>Depth and Breadth</strong></td>
<td>215</td>
<td><strong>Detailed Design</strong></td>
<td>212–214</td>
</tr>
<tr>
<td><strong>Elegance</strong></td>
<td>215</td>
<td><strong>Genre(s)</strong></td>
<td>196–200</td>
</tr>
<tr>
<td><strong>Game+Player System</strong></td>
<td>214</td>
<td><strong>Product Description</strong></td>
<td>206–212</td>
</tr>
<tr>
<td><strong>Questions to Consider</strong></td>
<td>216</td>
<td><strong>Target Audience</strong></td>
<td>200–204</td>
</tr>
<tr>
<td><strong>Themes</strong></td>
<td>214</td>
<td><strong>USPs (Unique Selling Points)</strong></td>
<td>204–205</td>
</tr>
<tr>
<td><strong>Working Title</strong></td>
<td>195</td>
<td><strong>Defining the Concept</strong></td>
<td>188</td>
</tr>
<tr>
<td><strong>Desired Experience</strong></td>
<td>192–193</td>
<td><strong>Game Definitions</strong></td>
<td>90–92, 95–96</td>
</tr>
<tr>
<td><strong>Game Design Theory</strong></td>
<td>119</td>
<td><strong>Game Design Theory Compared to Game Theory</strong></td>
<td>119</td>
</tr>
<tr>
<td><strong>Development of</strong></td>
<td>117–119</td>
<td><strong>Second-Order Design</strong></td>
<td>104–105</td>
</tr>
<tr>
<td><strong>Game Designers</strong></td>
<td>171</td>
<td><strong>Approaches to</strong></td>
<td>172–173</td>
</tr>
<tr>
<td><strong>Design Process</strong></td>
<td>172–173</td>
<td><strong>Designer’s Loop</strong></td>
<td>181–182</td>
</tr>
<tr>
<td><strong>Iterative Nature of</strong></td>
<td>177–178</td>
<td><strong>Structural Parts</strong></td>
<td>180–181</td>
</tr>
<tr>
<td><strong>Systemic Loops</strong></td>
<td>180</td>
<td><strong>Thematic Architecture of</strong></td>
<td>178–180</td>
</tr>
<tr>
<td><strong>Finding the Fun</strong></td>
<td>174–175, 386</td>
<td><strong>Getting Started</strong></td>
<td>172</td>
</tr>
<tr>
<td><strong>Game Analysis</strong></td>
<td>182–184</td>
<td><strong>Inventors</strong></td>
<td>174</td>
</tr>
<tr>
<td><strong>Knowing Your Strengths and Weaknesses</strong></td>
<td>173</td>
<td><strong>Playtesting</strong></td>
<td>184–185</td>
</tr>
<tr>
<td><strong>Prototyping</strong></td>
<td>184–185</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
storytellers, 173
teamwork, 356
  balancing with needs of individuals, 363
  communication, 361–362
  development team organization, 366–372
  executive teams, 364–365
  practices of successful teams, 356–357, 360–361
  principles for, 363–364
  product development, 359–360
  product vision, 357–358
  studio roles, 365–366
  team size, 372
  teams as systems, 372–373
toomakers, 174
game frameworks
  FBS (Function-Behavior-Structure), 94
  MDA (Mechanics-Dynamics-Aesthetics), 92–93
  other frameworks, 95
  SBF (Structure-Behavior-Function), 94
game genres. See genre(s)
game loops. See loops
game mechanics, 101, 220–232
Game of Life, 278–279
Game of War, 146
Game Outcomes Project
  communication, 361–362
  needs of individuals, 363
  practices of successful teams, 356–357, 360–361
  product development, 359–360
  product vision, 357–358
  summary, 363–364
game parts, 267–268
  attributes, 270–273
    first-order attributes, 271–272
    ranges, 272–273
    second-order attributes, 271–272
    third-order attributes, 271–272
  behaviors, 276–277
    emergence, 278–280
    feedback, 280–283
    generic, modular behaviors, 277–278
    local action, 277
    looping systems, 285–286
    defining, 268–269
design process, 286–287
documentation for
  design documents, 287–289
  spreadsheet details, 289–291
  updating, 288–289
internal state of, 270
nautical game example
  attributes, 275
  behaviors, 283–285
  core parts, 274–275
  detail design process, 275–276
  game concept, 273–274
  questions to consider, 291
  simple/atomic parts, 268
types of, 269–270
game+player system, 122, 214
game production, 399
  concept phase, 401
  finishing games, 406–407
  iterative design, 399–400
    iterative production, 400–401
    stage gating, 400
  preproduction phase, 401–402
  features and assets, 402
  length of, 403
  project plans, 402–403
  production phase, 403–404
    alpha milestone, 405–406
    beta/first release milestone, 406
    commercial release, 406
    production analytics, 404–405
game progression
  balancing, 338–339
  pacing, 339–340
  secondary progression, 341
  time and attention, 340–341
core loops, 213
designing, 179
progression and power curves
  cost/benefit definitions, 320–323
  exponential curves, 324–327
  linear curves, 323
  logistic curves, 327–328
  NAP (near-arithmetic progression) curves, 331
  piecewise linear curves, 329–331
  polynomial curves, 324
game prototyping, 184–185, 263–264. See also playtesting
 analog prototypes, 387
 answering questions with, 388–389
 definition of, 386–387
 digital prototypes, 387
 getting started with, 388
 intended audience for, 389–390
 keeping separate, 387
 moving fast, 390

