

# WRITING INTERACTIVE FICTION WITH TWINE

PLAY INSIDE A STORY



TECHNICAL  
ACCURACY  
GUARANTEED BY  
TWINE CREATOR,  
CHRIS KLIMAS!

MELISSA FORD



FREE SAMPLE CHAPTER

SHARE WITH OTHERS



# Writing Interactive Fiction with **TWINE**

Melissa Ford

**que**<sup>®</sup>

800 East 96th Street,  
Indianapolis, Indiana 46240 USA

## Writing Interactive Fiction with Twine

### Copyright © 2016 by Melissa Ford

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/](http://www.pearsoned.com/permissions/). No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-7897-5664-0

ISBN-10: 0-7897-5664-1

Library of Congress Control Number: 2015956273

First Printing: May 2016

### Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Que Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

### Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

### Special Sales

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](mailto:corpsales@pearsoned.com) or (800) 382-3419.

For government sales inquiries, please contact [governmentsales@pearsoned.com](mailto:governmentsales@pearsoned.com).

For questions about sales outside the U.S., please contact [intlcs@pearson.com](mailto:intlcs@pearson.com).

#### Editor-in-Chief

Greg Wiegand

#### Acquisitions Editor

Rick Kughen

#### Development Editor

Todd Brakke

#### Managing Editor

Sandra Schroeder

#### Project Editors

Tonya Simpson  
Seth Kerney

#### Copy Editor

Kitty Wilson

#### Indexer

Works Publishing

#### Proofreader

The Wordsmithery  
LLC

#### Technical Editor

Chris Klimas

#### Editorial Assistant

Cindy Teeters

#### Cover Designers

Mark Shirar  
Chuti Prasertsith

#### Compositor

Bronkella Publishing

# Contents at a Glance

	Foreword	xvi
	Introduction	1
<b>CHAPTER 1</b>	The Nuts and Bolts: Getting Started with Twine	5
<b>CHAPTER 2</b>	Using Choice to Create Agency	13
<b>CHAPTER 3</b>	Creating a Vivid Setting	29
<b>CHAPTER 4</b>	Designing Puzzles	49
<b>CHAPTER 5</b>	Building Objects with Variables	67
<b>CHAPTER 6</b>	Stasis, Catalyst, and Climax: Understanding Story Arc	91
<b>CHAPTER 7</b>	Exploring Interactive Fiction Genres	107
<b>CHAPTER 8</b>	Constructing Believable Characters	127
<b>CHAPTER 9</b>	Maintaining an Inventory	147
<b>CHAPTER 10</b>	Achieving Proper Pacing	171
<b>CHAPTER 11</b>	Foreshadowing Important Clues	183
<b>CHAPTER 12</b>	Making a Role-Playing Game	195
<b>CHAPTER 13</b>	Combat and Consequences in Role-Playing Games	213
<b>CHAPTER 14</b>	Wiggling Words: Changing the Text Appearance	233
<b>CHAPTER 15</b>	Story Style: Changing the Game Appearance in Harlowe	243
<b>CHAPTER 16</b>	Adding Images in a New Format	253
<b>CHAPTER 17</b>	Setting Up the Pre-Story, Central Story, and Post-Story	267
<b>CHAPTER 18</b>	Developing a Strong Player Character	283
<b>CHAPTER 19</b>	Balancing Pacing and Action	307
<b>CHAPTER 20</b>	Keeping Players Engaged	327
<b>CHAPTER 21</b>	Show, Don't Tell	347
<b>CHAPTER 22</b>	Story Style: Changing the Game Appearance in SugarCube	363
<b>CHAPTER 23</b>	Finishing Up and Clicking Publish	381
	Index	397

# Table of Contents

	Foreword	xvi
	<b>Introduction to Interactive Fiction</b>	<b>1</b>
	Wait a Second...What Is a Choose Your Own Adventure Book?	1
	Are They Games or Stories?	2
	What Type of Games Can I Make with Twine?	2
	Linear Storytelling Versus Interactive Storytelling	3
	How to Use This Book	4
<b>Chapter 1</b>	<b>The Nuts and Bolts: Getting Started with Twine</b>	<b>5</b>
	Installing Twine	5
	Starting Your First Story	6
	Saving in Twine	9
	Navigating Twine Menus	9
	Getting to Know the Stories Screen	9
	Getting to Know the Blue Grid Screen	9
	Getting to Know the Hover Menu	10
<b>Chapter 2</b>	<b>Using Choice to Create Agency</b>	<b>13</b>
	Designing Agency	14
	Providing Clues	17
	Crafting Meaning	18
	World-Building with Optional Details	19
	Using Choice to Affect Pacing	21
	Imagining Different Choices	22
	Giving Preference Choices	22
	Giving Value Choices	22
	Giving Adventure Choices	22
	Giving Ethical Choices	22
	Giving Cause/Effect Choices	22
	Giving Exploratory Choices	23
	Giving Obtainable Choices	23
	Giving Directional Choices	23
	Plotting Your Story's Choices	23

	Handling Endings	25
	Using the Fan Story Shape	25
	Using the Hourglass Story Shape	25
	Delaying Branching	27
	Creating Another Type of External Link	27
<b>Chapter 3</b>	<b>Creating a Vivid Setting</b>	<b>29</b>
	Understanding the Importance of Setting	29
	Creating Setting from Prompts	30
	Adding Descriptions with (link:)	31
	Repeating Text with (display:)	32
	Setting Tips and Exercises	35
	Writing What You Know	35
	Being the Player's Eyes	36
	Considering What a Character Would Notice	36
	Combining Motion with Surroundings	37
	Using Your Other Senses	38
	Considering the Mood of a Place	38
	Drawing the Player's Attention	40
	Using Descriptive Words	40
	Distinguishing Static and Dynamic Settings	42
	Building a Map-Based Game	43
	Creating a Maze	46
<b>Chapter 4</b>	<b>Designing Puzzles</b>	<b>49</b>
	Finding Puzzle Ideas	50
	Finding Puzzles in Everyday Life	50
	Making Your Puzzles Matter	51
	Building Puzzles in Twine	52
	Finding the Starting Point and End Goal	52
	Adding Layers to Your Puzzle	54
	Finishing the Puzzle	58
	Building Other Types of Puzzles	58
	Using Macros and Hooks	59
	Using Named Hooks	62

<b>Chapter 5</b>	<b>Building Objects with Variables</b>	<b>67</b>
	Using Objects in Interactive Fiction	67
	Building Interactive Objects with Variables	68
	Creating a Variable	68
	Changing a Value	69
	Using Strings	70
	Using Boolean Values	70
	Using Variables in Passages	71
	Building Conditional Statements	77
	Adding a Conditional Statement	78
	Adding More Than One Condition	81
	Keeping Track of the Player	83
	Setting Up Types of Quests	86
	Writing Search Quests	86
	Writing Fetch Quests	87
	Writing Drop-off Quests	87
	Writing End Quests	87
	Writing Rescue Quests	88
	Writing Escape Quests	88
	Writing Transformative Quests	89
	Writing Creation Quests	89
<b>Chapter 6</b>	<b>Stasis, Catalyst, and Climax: Understanding Story Arc</b>	<b>91</b>
	Drawing a Story Arc	91
	Understanding Time and Pacing	93
	Making a Game with Story Arc	94
	Opening a Game in Stasis	94
	Continuing to the Catalyst	96
	Building to the Climax	97
	Returning Back to Stasis	99
	Diagramming the Story	100
	Extending Story Arc	102
	Counting Turns with the <code>(history:)</code> Macro	103
	Controlling Time with <code>(history:)</code>	104
	Controlling Links with <code>(history:)</code>	105

<b>Chapter 7</b>	<b>Exploring Interactive Fiction Genres</b>	<b>107</b>
	Managing Reader Expectations	108
	Creating Chance with the (either:) and (random:) Macros	109
	Using the (either:) Macro	109
	Using the (random:) Macro	113
	Delving into Genres	115
	Exploring Horror	115
	Exploring Fantasy	116
	Exploring Science Fiction	117
	Exploring Mystery	118
	Exploring Espionage	118
	Exploring Adventure	119
	Exploring Magical Realism	120
	Exploring Historical	120
	Exploring Western	121
	Exploring Nautical	123
	Exploring Fairytale and Folklore	123
	Exploring Mythology	124
	Exploring Realistic Fiction	125
	Exploring Other Genres	125
	Understanding Motivations	126
<b>Chapter 8</b>	<b>Constructing Believable Characters</b>	<b>127</b>
	Recognizing the Traits of Successful Characters	127
	Using the (click:), (mouseover:), and (prompt:) Macros	129
	Using the (click:) Macro	129
	Using the (mouseover:) Macro	132
	Using the (prompt:) Macro	137
	Distinguishing Types of Characters	138
	Limiting Characters	139
	Creating a Clear Set of Traits	139
	Setting a Goal	142
	Being Vulnerable	143



	<b>Deepening Character Building</b>	<b>144</b>
	Assigning Purpose	144
	Providing Exposition	145
	Exploring Relationships	145
<b>Chapter 9</b>	<b>Maintaining an Inventory</b>	<b>147</b>
	Managing Inventory with Arrays	148
	Using Variables to Work Around Arrays	151
	Using the <code>(array:)</code> Macro	154
	Building a Basic Array	154
	Adding Items to an Array	156
	Dropping Items from an Array	158
	Checking Strings in an Array	159
	Checking the Array Size	161
	Limiting the Array Size	161
	Troubleshooting Arrays	162
	Using the <code>(count:)</code> Macro	164
	Utilizing Literary Techniques	165
	Using Symbolism	166
	Using Allegory	166
	Using Simile	167
	Using Metaphor	167
	Using Analogy	168
	Using Allusion	169
	Using Irony	169
	Using Hyperbole	170
<b>Chapter 10</b>	<b>Achieving Proper Pacing</b>	<b>171</b>
	Understanding Plot	172
	Setting the Pace with Language	177
	Determining Description Amount	177
	Employing Internal Thoughts	177
	Leaving Information Unknown	177
	Setting the Danger Level	177
	Varying Sentence Length	178
	Regulating Passage Length	178
	Limiting Word Count	178

Using Sentence Fragments	178
Moving with Vivid Verbs	179
Utilizing Setter Links	179
<b>Chapter 11 Foreshadowing Important Clues</b>	<b>183</b>
Using the (append:) Macro	184
Using the (prepend:) Macro	187
Foreshadowing Clues	189
Dropping Direct Clues	189
Dropping Subtle Clues	190
Dropping Dream Clues	190
Dropping Feeling Clues	190
Dropping Red Herrings	190
Creating a Hint System	191
Including Inside Jokes and Easter Eggs	194
<b>Chapter 12 Making a Role-Playing Game</b>	<b>195</b>
Using the (datamap:) Macro	196
Adding Values to the Datamap	197
Removing Values from a Datamap	198
Using a Datamap for a Stat System	198
Establishing Class	199
Becoming a Wizard	199
Becoming a Knight	199
Becoming a Thief	199
Becoming a Healer	199
Becoming an Elf	200
Becoming a Dwarf	200
Coding Class Selection	200
Establishing Names	202
Establishing Personal Characteristics	203
Choosing Gender, Height, and Weight	204
Troubleshooting User Play Decisions	204
Establishing Traits	206
Deciding on Traits	207
Determining Hit Points	207

Making the Next Character 208

Wrapping Up the Opening 210

Limiting an Array 211

## **Chapter 13 Combat and Consequences in Role-Playing Games 213**

Exploring the World 215

Using Personal Characteristics 215

Making Consequences 217

Using Traits 219

Using Items 220

Battling Monsters 223

Starting the Fight 224

Continuing the Fight 227

Solving Puzzles 229

Writing a Conclusion 231

Taking Your Role-Playing Game to the Next Level 231

Breaking Down Game Writing into Manageable Chunks 231

## **Chapter 14 Wiggling Words: Changing the Text Appearance 233**

Changing Fonts 234

Adding a New Font 234

Designing Fonts 235

Picking a Font 236

Changing Text Style 237

Adding Style 237

Changing Understanding 238

Changing Size 239

Changing Text Color 240

Changing Text Transitions 241

## **Chapter 15 Story Style: Changing the Game Appearance in Harlowe 243**

Navigating the Stylesheet 244

Setting Up Selectors and Declarations 245

Changing the Background 247

Changing the Sidebar 248

Changing the Font 249

Changing the Links 251

Getting Rid of Whitespace 252

**Chapter 16 Adding Images in a New Format 253**

Changing Over to SugarCube 254

Learning a New Macro Format 255

Deciding to Use Images 256

Knowing Which Image Files You Can Use 256

Enhancing Understanding with Images 257

Enhancing Player Mood with Images 257

Enhancing Game Play with Images 257

Adding Images to a Game 258

Sizing Images Before Uploading 258

Writing Image Code in a Passage 258

Using Images Creatively 260

Making Simple Links 260

Making Setter Links 262

Making Mouseover Clues 264

**Chapter 17 Setting Up the Pre-Story, Central Story, and Post-Story 267**

Learning New Macros 268

Using the `<<set>>`, `<<unset>>`, and `<<print>>` Macros 268Using the `<<if>>`, `<<else>>`, and `<<elseif>>` Macros 270Using the `<<nobr>>` and `<<silently>>` Macros 273

