CHAPTER TWELVE

Some Elements Are Game *Mechanics*

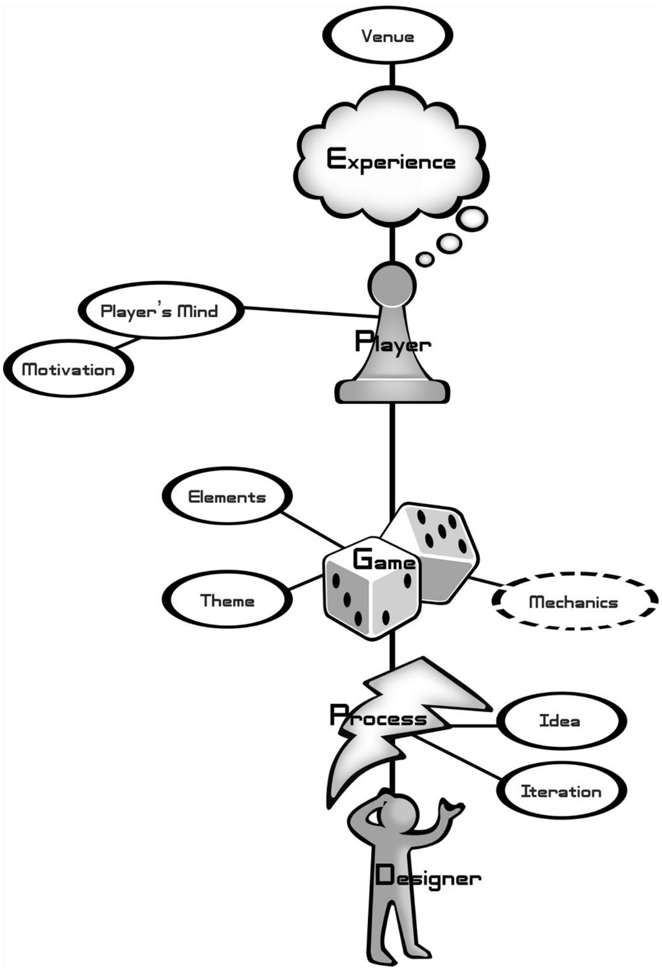


FIGURE **12.1**

We have talked a lot about designers, players, and the experience of gameplaying. It is time to talk nuts and bolts about what games are really made of. Game designers must learn to use their x-ray vision to be able to see past the skin of a game and quickly discern the skeleton, which is defined by the game mechanics.

But what are these mysterious mechanics?

Game mechanics are the core of what a game truly is. They are the interactions and relationships that remain when all of the aesthetics, technology, and story are stripped away.

As with many things in game design, we do not have a universally agreed-upon taxonomy of game mechanics. One reason for this is that the mechanics of gameplay, even for simple games, tend to be quite complex and very difficult to disentangle. Attempts at simplifying these complex mechanics to the point of perfect mathematical understanding result in systems of description that are obviously incomplete. Economic "game theory" is an example of this. You would think with a name like "game theory," it would be of great use to game designers, but in truth, it can only handle such simple systems that it is seldom useful for designing real games.

But there is another reason that taxonomies of game mechanics are incomplete. On one level, game mechanics are very objective, clearly stated sets of rules. On another level, though, they involve something more mysterious. Earlier, we discussed how the mind breaks down all games into mental models that it can easily manipulate. Part of game mechanics necessarily involves describing the structure of these mental models. Since these exist largely in the darkness of the subconscious mind, it is hard for us come up with a well-defined analytical taxonomy of how they work.

But that doesn't mean we shouldn't try. Some authors have approached this problem from a very academic perspective, more concerned with an analysis that is philosophically watertight than with one that might be useful to designers. We can't afford this kind of pedantry. Knowledge for the sake of knowledge is a fine thing, but our interest is in knowledge for the sake of great games, even if it means a taxonomy that has some gray areas. With that said, I present the taxonomy that I use to classify game mechanics. These mechanics fall largely into seven main categories, and each one can provide useful insights on your game design.

Mechanic 1: Space

Every game takes place in some kind of **space**. This space is the "magic circle" of gameplay. It defines the various places that can exist in a game and how those places are related to one another. As a game mechanic, space is a mathematical construct. We need to strip away all visuals, all aesthetics, and simply look at the abstract construction of a game's space.

There are no hard and fast rules for describing these abstract, stripped-down game spaces. Generally, though, game spaces

1. Are either discrete or continuous
2. Have some number of dimensions
3. Have bounded areas that may or may not be connected

The game of tic-tac-toe, for example, features a board that is discrete and 2D. What do we mean by "discrete"? Well, even though we commonly draw a tic-tac-toe board like this:

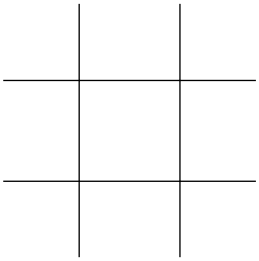


FIGURE 12.2

It is not really a continuous space, because we only care about boundaries, not the space within each cell. Whether you put your X...

FIGURE 12.3

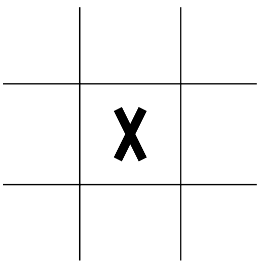


FIGURE 12.4

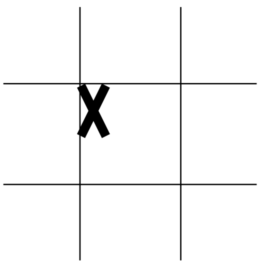
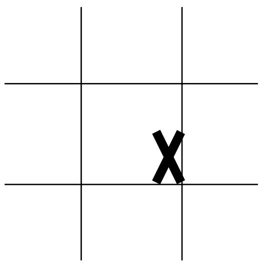
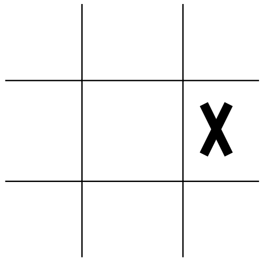


FIGURE 12.5



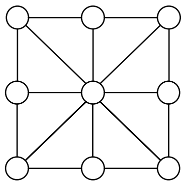
It doesn't really matter—all those are equivalent in terms of the game. But if you put your X here:

FIGURE
12.6



That is another matter entirely. So, even though the players can make their marks in an infinite number of places in a continuous 2D space, there are really only nine discrete places that have any actual meaning in the game. In a sense, we really have nine zero-dimensional cells, connected to each other in a 2D grid, like this:

FIGURE
12.7



Each circle represents a 0D place, and each line shows which places are connected to each other. In tic-tac-toe, there is no movement from place to place, but adjacency is very important. Without adjacency, it would just be nine disconnected points. With the adjacency, it becomes a discrete 2D space, with clear boundaries—the space is three cells wide and three cells high. The space for a chessboard is similar, except that it is an 8 × 8 space.

A game with fancy aesthetics can fool you into thinking that its functional space is more complex than it really is. Consider a Monopoly board.

At first glance, you might say it is a discrete 2D space, like a chessboard, with most of the middle cells missing. But it can be more simply represented as a 1D space—a single line of forty discrete points, which connects to itself in a loop. Sure, on the game board, the corner spaces look special because they are bigger, but functionally that doesn't matter, since each game square is a zero-dimensional space. Multiple game pieces can be in a single game square, but their relative positions within that square are meaningless.

But not all game spaces are discrete. A pool table is an example of a continuous 2D space. It has a fixed length and width, and the balls can freely move about on the table, ricocheting off of the walls or falling into the holes, which are in fixed positions. Everyone would agree that the space is continuous, but is it 2D? Since clever players can sometimes cause the balls to leave the table and hop over each other, you could certainly argue that this is really a 3D game space, and for some purposes, it is useful to think of it that way. There are no hard and fast rules for these abstract functional spaces. When designing a new game, there are times it will be useful for you to think of your space as 2D and there are times when thinking of it as 3D is more useful. The same goes for continuous vs. discrete. The purpose of stripping down a game into a functional space is so that you can more easily think about it, without the distractions of aesthetics or the real world. If you are thinking about modifying the game of soccer to a playing field with new boundaries, you will probably think about it in terms of a 2D continuous space.

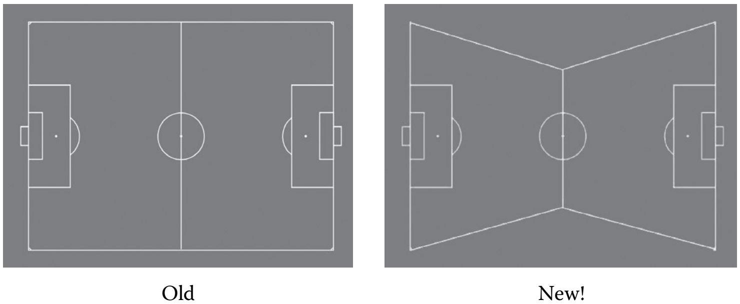


FIGURE
12.8

But if you are thinking about modifying the height of the goal, or changing the rules about how high the players can kick the ball, or adding hills and valleys to the field, it is useful to think of it as a continuous 3D space instead.

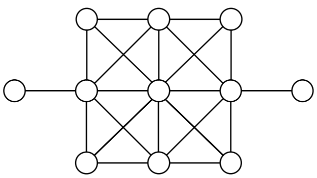
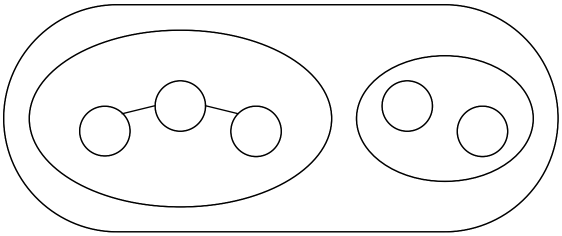


FIGURE
12.9

There might even be times you think about a soccer field as a discrete space—breaking it up into, say, nine major areas of play, with two extra areas on the left and right representing the goals. This mode of thinking might prove useful if you are analyzing the different kinds of play that take place in different parts of the field, for example. The important thing is that you come up with abstract models of your game space that help you better understand the interrelationships of your game.

Nested Spaces

FIGURE
12.10



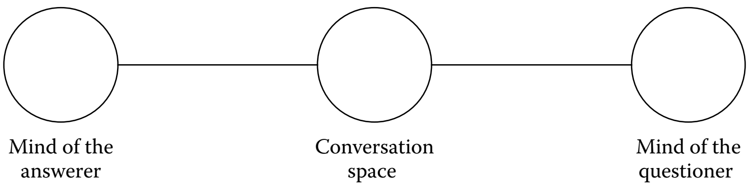
Many game spaces are more complex than the examples we have looked at here. Often, they feature "spaces within spaces." Computer-based fantasy role-playing games are a good example of this. Most of them feature an "outdoor space" that is continuous and 2D. A player traveling this space sometimes encounters little icons representing towns, or caves, or castles. Players can enter these as completely separate spaces, not really connected in any way to the "outdoor space" but through the gateway icon. This is not geographically realistic, of course—but it matches our mental models of how we think about spaces—when we are indoors we think about the space inside the building we are in, with little thought to how it exactly relates to the space outside. For this reason, these "spaces within spaces" are often a great way to create a simple representation of a complex world.

