Chapter 2. Emergence and Progression

In the previous chapter, we introduced five types of game mechanics: physics, internal economy, progression, tactical maneuvering, and social interaction. Of these categories, the mechanics of progression create what in game studies are called *games of progression*. The other four types of mechanics correspond fairly well to another category, *games of emergence*. For ease of reference, we will call the other four types of mechanics *mechanics of emergence* in this chapter.

The two categories of games of emergence and games of progression are considered important, alternative ways of creating gameplay. In this chapter, we explore this important distinction in more detail and provide examples of each category. We also explore the structural differences in the mechanics that generate emergence and progression and the problems and opportunities they create when a designer tries to integrate emergence and progression in a single game.

The History of Emergence and Progression

The categories of emergence and progression were originally introduced by game scholar Jesper Juul in his paper "The Open and the Closed: Games of Emergence and Games of Progression" (2002). Put simply, games of emergence are those games that have relatively simple rules but much variation. We use the term *emergence* because the game's challenges and its flow of events are not planned in advance but emerge during play. Emergence is produced by the many possible combinations of rules in board games, card games, strategy games, and some action games. According to Juul, "Emergence is the primordial game structure" (p. 324); that is, the earliest games were games of emergence, and in creating a new game, many people begin with emergent designs.

Games of this type can be in many different configurations, or states, during play. All possible arrangements of the playing pieces in chess constitute different game states, because the displacement of a single pawn by even one square can make a critical difference. The number of possible combinations of pieces on a chess board is huge, yet the rules easily fit on a single page. Something similar can be said of the placements of residential zones in the simulation game *SimCity* or the placement of units in the strategy game *StarCraft*.

Emergence and Progression Outside Video Games

In Juul's categorization, all board games are games of emergence. Games that start with randomized elements, such as cards or dominoes, also qualify. Such games typically have a small number of pieces and little or no predesigned data. The text on *Monopoly's* Chance and Community Chest cards are examples of predesigned data, but they require less than 1KB to store.

A game of progression requires a large amount of data, prepared in advance by the designer, that the player can access at arbitrary points (called *random access*). This is inconvenient for board games but easy for video games now that they can store many gigabytes of data. Progression is the newer structure, starting with the text-adventure games from the 1970s. However, progression is not limited to games running on computers. Pen-and-paper role-playing games like *Dungeons & Dragons* offer published scenarios, and these scenarios also constitute games of progression, as do the books in the Choose Your Own Adventure book series. Books are another medium that can handle a large amount of data and offer easy random access.

Tip

Don't confuse the term *games of progression* with other ideas about progression in games, such as leveling up, difficulty curves, skill trees, and so on. We use Juul's definition of the term: A game of progression is one that offers predesigned challenges, each of which often has exactly one solution, in a fixed (or only slightly variable) sequence.

In contrast, games of progression offer many predesigned challenges that the designer has ordered sequentially, usually through sophisticated level design. Progression relies on a tightly controlled sequence of events. A game designer dictates the challenges that a player encounters by designing levels in such a way that the player must encounter these events in a particular sequence. According to Juul, any game that has a walkthrough is a game of progression. In its most extreme form, the player is "railroaded" through a game, going from one challenge to the next or failing in the attempt. In a game of progression, the number of game states is relatively small, and the designer has total control over what is put in the game. This makes games of progression well suited to games that tell stories.

Comparing Emergence and Progression

In his original article, Juul expresses a preference for games that include emergence: "On a theoretical level, emergence is the more interesting structure" (2002, p. 328). He regards emergence as an approach that allows designers to create games in which the freedom of the player is balanced with the control of the designer. In a game of emergence, designers do not specify every event in detail before the game is published, though the rules may make certain events very likely. In practice, however, a game with an emergent structure often still follows fairly regular patterns. Juul discusses the gun fights that almost always erupt in a game of *Counter-Strike* (p. 327). Another example can be found in *Risk,* in which the players' territories are initially scattered all over the map, but over the course of play their ownership changes, and the players generally end up controlling one or a few areas of neighboring territories.

Data and Process Intensity

The game designer Chris Crawford's notions of *process intensity* and *data intensity* apply to progression and emergence in games. Computers differ from most other gaming media because computers are good at processing numbers. Computers also allow fast access to random locations within a large database, an ability put to good use within games of progression. But it is the ability to create new content on the fly and handle complex simulations where computers really shine. Like no other medium before, computers have the capacity to surprise players and designers with clever simulations and emergent gameplay. Crawford believes games should capitalize on this ability of the computer: Games should be process-intensive, rather than data-intensive. He says that video games should be games of emergence rather than games of progression.

In his later book *Half-Real*, Juul is more nuanced in his discussion of emergence and progression (2005). Most modern video games are hybrids; they include some features of both. *Grand Theft Auto: San Andreas* provides a vast open world but also has a mission structure that introduces new elements and unlocks this world piece by piece. In the story-driven first-person shooter game *Deus Ex*, the storyline dictates where the player needs to go next, but players have many different strategies and tactics available to deal with the problems they encounter on the way. It is possible to write a walkthrough for *Deus Ex*, which would make it a game of progression according to Juul's classification, but there are many possible walkthroughs for *Deus Ex*—just as, at least in theory, it is possible to create a walkthrough for a particular map in *SimCity*, instructing the player to build certain zones or infrastructure at a particular time in order to build an effective city. It would be hard to follow such a walkthrough, but creating one is possible.

Emergence is not better than progression. They are simply different. Pure games of emergence and pure games of progression represent two extremes on a bipolar scale. Many casual games, such as *Bejeweled*, are pure games of emergence. Pure games of progression are fairly rare. The most typical examples are adventure games such as *The Longest Journey,* but they are no longer the dominant genre they once were. Other games include elements of both, often by exhibiting emergent behavior *within* a given level but offering their levels in a strict sequence from which the player cannot depart (progressive behavior). Today, action-adventure games such as *Half-Life* and the *Legend of Zelda* series are much more common than traditional adventure games: Action-adventures include some form of emergent action as part of the gameplay. Among large games, hybrid forms are the most popular.