Designing the Pre-Story 275

Telling or Not Telling 276

Working the Backstory into the Game 277

Including Game Tutorials 278

Designing the Post-Story 279

Defining the Ending 279

Moving Beyond Happily Ever After 281

Making Sequels 282

**Chapter 18 Developing a Strong Player Character 283**Using the `<<button>>`, `<<radiobutton>>`, and `<<checkbox>>` Macros 284Learning the `<<button>>` Macro 284Learning the `<<radiobutton>>` Macro 285Learning the `<<checkbox>>` Macro 286

Using the <code>&lt;&lt;textarea&gt;&gt;</code> and <code>&lt;&lt;textbox&gt;&gt;</code> Macros	288
Learning the <code>&lt;&lt;textarea&gt;&gt;</code> Macro	289
Learning the <code>&lt;&lt;textbox&gt;&gt;</code> Macro	290
Using the <code>&lt;&lt;click&gt;&gt;</code> , <code>&lt;&lt;append&gt;&gt;</code> , <code>&lt;&lt;prepend&gt;&gt;</code> , <code>&lt;&lt;replace&gt;&gt;</code> , and <code>&lt;&lt;remove&gt;&gt;</code> Macros	291
Learning the <code>&lt;&lt;click&gt;&gt;</code> Macro	291
Learning the <code>&lt;&lt;append&gt;&gt;</code> and <code>&lt;&lt;prepend&gt;&gt;</code> Macros	292
Learning the <code>&lt;&lt;replace&gt;&gt;</code> and <code>&lt;&lt;remove&gt;&gt;</code> Macros	294
Working with Narrative Voice	296
Setting the Narrator	296
Being Consistent	297
Changing the Point of View	298
Playing for Research	299
Communicating with Non-Player Characters	299
Handling Dialogue	299
Using Twine Tricks for Conversation	301
Conveying Nonverbal Communication	302
Employing Juxtaposition and Contrast	303
Defining Goals and Assigning Purpose	303
Creating Blank or Defined Player Characters	303
Considering Gender	304
Slipping Into Character	304
<b>Chapter 19 Balancing Pacing and Action</b>	<b>307</b>
Using the <code>either()</code> and <code>random()</code> Functions	308
Learning the <code>either()</code> Function	309
Learning the <code>random()</code> Function	310
Using the <code>turns()</code> and <code>visited()</code> Functions	311
Learning the <code>turns()</code> Function	311
Learning the <code>visited()</code> Function	313
Using the <code>&lt;&lt;back&gt;&gt;</code> and <code>&lt;&lt;return&gt;&gt;</code> Macros	315
Learning the <code>&lt;&lt;back&gt;&gt;</code> Macro	315
Learning the <code>&lt;&lt;return&gt;&gt;</code> Macro	316
Using the <code>&lt;&lt;display&gt;&gt;</code> , <code>&lt;&lt;actions&gt;&gt;</code> , and <code>&lt;&lt;choice&gt;&gt;</code> Macros	317
Learning the <code>&lt;&lt;display&gt;&gt;</code> and <code>&lt;&lt;actions&gt;&gt;</code> Macros	317
Learning the <code>&lt;&lt;choice&gt;&gt;</code> Macro	319

<b>Saving Your Game</b>	<b>320</b>
Using the Save Feature	321
Retrieving a Saved Game	321
<b>Playing with the Pacing</b>	<b>322</b>
Speeding Up the Story	322
Slowing Down the Story	323
Striking a Pacing Balance	323
<b>Managing Expectations</b>	<b>324</b>
Highlighting Change	324
Using the Rule of Three	325
Distracting the Player with MacGuffins	325
<b>Chapter 20 Keeping Players Engaged</b>	<b>327</b>
<b>Returning to Arrays</b>	<b>328</b>
Building a New Array	328
Adding Items to an Array	329
Dropping Items from an Array	330
Checking Strings in an Array	332
Limiting the Array Size	333
Troubleshooting Arrays	334
<b>Checking Your Inventory</b>	<b>335</b>
Setting Up an Inventory Passage	335
<b>Understanding Special Passages</b>	<b>337</b>
Using the StoryMenu Passage	337
Using the StoryCaption Passage	338
<b>Knowing Your Audience</b>	<b>339</b>
Aiming for a Niche	339
Speaking Broadly	340
Writing for One Person	340
Deciding on Length	341
Piquing Interest with a Teaser	341
<b>Recognizing the Three Types of Players</b>	<b>342</b>
Making Games for Win Seekers	342
Making Games for Story Seekers	343
Making Games for Realism Seekers	343

Keeping Players Engaged	344
Having a Plan	344
Being Creative	344
Giving a Purpose	345
Rewarding Choices	345
Leaving Them Dreaming	345
<b>Chapter 21 Show, Don't Tell</b>	<b>347</b>
Learning About Other Formats	349
Adding New Formats in Twine	349
Utilizing the <code>time()</code> Function and <code>&lt;&lt;timed&gt;&gt;</code> and <code>&lt;&lt;goto&gt;&gt;</code> Macros	350
Counting Seconds	350
Building a Timer	352
Delayed Text	353
Practicing Showing, Not Telling	354
Problems with Telling	355
Describing Actions	356
Drawing Conclusions	357
Demonstrating Behavior	358
Transporting the Reader	359
Trusting the Reader	360
Telling, Not Showing	361
<b>Chapter 22 Story Style: Changing the Game Appearance in SugarCube</b>	<b>363</b>
Navigating the Stylesheet	363
Reviewing Selectors and Declarations	364
Changing the Background	367
Changing the Sidebar	368
Changing the Font	372
Changing Links	374
Creating a Class on the Stylesheet	376
Using HTML in a Passage	378
Bolding Words	378
Italicizing Words	379
Underlining Words	379
Resizing Words	379

Changing the Color of Words 379

Putting It All Together 380

## **Chapter 23 Finishing Up and Clicking Publish 381**

**Editing Techniques 381**

Tackling Different Types of Edits 382

*Editing for Structure* 382

*Editing for Details* 384

*Editing for Sound* 384

*Editing for Coding Errors* 384

*Editing for Mechanical and Grammatical Errors* 385

Navigating to the Proofing Copy 385

**Testing Your Game 386**

Questioning Beta Testers 386

Using Testing Notes 387

**Clicking Publish 388**

Hosting Your Project on Philome.la 389

Hosting Your Project on the Internet Fiction Database 389

Hosting Your Project on Your Own Site 391

Comparing the Options 392

**Reaching Your Audience 392**

Posting on IFDB.tads.org 392

Posting on Forums 392

**Joining the Gaming Community 393**

Participating in Jams 393

Participating in Competitions 394

Earning Interactive Fiction Awards 394

**Breaking the Rules 394**

**Index 397**



## Foreword

Some people set out to change the world when they build something new. I didn't. I built the first versions of what would become Twine with an audience of one in mind: myself. I had experimented with writing traditional interactive fiction (IF), the kind exemplified by games like *Zork*, that ask the reader to explore a world by typing commands. I wanted to try experimenting with IF in a different way, to create something that ran lighter on puzzle solving and heavier on storytelling.

The best reference point I had was an immensely popular series of books I had devoured when I was in elementary school, *Choose Your Own Adventure (CYOA)*. Though many of the stories they told were haphazard and madcap, there was something inescapably intriguing to me about stories that have no fixed plot, whose content is changed by the process of reading them.

Of course, this idea—which we'd now think of as a branch of the hypertext medium—has a history that stretches back much further than the CYOA series. It runs back nearly a century, in fact. But just as I had no great ambitions when I first started with Twine, I also had no great knowledge of the medium. My thoughts were humble. I saw a hill and wondered what was past it. Twine was the walking-stick I built to help myself get there.

Twine has grown and grown since then, to my delight. What I'm most proud of is how many people have used Twine to write their first interactive story. Some of these people only know the medium of interactive fiction in the same terms I did, with examples like CYOA and *Zork*. Many more, I suspect, come to the medium completely fresh. Whoever you are, it's truly a privilege to be able to offer a first glimpse of this territory. I hope you'll fall in love just as I did.

But there are hazards to exploration. False trails and dead ends abound. A wise explorer draws on every resource available—and this book, though there are no maps printed in its pages nor any descriptions of flora or fauna, is an excellent guidebook to have at your side on your first foray.

—Chris Klimas, creator of Twine

## About the Author

**Melissa Ford** is the author of numerous works of fiction and nonfiction. She has been a huge interactive fiction fan since 1982, when her dad gave her a copy of *Zork* to help her become a better speller. She is the blogging and social media editor at *BlogHer*, a contributor at *GeekDad*, and the Twine mentor at her local computer club. She is also the author of the award-winning blog *Stirrup Queens*. She earned her MFA from University of Massachusetts-Amherst.

## Dedication

*For my Dad, who brought home Zork and said, “Play this.”*

## Acknowledgments

First and foremost, a huge thank you to the Que team for all of your hard work: Rick, Todd, Tonya, Laura, Sandra, Seth, Cindy, Mark, Michelle, Greg, and Kitty. A shout-out to Charlotte Kughen, who helped with the middle of the book, and an enormous separate thank you to Rick Kughen for believing in this project when I wrote him, “So I have this idea for a book...” and he brought me into the Que family. I have never had so much fun working on a book.

Thank you to James Floyd Kelly, who held my hand when I needed to jump and made sure I landed somewhere good. And an enormous thank you to Chris Klimas for jumping with me. I would not have done this without you.

And on that note, there literally wouldn't be a book without Twine itself. Thank you to Chris Klimas for making a program that even someone with no coding background can pick up within minutes. You've really leveled the playing field. Thank you to Leon Arnott and Thomas Michael Edwards for your work with the two formats in Twine, and to Dan Cox and Greyelf for answering questions when I was first learning my way around Twine.

I would have never found Twine if not for Frank Hunleth and Josh Westgard asking, “Hey, Mel, want to learn an interactive fiction language?” And a huge thank you to Dave Lebling for answering questions during early days to get the next generation of interactive fiction creators on their way.

Thank you to one very special teacher, Andrea Siska, who not only nurtured the best in my kids but brought out the inner programmer in me.

No book would ever get finished without the help of my siblings and their families—Randall, Morgan, Wendy, Jonathan, Olivia, and Penelope—as well as my parents. In fact, most aspects of life would never get finished without the help of my parents, so thank you for everything from making me learn how to spell so I could play interactive fiction games to giving me writing days. I love all of you so much.

Much love and one thousand thank yous to Josh, who gives me space to dream and supports every outlandish idea that pops into my head. I love you for always having my back, for letting me vent when I get stuck, and for giving me the energy to try again. And thank you, of course, to the twins, who made me see computers as more than plastic and wire. There is no chance in a million years that I would have found Twine if not for you two, and the biggest happiness in my day is seeing the projects you create and the creativity that spills out of your brains. I love you two to pieces.

And, last but never least, Truman, my furry confidante who squeaks his many thoughts, especially ones about cookies.

## We Want to Hear from You!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

We welcome your comments. You can email or write to let us know what you did or didn't like about this book—as well as what we can do to make our books better.

*Please note that we cannot help you with technical problems related to the topic of this book.*

When you write, please be sure to include this book's title and author as well as your name and email address. We will carefully review your comments and share them with the author and editors who worked on the book.

Email: [feedback@quepublishing.com](mailto:feedback@quepublishing.com)

Mail: Que Publishing  
ATTN: Reader Feedback  
800 East 96th Street  
Indianapolis, IN 46240 USA

## Reader Services

Register your copy of *Writing Interactive Fiction with Twine* at [quepublishing.com](http://quepublishing.com) for convenient access to downloads, updates, and corrections as they become available. To start the registration process, go to [quepublishing.com/register](http://quepublishing.com/register) and log in or create an account\*. Enter the product ISBN, 9780789756640, and click Submit. Once the process is complete, you will find any available bonus content under Registered Products.

\*Be sure to check the box that you would like to hear from us to receive exclusive discounts on future editions of this product.

*This page intentionally left blank*

# Introduction

## Introduction to Interactive Fiction

Have you ever loved a book so much that you wanted to step inside it? Do you sometimes wish that you could walk around Narnia or become a student at Hogwarts? Do you think about alternate paths for the narrative or wish you could see the choices not taken for characters? What if all the bells and whistles were stripped away from a game, and all you were left with was the perfect, engaging story? What if I told you that there's a type of computer game that allows the player to enter and control the story, like in a Choose Your Own Adventure book?

It's called interactive fiction.

In this book you'll learn how to make your own interactive fiction using Twine.

