CHAPTER TWO

The Game Mechanic at Work

The game designer has a number of responsibilities in the game development process. The game designer directs the creative vision of the game from conception through to launch. Like the director of a film, the game designer is responsible for creating and maintaining the creative direction of the game, working with the artists, programmers and producers to bring the game to life. This includes a wide range of specific responsibilities from brainstorming concepts to writing rules to crafting levels for the game.

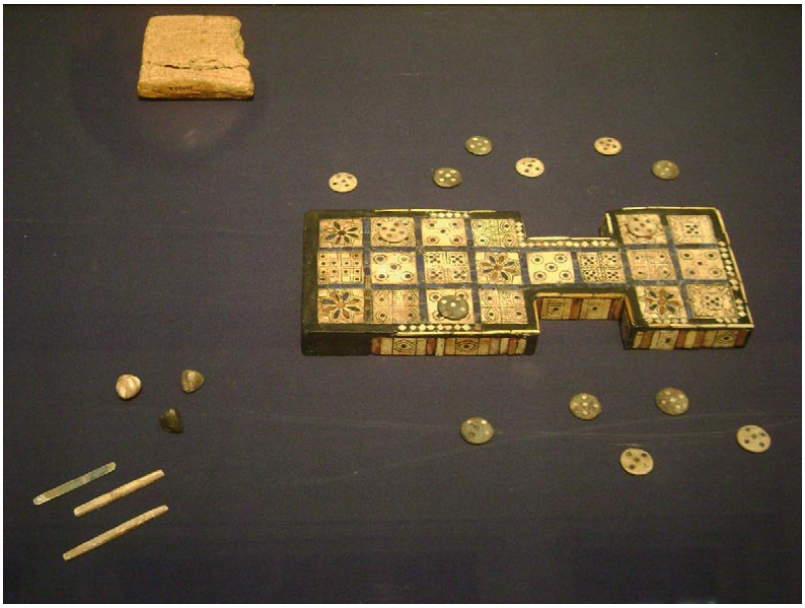
The Role of the Game Designer

People have been designing games for thousands of years. In the 1920s, Sir Leonard Woolley unearthed a board game in the Royal Tombs of Ur in what is now southern Iraq. *The Royal Game of Ur,* dating back to 2,600 B.C., is probably the world's oldest intact board game (Figure 2.1). The Egyptian game *Senet* is even older. Historians have found evidence dating *Senet* back to 3500 B.C. But we can confidently say games are far older than that. As long as leisure time has existed, we've had play and games to help fill those spare minutes. And as long as there have been games, there have been game designers, picking out stones, crafting boards and prescribing rules to govern play.

And while the tools of implementation may have changed—we now push pixels instead of round stones—the basic idea is still the same: Craft a set of rules that governs play. As an author strings words together into sentences and builds them up into stories, a game designer combines rules into mechanics and assembles those mechanics into games.

We all have some experience with game design. As children, we work with our friends to turn our play into games. On the playground, bored with simple tag, we conspire to add new rules to the basic mechanic of tag, building up new games, from *Freeze Tag* to *Television Tag* to *Zombie Tag,* gradually making the game more complex and interesting. Just think of the common refrain echoing from kids playing

FIGURE **2.1**



The *Royal Game of Ur* on view at the British Museum. (WikiCommons[1])

tag, "No tag backs! You gotta wait at least 10 seconds." These kids have instantly recognized a flaw in tag that often devolves the game into a stand-off. One simple rule tweak fixes the game: After being tagged, you must wait at least 10 seconds before you can tag the same player back.

The practice of adjusting games continues into adulthood. Pick-up basketball players adjust the rules of games like 21 or three-on-three to suit the players at their local court. They change rules that dictate courting the ball, committing fouls and even points per basket. Friends gathering to play board games like *Trivial Pursuit* or *Monopoly* add house rule variations to deal with perceived weaknesses with the game. For example, a common *Monopoly* house rule says landing on Free Parking pays out to the player all cash accumulated from Chance and Community Chest cards. Playing a game forces us into an intimate relationship with the rules and naturally leads us to adjust them to improve the experience of playing (or sometimes, more devilishly, to strengthen our own hand).

It's almost natural, then, that the job of game designer is somewhat overlooked. We take for granted that games exist. Many of the games most familiar to us—like tag—seem so elemental that you cannot imagine them not existing. Games have an active and rich folk tradition that we regularly interact with. Like ghost stories, it is often assumed that folk games like tag or hide and seek do not have authors. Games are one of the few media that still have such an active

[1]http://commons.wikimedia.org/wiki/File:Royal_game_of_Ur,at_the_British_Museum.jpg, User: Zzztriple2000

folk tradition. It can seem as if they've always existed. Even popular board games like *Monopoly* have been around long enough that we take their existence as a given. But *Monopoly* was designed by someone, Charles Darrow, in fact. His game is a descendant of *The Landlord's Game* designed by Elizabeth Magie Phillips, a Quaker follower of the economist Henry George, who created her game to help explain George's single tax theory.

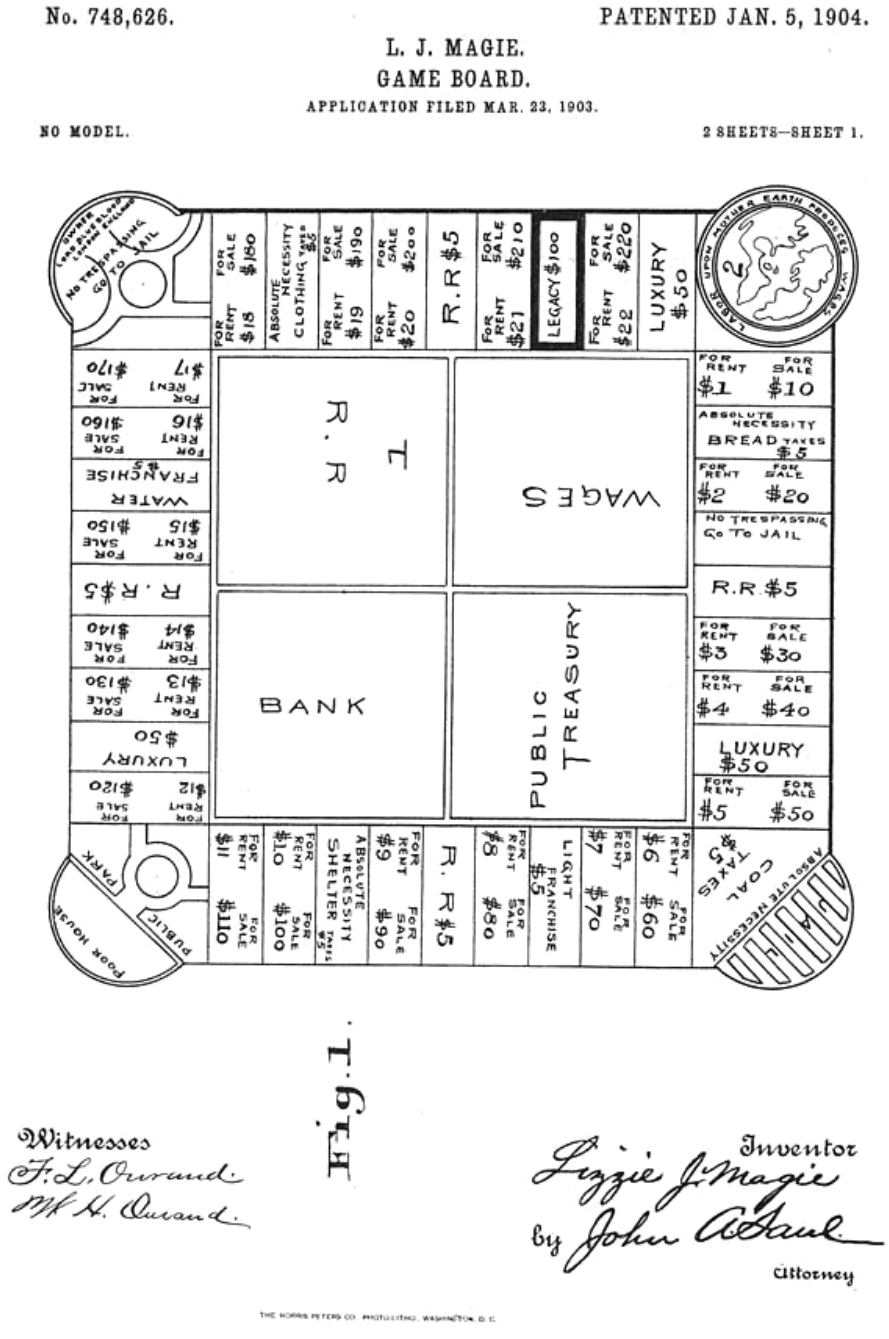
As video games came to dominate the game industry, people have become more familiar with the profession of game development. However, misconceptions still abound. The software and art are the most tangible parts of a video game. Video games fill the screen with lush graphics, moving around, responding to your input. Unlike a board game, you rarely sit down and read the rules to a video game. Those rules exist, but they are baked into the code. As a consequence, the general population equates game design with programming or 3D graphic design. Tell most people you work in game design, and they'll ask if you are a programmer or an artist. Both of those jobs are absolutely integral to video game development. But there are some other vital roles that need to be filled to bring a video game to market.

The same people doing game design may also be programming and drawing characters, but they are all separate arts requiring different training. In the independent development scene, with small teams, team members may find themselves conscripted into handling multiple facets of game development. This can be good. A healthy knowledge of programming and art creation is indispensable when designing games. That background gives the game designer a better sense of what is possible and the cost of implementing ideas. But in the end they are all separate tasks.

And it all starts with the game designer. Game design is the art of creating the system and experience of the game. The game designer generates the concept that underlies the game. She says, "This game is going to be about matching sets of three adjacent objects." This concept gives the game a basic shape and direction. The Game designer defines the space of the game: "The game will take place on a 10-by 10-grid with a red, blue, yellow, green or purple block in each square." After that, the game designer must sketch out the gameplay: "The player will make matches by swapping adjacent blocks to create sets of three." This begins to define how the player will go about making those matches of three. Once they have the initial concept, the game designer fleshes out this system, crafting the nature of the experience by adding rules. She adds a rule, "The player can only swap two adjacent blocks if one of the swapped blocks will wind up in a matching set of three or more." This prevents the player from swapping any two blocks, and adds a level of challenge to the game. Slowly, through the accumulation of rules, the game is constructed. Finally, you wind up with something like the system behind *Bejeweled*.

This doesn't all happen at once, and it certainly doesn't happen in a vacuum. The development of the game design is a living process that responds and changes as the game is built and the mechanics are tested. The game designer works in partnership with other team members to build the essential elements of the game. The game designer makes an initial guess at what will make a good game and then the team finds a way to prototype that idea and test it through play. The game

FIGURE **2.2**



The patent application for *The Landlord's Game* shows a game that evolved into Monopoly as different designers adapted and refined the mechanics. (WikiCommons[2])

[2]http://commons.wikimedia.org/wiki/File:BoardGamePatentMagie.png, User: Zzztriple2000

designer then refines the game idea, adding or removing elements and playtests the game again.

As the game progresses and grows more complicated, the game designers must keep the game clear and focused. Clarity is important in any game, but it is of the utmost importance in casual games. The game designer is an advocate for the player and must focus on delivering the smoothest experience possible.

Video Game Designers

These days, "game designer" seems synonymous with "video game designer" in the popular imagination. This means the job is part game design and part software developer.

A video game development team can generally be broken down along the following lines:

- Producers manage the development budget and team, helping organize the team and providing the necessary support so each team member can do his or her job.
- Game designers are responsible for the concept and vision of the game. If you were to equate the role to movies, the game designer would be the director. He or she works with the artists and the programmers, helping guide the implementation of the game from the overall visual tone to the bounciness of the physics engine.
- Visual artists produce the art assets for the game, be they hand-drawn animations or 3D models. The field of visual arts in game development is huge and encompasses a number of different specialties, from user-interface design to character design to animation.
- Audio artists create the sound effects and music for the game.
- Programmers write the software code that brings the game to life. They take the specifications and rules and turn them in to code. This is the heavy lifting in video game development.
- The Quality Assurance team puts the game through the paces, writing test plans and thoroughly testing every aspect of the game. QA is often a good entry job into the game industry, as it familiarizes you with all aspects of development, forcing you to work with producers, artists, game designers and programmers.

Board Game Designers

Video games get all of the attention and steal the headlines. But casual gamers are just as likely to play board games as video games. Fortunately, a number of talented game designers dedicate their energies to board games. Board game designers have many of the same responsibilities as video game designers. They conceive the

game, write the rules, work with artists, and continually iterate and refine the game. However, when creating board games, designers cannot rely on programmers and the software to do all of their heavy calculations. So they must pay close attention to balancing the game and the probabilities of different moves and actions.

The Responsibilities of the Game Designer

On a high level, the game designer is responsible for the concept and vision of the game. But what does that boil down to on a day-to-day basis? In practice, the game designer's job looks something like this:

- Work as part of the team
- Generate the concept
- Craft the rules
- Write game and software specifications
- Guide the implementation
- Playtest the game
- Refine the concept
- Design the levels

Work as Part of the Team

Good game development is a team effort, requiring the input of individuals with different skills and knowledge. The game designer has specific knowledge and skills relating to how the game system should work. But programmers and artists have very specialized knowledge that greatly influences the game. The game designer must communicate the structure and tone of the game to the other team members. The more effectively he or she can share the vision, the more the other members of the team can contribute ideas and suggestions that draw on their specialized knowledge. If the idea and requirements of the game are clear, the programmers will be able to suggest everything from an appropriate game engine to the variables they can expose to the game designer. A working grasp of programming is extremely useful for any game designer. It helps give you an idea of possibilities and makes communicating software needs easier.

The game designer also needs to speak the language of visual and audio artists. Artists will produce the art and audio assets that go into the game. But they will look to the game designer for guidance on the overall tone and direction of the game. The art should be in service of the gameplay, so the designer and the artist will need to be on the same page. Like programming, visual art has its own special language. The designer saying, "Make it happier," is not helpful and will leave the artist with

insufficient direction. But being able to say, "We need to lighten the color palette and increase the line weight to give the characters a more round, cartoonish feel" will go much further in helping the artist know how to make the game "happier." The same holds true for audio artists.

If you don't already have a knowledge of these disciplines, you should work on developing one. Pay attention to the way artists and programmers talk about their work. Work with these members of your team to develop a rapport. Ask them to teach you about their work and, in turn, share your knowledge of game design.

Through all of this, the game designer must work in partnership with the producer to make sure the game stays on track. The producer is responsible for keeping the game on schedule and budget. But the game designer must help him or her do that. The game designer must recognize when to cut features to save time and when to push hard for a feature that will greatly improve gameplay and the player experience. This can lead to some back and forth between the producer and the game designer. Maintaining a relationship of respect is vital. In the end, if both the producer and game designer are clearly communicating, the two will develop a close relationship which will help keep development focused.

Each studio will have its own variation on team structure. At very small developers, the same person may fill multiple roles, from designing the game to programming to drawing the art. At large console developers, the development team will consist of dozens of people, each one responsible for some small aspect within one of the production silos. But no matter what, the game designer needs to be the consummate team player. He or she needs to help everyone else on the team do their jobs better.

Generate the Concept

The game designer is involved from the very outset in generating the concept for the game. External forces will often provide certain constraints for the game. Commercial interests may dictate subject matter or a general game type. The game platform will push the gameplay in directions that suit the controls of the system. For example, console controllers are ill-suited for complex strategy games. The need for a keyboard and mouse tends to push strategy games onto PCs. Most importantly, your intended audience will drive certain gameplay decisions. But in any of these cases, the game designer needs to be there to help define the mechanic at the core of the game. The rest of the game, from the story to the art, should grow from the core mechanic.

At the outset of a project, the game designer will brainstorm initial concepts for the game. Some designers generate many wildly different concepts. Other designers quickly settle on one general idea and come up with variations. It is up to you to find a method that best fits with your way of generating ideas.

The easiest way to begin thinking about a game is to start with a game mechanic or interaction you know works. Many games are built on top of other games. It's

important that game designers play other games and closely analyze which mechanics work and which don't and why. Game designers may play a game and see new potential in a game mechanic. Perhaps you will see a way to reuse the mechanic with different content. Or you may envision a way to modify the mechanic and make a new game. What if you mixed *Solitaire* with *Scrabble* so you played with a deck of cards marked by letters, and your goal was to spell specific words, not just get cards in numerical order? There are millions of ways to modify and build new games. You just have to look around.

Some game studios will spend time prototyping mechanics. A game designer may mock up ideas in paper prototype form or work with a programmer to put together a small digital prototype. This prototype should demonstrate a game mechanic or interaction scheme or perhaps even just a possible visual content direction. When it comes time to develop a new game, the team can draw on these prototypes and pick and choose the prototypes they feel work the best. Spending time prototyping mechanics can be difficult for a studio to maintain. It ties up resources on work that doesn't immediately generate revenue. But it can prove invaluable in the long run, as it gives new game concept generation a set of departure points.

