3
RULES

Game rules are paradoxical: Rules and enjoyment may sound like quite different things, but rules are the most consistent source of player enjoyment in games. We may associate rules with being barred from doing something we really want, but in games, we voluntarily submit to rules. Game rules are designed to be easy to learn, to work without requiring any ingenuity from the players, but they also provide challenges that *require* ingenuity to overcome. Finally, the rules of a game tend to add up to more than the sum of their parts: For most games, the strategies needed to play are more complex than the rules themselves.

Fixed rules are a core feature of games, but rules do not appear out of nowhere; they are created by players in folk games and by game designers in commercial games. Many games are played using either playing cards or computers, but the rules appear to *be the same*, even if it is the players that uphold the rules when played with cards, but the computer that upholds the rules in the video game version. As a game can move between different media, so can the rules that make up the game. But then what are rules made of and what function do game rules serve?

Let us assume that games are enjoyable in part because players *enjoy* the sense of accomplishment that solving a challenge gives them. In a multiplayer game, the enjoyment may also come from the interaction with other players, the contest or the teamwork involved in playing the game. These are not the only enjoyable aspects of games, but they are surely among the most universal ones.

In short, rules work like this.

1. *Rules* are designed to be above discussion in the sense that a specific rule should be sufficiently clear that players can agree about how to use it. Rules describe what players can and cannot do, and what should

happen in response to player actions. Rules should be implementable without any ingenuity.

2. The rules of the game construct a *state machine*, a "machine" that responds to player action (regardless of whether the game is played using computer power or not).

3. The state machine of the game can be visualized as a landscape of possibilities or a branching *game tree* of possibilities from moment to moment during the playing of the game. To play a game is to interact with the state machine and to explore the game tree.

4. Since a game has multiple outcomes, the player must expend effort trying to reach as positive an outcome as possible. It is usually harder to reach a positive outcome than a negative one—harder to win than to lose. If the player works toward the positive outcome, the player therefore faces a *challenge*.

5. The way the game is actually played when the player tries to overcome its challenges is its *gameplay*. The gameplay is an interaction between the rules and the player's attempt at playing the game as well as possible.

6. Games are learning experiences, where the player improves his or her skills at playing the game. At any given point, the player will have a specific *repertoire* of skills and methods for overcoming the challenges of the game. Part of the attraction of a good game is that it continually challenges and makes new demands on the player's repertoire.

7. Any specific game can be more or less challenging, emphasize specific types of challenges, or even serve as a pretext for a social event. This is a way in which rules can give players *enjoyable experiences*, and different games can give different experiences.

There are two extreme ways of creating challenges for players: that of *emergence* (rules combining to provide variation) and *progression* (challenges presented serially by way of special-case rules). Emergence games are the historically dominant game form. Progression games are a historically new game form where the game designer explicitly determines the possible ways in which the game can progress. Rules in games of emergence present a paradox contained in the sentence *easy to learn but difficult to master*. This is a common description of the type of game with nominally simple rules where it nevertheless requires immense amounts of effort to gain proficiency in playing the game. The apparent paradox here is that the simplicity of the rules of a game may lead to very complex gameplay. Emergence in games comes in many different forms, and it explains many interesting aspects of games such as the fact that a game can be played for hundreds of years without being exhausted; how the actual playing of a game can be unpredictable even to its designer. The element of surprise in emergent games is special in that it is an interaction between the rules of the game and the fuzzy ways in which humans understand games.

What makes an enjoyable challenge? Using a combination of Marcel Danesi's discussion of puzzles (2002) and discussions from the game development community about what Sid Meier has coined *interesting choices* (Rollings and Morris 2000, 38) I will examine what *kinds of* challenges games provide and how. Rules are not the only source of game enjoyment; I discuss the enjoyment of games as fictional worlds in chapter 4.

What Are Rules?

There is generally a clear-cut split between the fiction and rules of a game: The rules of chess govern the movement of the pieces; the representation fiction of chess is the shape and color of the pieces. No matter how the pieces are shaped, the rules, gameplay, and strategies remain identical.

What are rules? One school of thought describes rules as primarily being *limitations*. In the previous chapter, I rejected Bernard Suits's definition of games as being based on allowing the player to reach a goal by only using the *less efficient means* available (1978, 34). In Suits's view, games limit the options of a player: in high jump, using a ladder is disallowed; in a track race, the player may not run across the midfield. My objection was that it made some sense in the choice of examples, but that it is not a general feature of games. In sports, we generally have the option of finding a more efficient way of reaching the game goal. On a basic level, this is because the human body and the laws of physics exist *before* the game, but in a game, they are appropriated for the game's purposes, and some limitations are added to how they can be used. This is similar to my racing game mentioned in the preface, where the movement of the cursor on a terminal was an existing system that was used to signify the movement of a car; the game then imposed additional rules on that system, disallowing the movement of the cursor into the characters that signified the racetrack. Likewise, Katie Salen and Eric Zimmerman describe rules as *limiting player action* with the following argument: "**Rules limit player action.** The chief way that rules operate is to limit the activities of the player. If you are playing the game Yatzee, think of all the things you could do with the dice in that game: you could light them on fire, eat them, juggle them, or make jewelry out of them.... Rules are 'sets of instructions,' and following those instructions means doing what the rules require and not doing something else instead" (2004, 122). This is technically true, but the limitation view of rules only paints half the picture: you *could* make jewelry of the dice, but it would be meaningless within the Yatzee game. The rules of a game also *set up potential actions*, actions that are meaningful inside the game but meaningless outside. It is the rules of chess that allow the player to perform a checkmate—without the rules, there is no checkmate, only meaningless moving of pieces across a board. Rules specify *limitations* and *affordances*. They prohibit players from performing actions such as making jewelry out of dice, but they also add meaning to the allowed actions and this *affords* players meaningful actions that were not otherwise available; rules give games *structure*. The board game needs rules that let the players move their pieces as well as preventing them from making illegal moves; the video game needs rules that let the characters move as well as rules that prevent the character from reaching the goal immediately.