Games of Emergence

The use of the term *emergence* in games, which predates Juul's categories, originates from the use of the term in complexity theory. There it refers to behavior of a system that cannot be derived (directly) from its constituent parts. At the same time, Juul cautions us not to confuse emergent behavior with games that display behavior the designer simply did not foresee (2002). In games, as in any complex system, the whole is more than the sum of its parts. Go and chess are famous for generating enormous depth of play with relatively simple elements and rules. Something similar can be said of relatively simple computer games such as *Tetris, Boulder Dash,* or *World of Goo*. These games consist of relatively simple parts, yet the number of strategies and approaches that they allow is enormous. No two play-throughs will feel the same. The emergent quality of the gameplay comes not from the complexity of individual parts but from the complexity that is the result of the many interactions among the parts.

Simple Parts in Complex Systems

The science of complexity studies all manner of complex systems in real life. While the active agents or active elements in these complex systems can be quite sophisticated in themselves, they are typically simulated with simple models. For example, to study the flow of pedestrians in different environments, great results have been achieved by simulating pedestrians with only a few behavioral rules and goals (Ball, 2004, pp. 131–147). In this book, we take a similar approach to games. Although it is possible to create emergent games with a few complex elements, we are more interested in the mechanics of game systems that work with simple parts but still create emergent gameplay. The advantage of our approach is that, in the end, these games are efficient to build, even if they are initially more difficult to understand.

Probability Space

In the previous chapter, we mentioned that games are often regarded as state machines: hypothetical machines that progress from one state to another based on their current state and the input provided by players. In games, the number of states can grow very fast, and yet not every state is possible. Not every random placement of pieces on a chess board represents a game state that can be reached through actual play. For example, it is not possible to have pawns in your color on the row closest to you in a real game or to have both your bishops on a square of the same color. When the number of possible states is very large, game scholars refer to them collectively as a *probability space*. The probability space represents all the possible states that can be reached from the current state. The probability space can be described as having a *wide* or a *deep* shape. When the shape of the space is wide, there are many different states that can be reached from the current state: Usually this means that players have many options. If the shape is deep, there are many different states that can be reached after many subsequent choices.

C. E. Shannon, in his early paper "Programming a Computer for Playing Chess," estimated that there are more possible game states in games like chess and Go than there are atoms on earth (1950). The rules of the game determine the number of possible states, but it is not necessarily true that more rules will lead to more possible states. In addition, when a game can create a large number of possible states without using many rules, the game will be more accessible to players.

Gameplay and Game States

When we speak of the path players take through the possible states of a game—its probability space—we sometimes describe this path as a *trajectory.* The possible game states and play trajectories through a game are emergent properties of the game rule system. Games that allow many different, interesting trajectories arguably have more gameplay than games that generate fewer trajectories or less interesting ones. However, determining the type and quality of the gameplay is hard, if not impossible, by simply looking at the rules. Comparing the rules of tic-tac-toe and *Connect Four* serves as a good illustration of these difficulties. The rules for tic-tac-toe are as follows:

1. The game is played on a three-by-three grid.
2. The players take turns to occupy a square.
3. A square can be occupied only once.

4. The first player to occupy three squares in a row (orthogonally or diagonally) wins.

The rules for *Connect Four* are as follows (with the differences emphasized):

1. The game is played on a *seven-by-six* grid.
2. The players take turns to occupy a square.
3. A square can be occupied only once.
4. *Only the bottom most unoccupied square in a given column can be occupied.*
5. The first player to occupy *four* squares in a row (orthogonally or diagonally) wins.

While there are only a few differences in the rules for these two games, the differences in gameplay are immense, much greater than the amount of mental effort needed to understand the rules. In the commercially available version of *Connect Four*, the most complicated rule (number 4) is enforced by gravity: A player's token will automatically fall to the lowest available space in the upright playing area (see **Figure 2.1**). This relieves players from manually enforcing this rule and allows them to focus on the rule's effects instead. Despite the small difference in the complexity of the rules, tic-tac-toe is suited only for small children, whereas *Connect Four* can also be enjoyed by adults. The latter game allows many different strategies, and it takes much longer to master the game. When two experienced players play the game, it will be an exciting match, instead of a certain draw as is the case with tic-tac-toe. It is hard to explain these differences just by looking at the differences in the rules.