Zero Dimensions

Does every game take place in a space? Consider a game like "Twenty Questions," where one player thinks of an object, and the other player asks "yes or no" questions trying to guess what it is. There is no game board and nothing moves—the game is just two people talking. You might argue that this game has no space. On the other hand, you might find it useful to think of the game happening in a space that looks like Figure 12.11.

The mind of the answerer contains the secret object. The mind of the questioner is where all the weighing of the previous answers is going on, and the conversation space between them is how they exchange information. Every game has some kind

FIGURE
12.11



of information or "state" (as we'll see later in Mechanic 2), and this has to exist somewhere. So, even if a game takes place in a single point of zero dimensions, it can be useful to think of it as a space. You may find that figuring out an abstract model for a game whose space seems to be trivial may lead you to insights about it that surprise you.

Being able to think about the space of your game in functional abstract terms is an essential perspective for a designer, and it is Lens #26.

Lens #26: The Lens of Functional Space

To use this lens, think about the space in which your game really takes place when all surface elements are stripped away.

Ask yourself these questions:

- Is the space of this game discrete or continuous?
- How many dimensions does it have?
- What are the boundaries of the space?
- Are there subspaces? How are they connected?
- Is there more than one useful way to abstractly model the space of this game?

When thinking about game spaces, it is easy to be swayed by aesthetics. There are many ways to represent your game space, and they are all good, as long as they work for you. When you can think of your space in these pure abstract terms, it helps you let go of assumptions about the real world, and it lets you focus on the kinds of gameplay interactions you would like to see. Of course, once you have manipulated the abstract space so that you are happy with its layout, you will want to apply aesthetics to it. The Lens of Functional Space works quite well with Lens #10, *Holographic Design*. If you can simultaneously see your abstract functional space and the aesthetic space the player will experience, as well as how they interrelate, you can make confident decisions about the shape of your game's world.

Mechanic 2: Time

In the real world, **time** is the most mysterious of dimensions. Against our will, we travel through it, ever forward, with no way to stop, turn around, slow down, or speed up. In the world of games, we often try to remedy this lack of control by creating toy worlds that let us play with time like gods.

Discrete and Continuous Time

Just as space in games can be discrete or continuous, so can time. We have a word for the unit of discrete time in a game: the "turn." Generally, in turn-based games, time matters little. Each turn counts as a discrete unit of time, and the time between turns, as far as the game is concerned, doesn't exist. Scrabble games, for example, are generally recorded as a series of moves, with no record of the amount of time that each move took, because real clock time is irrelevant to the game mechanics.

Of course, there are many games that are not turn based, but instead operate in continuous time. Most action videogames are this way, as are most sports. And some games use a mix of time systems. Tournament chess is turn-based but has a continuous clock to place time limits on each player.

Clocks and Races

Clocks of varying types are used in many games, to set absolute time limits for all kinds of things. The "sand timer" used in Boggle, the game clock in American football, and even the duration of Mario's jump in Donkey Kong are different kinds of "clock" mechanisms, designed to limit gameplay through absolute measure of time. Just as there can be nested spaces, sometimes time is nested, as well. Basketball, for instance, is often played with a game clock to limit the length of total play but also with a much shorter "shot clock" to help ensure players take more risks, keeping the gameplay interesting.

Other measures of time are more relative—we usually refer to these as "races." In the case of a race, there is not a fixed time limit, but rather pressure to be faster than another player. Sometimes this is very obvious, like in an auto race, but other races are more subtle, such as my race in *Space Invaders* to destroy all the invading aliens before they manage to touch the ground.

There are many games, of course, where time is not a limiting factor, but it is still a meaningful factor. In baseball, for example, innings are not timed, but if the game goes on too long, it can exhaust the pitcher, making time an important part of the game. In Chapter 13, "Balance," we will talk about different game factors that can control how long a game takes to play.

Controlling Time

Games give us the chance to do something we can never do in the real world: control time. This happens in a number of fascinating ways. Sometimes we stop time completely, as when a "time-out" is called in sporting match or when the "pause" button is pushed on a videogame. Occasionally, we speed up time, as happens in games like Civilization, so that we can see years pass in just seconds. But most often, we rewind time, which is what happens every time you die in a videogame and return to a previous checkpoint. Some games, such as *Braid*, go so far as to make manipulation of game time a central mechanic.

Since time is invisible and unstoppable, it is easy to forget. Take this lens to help you remember.

Lens #27: The Lens of Time

It is said that "timing is everything." Our goal as designers is to create experiences, and experiences are easily spoiled when they are too short or too long, too fast or too slow. Ask these questions to make yours just the right length:

- What is it that determines the length of my gameplay activities?
- Are my players frustrated because the game ends too early? How can I change that?
- Are my players bored because the game goes on too long? How can I change that?
- Would clocks or races make my gameplay more exciting?
- Time limits can irritate players. Would I better off without time limits?
- Would a hierarchy of time structures help my game? That is, several short rounds that together comprise a larger round?

Timing can be very difficult to get right, but it can make or break a game. Often, it makes sense to follow the old vaudevillian adage "Leave'em wanting more."

Mechanic 3: Objects, Attributes, and States

A space without anything in it is, well, just a space. Your game space will surely have **objects** in it. Characters, props, tokens, scoreboards, or anything that can be seen or manipulated in your game falls into this category. Objects are the "nouns" of game mechanics. Technically, there are times you might consider the space itself an object, but usually the space of your game is different enough from other objects that it stands apart. Objects generally have one or more **attributes**, one of which is often the current position in the game space.

Attributes are categories of information about an object. For example, in a racing game, a car might have maximum speed and current speed as attributes. Each attribute has a current **state**. The state of the "maximum speed" attribute might be 150 mph, while the state of the "current speed" attribute might be 75 mph if that is how fast the car is going. Maximum speed is not a state that will change much, unless perhaps you upgrade the engine in your car. Current speed, on the other hand, changes constantly as you play.

If objects are the nouns of game mechanics, attributes and their states are the adjectives.

Attributes can be static (such as the color of a checker), never changing throughout the game, or dynamic (the checker has a "movement mode" attribute with three possible states: "normal," "king," and "captured"). Primarily, we are interested in dynamic attributes.

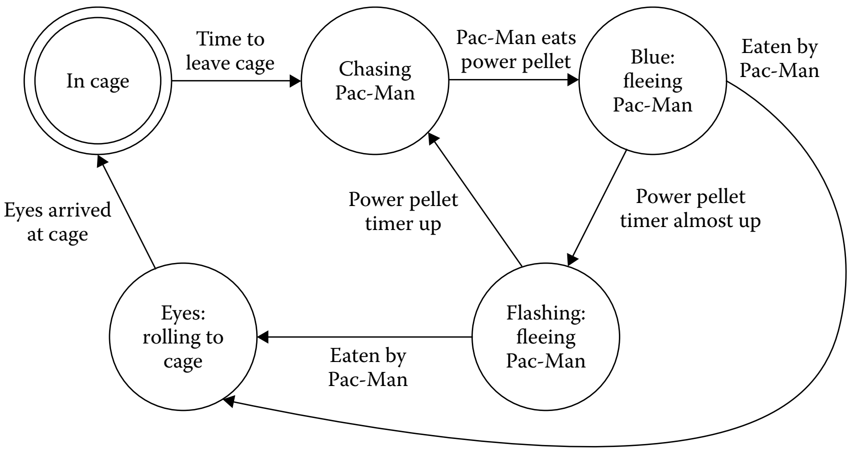
The following are two more examples:

1. In chess, the king has a "movement mode" attribute with three important states ("free to move," "in check," and "checkmated.")
2. In Monopoly, each property on the board can be considered an object with a dynamic "number of houses" attribute with six states (0, 1, 2, 3, 4, hotel) and a "mortgaged" attribute with two states (yes, no).

Is it important to communicate every state change to the player? Not necessarily. Some state changes are better hidden. But for others, it is crucial to be sure they are communicated to the player. A good rule of thumb is that if two objects behave the same way, they should look the same. If they behave differently, they should look different.

Videogame objects, especially ones that simulate intelligent characters, have so many attributes and states that it is easy for a designer to get confused. It is often useful to construct a state diagram for each attribute to make sure you understand which states are connected to which and what triggers state changes. In terms of game programming, implementing the state of an attribute as a "state machine" can be a very useful way to keep all this complexity tidy and easy to debug. Figure 12.12 is a sample state diagram for the "movement" attribute of the ghosts in *Pac Man*.

FIGURE
12.12



The circle that reads "In Cage" is the initial state for the ghosts (double circle is often used to indicate the start state). Each of the arrows indicates a possible state transition, with an event that triggers that transition. Diagrams like these are very useful when trying to design complex behaviors in a game. They force you to really think through everything that can happen to an object and what makes it happen. By implementing these state transitions in computer code, you automatically forbid illegal transitions (such as "In Cage" → "Blue"), which helps cut down on puzzling bugs. These diagrams can get quite complicated and are sometimes nested. For example, it is quite likely that the real Pac-Man algorithm actually has several substates in "Chasing Pac Man," such as "Scanning for Pac-Man," "On Pac-Man's Tail," and "Moving through a Tunnel."

Deciding which objects have what attributes and what states is up to you. There are often multiple ways to represent the same thing. In a game of poker, for example, you could define a player's hand as an area of the game space that has five card objects in it, or you could decide you don't want to think of cards as objects and just call the player's hand an object that has five different card attributes. As with everything in game design, the "right" way to think about something is whichever way is most useful at the moment.

Games that force the players to be aware of too many states (too many game pieces, too many statistics about each character) to play can confuse and overwhelm. In Chapter 13, "Balance," we'll discuss techniques for optimizing the amount of state the players have to deal with. Thinking of your game strictly as a set of objects and attributes with changing states can give a very useful perspective, and it serves as Lens #28.

Lens #28: The Lens of the State Machine

To use this lens, think about what information changes during your game. Ask yourself these questions:

- What are the objects in my game?
- What are the attributes of the objects?
- What are the possible states for each attribute?
- What triggers the state changes for each attribute?

Gameplaying is decision making. Decisions are made based on information. Deciding the different attributes, their states, and what changes them is core to the mechanics of your game.