If the project you're working on calls for a fresh concept, you may have to start with pure brainstorming to find a game concept and mechanic. This sort of open-ended assignment can be daunting. In order to reach a set of interesting and viable ideas, it's important to give your brainstorming some structure. First off, when brainstorming game ideas from scratch, define some parameters for the game before you begin, such as your goals for the game, the platform and audience. Blue sky brainstorms where you can head in any direction are often unproductive.

Before you come up with game concepts, it is important that you have an idea of what you want players to get out of the game. Are you aiming for commercial success or do you prefer to use your game to make an artistic statement? Do you want players to have a quick, intense, five-minute experience? If so, your game idea doesn't need to be as deep and strategic as a game that must hold a player's attention over 80 hours. Instead, you can focus on a novel or interesting interaction scheme. If you want players to engage with the game for hours on end over multiple weeks, then you will need to design a game with structures that draw players back again and again. You might decide to do this through a complex and strategic game. Or you might make the game about a persistent identity that players nurture by collecting items and experience points so they can level up the character. You'll generate concepts in the brainstorm, but it's important that you have an idea of how people will play your game before you start brainstorming. That way you can focus on ideas that meet these goals.

4 Minutes and 33 Seconds of Uniqueness is a beautiful little game created by Petri Purho. The only way to the win the game is to be the only person in the world playing it for four minutes and 33 seconds. During the game all you see is a black window (Figure 2.3). The game constantly checks with a server to see if anyone else anywhere in the world has started the game. If they have, the game simply closes and you lose. The game is a clever little idea, but hardly commercially viable. But that's okay, because the game is more of an art project than a commercial game,

FIGURE
2.3

***4 Minutes and 33 Seconds of Uniqueness*. (Reproduced by permission of Petri Purho, Heather Kelley and Jonatan Söderström)**

and Purho knew that when he set about making the game. The game was actually inspired by the piece *4'33"* which is sometimes referred to as "Four minutes and thirty-three seconds of silence" by the avant-garde composer John Cage. There is a large audience of people interested in exploring games as art, and this game strikes a chord. Knowing your intended audience will help you know what you can and cannot do in your game—how to craft it to satisfy your players.

Knowing who you are making the game for will suggest the level of complexity and engagement the game should offer. If you are making a game for the casual audience, it's important to keep the game simple and focused on the core mechanic. A casual game cannot demand players dedicate an hour to every play session. The game should be playable in small, discrete chunks. This will help the game match the amount of time casual game players can give to it.

Determining your audience can also help suggest content directions for the game. Attractive content often helps entice players to try the game. You want to find content that will draw players in, not turn them off. This doesn't mean that all content needs to be bland and market-tested. Instead, you just need to be aware of how your audience perceives your game. The audience for Web games on Kongregate has different taste than the players who download casual downloadable PC games from the portal Big Fish Games. Kongregate attracts more kids with greater taste for typical gamer tropes than a downloadable portal like Big Fish will. The downloadable portals attract a large number of adult women. You can make a zombie game for that market, but it will probably be more successful if the zombies look like the cute-cartoon zombies of *Plants vs. Zombies* (Figure 2.4) than if the zombies display the ghoulish gore of the zombies in the hardcore console title *Left 4 Dead* (Figure 2.5).

FIGURE
2.4

Plants vs. Zombies **has been a big hit in the casual market despite the undead content. PopCap made the content palatable by using a cartoonish style that made the zombies almost as endearing as the flowers. (Reproduced by permission of PopCap Games)**

Similarly, determining the platform will help you hone in on your audience, as well as highlight technical limitations for your game. Distributing your game on Xbox Live versus downloadable portals will put your game in front of very different people. Get familiar with the types of games on different platforms and portals. Play each outlet's games, look through the popular games and any user profile information you can find. Keep abreast of these outlets by revisiting them. The audience can shift and change over time. This will give you a greater idea about the length of game and depth of gameplay the audience will be expecting. Use this information in your brainstorming. If you've decided to make a Web game for Addicting Games, you probably don't want to make a strategy game that takes hours to play. Instead, you probably want to make a small game with a simple, yet repeatable, mechanic that may only take a few minutes or even seconds to play.

Each platform will have technical limitations to contend with. If you make a game for Xbox Live or PlayStation Home, you'll need a control scheme that works well with a joystick controller. As a general rule of thumb, PC downloadables should be playable with just the mouse. And if you want to be really safe, your downloadable

FIGURE
2.5

The zombies in Valve's *Left 4 Dead* aren't nearly as endearing as the undead in *Plants vs. Zombies*. They made the game a hit in the hardcore market, but it's hard to imagine these gruesome zombies striking a chord with casual gamers. (© Valve Software)

game should probably just use the left mouse button. This will no doubt change over time as casual downloadable players become more advanced. But it will still be a while before casual players are using WASD-keys and the mouse in combination to control games. If you develop games for an iPhone, a player's finger may operate much like a mouse, but it will obscure part of the screen in the way a mouse will not. Ninja Kiwi's port of their popular web game *Bloons* to the iPhone is marred by this very limitation. Touching your finger down to aim your shot obscures your aiming arrow, making it much harder to shoot accurately than when you play the game on the computer and you can use a mouse (Figure 2.6).

Meanwhile, *Gigaputt* (developed by my company, Gigantic Mechanic) takes advantage of the special abilities of the iPhone (Figure 2.7). In the golf game, you swing the phone like a golf club and knock the virtual ball over a Google Map of your surrounding neighborhood. The phone leverages the GPS and the accelerometer built into the phone to enable the gameplay. This is an example of a game where the specific constraints and abilities of the platform inspire the gameplay and mechanics.

The point designers should heed is clear: look into the technical and control limitations imposed by your platform and use these as parameters when you start brainstorming concepts. These constraints will actually help spark ideas, not limit them.

FIGURE
2.6

When you play the Web-version of *Bloons* on the PC the mouse does not obscure the game-playing area. The experience on the iPhone is flawed because the designers didn't properly account for the touch control scheme of the iPhone and how it would obscure the game area. (© Ninja Kiwi)

Once you have determined your audience and platform and researched what they entail, you should have a nice set of reference points and constraints to inform your brainstorming. You can add additional constraints if you like to further focus your idea generation.

FIGURE
2.7

***Gigaputt* uses the iPhone GPS and the accelerometer as key components of the gameplay. (Reproduced by permission of Gigantic Mechanic)**

After you and your team feel that you have produced a healthy basket of ideas, you need to trim and organize your ideas. The game designer should classify the ideas, looking for similarities and trends. Organize them by content ideas and game mechanic ideas. This doesn't necessarily need to be done in a group, as it could take some time. At this point, you should go through all of the ideas once to cut ideas which seem out of scope for the project or just plain impossible. With the remaining ideas, you can spend some time fleshing out the most viable ideas, imagining how the idea would look and play as a full game. At this stage, it's important that you recognize intriguing gameplay possibilities. When you are designing casual games, it's also important to keep the idea simple. At this point, it can be very tempting to start tacking features, goals and strategies on to the game. You certainly want to imagine a robust game, but be wary of going too far before testing the main interaction at the core of your game idea.

Once you have settled on a few ideas you like, the game designer should conduct some competitive analysis to see what other similar games exist. This includes looking for similar games, and researching how well they've sold. Envision how your game will fit into the market. Just because a similar game already exists doesn't mean that you should not pursue an idea. But you do need to think about how your game will differ and surprise players to stand out among the crowd.

If you are working for a publisher or client, you will need to pitch your ideas to them as well. Sometimes, they will want to see a playable prototype, while other times a well-written pitch document will suffice—more likely if you already have a proven track record making successful games. Your game pitch should be concise, describing the game as succinctly as possible. Include art mock-ups that help describe the gameplay and general look and feel of the game. Pictures always help convey abstract game system ideas.

With your team and client, settle on the game that best meets your objectives, audience expectations and platform requirements.

Craft the Rules

Rules provide the skeletal framework for your game. They give your game structure and solidity. They tell the players what they can and cannot do.

Once the team has settled on a concept, the designer writes the rules and fleshes out the game. It may seem silly to write rules for a video game, but it's still an important exercise. It often seems like video games don't have rules so much as interfaces. The software simply defines what the player can and can't do, so the player doesn't need to know the rules. It's true, the game guides the player, but the software still follows rules. As the game designer, you must define the behavior so you can write the software specifications for the programmer. Writing out the actual rules to the game serves as a good starting point for thinking about the software specifications.

There is no one perfect way to write rules for a game, but there are lots of ways to wind up with a confusing set of rules. Before you write your rules, make sure you have a clear idea of how the game plays and what experience you want the players to have. Then craft your rules accordingly. Keep your audience in mind. For example, hardcore board gamers are much more willing to pore through and study a long booklet of rules than most casual gamers. In my experience, most people want to get going with the game as soon as possible. They view rules as an unfortunate impediment to playing, not as the intricate and careful crafting of experience that game designers do. As always, it's best to give your audience what they want, not necessarily what you want.

Some general guidelines for writing rules:

- Be concise and exact
- Be firm
- Can't vs. must
- Instructions are rules too
- Avoid too many special cases
- State the game's goal upfront
- Tell the rules like a story

- Give examples
- Organize play into phases

Be Concise and Exact

Be as concise and exact as possible. Clearly state what will happen when a game event occurs. Unclear rules confuse and frustrate players. Many players will simply abandon a game before even starting if the rules are unclear. In addition, be careful not to contradict yourself with rules.

If you say a player has power-up, define exactly how many rounds or seconds the player can use the power-up. If someone loses or gains points, define exactly how many points. If you generalize by saying, "When you place the paint bomb, it changes the color of a bunch of the gems," players will have no idea how many gems change color, nor what color they take on. It is much better to say, "When you place the paint bomb, it changes the color of all gems in horizontally and vertically adjacent squares to a color of the player's choosing." This way, the outcome is clearly defined.

Be Firm

Your rules should not leave room for argument. The game system governs the game. It dictates to players how to play. People expect that of rules. Deliver that.

Use strong language like "will" instead of "may." This will become increasingly important when you begin to write specifications for your game.

In addition, you don't want players to have to interpret your rules during play. They'll wind up arguing with each other, sucking the life out of your game. You want them to stay lodged in the game system you create. Stepping out to argue about rules snaps players out of the game. It's like seeing the boom mike dip into the frame of a movie—it ruins the illusion of the game.

Can't vs. Must

It's easy to think of rules as just a list of "Can't Dos." Pawns can't move backwards; you can't touch the ball with your hands; you can't move while holding the basketball unless you dribble. While you will certainly need to have a number of Can't Dos in your list of rules, if you have too many, it may feel claustrophobic to players. Can't Do rules also do a poor job of telling the player what he or she can do, how he or she should actually be playing the game. Rather than simply thinking of Can't Dos, structure your rules as Musts. So instead of saying, "You can't move while holding the basketball, unless you dribble," try, "To move with the basketball, you must dribble." This makes the rule more affirmative and begins to help the player see what type of actions the game wants him or her to take.

Instructions Are Rules Too

There can sometimes be questions about what constitutes a rule. A rule is any information the player needs to play the game. At its core, each rule should reflect an element of the game system. However, when written, they may look more like instructions. That's because instructions describe how to interact with the game system. So the rule might be, "The number of spaces a player can move is determined by the roll of a six-sided die." But the rule as instructions may read "Roll the dice to determine how many spaces you can move."

Avoid Too Many Special Cases

You want your game to have an overall systematic consistency. Rules and the effects they dictate should naturally flow from one to the other. Special cases are instances where one particular element of the game behaves unlike all of the other elements, breaking the system's consistency. You know you're headed for a special case when you have to write, "You always do this, EXCEPT when this ONE thing occurs. Then you do this other thing instead." For example, it would be inconsistent game design if the rules of basketball were changed to say, "To move with the basketball, you must dribble, unless you have just scored a three-point basket." Not only does that rule not make any sense given the context of the game, it would also break up the action of the game and make playing much choppier as players began to behave in different ways based on specific events. Some special cases are necessary. But if you lard your games with them, players have a hard time keeping track of what they should do. In board games, they will be forced to constantly refer back to the rules. In video games, they will likely just be confused by constantly changing behaviors.

State the Game's Goal Upfront

Before you plunge players into lengthy descriptions of what they must do and what they can't, tell them what their overall goal is. Tell them how they win. This can be as simple as, "The goal is to score the most points." This puts all of the rest of the rules into context. Players can read the rules and situate them in their minds in relation to that goal. A rule might help a player earn points or might cost him or her points. If players have a clear idea of the game goal in their minds, they know immediately if that rule helps push them closer to winning or might cost them the game.

Tell the Rules Like a Story

Where you can, it helps to narrativize your rules. Don't just spit out rules in a long list. Put them into a natural flow that reflects the gameplay. Introduce high-level rules that frame the action of the game first. Then introduce rules in an order that reflects how players will encounter them in the game. Craft your set of rules into a walkthrough of the game, from set up to standard moves to the end game. This gives players a sense of the arc of the play experience so they will know where they are in

the game. Think of it like telling a story. You introduce the players to the setting, run them through the action and finally move on to the climax of the game. This method can enable players to begin playing the game before knowing all of the rules, letting them learn some of the rules by playing. Rules learned while playing usually make more sense because the player now has a feel for the game and its constraints.

Give Examples

Help players understand a rule by providing a description of the rule in action. Say someone was having a hard time understanding the rule, "To move with the basketball you must dribble," you could clear it up with a simple example, "If you catch the ball you can no longer move as freely. To move, you must dribble the ball. You can also shoot or pass the ball to another player. Once you get rid of the ball, you can move freely again."

Organize Play into Phases

Your game should have a natural rhythm to it. Each move should sensibly lead to the next action. This will help players move smoothly through the game. This doesn't mean each move needs to be one simple click or roll of the dice. If your game requires more complex interaction patterns, consider breaking each turn into phases. A lot of complex board games do this. This helps players chunk rules and instructions into more understandable moments. For example, the excellent board game *Settlers of Catan* breaks the game into a number of different phases, each with a set of instructions. Each turn a player takes is broken into three different phases: the Raw Material Production Phase, the Trade Phase and the Build Phase. Each phase has unique actions associated with it and involves each player in the game in different ways. By breaking each turn into these three phases, players can develop a rhythm to their play. Players know when to pay attention and when they can tune out.

It also makes reading the rules easier. The player can digest each phase in turn, rather than trying to understand, all at once, all of the rules about collecting materials, trading with others and building new resources.

This sort of organization can be valuable in video games as well. A game with a several step process can benefit by laying out each move as a series of steps or phases. This will help players internalize the structure of the game and navigate through the UI. The easier it is to understand and make moves, the more effort the player can expend on strategy and gameplay.

Write Game and Software Specifications

Before the programmers and artists can begin working on the game, the game designer will need to provide a software specification document. The spec, as it's often called, serves as the blueprint for the game. Spec documents can run to hundreds of pages

for large games, as they need to outline all of the functionality in the game. For small teams using agile project management methodologies, the documents will be much smaller, but grow over time as the game evolves. In this document, the game designer lays out all of the functions the game will need to perform and the variables the game designers will need to access to balance the game. The document should be detailed, but not so verbose that programmers refuse to read it.

Included in this document should be rough wireframes for the game's user interface, or UI, as it's commonly called. The specifications should also include notes on the visual direction for the artists. This can include everything from rough sketches to written descriptions to sample images from games, books, comics, photos and movies that capture the desired look.

Writing good, clear specifications is an art in itself. Game designers new to writing specs tend to underwrite them, which leaves big holes in the logic of the game that the programmer will then need to fill with his or her own assumptions. This will frustrate many programmers, especially if the game designer later asks for changes because the assumptions didn't match what the designer had in mind. It's best to think out the logic of the game as thoroughly as possible and then document it, so the programmers don't have to make guesses as to how the game should work. Another common mistake is overwriting the specs, putting in so much detail that the programmers feel they can't make key decisions about how to implement features.

Probably the best thing you can do before writing a specification document is talk with your team to fully understand how they want you to structure and write the spec. They are the ones who have to read and interpret the document, so you want to craft it for them. Some programmers may want very detailed specifications that list every possible iteration of a function. Others may prefer just a high-level sentence or two that describes the functionality. Still other programmers may prefer lots of pictures and wireframes showing how the game moves from state to state. Similarly, work with the artists to decide the best way to agree on an art style. Once your team has given you an idea of how they want to receive specifications you can set about crafting the document in this model.

The game designer will lay out the initial design document, but other members of the team may contribute to it as well, adding information about the narrative, art, UI and technology. The game designer will own the design document, but they maintain it to help facilitate conversation and to keep a record of creative decisions for the game.

Specs must be clear enough to convey the essential ideas, but concise enough that readers are not bogged down. Effective organization of your design document is key. Each game is different and will require a different structure. I recommend breaking your document into several key sections:

General Summary

State the purpose of the document and a very high-level view of the game. This is to orient anyone picking up the document and helps him or her know what to expect.