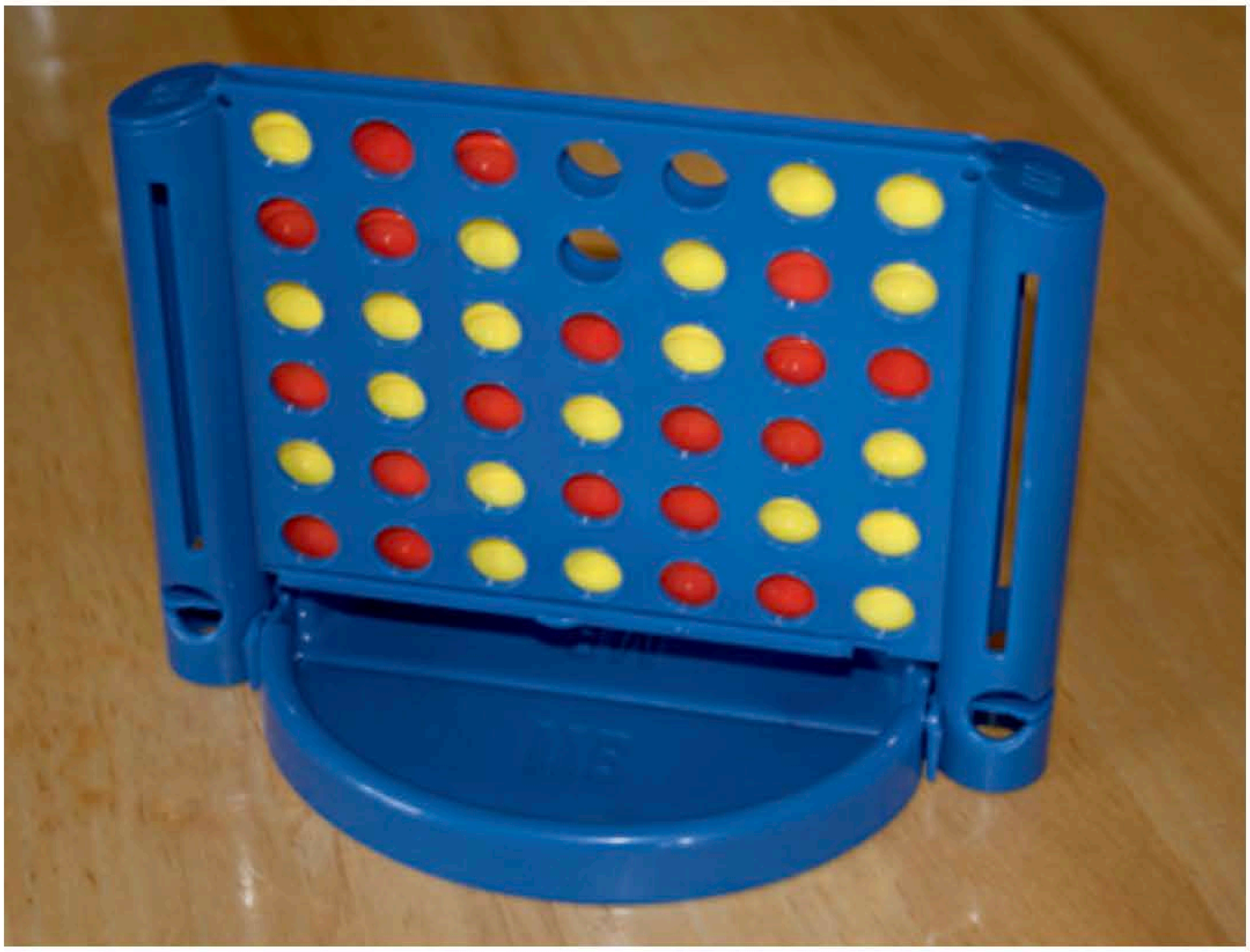

Figure 2.1. In *Connect Four*, gravity makes sure players can occupy only the bottom most unoccupied square in each column. (Image by permission of Wikimedia Commons contributor Popperipopp under a Creative Commons 3.0 license.)

Example: *Civilization*

Sid Meier's *Civilization* is a good example of a game of emergence. In *Civilization,* you lead a civilization as it evolves over roughly six millennia. During the game, you build cities, roads, farmlands, mines, and military units. You need to upgrade your cities by building temples, barracks, courthouses, stock markets, and so on. Cities produce money that you use to research new technology, to convert into luxuries to keep the population happy, or to speed up the production of units and upgrades. *Civilization* is a turn-based game set on a tile-based map, with each turn representing a number of years of your civilization's history. The choices you make determine how fast your civilization will grow, how sophisticated its technology is, and how powerful its military. Several other computer-controlled civilizations compete with you for space and resources on a finite map.

Civilization is a large game with many different game elements. However, the individual elements are surprisingly simple. The mechanics for city upgrades can easily be expressed with a few simple rules. For example, a temple costs one gold per turn and will reduce the number of unhappy citizens in a city by two. Units have simple integer values representing the number of tiles they can move and their respective offensive and defensive strengths. Some units have special capabilities. For example, settlers can be used to build new cities, and artillery can be used to bombard enemy units from a distance. Terrain modifies the capabilities of units. Mountains cost extra movement points to cross but also double a unit's defensive strength. Players can build roads to negate the extra movement costs imposed by mountains.

The Mechanics of *Civilization* Are Discrete

Inspecting *Civilization* reveals that most of its mechanics are discrete: The game is turn-based, the positions of units and locations of cities are restricted to tiles, and offensive and defensive strength is represented with whole numbers. Because the mechanics are discrete, they are easy to understand individually. You can, in principle, do all of the calculations to work out their effects in your head. Still, the probability space of *Civilization* is huge. *Civilization* is an excellent example of creating enormous variety with relatively simple discrete mechanics that invite players to interact with the game on a strategic level.

A complete description of all the mechanics of *Civilization* easily fills a book, especially if all the details of all unit types and city upgrades are listed. The game comes with its own encyclopedia to provide access to all these details. However, all these elements are easy to understand. And more importantly, there are many relations between the elements: Units are produced in cities, consuming vital resources that could have been used toward other ends. After a unit is produced, you will often have to pay gold for its upkeep every turn. Building roads also requires an investment in time and resources, but it allows you to deploy your forces more efficiently, which reduces the need to keep a large military. You can also invest in researching new technology to make sure your units are stronger than those of your opponent. In short, everything in *Civilization* is connected to almost everything else. This means that the choices you make will have many effects, sometimes unforeseen ones. Building a strong military early on allows you to capture a larger part of the map but will take a toll on other developments, which might set you back in the long term. To add to the complexity, the choices made by the civilizations surrounding yours will influence the effectiveness of your strategies.

There are many different strategies to play *Civilization*, and players often have to switch between strategies as the game progresses. Early on, it is important to capture territory so that your civilization can expand quickly. It also helps to develop technologies quickly so that you can identify and capture vital resources during this stage. Once you encounter other civilizations, you can attack them or befriend them. In the early stages of the game, it is easier to conquer other civilizations completely. Later in the game this will be much harder, and other strategies work better. When your civilization is wealthy and your neighbor is not, you can start a cultural offensive to persuade neighboring cities to join your realm. The game often progresses through a number of distinct phases: early expansion, investing in your economy, violent conflict, and eventually a technology and production race to space. The *Civilization* setup causes all these strategies and game phases to emerge quite naturally from its mechanics.

Civilization Gameplay Phases vs. Historical Periods and Golden Ages

In *Civilization*, your civilization will evolve through a number of historical periods in the game. It starts in a classical period and eventually will grow into a medieval period, renaissance, and modern period. The game uses these periods to keep the graphics and representation of your civilization in tune with your progress. The triggers that cause you to move into a new period are rather arbitrary. They don't emerge from the game mechanics as the different strategic phases of exploration, development, and conflict do. The historical periods are a fairly superficial addition that provides visual color; they aren't emergent game phases.

A golden age, a mechanism that can trigger a 20-turn span of increased production for your civilization, falls somewhere in between a gameplay phase and a historical period. The events that trigger a golden age are nearly as arbitrary as the triggers for the historical periods. However, the player does have more control over these events and can aim to trigger a golden age on purpose. Golden ages do not emerge from the gameplay but do affect the gameplay phase.

Note

We use the word *structures* to refer to the various ways that a game designer can set up game mechanics to influence or control one another. A feedback loop is a structure, for example, and so is a trigger that sets off an event when certain conditions are met.

Imagine that you are asked to design the mechanics for a game like *Civilization*.