Secrets

A very important decision about game attributes and their states is who is aware of which ones. In many board games, all information is public; that is, everyone knows it. In a game of chess, both players can see every piece on the board and every piece that has been captured—there are no secrets, except what the other player is thinking. In card games, hidden or private state is a big part of the game. You know what cards you hold, but which ones your opponents hold is a mystery for you to puzzle out. The game of poker, for example, is largely about trying to guess what cards your opponents have while attempting to conceal information about what cards you might have. Games become dramatically different when you change what information is public or private. In standard "draw poker," all states are private—players can only guess your hand based on how much you bet. In "stud poker," some of your cards are private and some are public. This gives opponents much more information about each other's situations, and the game feels very different. Board games such as Battleship and Stratego are all about guessing the states of your opponent's private attributes.

In videogames, we face something new: a state that only the game itself knows about. This raises a question about whether virtual opponents, from a game mechanics standpoint, should be thought of as players or just part of the game. This is well illustrated by a story: In 1980, my grandfather bought an Intellivision game console, which came with a "Las Vegas Poker and Blackjack" game cartridge. He had great fun with it, but my grandmother refused to play. "It cheats," she insisted. I told her that was silly—it was just a computer—how could it cheat? She explained her reasoning: "It knows what all my cards are, and all the cards in the deck! How can it *not* cheat?" And I had to admit that my explanation that the computer "doesn't look at those" when it is making decisions about playing the game sounded kind of weak. But it brings out the point that there were really three entities in that game who knew the states of different attributes: my grandfather, who was aware of the state of his hand; the virtual opponent algorithm, which was "aware" of the state of its hand; and lastly the main algorithm for the game, which was aware of both players' hands, every card in the deck, and everything else about the game.

So, it seems that from a public/private attribute point of view, it makes sense to consider virtual opponents as individual entities on par with players. The game itself, though, is yet another entity, with a special status, since it isn't really playing the game, although it may be making decisions that enable the game to happen. Celia Pearce points out another kind of information, which is private from all of the entities we have mentioned so far: randomly generated information, such as a die roll. Depending on your views about predestination, you might argue that this information doesn't even exist until it is generated and revealed, so that referring to it as private is a little silly. But it does fit well into a Venn diagram I call the "hierarchy of knowers," which helps to visualize the relationship between the public and private states:

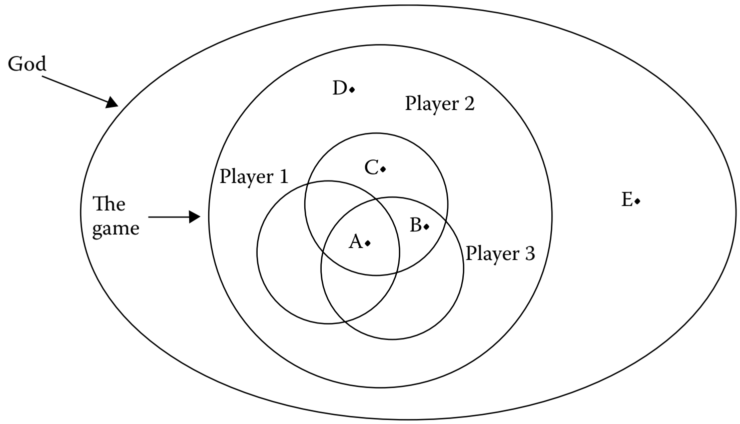


FIGURE
12.13

Each circle in Figure 12.13 represents a "knower." The "knowers" are god, the game, and players 1, 2, and 3. Each point represents some information in the game—the state of an attribute:

- **A** is information that is completely public, such as the position playing piece on a game board, or a face-up card. All the players are aware of it.
- **B** is the state that is shared between players 2 and 3 but kept secret from player 1. Perhaps 2 and 3 each had the opportunity to look at a face-down card, but player 1 didn't. Or maybe players 2 and 3 are virtual opponents of player 1, and their algorithm has them sharing information so they can team up against player 1.
- **C** is information private to a single player, in this case player 2. It could be cards he was dealt, for example.
- **D** is information that the game knows about, but not the players themselves. There are some mechanical board games where this kind of state exists in the physical structure of the board game, but is unknown to the players. *Stay Alive* was a classic example, with plastic sliders that when moved revealed holes in the board. *Touché* is another interesting example, where magnets of unknown polarity are placed under each square of the board. The states are "known" by the game, but not by the players. Another example is tabletop role-playing games, which feature a "dungeon master," or "game master," who is not one of the players and who privately knows a great deal of the game state, since he is the operational mechanism of the game, so to speak. Most computer games have a great deal of internal state that is not known to the players.
- **E** is randomly generated information, known only by the fates, god, etc.

Secrets are power. Lens #29 will help you channel that power to make your game as interesting as it can be. Take it, but don't tell anyone.

Lens #29: The Lens of Secrets

Change who has what information, and you change your game completely. To use this lens, think about who knows what and why.

Ask yourself these questions:

- What is known by the game only?
- What is known by all players?
- What is known by some or only one player?
- Would changing who knows what information improve my game in some way?

Gameplaying is decision making. Decisions are made based on information. Deciding the different attributes, their states, and who knows about them is core to the mechanics of your game. Small changes to who knows what information can radically change a game, sometimes for the better, sometimes for the worse. Who knows about what attributes can even change over the course of a game—a great way to create drama in your game is to make an important piece of private information suddenly become public.

Mechanic 4: Actions

The next important game mechanic is the **action**. Actions are the "verbs" of game mechanics. There are two perspectives on actions or, put another way, two ways to answer the question "What can the players do?"

The first kind of action is the **basic action**. These are simply the base actions a player can take. For example, in checkers, a player can perform only three basic operations:

1. Move a checker forward.
2. Jump an opponent's checker.
3. Move a checker backwards (kings only).

The second kind of action is **strategic action**. These are actions that are only meaningful in the larger picture of the game—they have to do with how the player is using basic actions to achieve a goal. The list of strategic actions is generally longer than the list of basic actions. Consider some possible strategic actions in checkers:

- Protect a checker from being captured by moving another checker behind it.
- Force an opponent into making an unwanted jump.
- Sacrifice a checker to trick his opponent.
- Build a “bridge” to protect his back row.
- Move a checker into the “king row” to make it a king.

Emergent Gameplay

The strategic actions often involve subtle interactions within the game and are often very strategic moves. These actions are mostly not part of the rules, per se, but rather actions and strategies that emerge naturally as the game is played. Most game designers agree that interesting emergent actions are the hallmark of a good game. Consequently, the ratio of meaningful strategic actions to basic actions is a good measure of how much emergent behavior your game features. It is an elegant game indeed that allows a player a small number of basic actions but a large number of strategic actions. It should be noted that this is a somewhat subjective measure, since the number of “meaningful” strategic actions is a matter of opinion.

Trying to create “emergent gameplay,” that is, interesting strategic actions, has been likened to tending a garden, since what emerges has a life of its own, but at the same time, it is fragile and easily destroyed. When you notice some interesting strategic actions showing up in your game, you must be able to recognize them and then do what you can to nurture them and give them a chance to flourish. But what makes these things spring up in the first place? It is not just luck—there are things you can do to increase the chances of interesting strategic actions appearing. Here are five tips for preparing the soil of your game and planting seeds of emergence.

1. **Add more verbs**: That is, add more basic actions. The strategic actions appear when basic actions interact with each other, with objects, and with the game space. When you add more basic actions, there are more opportunities for interaction and thus emergence. A game where you can run, jump, shoot, buy, sell, drive, and build is going to have a lot more potential for emergence than a game where you can just run and jump. Be careful, though—adding too many basic actions, especially ones that don’t interact with each other well, can lead to a game that is bloated, confusing, and inelegant. Keep in mind that the ratio of strategic actions to basic actions is more important than the sheer number of basic actions. It is usually better to add one good basic action than a slew of mediocre ones.

2. **Verbs that can act on many objects**: This is possibly the single most powerful thing you can do to make an elegant, interesting game. If you give a player a gun that can only shoot bad guys, you have a very simple game. But if that same gun can also be used to shoot a lock off a door, break a window, hunt for food, pop a car tire, or write messages on the wall, you now start to enter a world of many possibilities. You still only have one basic action: "shoot," but by increasing the number of things you can usefully shoot at, the number of meaningful strategic actions increases as well.
3. **Goals that can be achieved more than one way**: It's great to let players do all kinds of different things in your game, giving them lots of verbs, and verbs with lots of objects. But if the goals can only be achieved one way, players have no reason to look for unusual interactions and interesting strategies. To follow up with the "shoot" example, if you let players shoot all kinds of things, but the goal of your game is just "shoot the boss monster," the players will only do that. On the other hand, if you can shoot the monster, or shoot out a support chain so a chandelier could crash down on him, or maybe even not shoot him at all, but stop him through some nonviolent means, you will have a rich, dynamic gameplay, where lots of things are possible. The challenge with this approach is that the game becomes hard to balance, for if one of the options is always significantly easier than the others (a dominant strategy), players will always pursue that option. We will discuss that further in Chapter 13, "Balance."
4. **Many subjects**: If checkers involved just one red checker and one black one but had the same rules, the game would not be interesting at all. It is because the players have many different pieces they can move, pieces that can interact with one another, coordinating and sacrificing, that the game becomes interesting. This method obviously doesn't work for all games, but it can work in some surprising places. The number of strategic actions seems to have roughly a magnitude of subjects times verbs times objects, so adding more subjects is very likely to increase the number of strategic actions.
5. **Side effects that change constraints**: If every time you take an action, it has side effects that change the constraints on you or your opponent, very interesting gameplay is likely to result. Let us again look to checkers. Every time you move a piece, you not only change the squares that you threaten with capture, but you simultaneously change which squares your opponent (and you) can move into. In a sense, every move changes the very nature of the game space, whether or not you intended it to. Think how different checkers would be if multiple pieces could peacefully cohabitate on a single square. By forcing multiple aspects of the game to change with every basic action, you are very likely to cause interesting strategic actions to suddenly appear.

Lens #30: The Lens of Emergence

To make sure your game has interesting qualities of emergence, ask yourself these questions:

- How many verbs do my players have?
- How many objects can each verb act on?
- How many ways can players achieve their goals?
- How many subjects do the players control?
- How do side effects change constraints?

When comparing games with books and films, one of the most striking differences is the number of verbs. Games usually limit players to a very narrow range of potential actions, while in stories the number of possible actions that characters can engage in seems nearly limitless. This is a natural side effect of the fact that in games, the actions and all their effects must be simulated on the fly, while in stories it is all worked out ahead of time. In Chapter 18, "Indirect Control," we will discuss how this "action gap" can be bridged in the mind of the player, so that you can give the feeling of limitless possibilities while keeping the number of basic actions at a manageable limit.