- Table of contents
- Story/theme information
- General look and feel
- Summary of gameplay
- Information architecture
- Specifications for specific features

Table of Contents

As the document grows in detail and length, a table of contents will be an invaluable tool to help readers find specific information buried within the document.

Story/Theme Information

Set the narrative stage for the game. You want everything in your game to feel consistent, from the art to the gameplay to the user interface. Describing the narrative universe of the game will help get artists, programmers, audio designers and UI artists on the same page. If your game has a detailed story, give an overview. This section is still valuable, even if your game is an abstract game with no particular story. Even abstract games have a theme. Lay out that theme.

As the document grows, you may start to include more details about the narrative. If the story is short, you may include the actual script in the design document. For longer narratives, it may be easier to create a separate script document so you don't bloat your design document.

General Look and Feel

In this section, you'll begin to establish the art style for the game. The designer may do this by simply describing the art style he or she feels will work within the game. So you might say, "The game takes place in the world of high fashion and the art should reflect that. It should borrow the look and feel of fashion magazines like *Vogue* mixed with the fashion illustrations of Antonio Lopez and David Downton."

Ideally, your descriptions should be accompanied by some visual research, such as pictures that exemplify the general style you are envisioning. But don't go overboard—the game art director will want to more fully establish the style. The direction you suggest should be the result of conversations with the team about what styles the team thinks will work. Once a general art style is set, the game's art director will look for more comparable art to round out the visual research. Again, this information can be stored in the design document or housed in a separate file, depending on the volume.

Summary of Gameplay

You should provide a line-by-line description of the gameplay, from starting the game and picking a level to the moment-to-moment interaction within the game to beating a level and moving on to the next level. Looking at the experience as a whole will help you think about how the overall experience is crafted. It's very tempting to only think about the game as independent chunks. This will lead you to concentrate on the game screen interactions. This is necessary. You want that section to be well thought out, but you also want to view the game as a whole, taking into account how well the whole experience coheres together. This summary, along with the screen flow, will serve as the roadmap for features that need to be specified.

When describing the gameplay, describe the interaction hit upon the key beats within your game. You want to provide a procedural description of how to play the game.

I often find it helpful to use this summary to lay out basic terminology for the game that you will use later when you spec a feature.

Here's an example of an initial gameplay description for a simple video game version of *Solitaire*:

- Game Loads
  - When the game loads, the **Main Deck** is in the upper left corner.
  - **Foundations** appear as four ghosted rectangles in the upper right corner.
  - **Cards** will be dealt into the **Tableau** underneath the Main Deck and the Foundations.
- Dealing Sequence
  - The **Dealing Sequence** commences as soon as the level loads.
  - During the Dealing Sequence, cards are dealt from the Main Deck to seven **Piles**.
  - When the Dealing Sequence is complete, the player has:
    - Seven Piles of cards in a Tableau with the top card on each Pile face up.
    - Four empty slots on the Foundation.
    - The Main Deck placed face down.
- Play Begins
  - Player begins organizing cards.
  - The player organizes cards on the Tableau by placing face-up cards on top of one another. Cards are organized from the king down to the ace, alternating **Suit** colors on each card.
    - A sequence of organized cards can start with any card, but must proceed down from that point.
  - When a player moves a face-up card, revealing a face-down card, the face-down card is turned face up.

    - The player organizes as many cards as they can.
    - Stacks of face-up cards can be moved on top of other face-up cards if they maintain the proper sequence.
- Playing off the Main Deck
    - If the player has no moves left to make on the Tableau or Foundation, he or she can click the Main Deck, flipping over the top card and placing it in the **Main Deck Active Pile**.
    - Cards can be drawn off of the Main Deck Active Pile and placed on the Tableau or Foundation, if they are in the proper sequence.
- Playing on the Foundations
    - To win the game, the player must sort all of the cards by Suit into four separate stacks on the Foundation rectangles.
    - The Foundation stacks proceed from ace to king.
    - Once all four Foundation Stacks are complete, the player wins.
- Win Animation
    - When the player wins the game by completing all four Foundation stacks, the **Win Animation** plays.
    - After the Win Animation completes, the **New Game Dialog** appears and asks if the player wants to play again.

It can be very useful to include hyperlinks to key sections in your game design document. Using these hyperlinks, your team members can jump to a section where they will get more detail about a function or element of the game. This way, you can avoid cluttering your description with too many details about specific functions that don't concern the overall experience and gameplay. In the description above, each of the bolded words would be hyperlinks that link to a section with more detail about that element. As the game grows and you add features to your game, you can come back and add them to your initial gameplay description.

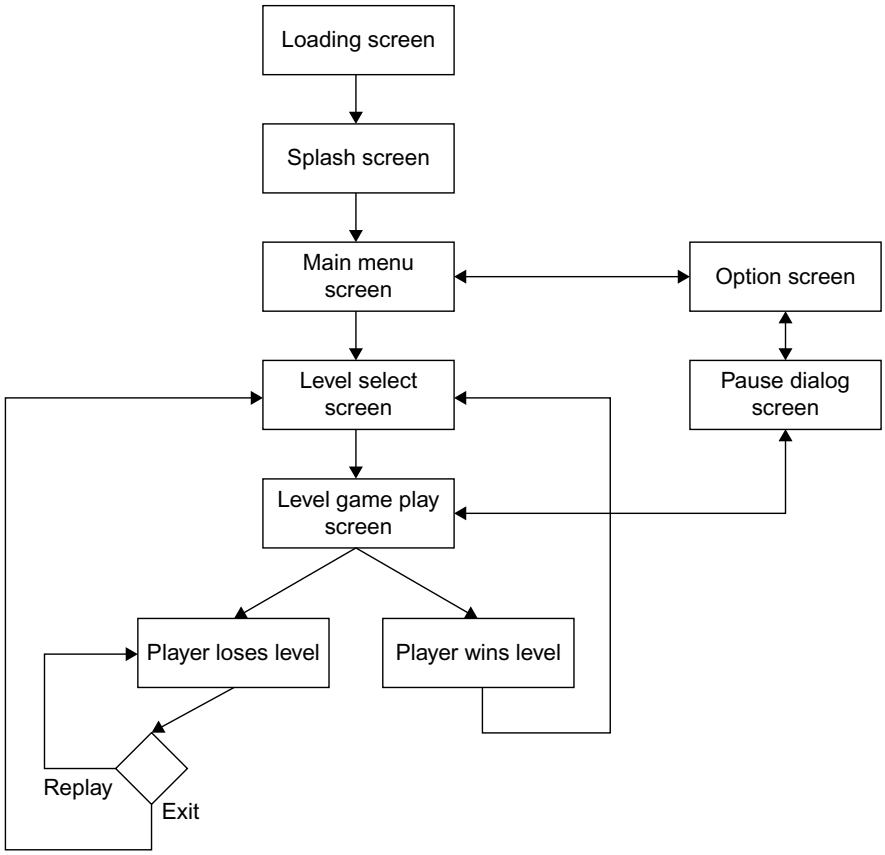
Information Architecture

Once you have your initial description of the gameplay, you should start working on the information architecture for the game. This includes a screenflow diagram that shows how all of the screens and processes in the game are connected. Once you have that screenflow, you can begin creating wireframes for each screen.

Screenflows can be laid out in programs dedicated to information architecture, like Visio, or pieced together in Illustrator or Photoshop. I've found that Microsoft Word has enough tools to create adequate screenflow diagrams for games.

Let's imagine our *Solitaire* video game is composed of a series of levels you have to play through. Each level is a specific assortment of *Solitaire* cards you must complete before moving on to the next level. Figure 2.8 is an example of a screenflow diagram for that level-based version of *Solitaire*.

FIGURE
2.8



A screenflow shows the process of the game and the various decisions made at each key point in navigating the software of the game.

Screenflows can become very complex very quickly, even for simple games. You can always break them up into multiple, smaller screenflows that cover specific interactions.

Once you have this screenflow, it gives you a good idea of what screens you need to wireframe and build in your game. It also provides a good map of the overall experience for other team members.

Wireframes are skeletal versions of each screen in your game. The wireframe should indicate all of the necessary elements for the screen. This includes buttons, score meters and gameplay elements. The wireframe does not indicate the final look and feel of the UI or even the exact layout; rather, it determines the weight and importance of each element. By looking at the wireframe, the artists should be able to see that the Pause button is smaller and lower than the Play button. That means Pause is less important than Play. The wireframe should also lay out the basic elements of the gameplay, illustrating how the pieces should generally be organized on the screen.

The amount of detail in a wireframe depends on how much you need to communicate and the roles on your team. Traditionally, a UI expert would be in charge of

producing wireframes for each screen. The first pass the game designer makes can be relatively bare. However, on small teams, the game designer may also be doing a fair amount of information architecture and layout, in which case he or she may need to make more detailed wireframes. There is a definite trade-off in making bare bones wireframes versus detailed interface mock-ups. Stripped-down wireframes are easy to change around and don't get locked down by the game "theme." They can be divorced from the art. However, this stripped-down approach means the artists will have to do more work to really make sure all of the elements you spec out actually fit on screen in a legible format.

Wireframes can be as simple as Figure 2.9. Adding more detail to the wireframes makes them more attractive and enables you to begin to explore exact layouts versus theoretical layouts. You can begin to see if the button will actually be large enough to be legible and clickable. This is very useful, but can add a lot of time to production. More detail also makes the wireframes harder to change, should the theme change. Beware when showing detailed screens to others. When people see art assets which look relatively polished, even if they are part of a mock-up, they begin to think of that asset as final. The more detailed your wireframe images, the more comments you will have to contend with about the "look" of them. This can be useful in some circumstances (say, if you're going to produce the art yourself), but it can also be a waste of time if the art is going to be entirely redone.

It's important to note—great UI is really hard. Like gameplay, you never get it exactly right on the first attempt. It requires refining as you see how players interact with it. Your wireframes and user interface will inevitably undergo revisions as you watch players stumble over certain interaction schemes you have designed.

Specifications for Specific Features

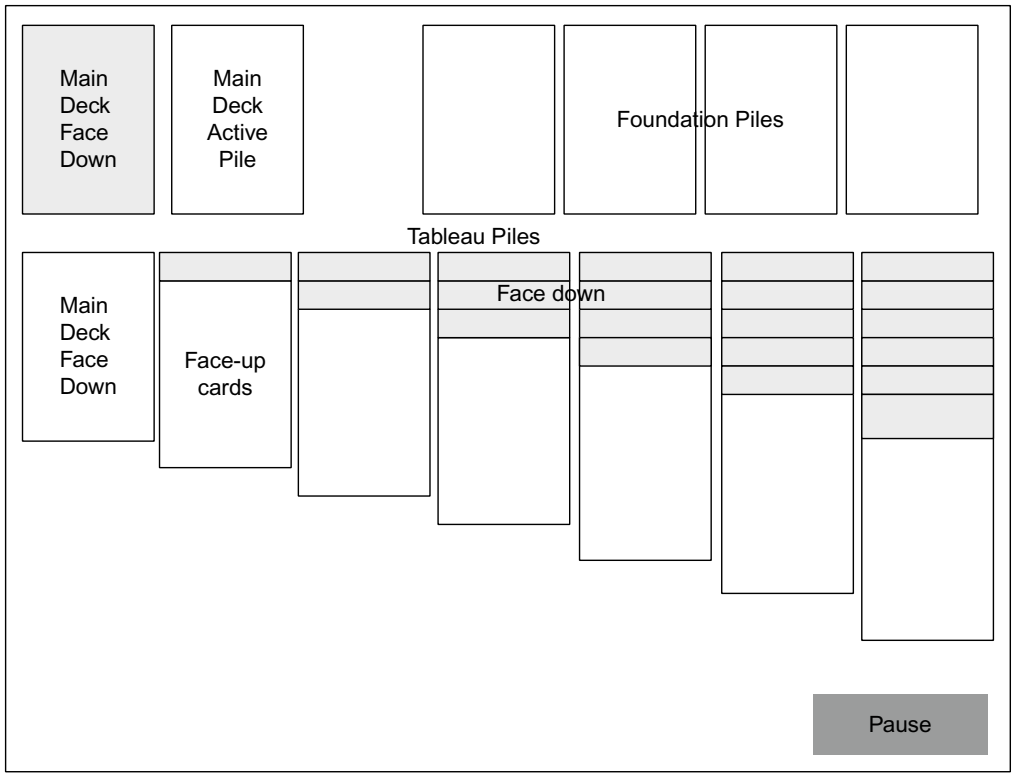
Once you have outlined the basic gameplay, you need to go into more detail on specific features. The specifications for each feature should walk through all of the behavior that surrounds the feature. This includes each possible state of the feature. So if this is a move in the game, what happens if the move is correct, what happens if the move is wrong, what happens if the move fails and what happens if the move collides with another move? You will need to do this for each feature. This is also your chance to define variables for features like the avatar, power-ups and enemies.

Defining the Main Deck Active Feature from our *Solitaire* game might look something like this:

Main Deck Active Pile

When a player clicks on the **Main Deck Face Down** pile, the program flips over X cards and lays them face up on the **Main Deck Active Pile**. The cards should be spaced out slightly so that the player can view the number and suit of each card. The player can only interact with the top card. The player can pick up the **Card** and place it on a **Foundation Pile** or a **Tableau Pile** if there is an available slot (see **Tableau Pile Sorting Method** and **Foundation Piles**

FIGURE
2.9



A basic wireframe for *Solitaire*.

Sorting Method). If a user successfully moves a card off of the **Main Deck Active Pile**, the next card in the stack becomes the top card.

Variables

Cards to Flip = X (this variable should be set in the level configuration. The range of possibilities is from 1 to 5.).

As you craft your game design document, you may find you need other sections not listed here. That's entirely possible, if not likely. Remember the game design document is a living document that changes and evolves with your game. It serves as a record of important design decisions. It should also be thorough enough that you could hand it off to another group to implement if for some reason you become unavailable to work on the project. It is the blueprint for the game.

Guide the Implementation

As development gets under way, the game designer must stay in constant contact with the rest of the team. The game designer will need to work with the team to sort

through issues as they arise. This means brainstorming creative solutions with programmers and artists. Games never progress the way you think they will. Even with a thorough specification document, unforeseen issues will arise as you playtest the game. These issues will require effort from the whole team to resolve. It's up to the game designer to see that the team solves the problems and makes the game better.

Be careful about the features you request. If the designer is constantly requesting features, variables and art they never uses, other team members may get frustrated and balk at future requests. Only request features you believe will truly make the game better. If you begin describing a feature by saying, "Maybe it would be cool if we could have the game do this . . ." that's a sure way to lose the faith of the programmer or artists. Be decisive. If you believe a feature is truly necessary and will make the game better, make the case for it. Help the rest of the team see why it really is an integral feature of the game.

The game designer should not seek to simply impose his or her vision on the game and the rest of the team. Instead, the designer should seek out the opinion of other team members. Use the expertise of your teammates to solve problems.

But at the end of the day, the decision about what direction to take often comes down to the game designer. The designer must have enough confidence in the game design to make a decisive decision.

Playtest the Game

The game designer must continuously play the game as it moves through each iteration. By the time the game launches, you won't be able stand playing the game anymore. But you will have an almost innate understanding of the game system. As the game designer, you must also watch others play the game and learn from their play. You take notes on where they succeed and where they fail, where they have fun and where they grow bored. You must do all of this with lips sealed. You'll desperately want to give the players hints on how to play the game, but you shouldn't. In the end, when the game ships, you won't be there to provide strategic hints to every player. The game must teach the player without your help. To produce a game that effectively teaches players requires playtesting, tweaks to the UI, gameplay and tutorials and more playtesting.

Different game companies have vastly different tools for playtesting, depending on their resources. A company like Microsoft working on a triple-A title like *Halo 3* can afford some very cool tools for playtesting, such as videotaping players, creating statistical maps that reveal where players die and running all manner of regression on their data. Other teams must take a more low-budget approach, such as offering pizza to playtesters and tracking player movements with old-fashioned pencil and paper. In both cases, though, the general philosophy and approach are the same. Observe as dispassionately as possible, record moments and places where players falter and get stuck, then turn this analysis into action items to revise in the game. Then do it all again.

It is always important that the game designers help craft and observe at least a portion of playtests. They need to see where the game breaks; they need the first-hand experience of observing players.

Game designers should begin playtesting their game as soon as possible, getting others on the team and at their company to take a spin at the controls of the game. This will give the designer some initial feedback about what needs to be adjusted. However, as everyone at the company becomes familiar with the game, the designer will need to begin playtesting with fresh eyes outside of the company. This has several advantages. First, people outside the company are less invested in the game. They care less about whether the game succeeds or fails. Your development colleagues are as invested in the game succeeding as you are. This makes their opinion somewhat suspect. They are more likely to see fun because they know how the game is supposed to work and want the game to be a success.