Sports and other physical games require an extra note here: Though the explicit rules of soccer only state the dimensions of the playing field, the ball's specifications, what the players can and cannot do, and the conditions for winning, the game of soccer is also governed by the laws of physics—the air resistance of the ball, gravity, the condition of the grass, and the limits of human anatomy. If we compare the physical sport of soccer with a video game version of soccer such as *FIFA 2002*, the video game adaptation requires that the laws of physics and the human anatomy be explicitly implemented in the programming *on the same level as the explicit rules of the game:* A computer-based soccer game needs to implement the physics of the players and the soccer pitch as well as the rules of the game. Gravity existed prior to the invention of soccer, and the human body existed prior to the invention of the foot race, so including them in a game is a choice that the creators of the game make. It therefore makes sense to see the laws of physics on the same level as the conventional rules in soccer: The main difference between the rules of a video game and the rules of a sport is that sports use the preexisting systems of the physical world in the game.

Strategies and State Machines

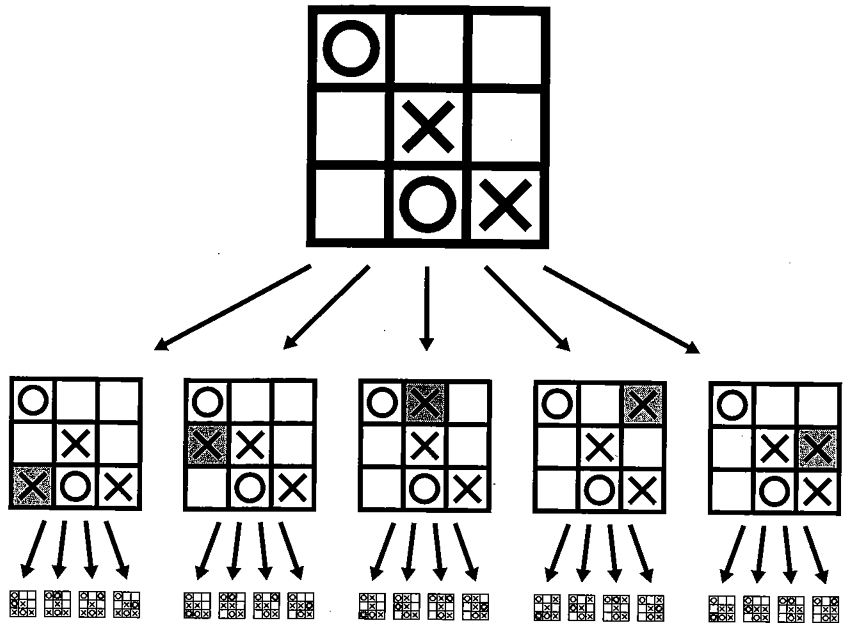
As explained in *game theory* by Neumann and Morgenstern, there is an important distinction between rules and strategies: "Finally, the *rules* of the game should not be confused with the *strategies* of the players.... Each player selects his strategy—i.e. the general principles governing his choices—freely. While any particular strategy may be good or bad—provided that these concepts can be interpreted in an exact sense—it is within the player's discretion to use or to reject it. The rules of the game, however, are absolute commands" (1953, 49). In game theory, a strategy is an overall plan for how to act in the variety of different states that the game may be in. A *complete strategy* is one that specifies unambiguously what the player should do for every possible game state. In actuality, humans tend to play games with incomplete and loosely defined strategies: A player may have a strategy that applies to only a small subset of the possible ways in which the game can be played, and will subsequently have to invent a new strategy if the game turns out differently than expected.[1] Actual strategies tend to group many possible game states into clusters in order to reduce the large number of potential game states to a manageable set of more generalized situations (Holland 1998, 41). A *dominant strategy* is one that is always better than all other strategies, regardless of the actions of any opponent.[2] A given game allows for any number of different strategies, some of which will be more effective than others. While the strategies of a game are different from the rules of the game, the relative effectiveness of a potential strategy is *a consequence* of the game rules.

Game theory also distinguishes between games of *perfect information* and games of *imperfect information:* In the former case, all players have complete knowledge of the game state at any given moment (Neumann and Morgenstern 1953, 30). In the latter case, players only have partial knowledge of the game state. Games of perfect information include many traditional board games and a few video games such as *Space Invaders* (Taito 1977), *Tetris* (Pazhitnov 1985), *ChuChu Rocket* (Sonic Team 2000), and *Tekken 3 Tag Tournament*.[3] Games of imperfect information include most card games (since the hands of the other players are hidden) and the majority of video games including all three-dimensional games (3-D graphics hide things from the player's view).

To borrow from computer science, the rules of a game provide a *state machine*. Briefly stated, a state machine is a machine that has an *initial state*, accepts a specific amount of *input events*, changes state in response to inputs using a *state transition function* (i.e., rules), and produces specific outputs using an *output function*.[4] In a literal sense, a game is a state machine: A game is a machine that can be in different states, it responds differently to the same input at different times, it contains input and output functions and definitions of what state and what input will lead to what following state (e.g., the piece can move from E2 to E4, but not to E5; if you are hit with the rocket launcher, you lose energy; if your base is taken, you have lost). When you play a game, you are interacting with the state machine that is the game. In a board game, this state is stored in the position of the pieces on the board; in sports, the game state is the score *and* the players; in computer-based games, the state is stored in memory and then represented on screen. Henceforth, I will be referring to the state of a game as the *game state*. If you cannot influence the game state in any way (as opposed to being unable to influence the game state in the *right* way), you are not playing a game.[5]

The game state only refers to the game, and not to the minds of the players. For example, in a game of perfect information, the players will not know in any detail the plans and thoughts of opponents; these are considered external to the game state. What happens in the mind of the players will be discussed later in the chapter. We can visualize the state machine of the game as a *game tree* where each game state can lead to a number of other game states, which can lead to other game states, until the game ends. Figure 3.1 is a small part of the game tree of tic-tac-toe.

It is often impractical to draw the complete game tree of a game, so the game tree is most useful as a way of understanding the large number of possibilities that a few simple rules can establish. In fact, a tic-tac-toe program I have written shows that there are 211,568 possible tic-tac-toe games—211,568 paths through the game tree of this simple game. The game tree visualizes the dynamic possibilities of a game as a map that players travel through when playing the game.



| **Figure 3.1** |
A partial game tree of tic-tac-toe.

► To see the complete listing of possible tic-tac-toe games, visit the book's Web site at http://www.half-real.net/tictactoe.