game structure
 architectural and thematic elements, 111–112
 autotelic experience, 114–115
 content and systems, 112–114
 meaning, 116–117
 narrative, 115–116
 themes, 116–117
 definition of, 94
 depth in, 84–86
 functional aspects, 102
 functional elements as machines, 103
 internal model of reality, 103–106
 meaningful decisions, 107–108
 opposition and conflict, 108–109
 player goals, 109–111
 player's mental model, 107
 possibilities for play, 102–103
 randomness, 106–107
 uncertainty, 106–107
 game mechanics, 101
 metagaming, 101–102
 repeated games, 102
 rules, 99–101
 structural parts
 design process for, 180–181
 game mechanics, 101
 metagaming, 101–102
 repeated games, 102
 rules, 99–101
 tokens, 98–99
 tokens, 98–99
 game systems, 253. See also documentation
 combat systems, 256
 construction systems, 257
 game+player system, 122
 progression systems, 253–256
 qualities of, 175–176
 skill and technological systems, 257
 social and political systems, 257
 tools for designing, 260
 rapid prototyping tools, 260–261
 spreadsheets, 261
 whiteboards, 260–261

game theory versus game design
 theory, 119

game types
 agon games, 91
 alea games, 91
 ilinx games, 91
 ludus-paidia spectrum, 91
 mimicry games, 91

game world
 concept document, 210
 designing, 179
 rules, 99–101

game-mediated social interaction, 144

playground loops, 235–236. See also interactive

loop

ecologies, 248–249
 ecological imbalances, 250–251
 kinds of, 249–250
 economies, 240–241
 currencies, 243
 economic issues, 245–248
 economies with engines, 243
 examples of, 244–245
 unfolding complexity in, 242–243
 engines, 235–236
 boosting engines, 236–239
 braking engines, 239–240

game's model loop, 225

Gaming the System (Tekinbas et al.), 33

Gaussian distribution, 306

Gell-Mann, Murray, 41

General Systems (Bertalanffy), 31

general systems theory, 33

generic, modular behaviors, 277–278

genre(s), 232

genres, 196–200

geocentric model of universe, 28

géo-héliocentric model of universe, 28

Gestalt psychology, 22, 30

Gilbert, Ron, 188–189
gluon field, 42
Go, 84–86, 98, 130, 142–143
goals
defining, 258
emotional interactivity, 150–151
identifying, 376–377
player goals, 109–111
playtesting, 392
Goethe, Johann Wolfgang von, 30
Gone Home, 161–162, 206–207
Google Docs, 261
greedy reductionism, 18–19
Greenspan, Alan, 74
Griesemer, James, 155–156
“grinding” gameplay, 259–260
The Grizzled, 147
grouping, 145
growth, limits to, 72–75
Guitar Hero, 196

Habitat, 342
habituation, 254–256
Halley, Edmund, 29
Halo 2, 155–156
Halo 3, 155–156
heating an oven, 23–25
hedonic fatigue, 254, 255
heliocentric model of solar system, 28–29
Hidden Order (Holland), 31–32
hierarchy of loops, 233–235
history
of game design, 117–119
of systems thinking, 28–30
Hobbes, Thomas, 38
holistic thinking, 21–22
Holland, John, 31–32
homeostasis, 82
Homo Ludens (Huizinga), 90
For Honor, 188
hooks, 131
“house rules,” 101–102
Howe, Chelsea, 157
Huizinga, Johan, 90
hydrogen atoms, 39
hypothesis-driven analysis, 17