## Wait a Second...What Is a Choose Your Own Adventure Book?

Choose Your Own Adventure books are unique novels that allow the reader to guide the story. After each page or two of text, you face a choice.

Let's pretend you're reading a Choose Your Own Adventure mystery set in a scary old house. After reading a few pages about the house, you get a choice: If you want to go in the kitchen, go to page 7. If you want to go to the living room, go to page 12.

If you turn to page 7, the story goes in one direction. If you turn to page 12, it goes in another. Every few pages, you make another choice. You can get 40 or so different story experiences inside a single book.

Of course, Twine doesn't operate with page numbers, but you can create a story with hyperlinked text that allows the reader to play over and over again, getting a different story outcome based on different choices.

That's just the jumping off point for Twine. Unlike with a Choose Your Own Adventure book, with Twine you're not limited to making a game where the player simply makes choices until reaching the end. By adding scripts, which are bits of programming used to run tasks within the game, you can have the player collect objects in an underground kingdom, race a timer to fix a space ship before it explodes, or have words wash off the screen in a story set on a beach.

Twine projects at their core are text based, but you can also add images or use the CSS and HTML code that you'll learn in this book to change the way the words appear on the screen.

Even if you've never written a line of code in your life, by the time you get to the end of this book, you will know how to build a text-based world, bring characters to life, and drop the player into a puzzle of your own creation. Sound good?

## Are They Games or Stories?

So are the projects made with Twine games or stories? Answer: Yes! Why do they have to be one or the other? With Twine, you get to play with your words.

In this book, I switch between the words *game* and *story*, but both words always apply. Some people like to think of using Twine as writing interactive stories and other people structure their projects more like games.

There are a lot of similarities between game structure and story structure. The same traits that make a good game also make a good book. Think about your favorite book and what you like about it. I'm willing to bet that it has interesting characters who have goals, action that moves the story along, and a plotline that makes sense. (For instance, characters learn there are problems first and find the solution later rather than the other way around.) In this book you'll learn how to use story structure in your Twine game, too.

I use a lot of writing terms in this book, and I promise that playing with Twine will make you a better writer, a better reader, a more logical thinker, and a generally more inquisitive and thoughtful person in the analog world.

I know, big promises, but I plan to deliver.

## What Type of Games Can I Make with Twine?

There's a whole wide world of interactive fiction out there, and it's divided into two main categories: choice-based games and parser games.

*Choice-based games* are digital versions of Choose Your Own Adventure books. The player reads some text and then clicks on a link to get the next piece of the story.

*Parser games* are another type of text-based adventure. Instead of clicking on a choice, the user types at a command line to tell the character what to do, such as "pick up the sword" or "go north." Lateral thinking is a hallmark of parser games, and many people find them difficult to play because they require the reader to solve puzzles by telling the character what to do in the game. If you can't guess the solution to the puzzle or find the right words, you can't move the story forward.

Twine straddles these two categories, offering the best of both worlds. You can make a straightforward Choose Your Own Adventure story, and you can also borrow elements of traditional parser games and incorporate them into a Twine project. Twine allows you to

give the player the freedom and complexity of a parser game even though all of the answers are technically right there on the screen, simultaneously giving it the simplicity of a choice-based game.

By adding scripting to Twine, you can have different text appear to the reader based on objects they're holding or tasks they've performed. You can create random events, such as the rolling of a virtual set of dice, which means you can make a complex role-playing game similar to *Dungeons and Dragons*, or you can make a scene where two wizards duel one another and have the game keep track of gained and lost points. You can build task puzzles, which are a hallmark of parser games, or you can create mazes. The type of game or story you make is limited only by your imagination.

## Linear Storytelling Versus Interactive Storytelling

I'm going to guess that you're pretty familiar with linear stories, in which you open the book on the first page and read straight through until you reach the last page.

You're probably also familiar with the first person and third person points of view. A first-person story uses pronouns such as *I* or *me*, and the reader understands the story through the main character's point of view. *The Hunger Games* is written in first person, with the story flowing through Katniss. If she doesn't experience it, the reader doesn't experience it. The book begins "When I wake up, the other side of the bed is cold." The pronoun *I* indicates first person.

Third person uses pronouns such as *he*, *she*, and *they*. The narrator floats above the story, usually following the main character but like an invisible ghost. Harry Potter is written in third person; the invisible narrator usually tells us what is happening in Harry's world, though every once in a while, we're following someone else who connects with Harry's story, such as Professor Snape. That series begins "Mr. and Mrs. Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much." The pronoun *they* indicates third person.

Unlike most linear stories, interactive stories are written in second person. By using the second-person pronoun *you*, an interactive story makes the reader feel as if he or she has been dropped into the story.

Look at how the first page of my favorite Choose Your Own Adventure book, *The Mystery of Chimney Rock*, begins: "Vacation is here, and you're visiting your cousins Michael and Jane for a few days at their new home in Connecticut." The story says you're visiting your cousins, so the main character is you.

When you construct a Twine story, you write it in second person so that the player becomes the main character in your story.



## How to Use This Book

Unlike an interactive story where you jump around the book, this one is a linear tale: You start here with the introduction and keep reading until you get to the last chapter.

But wait! This book is also interactive, in the sense that many times in each chapter, you'll set the book down and open up Twine on your computer and try writing scripts. These exercises come in the form of Try It Out sidebars, like this:

### TRY IT OUT: HERE'S WHAT A TRY IT OUT LOOKS LIKE

When you see this element in this book, you'll know you've reached a point at which you can try creating your own interactive fiction code. You'll receive a story prompt to get you started, though feel free to change the details to suit your literary tastes. The point is to get comfortable with the cool tasks you can do in Twine while simultaneously strengthening your writing abilities.

Make sure you do all the Try It Out exercises in the book because doing them will help you understand other ways you can creatively use scripting to construct unique aspects to your games.

I want to warn you now that there is a lot of information in this book, and if you try to perfect every aspect of storytelling in one fell swoop, you will quickly become overwhelmed. Authors and game makers spend many years trying to get all these aspects of storytelling to line up. My advice is to focus on getting comfortable with each facet of storytelling individually rather than trying to master everything at once. Once you're strong at creating memorable characters or building vivid settings, you will find that these skills are easy to pluck from your virtual toolbox each time you sit down to start a new project. Writing skills get under your skin the more you practice.

Ready to start making your first story? Turn the page.

## Creating a Vivid Setting

One of the first interactive fiction games, *Adventure*, explored a real space: Bedquilt Cave in Kentucky. The original author, Will Crowther, loved caving and wanted to re-create the sensation of cave exploration as a text-based game. So, he mapped out a real cave and coded the game, and he threw in some fantasy elements like an annoying dwarf and the magical word XYZZY. Later on, Don Woods expanded the game, giving gaming adventurers hours of play.

Players felt as if they were really crawling through a cave. It didn't matter if they had never been spelunking; Crowther acted as the player's eyes and ears. He led them through the subterranean scenery so they could imagine what it was like to crawl through narrow rock passages. I've played that game so many times that I think if I were ever to go to the real Bedquilt Cave, I could find my way around it just because of this game.

It's time to turn your eye toward *setting*, which refers to the physical space of a story. By writing a vivid description of a place, you help your players feel as if they're being transported there. Stories and games are magical because they make you feel as if you are traveling without leaving your home. This chapter teaches you how to write a strong setting and how to organize map-based games. Once you learn how to build a space out of words, you can forge off the beaten path and chart your own unique trail; maybe you'll even create your own land or planet.

## Understanding the Importance of Setting

Have you ever woken up somewhere unfamiliar and needed to take a few seconds to figure out where you are? Well, readers go through that sensation every time they start a Twine game.

The setting helps players figure out what's happening and helps them guess what will happen next. Location sets readers' expectations. For example, a story set on a distant planet is going to be very different from one set in a cave, or a futuristic restaurant, or a gloomy English manor.

You can use players' expectations to your advantage: You can give them exactly what they expect, or you can twist their expectations to surprise them. (This is called *subverting* expectations.)

Setting is the first thing readers encounter, so you have to make it good, and that means making it vivid. You need to make readers feel as if they're standing in a new place even though they're still

sitting in their living room. Think about your favorite settings: What makes you want to enter Narnia, or go to Hogwarts, or visit Mars?

A simple set of prompts can help you get started imagining a place and help you figure out what information you need to convey to your reader.

## Creating Setting from Prompts

These are some questions to ask yourself when you sit down to start writing a story or game:

**What's the location?** This is the general place or places where the game is set, such as New York City, Narnia, the beach, or the moon.

**Where does the player character start?** This is the specific place the game begins, such as the lamppost in Narnia or the front door of a house.

**Where does the game end?** Once you've decided the specific location where the game ends, you can focus on what motivates the characters in your story. Why do they want to get to that endpoint? More importantly, how do they get to that endpoint?

**What buildings are there?** What human-made structures are there? Which buildings are important to your story? Consider monuments, parks, and stores—anything built or shaped by humans.

**What does the geography look like?** Are there mountains that will become obstacles for the player? Water and coral reefs around the submarine? Giant canyons on the surface of the planet?

**What time is it?** Is your story set in the middle of winter? Is it near a holiday? Is your story taking place during the day, or at night, or both?

**What is the weather like?** Think about the climate and weather of your setting, which will impact how the player moves through the story. Is it a rainy place? Is it very hot? What do players need to wear to be comfortable?

**What are the other people like?** Maybe there aren't any other inhabitants in your story (creepy!), but if there are, what are they like? Are they helpful? Distrustful? Do they speak the same language as the player character or a different language? Of course, aliens count in this category, too.

**What is the culture of the area?** How does your player character relate to the culture of the area? Is the player character from that area and accepted by others? Is he or she an outsider from an area at war with the people in your setting?

**What is the area's past?** Think about the place's history. If your story is set on a distant planet, have other outsiders passed through there, changing the way the aliens think about outsiders?

Now that your imagination is percolating, I'll walk through two new Twine tools and how you might use them to create a space: `(link:)` and `(display:)`.

## Adding Descriptions with `(link:)`

You've already learned how to use a little scripting, though you may not have realized that your foray into coding a game has already begun. The `[[link]]` tool that you've been using to connect two passages, `[[hyperlinked words|Passage Name]]`, is a kind of script. Those double brackets around the words tell the program to link the current passage to the passage named on the right side of the vertical bar.

Now, however, you need to learn some scripting that involves using parentheses to set a tool apart from the plain text in the passage. The `(link:)` tool allows an action to occur inside the existing passage; namely, it allows you to give the reader small, optional details to the story without troubling the reader to enter a whole new passage.

With the `(link:)` tool, the text still appears as hypertext on the screen, and players instinctively know to click the links in order to explore where they go. But unlike the `[[link]]` tool, the `(link:)` tool keeps the player in the same passage (and, therefore, all the other text remains on the screen, too) and expands the paragraph to include the new text.

To create an internal link, write the link's text inside the quotation marks inside the parentheses and write the words you want to appear when the link is clicked inside single square brackets, like this:

```
(link: "Words you want linked.") [New words that appear and replace the linked text.]
```

You'll notice that the order of the hypertext word or words and the passage name mirrors the same order seen in the `[[link]]` tool, with the hypertext listed first and the name of the passage to the right.

What if you want quotation marks to appear with the link? For example, what if you want the linked text to be part of a conversation between two characters? For this, you put single quotation marks around the double quotes, as shown here:

```
(link: "'What the character is saying.'") [New words that appear and replace the linked piece of dialogue.]
```

Now the quotation marks remain as quotation marks when they appear onscreen. Try both of the preceding examples on your computer and click Play to see the `(link:)` macro in action.

Next try the following example. Open a new story in Twine and call it Link and type the following:

```
Every wall of the room is covered in ceiling-to-floor bookcases filled with old, dusty books. (link: "You pull an untitled book off the shelf.") [You stare at the strange symbols that dot the spine of many of the books, and feel a breeze move through the room even though all the windows are closed. The book you have pulled from the bookcase feels heavier than a normal book.]
```

When you play this example, you should see two sentences, one in plain text and one hyperlinked, as shown in Figure 3.1.

Every wall of the room is covered in ceiling-to-floor bookcases filled with old, dusty books. **You pull an untitled book off the shelf.**

**FIGURE 3.1** Example text using the `(link:)` tool.

When you click the hyperlinked sentence, it's replaced by the text you put in brackets, as shown in Figure 3.2.

Every wall of the room is covered in ceiling-to-floor bookcases filled with old, dusty books. You stare at the strange symbols that dot the spine of many of the books, and feel a breeze move through the room even though all the windows are closed. The book you have pulled from the bookcase feels heavier than a normal book.

**FIGURE 3.2** Once the linked text is clicked, the new text appears.

Notice that the hyperlinked text disappears from the screen when you click it. If you want any hyperlinked text to stay on the screen, you need to repeat it inside the brackets.

The small but mighty `(link:)` tool can help you build your setting by placing the power in players' hands. They can get as much description or as little description as they wish. When using the `(link:)` tool, remember to additionally use the `[[link]]` tool to continue the story and move players out of the current passage.