Even though it is easier to illustrate the game tree of a turn-based game, action games can also be seen as game trees, but with a much larger number of branches from moment to moment.

Algorithmic Rules

Is there anything special about game rules? Rules are limitations as well as affordances, but is there a limit to what can be a rule in a game? The rules in games are designed to be above discussion, not in the sense that it is above discussion *what* rules to use, nor in the sense that rules are never subject to disagreement, but in the sense that the *application* of a specific rule *should* be above discussion. If we think exclusively in terms of games played using computers, the question of what kinds of rules can be implemented in a computer program has already been specified in computer science with the concept of an *algorithm*. In Donald Knuth's classic computer science textbook, *The Art of Computer Programming*, he lists five important features for an algorithm:

1. **Finiteness.** An algorithm must always terminate after a finite number of steps. . . .
2. **Definiteness.** Each step of an algorithm must be precisely defined; the actions to be carried out must be rigorously and unambiguously specified for each case. . . .
3. **Input.** An algorithm has zero or more inputs . . .
4. **Output.** An algorithm has one or more outputs . . .
5. **Effectiveness.** An algorithm is also generally expected to be *effective*. This means that all of the operations to be performed in the algorithm must be sufficiently basic that they can in principle be done exactly and in a finite length of time. (Knuth 1968, 4–6)

For our purposes, *definiteness* corresponds to the description of rules as being unambiguous; *finiteness* and *effectiveness* imply that the rules of a game have to be practically usable; *input* and *output* relate to the input and output of the state machine described earlier. Knuth explains how a cookbook recipe *does not* qualify as an algorithm:

Let us try to compare the concept of an algorithm with that of a cookbook recipe: A recipe presumably has the qualities of finiteness (although it is said that a watched pot never boils), input (eggs, flour, etc.) and output (TV dinner, etc.) but notoriously lacks definiteness. There are frequent cases in which the definiteness is missing, e.g., "Add a dash of salt." A "dash" is defined as "less than $\frac{1}{8}$ teaspoon"; salt is perhaps well enough defined; but where should the salt be added (on top, side, etc.)? Instructions like "toss lightly until mixture is crumbly," "warm cognac in small saucepan," etc., are quite adequate as explanations to a trained cook, perhaps, but an algorithm must be specified to such a degree that even a computer can follow the directions. (Knuth 1968, 6)

Students of contemporary literary theory may find the demand for *definiteness* daring since it is well known that any piece of text or information can potentially be understood in any number of ways, but in actuality, algorithms can be definite because of the way they are constructed. In the recipe example, Knuth points to the fact that the recipe can be understood by a trained cook, but not by someone who has not cooked before. The recipe presupposes knowledge about the problem domain—in this case cooking. For something to be an algorithm, it has to be usable *without an understanding* of the domain. As such, what can qualify as an algorithm—and therefore what can be made a rule in a game—hinges on a *decontextualization:* an algorithm can work *because* it requires no understanding of the domain and because it only reacts to very selected aspects of the world—the state of the system; the well defined inputs; but generally *not* the weather,[6] the color of the computer case, the personality of the computer operators, or the current political climate.

This leads back to Goffman's notion of *rules of irrelevance:* playing a game involves ignoring many aspects of the current context: "any apparent interest in the aesthetic, sentimental, or monetary value of the equipment employed" (1972, 19). As such, *all game rules relate only to selected parts of the context in which they are played.* In state machine terms, this is because a game has a predefined number of *input events*—the state of the game does not change because the sky becomes overcast or because someone coughs; it only changes when someone performs a permissible move: *Game rules relate to selected and easily measurable aspects of the game context.* To rephrase Goffman's description, every game rule also has a *rule of relevance:* A rule includes a specification of what aspects of the game and game context are relevant to the rule. The rules of relevance are a place where rules and fiction meet in that learning a game also means learning to ignore the purely decorative aspects of that game. This is part of the process of *information reduction,* discussed later.

Compare two different possible versions of the checkmate rule in chess:

1. A player is in checkmate when his or her king is in a hopeless position.
2. A player is in checkmate when the king is checked [can be captured in the next move] and he or she is unable to bring the king into an unchecked position in one move.

The actual rule for checkmate is of course the second one, and it works because the aspects of the game situation that are relevant to the rule are well defined—only the positions of the pieces. Rule #1 would immediately lead to long discussions about what constituted a hopeless position; whether a position was hopeless would depend on how skilled the player was, which could then be subject to discussion and so on. Rule #1 fails to work because it does not specify what aspects of the context are relevant—in Knuth's terms, it lacks *definiteness*. Rule #2 works because it does specify what is relevant. Furthermore, even when the relevant aspects of the context are specified, rules still need to be specified in such a way that we can easily decide whether a condition is met or not. Imagine two rules in soccer:

1. The ball is out of play when it is far away.
2. The ball is out of play when it crosses the white line drawn on the grass.

Both rules specify what aspect of the game context is relevant—in this case the position of the ball—but the first one fails to specify it in sufficient detail to be of any use. Again, rule #1 would likely lead to much discussion because it is not easily decidable.[7] In a video game, the distinction is more likely to be between what *can* be an enforceable rule (since the computer keeps track of the game state, everything in the game is already measured) and what players find to be an *acceptable* rule (players tend to become frustrated if they cannot tell exactly what happened).

Making Rules, Changing Rules

How are the rules of a game determined? In a video game, the rules are explicitly designed by the game developers, and usually developed through play-testing. In a folk game, the rules of a game are developed, passed on, and changed by thousands or even millions of independent players. In the 1894 report "Mancala, the National Game of Africa" (Culin [1894] 1971), the American anthropologist Stewart Culin examined the diffusion and variations of the game of mancala (also known as kalaha) around Africa and the Middle East, and found considerable variations in the way the game was played. Given the historical and geographical spread of the game, this makes sense. Perhaps more surprising is that a game can undergo a considerable amount of development and variation within a confined area and time period. Jean Piaget has offered a more local and detailed description of the negotiations about the rules of a "folk" game among children, a marble game called "the square game":

> As we had occasion to verify, the rules of the Square game are not the same in four of the communes of Neuchâtel situated at two to three kilometers from each other. They are not the same in Geneva or Neuchâtel. They differ, on certain points, from one district to another, from one school to another in the same town. In addition to this, as through our collaborator's kindness we were able to establish, variations occur from one generation to another. A student of twenty assured us that in his village the game is no longer played as it was 'in his days'. These variations according to time and place are important, because children are often aware of their existence.
>
> Finally, and clearly as a result of the convergence of these local or historical currents, it will happen that one and the same game (like the Square game) played in the playground of one and the same school admits on certain points of several different rules. Children of 11 to 13 are familiar with these variants, and they generally agree before or during the game to choose a given usage to the exclusion of others. (Piaget 1976, 414–415)

This suggests that it is a common experience to discuss the variations in the rules of a game, changing them at will, and being aware of a number of different variations. Piaget documents that children are well versed in the art of discussing the rules of games, and he confirms what I postulated in chapter 2, that games generally require that the rules be agreed upon *before* the game starts.

