Chapter Seven
By Your Bootstraps

This chapter is here to help you through your first game. It's not intended as an instruction manual, as an encyclopedic reference, or even as a step-by-step guide. Rather, it's a series of tasks designed to stimulate you creatively, to get you thinking about game design decisions, and to actively involve you in working on a game—which is the most important part. Lots of would-be game creators spend such a long time planning dream games that their ambitions are far too large for them to achieve, or even to know where to start. For others, the whole notion of making a videogame is alien and it's difficult for them to know how to even approach the work from a technical perspective. The steps of this chapter attempt to introduce concepts of game making in a less intimidating way.

The tasks are purposefully left abstract so that they can be applied to any tool you choose to work with. This means that I'm not providing specific instructions for how to make a character move around in Game Maker, for example. Figuring that out is up to you. Fiddle around, read the documentation, ask questions online. By working it out yourself, you'll start to learn your way around the tool of your choice.

Some tools are very different than others. Twine and Inform, for example, are for making games presented largely through text alone. You'll need to be creative in adapting the tasks below to fit the kind of game you can make with those tools. Remember that these tasks aren't instructions: they're just here to point you in the right direction. Diverge from them as often as possible; do them out of order; ignore the ones that don't fit your vision; leave them behind once you've found your own way. This is just the start of the path. The end is everywhere.

Task #1: Choose a Tool

The first step is to pick a tool—either one that sounds like it would be useful for the game you have in mind, or one that seems interesting to you and that starts you thinking about what kind of games you might like to make, or that suggests possibilities that your game might explore.

To make the kind of games most people intuitively think of when they think of videogames, The Games Factory (for Windows) or Game Maker (for Windows or Mac) are strongly recommended for being fairly robust. They're capable of crafting a great variety of games that look and play differently without much difficulty. Game Maker is probably the best-documented tool of its kind, but the current version (rebranded "GameMaker" by the soulless corporation that currently controls the intellectual property) makes it difficult and unrewarding to self-publish. But starting with an older version of Game Maker that publishes games for free is a good idea.[77]

If Game Maker and The Games Factory are too intimidating, try Scratch. Scratch provides a good introduction to thinking about games in terms of rules, an essential habit for more advanced game making. Every rule in the Scratch editor is essentially a sentence made up of smaller phrases, which helps you understand how rules create the actions in a digital game. There's also a tool called Stencyl that uses the same concept as Scratch, but produces Flash games for online distribution.

I'm also partial to Twine for games that are closer to written fiction. Though you can create projects that include images and other graphical elements if you have knowledge

of HTML and CSS (in which Twine is based), Twine is essentially a tool for creating text-only games. It's intended for making branching stories in the vein of Choose Your Own Adventure books, and has a clean visual presentation that shows how passages in the story are connected. Other text game creators like Ren'Py and Inform 7, described more fully later, provide other simple options for creating this kind of game.

In the next chapter you'll find a fuller list of the tools that I think will make the most sense to people who have no programming or game making experience. I've tried to describe what each tool is good for and what the experience of working with each tool is like. Try more than one. You can always learn to use a new tool. Each one will teach you something different about design.

Task #2: Introduce a Character

A character could be abstract or concrete. It could be a person, an animal, a symbol. It could be represented by a photo—one of yourself or of a loved one or of a stranger that you've grabbed from Google's image search. Mold a character out of clay and take a picture. Open up MS Paint and doodle a cartoon. Find a picture of an American president and draw horns on it. Or just use an image file that comes with whatever tool you're using. Klik & Play includes lots of clip art, and Stencyl lets you grab animated characters from an online database.

Take this image of your character and find a way to start a game and display the character on screen.

Once you have that, imagine what this character might be capable of doing, and how that will characterize her. It can help to think of this in terms of a question: what job does this character have? Think about what kind of conflict the character might have: what makes it difficult for her to do that job? I once made a game about a robot gardener: her conflict was that coming in contact with water would cause her to short circuit.

Let's say your character starts from a picture of one of your friends, grabbed from Facebook. In the photo your friend is wearing a sombrero. Why would this character you're creating wear a sombrero? Perhaps she's a MATADOR. What forces come into conflict with a matador? Perhaps she'll have to face some RAGING BULLS. A story starts to develop from a single picture that you've taught your program to display on the game screen. (Or the story starts to develop and you change the picture to fit the new story.)