Getting outside opinions also enables you to approach players who are more likely to be in your intended audience. Like it or not, most game companies are not producing games for their employees. If you make games, you are most likely a relatively serious gamer, probably more so than the average casual gamer. Your taste in gameplay will probably run to the more complex and challenging. This makes it very important that you find some actual casual gamers to test your game. You should find your game fun, but you don't necessarily want to design it for yourself. You want a large, broad audience to find it fun.

In your initial brainstorming and concept development, you should have identified the key audience segments for your game. When you have a workable prototype of the game, reach out to members of your audience and ask them to playtest. Bribe them with pizza, cajole them with free games, flatter them with compliments. If you're making a casual game, reach out to your friends who don't devotedly play games. Then reach out to your friends' friends. The further removed from you the playtesters are, the more honest their feedback will likely be.

Before you begin your playtest, write out a script detailing the instructions you will give and the questions you will ask. Your script should introduce the players to the game, explaining what you will be doing, thanking the playtesters for participating and letting them know how long the session will last. Tell them they are free to stop playing at any time or to play as long as they want, but that you will be focusing on a specific aspect which will cover X amount of time. If your game is in the early stages of development, you may need to give people a sheet of paper with instructions on how to play. You can also simply read them the instructions. Use the playtest as an opportunity to test your tutorial text. Following a script formalizes the playtest and makes sure that each player gets the same information.

Before you start the playtest, it usually behooves the designer to have a short interview with each tester. Ask the playtesters what type of games they normally like to play and how often they play. Ask if they have ever played games similar to yours. You want to gather a little information about your playtesters to give some context to their play and comments. If a playtester claims she absolutely loves to play real-time strategy games like *StarCraft*, you can assume she is a hardcore player

well-versed in using the computer to perform complex operations. If he reports playing *Bejeweled* for three hours a day, you are probably dealing with a different breed of player. And if players say they don't play many games because they aren't comfortable with computers, then you'll be able to observe their interactions with the game system in that particular light.

After getting acquainted, sit the players down in front of the computer. Preface the session by telling your playtesters that you want to get their unadulterated feedback. There are no right or wrong answers. You'll be observing their play, and you will not give them much help. You'll largely be an impassive viewer and, in some cases, you may not even answer their questions. Give them as little instruction about the actual game as you can to get them started. With early prototypes missing key features and pieces of the user-interface, it may be necessary to set the stage for the game, giving players some key instructions about how to start up the executable. But ideally, tell them as little as possible. You don't want to taint your playtesting results by telling players how to avoid specific problems or by teaching them aspects of the game that the game should be teaching.

With early prototypes, it can be very helpful to structure the playtest to answer one specific question about the game mechanics, rather than try to address all aspects of the game. Your test could ask whether the main interaction—such as sorting cards—is intuitive or whether players feel comfortable clicking and moving cards with their mouse. As the game advances and UI and art get added to the game, your playtests can begin to test the overall game scheme. In each case, however, keep your remarks to a minimum and let the game do the talking.

Once you have your playtesters playing the game, you'll want to sit back at a vantage where you can observe their play, facial expressions and body language without making them uncomfortable. Sitting off to the side and back a few feet should give you a view to their play. As the playtesters make their way through the game, note moments where they get confused and what is onscreen at the time.

The playtester will no doubt ask questions about specific features, "What does this do?" or "How do I do this?" Don't give the answer. You can keep mum or respond politely, "What do you think it does?" This may entail watching players struggle in certain parts as they try to figure out how to start a level or beat a complex part of the game. Let the playtesters work through it and see if they can solve it. Often, the player will figure it out through trial and error. It's up to you as the designer to decide if finding the solution took a little too much trial and resulted in a little too much error. If players get completely stuck, explain to them quickly how to move past. Knowing when to step in and when to stay silent is a judgment call. If you're making casual games, your players won't want to struggle long, but even the simplest game requires a bit of thought, so give them room to do that.

As players move through the game, do not explain strategy to them or tell them the best way to score points. That is something the game must teach the player. Also refrain from making suggestions like, "Oh it's really fun if you do this. . ." That will immediately throw off the playtest results.

In addition to watching what game elements the players interact with, read their body language and facial expressions. Be empathetic. Watch to see when people really look happy, as if they are enjoying themselves. Watch for signs of frustration passing over their face. If they lean forward, intently staring at the screen, it's a good bet they are engaged. If they lean back, play with one hand and sigh a lot, they probably aren't very engaged. Most people aren't that obvious, but the more you playtest, the better sense you'll have of whether people are enjoying themselves.

Determine beforehand how much of the game you want the tester to play through. This may be a few levels if you just want to test interaction with the core mechanic. It might be a few hours if you want to see how well the game holds up over dedicated play. Have them play enough to answer the question you are testing.

During the playtest, it is a good idea to note data about the user's plays. Note the player's scores, how many times a player must attempt a level, and what power-ups are used. These will help with level balancing later.

After they finish playing, ask your playtesters some directed questions. You can let them tell you whether they liked the game and had fun. That's the first piece of info they'll want to share. But be prepared to ask some more pointed questions to really get at their feelings about the game. Generally playtesters will be generous and tell you they thought the game was fun. Often, they are flattered to have been asked to test the game and want the opportunity to do it again. They may be afraid that if they say they hated the game they won't be asked back. Assuage this fear and tell them you want their honest opinion. But then remember to read between the lines. Ask them how long they think they would keep playing. Ask them what parts they specifically liked and didn't like. Ask them about specific moments in the play. Each playtest will lead to different questions, again depending on the question you are trying to answer. If you are testing the UI, ask them to explain what certain buttons do and see if they really internalized the game flow. If you are playtesting levels, ask them which levels they enjoyed and which they didn't and why.

You can also ask what features or changes they might like to see to the game. But take all feature requests with a grain of salt. Players will often want features that might make the game easier or harder. It's up to the designer to decide if these requests would really improve the game. But if playtesters ask for the same feature, you should give it some serious consideration. There may be something about your game that demands this feature.

Remember that each playtester represents only one data point. You want to collect as many data points as possible so you don't have to put all of your faith in the taste or experience of a few players.

The key to good playtesting is remaining impartial and analytical. This can be very hard. Developers spend a lot of time trying to get a game just right and really want people to enjoy the game. It's tempting to get mad or dismiss players who don't seem to fall in love with the game. But you can actually learn more from people who get frustrated with your game than from testers who blindly adore it.

Refine the Concept

No good game is born fully formed. Every concept and game needs refinement to reach its full potential. The game designer must listen to the concerns of playtesters, team members and publishers and then synthesize all of their comments. From all of this feedback, the game designer must lay out a path for changing the game if necessary. Often, prototypes will show that what you thought would be fun isn't. Instead, some minor feature actually provides the joy in the game. In these cases, the designer must be willing to adapt his or her vision to the realities of the prototype. You'll make dozens of false starts. But if you're willing to quickly prototype, refine and throw out ideas that don't work, you'll eventually come to something great.

At other times, the game designer will need to stay the course if he or she believes the game is on the right path and will become fun with the implementation of certain features. This means the game designer must not just play the actual prototype, but be able to extrapolate and play the imagined game in his or her head. This can be very hard, but it's essential for you to be able to look at a game and begin to play out the possibilities in your head.

This is the sort of skill that requires a fair amount of practice and experience. When designers start out, they tend to be inflexible about their designs. The more you work with your team on different games, the more you will come to trust and utilize the opinions of others.

Just remember that your game and concept will need revision. No book reads perfectly on first draft. The rough cut of a movie is often a complete mess in dire need of editing. Games are the same way, possibly more so. They require an iterative process to build. A game is an interactive conversation between the game system and the player. The two sides need to react and dialogue with each other. During development, the designer needs to dip into the part of the game system for that conversation, listening to what the player has to say and responding with the appropriate refinements.

Design the Levels

Once the game system has been laid out, the game designer will need to build levels. Level design is the meat and potatoes of game design, where the designer crafts the moment-to-moment user experience. In level design, the game designer finally starts using all of the knobs and variables they asked the programmers to build. Tweaking these variables in different combinations, the game designer hopes to make a fun experience with just the right amount of challenge. The game designer must be willing to make levels with only a partial set of tools while waiting for the game to be fleshed out. Often these levels must be trashed later as new features become available. But constantly exploring the potential and limitations of the game system as it is built will help keep the game on track and focused on the most important features—those that actually make the game fun.

As the features are fully built out the game designer will craft the arc of the player experience. This means taking into account player learning curves. This can be particularly tricky with casual games, as the audience is very broad and has a wide range of abilities. As the designer, you'll need to design a path through the game to suit all types of players, from expert gamers to the first-time player uncomfortable with even using a mouse.

Level design is an art unto itself, to which you could no doubt dedicate an entire book. Here are some general guidelines and approaches for designing levels:

- Be empathetic
- If you can't beat your level, then it's waaaaaaaay too hard
- Design for the general audience, not the hardcore
- Ease players into the game
- Don't forget to challenge players
- Build levels around a central concept
- Teach players to play the level
- Give players room to explore
- Occasionally break your own rules (carefully)
- Create a plan
- Vary your levels
- Refine, play and refine
- Playtest

Be Empathetic

When first time designers sit down to try their hand at level design, they almost always produce incredibly complex, punishing levels. If all you had to go by were levels crafted by beginner designers, you would conclude game design was a blood sport. First-time designers mistakenly take level design as a contest between player and designer. The ethos seems to be, "But can you beat this!?" This is the wrong approach. Level design should not be a struggle between designer and player. No, the game designer should offer a helping hand to guide the players through the level, leading them toward enjoyment. At times this means giving the player challenge. But at other times it means letting the player succeed. You want the player to have a good time, not simply a hard time.

So how do you know if they are having a good time? That's a much harder thing to measure than success and failure rates. You need to employ your sense of empathy. You need be able to put yourself in the position of the players and see the game through their eyes. They don't know all of the tricks and secrets hidden in the level. The game designer creates the basic concept of the level. But then you must ask yourself, what would the players like to do? What will make them enjoy the level?

Players want to be challenged, but they don't want to be punished. They want to feel successful. They want to win. Your challenge is letting players do that without letting them see that you let them win.

If You Can't Beat Your Level, then It's Waaaaaaaay Too Hard

On multiple occasions, I've had new level designers tell me they cannot beat their own levels. That's unacceptable. Designers know every surprise the level will throw at the player. They know by heart where to step, what aliens will pop out, what gems to swap, the order in which customers should be served. They have mental access to all of the hidden information of the game. So if you can't beat the level, imagine how long it will take your average player to beat it. The average player will be totally out of his or her depth. Players can only discover this hidden information by the arduous process of trial and error.

As a general rule of thumb for casual games, I feel a designer should be able to beat early levels in a game with one arm tied behind his or her back. As the game progresses and gets more difficult, perhaps some concentration is required. But unless you are designing the boss levels in a hardcore console triple-A title, you really shouldn't be losing all that much. And even then, you need to be able to beat it to ensure that it is beatable. It's entirely possible to design a level that is unwinnable by setting a goal score too high or a jump too far. You have to be able to play through your level from beginning to end and prove it's winnable.

You designed the level. You know where all of the bombs and traps are hidden. For this reason, the level is easier for you than anyone else. Don't tune the level for your own enjoyment. Tune it for the player's enjoyment.

Design for the General Audience, not the Hardcore

The hardcore players always have the loudest voices. They are the ones on forums complaining a game is too easy. They clamor for greater and greater challenge. But you have to remember: they are a minority. Granted, they are a vocal minority, but a minority just the same. You need to take their demands into consideration, but like democracy, you need to answer the majority's needs first. The large majority of your casual games audience is not hardcore. They want challenge on the order of an invigorating hike, maybe some light scrambling over rocks. They don't want to scale a 1,000-foot cliff. Covered with ice.

Design your levels to please and thrill the general audience with intermediate skill levels. This is the audience that will make your game a success. You want them to be happy.

Ease Players into the Game

Starting any game represents a big challenge. You have to learn the rules. You have to suss out how to control the game. You have to take in the game narrative and

world, all while navigating a user interface tailored to the specific game. This puts a lot of cognitive demands on a new player. If you have super-challenging gameplay on the first level, you'll likely scare players off.

Ease players into the game. Introduce one element at a time. If your game has a lot of power-ups, dole them out one at a time. The same goes for enemies. If your control scheme is particularly complex, break it into pieces to enable players to master the different components. You want to make getting into the game as smooth as possible. Since players must spend so much energy learning the game in the first few levels, don't overwhelm them by making them learn tricky levels too. Let them get their feet under them before swiping them out.

Don't Forget to Challenge Players

While it's crucial to ease players into a game with a gentle initial learning curve, don't forget to challenge your players. Without a bit of challenge, the game will lose all sense of vitality, devolving to no more than an exercise with some very idiosyncratic constraints. Sometimes challenge means actually making players lose a level, just to remind them they are playing a game and keep them on their toes.

Wade Tinney, the co-founder of Large Animal Games, once related a story about how they used dynamic difficulty adjustment in one of their games. The game's algorithm monitored how many levels a player won in a row and modified the difficulty accordingly. If the player won several levels in a row, the game would make the next level harder, making it quite likely the player would lose the next level. After a loss, though, the game would adjust and make the game a bit easier again. In playtesting, they found this kept players intrigued and playing the game. If they won every level, they tended to stop playing sooner.

Build Levels Around a Central Concept

Before you start laying out all of the variables in a level, spend some time thinking about what concept underlies the level. The best levels are concise and clean. They focus on one central idea, running through different elaborations on that idea. In a color-matching game, the core idea of a level might be managing a preponderance of one color. A puzzle platformer might require a series of similar jumps.

A level is like a great pop song. It has a central melody that you can build variations around, but underlying the whole level is one catchy hook or idea. This gives your levels clarity and focus. Focusing on one idea will help you find the core element of fun in the level and let you polish that to a shine.

Teach Players to Play the Level

A level should offer signals to the player that indicate how to approach the level. You want to guide the player into the experience. If your level is all about a particular type of move, give your players space to try out the move and learn to master

it before placing them squarely into danger. So if your level requires wall jumps, give the players a safe place to try out wall jumps before you have them do those wall jumps over spikes or bombs. Otherwise they will repeatedly die and get frustrated.

Set up general patterns and rules that players can learn to "read." If your game requires a particular type of wall jump, set up similar structures for the wall jumps in the easy and hard parts. In this way players will begin to recognize what action they should perform to pass difficult spots. Players will begin to read the level and understand what actions to deploy.

Give Players Room to Explore

Levels that introduce a new feature should focus on teaching the player the basics of using the feature. Completing the level should require the player to interact with the new feature in some basic and straightforward way. You want the players to see the basic utility of the item, be it a power-up or new move. Forcing them to use it will help push them to use the feature and break them out of their established playing pattern.

In the next level, use the feature again, but open the play up to let the players explore other aspects of the feature. If it's a power-up, give them the chance to explore the different facets and ways they could use the power-up. The first interaction teaches them the basics; the second teaches them to creatively apply their new tools.

Occasionally Break Your Own Rules (With Care)

Once you have set up patterns in your game, you can break your own rules. Do this with care. You don't want to call into doubt the entire system of meaning you have created for players. But the occasional shift in the patterns of the game can surprise and delight your player. It keeps the gameplay fresh and enables the players to feel they have creatively applied the mechanics of the game.

Breaking your own rules should be done with care. You don't want the game to seem arbitrary. You still want it to read like there is a logic to the game.

Create a Plan

Just as you would outline a novel or screenplay, it's crucial to outline your level structure. Lay out where you think you will introduce different concepts, power-ups, enemies and content to the game. Make sure this level plan fits with the overall narrative and goals of the game. Make a big spreadsheet or list with an entry for every level in the game, detailing what elements will be used. Outlining the whole game helps you craft the overall experience of the game, progressing the game in complexity and difficulty. It will also help you see which elements you are using too often and which you are ignoring.

Vary Your Levels

This may seem obvious, but it's not easy to do. It's very easy to fall into a pattern you repeat over and over. As you produce levels, establishing consistency between levels is important. You can get a lot of mileage out of creating levels that are variations on a theme. However, you don't want those variations to feel repetitious.

Where possible, get multiple designers to contribute levels to the game. Different designers inject fresh perspectives into the game, with each designer finding new ways to use the level variable tools to create a slightly different experience. If you do use multiple designers, though, one lead designer should set some basic parameters and target goals for the all levels that the individual level designers follow. This lead designer must also play through all of the final levels to make sure they are balanced and consistent.

Refine, Play and Refine

Lay out the basics of the level. Play it. Make refinements. Play it again. Make more refinements. Play it again. Keep this up until you have polished the experience of the level. It will take a while. No one gets it right on the first stab. Like an author reworking a paragraph to get just the right syntax, a level designer reworks and plays a level to craft just the right experience. This is the most important part of level design. Sometimes this will require you take a break from the level for a day or so then come back to it and try it again. A little distance can you give you some much needed perspective on your work.

Playtest

Playtest, playtest, playtest. You can play and refine your level until you're sick to death of it. But you'll never be able to capture the fresh perspective of a new player without playtesting it with others. Get outsiders to look at it and carefully note how they play it, where they have fun and where they don't. Then modify the level to draw out the fun parts and reduce the not-so-fun parts.