The Joy of Arguing about Rules

The description of rules having to be defined before a game starts makes it sounds like disagreement about rules is *always* a problem, something that stands in the way of the enjoyment of playing a game. But any aspect of the enjoyment of games can potentially be placed in the background in favor of something else that was previously considered a dull obstacle, and discussing rules can in fact be enjoyable: In her article "Sex Differences in the Games Children Play," Janet Lever compared the play of some eight hundred children and concluded that the boys *enjoyed* discussing the rules of a game: "During the course of this study, boys were seen quarrelling all the time, but not once was a game terminated because of a quarrel and no game was interrupted for more than seven minutes. In the gravest debates, the final word was always, to 'repeat the play,' generally followed by a chorus of 'cheater's proof.' In fact, it seemed that the boys enjoyed the legal debates as much as they did the game itself" (1976, 482). I am not sure how far we can extend the gender-specific aspects of this research, and we must be wary of extending this description to cover all game players of all games, but it certainly points to the possibility that the emphasis of a game can be shifted so as to place the enjoyment somewhere else. Additionally, a few games exist that are in themselves about the discussion of rules—Peter Suber's game *Nomic* (1982) is a game *about* changing rules.[8] This does not challenge the main point here, that the dominant way of playing games is to agree on the rules *before* the game starts, and that arguing about rules is *usually* considered an impediment to game play.

Other Rules: Gaming, Sportsmanship, Gravity

Still, the explicit rules are not the only rules in a game. The most obvious example is the notion of sportsmanship: sports, especially, tend to have associated with them a number of ideas of how the noble player should perform. These ideas or conventions are loosely defined guidelines open to interpretation that *tend* to be followed by players. I can see three main kinds of sportsmanship:

1. *Preventing bodily harm:* Even if the rules would actually allow it, players try to prevent injury.

2. *Maintaining fairness in case of force majeure:* In tournament soccer, if one team has an injured player on the field, it is quite common for the other team to kick the ball out of the playing field, in effect pausing play, letting the injured player be treated, and giving the ball to the injured player's team. In a real-time strategy game, this corresponds to letting another player pause the game when they have to answer the phone.[9]

3. *Keeping the game interesting:* In some situations, a specific strategy may increase the player's chance of winning, but make the game a dull affair. In the game of *Counter-Strike*, violent discussion often breaks out around the issue of *camping:* This is a strategy where a player simply stays hidden most of the game waiting for unsuspecting players to walk by in order to promptly shoot them (Wright, Boria, and Breidenbach 2002). This is not explicitly disallowed by game, but players may agree not to

camp. The problem is that it is hard to describe camping in unambiguous terms—players do not move all the time, so for how long is a player allowed to stay still? Furthermore, a player that walks into a room and is immediately killed may feel that camping was involved—even if the other player entered the room just a few seconds earlier.

Sportsmanship is not strictly a rule according to my previous definition since its ambiguity makes it subject to continued discussion and social regulation.

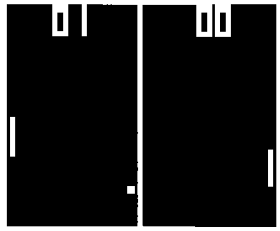
Generally, to play a game also requires competence in initiating or terminating game sessions and managing the interplay between the game and the context in which the game is played. This has been called *gaming* and *gaming rules*, and these are not rules as much as they are "rules for rules" in games (Hughes 1999, 195). This is a worthwhile study in its own right, but generally falls outside the scope of this book. Finally, many aspects of physical games are specified by preexisting systems such as the laws of physics, which are used as *objets trouvés* or *found objects*, appropriated for game purposes.

As such, the explicit game rules are not all there is to a game and games do not appear in a void. Games have an undetermined relationship to what is outside the game, and this is also part of the classic game model: the *negotiable consequences* of the game. Games often incorporate many things that are not specified in the rules.

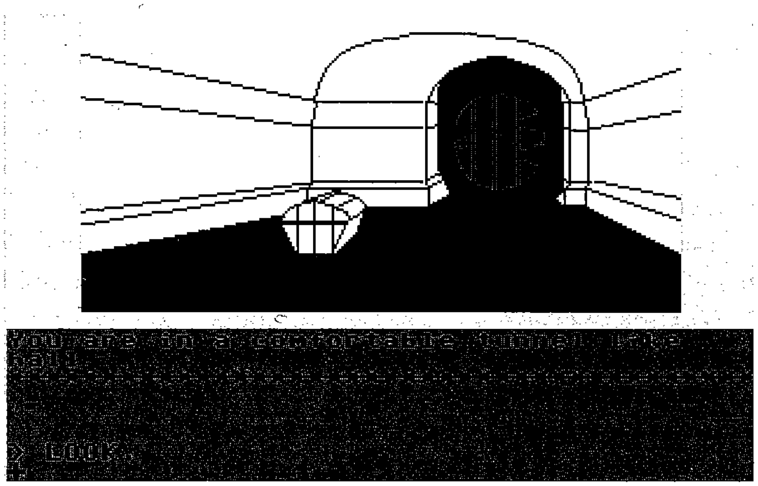
Rule Structures: Games of Emergence and Games of Progression

In the beginning of this chapter I discussed how games can present players with challenges. This can be done in several different ways, but the two most important ways are *games of progression* that directly set up each consecutive challenge in a game, and *games of emergence* that set up challenges indirectly because the rules of the game interact. To understand this, compare two old video games, the simple table-tennis game of *Pong* (figure 3.2) (Atari 1973) and the adventure game *The Hobbit* (figure 3.3) (Melbourne's House 1984), where the object of the game is to complete the travels of Bilbo as described in J. R. R. Tolkien's novel.