The reason so many games seem similar to one another is because they use the same set of actions. Look at the games that are considered "derivative," and you will see that they have the same set of actions as older games. Look at games that people call "innovative," and you will find that they give the players new kinds of actions, either basic or strategic. When *Donkey Kong* first appeared, it seemed very different because it was about running and jumping, which was new at the time. *Harvest Moon* was a game about farming. *Katamari Damacy* was about rolling a sticky ball. The actions a player can take are so crucial to defining a game's mechanics that changing a single action can give you a completely different game.

Some designers dream of games where any verb the player can think of is a possible action, and this is a beautiful dream. Some massively multiplayer games are starting to move in that direction, offering a wide range of verbs for combat, crafting, and social interaction. In a way, this is a return to the past—in the 1970s and 1980s, text adventures were very popular typically featuring dozens or hundreds of possible verbs. Only with the rise of more visual games did the number of verbs suddenly decrease, because it was not feasible to support all those actions in a visual-based game. The demise (or hibernation?) of the text adventure genre is usually attributed to the public's hunger for fancy visuals—but perhaps, from an action perspective, there is another explanation. Modern 3D videogames give you a very limited range of basic actions. The player generally knows every action they can possibly attempt. In text adventures, the complete set of basic actions was unclear, and discovering them was part of the game. Very often, the solution to a tricky puzzle was thinking to type an unusual verb, like "spin the fish" or "tickle the monkey." While this was all very creative, it was also often frustrating—for every one of the hundreds of verbs a game supported, there were thousands it did not. As a result, players did not really have the "complete freedom" that text adventure interfaces pretended to give them. It is possible that this frustration, more than anything else, caused text adventures to fall from favor.

Your choice of actions significantly defines your game structure, so let's make that Lens #31.

Lens #31: The Lens of Action

To use this lens, think about what your players can do and what they can't and why.

Ask yourself these questions:

- What are the basic actions in my game?
- What are the strategic actions?
- What strategic actions would I like to see? How can I change my game in order to make those possible?
- Am I happy with the ratio of strategic to basic actions?
- What actions do players wish they could do in my game that they cannot? Can I somehow enable these, either as basic or strategic actions?

A game without actions is like a sentence without verbs—nothing happens. Deciding the actions in your game will be the most fundamental decision you can make as a game designer. Tiny changes to these actions will have tremendous ripple effects with the possibility of either creating marvelous emergent gameplay or making a game that is predictable and tedious. Choose your actions carefully, and learn to listen to your game and your players to learn what is made possible by your choices.

Mechanic 5: Rules

The **rules** are really the most fundamental mechanic. They define the space, the timing, the objects, the actions, the consequences of the actions, the constraints on the actions, and the goals. In other words, they make possible all the mechanics we have seen so far and add the crucial thing that makes a game a game—goals.

Parlett's Rule Analysis

David Parlett, game historian, did a very good job of analyzing the different kinds of rules that are involved with gameplay, as shown in this diagram.

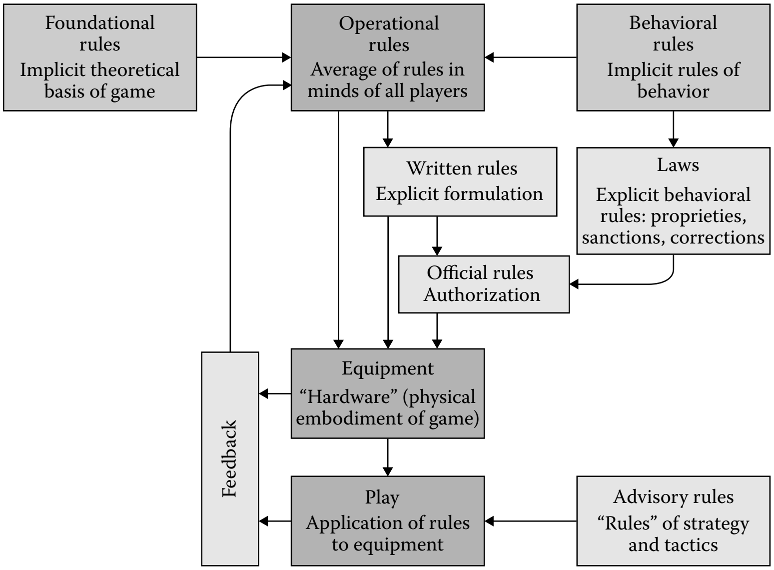


FIGURE 12.14

This shows the relationships between all the kinds of rules we are likely to encounter, so let's consider each.

1. **Operational rules**: These are the easiest to understand. These are basically "What the players do to play the game." When players understand the operational rules, they can play a game.
2. **Foundational rules**: The foundational rules are the underlying formal structure of the game. The operational rules might say "The player should roll a six-sided die, and collect that many power chips." The foundational rules would be more abstract: "The player's power value is increased by a random number from 1 to 6." Foundational rules are a mathematical representation of game state and how and when it changes. Boards, dice, chips, health meters, etc., are all just operational ways of keeping track of the foundational game state. As Parlett's diagram shows, foundational rules inform operational rules. There is not yet any standard notation for representing these rules, and there is some question about whether a complete notation is even possible. In real life, game designers learn to see the foundational rules on an as-needed basis, but seldom do they have any need to formally document the entire set of foundational rules in a completely abstract way.

3. **Behavioral rules**: These are rules that are implicit to gameplay, which most people naturally understand as part of "good sportsmanship." For example, during a game of chess, one should not tickle the other player while they are trying to think or take five hours to make a move. These are seldom stated explicitly—mostly, everyone knows them. The fact that they exist underlines the point that a game is a kind of social contract between players. These, too, inform the operational rules. Steven Sniderman has written an excellent essay about behavioral rules called "Unwritten Rules."
4. **Written rules**: These are the "rules that come with the game," the document that players have to read to gain an understanding of the operational rules. Of course, in reality, only a small number of people read this document—most people learn a game by having someone else explain how to play. Why? It is very hard to encode the nonlinear intricacies of how to play a game into a document and similarly hard to decode such a document. Modern videogames have gradually been doing away with written rules in favor of having the game itself teach players how to play through interactive tutorials. This hands-on approach is far more effective, though it can be challenging and time consuming to design and implement as it involves many iterations that cannot be completed until the game is in its final state. Every game designer must have a ready answer to the question: "How will players learn to play my game?" Because if someone can't figure out your game, they will not play it.
5. **Laws**: These are only formed when games are played in serious, competitive settings, where the stakes are high enough that a need is felt to explicitly record the rules of good sportsmanship or where there is need to clarify or modify the official written rules. These are often called "tournament rules," since during a serious tournament is when there is the most need for this kind of official clarification. Consider these tournament rules for playing *Tekken 5* (a fighting game) at the *2005 Penny Arcade Expo*:
   a. Single elimination.
   b. You may bring your own controller.
   c. Standard VS mode.
   d. 100% health.
   e. Random stage select.
   f. 60 second timer.
   g. Best 3 of 5 rounds.
   h. Best 2 of 3 games.
   i. Mokujin is banned.

   Most of these are just clarifying exactly which game settings will be used in the tournament. "You may bring your own controller" is a formalized decision about what is "fair play." The most interesting rule here is "Mokujin is banned."

Mokujin is one of the characters you can choose to play in *Tekken 5*. The general feeling among players is that Mokujin's "stun" move is so powerful that any player who chooses to play Mokujin is likely to win the game, making a tournament pointless. So this "law" is an attempt to improve the game, ensuring the tournament is balanced, fair, and fun.

6. **Official rules**: These are created when a game is played seriously enough that a group of players feels a need to merge the written rules with the laws. Over time, these official rules later become the written rules. In chess, when a player makes a move that puts the opponent's king in danger of checkmate, that player is obligated to warn the opponent by saying "check." At one time, this was a "law," not a written rule, but now it is part of the "official rules."
7. **Advisory rules**: Often called "rules of strategy," these are just tips to help you play better, and not really "rules" at all from a game mechanics standpoint.
8. **House rules**: These rules are not explicitly described by Parlett, but he does point out that as players play a game, they may find they want to tune the operational rules to make the game more fun. This is the "feedback" on his diagram, since house rules are usually created by players in response to a deficiency perceived after a few rounds of play.

Modes

Many games have very different rules during different parts of play. The rules often change completely from mode to mode, almost like completely separate games. One memorable instance was the racing game *Pitstop*. Most of the time, it was a typical racing game but with a twist—if you didn't pull over to change your tires periodically, they would burst. When you did pull over, the game changed completely—now you were not racing your car, but rather racing to change your tires, with a completely different game interface. When your game changes modes in a dramatic way like this, it is very important that you let your players know which mode you are in. Too many modes and the players can get confused. Very often, there is one main mode, with several submodes, which is a good hierarchical way to organize the different modes. Game designer Sid Meier proposes an excellent rule of thumb: players should never spend so much time in a subgame that they forget what they were doing in the main game.

Enforcer

One of the most significant differences between videogames and more traditional games is how the rules are enforced. In traditional games, rules are primarily enforced by the players themselves or by an impartial referee in high-stakes games, such as sporting events. With computer games, it becomes possible (and sometimes necessary) for the computer to enforce the rules. This is more than a convenience—it allows for the creation of games much more complex than was traditionally possible, because now the players don't have to memorize all the rules about what is and is not possible—they just try things in the game, and see what works and what doesn't work—they don't have to memorize it all, or look it up. In a sense, what used to be a "rule" now becomes a physical constraint of the game world. If a piece isn't allowed to move a certain way, it simply doesn't move that way. Many of the game rules are enforced by the design of the space, the objects, and the actions. A game like *Warcraft* could conceivably be a board game, but there would be so many rules to remember and state to keep track of that it would quickly become a dreary experience. By offloading the dull work of rules enforcement onto the computer, games can reach depths of complexity, subtlety, and richness that are not possible any other way. But proceed with caution—if the rules of your videogame are so complex that a player can't even form a rough idea of how the game works, they will be overwhelmed and confused. You must make the rules of a complex videogame something that players can discover and understand naturally—not something they have to memorize.

Cheatability

The reason games need an enforcer is to prevent cheating. Violating the rules is bad gamesmanship, certainly, but as we've seen throughout history, some players will stop at nothing to be perceived as the winner. Obviously, when you play a game, you want to make sure others don't cheat. But cheating has a more insidious effect than that—if players start believing your game is cheatable, even if it isn't, all the precious endogenous value you have worked to develop just slips away. Players imagine themselves working hard to win and imagine another player cheating, and it makes them feel like chumps. That is the danger of cheatability—if players feel like your game can be cheated, some will try to cheat, but most will just no longer want to play.