Becoming a Game Designer

There are many ways to become a game designer, though no sure path. Programs exist that teach people the fundamentals of game design and developing video games. These range from pre-professional programs like Full Sail to more arts-oriented programs like University of Southern California's Interactive Media programs. However, like film schools, these programs can train you in the fundamentals of game design, but it's up to each person to put what they've learned into practice. And in the end, practice is the most important part. If you want to be a game designer, you have to start making games however you can.

Use available game engines and level editors to make maps for your friends. This teaches you to work with game design tools and tweak variables while trying to

build a fun experience. Again, level design is an essential skill for any aspiring game designer. Creating your own levels is also a good way to start building a portfolio that shows off your design chops. Many employers looking for game designers ask applicants to build sample levels using one of the company's own level editors. This gives employers a good view into the type of game designer the applicant will make. Will they punish the player with impossibly hard levels, mistaking difficulty for fun? Will they make levels that are too easy and do not offer any challenge? Will they nurture the player along with levels that offer compelling and creative challenges?

Practice designing games nonstop. Play games, then imagine how you would modify them to make them better (or worse). Start practicing with board games. Read the rules to board games and consider how the game designers framed the game for you with their written description and rules. Contemplate a change you would like to make to the game and then modify or add rules that you think will produce that result. For example you might try to make a game go faster. Consider what rules you would have to change. Can you shorten each player's turn? Can players take turns simultaneously?

Also try taking out key rules to see where the game breaks. This can be very informative. Games are like buildings framed by rules. Sometimes taking out just one beam causes the whole thing to collapse. Other times, the game can withstand the removal of a number of rules before it ceases to be playable. Take soccer (or football, as the civilized world likes to call it) for example. On their Web site, the Fédération Internationale de Football Association (FIFA) offers a document that spells out the official "Laws of the Game." This document is 138 pages and covers everything from the size of the field to the duration of the game to player equipment.[3] How much of all of that, though, is really essential to soccer? How many rules can you strip away and still call the game you're playing soccer? Does offsides really matter? Does the number of players? Probably not. Does prohibiting the use of your hands when touching the ball? Most definitely. The game would cease to be football if every player could use their hands to carry the ball. Taking out rules will teach you to boil a game down to its primary core mechanic. This is an essential skill for casual game design.

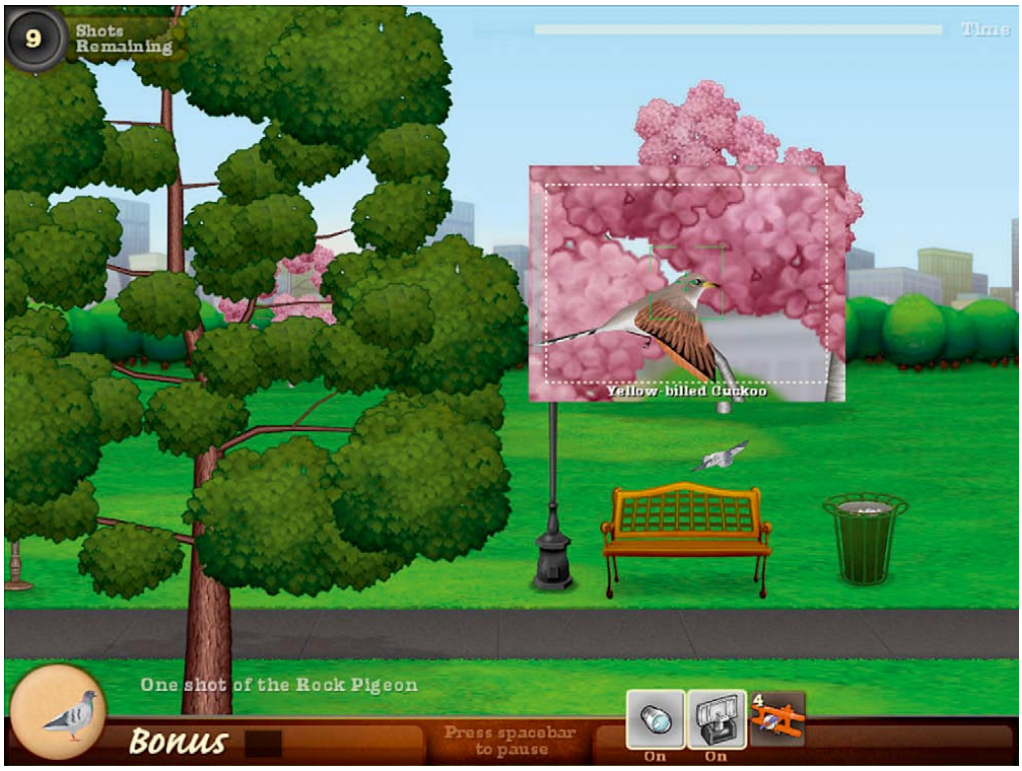
Don't stop with games though. Try giving goals and rules to different activities to see how you might make them into games. We already treat a lot of activities like games. How quickly can you chop three carrots for your salad? Can you walk to work without stepping on any cracks? How quickly can you sort all of your e-mail into folders? Why? It teaches you to look for obvious rules and play structures. The more obvious and intuitive your rules, the better casual games they will make. Plus, some of these activities may even make good video games. A lot of casual games, like *Cooking Mama* and *Snapshot Adventures,* are reinterpretations of familiar activities. *Cooking Mama* (Figure 2.10) has built a hit franchise around preparing meals.[4] *Snapshot Adventures* (Figure 2.11) makes a game out of bird watching and taking photographs.[5] And even if you don't make any of your daily activities into video games, you will learn to look for fun in unusual places.

[3] http://www.fifa.com/mm/document/affederation/federation/81/42/36/lotg_en.pdf
[4] http://www.cookingmamacookoff.com
[5] http://www.largeanimal.com/games/deluxe/snapshot-adventures-secret-of-bird-island

FIGURE
2.10

Cooking Mama **transforms cooking into a game. (© Majesco Entertainment)**

FIGURE
2.11



Snapshot Adventures **recognizes the implicit game like qualities of bird spotting. (Created by Large Animal Games. Published by iWin)**

Beyond these exercises, start making your own paper games. Many games start their lives as prototypes sketched out with pen and paper. Paper prototyping enables you to quickly test ideas to see if the general mechanic is fun. Building non-digital games, from board games to card games to new sports, exercises your game design muscles and forces you to confront general problems that you'll encounter again and again in game design, from creating core mechanics to closing player exploits to balancing statistics.

If you want to be a game designer, you have to make games. You can't be a rock star without practicing the guitar, and you can't become a writer without sitting down and actually typing out stories. The same is true of games. Like most arts, game design is one part inspiration and nine parts implementation. Build a portfolio that shows you understand games in all their aspects. Design levels, board games, card games and make up new sports. Each game project will teach you something new about designing games.

Becoming a Professional Game Designer

Traditionally, playtesting has been a way into the game development industry. All video game companies need people to play their games and test them not only for bugs, but for playability. For this, they hire playtesters. This helps people get their foot in the door and gives them the chance to work with programmers, artists, producers and game designers. Some of these playtesters go on to become producers or game designers.

Experience with software production in general can be a good way into the video game industry. At its core, making video games is just software development. The product just happens to be really fun. Working as a project manager or producer in software development gives you many of the same skills you will need to bring to games—from conceiving and specing projects to working with programmers and UI designers.

However, the best way to become a game designer is to start making your own games. One of the great things about developing casual games is the low barrier to entry. Not only can an individual or a small team make a Web game, but they can also find an audience on the Internet. In this sense, becoming a game designer is not unlike hanging your shingle: make a game and put it on the Internet. Monetizing your work can be hard. But there are some good channels, from Kongregate to the iPhone to the portals, that make self-publishing possible.

Why Be a Casual Game Designer?

So why do game designers work in casual games? Wouldn't they all rather be making big budget console games? After all, most game designers are likely gamers used to playing console games. No doubt, some casual game designers do long to make big-budget AAA console games. But there are many who prefer working in casual games. A number of things make creating casual games attractive.

First, casual games are small. They can take anywhere from a week to a year to develop. This is a stark contrast to the multiple years necessary to develop a console game. The short development time for casual games gives you the opportunity to work on more games. This prevents you from getting as burned out on a game. New projects offer fresh challenges and the chance to try out new ideas.

Designing casual games for a broad audience also means your friends and family may actually enjoy playing your game, even if they are not hardcore gamers. Casual games have the chance to reach a really broad audience in the way that only the best-selling hardcore games do. So while the *Grand Theft Auto* franchise has made more money than any casual game, more people have probably played *Bejeweled* than *Grand Theft Auto*. Casual games with core mechanics designed specifically to appeal to as many people as possible have the chance to become genuine pop culture phenomenon, spreading virally over the Internet and gossiped about around the water-cooler. That potential to reach a mass audience excites designers.

Casual games also offer more opportunity for innovation. Because they are small and less expensive to make, game designers can take more risks. They can try out strange new mechanics or base the game around off-beat themes and art. Casual games have their tropes and limitations, but the landscape often seems far more open than the console market. Even console games hailed for their innovation, like *Halo*, mostly just refine known mechanics. *Halo* very cleverly improved upon the health meter and inventory system, but in the end it hews very closely to first-person shooter conventions. The game designers innovated within an established genre. Contrast that to the puzzle game *Crayon Physics*, in which the player draws objects which are transformed into physical objects within the game.[6] *Crayon Physics* was created by the one-man development "team" Petri Purho, who designed, programmed and created the art for the game. The game grew out of a series of game design experiments by Purho, in which he designed and built a new game every seven days. *Crayon Physics* feels much more experimental and far afield than most any console game. Obviously, a lot of innovation happens on console games, but the conventions of established genres and the cost of doing business make it very risky. Smaller teams and budgets allow for more quick experimentation.

Summary

You want to be a game designer? Well, you've probably already taken the initial steps. Start with paper games, card games and physical games—things you can do all on your own. Then move on to video games. Game designers appreciate good design, whether it is in a card game or a video game. The art of game design is not about pixels or programming after all. It's about crafting fun experiences out of rules. The rest is practice and implementation. Like any art, to become a game designer you must design games. You must make games.

[6] http://www.crayonphysics.com

CHAPTER THREE

Play Is the Thing

A firm grasp of games starts with a good understanding of play. If you don't understand the attraction and enjoyment of play, it can be hard to fathom the attraction of games. Even more importantly for a burgeoning game designer, it will be hard to create activities with which people want to engage. The playful activity is the elemental building block of games. Almost all games have at their heart an activity which you could describe as playing. This playful activity often resides beneath the level of core mechanic. It supports and animates the core mechanic. In a game like *Bejeweled*, it's the playful activity of matching, which is then structured by a match-three game mechanic. In a sport like soccer, it's the kicking around of an object that supports the mechanic of kicking the ball into goals. Aspiring game designers must be able to identify these playful activities, pick out the fun ones and build games around them.

Play is the act through which we experience not only games, but much of the world around us. As children we use play to explore the world, sussing out everything from basic physics (if I throw a ball up in the air, it will inevitably come back down) to social relations (if I call that girl a name, she won't like me). Through play, we test the limits of our own actions and their impact on the world. This behavior continues into adulthood, though it may be less apparent and we may call it by a different name. There are obvious ways adults continue to play. They play golf; they get together with friends to play poker; they hold *Wii Tennis* tournaments. Then there are less obvious ways. Flirting is a form of play. Rough-housing with your son, painting your face orange and blue and rooting for the Mets, wildly cheering and calling out for "Freebird" at a concert are all forms of play. Even spinning your wedding ring on the conference room table during a meeting is play.

New parents engage in a form of mutual play with their infants, in which both baby and parent poke and prod at each other, exploring and establishing the ways in which they can interact. A father picks up his infant daughter. The baby kicks her legs wildly about, flailing below her father's arms and whimpers. The father repositions his daughter on his hip to try and get the girl to stop kicking. He swings her around and bounces her on his knee. They both look at each other with a moment of incertitude, as if asking, "Is this right? Have we reached a moment of mutual comfort and happiness?" The father tickles the girl and smiles. The girl smiles back

and the two reach a moment of attunement. She flaps her arms in excitement and he hoists her up above his head. The father makes note that next time his daughter is unhappy he should place her on his knee, bounce her and maybe throw in a few tickles. Through this playful interaction, both father and daughter begin to establish a grammar for interacting.

The Dutch anthropologist Johann Huizinga opened *Homo Ludens*, his seminal book on play, by declaring, "Play is older than culture, for culture, however inadequately defined, always presupposes human society, and animals have not waited for man to teach them their playing."[1] Huizinga points out that you can see all of the same elements of human play in the playful nipping of dogs or the rough-housing of lions. In these interactions, Huizinga saw play taking a place of primacy. He argued for play as an elemental form of interaction and a contributor to culture. He saw lions and boys alike playfully wrestling their brothers and sisters as a means of experimentation and socialization.

Play is semi-structured. There are no explicit rules to play. You can't write down the rules to rough-housing with your siblings. But most people develop a good sense for when the lines are crossed. They develop this sense through multiple play sessions, poking and prodding others to see what is acceptable and what isn't. When you wrestle your brother and twist his arm and he yelps in pain or shouts angrily, "Okay get offa me!" we cache that response and alter our mental guidelines for the play activity. Instead of rules, play operates along general guidelines that define how participants should interact and behave. Play is in some part defined by etiquette that develops out of repeated play sessions. The more formalized that etiquette and those guidelines become, the closer play inches toward structured games.

This makes play an incredibly fundamental part of our lives, one that we never escape. If you were to ask a group of adults if they "play," I imagine many would respond in the negative. Or they might admit to playing a few specific games, but few would declare they routinely engage in unstructured play. But most likely they play much more often than they realize. We all have favorite activities we rely on to kill time. Tapping feet on the subway, spinning pens around our fingers during meetings, counting cars as we walk, flirting with a co-worker. We just don't think of them as play. Perhaps we call them killing time, or possibly the activity is so natural that we don't even realize when we start doing it. The need to play is innate. Understanding how to tap into that primal urge to play gives game designers powerful tools with which to engage players.

The Liminal Moment

So play is ubiquitous and informs learning. How does that impact game designers? Most forms of play grow stale after a while. How long can you keep stacking up blocks just to watch them fall? Kicking a ball back and forth can get boring pretty

[1] Johann Huizinga, *Homo Ludens,* Beacon Press, 1971, p. 1

quickly. That's where games come in. If you give yourself the goal to build the highest tower you possibly can out of your blocks, the activity suddenly takes on new directed shape. You have something to strive for and something to measure yourself against. Every time the blocks tumble back to the floor, you know what you need to do next: Add at least one more block to your next stack. Even this slight goal and loose structure stretches out the entertainment that can be found in playing with blocks.

There is a liminal moment in play when a playful activity transforms into a game. The play provides the spark of fun. The game provides the framework for long-term interaction. It's the game designer's job to usher the player through these liminal moments, to transform moments of free play into structured games. Designers do this by providing the rules and goals that define the game. In doing so, they build a system that supports repeat play.

This moment of transformation should be of particular interest to game designers. For hardcore game designers, play can be buried under an accumulation of rules. For game designers looking to build casual game experiences, the closer they can keep their game to that liminal moment, the better. Many casual games hover close to the playful activity that inspired the game.

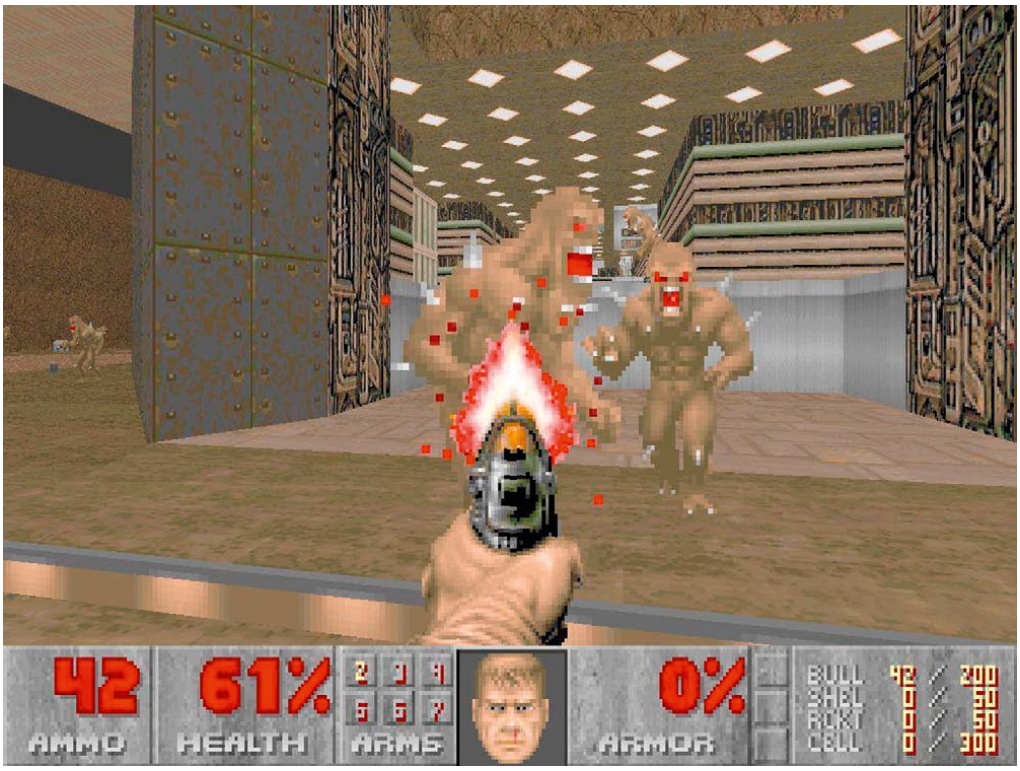
The Rush to Complexity

Hardcore players expect complexity and depth from their games. They play a lot of games. They have learned the basic rules and interactions that govern most games. Their familiarity with basic mechanics means that new games must offer fresh complexities to provide a real challenge. Video games are dominated by genres that rely on familiar mechanics. Just think about how many games for consoles are first-person or third-person shooters. Hardcore players are familiar with the mechanics, so the game must either provide a new mechanic or add other elements of complexity on top of the original mechanic. To meet the ever-increasing abilities of hardcore players, game designers must add new features to the game play. Sometimes this takes the form of a new game mechanic. Sometimes it manifests itself as more resources to manage.