The Hobbit is a text-graphics hybrid adventure game where some of the world is represented in graphics and most is represented in text. The player interacts with the game exclusively through text input. To complete



| **Figure 3.2** |
Pong (Atari 1973).



| **Figure 3.3** |
The Hobbit (Melbourne House 1984).

the game, the player has to overcome a number of challenges such as getting a key from some trolls without being eaten, escaping from a goblin, sailing a river on a barrel, slaying a dragon using a helper, and finally getting back home again. On the surface of things, *The Hobbit* is the more complex game, having large amounts of graphics, dialogue, and a quite varied setting. *Pong*, on the other hand is as simple as can possibly be. Katie Salen and Eric Zimmerman have discussed how it can be difficult to understand why *Pong* became a popular game:

People love pong.
They do. But why?
Really. What's to love? There isn't much to the game; a pair of paddles moves to blunt white lines on either side of a black screen, a blocky excuse for a ball bounces between them, and if you miss the ball, your opponent scores a point. The first player to score fifteen points win. (Salen and Zimmerman 2004, 13)

They then list six reasons for playing *Pong*, the second one being the following: "*Every game is unique*. Because the ball can travel anywhere on the screen, Pong is an open-ended game with endless possibilities. Pong rewards dedicated play: it is easy to learn, but difficult to master" (Salen and Zimmerman 2004, 15). Therefore, *Pong* gives us a very rudimentary example of how a game with very simple rules can provide variation and replayability. *The Hobbit*, on the other hand, contains a wider range of possible actions that the player can perform (picking up things, talking to people, using objects to manipulate the world). Even so, the complete solution to *The Hobbit* fits on a sheet of paper (figure 3.4).

When actually playing the game, the player will at first fail to find the sequence of commands needed to complete the game. Having completed *The Hobbit*, he or she finds little reason to play the game again; the possibilities of the game are exhausted once it has been completed. This is not to cast a value judgment on the two games, but simply to point out the difference in the economy of their rules. A shorthand description of the rules of *Pong* is as follows:

Pong: Players control a bat each (at the left and right side of the screen) using a paddle. A ball is served by the computer; it bounces off two lines at the top and bottom of the screen and off the player bats. Players can direct the direction of the ball by hitting the ball with different parts of their bats. The ball accelerates until a player fails to block the ball with his or her bat, whereupon the other player scores a point. The first player to gain 15 points wins.[10]

Pong has very few rules, yet it provides the players with a large possibility space. A shorthand version of the rules of *The Hobbit* would follow the general structure of the walkthrough above, first describing where Bilbo starts, where the different players are, and what the conditions are for the solution of each task in the game. In fact, the rules of *The Hobbit* are quite

(You start in your Hobbit Hole, a "comfortable tunnel like hall")

- READ MAP
- EAST
- EAST
- NORTH
- WAIT (until day dawns)
- SOUTH
- GET KEY
- NORTH
- UNLOCK DOOR
- OPEN DOOR
- NORTH
- GET ROPE AND SWORD
- SOUTH
- SOUTH
- GET MAP
- SOUTH EAST
- GIVE MAP TO ELROND
- SAY TO ELROND "HELLO"
- SAY TO ELROND "READ MAP"
- WAIT (until Elrond gives you some lunch)
- EAT LUNCH
- SAY TO ELROND "GIVE ME MAP"
- EAST
- SOUTH
- EAST
- NORTH
- NORTHWEST
- NORTH
- SOUTHEAST
- DOWN
- DOWN
- DOWN
- DOWN
- EAST
- GET GOLDEN KEY
- UP
- NORTH
- WEST
- SOUTH
- EAST
- NORTH
- WAIT (you should be by a crack in a wall, WAIT until it opens and you get captured and thrown in the goblin dungeon!)
- DIG
- SMASH TRAP DOOR (keep doing it until it breaks, there is a CURIOUS KEY underneath. Thorin will take the Key)
- SAY TO THORIN "OPEN WINDOW"
- SAY TO THORIN "PICK ME UP"
- SAY TO THORIN "WEST"
- SOUTHWEST
- WAIT (until a goblin appears)
- NORTH
- SOUTHEAST
- EAST
- GET RING
- NORTH
- SOUTH
- NORTHWEST
- EAST
- OPEN DOOR
- UP
- CLOSE DOOR
- EAST
- EAST
- OPEN CURTAIN
- OPEN CUPBOARD
- GET FOOD
- EAT FOOD
- NORTHEAST
- EAST
- EAST
- LOOK ACROSS RIVER (you should see a boat)
- THROW ROPE ACROSS RIVER (may need to try more than once)
- PULL ROPE
- SAY TO THORIN "CLIMB INTO BOAT"
- CLIMB OUT
- EAST
- SMASH WEB (until it breaks)
- NORTHEAST
- SMASH WEB
- NORTH
- WEAR RING
- EXAMINE DOOR
- WAIT (until the door opens)
- NORTHEAST
- SOUTH
- KILL BUTLER WITH SWORD
- GET RED KEY
- UNLOCK RED DOOR WITH RED KEY
- OPEN DOOR (if Thorin got captured earlier, he'll reappear now)
- OPEN BARREL
- OPEN TRAP DOOR
- GET BARREL
- THROW BARREL THROUGH TRAP DOOR
- SAY TO THORIN "JUMP ONTO BARREL"
- GET BARREL
- THROW BARREL THROUGH TRAP DOOR
- JUMP ONTO BARREL
- EAST
- PICK UP BARD
- WEST
- NORTH
- UP
- NORTH
- NORTHWEST
- NORTH
- WEST
- EAST
- NORTHWEST
- NORTH
- WAIT (keep waiting until sun shines on the rock and opens the SECRET DOOR)
- SAY TO THORIN "UNLOCK DOOR WITH CURIOUS KEY"
- DROP BARD
- EAST
- SAY TO THORIN "WEST"
- WEAR RING
- EAST
- GET TREASURE
- EAST
- WEST
- PICK UP BARD
- UP
- DROP BARD
- SAY TO BARD "GET STRONG ARROW FROM QUIVER"
- WAIT (until Smaug the Dragon shows up)
- SAY TO BARD "SHOOT THE DRAGON"
- SOUTH x 3
- DOWN
- SOUTH x 3
- WEAR RING
- WEST
- WAIT
- WAIT
- WEST
- WAIT
- WAIT
- WEST
- NORTH
- SOUTHWEST
- WEST x 4
- SOUTHWEST
- WEST
- OPEN CHEST
- PUT TREASURE IN CHEST