The Most Important Rule

Games have a lot of rules—how to move and what you can and cannot do—but there is one rule at the foundation of all the others: the object of the game. Games are about achieving goals—you must be able to state your game's goal and state it clearly. Often, there is not just one goal in a game, but a sequence of them—you will need to state each and how they relate to one another. A clumsy statement of your game's goal can be off-putting to players right from the beginning—if they don't completely understand the purpose of their actions, they cannot proceed with any certainty. Newcomers to chess are often stymied when someone awkwardly tries to explain the object of the game: "Your goal is to put the other king in checkmate… that means you move your pieces so he can't move without being in check… which, uh, means that one of your pieces could potentially capture him, except that, um, it's against the rules to capture the king." As a boy, I often wondered why a game considered to be so elegant could have such an inelegant goal. I played the game for years before I realized that the goal of chess is actually quite simple: "Capture your opponent's king." All the folderol about check and checkmate is simply there to politely warn your opponent that they are in imminent danger. It is remarkable how more interested a potential chess player becomes when you tell them that simple four-word goal. The same is true for any game you create—the more easily players understand the goal, the more easily they can visualize achieving it, and the more likely they are going to want to play your game.

When a goal is set in a player's mind, it gives them tremendous motivation to see it through. Having a clear set of well-constructed goals or quests is crucial to keeping your players engaged and motivated. Good game goals are as follows:

1. **Concrete**: Players understand and can clearly state what they are supposed to achieve.
2. **Achievable**: Players need to think that they have a chance of achieving the goal. If it seems impossible to them, they will quickly give up.
3. **Rewarding**: A lot goes into making an achieved goal rewarding. If the goal has the right level of challenge, just achieving it at all is a reward in itself. But why not go further? You can make your goal even more rewarding by giving the player something valuable upon reaching the goal—use Lens #20, *Pleasure*, to find different ways to reward the player and really make them proud of their achievement. And while it is important to reward players that achieve a goal, it is equally (or more) important that players appreciate that the goal is rewarding *before* they have achieved it, so that they are inspired to attempt to achieve it. Don't overinflate their expectations, though, for if they are disappointed with the reward for achieving a goal, they will not play again! We will talk much more about rewards in the next chapter.

And while it is important that each of the goals in your game has these qualities, it is also important that you have a good balance of goals in your game, with some short-term and some much longer term. This balance of goals will make your players feel they know what to do immediately and that ultimately they will achieve something important and magnificent.

It is easy to focus so much on the action of a game that you forget about the goals. To help us remember the importance of goals, let's add this lens to our toolbox.

Lens #32: The Lens of Goals

To ensure the goals of your game are appropriate and well balanced, ask yourself these questions:

- What is the ultimate goal of my game?
- Is that goal clear to players?
- If there is a series of goals, do the players understand that?
- Are the different goals related to each other in a meaningful way?
- Are my goals concrete, achievable, and rewarding?
- Do I have a good balance of short- and long-term goals?
- Do players have a chance to decide on their own goals?

It can be fascinating to pick up the Lens of the Toy, the Lens of Curiosity, and the Lens of Goals at the same time to see how these aspects of your game influence each other.

Wrapping Up Rules

Rules are the most fundamental of all game mechanics. A game is not just defined by its rules; a game *is* its rules. It is important to view your game from a rules perspective, and that is Lens #33.

Lens #33: The Lens of Rules

To use this lens, look deep into your game, until you can make out its most basic structure. Ask yourself these questions:

- What are the foundational rules of my game? How do these differ from the operational rules?
- Are there "laws" or "house rules" that are forming as the game develops? Should these be incorporated into my game directly?
- Are there different modes in my game? Do these modes make things simpler, or more complex? Would the game be better with fewer modes? More modes?

- Who enforces the rules?
- Are the rules easy to understand, or is there confusion about them? If there is confusion, should I fix it by changing the rules or by explaining them more clearly?

There is a common misconception that designers make games by sitting down and writing a set of rules. This usually isn't how it happens at all. A game's rules are arrived at gradually and experimentally. The designer's mind generally works in the domain of "operational rules," occasionally switching to the perspective of "foundational rules" when thinking about how to change or improve the game. The "written rules" usually come toward the end, once the game is playable. Part of the designer's job is to make sure there are rules that cover every circumstance. Be sure to take careful notes as you playtest, because it is during these tests that holes in your rules will appear—if you just patch them quickly and don't make a note, the same hole will just show up again later. A game is its rules—give them the time and consideration that they deserve.

Mechanic 6: Skill

In virtute sunt multi ascensus.

(There are many degrees in excellence.)

—Cicero

The mechanic of **skill** shifts the focus away from the game and onto the player. Every game requires players to exercise certain skills. If the player's skill level is a good match to the game's difficulty, the player will feel challenged and stay in the flow channel (as discussed in Chapter 10, "Player's Mind").

Most games do not just require one skill from a player—they require a blend of different skills. When you design a game, it is a worthwhile exercise to make a list of the skills that your game requires from the player. Even though there are thousands of possible skills that can go into a game, skills can generally be divided into three main categories:

1. **Physical skills**: These include skills involving strength, dexterity, coordination, and physical endurance. Physical skills are an important part of most sports. Effectively manipulating a game controller is a kind of physical skill, but many videogames (such as camera-based dance games) require a broader range of physical skills from players.
2. **Mental skills**: These include the skills of memory, observation, and puzzle solving. Although some people shy away from games that require too much in the

way of mental skills, it is the rare game that doesn't involve some mental skills, because games are interesting when there are interesting decisions to make, and decision making is a mental skill.

3. **Social skills**: These include, among other things, reading an opponent (guessing what he is thinking), fooling an opponent, and coordinating with teammates. Typically, we think of social skills in terms of your ability to make friends and influence people, but the range of social and communication skills in games is much wider. Poker is largely a social game, because so much of it rests on concealing your thoughts and guessing the thoughts of others. Sports are very social, as well, with their focus on teamwork and on "psyching out" your opponents.

Real vs. Virtual Skills

It is important to draw a distinction here: When we talk about skill as a game mechanic, we are talking about a **real skill** the player must have. In videogames, it is common to talk about your character's skill level. You might hear a player announce "My warrior just gained two points on his sword fighting skill!" But "sword fighting" is not a real skill required of the player—the player is really just pushing the right buttons on the control pad at the right time. Sword fighting, in this context, is a **virtual skill**—one that the player is pretending to have. The interesting thing about virtual skills is that they can improve even though the player's actual skill does not. The player might be just as sloppy at mashing the controller buttons as he ever was, but by mashing them enough times, he might be rewarded with a higher level of virtual skill, which allows his character to become a faster, more powerful swordfighter. Many "free-to-play" games have a whole monetization strategy based on the purchase of virtual skills.

Virtual skills are a great way to give a player a feeling of power. Taken too far, it can feel hollow—some critics of massively multiplayer games complain that there is too much emphasis on virtual skills, and not enough on real skills. Often, the key to a fun game is finding the right mix of real and virtual skills. Many novice designers confuse the two—it is important that you draw a clear distinction between them in your mind.

Enumerating Skills

Making a list of all the skills required in your game can be a very useful exercise. You might make a general list: "my game requires memory, problem solving, and pattern matching skills." Or you might make it very specific: "my game requires players to quickly identify and mentally rotate specific two-dimensional shapes in their heads, while solving a grid-based packing problem." Listing skills can be very tricky—one interesting example comes from the game *RC Pro Am*, a racing game for the NES. In it, players steer the car with the joypad (left thumb), accelerate with the

A button (right thumb), and fire weapons at opponents with the B button (also right thumb). To master this game, two surprising skills were required—the first was problem solving. Generally on NES games, you only push one button at a time—you take your thumb off the A button when you want to push the B button. But in *RC Pro Am*, this is disastrous—it means that if you want to fire a rocket (the B button), you have to release the car's accelerator (the A button), and your opponent quickly speeds away! How to solve this problem? Some players try using a thumb for one button and finger for the other, but this is awkward and makes the game too hard to play. The best solution seems to involve a new grip on the controller: you hold your thumb sideways on the A button, so that when you want to occasionally push the B button, you can roll it down onto the B button smoothly, without releasing the accelerator. Once the player has solved this problem, they then need to practice this very specific physical skill. And of course, there are many other skills involved in the game—managing resources (missiles and mines, so you don't run out), memorizing race courses, reacting to sharp turns and unexpected road hazards, and many more. The point is that even a game that seems somewhat simple might require many different skills from a player. As a designer, you need to know what these are.

It is easy to fool yourself into thinking your game is about one skill, when other skills are actually more important. Many action-based videogames seem, on the surface, to be mainly about quickly reacting to opponents, when in truth there is a lot of puzzle solving required to figure out the right way to react to them and a lot of memorization required to avoid being surprised next time you play a given level. Designers are often disappointed to realize that a game they thought was about quick decisions and thinking on your feet is really about memorizing which enemies pop out at what time—a very different (and much more tedious) experience for the player. The skills that a player exercises go a long way toward determining the nature of that player's experience, so you must know what these are. Viewing your game from this perspective is Lens #34.

Lens #34: The Lens of Skill

To use this lens, stop looking at your game, and start looking at the skills you are asking of your players.

Ask yourself these questions:

- What skills does my game require from the player?
- Are there categories of skill that this game is missing?
- Which skills are dominant?
- Are these skills creating the experience I want?

- Are some players much better at these skills than others? Does this make the game feel unfair?
- Can players improve their skills with practice, leading to a feeling of mastery?
- Does this game demand the right level of skill?

Exercising skills can be a joyful thing—it is one of the reasons that people love games. Of course, it is only joyful if the skills are interesting and rewarding and if the challenge level strikes that ideal balance between "too easy" and "too hard." Even dull skills (such as pushing buttons) can be made more interesting by dressing them up as virtual skills and providing the right level of challenge. Use this lens as a window into the experience the player is having. Because skills do so much to define experience, the Lens of Skill works quite well in conjunction with Lens #2, *Essential Experience*.

Mechanic 7: Chance

Our seventh and final game mechanic is **chance**. We deal with it last because it concerns interactions between all of the other six mechanics: space, time, objects, actions, rules, and skills.

Chance is an essential part of a fun game because chance means uncertainty, and uncertainty means surprises. And as we have discussed earlier, surprises are an important source of human pleasure and the secret ingredient of fun.

We must now proceed with caution. You can never take chance for granted, for it is very tricky—the math can be difficult, and our intuitions about it are often wrong. But a good game designer must become the master of chance and probability, sculpting it to his will, to create an experience that is always full of challenging decisions and interesting surprises. The challenges of understanding chance are well illustrated by a story about the invention of the mathematics of probability—invented, not surprisingly, for the express purpose of game design.

Invention of Probability

Il est tres bon ésprit, mais quel dommage, il n'est pas geometre.

(He's a nice guy, but unfortunately, no mathematician.)

—Pascal to Fermat regarding the Chevalier de Méré

It was the year 1654, and French nobleman Antoine Gombaud, the Chevalier de Méré (pronounced "Shevulyay duh Mayray"), had a problem. He was an avid gambler and had been playing a game where he would bet that if he rolled a single die four times, at least one time it would come up as a six. He had made some good money from this game, but his friends got tired of losing and refused to play it with him any further. Trying to find a new way to fleece his friends, he invented a new game that he believed had the same odds as the last one. In his new game, he would bet that if he rolled a pair of dice twenty-four times, a twelve would come up at least once. His friends were wary at first but soon grew to like his new game, because the Chevalier started losing money fast! He was confused, because by his math, both games had the same odds. Chevalier's reasoning was as follows:

First Game: In four rolls of a single die, the Chevalier wins if at least one six comes up.