For example, consider the difference in complexity between a game like *Doom* and *Ghost Recon: Advanced Warfighter* (*GRAW*). *Doom* is one of the seminal first-person shooter games. It helped establish many of the conventions of the first-person shooter, from the point of view of the player to the practice of navigating a large open space. At the time, *Doom* was quite complex. It required the player to quickly navigate the semblance of a three-dimensional space with one hand and to take aim at oncoming enemies with the other. All the while, you managed your remaining health, as well as various weapons and ammunition. But at the same time, the game was incredibly intuitive. The core mechanics of pointing at things and shooting felt familiar from the first play, as if the game tapped directly into our memories of childhood water gun battles. In some ways, *Doom* fulfilled a long-held desire among gamers: The video game would actually put you in the position of the

hero pointing the gun at the bad guys. These navigation and inventory challenges combined with the popular shooting mechanic made *Doom* a megahit in the 1990s. But play *Doom* today and the game feels incredibly simple next to modern first-person shooters. Next to *GRAW*, *Doom* looks almost, well, casual.

FIGURE **3.1**



***Doom* taps into the basic playful activity of pointing and shooting. (Doom® © 1993 id Software LLC, a ZeniMax Media company. DOOM, ID, and ZeniMax are registered trademarks owned by ZeniMax Media Inc. All Rights Reserved)**

The lineage is obvious. They both offer the same perspective and core mechanic of moving, aiming and shooting. But *GRAW* greatly increases the complexity. In *GRAW*, the player has greater camera control, enabling the player to look up and down. This alone greatly increases the complexity of the game, making it feel more like a full 3D space. *Doom* is limited to one plane.

Everything about *GRAW* is bigger. The maps and levels in *GRAW* are bigger, sprawling over vast areas. The player can pick any number of paths through the space. The inventory of weapons and ammunition the player must manage is larger and more varied. These larger spaces and inventories mean the players must constantly sort through more options as they play.

On top of these complications to the basic first-person shooter gameplay, *GRAW* stacks squad management. In *Doom*, you are simply responsible for your lone,

nameless space marine as he battles his way through an alien-infested space station on one of the moons of Mars. In *GRAW*, you guide an entire squad of soldiers. You have your own avatar you control. But on top of that character, you also give orders to other characters, picking out locations and targets for them to reach and attack. In effect, you play a first-person shooter with a strategy and resource management game layered over the FPS action.

FIGURE
3.2

***GRAW* layers tactical strategy and resource management on top of the FPS mechanic, making the game more complex. (WikiCommons[2] © Ubisoft)**

This combination of game mechanics places greater demands on the player, making the game more challenging. It adds depth to the game. The game takes longer to master because there are more systems to learn and control. There are more opportunities for decisions and a greater array of strategies to choose from. The players can barge into a firefight and try to pick off everyone. Or they can send members of their squad to take care of the action.

This added complexity also pushes the game further away from the base element of play. As the FPS game has gotten more complex, the new functions have served to bury that original playful element of *Doom*, aiming and shooting. Pointing and shooting is, of course, still very much at the heart of *GRAW*, but there is an abundance of other stuff the player must also manage. This added complexity and depth delights hardcore players. But very often it just obfuscates the fun for casual players. Casual players tend to be attracted to the base element of play—that element of play the game initially promises to deliver.

[2] http://commons.wikimedia.org/wiki/File:GRAW5.jpg

The Push toward Simplicity

Casual games grow out of a philosophical approach to game design. Unlike games that can claim a common game mechanic, like the first-person shooter, there is no definitive game from which all casual games have sprouted. Instead, games wind up being considered casual games in one of two ways. The first path explains why a game like *Doom* which used to seem hardcore now feels almost casual. Games evolve and yesterday's hardcore game begins to seem awfully simple and casual next to today's newer more complex fare. The casualness versus hardcore nature of a game is entirely contextual and changes as players evolve and grow more skilled. The other way is more intentional. Game designers look at the state of games, play and player skill and attempt to craft a game with a simple, understandable game mechanic at its core. She pushes her game toward simplicity and the core element of play in the game mechanic.

Casual games eschew complexity in favor of simplicity. The game is simple and boiled down. Sometimes, casual games barely cross that fine line between play and game. Instead of asking the player to navigate a complex set of rules, the game focuses primarily on delivering the promise of play initially held out by the game. When creating a casual game, the designer should focus on delivering the pointing and shooting, as opposed to the pointing and shooting with resource management and strategy game laid on top of it. The casual game should be closer to the activity of kicking the ball back and forth than the game of professional soccer laid out in 138 pages by FIFA.

Every game has a fundamental element of play. If you strip away all of the rules, you'll find it. You'll get back to the moment where the game transforms back into simple play. At that moment, you can see where the game most likely originated. What was the impulse that sparked the game? Basketball, with all of its complexities and strategies, with its bouncing balls, dunks and three-pointers, basically boils down to the impulse to throw an object into another object. Take away the jumping and the geometry, and checkers is about gobbling up your opponent's pieces. Chess is about cornering your opponent. How you accomplish this feat is complicated by all of the individualized movement patterns of each piece. At its core, a game like *Doom* is about pointing and shooting. In many ways, the pleasure and attraction of the game stems directly from playing cops and robbers as children, of putting your hands together like a gun and yelling "bang" as your friend walks through your sites.

This is the sort of straightforward, almost elemental play that most people seek when they decide to play a casual game. This is true of both self-identifying casual players and dedicated gamers electing to play a casual game. They want a short, pure burst of play. They want something they can learn and play quickly—a game with simple, stripped down core mechanics. Playful activities exist without the hindrance of complex rule sets. In their lack of complex rules, they must by definition be relatively simple and straightforward. To add complexity, you must add rules. This in turn makes the game harder to learn and master. Of course, if these players

really like the game and get better at it, they too may demand more depth and complexity. Eventually they may demand that the casual game evolve in complexity to match their growing skill and understanding of the game. But at this point they will be looking for a hardcore version of the casual game they started playing.

Patterns of Play

Casual game designers must walk the fine line between play and games. To effectively straddle that line, they must understand play. Play only seems unstructured at first glance. Upon closer examination, we can see definite patterns of play emerge. A number of scholars have dedicated themselves to examining the different manifestations of play. Huizinga got the conversation started, and scholars, academics and game designers alike have continually found new ground to probe.

The National Institute for Play takes play very seriously. They believe play has transformative qualities for children and adults. This non-profit is comprised of a number of psychologists and researchers looking at the effects of play on cognitive and social development.

They outline seven main types of play in an effort to create a holistic framework for studying play.[3]

- Attunement play: This very simple interaction produces measurable brain activity and powerful emotional connections. A mother makes eye contact with her child, and the child smiles. The mother smiles back, reinforcing the baby's smile and bringing the two into attunement.
- Body play and movement: We understand much of the world through our own bodies. From babies flailing their arms as they learn motor control, to kids tossing snowballs at each other, to adults learning the foxtrot, we explore our world and simultaneously entertain ourselves and explore our limits through movement.
- Object play: Much of our play is inspired by objects. We pick them up, shake them, turn them over, spin them, throw them. Through these interactions, we explore the limits of the object. How much stress can it take before it breaks? How far will it fly with a good heave? Babies do it. Grown-ups do it. Object play can also lead to problem solving as you learn to take apart an object and put it back together.
- Social play: Social play runs the gamut from rough-housing to more complex rituals like the Dozens, in which participants trade ribald insults. Animals also engage in social play. This play serves important socialization and cultural functions.
- Imaginative and pretend play: Playing house and other games of pretend may seem like child's play, but it does wonders for our creativity, at a young age and even later in life. It also helps kids build their own mental models of the world.
- Storytelling-narrative play: Stories are one of the atomic units through which we organize our understanding of the world around us. We group our lives and our

[3] http://www.nifplay.org/states_play.html

days into stories about the characters and events of our lives. Shaping them into stories helps give them meaning and lets us make sense of the random events of life. We largely learn to create those stories through play. We take on roles, we make up tall tales and we share them with each other.

- Transformative-integrative and creative play: We use our fantasy play to spark creativity and imagine new possibilities for our play. We imagine all of the different things we might do with a ball and test them out. Sometimes we use these ideas in our lives outside of play as well. Group brainstorming often takes on elements of creative play.

And the National Institute for Play isn't alone in categorizing and studying play. Scholar Brian Sutton-Smith has written very eloquently about play and games. In his book, *The Ambiguity of Play*, Sutton-Smith examines the role and perception of play in culture. Like many, Sutton-Smith acknowledges that play is hard to define, that play is often ambiguous and variable. Playful behavior can be found in many of our daily activities, from traveling to gossiping to reading. We may infuse these activities with playful competition, comparing who has been more places. In gossiping, we cheekily denigrate people behind their back. Or we may simply go through the imaginative exercise of envisioning and identifying with fictional characters, playing out their lives in our heads.[4]

At the outset, Sutton-Smith presents a list of activities he considers to be play and categorizes them into groups not unlike those presented by the National Institute for Play. Sutton-Smith lists broadly recognizable categories. But it is the activities and behaviors that he includes in each category that really provoke interest. He takes a broad view of play and sees playful behavior in an incredibly diverse set of our everyday activities. Many of them you might not consider play at first. But when grouped together with other similar activities, you can see how each does indeed possess elements of play.

Sutton-Smith's categories of play and a few of the examples he includes:

- Mind or subjective play: Here he gives the example of daydreaming, fantasizing and similar activities.
- Solitary play: hobbies, collecting, reading, building models
- Playful behaviors: playing tricks, playing by the rules, playing at something
- Informal social play: partying, joking, getting laid, babysitting, intimacy, amusement parks, speech play like riddles, gossip and jokes
- Vicarious audience play: watching television, films, Renaissance fairs, spectator sports
- Performance play: playing music, playacting
- Celebrations and festivals: birthdays, roasts, weddings, balls, masquerades

[4] Brian Sutton-Smith, *The Ambiguity of Play*, Harvard University Press, 2001, p. 3

- Contests (games and sports): athletics, golf, gambling, board games, card games, martial arts
- Risky or deep play: caving, hang-gliding, sky jumping, sport climbing

Tapping Play for Games

Thinking about different categories of play can be useful for game designers. It can give you a starting point for a new game. Looking at activities that already engender playful characteristics can spark directions for new games. This is true for everything from physical games to card games to video games.

There are obviously many different ways to categorize play. But these are some useful categories for starting to think about game design:

- Physical play
- Playing with others
- Playing with things
- Playacting
- Daredevilry or pushing your luck

Physical Play

Obviously many of our sports grow out of activities such as running, jumping, throwing, hitting and spinning, from simple games like the 100-meter dash to more complex games like American football. But these sorts of activities also inform smaller casual games, like arm-wrestling and thumb-wrestling. They also inspire the focus of video games from platformers like *Super Mario Bros.* to fighting games like *Tekken*. Mario's name was originally Jumpman, after all.

These activities have an immediacy that stems from their primacy. As children, we first explore ourselves and the world through physical play, making many of these activities intimately familiar. This makes building games out of them relatively easy as everyone already understands what they need to do. They are enjoyable, satisfying activities in and of themselves. Game designers can use them as building blocks and combine them into new structures. For players, the fun and challenge comes in mastering the actions in the new contexts.

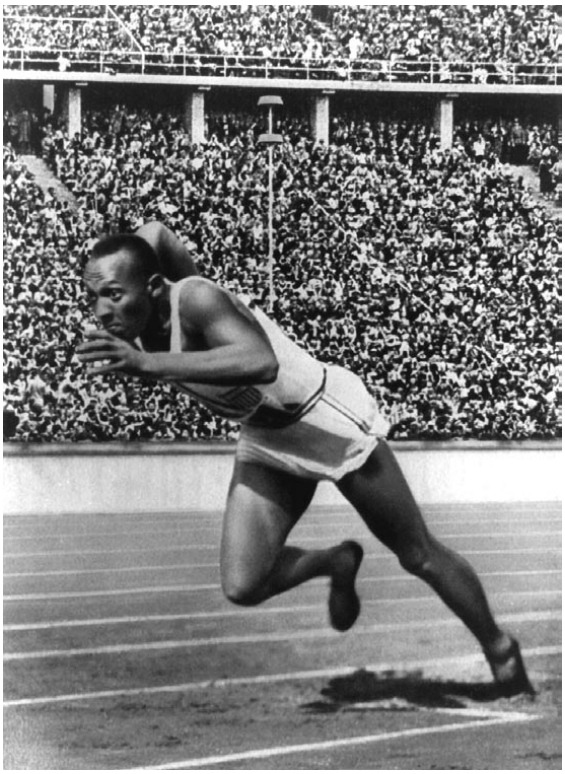
One of the great advantages of physical play is quantifiable outcomes. You can measure how fast someone runs, how hard they hits and how high they jumps. This makes building a game around the activity easier because you can keep score easily.

Physical activities do present problems for game designers though. The games are quickly limited by the physical abilities of the players. For all of your hard work balancing the game to make it fair to all players, some people will excel at the game based simply on physical prowess. This is especially true of games that hew closely

to the original physical activity. This does not break the game, but it can present some players with advantages and discourage others from participating.

Game designers can compensate for the dominance of physical ability by creating a system of rules to balance out the game and add more strategy. By offering overlapping strategies and multiple paths to victory, the game should theoretically allow for greater range of player abilities. The trade-off is complexity. Casual game designers must again be aware of this trade-off as they construct their game. The simpler the game, the faster players will understand and get into the game. However, simple games will likely be dominated by players with particular abilities that match the game's core mechanic.

FIGURE **3.3**



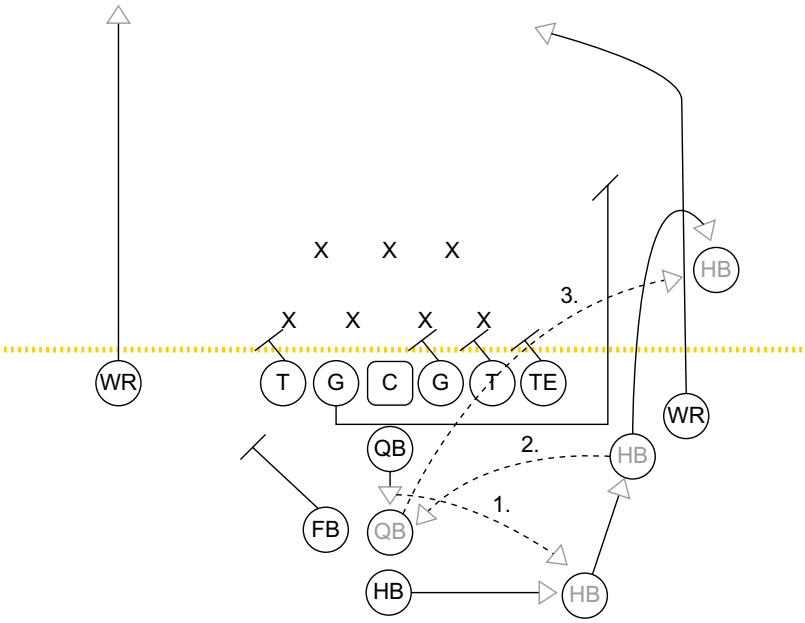
Run straight and run fast. Jesse Owens running the 200-meter dash at the 1936 Olympics. (WikiCommons[5])

Compare the 100-meter dash with American football. Anyone can immediately understand the 100-meter dash. There's really only one rule: be the first to cross the finish line. This simplicity and lack of complex strategies mean that 9 times out of 10, the faster runner will win the sprint. Contrast this with American football with

[5] http://commons.wikimedia.org/wiki/File:Jesse_Owens.jpg

its almost byzantine rule system governing ball control, downs, yards and ways to score. It's no wonder uninitiated viewers find themselves bewildered upon watching their first football game. But the complex rules of football allow for a wider range of strategies and outcomes. Yes, more physically adept players will generally be better at football, but players need not all be physically adept in the same way. There is no one trait that will always win out in football. A faster player may be outwitted by a clever player with agile feet. A stronger player may be avoided by a faster player. Football's complex rule system allows for different strategies and variable outcomes. However, you cannot explain football as quickly to a new player as you could the 100-meter dash. It is an extremely complex game, and the time necessary to master the rule system precludes it from meeting our definition of a casual game.