(You're back in your Hobbit Hole and rich, rich, rich. Congratulations!)
(Chesire 2001)

| Figure 3.4 |
Complete solution to *The Hobbit*.

similar to the preceding walkthrough. *The Hobbit* provides challenges via many rules, but even so, the possibility space of *The Hobbit* is quite small.

I have chosen these old examples to point to the dual origins of the video game. The history of video games can be seen as the product of two basic game structures, the *emergence* structure of *Pong* and the *progression* structure of adventure games. Though games of emergence are theoretically much more interesting, I emphasize that the distinction is purely descriptive. Chris Crawford has described a similar but not identical[11] normative distinction between information-rich and process-intensive games (1982, 46). Crawford argues that since the computer is a data-processing device, a game should take advantage of the computer's strengths by emphasizing processing over data storage.[12] The distinction is also present in Harvey Smith's call for systemic level design over special-case level design (2001), which will be discussed later.

Before I describe these two game types in more detail, I suggest the reader performs the *game guide test of emergence* on a number of games:

The game guide test of progression and emergence

Search for a guide to the game on the Internet. If the game guide is a walkthrough (describing step by step what to do), it is a game of progression. If the game guide is a strategy guide (describing rules of thumb for how to play), it is a game of emergence.

Many games can be found on a scale between emergence and progression, and their game guides are consequently a combination of step-by-step descriptions ("get the red key, walk north, and open the third door") and strategy guides ("in the large room, the best way of reaching the exit is to work your way around the side while keeping all enemies at a distance using the laser rifle"). There are two extremes of this scale and two primary ways of creating hybrids:

- *Pure progression games:* The traditional adventure game is the purest example of a progression game.
- *Pure emergence games:* The multiplayer board, card, action, or strategy games are the purest examples of emergence games.
- *Progression games with emergent components:* The single-player action game is usually a hybrid in that the player has to traverse a number of

areas each of which can be negotiated in a number of ways and are therefore emergence structures.

- *Emergence games with progression components:* Multiplayer role-playing games like *EverQuest* (Verant Interactive 1999) are hybrids where the overall game structure is emergent but contains a number of small *quests* where the player has to perform a sequence of events to complete the quest.[13]

Games of Progression

Progression is the historically newer structure that entered the computer game through the adventure genre. Most clear-cut progression games are adventure games. The first adventure game is the text-based *Adventure* (Crowther and Woods 1976). A typical start of *Adventure* looks like this (">" marks what the player types.)

Welcome to Adventure!

. . .

At End Of Road
You are standing at the end of a road before a small brick building. Around you is a forest. A small stream flows out of the building and down a gully.

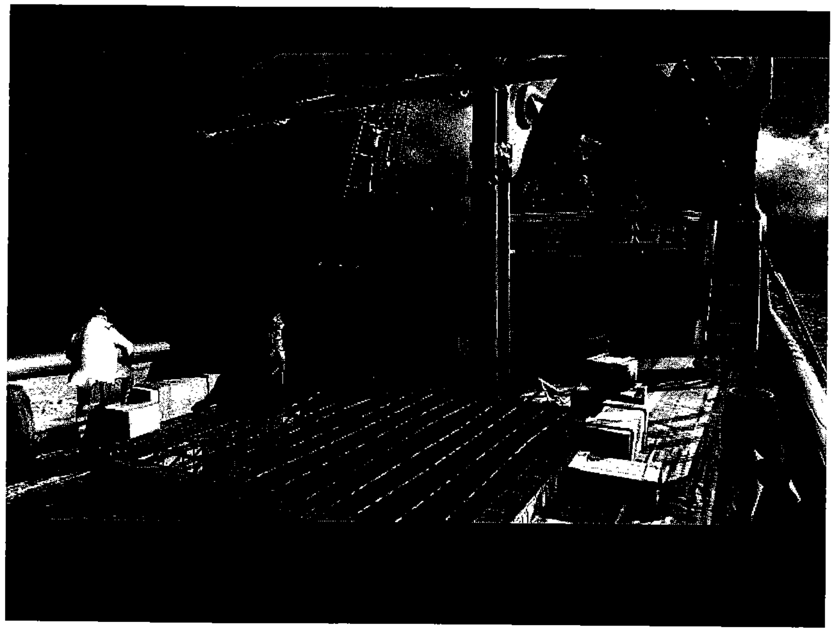
>enter building

Inside Building
You are inside a building, a well house for a large spring.
There are some keys on the ground here.
There is tasty food here.
There is a shiny brass lamp nearby.
There is an empty bottle here.

>get lamp

Taken.

The traditional adventure game was based loosely on the fantasy genre inspired by Tolkien: a world of elves, trolls, dragons, caves, and treasures. During the 1980s, the genre changed from being text-based to being primarily graphical.[14] In *The Longest Journey* (Funcom 2000) the game protagonist, April Ryan, is on board a ship threatened by a storm. To save