The Chevalier reasoned that the chance of a single die coming up 6 was 1/6, and therefore rolling a die four times should mean the chance of winning was

$$4 \times (1/6) = 4/6 = 66\%, \text{ which explained why he tended to win.}$$

Second Game: In twenty-four rolls of a pair of dice, the Chevalier wins if at least one 12 comes up.

The Chevalier determined that the chance of getting a 12 (double sixes) on a pair of dice was 1/36. He reasoned, then, that rolling the dice 24 times meant the odds should be

$$24 \times (1/36) = 24/36 = 2/3 = 66\%. \text{ The same odds as the last game!}$$

Confused and losing money, he wrote a letter to mathematician Blaise Pascal, asking for advice. Pascal found the problem intriguing—there was no established mathematics to answer these questions. Pascal then wrote to his father's friend, Pierre de Fermat, for help. Pascal and Fermat began a lengthy correspondence about this and similar problems and, in discovering methods of solving them, established probability theory as a new branch of mathematics.

What are the real odds of Chevalier's games? To understand that, we have to get into some math—don't fret, it's easy math that anyone can do. Fully covering the mathematics of probability is not necessary for game design (and beyond the scope of this book), but knowing some of the basics can be quite handy. If you are a math genius, you can skip this section, or at least read it smugly. For the rest of us, I present the following:

Ten Rules of Probability Every Game Designer Should Know

Rule #1: Fractions Are Decimals Are Percents

If you are one of those people who has always had a hard time with fractions and percents, it's time to face up and deal with them, because they are the language of probability. Don't stress—you can always use a calculator—no one is looking.

The thing you have to come to grips with is that fractions, decimals, and percents are all the same thing and can be used interchangeably. In other words, ½ = 0.5 = 50%. Those aren't three different numbers; they are just three ways of writing exactly the same number.

Converting from fractions to decimals is easy. Need to know the decimal equivalent of 33/50? Just type 33 ÷ 50 into your calculator, and you'll get 0.66. What about percents? They're easy too. If you look up the word "percent" in the dictionary, you'll see that it really means "per 100." So, 66% really means 66 per 100, or 66/100, or 0.66. If you look at Chevalier's previous math, you'll see why we need to convert back and forth so often—as humans, we like to talk in percents, but we also like to talk about "one chance in six"—so we need a way to convert between these forms. If you are the kind of person who suffers from math anxiety, just relax and practice a few of these on the calculator—you'll have the hang of it in no time.

Rule #2: Zero to One—and That's It!

This one's easy. Probabilities can only range from 0% to 100%, that is, from 0 to 1 (see Rule #1), no less and no more. While you can say there is a 10% chance of something happening, there is no such thing as a –10% chance and certainly no such thing as a 110% chance. A 0% chance of something happening means it won't happen, and a 100% chance means it definitely will. This all might sound obvious, but it points out a major problem with Chevalier's math. Consider his first game with the four dice. He believed that with four dice, he had a 4 × (1/6), or 4/6, or 0.66, or 66% chance of having a six come up. But what if he had seven dice? Then he would have had 7 × (1/6) or 7/6 or 1.17 or 117% chance of winning! And that is certainly wrong—if you roll a die seven times, it might be likely that a six will come up one of those times, but it is not guaranteed (in fact, it is about a 72% chance). Anytime you calculate a probability that comes up greater than 100% (or less than 0%), you know for certain that you've done something wrong.

Rule #3: "Looked For" Divided By "Possible Outcomes" Equals Probability

The first two rules lay some basic groundwork, but now we are going to talk about what probability really is—and it is quite simple. You just take the number of times your "looked for" outcome can come up and divide by the number of possible outcomes (assuming your outcomes are equally likely), and you've got it. What is the chance of a six coming up when you roll a die? Well, there are six possible outcomes, and only one of them is the one we are looking for, so the chance of a six coming up is 1 ÷ 6, or 1/6, or about 17%. What is the chance of an even number coming up when you roll a die? There are 3 even numbers, so the answer is 3/6, or 50%. What is the chance of drawing a face card from a deck of cards? There are twelve face cards in a deck, and fifty-two cards total, so your chances of getting a face card are 12/52, or about 23%. If you understand this, you've got the fundamental idea of probability.

Rule #4: Enumerate!

If Rule #3 is as simple as it sounds (and it is), you might wonder why probability is so tricky. The reason is that the two numbers we need (the number of "looked for" outcomes and the number of possible outcomes) are not always so obvious. For example, if I asked you what the odds of flipping a coin three times and getting "heads" at least twice, what is the number of "looked for" outcomes? I'd be surprised if you could answer that without writing anything down. An easy way to find out the answer is to enumerate all the possible outcomes:

1. HHH
2. HHT
3. HTH
4. HTT
5. THH
6. THT
7. TTH
8. TTT

There are exactly eight possible outcomes. Which ones have heads at least twice? #1, #2, #3, and #5. That's 4 outcomes out of 8 possibilities, so the answer is 4/8, or a 50% chance. Now, why didn't the Chevalier do this with his games? With his first game, there were four die rolls, which means 6 × 6 × 6 × 6, or 1296 possibilities. It would have been dull work, but he could have enumerated all the possibilities in an hour or so (the list would have looked like 1111, 1112, 1113, 1114, 1115, 1116, 1121, 1122, 1123, etc.), then counted up the number of combinations that had a six in them (671), and divided that by 1296 for his answer. Enumeration will let you solve almost any probability problem, if you have the time. Consider the Chevalier's second game, though: 24 rolls of 2 dice! There are 36 possible outcomes for 2 dice, and so enumerating all 24 rolls would have meant writing down 36^{24} (a number 37 digits long) combinations. Even if he could somehow write down one combination a second, it would have taken longer than the age of the universe to list them all. Enumeration is handy, but when it takes too long, you need to take shortcuts—and that's what the other rules are for.

Rule #5: In Certain Cases, OR Means Add

Very often, we want to determine the chances of "this OR that" happening, such as what are the chances of drawing a face card OR an ace from a deck of cards? When the two things we are talking about are mutually exclusive, that is, when it is impossible for both of them to happen simultaneously, you can add their individual probabilities to get an overall probability. For example, the chances of drawing a face card are 12/52, and the chances of drawing an ace are 4/52. Since these are mutually exclusive events (it is impossible for them both to happen at once), we can add them up: 12/52 + 4/52 = 16/52, or about a 31% chance.

But what if we asked a different question: What are the chances of drawing an ace from a deck of cards or a diamond? If we add these probabilities, we get 4/52 + 13/52 (13 diamonds in a deck) = 17/52. But, if we enumerate, we see this is wrong—the right answer is 16/52. Why? Because the two cases are not mutually exclusive—I could draw the ace of diamonds! Since this case is not mutually exclusive, "or" does not mean add.

Let's look at Chevalier's first game. He seems to be trying to use this rule for his die rolls—adding up four probabilities: 1/6 + 1/6 + 1/6 + 1/6. But he gets the wrong answer, because the four events are not mutually exclusive. The addition rule is handy, but you must be certain the events you are adding up are mutually exclusive from one another.

Rule #6: In Certain Cases, AND Means Multiply

This rule is almost the opposite of the previous one! If we want to find the probability of two things happening simultaneously, we can multiply their probabilities to get the answer—but ONLY if the two events are NOT mutually exclusive! Consider two die rolls. If we want to find the probability of rolling a six on both rolls, we can multiply together the probabilities of the two events: The chance of getting a six on one die roll is 1/6, and also 1/6 for a second die roll. So the chance of getting two sixes is 1/6 × 1/6 = 1/36. You could also have determined that by enumeration, of course, but this is a much speedier way to do it.

In Rule #5, we asked for the probability of drawing an ace OR a diamond from a deck of cards—the rule failed, because the two events were not mutually exclusive. So what if we asked about the probability of drawing an ace AND a diamond? In other words, what is the probability of drawing the ace of diamonds? It should be fairly intuitive that the answer is 1/52, but we can check that with Rule #6, since we know the two events are not mutually exclusive. The chance of getting an ace is 4/52, and the chance of a diamond is 13/52. Multiplying them, 4/52 × 13/52 = 52/2704 = 1/52. So, the rule works and matches our intuition.

Do we have enough rules yet to solve Chevalier's problems? Let's consider his first game:

First Game: In four rolls of a single die, the Chevalier wins if at least one six comes up.

We've already established that we could enumerate this and get the answer 671/1296, but that would take an hour. Is there a quicker way, using the rules we have?

(I'll warn you now—this gets a little hairy. If you don't really care that much, save yourself the headache, and just skip to Rule #7. If you do care, then press on—you will find it worth the effort.)

If the question was about the chances of rolling a die four times and getting four sixes, that would be an AND question for four events that are not mutually exclusive, and we could just use Rule #6: $1/6 \times 1/6 \times 1/6 \times 1/6 = 1/1296$. But that isn't what is asked. This is an OR question for four events that are not mutually exclusive (it is possible for the Chevalier to get multiple sixes on the four rolls). So what can we do? Well, one way is to break it down into events that are mutually exclusive and then add them up. Another way to phrase this game is

What are the chances of rolling four dice, and getting either

a. Four sixes, OR
b. Three sixes and one non-six, OR
c. Two sixes and two non-sixes, OR
d. One six and three non-sixes

That might sound a little complicated, but it is four different mutually exclusive events, and if we can figure the probability of each, we can just add them up and get our answer. We've already figured out the probability of (a), using Rule #6: 1/1296. So, how about (b)? Really, (b) is four different mutually exclusive possibilities:

1. 6, 6, 6, non-six
2. 6, 6, non-six, 6
3. 6, non-six, 6, 6,
4. Non-six, 6, 6, 6

The probability of rolling a six is 1/6, the probability of rolling a non-six is 5/6. So, the probability of each of those is $1/6 \times 1/6 \times 1/6 \times 5/6 = 5/1296$. Now, if we add up all four, that comes to 20/1296. So, the probability of (b) is 20/1296.

How about (c)? This one is the same as the last, but there are more combinations. It is tricky to figure out how many ways there are for exactly two sixes and two non-sixes to come up, but there are six ways:

1. 6, 6, non-six, non-six
2. 6, non-six, 6, non-six
3. 6, non-six, non-six, 6
4. non-six, 6, 6, non-six
5. non-six, 6, non-six, 6
6. non-six, non-six, 6, 6

And the probability of each of these is $1/6 \times 1/6 \times 5/6 \times 5/6 = 25/1296$. Adding up all six of them comes to 150/1296.