FIGURE
3.4



A diagram of the flea flicker play in football shows off the geometrical and strategic complexity of the game. (WikiCommons[6])

In Chapter 8, Hitting, we'll look at how tapping into primal physical actions can provide the basis for entire games. Games like *Whac-a-mole* and *Wii Tennis* rely heavily on physical sensation, giving them a place of primacy in the game over strategy or complexity. Then in Chapter 11, Bouncing, Tossing, Rolling and Stacking, we'll look at how physics systems can be modeled to emulate physical play and all the variability that accompanies objects careening through space.

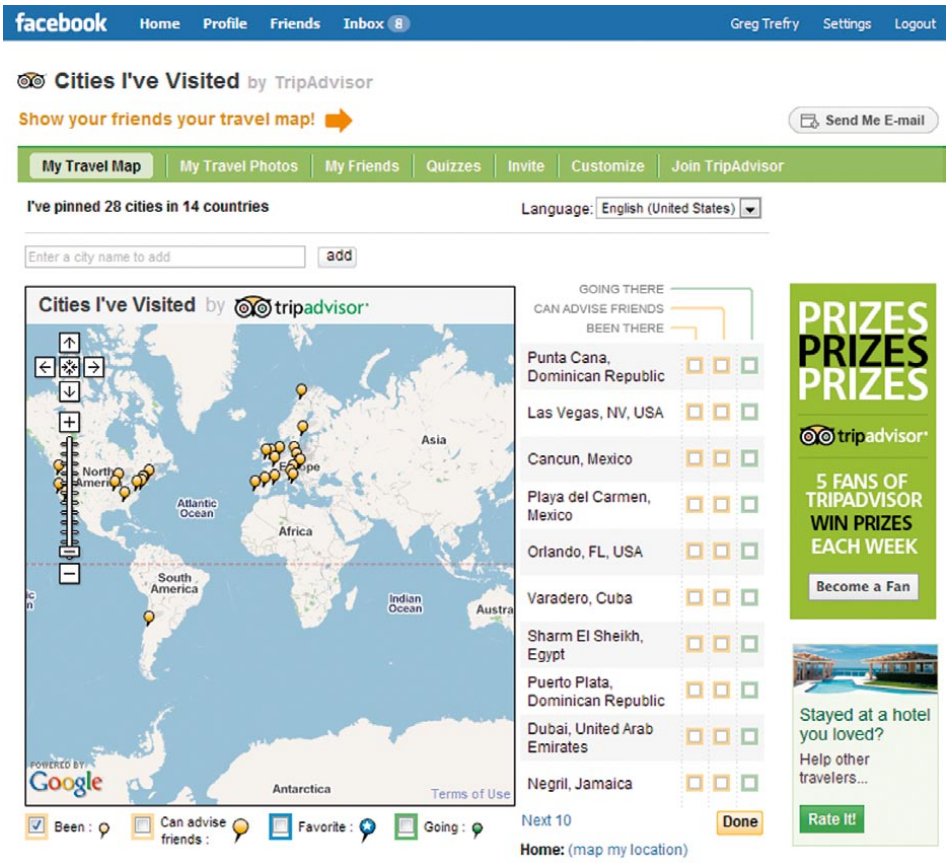
[6] http://commons.wikimedia.org/wiki/File:FleaFlicker.png, User: Jeffness

Playing with Others

Far more amorphous than physical play, but equally compelling, social play (talking, lying, accusing, betting, bluffing, bidding, joking, gossiping, comparing, copying, repeating, guessing, swarming) offers many activities which provide the roots for games. Lying, guessing and accusing form the basis for social games like *Mafia* and *Werewolf*. Card games like poker are as much about reading the tells of other players and calling their bluffs as the probability of building the right sets.

As Sutton-Smith points out in *Ambiguity of Play*, social play can include simple behaviors like comparing how many U.S. states you and your friend have each visited. This very impulse to compare/compete with your friends drives many of the applications on social networking sites like Facebook. Take the Facebook applications *Cities I've Visited* (Figure 3.5). The entire purpose of the application is to broadcast

FIGURE **3.5**



Applications like *Cities I've Visited* allow users to display how well-heeled they are while comparing their moves against their friends. This application is just shy of being an actual game. (Cities I've Visited ® is the registered trademark of TripAdvisor LLC. The screenshot is ©2009 TripAdvisor LLC. Used with permission)

how many places you've traveled to, while comparing your tally with your friends. This sort of social play manifests itself in many ways. People can find hours of fun in comparing their rental histories on services like Netflix. There is inherent competition in naming what movies you have seen and enjoyed.

Tapping into these behaviors offers game designers a rich palette of interaction. By making the social interactions between players a key part of the game, you can leverage the unpredictability of human behavior. There is inherent complexity in reading and reacting to others. Just figuring out if someone is telling the truth contains an implicit goal; figuring out the answer and calling them out can be incredibly satisfying. Adding rules and game structure can amplify the tension and excitement.

However, it also poses risks to the consistency of the game. The more the game relies on social interactions, the more dependent it becomes on the players and their ability and willingness to relate to one another. As opposed to physical play with hard quantifiable outcomes, social play can feel a bit squishy and subjective. Designers can address the subjectivity and the reluctance of players to fully engage socially by adding more structured rules. However, as the game becomes more formalized, the designer loses some of the spontaneity of true social interactions.

In Chapter 12, Socializing, we'll look at several games and how they use rules to craft social interactions. The card game *Apples to Apples* uses several simple rules to turn the arbitrary judgments of players into binding resolutions. Simply looking at the rules this dynamic might seem too loose and random. But in the context of the game it works quite effectively, given that players are willing to engage each other socially. *What to Wear*, a Facebook game about fashion, also uses the arbitrary judgments of other players to power a scoring system. In this game, the designers create a system to emulate the social milieu of fashion by harnessing the opinions of thousands of users.

Playing with Things

Little kids aren't the only ones with toys. From animals to kids to adults, we all occasionally find hours of enjoyment in simply playing with some object, spinning, bouncing, counting, sorting, matching and rolling it. Cats love paper bags, dashing in and out of them. Kids can be entertained by a good bag for hours too, even if the bag was originally used to deliver an expensive new toy. Heck, adults can find plenty of uses for a paper bag too, blowing them up and popping them behind an unsuspecting spouse. We use the object as a nexus or centerpiece for our play. Something in the physical characteristics of the object sparks our interest. We find that a ball bounces right back to our hand when thrown a certain way into the corner; or that a cell phone will spin for 10 revolutions on our desk. We explore and play with the cheapest of objects, reveling in our discoveries about how it behaves. Of course, as we get older, our taste for playful objects sometimes gets more expensive, running from fancy blenders to sports cars. There's joy to be had everywhere, even in flicking open menus with your thumb on an iPhone just to watch the little bounce it gives.

Object play is often mixed with physical play. What do you do with a ball? Kick it. Then there is object-based play that combines social play, like the always dangerous

party game *Spin-the-Bottle*. Other times, playing with objects requires more mental acumen than physical, as you struggle to make matches from memory or sort out sets.

As with other sorts of play, games sit just a few simple rules away. Turning this sort of play into a game can be as simple as seeing how long you can get the object to spin or how many catches you can make in a row.

Sorting and matching also falls into the category of object play. Humans seem to have a natural tendency to sort. Spread an array of different colored paper clips on a conference room table before a meeting, and I bet you by the end of the meeting, a significant portion of those paper clips will be grouped by color or strung into chains. We seem to have an almost subconscious need to produce order from chaos. Thousands of games have grown out of just this desire to sort and create sets. From the set creation in Poker to the split second group sorting of *Pit* to the color matching of *Snood*, countless games find real pleasure in making order of randomness.

Tapping into these activities offers fertile grounds for game designers. Games like *Memory* are bare bones games that rely primarily on the characteristics of the object. By building a game around play with an object, designers can take advantage of familiar interactions that already show an element of fun. The dart game cricket does exactly this. It provides a structure of rules and scoring to go with the joy of hurling sharp objects at a target. When you hold a dart in your hand, it's pretty obvious what you should do with it: throw the sharp end at anything you think it will stick into, preferably a dart board and not your brother. Simply throwing darts

FIGURE **3.6**

In Cricket, players must close out sections of the dart board by hitting them. The wide distribution of target areas ensures some amount of success, especially early in the game. (WikiCommons[7])

[7] http://commons.wikimedia.org/wiki/File:Darts_in_a_dartboard.jpg

and trying to hit the bull's eye is fun, but ultimately tiring over time. Unless you're quite practiced, you're not likely to hit the bull's eye very often, so your success to failure ratio will be pretty low. Cricket, on the other hand, demands that you hit regions all across the dart board. Players must close out regions of the dart board, like the area of the board labeled 19. You close the section by landing three darts in the wedge. The first team to close all of the sections in play wins. (You can also play Cricket where you begin to score points after closing a wedge.) Cricket gives players a range of new goals at which to take aim. Direct competition to close out scoring areas spurs players forward imbuing each throw with great meaning.

A game like *Guitar Hero* is an excellent example of a video game centered on an object. The designers at Harmonix, the original makers of *Guitar Hero*, had long dreamed of making a rhythm action game with a guitar controller. However, getting funding to make an expensive peripheral for a video game was no easy trick. So they cut their teeth developing rhythm games like *FreQuency* and *Amplitude*. These games were popular with players and game critics. But it wasn't until they were finally able to combine their stellar gameplay with a guitar-shaped controller designed in collaboration with Red Octane that they started winning over a whole new audience of players. The gameplay of their games *Amplitude* and *FreQuency* in many ways isn't that different from the main mechanic of *Guitar Hero*, but the little plastic guitar distributed with *Guitar Hero* makes all the difference. It immediately draws the player into the game because it's simply fun to play with. It's also familiar.

FIGURE
3.7

Much of the fun of *Guitar Hero* stems from rocking out on these simple almost silly looking plastic guitars. The object ties the game together. As soon as you pick it up you know how to hold it and you feel like a guitarist. (WikiCommons[8])

[8] http://commons.wikimedia.org/wiki/File:Guitar_Hero_series_controllers.jpg

Mentally mapping the activity of hitting streaming notes on the screen is much easier with an object that looks like a guitar than with a normal video game controller. The shape and feel of the object tells the players much of what they need to do.

The disadvantage of object play is, well, it requires you own the object and a specific set-up that accommodates play. This can limit the accessibility of the game. The more limited the accessibility, the more limited its chances for adoption.

In Chapter 4, Matching, and Chapter 5, Sorting, we'll look at how games like *Solitaire* and *Bejeweled* use the characteristics of objects to structure gameplay. The designers of these games create mechanics which highlight the features of the objects and force the player to interact with those features. Chapter 11, Bouncing, will look at games like *Bowman* that take the playful act of shooting an arrow and shape it into a simple goal-directed game that utilizes the physical properties of an arrow in flight.

Playacting

Any time we play, we playact to a greater or lesser degree. The roles we assume help guide our play and provide a thrust for our actions and unwritten rules for our behavior. Little kids playing cops and robbers take on the roles of heroes and villains in their play. They do not necessarily provide elaborate backstories and deep-seated motivations for these characters ("I'm a divorced cop with a serious authority problem and two mortgages on my house"). Instead, the roles simply provide a general behavioral framework that they use as departure points. Everyone knows the cops really want to catch the bad guys and will chase them down at all costs. So the kids playing the cops chase the robbers and yell, "Halt! Police!" Robbers, of course, rob. They knock over banks, grab the cash and run away. So kids playing robbers sneak around and then run like mad from the kids playing the cops when they're spotted. The simple sketch provides enough meaty roles for most childhood play. The action of play provides the forum to act out the roles.

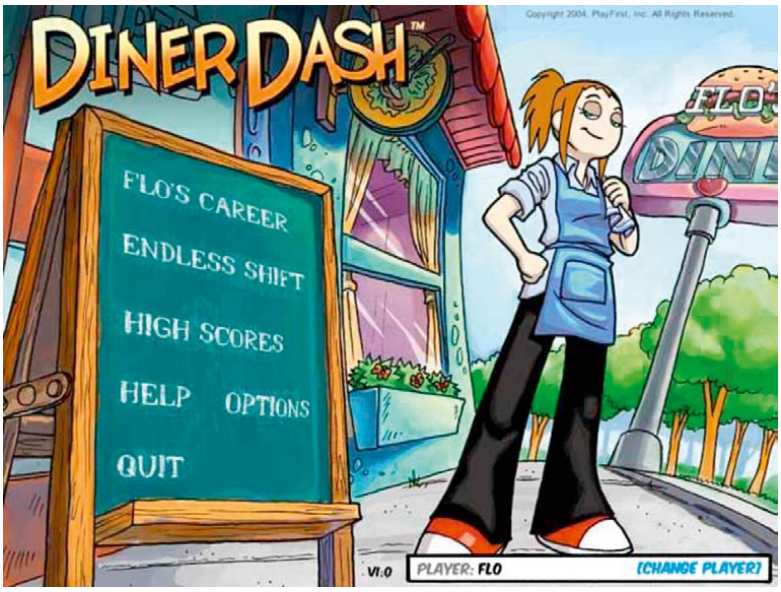
Role-playing seems to have garnered a bad reputation. People often associate it with Renaissance fairs and live-action role-playing. But these are instances of role-playing taken to extremes. There are many more subtle ways in which we enact characters.

From improv games to role-playing games to even video games we continue to adopt roles and characters to inform our actions. When we play a video game, we take cues from the character in the game. Depending on the type of game and the richness of the character, this will lead the player to inhabit the character and make decisions as that character. A game like *Diner Dash*, in which the player controls Flo, a former businesswoman who has chucked it all to run her own diner, requires little role-playing. You don't need to think about what it would be like to be Flo, owning your own restaurant and trying to pay your bills and all of the pressure that would entail to play the game. Frankly, the story of the game isn't very concerned with those issues either. Told in short and simple comic strip interstitials, the game story focuses on how Flo gets better and better at serving more and more people. Simply knowing that Flo is a waitress and that she needs to bring patrons their food is enough. The role of Flo gives the player an indication of how to behave.

FIGURE
3.8

Live-action role-playing takes role-playing to the extreme. There are many more subtle ways we enact roles. (WikiCommons[9])

FIGURE
3.9



By adopting the role of Flo the waitress, players instantly know how they should behave in levels: they should serve customers. (Copyright © 2003 PlayFirst, Inc. Reproduced by permission of PlayFirst, Inc.)

[9]http://commons.wikimedia.org/wiki/File:LARP_Sternenfeuer_Treffen.JPG

Similarly in a game like *Tom Clancy's Splinter Cell*, the player does some very light playacting. In *Splinter Cell*, the player controls Sam Fisher, a covert operative dropped into places like foreign embassies where you cannot risk detection. The need to sneak around undetected informs all of the player's actions within the game. The game allows for a certain amount of leeway. You can occasionally be seen and still pass the level, but players who take the game seriously will strive to never be seen, even if this makes the game harder to play. In this simple way, they enact the role of a covert operative even when the game does not strictly require it. They may not consider and inhabit Sam Fisher's personal motivations, but they do recognize the primary role of Fisher as a covert operative and enact that.

FIGURE **3.10**

In *Splinter Cell*, players adopt the role of the stealthy secret agent. Even in situations where you could jump out, players still make an effort to hide and go undetected. (WikiCommons[10] © Ubisoft)

As our playacting becomes more complex, so does our relationship to it. In some games, playacting not only provides motivations for actions within play, it also becomes a way to explore what it would be like to inhabit another persona. This is especially true of role-playing games, from table-top games like *Dungeons & Dragons* to video game RPGs like BioWare's *Star Wars: Knights of the Old Republic*. In *Dungeons & Dragons*, players craft characters with complicated back stories and a web of motivations. Then they play as those characters and attempt to make decisions consistent with the character's backstory and personality. The play of the game is a process of improvisational storytelling conducted between the Dungeon Master

[10] http://commons.wikimedia.org/wiki/File:Splinter_Cell_Screenshot_vision01.jpg

leading the game and the players enacting the characters in the game. In video game RPGs, players are more constrained by the software and the story embedded in the game. They can improvise and impact the story less. But they can still make significant choices as the character which greatly impact how other characters in the game react to them.