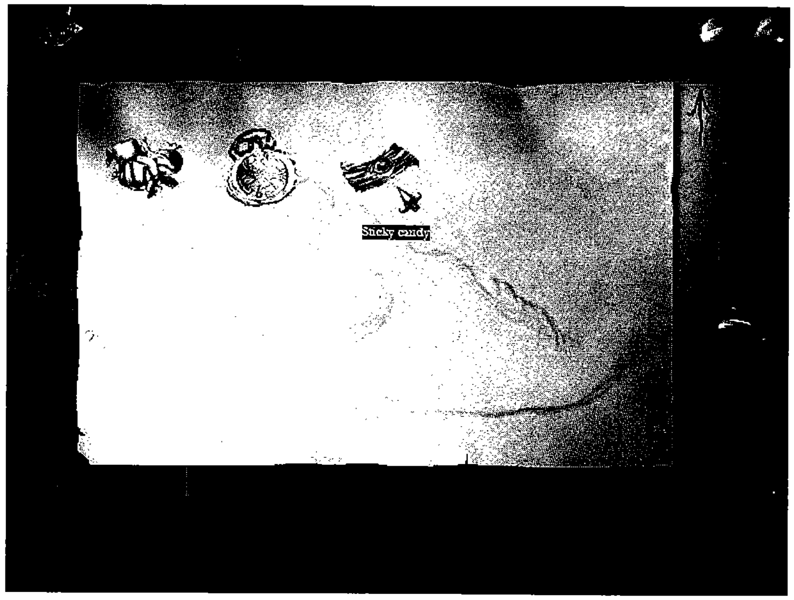


| **Figure 3.5** |
The Longest Journey (Funcom 2000): Go to the cargo bay.

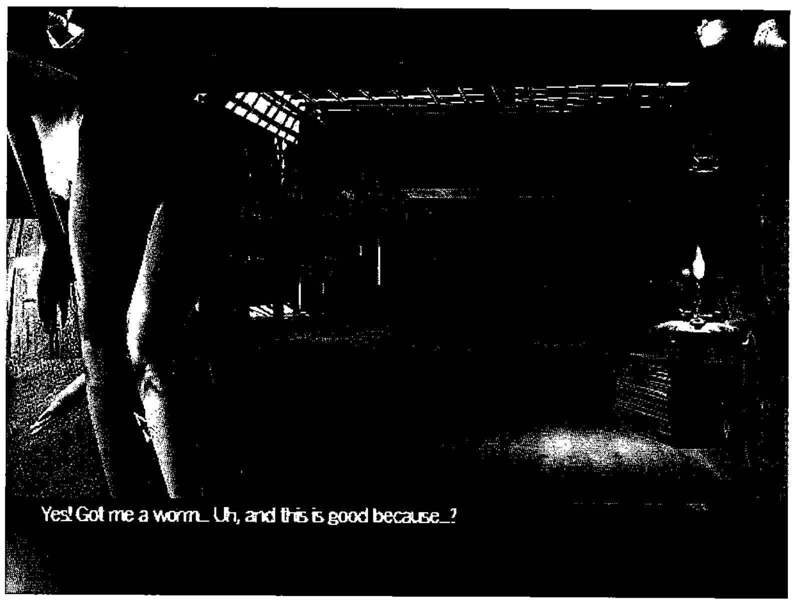
her, the player must perform a predefined sequence of events (figure 3.5–3.8). If the player does not perform the right actions, the game is over. It is characteristic of progression games that there are more ways to fail than to succeed (figure 3.9). The progression structure yields strong control to the game designer: Since the designer controls the sequence of events, this is also where we find the games with cinematic or storytelling ambitions.

Games of Emergence

Emergence is the primordial game structure where a game is specified as a small number of rules that combine and yield a large game tree, that is, a large number of game variations that the players deal with by designing strategies. Emergence is found in card and board games, most action, and all strategy games. Almost all multiplayer games are games of emergence. Games of emergence exhibit a *basic asymmetry* between the relative simplicity of the game rules and the relative complexity of the actual playing

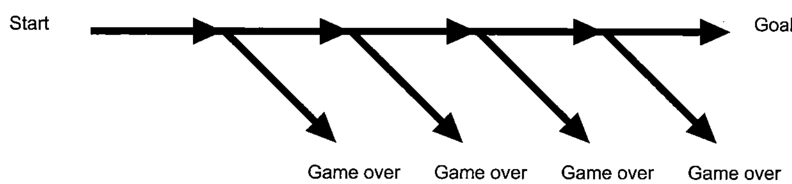


| **Figure 3.6** |
Eat some candy, leaving a candy wrapper.



| **Figure 3.7** |
Stop a hole with the wrapper and catch a worm.

| **Figure 3.8** |
A storm appears.



| **Figure 3.9** |
Progression games: to complete the game, the player has to perform exactly the actions that the game designer planned or the game ends.

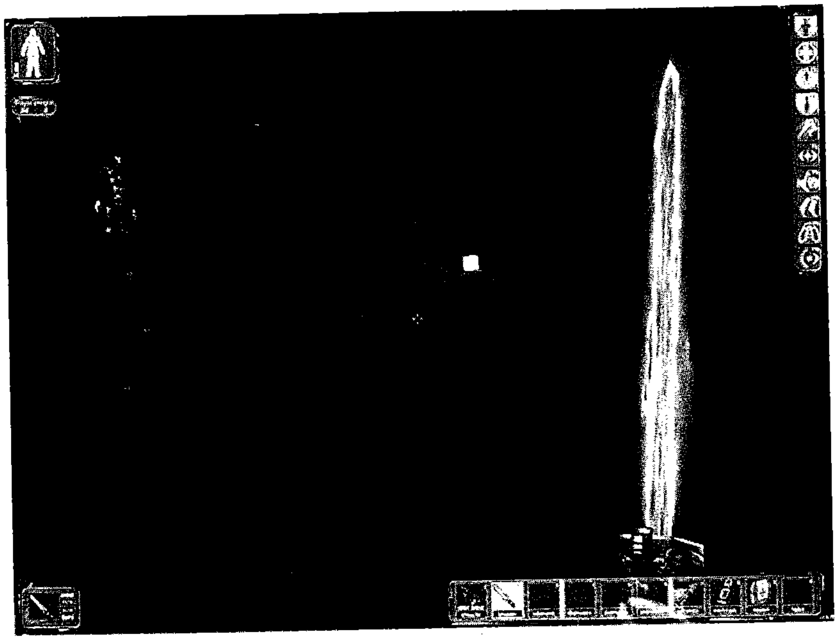
of the game. To give a non-electronic example, the rules of chess can be described on a sheet of paper, but a well stocked bookstore carries shelf after shelf of books on specific openings, gambits, endgames, and so on; there is more to playing such games than simply memorizing the rules. In a game of emergence, the game is therefore not as much a straight line as an open landscape of possibilities: In chess you win by checkmating your opponent—but there is a myriad of end positions in chess that qualify as checkmate, and each of these positions can be reached in an immense number of different ways. A game of emergence has a broadly defined goal—there are many game states that qualify as the goal—and a large number of ways to reach these states.