This leaves only (d), which is the inverse of (b):

a. Non-six, non-six, non-six, 6
b. Non-six, non-six, 6, non-six
c. Non-six, 6, non-six, non-six
d. 6, non-six, non-six, non-six

The probability of each is 5/6 × 5/6 × 5/6 × 1/6 = 125/1296. Adding up all four gives 500/1296.

So, we have now calculated the probability of the four mutually exclusive events:

a. Four sixes—(1/1296)
b. Three sixes and one non-six—(20/1296)
c. Two sixes and two non-sixes—(150/1296)
d. One six and three non-sixes—(500/1296)

Adding up those four probabilities (as Rule #5 allows) gives us a total of 671/1296, or about 51.77%. So, we can see that this was a good game for the Chevalier—by winning more than 50% of the time, he eventually was likely to make a profit, but the game was close enough to even that his friends believed they had a chance—at least for a while. It certainly is a very different result than the 66% chance of winning the Chevalier believed he had!

This is the same answer we could have gotten from enumeration, but much faster. Really, though, we did a kind of enumeration—it is just that the rules of addition and multiplication let us count everything up much faster. Could we do the same thing to get the answer to Chevalier's second game? We could, but with 24 rolls of two dice, it would probably take an hour or more! This is faster than enumeration, but we can do even better by being tricky—that's where Rule #7 comes in.

Rule #7: One Minus "Does" = "Doesn't"

This is a more intuitive rule. If the chance of something happening is 10%, the chance of it not happening is 90%. Why is this useful? Because often it is quite hard to figure out the chance of something happening but easy to figure out the chance of it NOT happening.

Consider Chevalier's second game. To figure out the chance of double sixes coming up at least once on twenty-four die rolls would be nightmarish to figure out, because you have so many different possible events to add together (1 double sixes, 23 non–double sixes; 2 double sixes, 22 non–double sixes; etc.). On the other hand, what if we ask a different question: What are the chances of rolling two dice twenty-four times and NOT getting double sixes? That is now an AND question, for events that are not mutually exclusive, so we can use Rule #6 to get the answer! But first we'll use Rule #7 twice—watch.

The chance of double sixes coming up on a single roll of the dice is 1/36. So, by Rule #7, the chance of not getting double sixes is 1 − 1/36, or 35/36.

So, using Rule #6 (multiplication), the chances of not getting double sixes 24 times in a row is 35/36 × 35/36 twenty-four times, or as we say $(35/36)^{24}$. You would not want to do this calculation by hand, but using a calculator, you find the answer is around 0.5086, or 50.86%. But that is the chance of the Chevalier losing. To find the chance of the Chevalier winning, we apply Rule #7 again: 1 − 0.5086 = 0.4914, or about 49.14%. Now it is clear why he lost this game! His chances of winning were close enough to even that it was hard for him to tell if this was a winning or losing game, but after playing many times, he was very likely to lose.

Even though all probability problems can be solved through enumeration, Rule #7 can be a really handy shortcut. In fact, we could have used the same rule to solve Chevalier's first game!

Rule #8: The Sum of Multiple Linear Random Selections Is NOT a Linear Random Selection!

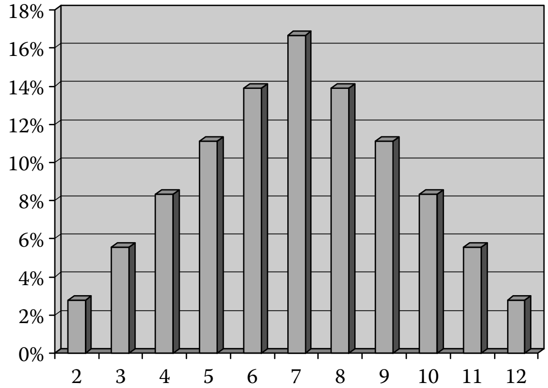
Don't panic. This one sounds hard, but it is really easy. A "linear random selection" is simply a random event where all the outcomes have an equal chance of happening. A die roll is a great example of a linear random selection. If you add up multiple die rolls, though, the possible outcomes do NOT have an equal chance of happening. If you roll two dice, for example, your chance of getting a seven is very good, while your chance of getting a twelve is small. Enumerating all the possibilities shows you why:

	1	**2**	**3**	**4**	**5**	**6**
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

Look at how many 7's there are and only one little twelve! We can show this in a graph, called a probability distribution curve, to visually see the chances of each total coming up:

Rule #8 might seem like a very obvious rule, but I frequently find novice game designers make the mistake of adding together two randomly selected numbers without realizing its effect. Sometimes, it is exactly the effect you want—in the game *Dungeons and Dragons*, players generate (virtual) skill attributes with values ranging from 3 to 18 by rolling three six-sided dice. As a result, you see a lot of attribute values around 10 or 11, but very few at 3 or 18, and this is exactly what the

FIGURE
12.15



designers wanted. How would the game be different if players simply rolled a single twenty-sided die to get their attributes?

Game designers who want to use mechanic of chance as a tool in their games must know what kind of probability distribution curve they want and know how to get it. With practice, probability distribution curves will be a very valuable tool in your toolbox.

Rule #9: Roll the Dice

All the probability we've been talking about so far is **theoretical probability**, that is, mathematically, what *ought to* happen. There is also **practical probability**, which is a measure of what *has* happened. For example, the theoretical probability of getting a 6 when I roll a die is a perfect 1/6, or about 16.67%. I could find the practical probability by rolling a six-sided die 100 times and recording how many times I get a six. I might record 20 sixes out of 100. In that case, my practical probability is 20%, which is not too far from the theoretical probability. Of course, the more trials I do, the closer I would expect the practical probability to get to the theoretical probability. This is sometimes known as the "Monte Carlo" method, after the famous casino.

The great thing about the Monte Carlo method of determining probability is that it doesn't involve any complex math—you just repeat the test over and over again and record how it comes out. It can sometimes give more useful results than theoretical probability too, because it is a measure of the real thing. If there is some factor that your mathematics didn't capture (e.g., perhaps your die is slightly weighted toward sixes), or if the math is just so complicated that you can't come up with a theoretical representation of your case, the Monte Carlo method can be just the thing. The Chevalier could easily have found good answers to his questions by just rolling the dice again and again, counting up wins, and dividing by the number of trials.

And here in the computer age, if you know how to do a little bit of programming (or know someone who can—see Rule #10), you can easily simulate millions of trials in just a few minutes. It isn't too hard to program simulations of games and get some very useful probability answers. For example, in Monopoly, which squares are landed on most frequently? It would be nearly impossible to figure this out theoretically—but a simple Monte Carlo simulation allows you to answer the question quickly by using a computer to roll the dice and move the pieces around the board a few million times. Alternatively, you could make use of the *Machinations* system created by Joris Dormans, which is specifically designed to model gameplay systems and show patterns of results through repeated simulations.

Rule #10: Geeks Love Showing Off (Gombaud's Law)

This is the most important of all the probability rules. If you forget all the others but remember this one, you'll get by just fine. There are many more difficult aspects of probability that we won't get into here—when you run into them, the easiest thing to do is to find someone who considers themselves a "math whiz." Generally, these people are thrilled to have someone actually needing their expertise, and they will bend over backwards to help you. I have used Rule #10 to solve hard game design probability questions again and again. If there aren't any experts around you, post your question on a forum or mailing list. If you really want a fast response, preface it with "This problem is probably too difficult for anyone to solve, but I thought I would ask anyway," for there are many math experts who love the ego boost of solving a problem that others think is impossible. In a sense, your hard problem is a game for them—why not use game design techniques to make it as attractive as possible?

You might even be doing your geek a favor! I like to call Rule #10 "Gombaud's Law," in honor of Antoine Gombaud, the Chevalier de Méré, who, through his awareness of this principle, not only solved his gambling problem (his mathematical one, anyway), but inadvertently initiated all of probability theory.

You might be afraid of exercising Rule #10, because you are afraid of asking stupid questions. If you feel that way, don't forget that Pascal and Fermat owed the Chevalier a great debt—without his stupid questions, they never would have made some of their greatest discoveries. Your stupid question might lead to a great truth of its own—but you'll never know unless you ask.

Expected Value

You will use probability in many ways in your designs, but one of the most useful will be to calculate **expected value**. Very often, when you take an action in a game, the action will have a value, either positive or negative. This might be points, tokens, or money gained or lost. The expected value of a transaction in a game is the average of all the possible values that could result.

For example, there might be a rule in a board game that when a player lands on a green space, he can roll a six-sided die and get that many power points. The expected value of this event is the average of all the possible outcomes. To get the average in this case, since all the probabilities are equal, we can add up all the possible die rolls, 1 + 2 + 3 + 4 + 5 + 6 = 21, and divide by 6, which gives us 3.5. As a game designer, it is very useful for you to know that each time someone lands on a green space, they will, on average, get 3.5 power points.

But not all examples are so simple—some involve negative outcomes, and outcomes that aren't evenly weighted. Consider a game where a player rolls two dice. If they get a 7's, or an 11's, they win $5, but if they get anything else, they lose $1. How do we figure out the expected value of this game?

The chance of rolling a 7 is 6/36.

The chance of rolling an 11 is 2/36.

Using Rule #8, the chance of rolling anything else is 1 − 8/36, or 28/36.

So, to calculate the expected value, we multiply the probabilities by the values for each and add them all up, like this:

Outcome	Chance×Outcome	Value
7	6/36×$5	$0.83
11	2/36×$5	$0.28
Everything else	28/36×− $1	−$0.78
Expected value		$0.33

So, we see that this is a good game to play, because in the long run, you will, on average, win thirty-three cents each time you play. But, what if we changed the game, so that only 7's are winning numbers and 11's make you lose a dollar, just like all the other numbers? This changes the expected value, like this:

Outcome	Chance×Outcome	Value
7	6/36×$5	$0.83
Everything else	30/36×− $1	−$0.83
Expected value		$0.00

An expected value of zero means that this game is just as good as flipping a coin in the long run. Wins and losses are completely balanced. What if we change it again, so that this time only eleven wins?

Outcome	Chance×Outcome	Value
11	2/36×$5	$0.28
Everything else	34/36×− $1	−$0.94
Expected value		−$0.86

Ouch! As you might expect, this is a losing game. You'll lose, on average, about eighty-six cents each time you play it. Of course, you could make it into a fair game, or even a winning game, by increasing the payoff for getting an eleven.

Consider Values Carefully

Expected value is an excellent tool for game balancing, which we will discuss more in the next chapter—but if you aren't careful about what the true value of an outcome is, it can be very misleading.

