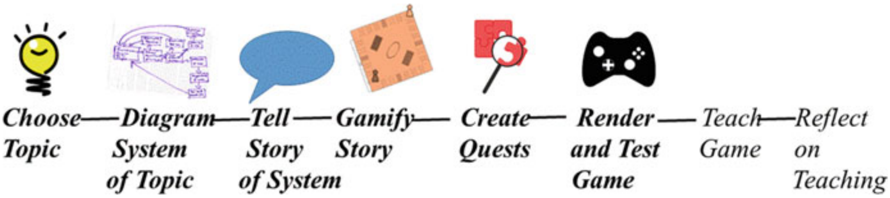
Chapter 7
Game Changer: Rendering and Testing the Game

> *A game isn't automatically fun just because it's about pirates … The difference between good and bad games is more in the polished game experience than in the content.*
>
> -Kurt Squire, 2011

Abstract This chapter dives into the specifics of creating the physical game with varying levels of technology. The chapter provides ways the reader can create a no-technology version of the curricular game. This no-tech version is then used to playtest the game with peers and with potential students. Once this feedback has been used to revise and refine the game, the chapter then discusses ways to repurpose technologies the reader is likely to be familiar with, such as PowerPoint, in order to make a low-technology version of the game. How to do this with medium technology, or technology readers may be unfamiliar with, and with programming, or high technology, is also explored.



J.J. Kellinger, *A Guide to Designing Curricular Games*,
Advances in Game-Based Learning, DOI 10.1007/978-3-319-42393-7_7

RENDERING QUEST

```
IF the primary performance objective has the word "understand" in it, THEN replace "understand" with a concrete action verb.
    ELSE WHILE primary performance objective does not match core game mechanic, DO revise one or both UNTIL they match.
        WHILE quests do not match core game mechanic, DO revise quests UNTIL they match.
            WHILE game story does not incorporate quests, DO revise game story and/or quests UNTIL all quests are embedded in the game story.
```

Day 90 is quickly approaching. After telling Amy you are designing a game for your class, running your ideas by her, and incorporating her feedback, you have finally gotten Amy's approval—something you have always longed for in your friendship with her. Excitedly you run to the Commander with your design document and description of quests, eager to hear her feedback. Instead, she glances at it quickly, stating, "What is this? This isn't a game. I can't play a word document. Don't come back to my office until you have a game I can play!"

Dejectedly you return to your living quarters. Your body is physically wrecked. Lack of sleep and lack of exposure to the sun has depleted your resources. You crash on your bed, resigning yourself to losing the Commander's game. There is no way you can program a video game in the time remaining. If only you could bring Amy into the future. Typically you would cry yourself to sleep over something like this; however, you are too weak to even do that. You fall into a deep slumber.

You dream that you are a pawn in a game of chess. You move around the board until you encounter a knight who quickly beheads you.

You wake up, shaking your head. Remembering your dream, you groan. You realize the knight is the Commander and you, just a pawn in her game. You have easily been defeated. You go through your routine of putting on your clothes and head to breakfast. As you chew on your institutional eggs, you reflect on your dream: "If that were the message of the dream, the Commander would be the queen. Perhaps that is not what the dream meant at all. Maybe I don't have to code an actual video game! She said, 'until you have a game I can play.' Chess is a game that can be played that is not a video game. I can create a learning game where students are the chess pieces, the elements of the game! I don't have to program a video game. I can create a game without any technology at all!"

You barely swallow your eggs as you abandon your breakfast and run at a full sprint back to your room.

ACTION: First, run your curricular game idea by some of your students or by people in that age range and get their feedback. Then, design a no-tech version of your game that does not rely on anything with an on/off switch. Playtest your game by playing it yourself, having a peer play it and give you feedback, and having students—or people in the age range of your intended audience—play it and give you feedback. After reflecting and revising based on that feedback, create a low-tech version of the game using software you are already familiar with. Then, if time, create a medium-tech version using software that is a stretch for you. Then, if time, create a high-tech version using game design software.

One evening I was out with a teacher friend who said she needed to go home to prepare her lesson for the next day. I wanted to stay out longer, so I asked her what the lesson was about, thinking I'd never be able to help her since she taught science. She said her students had to review for a test. I said, "Easy Peasy! Put them in groups and have each group design a sample test. Then, have the groups exchange tests, grade each other's tests, and give back the results. See, your planning is done." This was a generic curricular plan I came up with (I'm sure I wasn't the first) that had no time investment and a big pedagogical payoff. Students often feel tests are something done to them and do not even think about what goes into designing a test, let alone grading one. This plan allows students to think about what might be on the test, get exposed to what their peers think as well, think about what the best format is for assessing that information, and how to evaluate different possible answers (what counts as right? do you give partial credit?). In other words, it gets students to role-play the teacher and, in doing so, become better students because they learn how to play the "game" of test-taking by being a test-designer and evaluator. By focusing on what students do when test-taking instead of what students need to know, I was able to create a successful lesson.

As the quote at the beginning of the chapter suggests, you want to create a "polished game experience." You might be thinking, "How am I going to do THAT in my classroom?" Don't panic. What's most important are the four compelling game elements identified by Malone and Lepper (1987), all of which can be done without any technology:

> Malone and Lepper (1987) define four characteristics of games that contribute to increases in motivation and eagerness for learning. These are challenge, fantasy, curiosity, and control. Challenges in a game … keep [students] engaged with the activity by means of adjusted levels of difficulty. Fantasy in a game increases enthusiasm by providing an appealing imaginary context, whereas curiosity offers interesting, surprising, and novel contexts that stimulate students' needs to explore the unknown. Finally, the control characteristic gives learners the feeling of self-determination. (Akilli 2007, p. 6)

This might sound daunting, especially if you think about the amount of time that might be involved in creating a curricular game that embodies these four characteristics. As a teacher, I know firsthand that time is the biggest constraint in teaching. I have often said that teaching is like a goldfish—it grows to fit the size of its container. In other words, teaching is never done—there is always be more to do—so planning expands to fit the amount of time you allow it. I have spent hours and hours and hours preparing a lesson that completely falls flat. On the other hand, there have been times I have spent just a few minutes on a lesson plan that is a tremendous success. The point here is that designing an engaging lesson, even a game-based lesson, unit, or course, does not have to take a tremendous amount of time. It can even be a time-saver if you design a game where students contribute to the content. However, sometimes the time investment does pay off. After all, you get to play, I mean your students will get to play, your curricular game year after year.

Below is an example of a game-based weekly course about teaching that I designed in less than half an hour while writing this chapter (refinements and creating the materials which are not included took longer):

Day 1 Observation Quest—INTRO and develop teacher and student profiles, and watch teacher video and annotate until annotation matches teacher annotation.

Day 2 Tutoring Quest—pair up to teach a "language," study profile of tutee, must pass language quiz to move on, and if not, teacher must teach some more.

Day 3 Small Group Discussion Quest—lead a small group discussion on a topic of your choice. Each teacher will be given a secret goal you cannot explicitly name. Once goal is reached, you pass (5 goals for each group: a student challenges another student, a student refers back to what another student said, a student asks a question of another student, a student changes his/her mind, and have an exchange among three students in a row without teacher intervention).

Day 4 Rubric Quest—reverse engineer a holistic rubric for a set of essays by dividing into A, B, and C piles and describing each, convert to an analytic rubric, and use to grade new essays until a score that matches those given is reached.

Day 5 Feedback Quest—give feedback on a series of revisions to a college essay, and only get the next version after specific feedback is given.

Days 6–8 Branched Unit Quest—in small groups, plan a branched unit and execute parts of it with diagnostic assessment to determine first decision point; choose topic and initial planning, design diagnostic assessment, and give for homework.

Day 9 Analyzing Data Quest—score the diagnostic assessment, determine a plan of action, and plan a hook lesson.

Day 10 Hook Quest—tag teach (each student in a small group teaches 10 min of the same lesson); hook (must get every single student to contribute, have an option to pause, to consult, and to rewind).

Day 11 Objective Quest—plan a hidden objective lesson OR teach another hook.

Day 12 Objective Quest—tag teach a hidden objective lesson and do not say the objective of the lesson; if at least 75 % of students identify the objective, move on OR plan a hidden objective lesson OR teach another hook; they can have an option to pause and consult, but cannot rewind.

Day 13 Zinger Quest—plan zinger lesson OR teach or plan previous lessons not passed.

Day 14 Zinger Quest—teach zinger lesson (OR do past lessons) which will be videotaped with no pausing/consulting.

BOSS LEVEL: video commentary on your part must include classroom management, content, and pedagogy comments, and each type must have positive and negative comments with evidence and negative comments must have suggestions for the next time which include zingers

Day 15 Debrief class-group reflection on process/what was learned/what should have been done differently the next time the class can share and discuss as a whole.

Notice how each quest has a specific goal the player/student has to achieve before moving on to the next one, with each subsequent goal requiring a greater level of teaching skill, thus maximizing *challenge*. Students are role-playing students when they are not role-playing themselves as the teacher, thus engaging in the *fantasy* element. I embedded "zingers"—actions given to students based on their student character profiles that they would act out and teachers would have to respond to, and the student role-playing aspect lent itself to natural surprises, thus engendering *curiosity* as to what might happen next. For example, one "student as student" came out as gay during a teaching session, forcing the current "student as teacher" (as well as others) to think about what he/she might do if this happened in an actual classroom. Lastly, students are *controlling* their behaviors—as teachers and as students—in reaction to other characters' behaviors and to the game goals, both immediate goals and the overall goal. In less than half an hour, I constructed the outline of a game-based course that embodies all four engaging elements identified by Malone and Lepper (1987).

In addition to these four elements, this course configuration allows for "play," or "the free space of movement within a more rigid structure" (Salen and Zimmerman 2004, p. 304), an element I would argue should be added to Malone and Lepper's (1987) original four. Bogost (2007) describes this "possibility space" in a video game as "the myriad configurations the player might construct to see the ways the processes inscribed in the system work. This is what we really do when we play videogames: we explore the possibility space its rules afford by manipulating the game's controls" (pp. 42–43). This is what I did in the first version of the course outlined above. Students in groups designed a unit plan and taught it by role-playing teachers and students for their fellow students as teachers.

However, this new iteration adds, or increases, the "game" element by setting specific goals (as opposed to the old version which just had a general "design a unit and teach it" goal). These goals are fun for students to attain, I hope, and students must achieve these specific goals in order to move on to the next level. For example, having "students as teachers" try to get the "students as students" in their small discussion group to exhibit a certain behavior without explicitly saying it is similar to the premise of several party games. I toyed with the possibility of having students do the small group discussion challenge in front of an audience who knows the target behavior to make it more like a game show; however, I did not want to frustrate, embarrass, or make a student cry. Plus, putting it in front of an audience not only creates logistical issues with time but also requires me to come up with a lot more target behaviors. Note that there is still "play" in this—"play" in the sense of "play in a steering wheel" in that students choose how to achieve these goals. It is important to also note that these goals are embedded in larger contexts—both the hypothetical one created in my class and students' own goal of wanting to get licensed to teach. I added an element of realism by having the student profiles based on the demographics of the local school system, in this case, Boston Public Schools. This version, by the way, garnered me one of the highest course evaluation scores in all my years of teaching college (average of 3.968 on a 4-point scale, compared to an average of 3.3 for the times I taught this particular course in a more traditional manner).

Promotyping

Before you even begin to build your game, first you should "promotype" (Hirumi and Stapleton 2008, p. 134). Before investing the time and expense of building a prototype, designers will promotype—"test" their idea by floating it by potential consumers to see what people think. Look over your worksheets. One asked you to write a Twitter description of your game, another a title, and a third a two-word phrase to advertise your game. Put those together to develop an elevator pitch. An elevator pitch is a short descriptive promotion that can be said in the length of an elevator ride. Use your elevator pitch to promotype your game. This means pitching your game to your students and asking them what they think about playing such a game in class. This can give you a sense of the likelihood your students will find your game engaging. Maybe a student will even have a good suggestion that you can incorporate into your game. By promotyping before building your game, you can make major changes, or even ditch an idea and start over, before you get too far in the design process.

No Tech

There are several reasons to create a no-technology game. As teachers, we know that if something can go wrong, it will. Designing a "no-tech" version gives us a backup plan in case something does go wrong, such as the Internet going down. A no-tech version also might be most feasible for your first time teaching a curricular game. A no-tech version might be all you need to teach your curricular game. Sometimes, technologies can get in the way! I have a class set of clickers students can use to answer multiple choice questions, but by the time I get them distributed, deal with exchanging ones whose batteries have run out, and explain how to use them, I could have just used a show of hands or have students write their answers on a sheet of paper and hold up their answers. Therefore, I only use them if I have a large class or I have a need to get accurate data. Think carefully about how much technology is truly necessary to run your curricular game. You may find that your no-tech version is sufficient or, in some cases, even superior.