**“iceberg” approach, 382–384**
iconography for loop components, 223–224
ideation, 188. See also blue-sky design
identity
systems as things, 87–88
Theseus’ ship paradox, 38–39
idle games, 198
ilinx games, 91, 164
immersion-creativity motivations, 201
implicit goals, 109–110
individuals’ needs, balancing with team needs, 363
inflation, 246–248, 342
information distortion, sample size and, 301–302
instant goals, 110
integrative levels, 42
intentional choices, 127
interactive loops, 137–138, 225–226. See also gameplay loops
action/feedback interactivity, 138–141
moment-to-moment gameplay, 141
present-tense action, 139
reflexive attention, 139
stress and reward of fast action, 139–141
blending types of, 142–143
cognitive interactivity, 141–143
core loops, 127, 156–157, 226–227
concept document, 212–213
elements of, 227–231
game mechanics, 220–232
cultural interactivity, 152–153
designer’s loop, 128, 181–182
emotional interactivity, 147–152
challenges of, 151
core loops, 156–157
cycles of engagement, 157–158
narrative and interactive engagement, 158–159

flow in, 153–155
social interactivity, 143–146
game-mediated social interaction, 144
techniques for encouraging, 144–146
time-scale view of, 155–159
core loops, 156–157
cycles of engagement, 157–158
narrative and interactive engagement, 158–159
interactivity, 121
  behaviors, 124–125
    behavior extinction, 130
    game behaviors and feedback, 125–126
    intentional choices, 127
    player behaviors and cognitive load, 125–126, 159
  definition of, 104–105, 123
  depth, 167–168
  detailed design (concept document), 213
  elegance, 167–168
  fun, 163
    characteristics of, 164
    definition of, 165–166
    “finding the fun,” 174–175, 386
    negative affect in gameplay, 166–167
  game+player system, 122, 214
  interactive loops, 137–138
    action/feedback interactivity, 138–141
    blending types of, 142–143
    cognitive interactivity, 141–143
    core loops, 127, 156–157
    cultural interactivity, 152–153
    cycles of engagement, 157–158
    designer’s loop, 128
    emotional interactivity, 147–152
    flow in, 153–155
    narrative and interactive engagement, 158–159
    social interactivity, 143–146
    time-scale view of, 155–159
  interactivity budget, 161–163
  internal state, 124
  mental load, 159–161
  mental models
    arousal, 129–130, 132–133
    building, 131–132
    engagement, 129–130, 134–137
    systemic approach to, 122–123
    whole experience, 129
  interactivity budget, 161–163
  interconnected world, 34–36
  internal model of reality, 103–106
    Dwarf Fortress example, 105–106
    endogenous meaning, 104
    second-order design, 104–105
  internal state, 124, 270
  intransitive balance, 316
  inventors, 174
irrational exuberance, 73–74
iterative design, 399–400
  iterative production, 400–401
  stage gating, 400
  iterative production, 400–401

J
  Jobs, Steve, 190
  Journal of Computer Game Design, 118
  Journey, 147–148
  Joyce, James, 41
  junior game designers, 368

K
  key moments, 207–208
  key performance indicators (KPIs), 349
  Knights of the Old Republic, 116, 147–148
  Koffka, Kurt, 22, 30, 60
  Kohler, Wolfgang, 30
  KPIs (key performance indicators), 349
  Kristallnacht, 152
  Kuhn, Thomas, 17

L
  LAMP programmers, 370
  The Last of Us, 206–207
  Lau, Edmund, 35
  Lawrence, D. H. 44–45, 60, 96
  League of Legends, 162, 203
  “legacy” games, 101
  Legend of Zelda, 256
  leveling treadmill, 260
  levels of organization, 81–82
  lifetime value (LTV), 352
  limited-play pricing, 211
  limits to growth, 72–75
  linear curves, 323
  linear effects, 63–65
  linear interpolation, 329
  linked events, probability and, 305
  “living” documents, 288–289
  local action, 277
  localization, 212
  logistic curves, 327–328
  long-term cognitive interactivity, 141
  long-term goals, 111
Mechanics-Dynamics-Aesthetics (MDA) framework, 92–93
median, 306
meetings, pitch, 380–381
Meier, Sid, 92
mental load, 159–161
mental models, 107
arousal, 129–130, 132–133
building, 131–132
engagement, 129–130
becoming and staying engaged, 134
definition of, 134
neurochemical engagement, 134
player’s mental loop, 225
metacognition, 14
metagaming, 101–102
metastability, 42–45
Michigan Fish Test, 14–15
Microsoft Excel, 261
milestones
alpha milestone, 405–406
beta/first release milestone, 406
military training, transitive balance in, 313
mimicry games, 91
Minecraft, 117
minimum viable product (MVP), 400–401
misleading feedback, 283
MMOs (massively multiplayer online games), 95
definition of, 198
faucet/drain economies in, 55–57
game-mediated social interaction, 144
mockups, 263–264
mode, 306
modeling
mathematical, 65–67
systemic, 66–67
models of emotion, 148–150
molecules, 43–45
Mollenkopf, Stephen, 34
moment-to-moment gameplay, 141
monetization, 353
concept document, 210–211
designing, 179–180
Monopoly, 61, 101–102, 208–209
boosting engines, 237
braking engines, 239–240
reinforcing loops in, 221
tokens in, 99
zero-sum view in, 245–246
monthly active users (MAU), 351–352
Monty Hall problem, 307–309
Monument Valley, 100
Morningstar, Chip, 342
morphology, 30
motivations, 200–202
MVP (minimum viable product), 402