How would you approach that task? You will probably have to design and tune the mechanics over many different iterations and prototypes. If you are clever, you keep all the elements as simple as possible, but you create several relationships among them. In that way, you can be sure that the game will be complex, but that is little guarantee that interesting gameplay will emerge. To get it right, you will need to be aware of the structure of these mechanics. Some structures will cause more emergent behavior than others. Structures like feedback loops in the game mechanics are a good way of creating emergent behaviors, especially if this feedback operates on different scales and at different speeds. Right now, this will probably sound somewhat vague. In this and later chapters, we will explore these structures and feedback in much more detail.

Games of Progression

Despite the importance of emergence in games, no professional game designer can ignore the mechanics of progression. Many games contain a story to drive the gameplay, often told over the course of many levels. Individual levels typically have clearly defined missions that set the player's goals and structure the tasks they must complete to finish the level. The designer should plan the game and its levels in such a way that the game creates a coherent experience for the player. Often this means that designers use various mechanics to control how players can move through a game. In this book, we call such mechanics *mechanics of progression*. Understanding the mechanics of progression is key to designing games with great levels and games with interesting interactive stories.

An Academic Battle

The topic of stories and games has been the subject of fierce debates between two different camps within the field of game studies. One camp, the *narratologists*, put games in the tradition of other storytelling media, and they focus on the storytelling aspects of games. The other camp, the *ludologists*, argue that to understand games you should start by looking at the game mechanics and the gameplay first and foremost. For the ludologists, game stories are not an integral part of games. *Angry Birds* is a good example. The game has a story, but the story is told only between levels, and the events within a level aren't part of it. The story and the gameplay have no effect on one another. In the case of *Angry Birds,* the ludologists are right, but there are also games that make an effort to integrate their gameplay with their stories —role-playing and adventure games, in particular. When we talk about storytelling in games, we mean integrated stories that provide more than just a superficial context for the gameplay.

The mechanics of progression are an important aspect of designing game levels. They are a key instrument for the designer to dictate what game elements players will encounter first, what resources they will start with, and what tasks they must perform to proceed. As a game designer, you decide what abilities the player has, and use the layout of a level, including the clever placement of locks, keys, and vital power-ups, to control the player's progress through the game. This way, players are eased into the game. As players explore the game's space and gain abilities and skill, they will eventually have a storylike experience that consists of the events that take place in the level, clues discovered throughout the game, or cut-scenes that are triggered at certain locations.

Tutorials

Game designers apply the mechanics of progression to create tutorials and level designs to train the player in the skills necessary to complete a game. These days, the number of rules, interface elements, and gameplay options of a modern retail video game is usually larger than most players can grasp at once. Even smaller games found on the Internet frequently require the player to learn a multitude of rules, to recognize many different objects, and to try different strategies. Exposing a player to all these at the same time can result in an overwhelming experience, and players will quickly leave the game in favor of others. The best way to deal with these problems is to design the levels in a way that teaches the player the rules in easy-to-handle chunks. In early tutorial levels, players are allowed to experiment with the gameplay options in a safe and controlled environment, where errors have few consequences.

Narrative Architecture

Using tutorials and level design to train the player illustrates one of the strengths of video games: They can use the game's simulated physical space to structure player experience. Unlike literature or cinema, which are well suited to depict events in time, games are well suited to depict space. In his paper "Game Design as Narrative Architecture" (2004), Henry Jenkins calls this type of spatial storytelling technique *narrative architecture* and places games in the tradition of spatial stories, an honor they share with traditional myths and heroes' quests as well as modern works by J.R.R. Tolkien (2004). Simply by traveling through the game space, a story is told.

Storytelling in Games

Many games have used storytelling to great effect. The *Half-Life* series stands out as a particularly good example. The games from this series are first-person shooter action games in which the player traverses a virtual world that seems to be vast but which in reality is confined to a narrow path. The whole story of *Half-Life* is told within the game. There are no cut-scenes that take the player out of the game, all dialogue is performed by characters inside the game, and the player can choose to listen or ignore them altogether. *Half-Life* has perfected the art of guiding the player through the game, creating a well-structured experience for him. The practice is often referred to as *railroading*; in this light, it is probably no coincidence that in *Half-Life* and *Half-Life 2* the player arrives on a train (see **Figure 2.2**). The disadvantage of railroading is that the player's freedom is mostly an illusion. When players go in a direction that was not intended by the game, the illusion can break down very quickly. It takes a lot of design skill to prevent players from noticing the invisible boundaries that prevent the player from exploring in other directions.

Figure 2.2. In *Half-Life 2* the player arrives in the game by train but never leaves the rails.

Creating interactive stories for games is not easy. Traditional techniques such as using branching story trees have proven inefficient. You have to create a lot of content the player will not experience in a single play-through. Creating vast open worlds for the player to explore, as is often the case in many of the

Elder Scrolls games, offers much freedom to the player but often means that the players lose track of the main storyline altogether. To create a coherent, storylike game, a delicate balance between offering players freedom and restricting their freedom through the design of your levels is required.

Example: *The Legend of Zelda*

Almost all the games and levels in the *Legend of Zelda* series are good examples of games of progression. To give a detailed example of how progression works in games, let's examine the Forest Temple level in *The Legend of Zelda: Twilight Princess*. In this level, the player, controlling the game's main character Link, sets out to rescue eight monkeys from an evil presence that has infested an old temple in the forest. The mission consists of the player freeing eight monkeys, defeating the mini-boss (the misguided monkey king Ook), and finding and mastering the "gale boomerang" before finally defeating the level boss (the Twilit Parasite Diababa). **Figure 2.3** displays the Forest Temple level map. **Figure 2.4** summarizes the player's tasks and their interrelation in a graph. To reach the goal, Link needs to confront the level boss in a final fight. To get to that fight, Link must find a key and rescue four monkeys, for which he needs the gale boomerang, for which he needs to defeat the monkey king, and so on. Some tasks can be executed in a different order: It does not really matter in what order Link liberates the monkeys. Other tasks are optional but lead to useful rewards.