When I talk about creating a game with "no technology," I do not really include old technologies we take for granted in my definition of technology: "Technology is anything that was invented after you were born" (Alan Kay, late 80s). Unless you have students pantomime your game, it is hard to truly design a "no-tech" game as paper and pencil, chalk and chalkboard, and so forth technically are technologies. Technology is really just tools that humans use to accomplish goals. However, some technologies are so "old school" they are rarely considered technology any more. For our purposes, I will use the definition of technology given by flight attendants: "anything with an on/off switch." For our no-technology version of our game, we will design one only using technologies that lack on/off switches.

Combining some "no-tech" technologies can create some interesting affordances. For example, when I taught high school, I discovered that my chalkboard was magnetic. I went to a craft store and bought a roll of sticky magnets, found some sentence strips (thick paper with lines on them), put magnets on the back, and wrote on the sentence strips. These magnetic sentence strips allowed me to randomize elements in a sequence to create some one-shot games where students had to put the sentence strips in the proper order (usually with two teams competing against each other). For example, students had to put together the punctuation and elements in an MLA citation properly. While this one-shot competition does not constitute the type of curricular game you are developing, it is an example of repurposing common items to create a mini-game. Think about what "no-tech" technologies you already have or can create in your classrooms that you could repurpose for your game. For example, a no-tech version of the Odysseus Order sequencing game I described earlier could involve rearranging magnetic sentence strips, each with a different adventure from the *Odyssey*.

CHALLENGE 7.1: *You find yourself staring at the shoe you are using to prop open your bathroom door because you were getting so tired of scanning your finger every time you needed to run into the bathroom to receive a text from Amy. You think about the items in your old classroom you used to "repurpose"—use in a way other than its intended use.* **EXERCISE**: Find an item in your classroom or in your life and "repurpose" it.

Words can be one of the most effective ways of creating realism. One of my students wrote such realistic dialect in her curricular game, and at first I thought she was quoting a book! Another student used real-life Presidential campaign examples in his campaign simulation game to explain the results of each decision. Yet another student had his players get a "phone call" from a gruff police chief by recording his own voice. You supply the framework, "the player, through imagination, supplies whatever else is necessary to complete the 'construction' of the setting" (Swan 2010, p. 115). Remember, that creating a sense of realism does not have to be flashy graphics: "the goal of graphics is not aesthetic excellence or stunning realism, … the goal of graphics is first and foremost, to communicate" (Swan 2010, p. 116). Given enough scaffolding, we conjure up the rest of the mental image in our minds:

> If I tell you a short story: "The mailman stole my car yesterday,' I have actually told you very little, but already you have a picture of what happened. Weirdly, your picture is full of details that I didn't include in my story. Take a look at the mental image that formed, and answer these questions: What did the mailman look like? What kind of neighborhood was my car in when he stole it? What color was the car? What time of day did he steal it? How did he steal it? Why did he steal it? … This ability to automatically fill in gaps is very relevant for game design, for it means that our games don't need to give every detail, and players will be able to fill in the rest. The art comes in knowing what you should show the player, and what you should leave to their imagination. (Schell 2008, pp. 124–125)

Indeed, it is reader's ability to make inferences that allowed Ernest Hemingway to win a contest by writing a short story in six words: "For sale: Baby shoes, never worn."

Although some of the appeal of commercial video games are the graphics, and video game companies spend a lot of money on this, "Research suggests that there is no difference in motivation and learning in either low immersion or high immersion environments (Moreno & Mayer, 2002)" (Green and McNeese 2011, p. 99). Indeed, some of the first "video" games that I played were text-based adventure games where the player would type in commands like "turn left" and the computer would reply, "you are facing a fireplace." One of the "pleasant frustrations" (Gee 2007) of those games was the ability to spatially create the space in your imagination. Schell (2008) points out that "you don't need to perfectly replicate real experiences to make a good game. What you need to do is to capture the essence of those experiences for your game" (Schell 2008, p. 20). Begin by asking "What experience do I want the player to have? What is essential to that experience? How can my game capture that essence?" (Schell 2008, p. 21). For example, when designing Wii Sports, the designers

> realized they wouldn't have time to simulate every aspect of baseball as well as they wanted. So, they made a big decision—since swinging the controller was the most unique part of the game, they would focus all their attention on getting that part of the baseball experience right—what they felt was the essential part. They decided that other details (nine innings, stealing bases, etc.) were not part of the essential experience they were trying to create. (Schell 2008, p. 21)

Jesse Schell (2008) uses an example of finding the essence of a snowball fight by listing elements of a snowball fight and drawing a line through anything that was nonessential—that you could remove and still have it capture that experience (p. 20). Take those essential elements and think about how you can represent them. For example, if you decide that being cold is an essential part of a snowball fight, think about what represents being cold. Perhaps just having characters wearing scarves is enough to signify being cold.

Keep in mind that people can be used in different ways as well. For years I would think, "Wow. Wouldn't it be great to create a video game for my pre-service teachers to play to practice being teachers?" but then I would think about the complexity of designing a game to replicate something as unpredictable as human interaction in a classroom setting. I then realized, why design a video game when I have a class of students every year who can role-play middle and high school students! Similar to having students design their own tests to review for a test to learn how to think like a teacher, by role-playing middle and high school students, my students have reported to me that this has allowed them to better understand their students. One way to add realism is to have students dress up as their role to create human avatars. This does not need to be elaborate; a simple prop such as a Viking hat can conjure up a whole culture. Do not forget, you should dress up as your character as well.

CHALLENGE 7.2: Now that you have realized you do not have to code a video game, you realize you need to practice "no-tech-ing" a video game to get into the mindset. You take a video game and design it as a game without any materials (just verbal/physical), then as a card game, then as a ball game, and then as a board game. For example, verbal Angry Birds *would have the teacher say the height of the tower and the distance between the tower and the birds ("Instructor as game-engine" (Hodgson 2013, p. 57)), and the students would have to state the angle of their slingshot. As a card game, students would draw from the Height card pile and the Distance card pile to determine these variables. As a ball game, students could actually throw a ball to try to knock over the towers. And as a board game, students might have to build a tower in the middle of a checkerboard. Students would roll a die to determine their position on the checkerboard. They would then have to calculate the trajectory.*

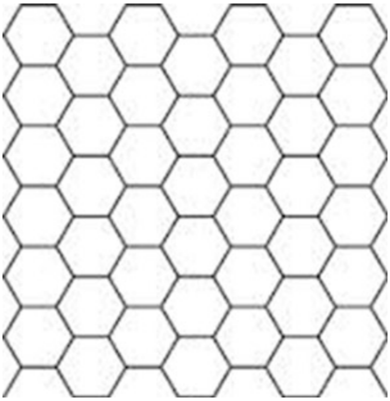
EXERCISE: Think about how you might redesign a video game without any materials, then as a card game, then a ball game, and then as a board game.

Even though your curricular game does not have to be photorealistic, you can use audio and visual details to convey meaning. When I played *MYST*, I was as drawn in by the music as much as I was by the visuals. Just hearing the introduction song playing would get me excited. Think about how often people use the *Jaws* theme to allude to fear and suspense or how compelling the *Star Wars* theme music is. I know, technically playing music requires something with an on/off switch, unless you are willing to sing. However, using a familiar technology to play music can be a powerful way to create mood.

Decorating the classroom is another way to create context. Imagine how excited students would get walking into a classroom decorated as the interior of a haunted mansion. I mentioned earlier putting masking tape on the floor to design the game space. Although admittedly this also involves on/off technology, if you can get a hold of four LCD projectors and four laptops, you could darken the classroom and project four pictures on your four walls to create the sensation of being in a video game and change the setting as play progresses. You could even have decorating/designing the room be the first quest of their mission. For example, in a game about ecology, students would have to decorate room like the ecology the game is enacting. To help determine what details you need, ask yourself: "What must the player understand to play my game? Can some element of imagination help them understand that better? … What details I provide inspire imagination? What details I provide stifle imagination?" (Schell 2008, p. 201). Earlier I described phasing—signaling progression in a game through objects. You can do this by changing the decorations in your classroom or by giving students certain props. In the stolen painting game two of my students designed, they described using props and decorations to transform their classroom into a train station one day, a café the next, and so forth. Imagine how exciting that would be for students to walk through a classroom door and enter a new world. Having a progress bar on the wall would make the game space even more meaningful as each change would clearly signal progression through the game.

Of course there is also good old poster paper. For example, if you wanted to create a Facebook game where students play animals (and/or plants) in an ecosystem, or fictional characters from a novel (or several novels), or historical figures, you can draw a Facebook template on several sheets of poster paper, post them around the room, and have students post their picture, update their status, "friend" each other, and post on each other's "walls." You can then introduce certain events such as a natural disaster or human intrusion that students have to react to in their Facebook statuses. Poster paper is a safe way to tap into students' personal technologies without facing potential consequences of having students go on the Internet. Be aware, though, of how characters are depicted and the messages that might send. In his storyboard, one of my students used animals to depict slave owners. Here is my response: "When I first looked at this strip, I wondered why the slaves were depicted as people and the white people as animals. Someone might read that as you trying to hide/blur the historical fact that slave owners were white people. But then I thought it actually turns slavery on its head as slaves were treated like animals, but it was really the white people who were acting like animals in the ways they treated slaves. If done well in the classroom, this could be used as a teachable moment." Having students design their own characters can also introduce conflicting messages. In the game where my students role-play middle and high school students, I worry about stereotyping; so in class we discuss how to realistically portray people without going overboard. While you do not want to interrupt the flow of the game, if a student says or does something disruptive, do not be afraid to pause game play for a teachable moment.

In her book *Reality is Broken*, Jane McGonigal (2011) describes several no-tech games. One is *Bounce*, where players call someone at least 20 year older than themselves and get points by finding similarities between themselves and their senior stranger. Another is *Cruel to be Kind* where players play a kind of rock, paper, scissors game with kindness acts such as smiling and complimenting others. Salen and Zimmerman (2004) describe a game created by a History teacher (Alfred Leonardi) in 1980 to simulate WWII dogfighting. He did so by creating two books, each page with an illustration of the current action from that player's perspective and the player's choices along the bottom. Both players announce their choice at the same time, and the grid along the bottom identifies which page the players then turn to (p. 425). We have become so wedded to technology that sometimes we forget that we do not need technology to do things[1] like have fun!



However, we can take cues from real video game designers on how to create no-tech games. Believe it or not, video game designers create no-tech versions of games in development to test them out. One game designer (Eric Zimmerman of SiSSYFiGHT 2000) describes his first prototype as being "played with post-it-notes around a conference table" (Fullerton et al 2004, p. 203). Video game designers even create a no-tech version for playtesting real-time war games. They do so by

[1] After 9/11, some felt the need to create "bomb-proof" trashcans but lamented the cost of the technology. Boston's subway system, however, created "bomb-proof" recycling bins by using very flimsy, transparent trash bags, making it impossible to hide a bomb in them since any bomb placed in there would be obvious to any passerby.

using HexPaper as the game board so characters can move in multiple directions, including diagonally. They then use something with a directional like an arrow to represent the players to indicate which way each player is facing. Something moveable like folded index cards is used to represent any barriers like walls or boundaries so they can test out different configurations. Even if the real game will be real time, in order to see how it might play out, the prototype is turn-based with a die, spinner, or turn cards to regulate game play. For initial playtesting, only the essential rules are used such as designating where players start, how turn taking is determined, how many spaces/which direction a player can move on a turn, how actions are determined, how scores are kept, how to know when a win state is achieved, and so forth.

You could use your classroom to create your own HexPaper game space by putting masking tape on the floor or purchasing hexagonal shaped floor tiles that can be mixed and matched. Desks or tables could be used to designate boundaries. Students, of course, are the players with their own directional built in—their faces. The rules can be written out on the board along with keeping track of turn taking. This, however, may not work for your game. Your game might be better built by repurposing an existing board game, say by changing the rules and printing up your own turn cards. You may not even need a game board at all. You could have students close their eyes as you describe a scenario, have students discuss it, and then vote on a decision with a show of hands. How you create your no-tech version is up to you, but keep in mind that your no-tech version is a great way to work out any kinks in your game and can be part of your playtesting process.