Consider these three attacks that might be part of a fantasy role-playing game:

Attack Name	Chance of Hitting (%)	Damage
Wind	100	4
Fireball	80	5
Lightning bolt	20	40

What is the expected value of each of these? Wind is easy—it always does exactly 4 damage, so the expected value of that attack is 4. Fireball hits 80% of the time and misses 20% of the time, so it's expected value is $(5 \times 0.8) + (0 \times 0.2) = 4$ points, the same as the wind attack. The lightning bolt attack doesn't hit very often, but when it does, it packs a wallop. Its expected value is $(40 \times 0.2) + (0 \times 0.8) = 8$ points.

Now, based on those values, one might conclude that players would always use the lightning bolt attack, since on average it does double the damage of the other two attacks. And if you are fighting an enemy that has 500 hit points, that might be correct. But what about an enemy with 15 hit points? Most players would not use lightning bolt in that case—they would opt for something weaker but surer. Why is this? Because even though the lightning bolt can do 40 damage points, only 15 of them are of any use in that situation—the real expected value of the lightning bolt against an enemy with 15 HP is $(0.2 \times 15) + (0.8 \times 0) = 3$ points, which is lower than both the wind and the fireball attack.

You must always take care to measure the real values of actions in your game. If something gives a benefit that a player can't use, or contains a hidden penalty, you must capture that in your calculations.

Human Element

You must also keep in mind that expected value calculations do not perfectly predict human behavior. You would expect players to always choose the option with the highest expected value, but that is not always the case. In some cases, this is due to ignorance—because players did not realize the actual expected value. For example, if you didn't tell players the respective chances of wind, fireball, and lightning bolt, but left it to them to discover them through trial and error, you might find that players who tried lightning bolt several times and never got a hit reached the conclusion that "lightning bolt never hits" and therefore has an expected value of zero. The estimates that players make about how often an event happens are often incorrect. You must be aware of the "perceived probabilities" that players have arrived at, because it will determine how they play.

But sometimes, even with perfect information, players still will not choose an option with the highest expected value. Two psychologists, Kahneman and Tversky, tried an interesting experiment, where they asked a number of subjects which of the two games they would like to play:

Game A:

66% chance of winning $2400
33% chance of winning $2500
1% chance of winning $0

Game B:

100% chance of winning $2400

These are both pretty great games to play! But is one better than the other? If you do the expected value calculations

$$\text{Expected Value of Game A: } 0.66 \times \$2400 + 0.33 \times \$2500 + 0.01 \times \$0 = \$2409$$

$$\text{Expected Value of Game B: } 1.00 \times 2400 = \$2400$$

You can see that Game A has a higher expected value. But only 18% of the subjects they surveyed picked A, while 82% preferred playing Game B.

Why? The reason is that the expected value calculation does not capture an important human element: regret. People not only seek out options that create the most pleasure, they also avoid the ones that cause the most pain. If you played Game A (and we're assuming you only get to play it once), and were unlucky enough to get that 1% and $0, it would feel pretty bad. People are often willing to pay a price to eliminate the potential of regret—"buying peace of mind," as the insurance salesmen say. Not only are they willing to pay a price to avoid regret, they are willing to take risks. This is why a gambler who has lost a little money is often willing to take more risks to try to get the money back. Tversky puts it this way: "When it comes to taking risks for gains, people are conservative. They will make a sure gain over a problem gain. But we are also finding that when people are faced with a choice between a small, certain loss and a large, probable loss, they will gamble." This appears to be a large part of the success of the "free-to-play" game *Puzzle & Dragons*. Players perform a series of puzzles and rack up treasures while making their way through a dungeon. Sometimes, though, they perish in the dungeon, and the game effectively says, "Oh, that's too bad, you're dying. Look at all the treasure you are going to lose. Are you sure you don't want to pay just a little bit of real money, so you have a shot at keeping what you have earned?" And many people respond by paying cash money to avoid that small, certain loss.

In some cases, the human mind inflates some risks completely out of proportion. In one study, Tversky asked people to estimate the likelihood of various causes of death and obtained the following results:

Cause of Death	Estimated Chance (%)	Actual Chance (%)
Heart disease	22	34
Cancer	18	23
Other natural causes	33	35
Accident	32	5
Homicide	10	1
Other unnatural causes	11	2

What is particularly interesting here is that the subjects making estimates *under-estimated* the top three categories (natural causes of death) and significantly *over-estimated* the bottom three (unnatural causes of death). This distortion of reality seems to be a reflection of the fears of the respondents. What bearing does this have on game design? As a designer, you must have not only a grasp of the actual probabilities of events in your game but also the *perceived* probabilities, which may be quite different for a number of reasons.

You will need to consider both actual and perceived probabilities when calculating expected values, which provide such useful information that they make Lens #35.

Lens #35: The Lens of Expected Value

To use this lens, think about the chance of different events occurring in your game and what those mean to your player.

Ask yourself these questions:

- What is the actual chance of a certain event occurring?
- What is the perceived chance?
- What value does the outcome of that event have? Can the value be quantified? Are there intangible aspects of value that I am not considering?
- Each action a player can take has a different expected value when I add up all the possible outcomes. Am I happy with these values? Do they give the player interesting choices? Are they too rewarding, or too punishing?

Expected value is one of your most valuable tools for analyzing game balance. The challenge of using it is finding a way to numerically represent everything that can happen to a player. Gaining and losing money is easy to represent. But what is the numerical value of "boots of speed" that let you run faster or a "warp gate" that lets you skip two levels? These are difficult to quantify perfectly—but that doesn't mean you can't take a guess. As we'll see in the next chapter, as you go through multiple iterations of game testing and tweaking parameters and values in your game, you will also be tweaking your own estimations of the values of different outcomes. Quantifying these less tangible elements can be quite enlightening, because it makes you think concretely about what is valuable to the player and why—and this concrete knowledge will put you in control of the balance of your game.

Skill and Chance Get Tangled

As tricky as probability and the difference between actual and perceived values might be, the game mechanic of chance has more tricks up its sleeve. As much as we like to think that chance and skill are completely separate mechanics, there are important interactions between them that we cannot ignore. Here are five of the most important skill/chance interactions for a game designer to consider.

1. **Estimating chance is a skill**: In many games, what separates the skilled players from the unskilled is their ability to predict what is going to happen next, often through calculating probabilities. The game of blackjack, for example, is almost entirely about knowing the odds. Some players even practice "card counting," which is the practice of keeping track of what cards have already been played, since each card played changes the odds of what subsequent cards can appear. The perceived probabilities in your game can vary a great deal between players who are skilled estimators and those who are not.
2. **Skills have a probability of success**: Naively, one might think that completely skill-based games, such as chess or baseball, have no aspects of randomness or risk in them. But from a player's point of view, this simply isn't true. Every action has some level of risk, and players are constantly making expected value decisions, deciding when to play it safe and when to take a big risk. These risks can be difficult to quantify (what are the odds that I can successfully steal a base or that I can trap my opponent's queen without him noticing?), but they are still risks. When designing a game, you need to make sure they are balanced just as you would balance "pure chance" game elements, like drawn cards or die rolls.
3. **Estimating an opponent's skill is a skill**: A big part of a player's ability to determine the chances of success for a particular action rests on their ability to estimate their opponent's skill. A fascinating part of many games is trying to fool your opponent into thinking your skills are greater than they are, to prevent him from trying

anything too bold and to make him uncertain of himself. Likewise, sometimes the opposite is true—in some games, it is a good strategy to make a player think your skills are less than they really are, so that your opponent will not notice your subtle strategies and will perhaps try actions that would be risky against a skilled player.

4. **Predicting pure chance is an imagined skill**: Humans look for patterns, consciously and subconsciously, to help predict what is going to happen next. Our mania for patterns often leads us to look for and find patterns where none exist. Two of the most common false patterns are the "lucky streak fallacy" (I've had several wins in a row, and therefore another is likely) and, its opposite, the "gambler's fallacy" (I've had several losses, so I must be due for a win). It is easy to scoff at these as ignorant, but in the all-important mind of the player, detecting these bogus patterns feels like the exercise of a real skill, and as a designer, you should find ways to use that to your advantage.
5. **Controlling pure chance is an imagined skill**: Not only do our brains actively seek patterns, but they also actively and desperately seek cause-and-effect relationships. With pure chance, there is no way to control the outcome—but that doesn't stop people from rolling the dice a certain way, carrying lucky charms, or engaging in other superstitious rituals. This feeling that it might be possible to control fate is part of what makes gambling games so exciting. Intellectually, we know it isn't possible, but when you are up there rolling the dice, saying "come on, come on…," it certainly feels like it might be possible, especially when you get lucky! If you try playing games of pure chance, but completely disengage yourself from the idea that anything you think or do can influence the outcome, much of the fun suddenly drains away. Our natural tendency to try to control fate can make games of chance feel like games of skill.

Chance is tricky stuff, because it intertwines hard math, human psychology, and all of the basic game mechanics. But this trickiness is what gives games their richness, complexity, and depth. The last of our seven basic game mechanics gives us Lens #36.

Lens #36: The Lens of Chance

To use this lens, focus on the parts of your game that involve randomness and risk, keeping in mind that those two things are not the same.

Ask yourself these questions:

- What in my game is truly random? What parts just feel random?
- Does the randomness give the players positive feelings of excitement and challenge, or does it give them negative feelings of hopelessness and lack of control?

- Would changing my probability distribution curves improve my game?
- Do players have the opportunity to take interesting risks in the game?
- What is the relationship between chance and skill in my game? Are there ways I can make random elements feel more like the exercise of a skill? Are there ways I can make exercising skills feel more like risk taking?

Risk and randomness are like spices. A game without any hint of them can be completely bland, but put in too much and they overwhelm everything else. But get them just right, and they bring out the flavor of everything else in your game. Unfortunately, using them in your game is not as simple as sprinkling them on top. You must look into your game to see where elements of risk and randomness naturally arise, and then decide how you can best tame them to do your bidding. Don't fall into the trap of thinking that elements of chance only occur around die rolls or randomly generated numbers. On the contrary, you can find them wherever a player encounters the unknown.

At long last, we have made it through all seven of the basic game mechanics. Soon, we will move onto more advanced mechanics that are built from these, such as puzzles and interactive story structures. But first, we need to explore methods of bringing these basic elements into balance.

Other Reading to Consider

***Game Mechanics*: *Advanced Game Design* by Ernest Adams and Joris Dormans**. This book gets into a lot of wonderfully nitty-gritty details about the interactions of various game mechanics and gives an introduction to the fascinating *Machinations* system for simulating your game design.

***The Oxford Book of Board Games* by David Parlett**. Contains more details on Parlett's Rule Analysis, as well as descriptions of some amazing but little-known board games from previous centuries.

***Uncertainty in Games* by Greg Costikyan**. An incredibly insightful book about the nature of chance and uncertainty in games. I get something new from it every time I read it.

***The Unfinished Game*: *Pascal, Fermat, and the Unfinished Letter that Made the World Modern* by Keith Devlin**. If you want even more details of the story of how probability came to be, this is the definitive book.