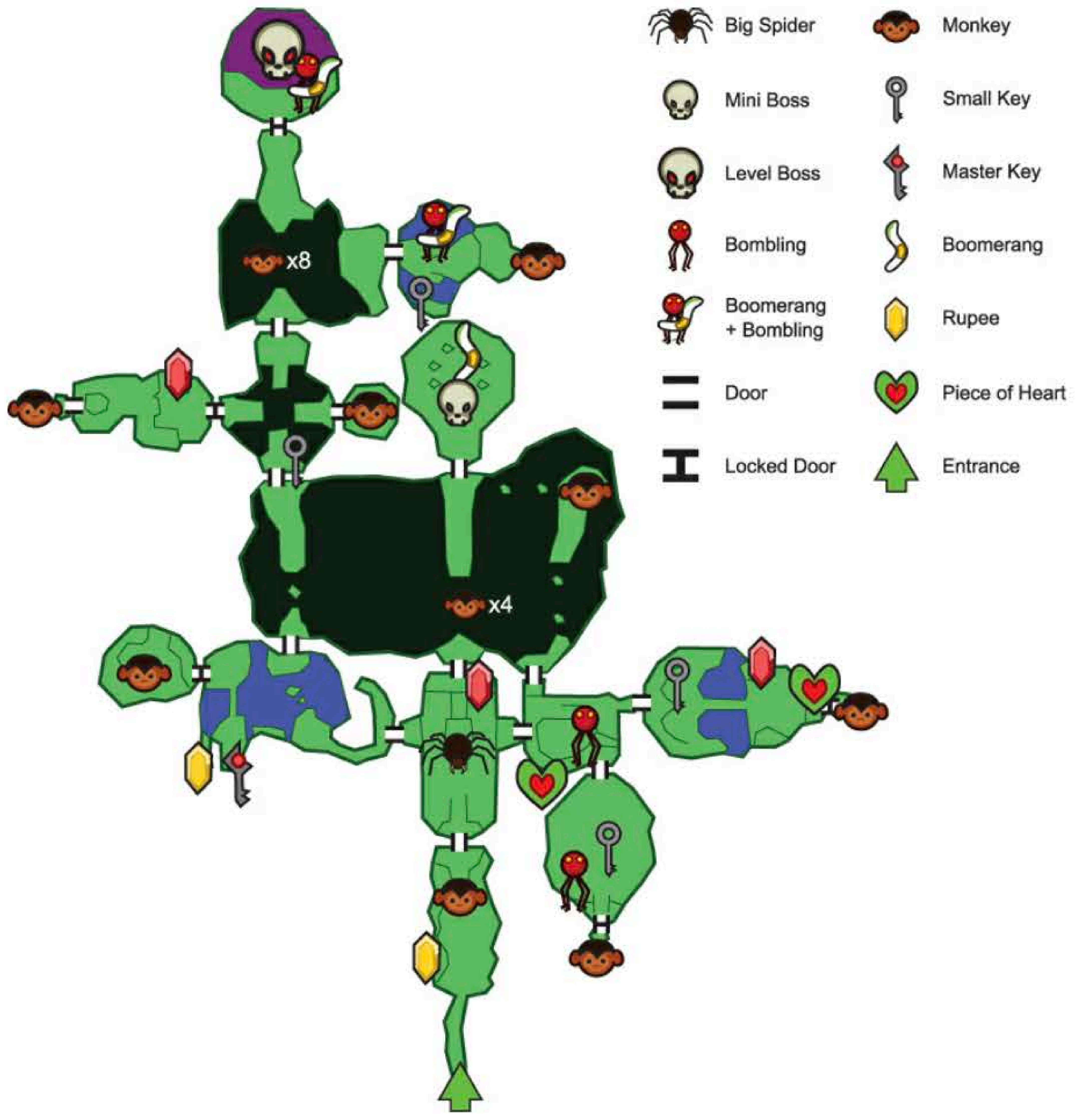

Figure 2.3. A map of the Forest Temple

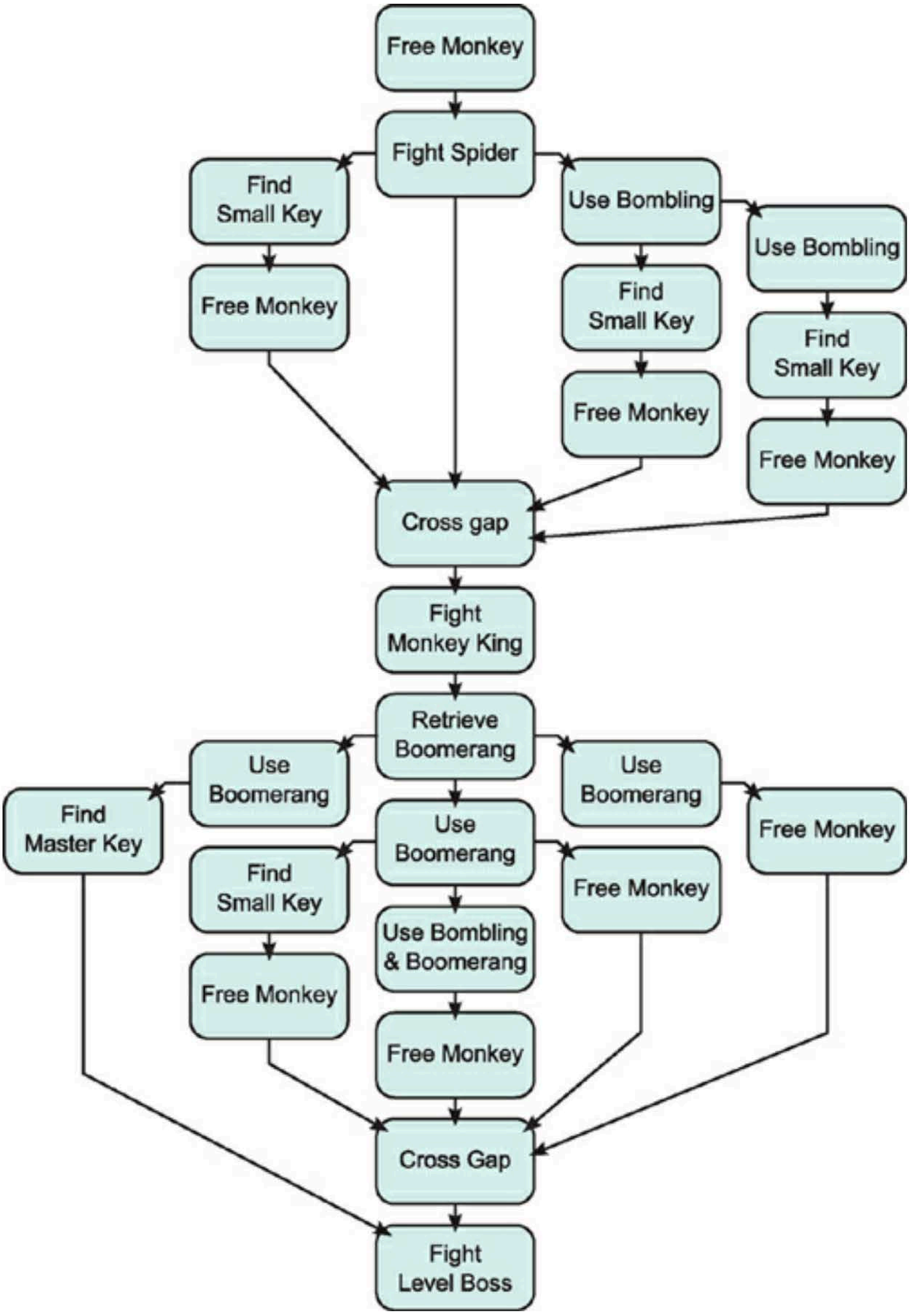

Figure 2.4. A graph of the forest temple mission

Zelda is Not a Pure Game of Progression

All games in the *Legend of Zelda* series combine mechanics of progression with emergent gameplay. Combat, for example, features a lot of emergent mechanics in which the players must learn and master many different fighting techniques and discover for themselves which strategy works best against what enemy. As already noted, pure games of progression are quite rare these days. However, the *Legend of Zelda* games do include strong mechanics of progression: They have cleverly designed levels and a long story line to structure the gameplay experience. As such, it is a prime example of progression in games.

The mission structure for the Forest Temple level has a few striking features. One is the bottleneck formed by the fight with the mini-boss and the retrieval of the boomerang halfway through the mission and the two sections of parallel options before and after the bottleneck. The game space features a hub-and-spoke layout (see the "Hub-and-Spoke Layout in Zelda Games" sidebar) that supports the parallel tasks of the mission structure. From the central hall (where the big spider is fought), the player can go in three directions. The pathway that leads to the right quickly branches into three more pathways. Three pathways lead to captured monkeys and one to the mini-boss. The last pathway is open to Link only after he has freed the first four monkeys. After the player has retrieved the gale boomerang, he can reach additional spaces in the first hub-and-spoke structure and a new hub.

Hub-and-Spoke Layout in Zelda Games

The dungeons of Zelda games are frequently arranged in a hub-and-spoke layout. One central room in the dungeon acts as a hub. From this location the player can venture into different parts of the dungeon: the spokes. Players frequently return to the hub after completing a particular task in a spoke. The advantages of a hub-and-spoke layout are that it lets players choose which tasks to complete first (and lets them choose a new one if the first one they try is too difficult), and hubs are good locations for save points or dungeon entrances. Using a hub-and-spoke layout minimizes backtracking through areas players have already seen. For a more detailed discussion on the hub-and-spoke layout technique, refer to Chapter 12 of *Fundamentals of Game Design.*

The gale boomerang itself is a good example of the lock and key mechanisms typical of the series. This is used in many action-adventure games, as Ashmore and Nietsche observed in their paper "The Quest in a Generated World" (2007). Lock and key mechanisms are one way to translate strong prerequisites in a mission into spatial constructions that enforce the relationships between tasks. The boomerang is both a weapon and a key that can be used in different ways. It has the capability to activate switches operated by wind. Link needs to operate these switches to control a few turning bridges to give him access to new areas. To get to the master key that unlocks the door to the final room with the level boss, he must use the boomerang to activate four switches in the correct order. At the same time, the boomerang can be used to collect distant objects (it has the power to pick up small items and creatures) and can be used as a weapon. This allows the designer to place elements of the second half of the mission (after the mini-boss has been defeated) in the same space that is used for the first half of the mission. This means players will initially run into obstacles they cannot overcome until they have found the right key—the boomerang.

Discrete Mechanics in Zelda Games