PLAYTESTING QUEST: You have designed a no-tech prototype of your game but you are unsure of your next step. You think you might need to test out your no-tech version, but maybe you should wait until you ramp up the technology. You recall Amy dreading the playtesting process but always being glad she did it afterward. You used to tease her, claiming that teaching was your playtesting process. "Used to," you think. "I'm acting like Amy is dead. Well, I guess she probably is in a way since I am in the future, but then again, perhaps I appear dead to her?" Thinking about the logistics of time traveling hurts your head. You remind yourself that Amy is alive in the sense that you can communicate with her, but decide to do so only as a last resort. For now, you need to figure out what to do.

```
IF your no-tech version can be playtested as is, THEN read/do the playtesting
section next.

ELSE

    simplify your no-tech version so it can be playtested and read/do the playtesting
    section next

    OR

    skip to the low-tech version section, read/do that, and then

    RETURN to the playtesting section.
```

Playtesting

Playtesting is when you get various people to try your game to gather feedback in order to revise your game. Although one of my students said playtesting “was my favorite part of the experience, because I actually had the opportunity to see my game in action and talk through it with others”, playtesting can be a painful process. Jesse Schell (2008) admits he hates playtesting—he says it is like sending out an engraved invitation that says, “You are cordially invited to tell me why I suck. Bring a friend—refreshments served” (p. 391). However, the same can be said about teaching as students constantly evaluate you. I have a friend who told me about a religious guru you can pay a lot of money to in order to sit down with him and have him “call you on your shit.” When she told me this, I laughed out loud and said, “I’m a high school teacher. I get paid to be called on my shit every day!” Now, of course, being “called on your shit” is even available online through websites like “Rate my professor.com.”[2] Teaching is vulnerable, and the more nontraditional your teaching is and the more passion you put into it, the more vulnerable you feel. However, the payoff can be much bigger as well.

Not playtesting can be dangerous. When Massachusetts rolled out their initial health-care reform, they playtested their website interface extensively by having state workers role-play different possible applicant situations. On the other hand, when the federal government rolled out their health-care website, very little playtesting was done, resulting in the federal government having to majorly overhaul the website after it had been made live. Diane Ravitch roundly, and justly, criticized the Common Core standards for not being field tested in order to “find out how the standards worked in real classrooms with real teachers and real students” (Strauss 2014). I will not bore you with my long list of pedagogical strategies that worked in my head but failed in the classroom. Fortunately, as teachers, we have built in opportunities to field test every day as we can use formative assessments to adjust our teaching midstream and summative assessments to make changes for the next time. The key, however, is to know when and how to solicit data and what to do with that data. I worked with a teacher who clearly did not do any playtesting in her classroom. When her students complained that her class was boring, her response was, “You think you’re bored. I’ve been teaching this same material for twenty years.” If you ever find yourself saying that, you know it is time to get out of teaching.

[2] If you ever do look yourself up on one of these websites, keep in mind that the unhappy students are usually the only ones motivated enough to post—and they are usually unhappy because they did not do well in your class. To counter this, keep your own “smile file” of positive things students say about your teaching. Teaching is emotionally hard and we all need to look at a “smile file” every once in a while to keep us going.

Gray Box Testing

A black box is a term used to describe something where the inner workings are unknown. My television set is a black box to me. I know how to use it as a user, but I do not understand how it works inside. Sometimes I turn it on and the screen is pink. When that happens, I just turn it off and hope when I turn it back on that it returns to normal. A white box is something where you do understand the inner workings, particularly if you designed it. Gray box testing, then, refers to a designer (white box view) interacting with their product as a user might (black box view) and using that feedback to revise. If you have ever designed a test and then taken it yourself before administering it to your students, you have gray box tested. A new learning management system adopted by my university had a student view for professors, but that student view was only available once the class was live, making gray box testing beforehand impossible. Fortunately, in the wake of multiple complaints, they have since changed that. Gray box testing should be the first step in testing the actual product. Your first goal in playtesting is fixing any flaws. After gray box testing, one of my students posted "my reaction was disbelief at my inability to notice these flaws while I was creating the game." As much as you think everything is perfect, it hardly ever is.

Next is to make sure your game is engendering the types of experiences you want your players to experience. Placing the tasks in a play matrix (Fullerton et al. 2004, p. 209) with one axis being "skill-to-chance" and the other "mental calculation-to-physical dexterity" can help give you a sense of the types of experiences your players might have. Sometimes there are activities that you think will involve skill, but actually are largely determined by chance. In these cases, "the solution might be to change a variable determined by chance into a variable determined by player choice" (Fullerton et al. 2004, p. 210). Two of my students discovered that while they thought "free reign" would require a lot of skill, they found that too much free reign and not enough guidance resulted in students using trial and error, i.e., chance. Once again, balance is key for designing games that teach. Clearly for academic reasons, you want your game to mainly fall into the skill box. As for mental calculation versus physical dexterity, that clearly depends on the type of skills you game is designed to teach.

When you gray box test, "if it feels unnatural, it is" (Dempsey 2010, p. 100). Anything that feels contrived to you will feel really contrived to your players. If so, think about how you can change the story or the game to make the game play feel more natural. Remember our content-swapping test. If part of the game is to answer three questions posed by a monster in order for it to step aside to let you move on to the next level, all you have done is dressed up a quiz as a game. You do not want your game to be "Alex Trebek[3] wearing a mask" (Amtzis 2014). Make sure the learning is in the content, and the context, of your game.

[3] Host of the trivia game show *Jeopardy*.

The next step is to make sure your game is engaging. Over a decade ago, a game designer said "When you add an instructional designer to a game design team, the first thing they do is suck the fun out" (quoted in Prensky 2011, p. 261). By creating a fantastical story where the player is the hero and embedding puzzles and challenges in it, your game will not suck. However, sometimes it is hard to break free of old patterns of didactic teaching. While you play your game, ask yourself "What sucks?" and removing anything that sucks (Prensky 2011). You will know something "sucks" if you find your mind drifting off. Anything that too closely resembles traditional classroom assignments might also suck. To flip around the teacher statement I quoted earlier, if you are bored, you know your students will really be bored.

Of course this is more complicated than simply removing everything that sucks. Often removing one thing means making adjustments elsewhere. As teachers, we know that we cannot teach everything about a subject. An important component of teaching is selecting what to include, which means leaving some things out—and makes teaching, by its very nature, political. I recognize that in today's day and age, teachers feel beholden to cover the curricular content dictated by the state—especially knowing that this content will be on high-stakes tests which are becoming more and more not just high stakes for students but high stakes for schools and for teachers. However, if done well, leaving certain content out can actually motivate students to learn more. As Prensky (2011) puts it, "the engaged student is a lot more likely to find any missing content than the bored student is to remember any content covered in an uninteresting way" (p. 272). Alluding to the heated rivalry between Tesla and Edison or the tragic end to Alan Turing's life might inspire students to find out the details.

Prensky (2011) argues that teachers focus too much on "covering the content" and not enough on "keeping students engaged" (p. 271). A major mistake I make (and still make) as a teacher is to assume that students cannot learn something unless I tell it to them. However, what is closer to the truth is that students probably have not learned something if *they* have not said it, thought it, or done it. The goal of your curricular game should be to get students to say, think, or do, i.e., enact, the material. If you find as a user of your game that the game and/or the teacher does the thinking for you, you need to overhaul your game.

Besides assuming that students learned something just because I said it, another common trap I have fallen into as a teacher is the "father [or, in this case, teacher] knows best" syndrome where I, as the teacher, am the "decider." When in this mode, I decide what content gets covered, how it gets covered, how it gets assessed, and leave very little decision-making up to the students. The little decision-making left is voluntary, i.e., raising hands to answer a teacher-initiated question. This allows students to mentally check out. Because of my fear of the potential emotional damage of "putting students on the spot," I have short-circuited the Kolb Learning Cycle (1984) of decision-action-feedback-reflection by making the only student decisions be whether or not they should answer a teacher-posed question. By only calling on those who decided in the affirmative, I had no sense of whether or not non-volunteers were following the lesson. As Prensky (2011) points out, "we have left decision making out of most of our instruction, letting our students slide by with volunteering decisions (i.e., raising their hand) only when the feel like it. We typically leave the *required* decision making to the testing, where it comes with no feedback at all, or until generally too late to be useful for learning" (pp. 272–273). Games, on the other hand, should be "a series of interesting decisions that lead to a satisfying conclusion" (Sid Meier, designer of *Civilization*, quoted by Prensky 2011, p. 272). As you play your game, think about who, or what, is making the decisions.

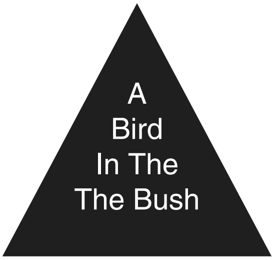
You should play your game several times. At least one time you should play your game as if you are an average student in your class to assess the "usability" of your game. How user-friendly is your game? How intuitive is it? Are the decisions being made and the thinking being done in service of the learning or does the player have to think too much to figure out something that is irrelevant to the learning objectives. Then, you should play your game several other times from different perspectives to adjust for accessibility. Accessibility means making your game playable by a range of players including those with visual, hearing, physical, and cognitive impairments as well as English (or other) Language Learners. If you attended a teacher preparation program, you most likely took classes on accommodating a range of students, commonly called inclusion. For a glimpse into ways the video game industry strives to do this, read *Includification: A practical guide to game accessibility* (Barlet and Spohn 2012). If your game requires players to measure something, how will those with visual impairments accomplish this? Will your students from other cultures know what a yard is? Should you use the metric system for everyone? Or should you allow your students to choose? One of my students described an example of a math word problem on a standardized test using the phrase "tennis racket." He pointed out that even if language learners are allowed to use a dictionary, this is what they might find in their dictionary if they look up the word "racket" (from dictionary.com):

1. A loud noise or clamor, especially of a disturbing or confusing kind; din; uproar: *The traffic made a terrible racket in the street below*
2. Social excitement, gaiety, or dissipation
3. An organized illegal activity, such as bootlegging or the extortion of money from legitimate business people by threat or violence
4. A dishonest scheme, trick, business, activity, etc.: *the latest weight-reducing racket*

None of these even come close to what a tennis racket is. Think about cultural assumptions you might be making. When I was a high school teacher, I designed some unit tests for the district I worked for. When these tests were field tested, teachers pointed out several of the cultural assumptions I had made. For example, asking students to identify which fairy tale the plot of a short story resembled. When testing for accessibility, remember the principle of Universal Design for Learning (UDL): an accommodation for one student often can help other students as well. When you have students playtest, you should also have a range of students, including average students, test it, but you can at least start thinking about how to make your game both usable and accessible at the gray box testing stage by putting yourself in your students' shoes.

Alpha Testing

Why not just end at gray box testing? For the same reasons you get someone else to proofread what you write. Alpha testers are peers you get to test your product before you test it with people from the target demographics, in this case, students. My students were very thankful for the feedback they got from their alpha testers. I am eternally grateful to my alpha tester, my sister, for catching my many mistakes and making innumerable suggestions, most of which I incorporated into this book. One of the major problems with proofreading your own writing or stopping at the gray box stage of testing is that we tend to see what we expect to see. This is called the Stroop effect (Driscoll 2005, p. 85). Look at the statement in the triangle below. Quickly, what does it say?



Now reread it. What does it actually say? We see what we expect to see and, as teachers, we know that students often hear what they want to hear. When we read our own writing or play our own game, we know what we intended and therefore see that:

> Nothing is quite as humbling as being forced to watch in silence as some poor play-tester stumbles around your level for 20 minutes, unable to figure out the "obvious" answer that you now realize is completely arbitrary and impossible to figure out. (Birdwell 1999/2006, p. 219)

As teachers, a dangerous trap we can easily fall into is playing the "guess what the teacher is thinking game." I know a teacher who used to teach by dropping the last word of every sentence and expecting her students to fill in that word.[4] She was literally playing the "guess what the teacher is thinking" game. Right now I have a college essay written by a high school student that underwent several major transformations partially due to feedback I gave her. I would like to incorporate a quest into my English Methods class where students have to read the first version and only get the second version after providing specific feedback and so forth until the hypothetical student "gets into college." I am excited about giving my students practice providing feedback on written essays—a large part of being an English teacher and something I was never taught. However, I am concerned that this will easily turn into a "Guess what I was thinking" puzzle, particularly since feedback on college essays can be so subjective. The solution? I will wait for you to think of it. Yes, of course! Scaffolding! I will design it so that students will get a series of lessons about

[4] Like the economics teacher in the movie *Ferris Bueller's Day Off*.

giving feedback on writing, guidelines, suggestions, and hints (feedback on their feedback!) with each iteration. However, I also do not want students to think there is only one way, my way, to give feedback. Ideally, I would have this puzzle branch where certain feedback would lead to different versions with multiple paths to writing the final essay that gets the student into college. This, of course, will not account for every possible path, but for this type of puzzle, that would be impossible, and at least branching would convey that there are multiple ways to give feedback.

When gray box testing, as you played the game you knew what you were thinking when you designed the game. A common classroom activity in teacher preparation programs is to have one student give instructions about fixing a peanut butter sandwich to another student. Following the exact directions often results in the peanut butter ending up anywhere but between two slices of bread because of the number of assumptions the instructor makes. When someone else tests your game, "We never truly see the output of our work, since it is an experience had by someone else and, ultimately, unsharable. This is why deep listening is so essential for game design" (Schell 2008, p. 11). Not only should you look to their behaviors—Where do they look? What do they do? What is the sequence of their "clickstream" (in a computer world, where they click; in a classroom world, where they take action)?—you should have your playtesters think out loud: "We are bound by believing that to understand the meanings of game play we can simply look at the rules when we, in fact, need to look at players' performance and understand *their* understandings of them" (Salen 2008, p. 15). Have your playtesters verbalize their decision-making, tell you what they are looking at, and report their emotions.

