WRITING INTERACTIVE FICTION WITH TWINE

PLAY INSIDE A STORY

TECHNICAL ACCURACY GUARANTEED BY TWINE CREATOR, CHRIS KLIMAS!

MELISSA FORD

QUE®

Writing Interactive Fiction with TWINE

Melissa Ford

QUE®

800 East 96th Street,
Indianapolis, Indiana 46240 USA

Writing Interactive Fiction with Twine

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 or (800) 382-3419.

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

For questions about sales outside the U.S., please contact 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

Table of Contents

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

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 for convenient access to downloads, updates, and corrections as they become available. To start the registration process, go to 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 feels 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:

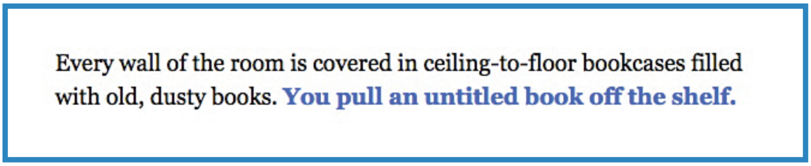
```
(link: '"What the character is saying."')[New words that appear and replace the
linked piece of dialogue.]
```

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.]
```

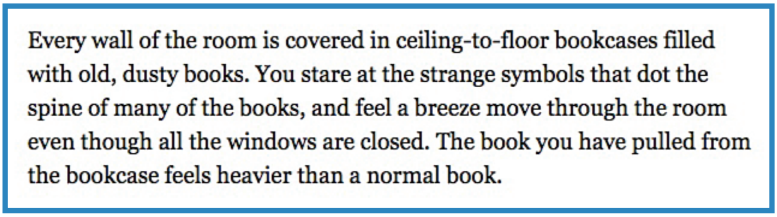
When you play this example, you should see two sentences, one in plain text and one hyperlinked, as shown in Figure 3.1.



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.

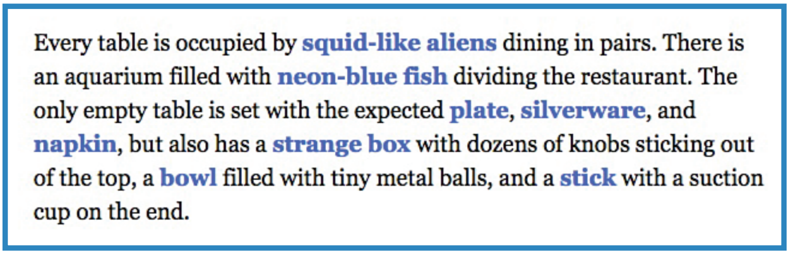
To create using this tool, add this line to any passages where you want to repeat the text of another passage:

```
(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.

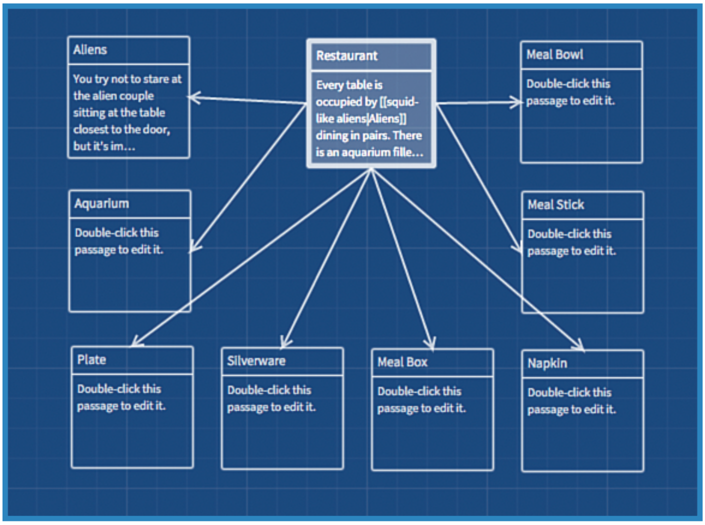
You want players to be able to click on any of the links in the passage to receive more information about the squid-like aliens, the neon-blue fish, or the strange box, but you also want players to be able to make second, third, and fourth choices instead of moving into another section of the story.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



Aliens

You try not to stare at the alien couple sitting at the table closest to the door, but it's im...