Zelda games mix discrete mechanics with continuous mechanics for physics. Space in Zelda is continuous, as are most physical challenges. However, a great many mechanics in Zelda are discrete. The hearts on the health bar and the damage done by Link and his enemies are discrete: A particular enemy will always cause the same damage to you, and you will be able to defeat that enemy with a constant number of hits with your sword. Likewise, the mechanics controlling progression are all discrete as well. You require a small key to unlock a door, a particular number of monkeys to help you past a gap, and so on.

Note

We do not have space to discuss the hero's journey story-pattern in more detail here, but there are many resources to guide you if you are interested in learning more about it. One of the more popular works is Christopher Vogler's book, *The Writer's Journey: Mythic Structure for Writers* (1998).

Using this particular layout and lock and key elements, the Forest Temple level in *The Legend of Zelda: Twilight Princess* is structured to generate play trajectories that feel like heroic tales. When Link enters the temple, he receives a challenge to adventure as he finds the first of eight monkeys he needs to liberate. Shortly after this, he encounters a large spider guarding the first hub in the level. Defeating this spider grants access to many locations in the first part of the level. What follows are many tests and obstacles, during which the hero Link meets new friends and enemies. Halfway through the level, Link confronts the monkey king, but there is a twist in the plot as he discovers that the monkey king is not his major adversary after all. He escapes with a magic item, the gale boomerang, which unlocks the second part of the dungeon and helps him defeat his real adversary in a final climactic battle. Just as the same structure—the hero's journey—never seems to grow stale for fairy tales and adventure films, this structure can be found in many of Link's adventures and in many other games as well.

Each enemy, trigger, or lock serves as a simple mechanism to control access to the next part of the story. Designing games of progression involves careful planning. Remember that the physical layout of your level and the location of key items inside it are your most important tools to control how players progress. You should use these elements to create a smooth and coherent experience for the players. At the same time, you must make sure that your players have had a chance to learn and practice the skills required to complete a level, but most of all make sure that they enjoy their own progress by letting them overcome obstacles that they were initially unable to defeat.

Structural Differences

To understand the difference between progression and emergence a little better, let's look at the structure of the mechanics that create the two different types of gameplay. Games of emergence are characterized by only a few rules. In a game of emergence, complexity is created by many connections and interactions *between* the rules, rather than large numbers of rules. What is interesting with this type of game is that the complexity of the gameplay leaps up after reaching a certain point in the complexity of the rules. We already saw a similar jump in gameplay complexity in the discussion of tic-tac-toe and *Connect Four*. **Figure 2.5** illustrates that turning point, which we refer to as the *complexity barrier*. Beyond a certain point, interactions among the rules create an effect that is sometimes called the *explosion of the probability space*. In general, mechanisms that contribute to emergence in games add many different possible states to the game. Large probability spaces make games more replayable; as a player, you can be confident no two games will be exactly alike. This adds to a game's appeal, especially if the outcome of each play-through is as unpredictable as the first.