You should instruct your playtesters to be brutally honest including stating when they are bored or identifying things that "suck." However, you need to think about how you can hear critical feedback without breaking into tears:

> Ignore your ego. If you're going to gain anything from a playtesting session, you have to transform yourself. Imagine that you are someone else. You are no longer the designer of this game. Instead, you are an analyst hired to uncover the truth. Your job is not to have these people love the game or you, it's to discover what they don't want to tell you or know how to tell you … [and then] embrace the criticism … remind yourself that you need to hear the problems because you cannot fix the problems if you don't know what they are. (Fullerton et al. 2004, p. 206)

Just like with any assessment where you look for patterns in the data to identify what your students are thinking, look for patterns in your testers' game play to infer what they are experiencing:

> The most common mistake is for the game designer to sit down and begin reading off the rules and describing their vision for the game. … Remember, people learn by playing, not by listening. Let them start playing allow them to make mistakes. … let your testers figure it out. You will tend to learn more the less you speak. … Remind playtesters they are testing the game, not their skills. Any difficulties in playing the game will help you to improve your design … Ask them to talk out loud throughout the game about what they are thinking, questions they may have. Warn them that you won't be able to answer their questions, you just want to know what they are. You are just an observer here. You won't be stepping in to help them, not because you don't want to, but because you need to see where problems exist with the game and how they solve those problems. (Fullerton et al. 2004, p. 201)

You should have all your playtesters think out loud. You could even think out loud while you gray box test and record it so you can capture all your thoughts. However, you should especially have your alpha testers think out loud so their thinking can help shape the expert model you use as the benchmark for assessment. In other words, this testing process provides data you can use to model how an expert approaches the game, how an intermediate approaches the game, and how a novice approaches the game. Since your goal is to move novice thinking toward expert thinking, data from alpha testing can then inform your assessment criteria.

Beta Testing

Why not stop at alpha testing? After all, these are the experts who have been weighing in:

> Testing with friends and family may feel like it works, and it does in the early stages, but it won't suffice once the game matures. The reason is that your friends and family have a personal relationship with you, and this obscures their objectivity. You'll find that most of them are either too harsh or too forgiving. It all depends on how they're used to interacting with you. Even if you believe that your confidants are providing balanced feedback, it's best not to rely too heavily on a small group of individuals. They will never give you the objective, broad criticism that you require to take your design to the next level. (Fullerton et al. 2004, p. 198)

Beta testers are playtesters drawn from the target audience, your students. They may not be your exact students, but should be from about the same age range and skill. As a matter of fact, you may not want a current or a prospective student since your role as teacher involves giving them a grade and so may influence their feedback:

> It is only through the process of inviting total strangers into your office or home and allowing them to fiddle around and criticize your creation that you will gain the fresh perspective and insight you require to improve your design. This is because outsiders have nothing to lose or gain by telling you honestly how they feel. (Fullerton et al. 2004, p. 199)

Beta testing is the most important part of the playtesting process since beta testers represent your audience. For example, a couple of my students discovered that while the word "research" did not bother their alpha testers, it was a complete turn off for their beta testers.

When you solicit and select your beta testers, you want a range of testers to help you balance your game so that it is a challenge for advanced players without being frustrating for beginners, i.e. a low floor and high ceiling. Video game designers use the phrase "tissue testers" to refer to newbies because "like a Kleenex tissue, they can only be used once" (Schell 2008, p. 394) before they are no longer "virgins" to your game. As teachers, we get "tissue testers" constantly—except that they can be tainted by the snot of previous players or even students in classes earlier in the day. On the other hand, by keeping the same testers while you make changes to your game, "You may even find that features which you removed or changed don't work as well, and these testers will be able to point that out. But don't become too dependent on a handful of testers. It's still smart to keep fresh recruits streaming in throughout the process. There's nothing like a pair of virgin eyes" (Fullerton et al. 2004, p. 200). A combination of veteran testers and new testers as you go through your testing cycle can best help you refine your game. Whether new to your game or old-timers, players will crawl all over your game like spiders, finding hidden spaces and things you did not think players could do. It can make you feel very vulnerable, but better to find these things out with playtesters than a class full of students. However, even if you end up being pressed for time and have your first set of students be your beta testers, I have found students to be very forgiving largely because they recognize the effort you are putting into their learning.

When designing puzzles, keep in mind that while the designer "sees" the answer, it is often much harder for others to see it. In the case of students, they might lack the background knowledge needed. Students too, by virtue of the generation gap, come from a different set of experiences. Olson (2007) provides an example of an English teacher who read that the rabid dog in *To Kill a Mockingbird* symbolized the town of Maycomb. He decided to have his students discuss the book and come up with this themselves. This is not even an instance of "guess what the teacher is thinking"; it is "guess what the teacher read about the book and did not even come up with himself." Predictably, he got frustrated when his students did not arrive at the answer. As one of my research participants said, "No student is going to discover the Pythagorean Theorem on their own." On the other hand, we can scaffold student learning to help students figure something out. For example, to help students make the connection between the rabid dog and the town of Maycomb, the teacher could have had students fill out this analogy: "If Maycomb were an animal, it would be ______________ because ______________" (Olson 2007, p. 140). Students might not think of the rabid dog, but it would stretch their thinking and could be the first step in leading them to that answer. On the other hand, sometimes we design something that we think is really hard, and students get it immediately. Yet another reason to playtest!

While your beta testers play your game and think out loud, you want to ask them key questions. However, you do not want to ask them questions constantly as that will distract them from playing your game and may lead them to thinking about things they typically would not: "remember, it's not the number of questions you ask but the quality of the responses" (Fullerton et al. 2004, p. 214). Questions you ask during game play should not direct their thinking, but help make their thinking visible such as "Why did you do that?" Wait until playtesters finish playing before asking questions about overall game play like: "What is the goal of the game? What did you learn? What makes it fun? What detracts from learning? What detracts from fun?"

While playtesting, look for plausibility, being internally complete, a sense of balance, dead ends where players get to an unwinnable game state, loopholes that "allow players to circumvent the intended conflict" (Fullerton et al. 2004, p. 131), confusion where players end up arguing over the rules, a sense of fairness where "all players [have] an equal opportunity to achieve the game goals" (Fullerton et al. 2004, p. 132), and fun and challenging game play. How will you know when these things happen? Will Wright and Katie Salen identify key moments that indicate a well-designed game (Salen 2008, pp. 11–13):

- "Can I try?" indicates not only a willingness but a confidence that one can play.
- "Can I save it?" indicates an investment and sense of ownership.
- "Want me to show you?" indicates a willingness to convey understanding which means a sense of understanding of the rules/system of the game has occurred.
- "How did you do that?" indicates a "community of practice" has developed—a willingness to ask and receive.

Ideally you would have the opportunity to beta test your curricular game multiple times as you engage in the iterative process of soliciting feedback, revising, and soliciting more feedback on that revision: "in the hopes of solving one problem, introduces a host of new problems. … Games are fragile systems, and each element is inextricably linked to the others, so a change in one variable can send disruptive ripples throughout" (Fullerton et al. 2004, p. 7). Because of this, if possible, make only one change at a time between cycles: "If you change two or more variables at once, it becomes difficult to tell what affect each of those changes has on the overall system" (Fullerton et al. 2004, p. 253). Keep in mind that playtesters, particularly veteran ones, do not have to play the game from the beginning every time:

> You can have [playtesters] experience a particular feature/situation/condition by having them start at a "control point"—somewhere in the middle—and play something out. This is one of the reasons that cheat codes exist for electronic games. They are tools that the game developers use so that the team can test controlled situations. (Fullerton et al. 2004, p. 217)

You can also make playtesting cumulative where every time you test, you add a new rule or a new feature. You can also have each playtesting session designed to answer a particular question. However, make sure these questions are not solely from you, but that you also playtest for questions that come from your playtesters.

Schell (2008) calls this iterative process "looping": "The Rule of the Loop: The more times you test and improve your design, the better your game will be" (Schell 2008, p. 80); therefore, "the work is never finished—only abandoned" (Schell 2008, p. 94). As teachers, even though, we have the advantage of constant playtesting since every time we teach a class, we get another group of playtesters, we do want to playtest prior to implementation in a classroom setting to work out the major kinks. However, video game designers warn against changing the game for every playtester's whim. If I changed my teaching based on what every single student wanted, my teaching would be in a constant state of flux since what one student likes another one hates. When we as teachers look at course evaluations or any solicitation for student feedback, we have to look across all of them and dismiss the outliers. Even having a consensus, though, does not mean we should necessarily make a change, otherwise we would be showing movies all day! Keep in mind that often teaching is not appreciated until later:

> Before a game is completely finished, playtesters may reject an unusual idea. They sometimes need to see it completed before they can really appreciate it. If you don't trust your own feelings about what is good and bad, you may, at the advice of your playtesters, throw out an 'ugly duckling' that could have grown up to be a beautiful swan. (Schell 2008, pp. 16–17)

I had a former high school student come to me and say, “I know I was a knucklehead when I was in your class, but what you taught me about writing has been invaluable.” Students may not appreciate your teaching at the time. The ultimate test for your game is not whether or not students like it, but rather whether or not students are learning.

How do you know what feedback to consider and what feedback to throw out? Part of it comes from gut instinct, which you should not discount. After all, your gut is drawing on your experiences teaching *your* content to *your* students. You can also, though, follow the advice of Chris Crawford (2006), who has designed 14 computer games, his first in 1978:

> Most suggestions are additions; some are embellishments, some are corrections, and some are consolidations. The additions are new features; those I dumped instantly. You don’t add new features to a game during playtesting. If the game needs major improvement, then it should be redesigned; if it doesn’t need major improvement, you shouldn’t go adding features this late in the design cycle. Embellishments are improvements on existing elements in the game; these got a few seconds’ consideration. Again, the burden of proof falls on the embellishment; if I can’t see a compelling reason for adding the embellishment, then I don’t want to mess around with it. Corrections fix clumsy aspects of the design; these I relish. My only concern in hearing such suggestions is that they constitute genuine corrections, that they really do fix a problem in the design. If they do, then there's nothing to discuss; it’s a go. Lastly, consolidations are ways of bringing two dissonant aspects of the game into harmony; these I also embrace. It’s rare that a playtester sees some deeper connection that escaped me, but when they do, I grab the idea and run with it as if it were my own. (p. 723)

Use the feedback to adjust your game in ways that support student learning, not to make a cool game for your students. Keep in mind your agenda is for your students to achieve the learning objectives. Your playtesters and students may have a completely different agenda of just having fun.

As you go through these cycles of playtesting, your game should go from a rough prototype where major changes are made to an almost finished game where just the details are tweaked, what Robinett (2006), designer of first graphic adventure game *Adventure*, calls “tuning” your game:

> The length of recoil between biting and swallowing is quite important. If it is too long, it is trivial to avoid being eaten, and players can ignore the dragon and do whatever they want. If the interval is too short, players never succeed in recoiling, and their cursors get eaten every time. There is a middle ground between “trivial” and “impossible” called “challenging.” Trying out the game with various players and watching how well they do is the best way to adjust a game’s timing. … Varying the length of the recoil interval turned out to be an effective means of varying the games difficulty. (Robinett 1984/2006, p. 702)

Posing problem statements about your game allows you to “make every loop count” (Schell 2008, p. 80) by seeing the effects of these subtle adjustments. Because we, as teachers, have the advantage of built-in looping, after playtesting we can continue playtesting by using the no-tech version the first time we teach our curricular game; then building a low-tech version, playtesting, and teaching it; then building a medium-tech version; and so forth.

Low Tech

For your low-tech version of your game, you are going to repurpose a technology tool to branch your narrative. In other words, students will be offered a series of decisions, will choose a decision, and then be directed where to go from there. You can have students do this individually, in partners, in small groups, or as a whole class. I suggest using either small groups or a whole class so that students have to deliberate before making a decision. Having students do it as a whole class allows you to ask guiding questions and hear student thinking. Requiring unanimous decision-making will likely engender even livelier discussion but also could result in a stalemate. Using whole-class decision-making also solves a common problem of not having enough technology in the classroom and of students randomly clicking through your game, getting through by trial and error.

In order to branch your narrative, you will create internal links—links within your product that lead to alternative paths. You can even generate conversations between the player and NPCs by using a dialogue tree where the player chooses from a menu what to say to an NPC, the NPC responds to the player's choice, the player is then given another menu of choices, etc. Go back to the branched narrative you designed in the story building chapter. You should have identified key leverage points and mapped out the results of various choices. In addition to storyboarding, video game designers create wire frames, a blueprint for each type of screen, and then story board the different actions that might take place within that wire frame. When designing your branched narrative, you might want to do the same by creating templates for different scenes or simply copying and pasting or duplicating a slide if you are using a presentation tool. By using the same format or similar formats, not only do you cut down on the time players need to figure out what to do so they can concentrate on the game, it also conveys to the players that they are in the same game, program, or website. This is important generally, but particularly important if you use a lot of external links such as having players do research on the Internet.