N
name-value pairs, 270
NAP (near-arithmetic progression) curves, 331
narrative, 115–116
designing, 179
detailed design (concept document), 213
interactive engagement, 158–159
nautical game example, 275–276
attributes, 275
behaviors, 283–285
core parts, 274–275
detail design process, 275–276
game concept, 273–274
near misses, 310–311
near-arithmetic progression (NAP) curves, 331
negative affect in gameplay, 166–167
negative feedback loops, 60–62
Nesbitt, Richard, 14
NetLogo, 260–261
neurochemical engagement, 135–137
neuroticism, 202
neurons, 41, 44
Newton, Isaac, 16
No Man’s Sky, 117
nonlinear effects, 63–66
nonsensical feedback, 283
norepinephrine, 136
normal distribution, 306
normalization, 272

O
objectives (detailed design document), 213
Objectivism, 153
observation of playtests, 397
“The One Question,” 196
oneness, 47
openness to experience, 202
OpenOffice, 261
opposition, 108–109
organization, levels of, 81–82
organization charts, 366
oven-heating loop, 23–25
oxytocin, 136

P
P&L responsibility, 365
pacing, 339–340
Pac-Man, 140
paidia, 91, 95
pairs, name-value, 270
Papers, Please, 147, 161–162
Parable of the Polygons, 279
paradigm shifts, 17
parasympathetic nervous system, 230
parts of games, 267–268. See also loops
attributes, 270–273
first-order attributes, 271–272
ranges, 272–273
second-order attributes, 271–272
third-order attributes, 271–272
balancing, 332–334
attribute values, 335–336
attribute weights, 334–335
decoupling cost from value, 336–338
behaviors, 276–277
emergence, 278–280
feedback, 280–283
generic, modular behaviors, 277–278
local action, 277
looping systems, 285–286
defining, 268–269
design process, 286–287
documentation for
design documents, 287–289
spreadsheet details, 289–291
updating, 288–289
game mechanics, 101
internal state of, 270
metagaming, 101–102
nautical game example
attributes, 275
behaviors, 283–285
core parts, 274–275
detail design process, 275–276
game concept, 273–274
questions to consider, 291
repeated games, 102
rules, 99–101
simple/atomic parts, 268
tokens, 98–99
types of, 269–270
parts of systems
behaviors, 53
boundaries, 52–53
converters, 57–58
deciders, 57–58
flow between, 54
resources, 54
sinks, 55–57
sources, 54
state, 51–52
stocks, 54–55
A Pattern Language (Alexander), 32, 46–47
patterns, 46–47
payload of pre-processed information, 158
Peggle, 140–141
pendulum, path of, 19–20
persistence, 51, 82
personality traits, 202
phases of game production
concept, 401
finishing games, 406–407
preproduction, 401–402
features and assets, 402
length of, 403
project plans, 402–403
production, 403–404
alpha milestone, 405–406
beta/first release milestone, 406
commercial release, 406
production analytics, 404–405
phenomenological thinking, 15
physiological needs (Maslow's hierarchy), 149
piecewise linear curves, 329–331
pitching games, 376
elevator pitches, 379–380
follow-up, 384–385
pitch content, 381–382
call to action, 384
“iceberg” approach, 382–384