Almost any game with content that involves people, gods or even animals has an element of playacting. These games provide a role which is recognizable to players. The more abstract the game, the less playacting involved. So games like *Bejeweled* and *Tetris* involve next to no playacting. Players simply interact with the game system as the player. *Diner Dash*, with its recognizable, but fairly iconic character, allows for light playacting. The player does not become Flo, but does enact the basic physical actions of Flo. Games like *Dungeons & Dragons*, where the player creates a character and then inhabits the mindset of that character, involves a great deal of role-playing. You cannot really play without inhabiting the character.

Role-playing extends far outside of games and even play. We also apply role-playing and playacting to more serious pursuits, from acting to adopting slightly altered personas to fit different situations. When we answer the phone at work, many of us use an altogether different tone and voice than we do at home. We assume the role of office worker and project a certain persona to meet the situation. While this may not fall under our typical definition of play and fun, it shares similar qualities with playacting.

Basing games around acting and roles can provide powerful motivations for players. It also helps illuminate the context of the game for players. Allowing for playacting and building in opportunities for role-playing can greatly intensify the experience of a game. Games where the player can really get into character and assume a role tend to be more immersive. However, creating the right situations for role-playing can be very difficult. Players are often willing to engage in light playacting. But many players prove quite reluctant to go beyond that to actual role-playing. They are willing to consider and inhabit the role of an iconic spy much more than they are to try and assume the role of a specific spy with specific personal problems and bugaboos. Perhaps this is because of social taboos and social anxiety at the idea of performing. As a result, most players new to role-playing will try to be funny, hamming up the role rather than playing towards a more serious side. Laughter tends to cut the tension and relieve any social awkwardness. Even experienced role-players will tend to play for laughs when they play with strangers.

This reluctance to role-play makes designing games with deep elements of role-playing difficult, especially for casual players. Serious role-players tend to be serious gamers. However, casual gamers are often willing to engage in a bit of playacting. While many casual games like *Solitaire* and *Bejeweled* have quite abstract themes which allow for very little playacting, a number of successful games are beginning to involve more recognizable character and so more playacting. *Diner Dash* not only popularized the time-management mechanic among casual downloadables, it also put a recognizable avatar character that players could identify and playact at the center of the game. The publisher, Playfirst, went on to build an entire franchise

around Flo and even repeated the formula in other games like *Wedding Dash* with the character of the wedding planner Quinn.

How to Host a Murder has been a popular party game for more than 20 years. While it might seem like *How to Host a Murder* is a role-playing game closer to *D&D*, it is actually quite casual. It requires more playacting than it does actual role-playing. Players are assigned iconic roles which they can decide to act out however they choose. The only thing the game requires is that players reveal certain pieces of information at specific points to aid partygoers in solving the murder. You could play the game simply as a deductive logic exercise not unlike *Mastermind* or *Clue*. Of course, you would miss out on some silly, performative fun, but the game would work much the same.

Daredevilry/Push-Your-Luck

The last category of play involves the joy and fascination that comes from pushing limits. These activities are playful exertions that test your abilities and appetite for risk (sneaking, rock-climbing, cliff-diving, sticking your hand near a moving fan-blade, playing "I'm not touching you" with your big brother, ski jumping). In many ways they resemble the physical activities at the center of games. However, unlike those activities—which revolve around the joy of performing a simple action—these activities offer the thrill of seeing just what you can get away with before crashing and burning.

This sort of play naturally provides a visceral thrill. Rules are also naturally embedded in the activity. These rules are partially defined by physical elements, with the moment of failure or consequence serving as an upper bound. The player's goal is simply to get as close to that upper bound as possible without failing.

Jenga is the perfect example of this sort of activity formalized as a game. In *Jenga*, players stack a set of wood sticks, creating a tower. Then players take turns pulling out sticks until finally someone removes a piece that causes the entire mass of blocks to come crashing down. The game has a natural drama which increases with each piece pulled out, leaving all the players in a state of agitated anticipation until finally the crucial block is pulled and the tower collapses. At this point, inevitably a chorus of cheers and screams breaks out as the anxiety is released like a coiled spring.

It's not always easy to simulate daredevilry in video games where the risks are, by definition, lower. Sure you may lose, but you can always start over again with little penalty. But there are ways to do it. And successfully simulating this moment of anticipation and release can imbue the video game with great drama. *Tom Clancy's Splinter Cell* does an excellent job of drawing tension out of its stealth mechanic. Players are required to sneak around embassies avoiding guards, cameras and all manner of surveillance equipment. The player must quietly walk and climb by all of these guards, often passing only inches from them without raising an alarm. The player knows that as soon as they are detected, alarms will sound, more guards will rush in and quite likely the game will be over. This gives the game an interesting dichotomy between coiled silence as the player meticulously avoids detection and

FIGURE
3.11

All of *Jenga* builds up to the moment of collapse as players are forced to make increasingly dangerous moves.

frantic scrambling if they do set off the alarm. The player pushes as close as possible to the envelope, increasing the excitement.

Puzzle games like *Tetris* also exhibit similar qualities. Completely filling four lines all at once to form a "tetris" is the ultimate way to score in the puzzle game. To build up a layer of four lines with only a few key blocks missing is dangerous. It is entirely possible that you will never get the one long piece you need to complete the lines. And while you wait, other blocks will stack up and pile closer and closer to the top of the screen. Obsessively trying to score a tetris is a good way to lose the game, but doing so is attractive nonetheless. The reward doesn't necessarily offset the risk in terms of points, but the joy of flaunting the danger and succeeding brings its own psychic rewards.

Building this sort of play into video games often results in push-your-luck mechanics. Players are offered a choice between great penalty and reward if they can complete a complex set of moves that requires great precision. From shooting the moon in *Hearts* to rock climbing, these types of play offset risk, reward and skill. Game designers building this tension into a game benefit from the natural drama and excitement of success. However, they run the risk that this one move will be viewed as the only way to play the game. Shooting the moon in *Hearts* must be balanced out with a more normative way to play the game so that every player isn't constantly trying to shoot the moon on each hand, or simply giving up if the conditions aren't right.

Designers must also weigh whether their players are keen on risk. Casual players often have a strong desire to succeed rather than get hung up on a particular element or level over and over. So if the element of pushing your luck requires too exacting a skill, it will likely frustrate players. This frustration must be offset by easier, but equally valid ways to play and win.

All of these types of play provide game designers with a list of activities that we know engages players. The designer's job is figuring out how to take the basics of play and formalize them into games. This involves finding ways to structure and lengthen those play experiences. A key element to this process is adding goals and rules. As we define games, we'll see that rules will help us give shape to play experiences.

Defining Games

Play and games exist on a spectrum of formality. But that spectrum can sometimes look very broad and all-encompassing. Activities like poker or basketball are clearly games. But what about more ambiguous activities that follow strict rules, like trading stocks? Stock trading would seem to share many qualities with a game: it's rule-based, you have goals, there's a lot of randomness involved. Does that make it a game? Well, that depends on how we define game. And defining games is a favorite pastime of game designers.

The French philosopher Roger Caillois built upon the definition of play laid out by Johann Huizinga in *Homo Ludens*. Huizinga focused largely on play and how it interacted with culture. He saw play in all manner of activity, from rough-housing to politics to war. Caillois meanwhile focused the study of play and honed in on games. In his book *Man, Play and Games*, Caillois offered a definition of play and then provided a system of classification for games. Caillois recognized that games and play existed along a spectrum, with unstructured activities on one end and more formalized games rounding out the other.

Caillois provides the following definition of play:[11]

1. Free: in which playing is not obligatory; if it were, it would at once lose its attractive and joyous quality as diversion;
2. Separate: circumscribed within limits of space and time, defined and fixed in advance;
3. Uncertain: the course of which cannot be determined, nor the result attained beforehand, and some latitude for innovations being left to the player's initiative;
4. Unproductive: creating neither goods, nor wealth, nor new elements of any kind; and, except for the exchange of property among the players, ending in a situation identical to that prevailing at the beginning of the game;
5. Governed by rules: under conventions that suspend ordinary laws, and for the moment establish new legislation, which alone counts;
6. Make-believe: accompanied by a special awareness of a second reality or of a free unreality, as against real-life.

[11] Roger Caillois, *Man, Play and Games*, University of Illinois Press, 2001, pp. 9-10

This definition closely resembles Huizinga's definition of play. And Caillois recognized that play and games existed along a continuous spectrum, with looser play activities on one end and structured, rule-based games on the other. Caillois called the unstructured end of his spectrum "paidia." These activities are closer to the natural root of play where, as he put it, "an almost indivisible principle, common to diversion, turbulence, free improvisation, and carefree gaiety is dominant." The opposite of this freewheeling play he called, "ludus." On the ludus end of the spectrum, conventions and rules have been firmly established, meting out the loose, improvisational nature of paidia activities.

Perhaps Caillois had a penchant for casual game design. He seems to take a dim view of games as they move along the spectrum towards ludus. His view almost sounds negative when he writes, "this frolicsome and impulsive exuberance is almost entirely absorbed or disciplined by a complementary, and in some respects inverse, tendency to its anarchic and capricious nature: there is a growing tendency to bind it with arbitrary, imperative and purposely tedious conventions, to oppose it still more by ceaselessly practicing the most embarrassing chicanery upon it, in order to make it more uncertain of attaining its desired effect."[12] While he by no means dismisses formalized games in favor of play, he does point out the fact that as games become overly formalized and rigid, they lose some of their playful looseness that makes them attractive in the first place.

He recognizes that paidia-like activities will naturally progress towards the ludus end of the spectrum as the players formalizes their interaction with the activity. This mirrors what we know about games progressing towards complexity as you become more skilled.

Caillois went on to divide games into four different categories:

- Agon—games of competition in which players go head-to-head with one another
- Alea—games of chance in which a role of the dice and fate take primacy
- Mimicry—games of simulation dominated by playacting, spectacle and actual theater
- Ilinx—games of vertigo that draw key pleasure from the physical sensation and disorientation

These categories reflect many of the same traits we see in play.

A Designer's Definition

For the purposes of designing games, it can be helpful to lay out a definition for games. At the most basic level, it may help you determine if what you are designing is really a game or more of a toy or experience. That check may seem silly, but it's

[12] Roger Caillois, *Man, Play and Games*, University of Illinois Press, 2001, p. 13

actually more common than you might think. Game design can be applied in many ways, from crafting Web sites to designing toys to making simulations. In designing those experiences, you may intentionally break the definition of formal games and leave off some aspect. If you're making an interactive application, you may find you want the directed experience of games, but not the uncertain outcomes associated with winners and losers. Having a mental model of games serves as a useful sounding board for all interaction design.

Caillois lays out a very helpful definition of play and games. However, I think his definition could be divided into intrinsic and extrinsic qualities. The intrinsic qualities are inherent to the game and its system of rules. They don't speak at all to how people actually interact with it or view it. The extrinsic qualities help us contextualize games and define how it is viewed in a social context. These qualities speak to how we perceive play and games in society. They let us know that we are playing and not working or doing something else.

Intrinsic Qualities

Some of the qualities of games are intrinsic to the game. They are established and defined by the game itself. These are the qualities the game designer has the most control over, because they are the qualities laid out by the actual design of the game.

Governed by Rules

Of the six qualities that Caillois offers, rules are the most important. Rules are the primary characteristic of games. A game is a system of rules that govern behavior. The rules, written down on a piece of paper in a *Monopoly* board game set or explained out loud to a bunch of first-time basketball players, provide the structure for the game. They exist outside of the actual play of the game. Once the game is in play, players interact with the rules and adjust their behavior accordingly. At a base level to qualify as a game and not just play, the rules must encompass an outcome or goal. The rules must include an end state. Whether or not that is considered winning is up to the designer of the game.

Uncertain

Books and movies end the same way every time. No matter how many times you watch *Citizen Kane*, the last shot is always going to be of that darn sled. The fixed nature of a strip of linear film enables directors and writers to tell consistent stories. Games are more variable. Ideally, the outcome of a game is uncertain. You will win or you won't but you won't know which outcome will occur at the start of each game. However, even in games where the outcome is fairly certain—you playing one-on-one basketball against an all-star basketball player like LeBron James, for

instance—there is still a high degree of variability in the game. How many points will LeBron win by? Will he shoot jump shots? Will he dunk? Will he fall for your patented head fake that works so well against your little brother? Games provide differing levels of interaction and choices within them. It is this choice that lends games their possibility space. Simple games have more limited choices and possibility space. Good games offer players more interesting complex choices.

The more tightly a game restricts choices and follows a prescribed script, the more it begins to feel like a movie or a book. If there is no room for any choice for the player in the game, even a simplistic one question choice, then the game ceases to be a game. Games require some level of uncertainty informed by choice.

Extrinsic Qualities

The rest of the qualities that Caillois outlines describe how we perceive the game and are therefore extrinsic qualities to games. They help us understand the game in a cultural context, but they are not intrinsic to the system of rules that govern the actual game. We think of games as voluntary, fun, contained and offering up the possibility for a different winner every time. But none of this is intrinsic to the game. You could easily imagine situations where any one of these qualities could be negated and you would still not deny that you were playing a game.

Voluntary

Games and play are generally voluntary activities. We opt to play the game. This distinguishes it from work where we are required to go. Caillois points to professional athletes and says they are not playing a game, they are going to work. They have to be there to fulfill their job. While this is true—you could consider a professional soccer player an employee or a worker—there is no denying that when on the pitch kicking a ball around, soccer workers are engaged in the game of soccer. Their behavior is being governed by a set of rules. They are playing a game.

Boundaries

Games take place on fields and across boards. They last for several minutes or for several hours. But in that space and during that time, you are playing the game. When the game is over, you step back into the real-world. Huizinga referred to this idea as the "magic circle." Because games are separate, we are safe playing them and we know how to behave while playing them. There is real legitimacy to this claim, though a number of games have attempted to blur the line, from alternate reality games that masquerade as real mysteries to real-world games like *Assassins*.

In *Assassins,* players are given another player as a target. They must find this player and tag or squirt them with a water gun. When they do that player is

eliminated and he must pass on his target. The game may run over several weeks and integrates with your everyday life. You can be assassinated anywhere and any time, from your doorstep to your office desk. The lines of the game, of who is playing and who isn't, begin to blur, instilling a sense of paranoia in players. You could argue that this game blurs the boundaries between the game and the rest of the world. Players know, of course, that they are playing the game, but at times, their awareness may slip and game moves may intrude on their everyday life. Boundaries are important to shape a game, but they are not absolutely intrinsic to the game.

Caillois also says that games must be inherently unproductive. Indeed, most games are unproductive. Most exist simply to bring you some amount of joy while playing, but this does not necessarily preclude games from being productive. For one thing, good games produce joy and fun, which is a value in and of itself. And you could imagine a game that produces some other good as a side-effect. In our culture, games are certainly not designed to be productive, but that doesn't mean they couldn't be. It also ignores the production of fun, which is what most game designers aim for.

Bernard Suits described games quite elegantly when he wrote that games are the "voluntary attempt to overcome unnecessary obstacles."[13] Put more plainly, games take simple activities and make them hard. If we look at the activity behind almost every game, we can see that there is a simpler straight line to success than the game allows. When we talk about game design, these straight lines are considered exploits, and we create rules to prevent players from taking that path. This may seem silly. But it is in the challenge and effort to overcome these self-imposed obstacles that we find fun. Creating the right obstacles and striking the correct balance between your ability to reach your goal and the effort required to get there are the crux of game design.

There are, of course, many ways to look at games and many ways to define them. Designers keep in the back of their head a notion of what they believe they are up to when they make games. All of their work is checked against this mental model. Sometimes they choose to push against that model, imagining games that stretch their personal definition of games by circumventing one of these qualities. Indeed this can be a very productive way to imagine wholly new types of games. Having a mental model of what you are making is an invaluable aid in designing. Your definition can be rigid or loose, long or short, but you need to have an idea of what you are doing.

For novice game designers, it is best to start with a more rigid definition of games. Just like rules help direct a player's efforts toward a game's goal, a definition of games helps direct you towards creating a coherent and whole game. Sticking to the formal elements of games will help you get an idea of how to fulfill the promises of games. Once you understand the qualities of games you can begin to experiment with the form. Picasso may have made some wonderfully abstract paintings that challenged a viewers very notion of art, but he could also create heartbreakingly realist paintings as well. Breaking the rules was intentional.

[13] Bernard Suits, *The Grasshopper: Games, Life and Utopia*, Broadview Press, 2005, p. 55

Summary

Games grow out of play. As a game designer you must look at all manner of playful activities and recognize their potential as games. By girding play with rules and goals we can begin to transform play into games. And by transforming playful activities into games we can lengthen and deepen the experience. In designing casual games it's important to pay attention to this line between play and games. We don't want to overly discipline the frolicsome exuberance of play, as Caillois would say. Instead we want to highlight and enhance the natural playfulness of the activity while shaping it.

Now that we have a general framework for considering play we can look closely at some specific games and see how the designers shaped the experience through the careful application and combination of game-specific mechanics.