CHALLENGE 7.3: Frustrated with your lack of technology skills, you decide it is worth the risk to contact Amy again. You text her to ask how to create links so students can choose to go to different slides in a presentation. Her response infuriates you: "Performance before competence: Open up whatever software tool you use to create presentations. Try to create a link that users can click to take them to another slide, or frame, that is not the next one in the sequence.[5]"

"Does she not realize the pressure I am up against?" you say out loud, not thinking about the fact that your voice might travel outside the bathroom walls and into any "bugs" that might be planted in your room. Once you calm down, you realize

[5] To do this, you need to create at least three slides. Number them 1, 2, and 3. On the first slide, type in "link to 3," highlight it, and go to insert link (or hyperlink) to see if there is an option of linking to another slide in that same document (in PowerPoint, you need to go to "Place in this document"), and select the third slide. To test out the link, it must be in presentation mode and not edit mode (to know if it is in presentation mode, the slide should take up the whole screen). Click the link to see if it takes you to the slide labeled "3."

Amy is just being Amy. Plus, she has no idea about your situation. You know that outside of this context, she would be right. Better to learn it by figuring it out on your own than by slavishly following her instructions. You laugh at yourself as you remember dropping a friend off at her destination and then realizing you had no idea how to get back because you just followed the friend's step-by-step directions for getting there without thinking about where you were going or how to get back.
EXERCISE: Try to figure out how to create an internal link, a link from one slide to another one that is not the next slide, in a presentation.

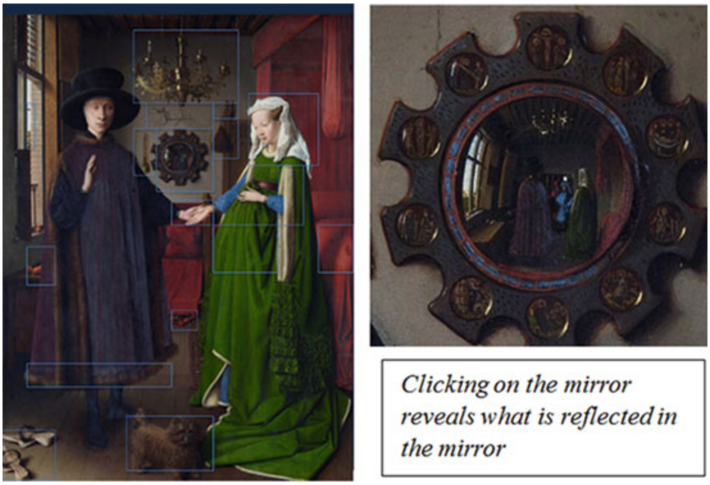
You probably learned a few things by doing the above challenge. First, you may have encountered presentation software that does not allow you to create internal links. If you were unable to create an internal link, search the Internet to see if it was you or the software. If it is the software, find another tool (PowerPoint is one that has this feature). If you did discover your presentation software tool does allow for internal links, you quickly learned that you cannot link to something that does not exist. You will either have to design it completely backward—creating all the end points, the links to those end points, the links to those links, and so forth OR by "swinging" back and forth between the starting point(s), creating the next set of paths, and going back to the starting point to insert the links. You also learned by doing the challenge that once you linked to the other slide, you had no way of getting back to the original slide. This is called "orphaning your user." Think about whether or not you want your players to return to previous points in the game or even a Table of Contents slide that they can use as a home base to explore a number of options. If so, include a return link. Remember, these links do not have to be exponential—you can have links that dead end, multiple links that go to the same path, and so forth. For most presentation tools, internal links will only work when in presentation mode, not when in edit mode.

What I Recommend

Because of their ability to easily create visual and audio effects, as well as their branching capability (for most of them), I recommend using a presentation tool (versus a wiki, a website, or a site designed for designing branched narratives such as Twine). Not all presentation tools have the option of linking to different slides, so make sure your presentation tool does have this option before diving in. Most software tools will change the color of a text link once it has been chosen. This can be useful if you need your users to keep track of where they have been. It will be reset, however, once the document is closed and reopened. Sometimes the color scheme that is automatically chosen makes it hard to distinguish the followed link from the background. If you want to change the color of the links—both followed and not followed—you can customize the color scheme.

Internal links: You can also hyperlink from a picture or a shape. Some programs allow you to create "action buttons" that allow you to do things like ending the slide show, link to other slides, have a sound effect play, and other things both for clicking on the action button or for mousing over the action button. You can even cover a whole slide with an action button, make it transparent,[6] and program it to do what you want—forcing the user to take a certain action. For example, you might have a dead end but have more slides after it that are on another path so you can use a transparent action button to end the slideshow. If you do not do this, your users will encounter the next slide after the “Game Over” one.

You can also use hyperlinked transparent shapes or action buttons to demarcate a section of a picture as clickable. You might do this so your player can click to go to an island on a treasure map, cause an action to happen, or allow the player to take a closer look at something. For example, one of my students pasted a painting on a slide and then had different areas link to other slides with a close-up of that section. This allows the user a closer look at what is reflected in the mirror, the shoes under the chair, the artist’s autograph, and so forth. To do so, she took a screenshot[7] of the painting from a website and then pasted the whole picture linked to slides, selected the crop tool to select the close-up section, and then stretched that section to fill the whole slide. She then went back to the original slide, pasted the painting there, and put action buttons over those sections that linked to the corresponding close-up slides.



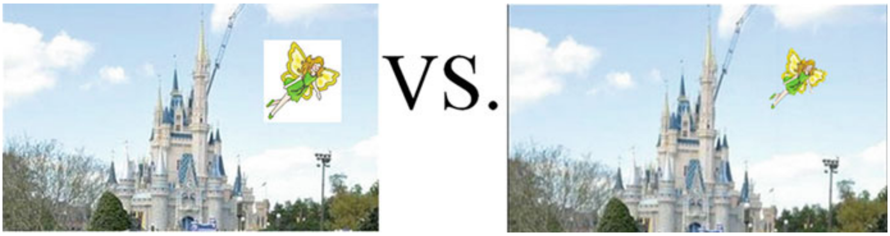
Clicking on the mirror reveals what is reflected in the mirror

[6] Or mostly transparent. When items are completely transparent, some programs think they are not there. I discovered this when I upgraded to a new version of PowerPoint. My old version could make action buttons completely transparent. I could not for the life of me figure out why my transparent action buttons were not working in this new version. I only figured it out after searching online for the answer. In PowerPoint, you can right-click on the action button, select “Format Shape,” and move the transparency slider button.

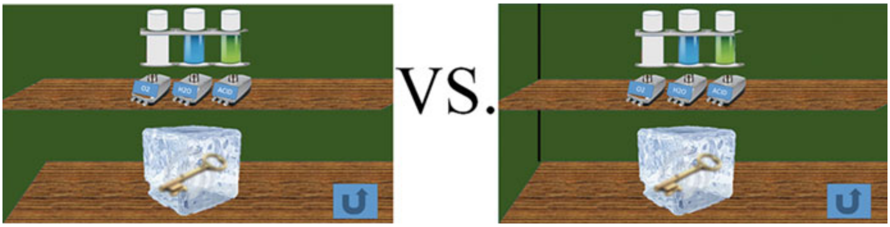
[7] This means to take a picture of what is on the computer screen which then can be pasted elsewhere. Typically you can use CTRL+Print screen on PCs and Command+CTRL+3 for Macs, although my latest laptop uses a different combination of keys, which I had to use Google to figure out.

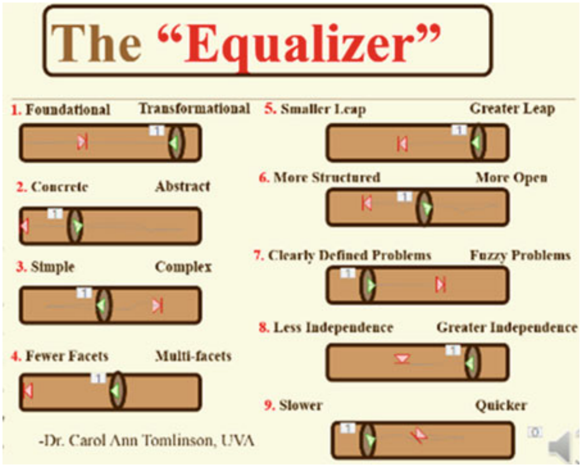
You can also use this technique to create "Easter eggs," or hidden items, by creating clickable areas that are not demarcated for the user. One of my students placed several Easter eggs in a picture of a coral reef so her students could learn about coral reefs by "exploring" one. Clicking on the ocean linked to a slide that revealed the shark lurking in the distance. Clicking on the sea anemone exposed the clownfish hiding within and so forth. Telling her students there are ten Easter eggs (which you might want to wait to do until after they discover the first one) gives her students a goal and allows those achievers to know when they have found everything while also satisfying explorers need for discovery. You do need to make sure the action button, hyperlinked shape, hyperlinked picture, or hyperlinked text is on top of everything by clicking on it and choosing bring to front or bring forward, otherwise you can have a picture or other object blocking the action. This means every time you edit a slide or a picture covered by an action button, you have to move the action button, do your editing, and slide the action button back onto that slide or picture. You can also put action buttons on top of other action buttons. If you do, make sure the smaller action buttons are "on top" by bringing them to the front. This can be one way to mask where the clickable areas are. In the coral reef example, students could discover where the clickable areas are by scrolling over the picture to see when the arrow turns into a hand, what is usually used to indicate something that is clickable. However, if you want to force your users to figure out what they need to click on on their own, you can place a transparent action button over the whole slide and either set it to end the show or to link to itself and then place the other transparent action buttons on top.

Visual effects: There are lots of ways to create visual effects. Different presentation tools have different options so there may be some described here that are not available in the tool you are using. Some visual effects are not special effects, but simply used to make the picture more realistic. For example, when combining images, you can make a color on an inserted picture transparent, in this case making white transparent, so that the background shows through.



Something as simple as a line can create more of a 3-D effect





The "Equalizer"

1. Foundational — Transformational

2. Concrete — Abstract

3. Simple — Complex

4. Fewer Facets — Multi-facets

5. Smaller Leap — Greater Leap

6. More Structured — More Open

7. Clearly Defined Problems — Fuzzy Problems

8. Less Independence — Greater Independence

9. Slower — Quicker

-Dr. Carol Ann Tomlinson, UVA

In some presentation tools, you can use what is commonly called animation to make objects appear and disappear. Having something "disappear" is a great way to have a player click to look under something to make a discovery. Do not forget, you can also hide text and objects by making them the same color as the background or place a shape over something that is the same color as the background that disappears when clicked to create an Easter egg in the same slide. You can also have objects follow certain trajectories. You can even design your own motion path for objects to follow. For example, I wanted to create the effect of sliders being moved on an equalizer so I drew horizontal rectangles and used an oval shape to indicate the slider. I then assigned each oval shape its own path moving back and forth so when I click, all the sliders move back and forth at the same time. You can even assign sound effects to your animations, control the timing, and determine if it happens automatically or after a click.

Although using animation is probably the most common way to create visual effects, you can also use transitions to create certain visual effects. For example, you can have a picture of a set of blinds take up the whole screen and then use the "blinds" transition so it looks like blinds are opening to reveal whatever you have pictured on the next slide such as a person wielding a knife! You can also use transitions to open a set of double doors, make it look like a page is turning, or even like a package of food is being peeled open. Play around with transitions to see what effects you can create.

You can also use a lack of transitions to create certain effects. If you want something to appear, instead of using animations, you can duplicate the slide and put the new item on the second slide. You can even create your own animated cartoon by creating a "flip book"—each slide containing a slight variation on a picture—and insert timing so that it appears as though the item is moving. You can also record narration and sound effects as well as inserting music to make your animation more complete.

Audio effects: Transparent action buttons placed over pictures can create certain effects, including sound effects. For example, I just inserted a picture taken from a recent trip to Disney World, covered it with a transparent action button, and clicked "Play Sound" under the Mouse Over tab so it makes a "cha-ching" sound every time a user scrolls over the Disney picture. Remember, sound is easily mapped to touch, so doing something like having an object that when clicked causes a buzzing sound can give the player the impression of being shocked. Sound effects are a great way to add realism to your curricular game and to convey meaning:

> From environment to interface, games present visible and audible experiences. Recorded effects and voices bring games to life. But it is easy to overlook the importance of strong audio cues in your interface. Audio is that underappreciated tool that gets players through menus and options with a minimum of confusion or need for a manual. Beeps, pops, clicks, and other abstractions communicate to users when their controls have made an appropriate choice such as a selection, or an inappropriate one, like trying to use a deactivated button. Your interface should offer a small vocabulary of audio effects, standardized sounds for things like selections, advancing (start), return (back), and errors in input as well as warnings and confirmations. As a general rule of thumb, if an element of the interface (button, slider, window) produces a visible reaction to player input (pressing, closing), there should be an accompanying sound effect to reinforce the action. (Rabin 2009, p. 105)

The more senses that are involved, the more "real" your game will feel. However, if a sound does not match the visual, this can jar a player out of his or her "suspension of disbelief." Use sounds to reinforce actions, let the player know his or her input has been recorded, to create mood, and to provide hints and guidance. Sound should complement game play, not disrupt it.