PITCHING GAMES
pitch meetings, 380–381
preparation
  audience identification, 378
goal identification, 376–377
  knowing your material, 378–379
Pixar, 184
plans, project, 402–403
platformer games, 198
platforms, defining in concept document, 211–212
player-based balancing, 298
players. See also playtesting
  agency, 100
  behaviors and cognitive load, 125–126, 159
  choosing for playtesting, 392–393
  game+player system, 122, 214
  goals, 109–111
  intuition, 298
  mental loops, 225
  mental models, 107, 225
  motivations, 200–202
  as part of larger system, 97–98
  personality traits, 202
  player agency, 100
  player behavior data
    acquisition and first experience, 350–351
    community, 354
    conversion, 352–353
    retention, 351–352
    usage, 353–354
  player cohorts, 349–350
  player expectations, 281
  player experience, 206
    emotions and meaning, 208–209
    fantasy, 206–207
    key moments, 207–208
    linking with system design, 259–260
  player information, collecting, 349
  player-based balancing, 298
  player-to-player economies, 244–245
  target audience, 200–204
    demographics, 202–203
    environmental context, 203
    motivations, 200–202
    psychographics, 200–202
    as type of opposition, 109
  player-to-player economies, 244–245
playtesting, 184–185. See also prototyping
  feedback, analyzing, 398–399
  finishing, 397
  goals of, 392
  importance of, 391
  player-based balancing, 298
  preparation, 393
    final checks, 396
    scripts, 393–394
    surveys, 394–396
  running playtests, 396–397
  test subjects, 392–393
  testing methods, 397–398
  when to test, 391
Poincaré, Henri, 46
Poker, 95
political systems, 257
polynomial curves, 324
Portal, 197
positive feedback loops, 60–62
possibilities for play, 102–103
postmortems, 405
power curves. See progression and power curves
Power Grid, 183, 221
predator-prey equations, 65–66
predictability, 175–176
premium pricing, 210
preparation
  for pitching games
    audience identification, 378
    goal identification, 376–377
    knowing your material, 378–379
  for playtesting, 393
    final checks, 396
    scripts, 393–394
    surveys, 394–396
preproduction phase (game production), 401–402
  features and assets, 402
  length of, 403
  project plans, 402–403
  presentations (pitches), 382–384
  present-tense action, 139
  prestige loops, 247
  price boundaries, 345–346
  primary emotions, 147
Principia Mathematica (Newton), 29
probability
  changing probabilities, 307
cognitive biases, 307–309
definition of, 304
fairness, 309–310
likely occurrence of unlikely events, 310–311
probability distributions, 305–307
randomization, 304–305
separate and linked events, 305
process of game design
designer’s loop, 181–182
iterative nature of, 177–178
structural parts, 180–181
systemic loops, 180
thematic architecture of, 178–180
processes, complicated, 58–59
producers, 366–368
product descriptions, 206
game world fiction, 210
monetization, 210–211
player experience, 206
scope, 212
technology, tools, and platforms, 211–212
visual and audio style, 209–210
product development
development teams
  art and sound, 371–372
game designers, 368
organization chart, 366
other team members, 372
producers, 366–368
programmers, 369–370
QA (quality assurance), 370–371
UI/UX, 369
top-priority items for, 359–360
product managers, 367–368
product vision, 357–358
production, 399
concept phase, 401
finishing games, 406–407
iterative design, 399–400
  iterative production, 400–401
  stage gating, 400
preproduction phase, 401–402
  features and assets, 402
  length of, 403
  project plans, 402–403
production phase, 403–404
  alpha milestone, 405–406
  beta/first release milestone, 406
  commercial release, 406
  production analytics, 404–405
production chains, 222
production phase (game production), 403–404
  alpha milestone, 405–406
  beta/first release milestone, 406
  commercial release, 406
  production analytics, 404–405
programmers, 369–370
progression
  balancing, 338–339
  pacing, 339–340
  secondary progression, 341
time and attention, 340–341
core loops, 213
designing, 179
progression and power curves
  cost/benefit definitions, 320–323
  exponential curves, 324–327
  linear curves, 323
  logistic curves, 327–328
NAP (near-arithmetic progression) curves, 331
  piecewise linear curves, 329–331
  polynomial curves, 324
progression systems, 253–256
progression and power curves, 320
  cost/benefit definitions, 320–323
  exponential curves, 324–327
  linear curves, 323
  logistic curves, 327–328
NAP (near-arithmetic progression) curves, 331
  piecewise linear curves, 329–331
  polynomial curves, 324
project managers, 367–368
project plans, 402–403
protons, 39–42, 44
prototyping, 184–185, 263–264. See also playtesting
  analog prototypes, 387
  answering questions with, 388–389
  definition of, 386–387
digital prototypes, 387
going started with, 388
intended audience for, 389–390
keeping separate, 387
moving fast, 390
psychographics, 200–202
psychological engagement, 137

Q
QA (quality assurance), 370–371
Qualcomm, 34
quality assurance (QA), 370–371
“quality without a name,” 47, 60, 96
Quantic Foundry
cognitive threshold diagram, 160–161
gamer motivations, 200–202
quarks, 41–44