## Repeating Text with `(display:)`

Sometimes you might want to repeat the description of a complicated space, especially if you want the player to visualize the layout and have access to the same choices several times. The `(display:)` tool gives you an easy way to repeat text. It's perfect for creating small spaces in a story, such as looking at a room in detail or describing the layout of a town square.

In fact, Anna Anthropy's Twine game called *Town* does just that. You start out in a plaza and see a palace, an armory, and a bank. Each time you click on one of the links, you get a little bit of information about that building but still see the description of the plaza. You feel as if you're turning to examine each building while staying in the same space, since the overall description of the setting doesn't change. Of course, there is a link to the next part of the story in the description of one of the buildings, so you can continue once you've explored the small area as much as you like.

To create using this tool, add this line to any passages where you want to repeat the text of another passage:

```
(display: "Name of Passage")
```

Get ready to work through an example that shows how you might use the `(display:)` tool. Open a new story in Twine and call it Alien Restaurant. Change the title of the first passage from Untitled Passage to Restaurant and type this in it:

```
Every table is occupied by [[squid-like aliens|Aliens]] dining in pairs.  
There is an aquarium filled with [[neon-blue fish|Aquarium]] dividing the  
restaurant. The only empty table is set with the expected [[plate|Plate]],  
[[silverware|Silverware]], and [[napkin|Napkin]], but also has a  
[[strange box|Meal Box]] with dozens of knobs sticking out of the top,  
a [[bowl|Meal Bowl]] filled with tiny metal balls, and a [[stick|Meal Stick]]  
with a suction cup on the end.
```

This is the passage that you repeat when you use the `(display:)` tool. Every time you add `(display: "Restaurant")` to future passages, this entire passage repeats on the screen, as shown in Figure 3.3.

Every table is occupied by **squid-like aliens** dining in pairs. There is an aquarium filled with **neon-blue fish** dividing the restaurant. The only empty table is set with the expected **plate**, **silverware**, and **napkin**, but also has a **strange box** with dozens of knobs sticking out of the top, a **bowl** filled with tiny metal balls, and a **stick** with a suction cup on the end.

**FIGURE 3.3** This passage repeats every time you use the `(display:)` tool in a future passage.

You want players to be able to click on any of the links in the passage to receive more information about the squid-like aliens, the neon-blue fish, or the strange box, but you also want players to be able to make second, third, and fourth choices instead of moving into another section of the story. In other words, you want them to stay right here in this restaurant and explore.

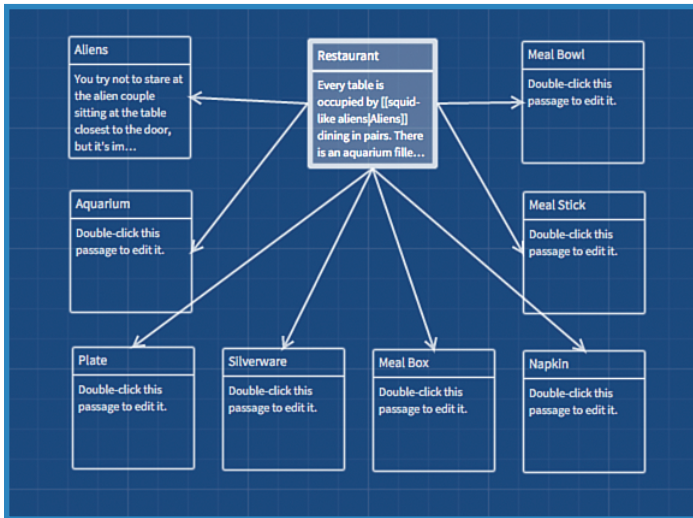
To make this happen, you need to start filling the newly created passages that describe parts of the restaurant. You're going to keep the description simple and then display the description of the restaurant again. The reason for keeping the description brief is that the Restaurant passage is already pretty long, and you don't want to give readers too much text to read through.

In the passage titled Aliens, add the following description and script:

You try not to stare at the alien couple sitting at the table closest to the door, but it's impossible not to gape at their waving tentacles. They sense you staring at them, and turn around to stare back at . . . YOU.

```
(display: "Restaurant")
```

You can see the additional passages fanning out around the Restaurant passage in Figure 3.4, even though only the Aliens passage is filled.



**FIGURE 3.4** The additional passages fanning out around the repeating Restaurant passage in the blue grid screen.

You need to play this story so far to make sure everything is working. Click the Play button and then click the text `squid-like aliens`. Your screen should look like the one shown in Figure 3.5.

Not only do players see the description of the squid-like aliens, but they can now choose to learn more about the silverware or that bowl filled with tiny metal balls, since you've repeated the Restaurant passage. This way, players can continue to explore the restaurant. Fill in the additional information in the other passages and make sure to always end each passage with `(display: "Restaurant")` so the player can keep exploring.

Be sure the capitalization of the title and the capitalization of the room name inside the display tool match. For instance, because the passage title is Restaurant, you need to capitalize *restaurant* when you write the name of the passage with `display:`, like this:

```
(display: "Restaurant").
```



You try not to stare at the alien couple sitting at the table closest to the door, but it's impossible not to gape at their waving tentacles. They sense you staring at them, and turn around to stare back at... YOU.

Every table is occupied by **squid-like aliens** dining in pairs. There is an aquarium filled with **neon-blue fish** dividing the restaurant. The only empty table is set with the expected **plate, silverware, and napkin**, but also has a **strange box** with dozens of knobs sticking out of the top, a **bowl** filled with tiny metal balls, and a **stick** with a suction cup on the end.

**FIGURE 3.5** The passage shows new text in addition to all the text and links in the Restaurant passage.

Remember that Twine is case-sensitive, and if the capitalization doesn't match, Twine gives you an error message telling you that a passage doesn't exist. Your first stop whenever you receive an error message should be to ensure that the capitalization in links matches the capitalization in a passage title.

## Setting Tips and Exercises

It's time to dive deeper into how location influences readers' experiences. The following exercises are meant to spark your creativity and to encourage you to look at setting in a whole new way. All the exercises use `(link:)`, `(display:)`, and `[[link]]`, or a combination of those tools. (If you need a refresher on how to use `[[link]]`, refer to Chapter 2, "Using Choice to Create Agency.")

### Writing What You Know

Writers are always told to write what they know. If you took that advice literally, you could never write fantasy stories or science fiction. Books would have no unicorns or dragons or magical schools. Fiction would be a little dull.

Still, it does make sense to draw from personal experience, even when you're trying to write about imaginary things. Maybe you've never been inside a real castle, but you've been inside a house. Use the layout of a house you know well, change the walls to worn stone and the grass outside the front door into a moat, and you've got yourself a castle!

If you want to write about something that you don't have any firsthand experience with, do a little research. Maybe you won't be able to go up in space, but you can certainly read firsthand accounts so you can accurately describe what liftoff feels like.



## TRY IT OUT: BUILD A CASTLE

To turn your home into a castle, open a new story and name it Castle.

Pretend your house or apartment is a castle. Using the layout of your actual home, create rooms in the house and change the description so it sounds like what you imagine a person would see if he or she were standing in an old stone castle.

Ask someone in your home to play the game. After they've gotten a chance to explore, ask them if they recognize the space.

## Being the Player's Eyes

Here's a cool fact: There is a longstanding relationship between the interactive fiction and blind communities. Text adventures—unlike graphical games—don't require a lot of work to become accessible to all players. Because of this, many blind gamers gravitate toward interactive fiction. Keep this in mind when building your game because you, the writer, are always the player's eyes.

When you're building setting, you're describing a place for someone who can't see it. Imagine a very specific place—real or fictional—and then start describing it to yourself. How detailed can you be? Focus on each small unit of the space, describing it as fully as possible before moving on to the next section.

## TRY IT OUT: DETAILED DESCRIPTIONS

Open the Castle game you created earlier. Choose one of the rooms and open the passage. Describe it in close detail; try creating passages for at least eight objects in the room. Describe the walls, the floors, the ceiling, and any fixtures or furniture. What things are out of place in the room? Don't stop describing until you sense that the reader would see the room exactly as you see it in your head.

Use either the `(display:)` tool or the `(link:)` tool to accomplish this exercise.

## Considering What a Character Would Notice

Consider what your player character is like and how that might affect what he or she notices. A short player character isn't going to be able to see high places, for instance, whereas a tall player character may not be focused on looking at the floor. Take a walk around the room you're in right now, taking note of what you notice at your eye level. Now drop down to the floor and crawl through the room. Do you see how your description of the same place would change depending on your height?

Now factor in the player character's personality. Is the player character someone who is very neat and tidy? A neat character would probably comment on a messy space or notice that things are out of place. Is the player character someone who moves quickly through an area, or is the character nervous or scared? A fast-moving or nervous player character may not notice the tiny details while running through the space.

## TRY IT OUT: NEW POINTS OF VIEW

Open the Castle game you created earlier. Choose one of the eight objects in the room described in a passage and have that object magically shrink or enlarge the player character temporarily. Describe the room from this new vantage point and then have the player character return to normal size before the player character heads back to the main description of the room.

## Combining Motion with Surroundings

The world looks very different when you're walking slowly through an area than when you're seeing it from the window of a moving train. Creating settings that keep changing can help capture the reader's interest. People might get restless staying in one place for too long, so think about keeping your players moving.

It's easy to keep players moving; all you need to do is give your characters a reason to explore. Why do they need to get out of the spaceship or go into the cave? Why do they need to move from one room to another? Are they looking for clues, trying to escape another character, or merely exploring?

## TRY IT OUT: GAZING OUT THE WINDOW OF A TRAIN

Start a new story and name it Train.

Set your player character on a moving train. Where is the player character going? Is the player character excited to be traveling or dreading arriving at the destination? Have your player character look out the window. What does the player character notice in the passing landscape? Give the player character the magical ability to slow down time using a spell so he or she can take a closer look at the world outside.

Use the `(link:)` tool to reveal more information. For instance, if the player character notices a shed outside the train window, he or she can use a spell to slow down the train, click the linked text, and reveal the inner contents of the shed.

## Using Your Other Senses

You're not just the players' eyes. You're also their ears, nose, tongue, and hands. Setting isn't just about what you see; it's about what you hear, smell, taste, or feel in the setting. What are the sounds of the forest? What is the smell of the Irish countryside after a rainstorm? What does a strawberry taste like? Does the tabletop feel bumpy or smooth?

### TRY IT OUT: EXPLORING OTHER SENSES

Open the Train game you created earlier. Visit the dining car. What does the character see, hear, smell, taste, and feel? Make sure to utilize all five senses and use the `(link:)` tool to reveal more sensory information.

## Considering the Mood of a Place

Some places are quiet and creepy, especially at night. Other places feel lonely, like a desolate planet on the edge of the solar system. Other places are lush and relaxing, like a jungle landscape thinning out onto a pristine beach.

Think of your setting as an extra character. What is the personality of your setting? Is it a loud place, a quiet place, an isolating place, a crowded place? Is your society a utopia or a dystopia? Is it a formal place where characters are dressed up or a casual place where jeans rule? Is the architecture ornate or plain? Is it an unforgiving landscape with prickly, unwelcoming foliage, or is it a warm, embracing small town surrounded by farmland? Is the place exciting like an amusement park or relaxing like a library?

Just as people have personalities, places do, too. The personalities of the characters may clash or work well with the setting, and the setting may even help you to create your characters. Think about the type of people who are drawn to or repelled by your setting.

### TRY IT OUT: DESCRIBING MOOD

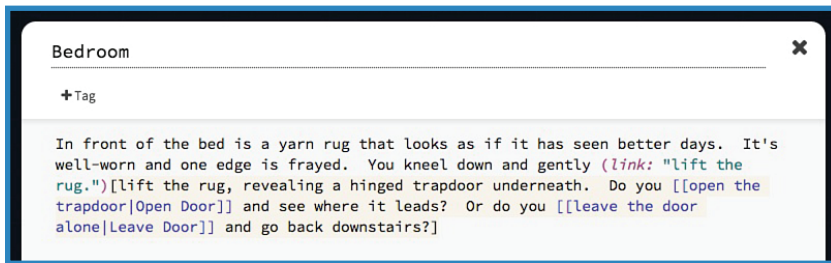
Open the Train game you created earlier. Leave the dining car and find yourself in an empty compartment. The emptiness fills you with dread. Where did all the people go? Why did the door suddenly lock behind you, trapping you inside the train car? Describe the creepiness of the space.

You can use a fun little trick to hide the exit of a room. Place it inside a `(link:)` by nesting the passage choice using this template:

```
(link: "hyperlinked words") [words to replace linked words plus
[[exit to another passage|Passage Name]]]
```

Notice that there are three square brackets at the end of the line. The first two close off the link to the next passage, and the last square bracket closes off the new words that replace the original linked words. As you type, you'll see the linked text change color to help you keep the brackets straight.