Cutscenes: Even though you technically can create a cutscene using the "flip book" method described above, there are several free tools on the web you can use to create a cutscene by making a cartoon. When choosing your cartoon-making tool, beware of ones that fall into the "uncanny valley" realm.[8] The term "uncanny valley" comes from people reporting an increase in empathy as creatures approach human looking but when an animated human is too realistic, there is a dip in empathy that then rises again when viewing real people. When an animated character resembles real human beings too closely, people report feelings of discomfort, revulsion, and even nausea, as it throws off their sense of what is real and what is not: "The cause of this uneasy feeling may be that when we see things that almost

[8] For me, the characters in the movie *The Polar Expr*ess fall into the "uncanny valley" category.

look like people, our brains register them as 'diseased people' who might be dangerous to be around. Zombies are a canonical example of the creepy things that live at the bottom of the uncanny valley" (Schell 2008, p. 328). Regardless of why "almost human" animations jar our senses, you want your players to be challenged by your quests, not by how they regard your computer graphics.[9]

You can insert these cutscenes into your presentation as a video or link to them externally. If your software tool does not have the option of downloading (or you have to pay for that option), try using a free video download application. Also, your presentation tool might have an add-in which will allow you to display a webpage within a presentation slide.

You can also create your own video—either by recording or using a video editing tool which allows you to not only edit video but also use images to create a movie by stringing them together. If you have watched the video *White Teacher* by Candance Doerr-Stevens, you probably felt like you were watching a video. If you watch it again carefully, you will see it is actually a series of pictures, i.e., a slideshow, with the feel of a movie. This is because the editor used the Ken Burns effect of camera movement such as zooming out of a picture makes what is essentially a slide show feel like a movie. In some presentation tools, you can even save a slide as a picture and insert it into your movie. If you do use a video editing tool, make sure you use a version that allows you to record your own narration. Otherwise you'll have to record it in a separate tool, import it in, and sync it with your images.

Animated GIFs: You can also create animated GIFs that you then insert into your presentation. Animated GIFs are graphic images that move. You create them using the flipbook technique I described above—small changes cycled through rapidly to create the illusion of motion. For example, my wife and I took surfing lessons on our honeymoon and; as part of the package, they took rapid fire photos of each of us learning to surf. I put the photos of my wife in order to create an animated GIF so it looked like a little movie. However, I did have to convert JPEG files (JPEGs are great for photos because they code for every pixel and therefore show nuances in colors) into GIFs (GIFs, on the other hand, code for blocks of color, e.g., make a ten by ten block of pixels yellow, and thus take up less file space but also look like graphics and not photos). My students have found several free animated GIF tools online, including one that creates water effects (imagine creating a scene in PowerPoint where your waterfall looks like it is flowing!). To create a series of slightly different GIFs, you can use an image editing tool to make each sequential change to the image. You can also create slides in your presentation tool, duplicate the slide, and make the slight change and then use Save As to save the slides as GIFs. Do not forget to save each one separately and to save them as GIFs. I tend to just number them so I know I am putting them in the right order. You can then insert the finalized GIF into your presentation just as you would a picture.

[9] This advice is a variation on the advice given to me by my dissertation chair, Dr. Marilyn Cochran-Smith, who said, "You want your readers to be challenged by your ideas, not by your writing."

Some examples: Previously, I described creating a game based on the *Odyssey* where the story changes depending on the order of the episodes. To do this in a presentation tool, you could create a map of the islands/locations Odysseus visits and have the players choose the order. Some presentation tools can "remember" the links you visited by color coding followed links; however, none that I know of remember the order in which you visited them. So, if you want order to matter, you would have to have multiple maps for the various paths and, as we know from the section on scaffolding, the locations marked with the order in which Odysseus has visited them. This does create a huge presentation from the designer's perspective, but you can duplicate slides so you do not have to recreate each from scratch. You can also cut down the number of options by chaining, making some episodes have to follow others or by including only some of episodes.

If, however, you are creating something where order does not matter but you still want the map to indicate which locations the player visited by placing Xs over the locations, you can animate the Xs so they appear only after a location on the map has been selected by that picture as the trigger. You can then have that slide link back to the map. When it does, the X will be there since it was triggered by clicking on that location.

I had one student create a curricular game in a PowerPoint where the game appeared to remember where the player had been. I could not figure out how my student did that until I investigated the internal links and discovered he used action buttons to return to "the last viewed slide." Often when you want to do something, if you investigate enough—either by playing around with menu options or searching on the web to see how others did it—you can figure it out. As one of my students said, "reverse engineering[10] [is] a great way to learn how a system works."

CHALLENGE 7.4: *While eating dinner, you suddenly feel your pocket buzz. You look up to make sure nobody noticed. Thank goodness you set your cell phone to vibrate! You sneak off to the bathroom to find an unsolicited text from Amy: "Take a presentation created by someone else that has a lot of special effects and 'reverse engineer' it. Reverse engineering means you take apart something in order to figure out how it works."*

"She thinks I failed the last challenge she gave me and now she's trying to help me out. I'll show her," you think. You realize you might even have one of her presentations on a flash drive in your satchel back from when you were roommates. After dinner you go back to your room and do just that—reverse engineer her presentation. **EXERCISE**: Take a presentation, game, website, video, or any other product created by someone else that uses technology beyond your ability and "reverse engineer" by examining it in edit mode to see if you can figure out what the creator did.

I created a *Can you escape?* type puzzle in a presentation tool by hijacking an idea from one of my students who created a *Can you escape*? game by designing physics puzzles that must be solved in order to escape from each room. I thought I might

[10] Reverse engineering, or "looking under the hood," is different from Backward Design. Reverse engineering starts with an end product that has already been developed. Backward Design is developing something from scratch by imagining what the end product should look like.

create a chemistry version for an upcoming presentation I am doing on game-based teaching. My chemistry is rusty, so the solution to the puzzle is rather simple, but it demonstrates how presentation tools can be used to create puzzles. I first pasted in a picture of the room. Clicking on the lab on the table shows you the lab close-up which has a key frozen in a block of ice and three vials above it labeled “O2,” “H2O,” and “Acid.” Clicking on the O2 vial makes a breeze sound to indicate the oxygen is just released into the air. Clicking on the H2O shows water being poured onto the ice but nothing happens. Clicking on the acid makes the vial pour acid onto the block of ice and disintegrating the block of ice and the key. The trick is that the player has to realize that the objects with the labels are actually Bunsen burners. Turning on the Bunsen burner below the oxygen and the acid causes an explosion, but turning on the Bunsen burner below the water causes bubbles to float around along with a blue flame that dances. Only then, when you click on the water does the hot water pour over the block of ice, melting the ice but not the key. Using motion paths, action buttons, and animations, I was able to do all this in PowerPoint.

Presentation tools can be used to do much more than present. By taking advantage of internal links, visual, and audio effects, you can create a powerful curricular game with a lot of player interaction in ways that immerse players in the game play.

Other software tools you can use to branch:

Word processors: A surprising number of software tools also have internal links that will allow you to branch your story. Even word processing programs have what are known as “bookmarks” that allow users to click on a link that will take them to a specified section of the document.[11] However, I would caution against using a word processing program as users are very used to simply scrolling down which would make all options visible. A tool that is a collection of discrete pages as opposed to one long document of pages allows you to control what users see based on the links a user clicks. Of course you could create a whole bunch of individual word document files, but if you move them around at all—say from one computer to another—the link will be pointing to somewhere that the document used to be and no longer is. Using cloud storage can help take care of this, but there are better tools out there for branching.

Survey tools: Commonly survey, quiz, and form software tools allow for branching. For example, you could write your narrative in the prompt area, select multiple choice for the question type, and select “Go to page based on answer” or whatever the branching prompt is. In order to do this, you would have to create multiple pages. While you can insert images and even video into a lot of survey tools, they do not offer the most options in terms of visuals and user interaction.

Wikis: Using a website or a wiki is another option. A common convention in websites is that anything underlined is a link. To make your wiki or website more

[11] For example, if you had a long list of alphabetized items, you might want to have the alphabet across the top of the document so users can link directly to the section of items that begin with that letter. You have to insert the bookmark first, name it, and then go back to where the decision is made to insert a hyperlink to that bookmark (in Word you go to “Place in this document”).

user-friendly, only underline links. Another common convention is to have links to another page in your wiki or website open up in the same window, while links to external websites open up in a new window. Websites have two types of internal links—ones that go to a specific spot on a page and ones that go to another page. The ones that go to a specific spot on the page are analogous to Bookmarks in word documents except in web lingo, and these are called “Named Anchors.” More commonly, you would be linking to other webpages. In a wiki or a WYSIWYG (What You See Is What You Get) HTML[12] editor, you would use the toolbar to insert a link to another page within that wiki. Do not forget, you have to create that other page first! If you are writing out the HTML, you would use the tag <a href=“link”> text on the page </a>. If the “linked to” page is in the same folder as the “linked from” page, you just need to name the file. If it is in another folder, you would have to indicate the path using a backslash for a subfolder and double dots to go up a level. If you are linking to an external site, add target=“blank” as an attribute to the “a href” tag so it opens in a new window: <a href=“link” target=“blank”> text on the page </a>. “Why is she talking about programming in the low-tech section?” you are probably asking yourself. It is because HTML, while it can get complicated, at its core is an easy language to learn.

CHALLENGE 7.5: *“Another text from Amy! She is getting way too into this. All right. What does she want me to do now?” You read the following text: “Open up the text editor on your computer*[13] *and type the following:*

```
<HTML>
<head>
<title> This is the title of my webpage</title>
</head>
<body>
<center> <strong> This book rocks!</strong> </center> </br>
</body>
</HTML>
```

Now look for patterns.”

“Really?” you think. You remember one of the issues you had with Amy as a roommate was how bossy she was. Still, she was usually right. Sighing, you do what she says. You notice that this has a “head” and a “body.” You get yet another text:

[12] HTML stands for HyperText Markup Language. Markup languages are what publishers use to indicate how a page in a manuscript should look when printed—for example, indicating how much the text should be indented. HTML tells web browsers how to render a webpage. However, each browser translates the HTML instructions slightly differently which is why webpages will look different in different browsers.

[13] Do not use a word processing program—it embeds its own meta-language behind what you write—instead for PCs, most likely you will have Notepad or Wordpad; for Macs, you’ll likely have TextEdit although you may have to change some settings. Things change quickly so you may need to look this up on the Internet for the settings for your Mac, but you can try this: under “New Document” tab, select Plain Text for format and deselect wrap text; under “Open and Save,” select “ignore rich text comments” and deselect “Add .txt extension for Plain Text files.”

"Go to cnn.com. Right-click and choose 'View Page Source.' You will see the HTML for that page. In the head, you will see a description of CNN. That description does not show up on the page itself. The head contains the 'meta data'—data about the webpage that does not show up on the page. You will also see a list of keywords. Search engines used to rely on those keywords so web designers would choose them very carefully. Now, however, search engines also search the content of webpages. The body contains the instructions for how the content of that page should show up on your computer screen."

You do what she says, discovering that CNN still exists in the future and HTML is still used. That was informative, but that still doesn't explain how the HTML instructs the browser to display the page. On closer inspection, you notice that most of the text is encased in angle brackets. Again, another text from Amy! It is like she is reading your mind, or really, anticipating what patterns you will see: "The angle brackets designate 'tags.' Like stage directions, tags tell the browser what to do, but the words in the tags do not actually appear on the screen. Most tags have follow-up tags, indicated by a back slash. Now save your document as a text file but in the filename, replace the file extension .txt with .html. Now click on the filename. It should open up in your default browser. Notice that the text that is not enclosed by angle brackets is what actually shows up on the webpage. Now go to a website that has a basic list of HTML tags and play around with your HTML code. See what happens. Create a second page and try linking to it from your first page."

After following her instructions, you think, "Wow!! That was crazy. I am ashamed to admit that I had absolutely no idea how websites were coded. I can't believe that worked!"[14]

One more text from Amy: "Now pat yourself on the back! You just coded your first website!" You smile as you recall how sweet praise from Amy could be, especially because she was so stingy with it, although not as stingy as the Commander, apparently. **EXERCISE**: Go to www.cnn.com and view the source code. Then, open notepad (PC) or textedit (Mac) and type in the HTML code from this challenge. When you save it, change the extension (letters after the dot) with html. Then, close the file and open it again. The words "This book rocks!" should appear in your default browser. Find a website that lists common HTML tags and what they do and play around with the original HTML by changing the background color, the text, inserting images, etc. Don't forget to refresh the webpage each time to see the changes. Then, create a website by creating a second page and linking to the first using <a href="filename.html"> text</a> (with filename being the name of the first webpage). Test your link to make sure it works.[15]

[14] This quote is from a discussion post by one of my students after doing this challenge.