R
rabbits, introduction into Australia, 25–26
The Rainbow (Lawrence), 44–45
Rampart, 238
random determination, 108
randomization, 304–305
randomness
creating uncertainty with, 106–107
random determination, 108
random effects, 67–68
randomization, 304–305
range, definition of, 306
ranges, attribute, 272–273
rapid prototyping tools, 260–261
reality, internal model of, 103–106
Dwarf Fortress example, 105–106
endogenous meaning, 104
second-order design, 104–105
Realm of the Mad God, 145
Realm vs. Realm combat, 249–250, 313
reductionist thinking, 16–21
reflexive attention, 139
reinforcing loops, 61–62, 220
repeated games, 102
The Resistance, 99
resources, 54, 221–222
core resources, 321–322
subsidiary resources, 322
retention, player, 351–352
reward of fast action, 139–141
Reynolds, Craig, 36
rhythm games, 198
Risk, 245–246
Rock Band, 196
Rock-Paper-Scissors, 102, 311–313
Rock-Paper-Scissors-Lizard-Spock, 312
roguelike games, 199
role-playing games. See RPGs (role-playing games)
roles
complementary roles, 146
team roles, 364
art and sound, 371–372
development team organization chart, 366
executive teams, 364–365
game designers, 368
other team roles, 372
producers, 366–368
programmers, 369–370
QA (quality assurance), 370–371
studio roles, 365–366
UI/UX, 369
Romantic philosophers, 30
Romero, Brenda, 152
Rovio, 172
RPGs (role-playing games), 252
definition of, 199
experience points in, 62
rules
design process for, 180–181
“house rules,” 101–102
metagaming, 101–102
purpose of, 99–101
as type of opposition, 108–109
RuneScape, 325
running playtests, 396–397
RvR (Realm vs. Realm combat), 313