In Figure 3.6, you can see that the links to the passages that leads out of the room, `Open Door` and `Leave Door`, appear only when the player clicks on "lift the rug".



**FIGURE 3.6** The passage hides the links to other passages from the player inside the `(link:)` tool.

You could use this trick in your game to hide the exit by placing numerous links into the passage so the player needs to find the one that contains the exit out of the passage. Here's how you could create one where the exit is hidden inside a book left behind on a seat:

```
The train car is completely empty except for a (link: "book left behind on the
first seat.") [book that fans open as you pick it up, revealing a [[tiny door|Exit]]
drawn on the page.] There are four windows, two on the [[left side|Left Windows]]
of the train and two on the [[right|Right Windows]], though all of them appear to
be locked.
```

```
There is a light flickering from the ceiling with an (link: "odd hinged door in
the glass cover.") [odd hinged door that breaks off in your hand when you try
to open it.] All of the seats are covered in [[plush velvet|Velvet]]; strange
for an ordinary passenger train.
```

The additional links in the passage help the reader visualize the space, but only one of them contains a way out of the train car. The extra passages, such as `Left Windows`, `Right Windows`, and `Velvet`, are all detail passages that give the player a little bit more information about the space and bring the player back to the train compartment passage.

## Drawing the Player's Attention

Vivid description doesn't necessarily mean long description. In fact, most people don't like to receive the setting as one big chunk of text at the beginning of the story. Write so that different aspects of the setting appear as the player character moves through the story. When you start your story, your player doesn't need to know about every feature of the land. Instead, as the character moves around and encounters new locations, you can gradually reveal that there is a cave guarded by a dragon, that mountains blot out the sky, or that a town of fairy houses is nestled deep in the forest.

Balance barebones, practical descriptions of unimportant places with vibrant descriptions of important places. This helps draw the reader's attention to whatever it is you want them to notice.

Let's say that you're describing a town. You may write a long description of the important buildings where the action will take place (the library, school, and bakery) but merely mention the fact that there is also a post office and hospital since the story will never enter those spaces.

### TRY IT OUT: GUIDING THE PLAYER THROUGH THE TOWN SQUARE

Start a new story and name it Town Square.

Pretend your player character is an alien who has landed in a sleepy little town on Earth. The alien is standing in the town square, trying to figure out the landscape. Remember, the player character has never been to Earth and has no clue what he or she is looking at.

Using the `(display:)` tool, label a repeating passage Town Square. Then create links to the various buildings around the town square so that the alien player character can get a bit more information about each building. Give a deeper description for each important building and give a cursory description of each unimportant building.

Don't forget to include `(display: "Town Square")` at the bottom of each passage so the player can see all the options over and over again. Ask a friend to play your game and see if he or she can tell which are the important buildings.

## Using Descriptive Words

The more descriptive the words you use, the fewer of them you need to use. Write out a description of the setting and then judge each word. Ask yourself if there's a more specific word that says the same thing but goes a step further. For example, red is certainly a specific

color, but crimson or maroon goes a step further in helping the reader visualize the shade of red you have in mind.

A thesaurus can come in handy! Look up synonyms for any words you think you could replace with something better.

## TRY IT OUT: A VERY DESCRIPTIVE HAT STORE

Returning to the Town Square story you began earlier, you're going to create a few new buildings. The first building is the movie theater. Create a passage for the movie theater (titling it Movie Theater) and write a brief description of the space. Create a second space called Hat Store 1 and link it to the movie theater.

Now create a passage for a library (titling it Library) and write a brief description of the space. Create a space called Hat Store 2 and link it to the library.

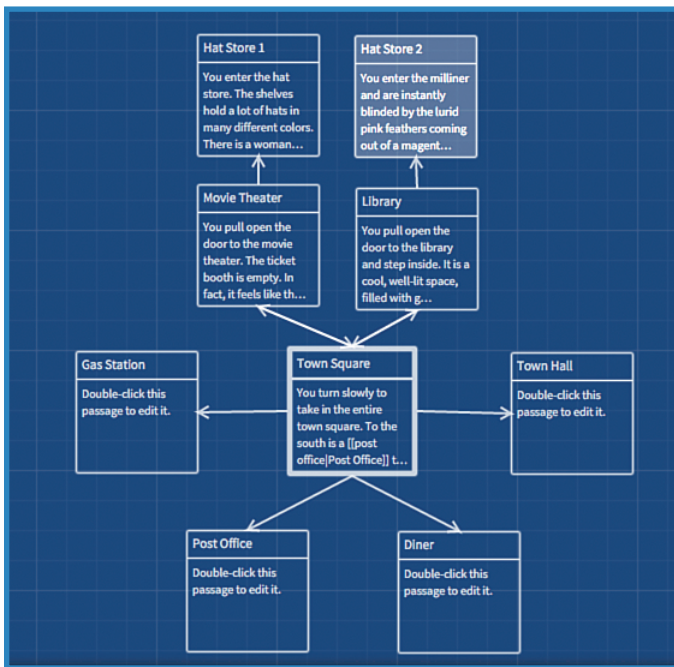
Now open the passage titled Hat Store 1. Write a basic description of the store, keeping it as simple as possible. Copy that description and also place it in the passage for Hat Store 2, but this time, edit that description to make it specific. For instance, you could say that it's a milliner, which is a store that specifically sells women's dress hats. See, the description has gone from general hats, which could include everything from top hats to baseball caps, to fancy women's hats, like fascinators. How specific can *you* become?

Which hat store description a player receives is dependent on whether the player enters the store through the movie theater or the library as seen in Figure 3.7. Pretend that the library in this town instantly fills your brain with great vocabulary words, just by walking through the doors.

If the player has been to the library, he or she will see a descriptive paragraph. If the player has been to the movie theater, he or she will see a basic paragraph.

For example, for Hat Store 1, you could write something simple like this:

You enter the hat store. The shelves hold a lot of hats in many different colors. There is a woman behind the counter who looks up from her book as you enter.



**FIGURE 3.7** The two Hat Store passages give readers two very different levels of description on the same space.

In Hat Store 2, you can kick that description up a notch by choosing very specific words: You enter the milliner and are instantly blinded by the lurid pink feathers coming out of a magenta, velvet cloche set on a shelf by the front door. The store bell tinkles like laughter as you enter, and the elderly woman behind the counter lifts one gnarled finger in the air as she continues to read her book, indicating that you should wait to speak until she is done with the page.

Which description gives you a very specific visual of the space?

## Distinguishing Static and Dynamic Settings

Sometimes the setting remains exactly the same throughout a whole story; this is known as a *static* setting. In other stories, the setting may change a lot, and this is known as a *dynamic* setting. Whether or not the setting changes plays a big role in the action of the story.

There are two ways a setting can change: An event may change the landscape, such as a fire burning down an important building, or the player character may move, traveling from place to place. Both options create a sense of movement, which keeps the story trucking along. Static landscapes are great for conveying drudgery, and landscapes that transform help move the action.

## TRY IT OUT: TRAVEL THROUGH TIME

Want one story with a lot of settings? Consider a time travel tale! Open up a new story and title it Time Travel. Start the story in a time machine. Where will you go? This passage will be repeated using the `(display:)` tool, so make sure that your wording works and that you've labeled the starting passage Time Machine.

Give the reader six possible destinations and repeat the code `(display: "Time Machine")` at the bottom of each of the six passages.

So how could you tackle this exercise? First, set up the passage you need to repeat and title it Time Machine:

```
There is a time machine here with six buttons, each a different color.  
Do you want to press the [[red button|Red]], [[orange button|Orange]],  
[[yellow button|Yellow]], [[green button|Green]], [[blue button|Blue]],  
or [[purple button|Purple]]?
```

Then, in the new passages that you just created, describe what happens when someone pushes one of those buttons. Make sure you add `(display: "Time Machine")` at the bottom of the paragraph so the player can still see the description of the time machine.

For instance, in the green button passage, I wrote a brief introduction that sends the player on his or her way or gives the player the option to choose a different button, using `(display: "Time Machine")`:

```
You press the green button and an image of the rolling fields of Ireland flash  
on the screen. A cool voice comes out of the speakers: "Would you like to  
[[visit Ireland|Ireland]] in the year 2045?"
```

```
You debate what to do. You have always wanted to visit [[Ireland|Ireland]].  
(display: "Time Machine")
```

Sketch out the diverse settings, write layers of description with the `(link:)` tool, and write short passages. How does the player character interact with these very different settings? Does the player character enjoy some settings more than others?

This is your longest, most complicated story yet, so take your time with this exercise before going onto the next section.

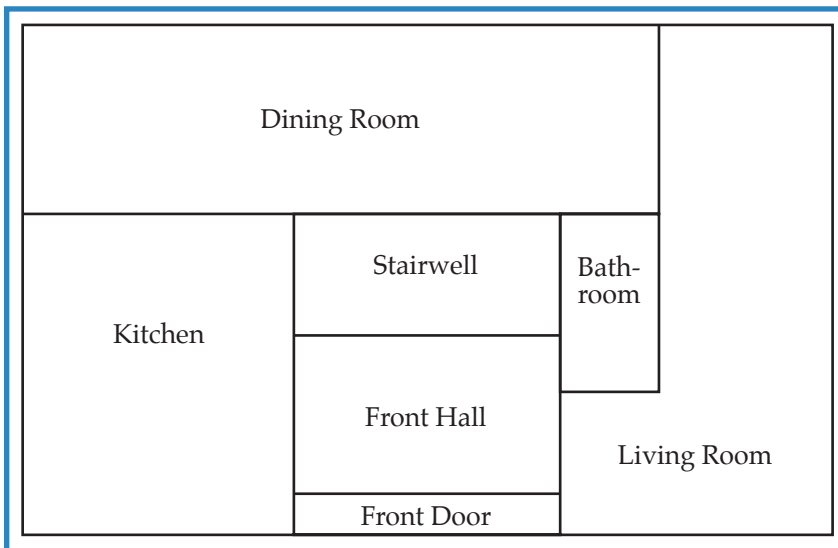
## Building a Map-Based Game

Creating a map-based game involves creating a physical setting that the player can explore. With directional choices like north, south, up, down, in, or out, the player controls where he or she goes. A map-based game may not have a plotline; rather, it may be fun because it allows a player to “travel” to a space and explore without leaving home. For example, you



could construct an interactive fiction Hogwarts and allow players to move from room to room in the castle.

You can practice this process by turning your home into a game space. Think of this as an unofficial but more elaborate Try It Out. Start by constructing a map of your house or apartment. Draw the layout of the rooms and pay attention to the flow of the space so you know how players can move from room to room. Figure 3.8 shows an example of a map of a house.



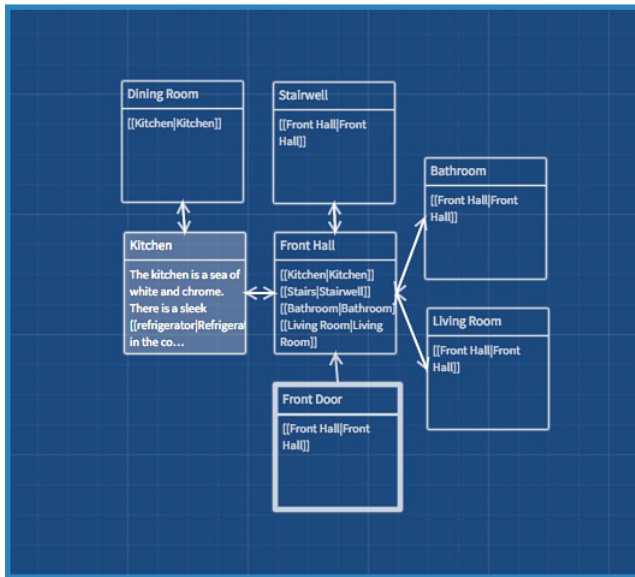
**FIGURE 3.8** A map of a house.

Once you have a rough sketch of your house, you can translate it into Twine. Create a passage for each room on your map. Figure 3.9 shows what the map from Figure 3.8 looks like all set up in Twine. Click and drag the boxes to arrange them so they're in the same order as the rooms on your map.

Link the rooms together as you would actually walk through them. In Figure 3.9, the player can only go to the dining room or front hall from the kitchen, but he or she can go to four different rooms from the front hall.

Now write some choices for your player to make when navigating this house. You can give the player directional choices such as go left or go right, or you can write room-based choices, such as “enter the living room” or “enter the bathroom.”

Open each passage and write a description of the room. Try actually walking to that room in your house and taking a look around. Where do your eyes go first when you enter that room? What furniture is in that room? Remember that you are not only the players' eyes, but also their ears, nose, tongue, and hands. Use all five senses when writing your descriptions.