[15] Don't worry. Your new website is not visible to everyone on the web. It is just on your computer. However, if you want to make it visible to others, you can always put the .html files in cloud-based storage and provide others with the link. Be sure to store everything in the same folder, including any images you want in your website, because those links are "relative," meaning they depend on where you start (like in these directions: "walk three blocks, turn left, and it is the second house on the right"—where you end up depends on where you begin), and not absolute (as in 36 Elm St. Smithville, NY which is the same spot no matter where you start, just like a URL).

If you want players to interact with each other online, I suggest you use a wiki. Wiki is Hawaiian for fast because users can quickly upload content. Not only can users edit the page itself, a lot of wikis have the ability for users to comment on a page or even create a backchannel discussion. Wikis are also designed, or at least should be designed, so it is easy for people to create websites. However, this interactivity can create a problem if your students edit something you do not want to be edited. Do not worry, most wikis keep track of the history of the document, allowing you to recover any previous version, but you do not want to have to do this if you do not have to. Wikis often have the option of "locking" certain pages so only the designer can edit it, but sometimes you have to pay money for this option. If you do decide to pay money for any software, look to see if there is an educator discount.

You may also be able to design templates, such as a Facebook template, for your students to use in a wiki. Depending on the wiki site you use, this template may or may not be locked. Wikis are a great alternative for designing social media games if you do not, or cannot, use social media in your teaching. While using social media taps into literacies most students are already using, it also comes fraught with problems ranging from schools restricting access; to opening yourself up to students see something age-inappropriate on the site or students posting something inappropriate; to certain sites, including Facebook, limiting access to people over 13 years old due to COPPA (Children's Online Privacy Protection Act); to sites, including Facebook, having rules about only creating accounts for real people (thus limiting having students creating accounts as Thomas Jefferson and so forth); and to subverting parental rules limiting their child's social media use. Because of these reasons, wikis can be a safe alternative, particularly if you password protect it or at least make it unlisted, i.e., so it is not searchable on the web. You can also use what I call "closed circuit" educational social media sites. These are sites designed specifically for educational purposes where students can only interact with other students in their class. Thanks to the TEACH (Technology, Education and Copyright Harmonization) Act, one advantage of these sites is that because they are password restricted, you can post copyrighted material as long as it is for educational use, you can use only the amount necessary (and not a whole book!), and students only have access to it while they are in your class.

Video: You can also create a completely video-based branched narrative by using YouTube's video editing tool. There may be others, but this is the only one I have found that allows users to insert hyperlinks directly onto a video. First, create the all the videos for all the paths on your flowcharted storyboard. Then, upload them into YouTube. Under the privacy settings, make them unlisted. Then, go to the first branch and copy the URL to that video. Go to the starting video and when you get to the decision point (either the end or if one of the decisions allows the video to continue playing wherever that decision point is), insert an annotation. In that annotation, put the text of that decision point and click the "link" box. You can then insert the URL of the video that decision links to. You may also want to insert a pause, although you can also not pause the video in order to give the player only a certain amount of time to make a decision.

Spreadsheets: You can also use a spreadsheet to "branch" your narrative. In the medium-tech section, we will get into ways you can use a spreadsheet to do some calculations for you, but for now, let's see what we can do just using some basic functionalities of a spreadsheet. One nice thing about a spreadsheet is that you can create a board game effect by coloring the cells that trace the path(s) and then turning off the gridlines (usually just by clicking Gridlines under View). Unlike a page in a word document, a wiki page, or a presentation slide, spreadsheets allow an (almost) unlimited canvas horizontally and vertically. However, you could create something similar to a children's book I loved as a child where users follow a path which branches at the end of a page (in a spreadsheet, where you determine that is—although if you end up printing it out which is always a good idea in case you have technology problems, you may want to use Page Break view to do it), and then users click on their decision which takes them to another sheet or, if you want users to see the alternative paths, both (or all) could take them to the same sheet, but different cells with new paths that branch. You can also use a spreadsheet to create a circular or rectangular path like *Monopoly* or *Sorry!* or a linear path like the game of *Life*. Paths do not just have to be a series of cells filled in; you can also insert pictures so your path has bridges, stops along the way, and so forth. You could program your spreadsheet to generate random numbers in a certain range to simulate rolling a die, OR you could just have players roll an actual die!

Scaffolding

One of my students discovered that the predominant feedback he got from playtesting was the need for more scaffolding. In some programs, mousing over external or internal links produces a comment box. If your software has this option, when you create a link, you will likely see a button that allows you to type in this mouseover text (sometimes called a "screen tip"). This is a handy way to insert some scaffolding into your curricular game, especially because it allows the user to choose which hint to see. If you want a balloon with text to pop up only when a user scrolls over something but you do not actually want to link anywhere, you can create a bookmark and then put the link to that bookmark in the same location as a workaround. Another way to control the amount of scaffolding shown is to use a selection pane which has a picture of an eye either open or closed that you can use to make items visible or not. You can even group items together and name those groups in the selection pane so the user can choose what to see. For example, you might group all the vocabulary definitions together and name them vocab. The user could then click the eye to make the vocabulary scaffolding visible or not. You can do something similar with comments and use keywords in place of the reviewers' names. However, you would need to include directions so students know to and how to do this. Just as we are "repurposing" software tools designed to do other things in order to make our game, there are lots of functions within those software tools we can "repurpose" to do what we want. Often, it just takes playing around to see what is possible.

CHALLENGE 7.6: *"Enough!" you think, as you feel your phone buzz once again: "Take a software tool you are relatively familiar with. Go through all the tabs, menus, and toolbars. If you see something and you do not know what it does, click on it to find out." Your one question has led to a flood of directives from Amy. You feel like she has wanted to teach you these things all along and was just waiting for an invitation to do so.* **EXERCISE**: Explore the menus in a software tool you think you are very familiar with. See if you can discover anything new.

Hybridization

Perhaps you want both special effects and interaction among players. A presentation tool is best for creating special effects, but wikis allow for communication among players. You can always link to a wiki from your presentation. It would be even better if you had an add-on so the live wiki page is displayed within a slide. Don't forget, not only can you combine software tools, for example, landing on a certain cell may link to a cutscene video or even a puzzle, you can also use no-tech elements as well. If your game uses turn cards, you can write them out on index cards OR you can use a flashcard quiz site to create turn cards that will be chosen at random. Blending no-tech and low-tech options can help create a powerful curricular game. As one of my students said, "low tech does not mean low quality."

Medium Tech

In our medium-tech version, you will do some "mini-programming" in order to perform calculations. Remember when the Commander said she cannot play a word document? This is not true. You can create mini-programs in word processing programs and other software tools so that one action sets off a series of other actions. These mini- or microprograms are, ironically, called "macros." To do so, you record a macro by assigning shortcut keys (you might have to try different combinations to find ones that are unassigned) and then perform the actions you want the program to do automatically. Press stop recording when you are done. The program will "record" these actions and "play" them automatically when that shortcut key is typed. For example, you could write the first part of your game story and then instruct players to type one shortcut key or another depending on which decision they make. The program will then execute the series of actions you programmed it to based on the shortcut key pressed. Of course it might be easier to use a presentation tool with links to different slides that are animated. Remember to follow the technology version of KISS (Keep It Simple Stupid)—use the simplest technology you can use to create the effects you want.

What I Recommend

Spreadsheet programs can also be used to "program" your game. Some even connect to the Internet to provide real-time data. You can have players input data, have the spreadsheet perform calculations on these data, and produce a result. This might be particularly useful for puzzles within your game, but also for simulations of systems where players can test their hypotheses by adjusting one variable to see the impact on the whole system. For example, perhaps you have a game where players have to decide whether to buy a house or rent an apartment. Players could then adjust the salary input to see how much money they would need to earn to afford a house payment including, of course, how long it would take to save for the down payment. This can be made even more complicated by having players choose geographic location as well as other expenses. You could even have players choose different occupations and have salary determined by average starting salary for that occupation. The big advantage of spreadsheets is that you can do much more complex and nuanced calculations based on student input than branching does. You can even "carry the work forward" by using Paste Special to paste formulas onto the next worksheet so if a change is made on a previous worksheet, it then changes the subsequent worksheets. You can also have your calculations done in hidden cells or sheets so your players only see the interface.

Although spreadsheets are known as data crunchers, data are not necessarily numbers. For example, foreign language teachers could use if-then-else statements to compare student input with the expert model to see if students translated or said something correctly.

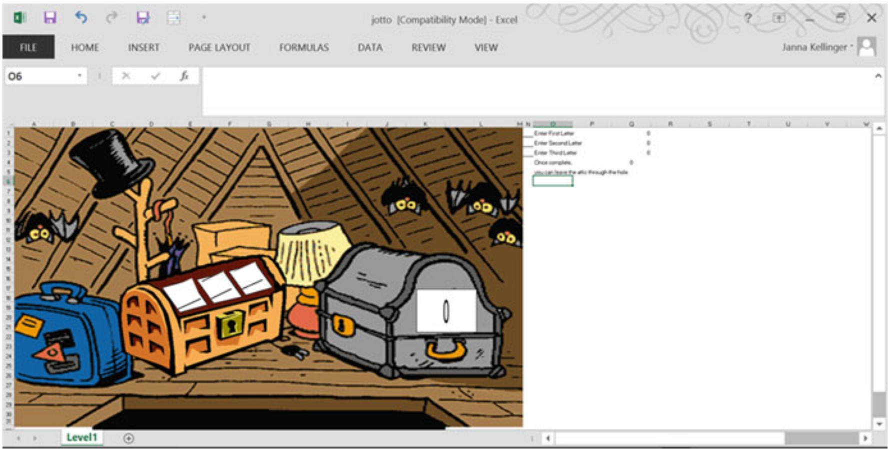
Ex: =IF (A2="Bonjour tout le monde", "Bonjour étudiant", "Je ne comprends pas")

If I remember my French correctly, this statement says: If the player typed "Bonjour tout le monde" (Hello, world) into cell A2, then print "Bonjour étudiant" (Hello, student) in this cell, otherwise, print "Je ne comprends pas" (I don't understand). Keep in mind that every spreadsheet has its own conventions, so you might have to do some online searching or just plain old trial and error to see exactly what format your program requires. I was really getting frustrated one time when working in a database because it seemed to be random which statements it would accept and which it rejected. Because the box in the database program was so small, I had been writing some statements in a word processing program and others right into the database. I finally figured out that the word processing program used curly quotation marks and the database required straight quotation marks. The vast majority of the time if you are having a problem, someone else has had that same problem and you can find the solution—or sometimes even find out that the program does not do what you want it to do—on the web.

You could even use a spreadsheet to create a text adventure game. Perhaps when a player encounters a door, the player can only open that door if the player has first found the key. You could program the spreadsheet to check the player's inventory for the key which is only stored there if the player moved aside a rock previously. First you would need to program the spreadsheet so that when the player typed “move rock,” another cell in the inventory (which can be a separate worksheet named inventory so the player can check to see what she has stored) outputs “key” (IF (B6 = “move rock”, “key”, “”). When the player gets to the door, the door is only opened (IF (Inventory!A3 = “key”, “open”, “closed”). You can even create more complicated if-then statements by using Boolean operators (AND, OR, NOT) and nested if-then statements which are if-then statements within if-then statements. For example, you might want to say IF (C8 = “move rock” AND Strength > 20, “cave entrance open”, (IF (C8 = “move rock” AND Strength >10, “cave entrance partially blocked”, (IF (C8 = “move rock”, “cave blocked”, “”))).

CHALLENGE 7.7: *As your phone buzzes again, you find yourself hoping this is the last time Amy texts: “Program a spreadsheet to play Jotto where players try to guess a target word and the spreadsheet indicates how many letters in their guessed word match the letters in the target word.” Although you are eager to get back to designing your game, once again you comply with Amy’s directive partially because of the hold she has over you and because her exercises have been so enlightening.* **EXERCISE**: Use a spreadsheet to design the game Jotto by choosing a secret target word and then comparing the words a player inputs to the target word, outputting the number of letters that are the same. Be sure to program a congratulatory message when the player figures out the target word.

So far I have been talking as if spreadsheets have little to no visuals. However, as mentioned at the end of the low-tech section, you can insert pictures into most spreadsheets and create clickable sections by inserting transparent shapes and hyperlinking them. For example, I used Excel to create a Jotto game with three different levels. The first level was a picture of an attic with a hole at the bottom for players to exit the attic. However, clicking on the hole did not link to the next level unless the Jotto puzzle was solved.