If you're working with something text heavy like Twine or Inform or Ren'Py, you can, instead of using an image, write an opening passage that characterizes the protagonist, what she's capable of doing, and what her relationship is with the world. In either case—text or image—this is the first piece of information the player will see when she loads your game. So you should choose something that tells her about the game and its world.

Task #3: Teach Your Character to Do Something

Specifically, teach your character to do something in response to the player. This means teaching the game to TAKE INPUT. The player should be able to press the spacebar, or click the mouse, or type the letter A, or *something*, and your character should respond to that input somehow. Maybe your character turns invisible while the SHIFT key is held down. Maybe your character flies up the screen when the player hits the UP arrow key. Maybe when the player clicks somewhere on the screen with the mouse, the character teleports there.

As you do this, think about how what you're teaching the character to do characterizes her. How does being able to teleport anywhere on the screen fit her character? What does that ability tell the player about her? Maybe she's not just any matador, but a MATADOR FROM THE FUTURE. Maybe the raging bulls she comes into conflict with are charging from one side of the screen to the other, and she has to teleport to stay out of their way.

This is your game's first rule. When the player does X, the protagonist does Q. This is the kind of rule I like to call a "verb," because that's the part of speech this rule would correspond to in a sentence describing your game, one that governs the relationship between the player's character and the other characters in your game. In other words: FUTURE MATADOR (subject) TELEPORTS (verb) to avoid RAGING BULLS (object.)

If you're working on a text game, write a second passage that describes a location, be it a physical place or a state of mind, so long as it's different from the one the game starts in. Maybe the player starts in a desert, but she can see an oasis in the distance. This second location can be the oasis. Now give the player some means of getting her character to the new location.

Task #4: Introduce a Second Character

This is the object of the previous sentence. This character should have a relationship to the first character, who we'll start calling the player for short. Maybe the second character represents something the player wants to avoid (raging bulls) or something she wants to acquire (a bag of cash). (Or maybe she wants to avoid the corrupting influence of the bag of cash and touch the confused raging bull, gently, in order to calm it.)

The relationship between the characters should have something to do with the player's verb. For example, if the player can FLY UPWARD and the object is THE SUN, which the player wants to touch, then writing the game's rules such that the player can fly upward high enough to reach the sun will change the experience of the game. If the player can fly, but never high enough to reach the sun (or perhaps the player can only fly straight upward, and the sun is a little to the left or right), you'll have made a game about unattainable dreams, ones that remain just out of reach. If the protagonist can reach the sun but it burns her wings off, causing her to fall back to the earth and the game to end, you'll have made a cautionary fable. If she can reach the sun, put it in her pocket, and earn one hundred points for it, you'll have made a power fantasy.

Rules change not only the meaning but also the feeling of a game. In our future matador game, maybe you'll decide that a raging bull always moves left and right across the screen, or instead, that it always moves toward the protagonist. The latter bull is probably more difficult to avoid than the former. Once you have your rule in mind, experiment with different ways of implementing it to find a feeling and balance that you like.

It's most likely that you want this character to abide by a rule that's internal to the game—that is, one kept by the computer, not controlled by the player. You could, of course, experiment with giving the player two characters to control in different ways (maybe one of them is a mirror image that moves left when the other moves right), or with making two different characters that are controlled by two different human players.

If you're writing a text game, introduce an object to the story, one whose relationship with the player introduces a conflict. Maybe when the player reaches into the oasis for a

drink, she discovers what appears to be an old, waterlogged map.

Task #5: Make Some Noise

Put a sound effect in your game. Think about which interactions in your game would be clarified by the playing of a corresponding sound. For example, if our matador makes a sci-fi BZZIP sound when she teleports, it would definitely characterize her as a matador that uses future technology. And it would act as punctuation to the teleport, helping the player to understand that clicking to move the character isn't a mistake, but what she should focus on. The bull striking the matador might be accompanied by a drawn-out scream to let the player know that it's an undesirable outcome (or a choir of angels singing if the future matador secretly seeks a glorious death in the ring).