**FIGURE 3.9** The earlier house map now constructed out of Twine boxes.

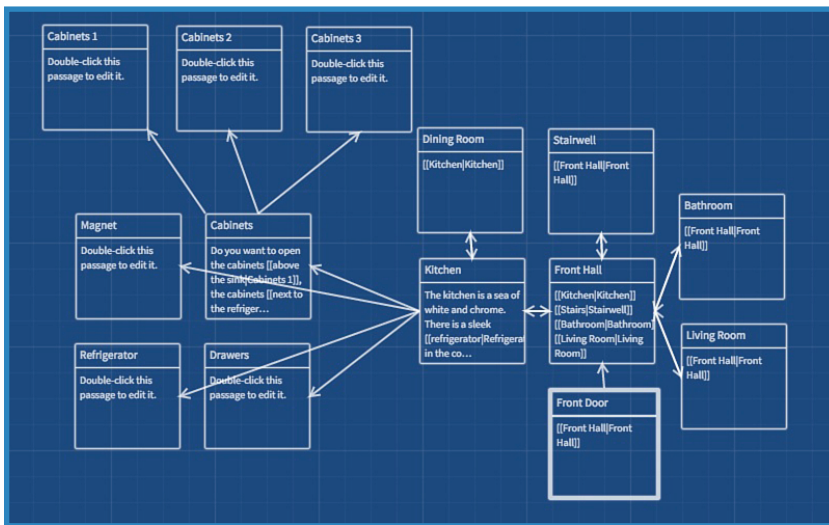
Now that you've written some description to start, you can add even more details to create opportunities for deeper exploration. Write a few short detail passages that connect to each room or use the `(link:)` tool to hide information for the player to discover.

Although there are only seven locations in this house, each passage may have dozens of detail passages that provide further detail. Each of those passages may have multiple passages connecting to them, as you can start to see in Figure 3.10.

In the kitchen, the player can look in the cabinets. In this case I've used detail passages to give additional information, but I could just as easily use the `(link:)` tool to create richer description.

Unless you're talking about places that people would love to visit, such as Hogwarts or the Shire, moving from room to room probably isn't very interesting if there isn't a goal, so you should give the player a goal. The player character could discover a ripped piece of paper with part of a secret message on it, and they must search the house to find the rest of the pieces. Don't make it too easy! Create nested links to make players poke around to accomplish the goal.

Houses are often small, but you can use the technique just described with larger spaces, such as a town or a country. Just draw your map, create your corresponding passages, arrange the boxes on the grid screen, and then add shorter passages to the main passages.



**FIGURE 3.10** The house contains several rooms. Each of those rooms has detail passages connected to the rooms to encourage deeper exploration.

## Creating a Maze

Mazes are a beloved and dreaded hallmark of interactive fiction. They have fallen out of vogue in contemporary interactive fiction because people find them fun to make but annoying to play. Still, they played a huge role in the early interactive fiction games, and, therefore, it's a rite of passage to make your own.

In fact, *Adventure*, that interactive fiction game I mentioned at the beginning of this chapter, contains a “maze of twisty little passages, all alike.” What makes mazes interesting is that the path through them isn't obvious, so players must pay attention and keep track of where they've been.

### TRY IT OUT: DESIGN YOUR OWN MAZE

Make a simple maze by arranging a grid of 16 boxes on the Twine screen, with 1 additional box to serve as the exit. Designate one of the passages as the entrance to the maze and then link various passages together. Try making some of your paths go diagonally (do you want to go northeast, or do you want to go southwest?) and throw in a few one-way options and dead ends.

Subtly change the text in the passages to convey that the reader is in a similar but slightly different space. A player wins when he or she gets to the exit passage, so make sure you have some fun text in that box to reward the player for making it through the maze.

Some other common spatial tropes in interactive fiction are caves and tunnels. It's good to know what commonly pops up in text adventure games so you can get creative and twist these elements into something new.

Map-based games are a great blank slate for exploring other aspects of Twine, including collecting objects or creating a points system for a traditional adventure-style role-playing game (RPG). As long as the exploration is meaningful and the players have a goal, they'll keep coming back to your game to play it again and again.

You've only scratched the surface of using Twine. You may be worried that the tasks are going to get harder, but I have a secret to tell you: Without even realizing it, you've been using macros, hooks, and strings. In other words, you've already done a lot of computer programming! I'll discuss those things further in the next chapter and teach you what macros are and how they work in Twine. Soon you'll be able to do some pretty cool things in your stories.

*This page intentionally left blank*

# Index

## Symbols

- 80 Days*, 49, 327
- + (add operator), 69
- / (divide operator), 69
- "" (empty quotation marks), 289
- > (greater than operator), 69
- >= (greater than or equals operator), 69
- < (less than operator), 69
- <= (less than or equals operator), 69
- \* (multiply operator), 69
- (subtract operator), 69

## A

### actions

- defined, 96
- describing, 356
- falling, 175

<<actions>> macro, 317-318

**Adam, Douglas**, 275

**add operator (+)**, 69

*Adventure*, 29

**adventure choices**, 22

**adventure genre**, 119

**agency**, 13-17

- choice frequency, 21
- clues, providing, 17-18
- detail passages, 19-21
- freedom, 14
- high versus low agency, 15-16
- link placement, 19
- making decisions matter, 14
- meaning, 18-19

**agreeableness (characters)**, 140

**Aikin, Jim**, 92

*Alice Falling*, 234

**alignment (text)**, 249

**allegories**, 166

**allusions**, 169

**analogies**, 168

**and operator**, 271

**anonymous hooks**, 62

**antagonists**, 138

**Anthropy, Anna**, 32

**appearance**. *See also* stylesheets

- backgrounds, 247-248
- fonts, 249-250
- links, 251-252
- sidebars, 248
- text

- color*, 240-241

- fonts*, 234-237

- size*, 239-240

- styles*, 237-238

- transitions*, 241-242

- whitespace, deleting, 252

<<append>> macro, 292-294

**(append:) macro**, 184-187

**archiving stories**, 9

**Arnott, Leon**, 59

**(array:) macro**, 148

- arrays

- building*, 154-156

- size*, 161-162

- troubleshooting, 162-164

- values, 156-161

**arrays**

- checking inside, 149

- conditional text, 150

- creating, 149-150, 154-156, 328

- defined, 148

## items

- adding*, 156-158, 329-330
- checking for*, 159-161
- counting*, 164-165
- deleting*, 158, 330-331

length, checking, 150

limiting, 211

size, 161-162, 333

strings, checking, 332

testing, 329

troubleshooting, 162-164, 334

variable work around, 151-153

**assignment macros**, 77

**audience**. *See* **players**

**awards**, 394

**B**

**background-color property**, 245

**backgrounds**, 245-248, 367

**<<back>> macro**, 315-316

**backstory**. *See* **pre-story**

**battle scenes**, 223

actions, 225-226

continuing fights, 227-229

player death, 226

starting fights, 224-227

**Bedquilt Cave (Kentucky)**, 29

**behavior, demonstrating**, 358-359

**believable characters**. *See* **characters**, **believable**

**beta testers, questioning**, 386-387

***Beware the Faerie Food You Eat***, 183

**Big 5 character traits**, 140

**blank characters, creating**, 303

**blue grid screen**, 6, 9

**bolding words**, 378

**Boolean values**, 70-71

**breaking rules**, 394-395

**browser compatibility (Twine)**, 5

**<b> tag**, 378

**<<button>> macro**, 284-285

**buttons**

creating, 284-285

multiple choice, 285-286

Play, 8

Save to Disk, 321

Saves, 321

**C**

**CANOE (conscientiousness, agreeableness, neuroticism, openness, extraversion/introversion)**, 140

**cardboard characters**, 126

**Cascading Style Sheets**. *See* **CSS**

**case sensitivity**, 35

**catalyst**, 91, 96-97

**cause-and-effect choices**, 22

**CC (Creative Commons) license**, 256

**chance, creating**, 109

numbers, 113-115

text, 109-113

**changes, highlighting**, 324-325

**characters**

agreeableness, 140

antagonists, 138

believable, 128-129

*displaying additional text when mousing over*

*chosen words*, 133

*exposition, providing*, 145

*goals*, 128, 142

*purpose, assigning*, 144-145

*relationships*, 145-146

*successful traits*, 127

*traits*, 139-141

*vulnerability*, 128

blank, creating, 303

cardboard, 126

charm levels, 207

classifications, 199-202

conscientiousness, 140

contrast, 303

defined, creating, 303

defining with objects, 67

dexterity levels, 207

dialogue, 299-302

exposition, providing, 145

extraversion, 140

foils, 138

gender, 304

- goals, 128
  - setting*, 142
  - telling players*, 210
- hit points, 207
- intelligence levels, 207
- introversion, 140
- juxtaposition, 303
- limiting, 139
- motivations, 126
- multiple player characters, 208-209
- neuroticism, 140
- non-human, creating, 140
- non-player, 14, 220-221
- nonverbal communication, 302
- openness, 140
- personal characteristics, 204-205, 215-217
- player. *See* player characters
- protagonists, 138
- purpose, 13, 144-145
- relationships, 145-146
- sidekicks, 139
- slipping into characters, 304
- stories told between, 278
- strength levels, 207
- traits
  - Big 5*, 140
  - creating*, 139-141
  - exploration replayability*, 219-220
  - selecting*, 207
  - setting*, 206
- types, 138-139
- vulnerability, 128, 143-144
- wandering non-player, creating, 220-221
- charm levels (characters), 207**
- check boxes, creating, 286-288**
- <<checkbox>> macro, 286-288**
- Chekhov, Anton, 13**
- choice-based games, 2**
- <<choice>> macro, 319-320**
- Choice of the Dragon*, 195**
- choice puzzles, 49**
- choices**
  - check boxes, creating, 286-288
  - choosing once, 319-320
  - clues, providing, 17-18
  - consequences, 18
  - creating, 7-8
  - dead ends for some, creating, 101
  - detail passages, 19-21
  - exploration
    - consequences*, 217-219
    - inventory-based*, 222
    - player character personal characteristics*, 215-217
    - replayability with character traits*, 219-220
  - fair, 17
  - four choices example, 19
  - frequency, 21
  - lighthearted, 18
  - link placement, 19
  - making decisions matter, 14
  - meaning, 18-19
  - outlining, 23-24
  - radio buttons, creating, 285-286
  - rewarding, 345
  - single, 19
  - types, 22-23
- Choose Your Own Adventure books, 1**
- classes, 364, 376-377**
- classifications (characters), 199-202**
- <<click>> macro, 291-292**
- (click:) macro, 62, 129-132**
- (click-append:) macro, 184**
- (click-prepend:) macro, 187**
- (click-replace:) macro, 131**
- clickable links, creating, 291-292**
- cliff-hangers, 177, 267, 322**
- climax, 91, 97-99**
- closed endings, 267**
- clues**
  - foreshadowing, 189-190
  - hint systems, creating, 191-194
  - mouseover images, creating, 264
  - providing, 17-18
- coding**
  - character classifications, 200-202
  - errors, troubleshooting, 384
  - spacing, 274
- collecting objects, 67**
  - organizing with variables, 151-153
  - realism, 148
  - storing in containers, 148



**collection puzzles, 49****color**

- background, 247-248
- hexadecimal code, 240
- links, 251, 374-375
- passage background, 245
- text, 209, 240-241, 379

**communities (gaming), 393-394****competitions, 394****completion endings, 281****conclusions, drawing, 231, 357****concrete endings, 280****conditional statements**

- adding, 78-80
- arrays, 150
- building, 77
- hooks, 79
- multiple, 81-83
- operators, 271
- player characters, tracking, 83-86
- structure, 78
- SugarCube, 270-272
- turns() function, 312
- visited() function, 313

**confidantes, 139****Conrady, Matthias, 234****conscientiousness (characters), 140****conscientiousness, agreeableness, neuroticism, openness, extraversion/introversion (CANOE), 140****consequences, 18, 217-219****containers (objects), 148****continuing fights, 227-229****contrast (characters), 303****(count:) macro, 164-165*****Counterfeit Monkey*, 243****creation quests, 89****Creative Commons (CC) license, 256****Creative Commons website, 257****crossroads passages, 216****Crowther, Will, 29****cryptogram puzzles, 59****CSS (Cascading Style Sheets), 244**

- accessing, 243
- backgrounds, 247-248, 367
- classes, creating, 376-377

- declarations, 245-247, 364-366

- fonts, 249-250, 372-374

- links, 251-252, 374-375

- navigating, 244-245

- selectors, 245-247, 364-366

- sidebars, 248, 368-371

- whitespace, 252

**cursive fonts, 236****D****Dalmady, Astrid, 183****danger level (pace), 177****data values**

- adding, 200-202
- consequences, creating, 218
- establishing, 199-200

**(datamap:) macro, 197-198****datamaps, 196**

- benefits, 196
- characters, 204-207
- data values, 197-202, 218
- function, 196
- Health stat, creating, 207
- player decisions, troubleshooting, 205
- stat systems, creating, 198-199

**dead ends for story choices, 101****decision trees, 23-24****declarations, 245-247, 364-366****decorative fonts, 236****defined characters, creating, 303****delayed branching, 27****delaying text, 353-354****deleting**