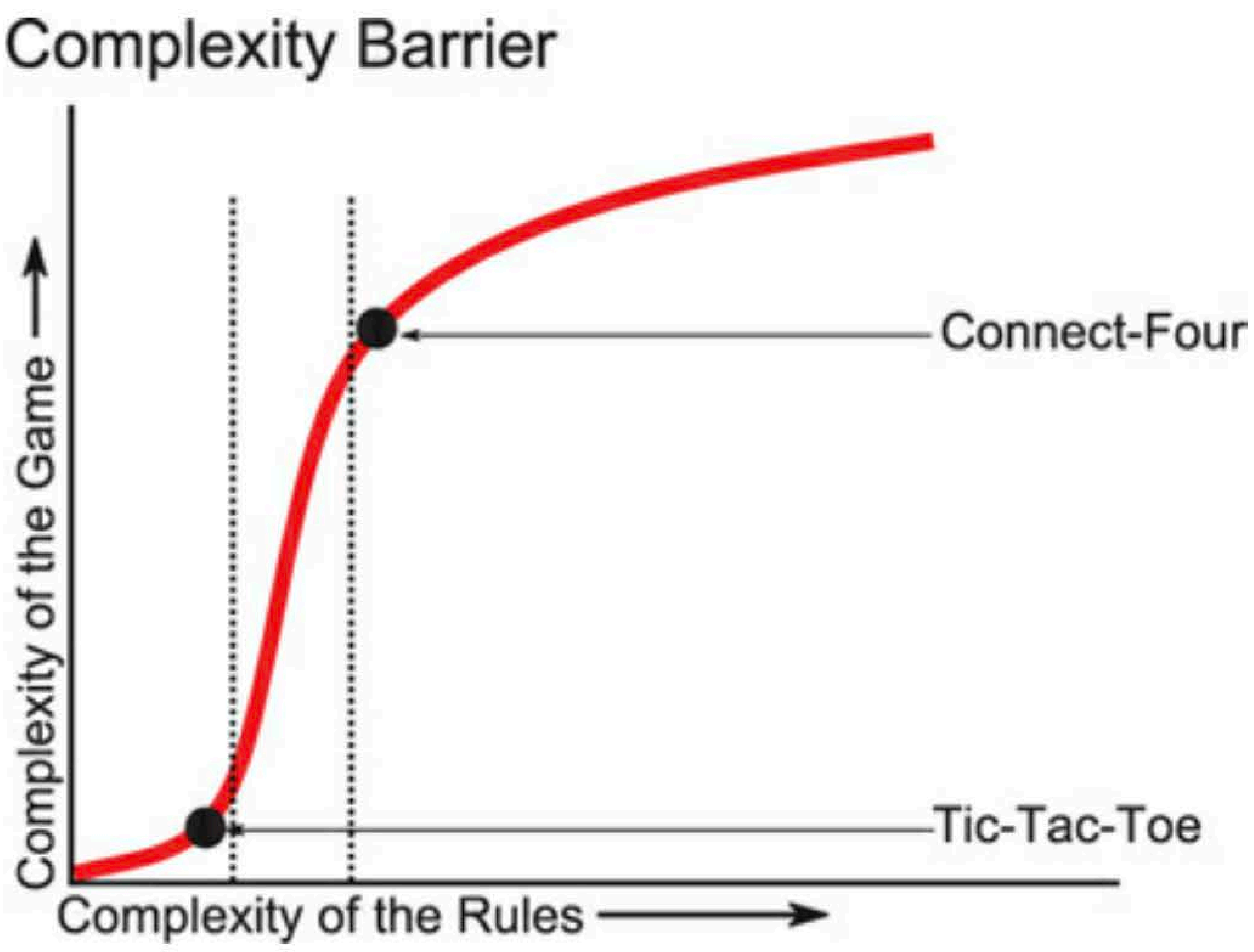

Figure 2.5. The complexity barrier is the region between the two dotted lines.

Games of progression usually possess many rules but far fewer interactions among the rules. The mechanics that control player progress through a level hardly interact with similar mechanisms in the game. Many of the mechanisms serve a single purpose: to keep players from reaching a certain place until they have accomplished some other task first. In effect, these mechanisms can be in one of two simple states: A door can be open or closed, a key can be found or not. Mechanisms that contribute to progression rarely add many different states to the game, but they are easily controlled by the game designer. The advantage of progression is that the designer can dictate the order in which players will face challenges and learn skills and can integrate constantly increasing challenges into an overarching story line. The overall experience is much easier to design in a game of progression than it is in a game of emergence.

The shape of the probability space generated by typical mechanics of emergence and mechanics of progression is quite different. Games of emergence have a probability space that is large and wide, because the game presents players with many options, and the game's direction is often subject to factors outside the player's direct control (such as die-rolling). In contrast, the probability space of games of progression tends to be small but deep. For a designer, it is easier to create a sequence of many gameplay choices—but with fewer options at each decision point—and still have a good idea of, and control over, the possible outcomes. This is why games of progression are usually longer than games of emergence and can deliver coherent stories. Games of emergence, such as checkers, tend to be shorter. In a long game of emergence, you run the risk that the player will make a small mistake early on that makes the game unwinnable hours later—a design flaw. *X-COM: UFO Defense,* although an excellent game in many respects, exhibited this property.

The mechanics of emergence are efficient at creating a large probability space. The mechanics that control progression do the opposite, restricting the probability space by limiting the number of options that players have at any one time—they cannot proceed until a particular problem is solved. As a designer, the mechanics of progression allow you to carefully structure the player's experience and deliver a well-told story. They also enable you to control the difficulty of the game so that players do not encounter challenges for which they are not yet prepared. **Table 2.1** summarizes these differences.