Restaurant

Every table is occupied by [[squid-like aliens|Aliens]] dining in pairs. There is an aquarium fille...

Meal Bowl

Double-click this passage to edit it.

Aquarium

Double-click this passage to edit it.

Meal Stick

Double-click this passage to edit it.

Plate

Double-click this passage to edit it.

Silverware

Double-click this passage to edit it.

Meal Box

Double-click this passage to edit it.

Napkin

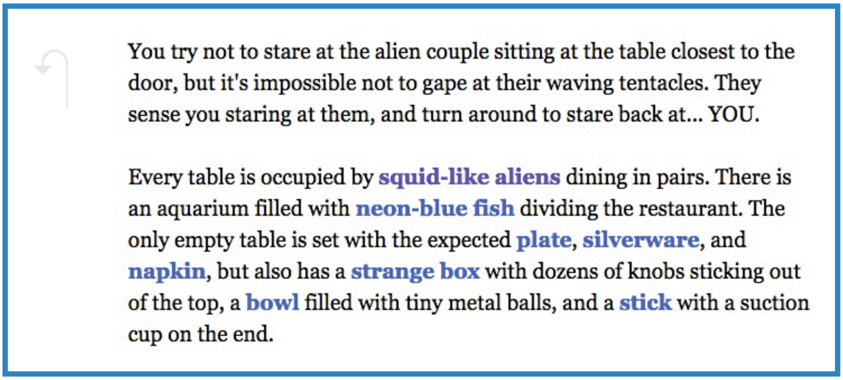
Double-click this passage to edit it.

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

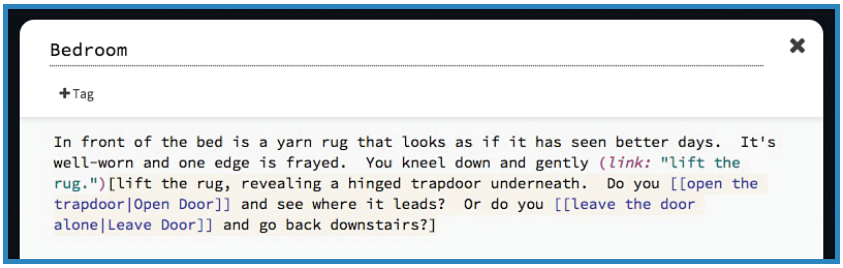
Open the Train game you created earlier. Leave the dining car and find yourself in an empty compartment. The emptiness fills you with dread. Where did all the people go? Why did the door suddenly lock behind you, trapping you inside the train car? Describe the creepiness of the space.

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"`.



```
In front of the bed is a yarn rug that looks as if it has seen better days.  It's
well-worn and one edge is frayed.  You kneel down and gently (link: "lift the
rug.")[lift the rug, revealing a hinged trapdoor underneath.  Do you [[open the
trapdoor|Open Door]] and see where it leads?  Or do you [[leave the door
alone|Leave Door]] and go back downstairs?]
```

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.