How do you find sounds to put in your game? Open the sound recording program that comes with your computer (or download a free one, like Audacity[78]) and record yourself making sounds into your computer's microphone. Try shifting the pitch or the tempo, reversing the sound, adding a thousand phase or distortion effects. Andrew Plotkin made all the sound effects for his 1994 shareware game, *System's Twilight*, by recording himself making silly noises into a cheap mic. You can even download his sounds if you want; he's made them available.[79] You can find plenty of sounds online, at sites like Freesound.org. And many tools, in particular Game Maker and Klik & Play, come with many of sounds for you to use. There are programs that will let you generate your own sounds directly, without a mic: a free program called SFXR[80] lets you easily generate Nintendo-like sounds.

In a typical videogame, the screen communicates a lot of visual information. In a busy game, it can be hard to follow absolutely all of what's happening on the screen. Sound communicates on a different channel, giving you a separate and sometimes more immediate layer of information. Use sound to underline the important relationships in your game.

If you're making a text game, the tool you're using may not have an easy way to play sound, or playing a sound effect might be jarring in a game that's all read. Your story might not need sound at all. You might try fiddling with some kind of background ambiance, like a loop of birds chirping or water running that only plays at the oasis, to differentiate that location from the desert. Or maybe a quick sound like a SPLOOSH of water or a CHIME of discovery upon pulling the map out of the water.

Task #6: Round Out the Player's Vocabulary

Give the player's character more verbs; give the player more ways to communicate with the game. Try to think of new rules that will interact with the existing ones in interesting ways. For example, what if you make your future matador into a future cowboy? In addition to being able to teleport, she can also drop a ZAPPO LASSO, which will catch bulls that run across it and return them to the future. But maybe being able to get rid of the bulls makes the game feel too easy, so maybe she can only drop the lasso where she's standing. This means that in order to trap a bull, the player will have to use her teleport to put her in the path of the bull, set the trap, and teleport away again. The conflict escalates.

In the case of the Icarus character who flies UP to reach the sun, maybe she can move LEFT or RIGHT as well. If she moves off the left or right side of the screen, she'll travel around to the other side of the planet, where it's night and where she can land safely on the moon instead. (The transition can be as simple as changing the background from a

sky blue to a night black, and the sun to the moon.)

Think about other rules you can add. Rules don't only have to be verbs; rules can also be *adverbs*. Your future cowboy might be able to teleport SLOWLY. In practice, this means that it takes a few fractions of a second between the player clicking the button and the cowboy arriving at her destination, which forces the player to plan more carefully and react more quickly. Maybe Icarus is affected by gravity, and whenever she's not flying up, she's falling slowly down. You could give her the ability to move up, left, and right when the player presses buttons, but she has to depend on gravity to move down.

Try to ensure that all the rules in the game have relationships with each other. If you think of them as a vocabulary—nouns, verbs, and adverbs—try to construct a vocabulary with which you can tell engaging stories.

In your text game, give the player a way to interact with the object she's encountered. Maybe if she tries to read it, she's told it's too waterlogged to make sense of. Maybe she has to take it back to the desert to dry it off. The way she interacts with the object should advance the story. Now that the player's returned to the desert, maybe the dried map leads to a treasure hidden somewhere in the sands.

Task #7: Design a Level

I'm using the term *level* to mean a sequence of events the player has to negotiate using her vocabulary of verbs. In other words, a level is a story: the tension that rises before the climax of the game resolves it.

A story doesn't have to be complicated. Levels for *Future Cowboy* could just be an escalation. Every time our cowboy catches a bull, it enrages two new bulls who charge onto the screen, causing the bull population to gradually increase with the player's performance, and making the bulls harder for the player to manage. This is rising tension too, and each new bull on the screen is a new chapter, ultimately telling a story about human performance against overwhelming odds (or whatever you like).

However, some stories you want to tell might demand more complicated levels. What if the sun our Icarus wants to reach is blocked by storm clouds that will electrocute her if she touches them? The storm clouds make the sun a more valuable goal—even though, in the version of the game where Icarus burns up when she touches the sun, it's a negative goal—because they require Icarus to work harder in order to reach it. They also force the player to recognize Icarus's ability to move left and right, making it more likely that she'll decide to fly Icarus off the edge of the screen, and thus more likely that she'll realize that the moon might also be a valuable goal. You can push this possibility further by leaving clues: plant a trail of storm clouds along the side of the screen, leading to the edge, or—to be even more obvious—make the side of the screen a wall of clouds with an Icarus-sized hole in it.