Table 2.1. Structural Differences Between Mechanics of Emergence and Mechanics of Progression

STRUCTURE	EMERGENCE	PROGRESSION
Number of rules	Low	High
Number of game elements	High	Low-high
Interactions among elements	High	Low
Probability space	Large, wide	Small, deep
Replay value	High	Low
Designer control of game sequence	Low	High
Length of game	Tends to be short (*Civilization* is a rare exception)	Tends to be long
Learning curve	Tends to be steep	Tends to be gentle

Emergence and Progression Integration

Although emergence and progression are considered two different ways of creating challenges in games, many games have elements of both. By integrating emergence and progression, designers strive to combine the best of both worlds: freedom and openness of play through emergence and the structured storylike experience through progression. Progression is normally used for storytelling, but it is difficult to create a coherent plot if the player has great freedom of action, as in emergent games. In practice, these generally alternate: An emergent level or mission unlocks a little story progress between levels, followed by another emergent level, and so on. The *Grand Theft Auto* games provide good examples. In those games, players may achieve victory in a mission by a wide variety of means, but their gameplay choices don't really affect the story; it occurs only between missions. So far, not many games have succeeded in integrating the two different structures so that players experience them simultaneously. There are many reasons for this:

- Video games are still a relatively young medium. No one can expect all these problems to be solved already.
- As Noah Wardrip-Fruin argues (see the "A Mismatch in the Mechanics of Games and Stories" sidebar), there is a disparity between the level of sophistication of the mechanics of progression and emergence: Mechanics of emergence have evolved much further and quicker in the past years than mechanics of progression have.
- In the past, the lack of solid formal theory of what game mechanics are and how they are structured made it difficult to approach such problems. One of the goals of this book is to present a methodological approach to designing game mechanics and to use this method to deal with these sorts of problems.

In addition, in the short history of video games there are a few interesting examples of games that have come up with ways to combine the two structures. Let's take a look at one of the more recent examples.

A Mismatch in the Mechanics of Games and Stories

In his book *Expressive Processing* (2009), Noah Wardrip-Fruin observes that the mechanics that govern a game's interactive story have not evolved as much as the mechanics to handle movement, combat, and other aspects of the game's (physics) simulation. Simulation mechanics are currently very evolved and detailed, but the player's progress through a story is simply tracked by setting up a few bottlenecks or gates to act as milestones. Once the player fulfills the task associated with a milestone, the story advances. As Wardrip-Fruin argues, the underlying shape of these story progression mechanics is not as interesting as the underlying shape of the mechanics of the rest of the game.

Example: From *StarCraft* to *StarCraft 2*

The original version of *StarCraft* is an excellent example of a game of emergence. *StarCraft* helped define the real-time strategy genre. Like *Civilization,* the individual game elements are fairly simple, but there are many interrelationships among them, setting up a system of game mechanics that has many interesting emergent properties. During the single-player campaign, you play through 30 missions, which nearly all have you build a base, manage your resources, and construct and upgrade an attack force before obliterating your opponents. The progression within a single *StarCraft* level is almost always the same, and because it is predictable, a given level does not feel very storylike.

StarCraft also tells a story *around* the levels. In many ways, it is a good example of storytelling in games, with a narrative that is more dramatic than most games of its time. In fact, its storyline follows a structure similar to a classical tragedy, which is definitely rare in games. However, the story is only a framing device around the core gameplay. The player's performance and choices have no impact on the plot, apart from the fact that the player must complete missions to advance the story. The story provides context and motivation for the game but is not an integrated part of the gameplay.

When *StarCraft 2* came out, more than a decade later, the story and its integration into the game was probably the biggest change. *StarCraft 2* changed little about the core mechanics of the original game. You can still build a base, manage resources, and construct and upgrade your force. However, the missions of the single-player campaign are much more varied than they were in the original game. For example, in the level "The Devil's Playground," the lower areas of the play field are periodically submerged in lava, destroying everything that is caught there (see **Figure 2.6**). The mission's objective is not to defeat an enemy base but to survive under these harsh conditions and harvest a number of resources in the meantime. This creates a different rhythm and progression from those of the typical missions in the original version of *StarCraft*. Another good example is the earlier mission called "The Evacuation." In this mission, it is your objective to protect a number of civilian colonists as they try to escape a planet overrun by aliens. To this end, you need to escort four caravans of civilians trying to break through to the safety of a nearby spaceport. You will build a base and an attack force but in this case, to protect the route and civilians. Again, this creates a different play experience from the typical *StarCraft* mission. In the single-player campaign of *StarCraft 2,* it is rare to find a mission that progresses through the typical stages of the missions of the original game. No longer can you simply build your base, carefully explore the map, and attack enemy bases one by one. In *StarCraft 2*, you find yourself pressed by events and scenarios that were predesigned—a classic progression mechanic. As a result, the missions are much more varied and engaging, forcing players to adapt their strategies and common patterns of play to new circumstances all the time. Because they are not repetitive, they feel more storylike.

Figure 2.6. “The Devil’s Playground” mission in *StarCraft 2*

In *StarCraft 2*, the player has more control over the larger story line of the game. To a certain extent, players can choose the order of missions and sometimes can choose between two options. Although this integrates the overall storyline into the game slightly better than the original *StarCraft* did, it is not as sophisticated as the integration between progression and emergence on the level of the individual mission.

Summary

In this chapter, you explored the categories of games of emergence and games of progression. These categories are frequently used in game studies to indicate two alternative ways of creating gameplay and challenges in games. There seems to be a tendency within game studies and among certain game designers to favor games of emergence over games of progression. This tendency can be attributed to the more interesting structure of the mechanics that create emergence in games and the size of the possibility space created by mechanics of emergence.

Games of emergence are characterized by relatively few rules, many interrelated game elements, and a large and wide possibility space. Games of progression are characterized by relatively many rules, fewer interrelation between game elements, and a smaller possibility space that is usually narrow and deep.

Modern video games include elements from both games of emergence and games of progression. However, integrating emergence and progression so that the player experiences both at once is not straightforward. It requires keen insight in the structure of the mechanics that create them. In this book, we will teach you a more structured method for looking at mechanics. In the later chapters, we will return to the integration of emergence and progressions and use these methods to shed new light on the problem of integrating them.

Exercise

Games of chess are commonly regarded to have three phases: the opening, the middle game, and the end game—and yet the rules never change throughout the game. This effect is an emergent property of the rules themselves, not an artificial construct created by mechanics of progression.

1. Find another game that progresses through different gameplay phases. (You should consider tabletop games as well as video games in your search.)
2. What causes this to happen?
3. Are the different phases truly emergent, or are they the result of a predesigned scenario or arbitrary triggers?