- items from arrays, 330-331
- line breaks, 273-274
- passages, 11
- text after clicking hypertext words, 295
- values
  - arrays*, 158
  - datamaps*, 198
  - SugarCube*, 269
- whitespace, 252

**descriptive words, 40-42****detail passages, 19-21, 384****dexterity levels (characters), 207**

**diagramming, 100-101, 175-176**

**dialogue**

- handling, 299-300
- narrative voice, 284
- tricks, 301-302

**dice rolling simulation, 113-115**

**direct clues, dropping, 189**

**directional choices, 23**

**<<display>> macro, 317-319**

**(display:) macro, 32-34**

**distracting players, 325-326**

**divide operator (/), 69**

**does not equal operator (is not), 69**

**dreams, 190, 278**

**drop-off quests, 87**

**dungeon crawl games, 147**

**dwarf data name value, 200**

**dynamic setting, 42-43**

## E

**Easter eggs, 194**

**Edwards, Ron, 342**

**Edwards, Thomas Michael, 254**

**either() function, 309-310**

**(either:) macro, 109-113**

**elements (HTML), 292**

**elf data name value, 200**

**<<else>> macro, 271**

**(else:) macro, 77-78, 312**

**<<elseif>> macro, 271-272**

**(elseif:) macro, 81-83, 314**

**emphasis (text styles), 238**

**empty quotation marks (“”), 289**

**Enchanter, 147**

**end goal (puzzles), 52-53**

**end quests, 87**

**<<endif>> macro, 271**

**endings**

- categories, 267
- completion, 281
- concrete, 280
- defining, 279-281
- epilogue, 280
- happy, 281
- interpreted, 280

mixed, 281

open, 280

sad, 281

story shapes, 25-27

surprise, 280

types, 281

ultimate, 280

understanding, 281

**engaging players, 344**

creativity, 344

giving players purpose, 345

leaving players dreaming, 345

planning, 344

rewards, 345

showing not telling, 354-360

telling not showing, 361

**entering words, 7**

**epilogue endings, 280**

**eq operator, 271**

**equality operator, 271**

**equals operator (is), 69**

**escape puzzles, 59**

**escape quests, 88**

**espionage genre, 118**

**ethical choices, 22**

**Eve, Eric, 92**

**expectations, 108, 324-326**

**exploration**

choices, 23

consequences, 217-219

inventory-based choices, 222

player character personal characteristics,  
215-217

replayability with character traits, 219-220

**exposition, 91, 145**

**external links, 27**

**extraversion (characters), 140**

## F

**fair choices, 17**

**fairytale genre, 123**

**falling action, 175**

**fan story shape, 25**

**fantasy genre, 116**

**feeling clues, 190**

**feelings, 238**

*Fellowship of the Ring*, 213

**fetch quests, 87**

**fiction, 125**

**fighths, 225**

continuing, 227-229

starting, 225-227

**final game statistics, printing, 231**

**first person point of view, 3, 283**

**flashbacks, 278**

**flat text, 165**

**Floyd from *Planetfall*, 127**

**fluctuating pacing, 308**

**foils, 138**

**folklore, 123**

**(font:) macro, 234**

**fonts**

categories, 235

changing, 234-235, 249-250, 372-374

choosing, 236-237

defined, 234

links, 252

**foreshadowing**

(append:) macro, 184-187

clues, 189-190

defined, 183

drawing attention to details, 184

hint systems, 191-194

misleading players, 184

(prepend:) macro, 187-188

tone, setting, 183

**formats**

macros, 60

SugarCube, 254-256, 349

Twine, 59, 349-350

**four choices example, 19**

**fragments (sentences), 178**

**freedom, 14**

## G

**game masters (GMs), 214**

**games**

choice-based, 2

competitions, 394

descriptions, writing, 391

dungeon crawl, 147

editing, 381-386

jams, 393

map-based, 43-45

parser, 2

publicizing, 392

publishing, 388-392

saving, 320-321

short stories, compared, 171-172

testing, 386-387

tutorials, 278

versus stories, 2

writing, 232

**gaming communities, 394**

**gamist, narrativist, simulationist (GNS), 342-343**

**garbage can icon (Hover menu), 11**

**gender (characters), 304**

**genres**

adventure, 119

defined, 115

espionage, 118

fairytale/folklore, 123

fantasy, 116

historical, 120

horror, 115-116

magical realism, 120

mystery, 118

mythology, 124

nautical, 123

realistic fiction, 125

science fiction, 117

subgenres, 115

Western, 121-122

**GMs (game masters), 214**

**GNS(gamist, narrativist, simulationist), 342-343**

**Google**

Developers image attributes tutorial  
website, 258

Fonts, 250, 373

**<<goto>> macro, 351-353**

**(goto:) macro, 98, 134-135, 180**

**gradient backgrounds, creating, 367**

**grammatical errors, troubleshooting, 385**

graphical interactive fiction, 253  
 greater than (gt) operator, 271  
 greater than operator (>), 69  
 greater than or equal operator (>=), 69  
 greater than or equal to (gte) operator, 271

## H

happy endings, 281  
 Harlowe format, 59  
 healer data name value, 199  
 health levels, 224-226  
 Health stat, creating, 207  
 hexadecimal code, 240  
 hidden object puzzles, 58, 63-64  
 high agency, 15-16  
 highlighting changes, 324-325  
 hint systems, 191-194  
 historical genre, 120  
 (history:) macro  
   passing time, 104-105  
   player characters, tracking, 84-86  
   turns, counting, 103  
   when links appear, 105-106  
 hit points, setting, 207  
*The Hitchhiker's Guide to the Galaxy*, 275  
 hooks  
   anonymous, 62  
   brackets, 60  
   conditional statements, 79  
   defined, 60  
   example, 60  
   named, 62-64, 129-132  
   puzzle example, 61  
 horror genre, 115-116  
 hosting  
   comparison, 392  
   Internet Fiction Database, 389-391  
   personal sites, 391  
   Philome.la, 389  
 hourglass story shape, 25-26  
 Hover menu, 10-11  
 HTML (Hypertext Markup Language), 378  
   defined, 378  
   elements, 292

game files, creating, 388  
 tags, 239, 376-379  
 words, 378-380  
 hyperboles, 170  
 [hyperlinked words | Passage Name], 7

## I

<i> tag, 379  
 iceberg effect, 347, 361  
 icons, 11  
 IDs, 364  
 IFDB (Internet Fiction Database), 389-392  
 <<if>> macro, 270-271  
 (if:) macro, 77-78, 312-313  
 images  
   adding, 258-260  
   Creative Commons (CC) license, 256  
   finding legally, 256-257  
   game play enhancement, 257  
   history of graphics in games, 253  
   hovering words over, 264  
   as links, creating, 260-261  
   mood enhancers, 257  
   as setter links, creating, 262-263  
   two-dimensional, 253  
   understanding, enhancing, 257  
 in medias res, 267  
 indexes, 149, 196-198  
 indexOf() method, 332  
 Infocom  
   *Moonmist*, 13  
   *Zork*, 67  
 inside jokes, 194  
 installing Twine, 5  
 intelligence levels (characters), 207  
 interacting-objects puzzles, 59  
 Interactive Fiction awards, 394  
 Interactive Fiction Database (IFDB), 389-392  
 internal links, 31-32  
 internal thoughts (pace), 177  
 interpreted endings, 280  
 Intfiction.org forum website, 393  
 introversion (characters), 140

**inventory**

- checking, 335-336
- creating, 335-336
- exploration choices, making, 222
- limiting, 211
- organizing with variables, 151-153
- storing objects in arrays, 148

**InvisiClue booklets, 191****irony, 169****is (equals operator), 69****is not (does not equal operator), 69****is operator, 271****italicizing words, 379****J****jams, 393****JavaScript**

- indexOf() method, 332
- pop() method, 330
- push() method, 330
- splice() method, 330
- time() method, 351

**Jonah, 349****juxtaposition, 303****K*****King's Quest*, 253****Klimas, Chris, xvi, 349****knights data name value, 199****L****language, pacing, 177-179****less than (lt) operator, 271****less than operator (<), 69****less than or equals operator (<=), 69****less than or equal to (lte) operator, 271****leveling up, 231****levels (story), 14*****Lifeline*, 307****light-source puzzles, 59****lighthearted choices, 18****line breaks, deleting, 273-274****linear stories, 3****(link:) macro, 31-32, 180****links**

- appearance, customizing, 251-252
- clickable, 291-292
- colors, 374-375
- external, 27
- fonts, 252
- images as, creating, 260-261
- internal, 31-32
- lists of objects, listing, 151-153
- placement, 19
- setter

*images as, creating, 262-263*

*pacing, 179-181*

text, 129-132

when appear, controlling, 105-106

**literary techniques**

- allegories, 166
- allusions, 169
- analogies, 168
- hyperboles, 170
- irony, 169
- metaphors, 165-167
- similes, 165-167
- symbolism, 166

**<<live:>> macro, 98****location. *See* setting****locked-door puzzles, 59****Lovecraftian tales, 116****low agency, 15****lt (less than) operator, 271****lte (less than or equal to) operator, 271****M****MacGuffins, 325-326****macros, 60. *See also specific names of macros*****magical realism genre, 120****manipulation puzzles, 59****map-based games, 43-45****mazes, 46-47****meaning, crafting, 18-19****mechanical errors, troubleshooting, 385****Meretzky, Steve, 127****metaphors, 165-167**

**methods, 330-332**  
**misleading players, 184**  
**mixed endings, 281**  
**monsters, battling, 223**  
     actions, 226  
     continuing fights, 227-229  
     player death, 226  
     starting fights, 224-227  
**mood**  
     describing, 38-39  
     enhancing with images, 257  
     text styles, 238  
***Moonmist*, 13, 267**  
**Moriarty, Brian, 49**  
**motion, 37**  
**mouseover clues, 264**  
**(mouseover:) macro, 132-136**  
**(mouseover-append:) macro, 185**  
**(mouseover-prepend:) macro, 187-188**  
**(mouseover-replace:) macro, 135-136**  
**(move:) macro, 198**  
**moves, counting, 311**  
***Mrs. Pepper's Nasty Secret*, 92**  
**multiple choice buttons, 285-286**  
**multiple conditional statements, 81-83**  
**multiply operator (\*), 69**  
**mystery genre, 118**  
**mythology genre, 124**

## N

**named hooks, 62-64, 129-132**  
**narrative voice, 283**  
     consistency, 297-298  
     dialogue, 284  
     first person point of view, 283  
     narrators, 296-297  
     point of view, changing, 298  
     researching through other games, 299  
     third-person point of view, 283  
**narrators, 296-297**  
**narrow-space puzzles, 59**  
**nautical genre, 123**  
**navigating**  
     IFDB, 389  
     proofing copies, 385-386

    stylesheets, 244-245  
     Twine, 9-11  
**neuroticism (characters), 140**  
**New Game Listing page (IFDB), 389**  
**<<nobr>> macro, 273-274**  
**non-human characters, 140**  
**non-player characters, 14, 220-221. *See also* characters**  
**nonverbal communication, 302**  
**notes (testing), 387**  
**notifications, 126**  
**numbers**  
     randomly displaying, 113-115  
     variables, 69-70

## O

**objects**  
     characters, defining, 67  
     collecting, 67  
         *organizing with variables, 151-153*  
         *realism, 148*  
         *storing in containers, 148*  
     conditional statements, 78-83  
     exploration choices, making, 222  
     inventory, checking, 335-336  
     limiting, 211  
     quests, 68  
     variables. *See* variables  
**obtainable choices, 23**  
**open endings, 280**  
**openness (characters), 140**  
**operators, 69, 271**  
**or operator, 271**  
**outlining, 23-24**

## P

**pacing, 93**  
     balance, 323  
     choice frequency, 21  
     expectations, 324-326  
     fluctuating, 308  
     with language, 177-179  
     plots, 174

with setter links, 179-181  
 slowing down, 323  
 speeding up, 322

### **parser games, 2**

#### **passages**

background colors, changing, 245  
 choices, creating, 7-8  
 connecting, 7  
 crossroads, 216  
 deleting, 11  
 detail, 19-21  
 length, 178  
 link placement, 19  
 Punch, 226  
 read, counting, 103  
 repeating text from another, 32-34  
 sending players  
   *backward one, 315*  
   *forward one, 316*  
   *to randomly chosen, 310*  
 single choices, 19  
 special StoryCube, 337-339  
 Spell, 226  
 text, 317-319  
 times player accessed, counting, 313  
 transporting players between, 134-135  
 variables, 71-76  
 World Beyond, 226

### **passing of time, 104-105**

#### **Password game show, 356**

#### **pencil icon (Hover menu), 11**

#### **personal characteristics (characters), 204-205**

#### **Philome.la, 389**

#### **Planetfall, 127**

#### **Play button, 8**

#### **player characters, 14. *See also* characters**

personalities, 37  
 settings based on, 36-37  
 slipping into characters, 304  
 tracking, 83-86