Once you've gotten the player on the night side of the planet, you might give her a trickier maze of clouds to navigate, cluing her in that the moon goal is more desirable. Now you're starting to tell a really interactive story: the player can take the easier and more obvious route of flying to the sun and being burned, or she can take the more difficult and oblique route to discover this newer goal. (Notice that I haven't said anything here about telling the story explicitly, having Icarus shout out a text bubble saying, "I MUST REACH THE GLORIOUS MOON!"—if done carefully, letting the player discover the moon on her own by accident or by following clues, the player will get the idea, and consider it that much more interesting and personally relevant because she discovered it herself). The most difficult part of the story then becomes the climax: now

that the player's goal has changed, she uses all her skill and patience to reach it.

If navigating the storm cloud maze around the moon is the climax of the game's tension—then maybe on reaching the moon Icarus arrives at a third, easier scene that can serve as the game's denouement. On this screen the player just has to guide Icarus to a Greek-columned house sitting at the bottom of the screen, moving left and right to avoid obstacles (maybe they can be hot stars instead of storm clouds) while gravity pulls her downward. This adds some variety and makes the scene distinct. What literary critics call the "falling action" of the story could be Icarus literally falling toward the surface of the moon and the game's conclusion.

Notice how this story was entirely told using our basic verb set: all Icarus is doing is flying, but we've given her interesting goals that cause her to use her verb set in interesting ways. Although the *Future Cowboy* story is maybe less complex, it's still a story entirely told through the player's two major verbs—TELEPORT and LASSO—and the relationship they put her in with the objects in her world. Plan a story that will develop the player's relationship to the rules of your game, especially her verb set.

If you're writing a text game, give the player somewhere to go. Maybe the map says something like, "Go west until you reach a mushroom-shaped rock, then travel north until you find the skull of a giant wizard, then head east to the secret cave." Maybe there are things for the player to do at these places, maybe not. But they should be at least interesting to visit: make them distinctive, write them descriptions that are evocative or silly or weird.

Task #8: Finish the Story

The beginning and resolution of a game do a lot to tell the player how to think about the experience in between. Think about *Gay Sniper*, which I described in the previous chapter. The opening screen—with the words "Gay Sniper" and the opening tinkle of a designed-to-tug-at-your-hetero-heartstrings public service video—and the closing screen—"America is destroyed"—do a lot to characterize the absurdity of the political message they bracket.

Not every game needs a title screen, but a title screen is very good at giving the player a context with which to understand the experience that follows. Think about what the first thing the player sees tells her about the game she's about to play. The title *Future Cowboy* goes a long way toward characterizing the rules of that game: along with whatever art you choose, the title gives a context both to the teleportation and to the bull-trapping in that game. If the title screen of *Icarus* shows a picture of Icarus on the ground, reaching toward the sun, that effectively conveys to the player that the sun should be Icarus's goal—and also makes the player more excited when she realizes that the title screen is not being entirely straightforward with her about what the game's true goal might be.[81]

The way your game ends is the last thing the player will see, and will similarly have a great effect on how she interprets the experience as a whole. The ending could resolve the story or subvert it. Icarus could freeze to death away from the sun's heat, or might live happily ever after soaring in the lower gravity of the moon. If the game program closes, it lets the player know she's reached the resolution of the story and there's no farther to go. But what about the other game-ending scenario in the Icarus game, when Icarus touches a storm cloud and is electrocuted? When this happens, the game could restart rather than close, telling the player she hasn't found the resolution of the story yet. Getting burned by the sun could lead to the game restarting as well, to clue the player in that there's another, more desirable resolution.

In the text adventure cave, maybe the player finds something valuable or maybe something useless. Maybe, after traveling so far from the oasis, she frees a genie who offers her "the greatest gift one can give in the desert: water!" Maybe to enter the darkness of the cave, she has to set the map on fire and use it as a torch, destroying her

means of finding her way back. Whatever she discovers in the cave, the text should make it clear to the player that this is the story's resolution: she's found the ending.

Think about how the beginning of your game, what the player experiences during your game, and the ending(s) of your game form a story. What story are you telling, and how will the beginning and ending help provide the context you want for the game?

Task #9: Have Someone Play It, Then Change It

You should be playing and replaying your game with every change that you make to see if it works in the way you expect. But you should be having other people play it, too. Because of your perfect knowledge of how the game works and what it contains, you can't get an accurate sense of how a player will experience your game from your own plays alone. You'll get the most information about what the player's experience actually is by having people who are unfamiliar with the game play it.