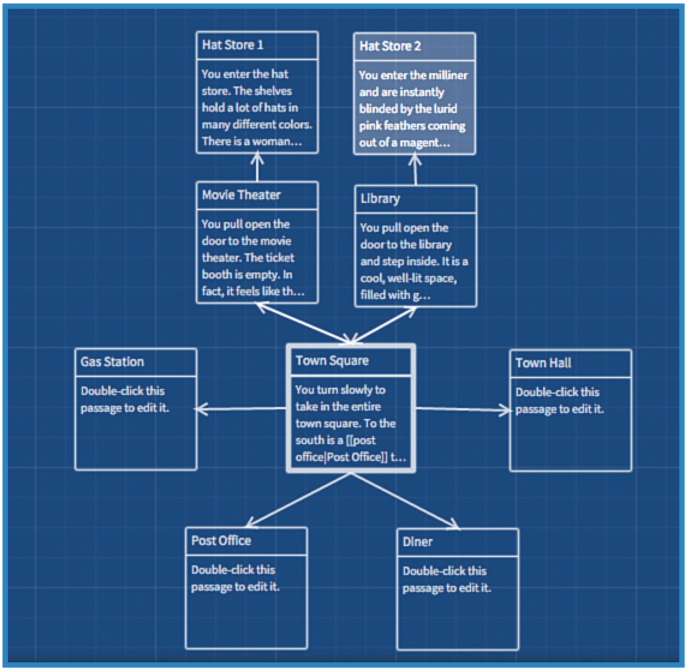
Now open the passage titled Hat Store 1. Write a basic description of the store, keeping it as simple as possible. Copy that description and also place it in the passage for Hat Store 2, but this time, edit that description to make it specific. For instance, you could say that it's a milliner, which 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.
```



Hat Store 1

You enter the hat store. The shelves hold a lot of hats in many different colors. There is a woman...

Hat Store 2

You enter the milliner and are instantly blinded by the lurid pink feathers coming out of a magent...

Movie Theater

You pull open the door to the movie theater. The ticket booth is empty. In fact, it feels like th...

Library

You pull open the door to the library and step inside. It is a cool, well-lit space, filled with g...

Gas Station

Double-click this passage to edit it.

Town Square

You turn slowly to take in the entire town square. To the south is a [[post office|Post Office]] t...

Town Hall

Double-click this passage to edit it.

Post Office

Double-click this passage to edit it.

Diner

Double-click this passage to edit it.

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.

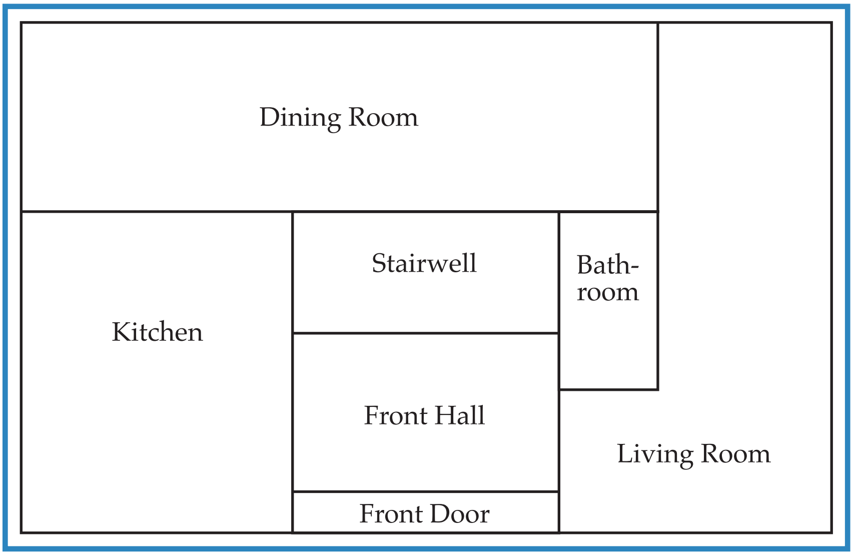


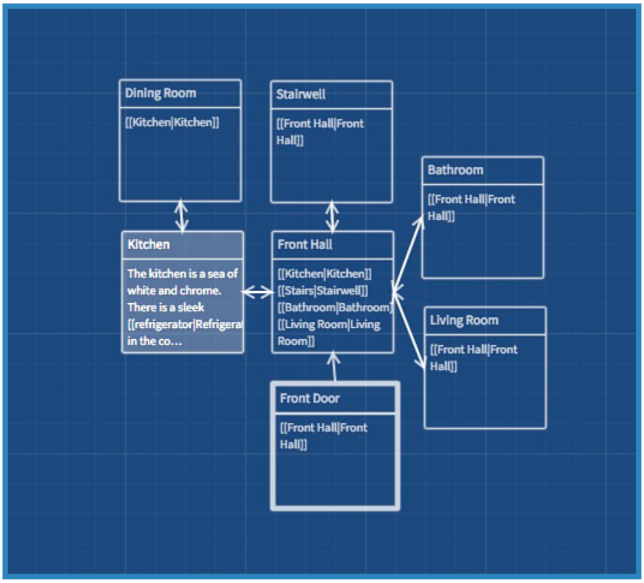
FIGURE 3.8 A map of a house.