S
safety (Maslow’s hierarchy of needs), 149
Salen, Katie, 92
sample size, 301–302
satiation, 255
SBF (Structure-Behavior-Function) framework, 94
schedules, variable, 244
schools of fish, shape of, 78–79
scientific method, 16–21
scope
defining in concept document, 212
scope creep, 404
scripts for playtesting, writing, 393–394
scrum, 400–401
second difference, 324, 325
second-order attributes, 271–272
second-order design, 104
security (Maslow's hierarchy of needs), 149
self-actualization (Maslow's hierarchy of needs), 149
selling points, unique, 204
semantic distance, 125
Senet, 118
separate events, probability and, 305
serotonin, 135
Settlers of Catan, 99, 246
shape of schools of fish, 78–79
shooters, 196–197
short-term cognitive interactivity, 141
short-term goals, 110
Sid Meier's Pirates, 116
Siebert, Horst, 19
sigmoid (logistic) curves, 327–328
SimCity, 172, 209
Simkin, Marvin, 257
simple collections, 58–59
simple resources, 222
simple/atomic parts, 268. See also game parts
The Sims, 209
simulation games, 199
single-player games, 97
sinks, 55–57, 223
size of teams, 372
skill and attainment (Maslow's hierarchy of needs), 149
skill and technological systems, 257
Slime Rancher, 279–280
Smuts, Jan Christian, 22, 30, 60
social casino games, 310–311
social goals, 111
social interactivity, 143–146
game-mediated social interaction, 144
techniques for encouraging, 144–146
social needs (Maslow's hierarchy), 149
social reciprocity, 146
social referents, 145
social systems, 257
software engineers, 369–370
solar system, views of, 28–29
sound design, 371–372
sources, 54, 222
space for play, designing, 180
Splendor, 117, 184, 237–238
sports games, 199
spreadsheet documentation, 261, 268–269, 289–291
spreadsheet specific, 268–269
sprints, 400–401
spurious correlations, 21
stage gating, 400
stagnation, 248, 342
Star Realms, 256
state
definition of, 51–52
internal state, 124, 270
statements
concept statement, 195–196, 381–382
x-statements, 205
static engines, 236
Steambirds: Survival, 114
Stellaris, 107, 162
stock market crash of 1929, 73–74
stocks, 54–55, 222–223
story-driven games, 115–116
storytellers, 173
strategy games, 159–161, 199
strengths, knowing, 173
stress of fast action, 139–141
structural coupling, 82
structural parts
design process for, 180–181
game mechanics, 101
metagaming, 101–102
repeated games, 102
rules, 99–101
tokens, 98–99
structure (game)
arbitrary and thematic elements,
111–112
autotelic experience, 114–115
content and systems, 112–114
meaning, 116–117
narrative, 115–116
themes, 116–117
definition of, 94
depth in, 84–86
functional aspects, 102
  functional elements as machines, 103
  internal model of reality, 103–106
meaningful decisions, 107–108
opposition and conflict, 108–109
player goals, 109–111
player's mental model, 107
possibilities for play, 102–103
randomness, 106–107
uncertainty, 106–107
game mechanics, 101
metagaming, 101–102
repeated games, 102
rules, 99–101
structural parts
  design process for, 180–181
  game mechanics, 101
  metagaming, 101–102
  repeated games, 102
  rules, 99–101
tokens, 98–99
tokens, 98–99
Structure-Behavior-Function (SBF)
  framework, 94
studio roles, 365–366
style guides, 210
subatomic structure, 39–42
subjective contour, 22
subprime lending, 74–75
subsidiary resources, 322
Sudoku, 141
Sumitomo Chemical Plant, 34–35
Super Meat Boy, 297
supply and demand, 245
surveys, creating, 394–396
symmetrical distribution, 307
sympathetic nervous system, 230
synergy, metastability and, 42–45
system design documents, 262
system technical design documents, 263
systema, 28
systemic depth and elegance, 83–86
systemic game designers. See
  game designers
systemic games, 113–114
systemic loops, 180
systemic machines, 235–236
ecologies, 248–249
  ecological imbalances, 250–251
  kinds of, 249–250
economies, 240–241
  currencies, 243
  economic issues, 245–248
  economies with engines, 243
  examples of, 244–245
  unfolding complexity in, 242–243
  engines, 235–236
  boosting engines, 236–239
  braking engines, 239–240
systemic modeling
  overview of, 66–67
systemic organization of games, 96–97
  architectural and thematic elements, 111–117
  functional aspects, 102–111
  player as part of larger system, 97–98
  structural parts, 98–102
systemic organization of games, 96–97
  player as part of larger system, 97–98
structural parts
  game mechanics, 101
  metagaming, 101–102
  repeated games, 102
  rules, 99–101
tokens, 98–99
systemic thinking. See systems thinking
systems
  adaptability, 82
  complicated versus complex, 58–60
  definition of, 49–51
downward causality, 80–81
  emergence, 78–80
game systems, 112, 253
  balancing, 114
  combat systems, 256
  construction systems, 257
  content-driven games, 112–113
  progression systems, 253–256
  skill and technological systems, 257
  social and political systems, 257
  systemic games, 113–114
levels of organization, 81–82
loops, 60–61
  balancing loops, 61–62
  chaotic effects, 68–70
combined effects, 63–65
limits to growth, 72–75
linear effects, 63–65
mathematical modeling, 65–67
nonlinear effects, 63–66
random effects, 67–68
reinforcing loops, 61–62
systemic modeling, 66–67
tragedy of the commons, 76–77
trophic cascades, 77–78
unintended consequences loops, 70–71
metacognition, 14
parts of
behaviors, 53
boundaries, 52–53
complicated processes, 58–59
converters, 57–58
deciders, 57–58
flow between, 54
resources, 54
sinks, 55–57
sources, 54
state, 51–52
stocks, 54–55
persistence, 82
structural coupling, 82
systemic depth and elegance, 83–86
systems thinking, 23–27
current state of, 32–33
examples of, 23–27
experiencing systems, 36–37
history of, 28–30
importance of, 33
and interconnected world, 34–36
metastability and synergy, 42–45
patterns and qualities, 46–47
rise of, 30–32
subatomic structure, 39–42
Theseus’ ship paradox, 38–39, 45–46
The Systems View of Life (Capra), 32–33

T

tabletop games, 176
Tacoma Narrows Bridge, 68–69
The Tao of Physics (Capra), 32
target audience, 200–204
demographics, 202–203
environmental context, 203
identifying, 378, 389–390
motivations, 200–202
psychographics, 200–202

teamwork, 356
balancing with needs of individuals, 363
communication, 361–362
development teams
art and sound, 371–372
game designers, 368
organization chart, 366
other team members, 372
producers, 366–368
programmers, 369–370
QA (quality assurance), 370–371
UI/UX, 369
executive teams, 364–365
practices of successful teams, 356–357, 360–361
principles for, 363–364
product development, 359–360
product vision, 357–358
studio roles, 365–366
team size, 372
teams as systems, 372–373