Enter First Letter 0
Enter Second Letter 0
Enter Third Letter 0
Once complete, 0
you can leave the attic through the hole

The typed letters were displayed on one treasure chest and the number of letters that matched the target word on another using the camera tool. Once the three-letter Jotto puzzle is solved and the player clicks on the exit, the player is taken to the next level (a separate Excel file) which is a bedroom which requires guessing a five-letter target word to be able to leave through the door. That door takes you to a third level (another Excel file) with a picture of a living room. Players have to guess a seven-letter target word in order to escape from the house by leaving through the front door.

High Tech

There are a lot of free and not so free gaming software programs out there that range from ones that do the programming for you to ones that are programming intensive. There are several game creation tools such as *Scratch* by MIT that utilize drag-and-drop software programming. Before you get involved in programming a game, however, you should create an "ugly" software prototype that is text only so you can do the "tuning" before the programming. This can be as simple as creating a spreadsheet to do the calculations or drawing out a flowchart. Below are some suggestions for building an "ugly" prototype from Nikita Mikros, lead programmer of *I-Spy Challenger:*

1. "Everything is a variable."
2. "Try to avoid any literal constants in your code, in other words a code snippet that looks like this: totalOutput = 15*2 should look like this: totalOutput = rateOfProduction*numFactories."
3. "Expose as many variables in the interface as possible."
4. "Litter your prototyping tool with editable text fields; any value that has a remote possibility of changing should be editable through these fields. Your tool will be as ugly to look at as your high school yearbook picture, but you'll be happy when you don't need to recompile or go rifling through your code looking for a variable in the middle of a playtesting session."
5. "Don't even think of reusing this code" (Fullerton et al. 2004, pp. 167–168).

When programming the game, think about how the player will interact with the game: "the goal of an interface is to make players feel in control of their experience" (Schell 2008, p. 222). The goal of the game is for the players to learn the content and skills, not how to use a controller or interact with computer, so "Keep It Simple Stupid" (KISS). Crawford (2006) calls an interface that is "simple" yet allows players to be "expressive" (Swan 2010, p. 116), "elegant controls":

> A qualitative criteria for elegant controls is that using the controls should recede into the background of the player's consciousness, while player action in the game moves into the foreground (Rollings & Adams, 2003). … Ideally, using the controls should become like walking, or riding a bike; they can be done without consciously thinking about them, freeing up the conscious mind to deal with other matters—like winning the game. (Swan 2010, p. 117)

Do not get too caught up in trying to emulate commercial video games. Those designers work with a team. Often, each member of the team has their own team of specialists. It is worth reiterating what I said in Chap. 2 that Squire (2011) found that the “floor” was the point of comparison:

> Kids compared [*Supercharged!*—the educational video game Squire helped create] to ‘what they did at school’ rather than ‘the games they played at home’. We saw no evidence of kids rejecting *Supercharged!* Because it wasn’t *Grand Theft Auto*. There was not one complaint about the graphics or lack of violent content. We presented *Supercharged!* As a game, and students played it. (Squire 2011, p. 96)

However, Squire and his team did find that:

> These kids *were* critical of bad design. … As kids grow up awash in software, their expectations evolve. Twenty years ago, when I was a kid, the computer was so interesting it really didn’t matter what we did with it. We were happy just to be on the computer. Now, almost every kid has access to an iPod touch, gaming console, and personal computer. They are sophisticated consumers who expect good design. (Squire 2011, p. 96)

What I did not include earlier were some of the criticisms students had of Squire’s game:

> Poorly arranged levels that didn’t match the ship’s controls (e.g., levels that were too big or too small) were criticized. Likewise, when the pacing was off—when new levels did not introduce new challenges or challenges graduated too quickly, students tuned out. Finally, sometimes the collision detection clipped or players got stuck near a wall, which was deadly for engagement. … Kids don’t expect educational games to be *Grand Theft Auto*, but they *do* expect good design. This means clear, compelling objectives; intuitive controls; clean interfaces; aesthetically pleasing worlds; and difficulty curves that ramp well. (Squire 2011, p. 96)

As you read through these criticisms, you will see that these are the type of design issues that will bubble up during playtesting.

Conclusion

Game designers sometimes get stuck thinking there is only one way to render their game. This can be extremely frustrating to novice game designers who think they lack the skills to make their game playable. I hope this chapter has shown you that not only are there multiple ways to create your game, you can also repurpose and combine a number of tools, including ones without on/off switches, to create your game. I also hope this chapter has taught you new features of familiar software tools and introduced you to some new software tools. Most of all, though, I hope this chapter has encouraged you to take risks with technology. Just like I wrote about replicating your system in a spreadsheet so players can “play” with the variables to see what happens, technology is a system where you can “play” with various menu options and features to accomplish your goals. Lastly, I hope this chapter has impressed upon you the importance of playtesting. It can feel very vulnerable, but like my alpha tester sister said of her husband who is a gaming aficionado and therefore someone I am a little intimidated to show a rough draft of this book to, he is going to read this book someday, it might as well be when I can take action on his suggestions and criticisms.

> *Design is really an act of communication, which means having a deep understanding of the person with whom the designer is communicating.*
>
> -Donald Norman, former VP of Apple’s Computer’s Advanced Technology Group, 2002

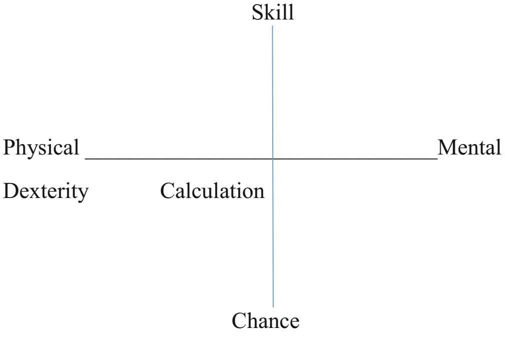
Appendix

RENDER QUEST WORKSHEET

Title: ____________________

Elevator Pitch: ____

Promotyping	Feedback	
	Revisions	
No Tech description		

Graybox testing	Play Matrix—place the different tasks in your game in the appropriate quadrant	 Skill Physical Mental Dexterity Calculation Chance
	What feels unnatural?	
	What sucks?	
	Revisions made	
Alpha testing	Description of expert thinking	
	Feedback	
	Revisions	
Beta testing	Description of novice student thinking	
	Description of intermediate student thinking	
	Description of advanced student thinking	
	Feedback	
	Revisions	
Low tech version	Link or attach	
Medium tech version	Link or attach	
High tech version	Link	

Suggested Curricular Game Rubric

Criteria	"Wow! I mean, I think this might work" (6)	"Hmm, this might be acceptable" (4)	"I need more convincing" (2)	"Go back to the drawing board" (0)
Compelling storyline	Multiple compelling storylines engage players, use second person, and are doled out to create suspense and just-in-time learning	Story furthers pedagogical goals and encompasses quests	Little to no connection between story and pedagogical goals	Lack of a storyline or storyline unclear or confusing
Engaging quests	Quests require critical thinking	Quests at the comprehension level	Quests at the knowledge level	Little to no quests OR core game mechanics of quests does not match learning objectives
Supportive scaffolding	Tiered scaffolding comes just in time	Student chooses type of scaffolding	Scaffolding supports learning	Little to no scaffolding
Making thinking visible/ audible	Game is designed so students have to constantly make thinking visible and/ or audible	Multiple prompts for students to critically reflect on their decision-making throughout	Opportunity for students to reflect on their decision-making at the end	No opportunities for students to reflect on their decision-making
Playtesting	Description of how results of gray box, alpha, and beta testing used to revise product	Three levels of testing described	Only two levels of testing described	Only one level of testing described
Techie (2 extra points)				
Both low-tech and medium-tech versions turned in				
Tech savvy (4 extra points)				
High tech—uses drag-and-drop game creation software like Scratch				
Tech guru (6 extra points)				
Even higher tech—codes own game				

Suggested Reading

Fullerton, T. (2008). *Game design workshop: A playcentric approach to creating innovative games*. Burlington, MA: Elsevier.

McGonigal, J. (2011b). *Reality is broken: Why games make us better and how they can change the world*. London: Penguin Books.

Rushkoff, D. (2011). *Program or be programmed: Ten commands for a digital age*. Berkeley, CA: Soft Skull Press.

Schell, J. (2008b). *The art of game design: A book of lenses*. Burlington, MA: Morgan Kaufmann.

References

Akilli, G. (2007). Games and simulations: A new approach in education? In D. Gibson, C. Aldrich, & M. Prensky (Eds.), *Games and simulations in online learning: Research and development frameworks* (pp. 1–20). Hershey, PA: Information Science Publishing.

Amtzis, A. (4/2/2014). Personal communication.

Barlet, M., & Spohn, S. (2012). *Includification: A practical guide to game accessibility*. Charles Town, WV: Ablegamers Foundation.

Birdwell, K. (1999/2006). The Cabal: Valve's design process for creating Half-Life. In K. Salen & E. Zimmerman (Eds.), *The game designer reader: A rules of play anthology* (pp. 212–225). Cambridge, MA: MIT Press.

Bogost, I. (2007). *Persuasive games: The expressive power of videogames*. Cambridge, MA: MIT Press.

Crawford, C. (1982/2006). Eastern front (1941). In K. Salen & E. Zimmerman (Eds.), *The game designer reader: A rules of play anthology* (pp. 714–724). Cambridge, MA: MIT Press.

Dempsey, J. V. (2010). Elemental learning and the pyramid of fidelity. In R. Van Eck (Ed.), *Gaming and cognition: Theories and practice from the learning sciences* (pp. 82–107). Hershey, PA: Information Science Reference.

Driscoll, M. (2005). *Psychology of learning for instruction*. Boston, MA: Allyn and Bacon.

Fullerton, T., Swain, C., & Hoffman, S. (2004). *Game design workshop: Designing, prototyping, and playtesting games*. San Francisco, CA: CMP Books.

Gee, J. P. (2007). *Good videogames + good learning: Collected essays on videogames, learning, and literacy*. New York: Peter Lang.

Green, M. E., & McNeese, M. N. (2011). Using digital games and virtual environments to enhance learning. In M. S. Khine (Ed.), *Learning to play: Exploring the future of education with video games* (pp. 79–105). New York: Peter Lang.

Hirumi, A., & Stapleton, C. (2008). Applying pedagogy during game development to enhance game-based learning. In C. T. Miller (Ed.), *Games: Purpose and potential in education* (pp. 127–162). New York: Springer.

Hodgson, J. (2013). Developing and extending gaming pedagogy: Designing a course as game. In R. Colby, M. S. S. Johnson, & R. Shultz Colby (Eds.), *Rhetoric/composition/play through video games* (pp. 45–62). New York: Palgrave.

Kolb, D. (1984). *Experiential learning: Experience as the source of learning and development*. Upper Saddle River, NJ: Prentice Hall.

Malone, T., & Lepper, M. (1987). Making learning fun: A taxonomy of intrinsic motivation. In R. Snow & M. Farr (Eds.), *Aptitude, learning, and instruction* (Conative and affective process analyses, Vol. 3). Hillsdale, NJ: Lawrence Erlbaum.

McGonigal, J. (2011a). *Reality is broken: Why games make us better and how they can change the world*. London: Penguin Books.

Olson, C. B. (2007). *Reading/writing connection*. Upper Saddle River, NJ: Pearson.

Prensky, M. (2011). Comments on research comparing games to other instructional methods. In S. Tobias & J. D. Fletcher (Eds.), *Computer games and instruction* (pp. 251–280). Charlotte, NC: Information Age Publishers.

Rabin, S. (2009). *Introduction to game development* (2nd ed.). Herndon, VA: Cengage Learning.

Robinett, W. (1984/2006). Adventure as a video game: Adventure for the Atari 2600 from Inventing the Adventure Game. In K. Salen & E. Zimmerman (Eds.), *The game designer reader: A rules of play anthology* (pp. 690–713). Cambridge, MA: MIT Press.

Salen, K. (2008). Toward an ecology of gaming. In K. Salen (Ed.), *The ecology of games: Connecting youth, games, and learning* (pp. 1–17). Cambridge, MA: MIT Press.

Salen, K., & Zimmerman, E. (2004). *Rules of play: Game design fundamentals*. Cambridge, MA: MIT Press.

Schell, J. (2008a). *The art of game design: A book of lenses*. Burlington, MA: Morgan Kaufmann.

Squire, K. (2011). *Video games and learning: Teaching and participatory culture in the digital age*. New York: Teachers College Press.

Strauss, V. (2014, Jan 18) Everything you need to know about Common Core—Ravitch. The Answer Sheet Blog, *Washington Post*. Retrieved January 9, 2016, from http://www.washingtonpost.com/blogs/answer-sheet/wp/2014/01/18/everything-you-need-to-know-about-common-core-ravitch/

Swan, R. (2010). Feedforward as an essential active principle of engagement in computer games. In R. Van Eck (Ed.), *Gaming and cognition: Theories and practice from the learning sciences* (pp. 108–136). Hershey, PA: Information Science Reference.