Get as many people as you can to play. Try to have an idea of how experienced these players are with videogames to give you some idea of how players with different backgrounds will react to your game. I usually try to have the widest variety of players as I can, and to have people play the game both while it's being constructed and after I consider it finished. Often I'll have different people play the game at different stages, so that I can ensure that players playing the "finished" game are seeing it for the first time.

If you're still putting the game together, give your player an idea of where you want to take the game so that she can give you ideas for how to take it there. If it's finished, don't tell her anything. Watch her play the game and don't make any comments or assist her unless she asks. Make notes on what she does first, where she gets hung up, whether she reacts as you expected, at what points she gets confused. Also make notes on anything that doesn't work as it should—and if any interesting consequences result when things don't work as they should. Someone less familiar with the game than you is more likely to stumble upon mechanical errors because she doesn't know the range of possibilities in the game. What if the player flies Icarus off the top of the screen and disappears? Nothing prevents the player from doing so because you never expected her to: you knew exactly what the goal was, so in all your plays of the game you've never flown off the top of the screen.

When you sit someone down to play the game, keep in mind specific parts of the game that you're uncertain about. Watch how she deals with them, then after she's done, ask her what she thought and whether those parts of the game worked or not. She may have some suggestions of how they or the game as a whole can work better.

Now that you have a better idea of how the player experiences your game, change it! Fix errors, smooth snags, add whole levels if you need to, implement some of the ideas your players suggested, and see if they improve the game. Have them play the game again. Have someone new play the changed game. Change everything, have the original player play again, and see which version they prefer.

Task #10: Distribute Your Game

What's the use of a game that no one can play?

Before broadband Internet, small game authors either had to find a publisher to pay the cost of manufacturing the games and putting them on store shelves, or they had to come up with clever solutions—shareware, which we talked about in chapter 2, being a popular one—for distributing their games. You, however, can take advantage of the speed and accessibility of the Internet to digitally distribute your game.

Upload your finished game to a free file sharing website like MediaFire.com. Some tools, such as Stencyl, have official websites where authors can make their games available. Start a free blog to keep track of all your games; put them on your ancient Myspace page; ask your significant other to upload them to a secret directory of the web server at her job. All you need is some web space for the files and an address you can send people a link to.

Once you have that, spread the link around! Put it on your Facebook page, on your e-mail signature, on a postcard you staple to local telephone poles, everywhere. E-mail it to IndieGames.com. A good place to start: most tools I talk about in this book have communities with attached forums where you can show your game to peers. Post your link there, then read any feedback or criticism you receive (even though most of it will be dumb). Keep it in mind when you make your next game.

Just because digital distribution makes things easier, however, doesn't mean you shouldn't be creative about passing out your game. A friend of mine who puts out his games under the label "Amon26" used the online publishing site Lulu.com to sell CDs containing two games of his that had previously been available on the Internet, along with scans of concept art and MP3 files of the soundtrack.[82] Rob Fearon, a British author and father, asks a one-time donation in the amount of the buyer's choosing for a collection of all of his games.[83] Burn your game to CD and sneak it onto a store shelf, leave a basket full of free copies at a local coffeehouse.

Task #11: Make Another Game

Do all of this again, using what you've learned from your first game. Make an entirely different game, use an entirely different tool, use completely different verbs than you used before. Or don't: make the same game over again, but slightly better: *Icarus 2*, with day, night, and twilight levels.

You're a zinester, after all. Whatever you're doing is right because you're doing it, and that's valuable. Don't worry about being brilliant or original—just make sure you're creative.

Footnotes

77 You can download this older version at http://www.auntiepixelante.com/?p=1240.

78 http://audacity.sourceforge.net.

79 http://eblong.com/zarf/twilight.html.

80 http://www.drpetter.se/project_sfxr.html.

81 You might also be clever: if the design and presentation of the game *Icarus* is exactly as described above, but the title screen shows a picture of Samuel Gompers and bears the title *The Story of Organized Labor in America 1886–1894*, the entire experience essentially changes. A little of this goes a very long way.

82 http://www.lulu.com/product/cd/and-you-will-be-fit-for-more-than-ashes/6053570.

83 http://bagfullofwrong.co.uk/bagfullofwords/whats-in-bundle-of-wrong.