technology
defining in concept document, 211–212
technological systems, 257
Temple Run, 140
Terraria, 116
testing. See playtesting
themetic architecture, 178–180
themetic game elements, 111–112
autotelic experience, 114–115
content and systems, 112
balancing, 114
content-driven games, 112–113
systemic games, 113–114
meaning, 116–117
narrative, 115–116
themes, 116–117
themes, 116–117
concept document, 214
themetic architecture, 178–180
themetic game elements, 111–112
autotelic experience, 114–115
content and systems, 112
meaning, 116–117
narrative, 115–116
themes, 116–117
Theseus’ ship paradox, 38–39, 45–46
“thingness”
systems as things, 87–88
Theseus’ ship paradox, 38–39
thinking aloud in playtesting, 398
Thinking in Systems (Meadows), 32
“The Third Thing” (Lawrence), 44, 60, 96
third-order attributes, 271–272
This War of Mine, 147, 208–209
thought processes
differences in, 14–15
holistic thinking, 21–22
metacognition, 14
phenomenological thinking, 15
reductionist thinking, 16–21
systems thinking, 23–27, 182–184
current state of, 32–33
experiencing systems, 36–37
history of, 28–30
importance of, 33
and interconnected world, 34–36
metastability and synergy, 42–45
patterns and qualities, 46–47
rise of, 30–32
subatomic structure, 39–42
Theseus’ ship paradox, 38–39, 45–46
three-door problem, 307–309
Tic-Tac-Toe, 107
time
balancing progression with, 340–341
time-scale view of interactive loops, 155–159
core loops, 156–157
cycles of engagement, 157–158
narrative and interactive engagement, 158–159
timestamps in documentation, 291
The Timeless Way of Building (Alexander), 32, 47
time-scale view of interactive loops, 155–159
core loops, 156–157
cycles of engagement, 157–158
narrative and interactive engagement, 158–159
timestamps in documentation, 291
timing of feedback, 282
title (concept document), 195
tokens, 98–99, 180–181, 221–222
tools, 260
defining in concept document, 211–212
rapid prototyping tools, 260–261
spreadsheets, 261
whiteboards, 260–261
Torg, 252
total progression ratio, 324
tower defense games, 199–200
Townsend, Michael, 325
toymakers, 174
Tozour, Paul, 356
trading economies, 244
tragedy of the commons, 76–77
Train, 152–153, 162–163, 167
transitive balance
achieving, 315–316
examples of, 311–314
requirements for, 314–315
trrophic cascades, 77–78
Tumbleseed, 299–301, 351
Twilight Struggle, 90
twisting ideas, 189–190
Tycho Brahe, 28
U
UI (user interface) team role, 369
Ultima Online, 55–57
uncertainty, 106–107
unfolding complexity, 242–243
unintended consequences loops, 70–71
unique selling points (USPs), 204–205
unlikely events, likely occurrence of, 310–311
Up, 184
updating documentation, 288–289
upward causality, 80–81
U.S. military training, transitive balance in, 313
usage-based metrics, 353–354
user interactions (UX) team role, 369
user interface (UI) team role, 369
USPs (unique selling points), 204–205
UX (user interactions) team role, 369

V
valence, 148
values
assigning to attributes, 335–336
decoupling cost from, 336–338
design process for, 180–181
VandenBerghe, Jason, 178
Varela, Francisco, 32–33
variable schedules, 244
vasopressin, 136
vertical slice of game, 389
vigor, 134
vision, product, 357–358
visual artists, 371–372
visual feedback, 281–282
visual style, 209–210

W
water molecules, 43–45
ways of thinking. See thought processes
weaknesses, working to, 173
weapons, balancing, 332–334
attribute values, 335–336
attribute weights, 334–335
decoupling cost from value, 336–338
weight coefficients, assigning to attributes,
Wertheimer, Max, 30
whiteboards, 260–261
whole experience. See game concept
wholes, 86
Wiener, Norbert, 31
The Witcher 3, 130
Wittgenstein, Ludwig, 95–96
“Wizard of Oz” protocol, 398
wolves, reintroduction into Yellowstone
National Park, 26–27
working title (concept document), 195
world (game)
concept document, 210
designing, 179
rules, 99–101
World of Warcraft, 130, 249–250, 314
arbitrage, 343–344
NAP (near-arithmetic progression) curves, 331
stock limitations in, 256
Wright, Will, 172, 209
writing scripts for playtesting, 393–394
Wu Xing, 312

X
XP (experience points), 62, 252
x-statements, 205

Y
Yellowstone National Park, reintroduction of
wolves into, 26–27
Yerkes-Dodson curve, 153–154
Yerkes-Dodson Law, 132–133

Z
zero-sum view, 245–246
Zimmerman, Eric, 92