#### **players**

aiming for specific group, 339-340  
 attention, drawing, 40  
 being the eyes of, 36

categories, 327  
 connecting with, 392  
 decisions, troubleshooting, 205  
 defined, 14  
 distracting (MacGuffins), 325-326  
 engaging, 344-345, 354-361  
 entering text, 289-290  
 expectations, 324-326  
 health levels, 224-226  
 keeping moving, 37  
 knowing, 339-341  
 misleading, 184  
 passages accessed, counting, 313  
 sending  
   *backward one passage, 315*  
   *forward one passage, 316*  
   *to randomly chosen passages, 310*  
 text, entering, 137-138  
 transporting, 134-135, 359-360  
 trusting, 360  
 turns, counting, 103  
 types, 342-343

#### **plots**

building, 174  
 defined, 172  
 diagrams, 100-101, 175-176  
 lines, 172  
 pacing, 174  
 points, 172  
 roller coaster analogy, 173  
 story moments versus plot points, 172  
 twists, 126

#### **pointless puzzles, 51**

#### **points of view**

changing, 298  
 first person, 283  
 player characters, setting, 36-37  
 third-person, 283

#### **pop() method, 330**

#### **popping, 158**

#### **post-story**

creating, 279  
 defined, 267  
 endings, 279-281  
 sequels, 282

**preference choices, 22**

**<<prepend>> macro, 292-294**

**(prepend:) macro, 187-188**

**pre-story**

defined, 267

dreams, 278

flashbacks, 278

game tutorials, 278

recollections, 277

stories told between characters, 278

telling versus not telling, 276

working into games, 277-278

**<<print>> macro, 269, 329**

**(print:) macro, 76, 309-313**

**printing**

final game statistics, 231

health levels, 224

randomly chosen words from lists, 309

variable values, 269

**(prompt:) macro, 137-138, 202**

**proofing copies, 385-386**

**protagonists, 138**

**publicizing games, 392**

**publishing games, 388-392**

**Punch passage, 226**

**purpose (characters), 13, 144-145**

**push() method, 330**

**pushing, 156**

**puzzles**

*80 Days* example, 49

building

*end goal, 52-53*

*finishing, 58*

*layers, adding, 54-57*

*starting point, 52-53*

choice, 49

collection, 49

finding, 50-52

hidden object, 63-64

layers, adding, 54-57

macros/hooks example, 61

replayability, creating, 229-230

soup can, 51

types, 58-59

*Wishbringer* example, 49

## Q

**quests**

conditional statements

*adding, 78-80*

*hooks, 79*

*multiple, 81-83*

*player characters, tracking, 83-86*

*structure, 78*

creation, 89

defined, 68

drop-off, 87

end, 87

escape, 88

fetch, 87

rescue, 88

search, 86

transformative, 89

variables, 71-76

## R

**<<radiobutton>> macro, 285-286**

**radiobuttons, creating, 285-288**

**random() function, 310-311**

**(random:) macro, 113-115**

**randomization, 109**

chosen words from lists, printing, 309

numbers, 113-115

text, 109-113

**rationing puzzles, 59**

**reader expectations, 108**

**realism**

object collection, 148

seeker players, 343

**realistic fiction genre, 125**

**recollections, 277**

**red herrings, 184, 190**

**relationships (characters), 145-146**

**<<remove>> macro, 295**

**repeating text, 32-34, 317-319**

**<<replace>> macro, 294-295**

**(replace:) macro, 131, 135-136**

**replacing text after**

clicking hypertext words, 294-295

mouse travels over designated text, 135-136



**replayability**

- battle scenes, 223
  - continuing fights*, 227-229
  - starting fights*, 224-227
- characters, 219-221
- creating, 214
- inventory-based, 222
- puzzles, 229-230

**rescue quests, 88****researching narrative voice, 299****retrieving saved games, 321****<<return>> macro, 316****returning stasis, 99****rewarding choices, 345****rocket ship icon (Hover menu), 11****rolling dice simulation, 113-115****RPGs (role playing games), 47****rule of three, 325****rules, breaking, 394-395****S****sad endings, 281****Saint, Puck, 363****sans serif fonts, 236****Save to Disk button, 321****Saves button, 321****saving**

- games, 320-321
- stories, 9

**science fiction genre, 117****search quests**

- conditional statements
  - adding*, 78-80
  - hooks*, 79
  - multiple*, 81-83
  - player characters, tracking*, 83-86
  - structure*, 78
- variable values, 71-76
- writing, 86

**second-person stories, 3****seconds, counting, 350-352****selectors**

- defined, 245
- setting, 245-247
- SugarCube, 364-366

**sending players**

- backward one passage, 315
- forward one passage, 316
- to randomly chosen passages, 310

**senses, 38****sequels, creating, 282****sequencing puzzles, 59****serif fonts, 236****(set-either:) macro, 110****<<set>> macro, 268, 309-310, 328****(set:) macro, 72-75**

- datamap data name values, adding, 200
- setter links, 180

**(set-random:) macro, 114****setter links**

- images as, 262-263
- pacing, 179-181

**setting**

- being the player's eyes, 36
- character traits, 206
- defined, 29
- descriptions, adding, 31-32
- descriptive words, 40-42
- drawing player attention, 40
- dynamic, 42-43
- importance, 29
- map-based games, creating, 43-45
- mazes, 46-47
- mood, 38-39
- motion, 37
- pacing, 177-181
- personal experience, 35
- player characters, 36-37
- prompts, 30
- repeating text, 32-34
- scenes. *See* exploration
- senses, 38
- static, 42-43
- variable values, 72-75

**shapes (stories), 25-27****Short, Emily, 243****short stories, 171-172****showing not telling, 347-348, 354**

- actions, describing, 356
- behavior, demonstrating, 358-359
- conclusions, drawing, 357

- problems with telling, 355
- transporting players, 359-360
- trusting players, 360
- sidebars, 248, 368-371**
- sidekicks, 139**
- <<silently>> macro, 274**
- similes, 165-167**
- simulating waiting, 105-106**
- size**
  - arrays, 161-162, 333
  - images, 258
  - text, 239-240, 379
- slipping into characters, 304**
- slowing down stories, 323**
- Snowman, 255**
- sound, editing, 384**
- soup can puzzles, 51**
- span class element, 292**
- <span class> tag, 376**
- <span style> tag, 379**
- special passages (StoryCube), 337-339**
- speeding up stories, 322**
- Spell passage, 226**
- splice() method, 330**
- Spunky Spelunky, 363**
- spy thrillers, 118**
- Star Wars: A New Hope text, 233**
- starting points, 6, 11, 52-53**
- stasis, 91-95, 99**
- stat systems, 198-199, 207**
- static setting, 42-43**
- Steampunk, 120**
- stories**
  - archiving, 9
  - diagramming, 100-101
  - endings, 267
  - first-person, 3
  - four choices example, 19
  - levels, 14
  - linear, 3
  - in medias res, 267
  - moments versus plot points, 172
  - outlining, 23-24
  - pacing. *See* pacing
  - plots, 172-176
  - post-story. *See* post-story
  - pre-story. *See* pre-story
  - reasons for details, 13
  - saving, 9
  - second-person, 3
  - shapes, 25-27
  - starting, 6, 11
  - tense, 14
  - third person, 3
  - told between characters, 278
  - versus games, 2
- Stories screen, 5, 9**
- story arcs**
  - catalyst, 91, 96-99
  - climax, 91
  - dead ends for some choices, creating, 101
  - defined, 91
  - different locations, 100
  - drawing, 91-93
  - extending, 102-103
  - stasis, 91-95, 99
  - time, controlling, 104-105
  - turns, counting, 103
  - when links appear, controlling, 105-106
- story seeker players, 343**
- StoryCaption passage, 338-339**
- StoryMenu passage, 337**
- strength levels (characters), 207**
- strings**
  - arrays. *See* arrays
  - values, 196, 268-269
  - variables, 70
- stylesheets. *See* CSS**
- subgenres, 115**
- subtle clues, 190**
- subtract operator (-), 69**
- successful characters, 127**
- Sugarcane, 349**
- SugarCube, 254, 349**
  - arrays, 328
    - adding items, 329-330*
    - creating, 328*
    - deleting items, 330-331*
    - size limits, 333*
    - strings, checking, 332*

- testing*, 329
  - troubleshooting*, 334
  - customizing
    - backgrounds*, 367
    - fonts*, 372-374
    - link colors*, 374-375
    - sidebars*, 368-371
  - declarations, 364-366
  - as default format, setting, 255
  - downloads website, 349
  - functions, 309-314
  - inventory, checking, 335-336
  - macros. *See specific names of macros*
  - saving games, 320-321
  - selectors, 364-366
  - special passages, 337-339
  - stylesheets, 363, 376-377
  - switching over, 254
  - time, 350-354
  - white space, condensing, 273-274
  - words, 378-380
  - surprise endings**, 280
  - sustenance puzzles**, 59
  - symbolism**, 166
- T**
- tabletop role-playing games**, 214, 231
  - teasers**, 341
  - telling not showing**, 361
  - tense**, 14
  - tension**
    - adding, 103
    - building, 177
    - timers, creating, 98-99
  - text**
    - adding, 292-294
    - alignment, 249
    - bolding, 378
    - changing after clicking hypertext words, 294-295
    - color, 209, 240-241, 379
    - customizing, 380
    - delaying, 353-354
    - deleting after clicking hypertext words, 295
    - entering, 289-290
    - flat, 165
    - fonts. *See fonts*
    - hiding/displaying, 129-133
    - italicizing, 379
    - players, entering, 137-138
    - randomly displaying, 109-113
    - repeating, 32-34, 317-319
    - replacing when mousing over designated text, 135-136
    - size, changing, 239-240, 379
    - Star Wars: A New Hope*, 233
    - styles, 237-238
    - transitions, changing, 241-242
    - underlining, 379
    - words hovering over images, 264
  - text adventure players**. *See players*
  - text-align declaration**, 249
  - (text-color:) macro**, 209
  - (text-style:) macro**, 237
  - <<textarea>> macro**, 289
  - <<textbox>> macro**, 290
  - thief data name value**, 199
  - third-person point of view**, 3, 283
  - time**
    - counting seconds, 350-352
    - links appearance, 105-106
    - pacing, 93
    - passing, 104-105
    - text, delaying, 353-354
    - timers, creating, 98-99, 352-353
    - waiting, 105-106
  - time() method**, 351
  - <<timed>> macro**, 352-353
  - Tolkien, J.R.R.**, 213
  - tone, setting**, 183
  - Town**, 32
  - traits**. *See characters, traits*
  - transformative quests**, 89
  - (transition:) macro**, 241
  - transitions**, 241-242
  - transportation puzzles**, 58
  - tropes**, 108
  - troublesome person puzzle**, 52-53
  - trusting players**, 360

**turn-based battles, 224**

- actions, 226
- continuing fights, 227-229
- player death, 226
- starting fights, 225-227

**turn-based systems, 223****turns, counting, 103****turns() method, 311-313****tutorials, 278****tw-link:hover selector, 251****tw-passage selector, 245-249****tw-sidebar selector, 249****Twine**

- archiving, 9
- blue grid screen, 6, 9
- browser compatibility, 5
- case-sensitivity, 35
- external links, creating, 27
- formats, 59, 349-350
- forum, 393
- installing, 5
- saving, 9
- website, 5, 393

**two-dimensional images, 253****U****<u> tag, 379****#ui-bar selector, 369****ultimate endings, 280****underlining words, 379****understanding endings, 281****<<unset>> macro, 269****uploading images, 258****useful-item puzzles, 58****V****value choices, 22****variables**

- creating, 68-69
- defined, 68
- lists of objects, organizing, 151-153
- non-player characters, creating, 220-221

## passages, adding, 71

## strings, 70

## tracking, 75

## true/false values, 70-71

## values

*assigning, 268, 285-288*

*changing, 69-70*

*displaying, 76*

*numbers, 69-70*

*printing, 269*

*random, assigning, 110, 309-310*

*setting, 72-75*

*text based on. See conditional statements*

*unassigning, 269*

**verbs, 179****visited() function, 313-314****vulnerability (characters), 128, 143-144****W - Z****waiting, simulating, 105-106****wandering non-player characters, creating, 220-221****websites**

Creative Commons, 257

Google

*Developers image attributes tutorial, 258*

*Fonts, 250, 373*

hexadecimal color values, 241

Internet Fiction Database, 389-391

Intfiction.org, 393

Philome.la, 389

SugarCube downloads, 349

Twine, 5, 393

**Western genre, 121-122****whitespace, deleting, 252, 273-274****win seeker players, 342*****Wishbringer*, 49****Wittgenstein, Ludwig, 348****wizards data name value, 199-200****Woods, Don, 29****word count, limiting, 178****words. See text****World Beyond passage, 226****Zork, 67, 147**