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.



Dining Room

[[Kitchen|Kitchen]]

Stairwell

[[Front Hall|Front Hall]]

Bathroom

[[Front Hall|Front Hall]]

Kitchen

The kitchen is a sea of white and chrome. There is a sleek [[refrigerator|Refrigera in the co...

Front Hall

[[Kitchen|Kitchen]]
[[Stairs|Stairwell]]
[[Bathroom|Bathroom]]
[[Living Room|Living Room]]

Living Room

[[Front Hall|Front Hall]]

Front Door

[[Front Hall|Front Hall]]

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.

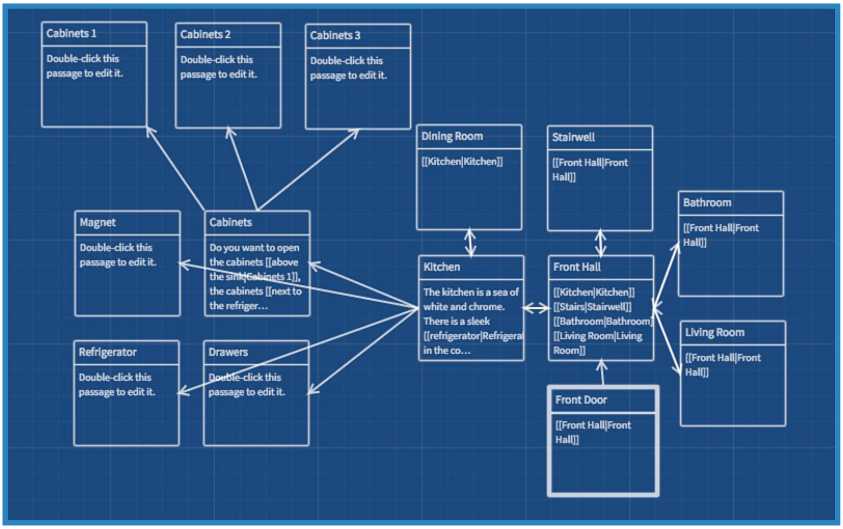

Cabinets 1
Double-click this passage to edit it.
Cabinets 2
Double-click this passage to edit it.
Cabinets 3
Double-click this passage to edit it.
Dining Room
[[Kitchen|Kitchen]]
Stairwell
[[Front Hall|Front Hall]]
Bathroom
[[Front Hall|Front Hall]]
Magnet
Double-click this passage to edit it.
Cabinets
Do you want to open the cabinets [[above the sink|Cabinets 1]], the cabinets [[next to the refriger…
Kitchen
The kitchen is a sea of white and chrome. There is a sleek [[refrigerator|Refrigera in the co…
Front Hall
[[Kitchen|Kitchen]] [[Stairs|Stairwell]] [[Bathroom|Bathroom]] [[Living Room|Living Room]]
Living Room
[[Front Hall|Front Hall]]
Refrigerator
Double-click this passage to edit it.
Drawers
Double-click this passage to edit it.
Front Door
[[Front Hall|Front Hall]]

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