A terminological caveat about *emergence:* The term is commonly used very loosely, and even scientific literature on emergence is often contradictory. My goal here is not to write a treatise on the general phenomenon of emergence, but to understand game rules. At a most basic level, the question is whether emergence is a feature of the game systems themselves or a feature of human cognition. There are good arguments for both positions, and I will therefore borrow from a number of different descriptions of emergence in order to distinguish between different types of emergence in games.

Emergence in games has recently received much attention under the heading of *emergent gameplay*. Emergent gameplay is usually taken to be situations where a game is played in a way that the game designer did not predict. The game designer Harvey Smith has argued extensively for *systemic level design;* game design that allows for emergent gameplay. He makes the distinction between *desirable* emergence, where the interaction between the different elements of the game leads to interesting gameplay, and *undesirable* emergence, where players find ways to exploit the rules in ways that make the game less enjoyable. The best-known example of the latter is the *proximity mine* problem in *Deus Ex* (Ion Storm 2000), illustrated in figure 3.10:

Some clever players figured out that they could attach a proximity mine to the wall and hop up onto it (because it was physically solid and therefore became a small ledge, essentially). So then these players would attach a second mine a bit higher, hop up onto the prox[imity] mine, reach back and remove the first proximity mine, replace it higher on the wall, hop up one step higher, and then repeat, thus climbing any wall in the game, escaping our carefully predefined boundaries. (Smith 2001)

Smith's distinction corresponds closely to the distinction between emergence and progression games. Harvey Smith's aesthetic argument for systemic level design is that it allows for more *self-expression* on the



| **Figure 3.10** |

Deus Ex (Ion Storm 2000): Climbing the wall using mines (from Smith 2001).

players' part; the players can solve problems the way they want to solve them rather than in the way the game designers planned. The *practical* argument is that it allows content to be created faster, which mirrors the basic asymmetry mentioned.

That rules and gameplay are asymmetrical, and that emergence games give the player freedom to play a game using different strategies, are in many ways flip sides of the same coin. This can be understood by way of what is broadly called "the sciences of complexity" (cf. Waldrop 1994), the study of systems (biological, economical, or otherwise) that exhibit an asymmetry between the simplicity of some basic rules and the complexity of the system. As Stephen Wolfram puts it: "Whenever you look at very complicated systems in physics or biology . . . you generally find that the basic components and the basic laws are quite simple; the complexity arises because you have a great many of these simple components interacting simultaneously. The complexity is actually in the organization—the myriad of possible ways that the components of the system can interact" (qtd. in Waldrop 1994, 86).

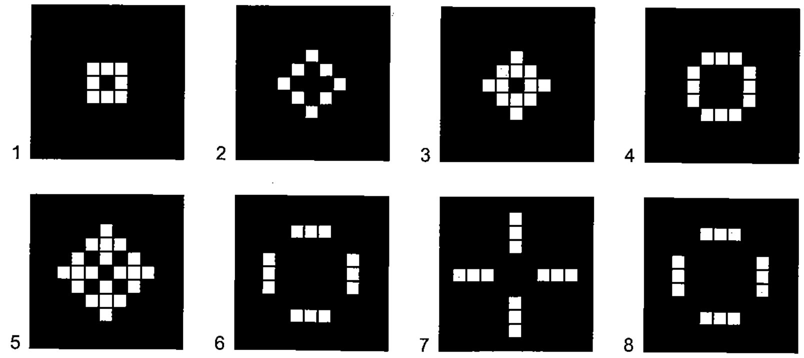
This corresponds quite well to the asymmetry between rules and gameplay in most games. What the sciences of complexity can provide is a framework for understanding how this happens. As for the term *emergence*, this is often taken to mean a higher-level pattern that is the result of interaction between many lower-level entities. Classical examples of emergence are life (life is molecules), consciousness (the result of interactions between brain cells), anthills (there is no central command in an anthill), bird flocks (there is no leader in a bird flock) (cf. Johnson 2001). As you read this, no cell in your brain is the one that is *really conscious;* your consciousness is an emergent property of the interactions between all your brain cells. In John Holland's description: "Emergence, in the sense used here, occurs only when the activities of the parts do *not* simply sum to give activity of the whole. For emergence, the whole is indeed more than the sum of its parts. To see this, let us look again at chess. We *cannot* get a representative picture of a game in progress by simply adding the values of the pieces on the board. The pieces interact to support one another and to control various parts of the board" (1998, 14).

A more concrete example of how something complex can arise from something simple is John Conway's *Game of Life* (Holland 1998, 136–142). Note that this is not a game, but an example of the emergent properties of some simple rules. Conway's *Game of Life* takes place on a grid of squares, each of which can be *on* (white) or *off* (black). The grid goes through a number of steps, in each of which the following rules are applied:

- If a square is on, it dies with less than two neighbors (from loneliness) or more than three neighbors (overcrowding).
- If a square is off, it is turned on if it has exactly three neighbors.

This may not sound very interesting, but it turns out to be a simple system that generates a large number of different patterns (figure 3.11).

► It is hard to do justice to the *Game of Life* on paper; an online version is available at http://www.half-real.net/gameoflife.



| **Figure 3.11** |

Game of Life: eight steps of a pattern generated by simple rules. It eventually cycles between steps seven and eight.

| **Figure 3.12** |

The glider: A pattern that moves across the grid in four steps.

All emergent systems are heavily *connected*. Their separate elements can all potentially influence each other in due time. The *Game of Life* has interesting properties because all its elements can interact with each other. Much effort has been dedicated to the study of the *Game of Life* and the more generalized field of *cellular automata* (for an example, see Wolfram 2002). One of the discoveries is that the *Game of Life* can support a number of regular patterns, the most famous of which is the *glider* (figure 3.12). The glider is a pattern that changes over the course of four steps, and finally reappears shifted one position on the grid (position five is identical to position one except it is shifted one position left and up). An animated version of the *Game of Life* shows the glider crawling (or gliding) across the computer screen. Patterns that are much more complex exist, such as the *glider gun*, a pattern that regularly creates new gliders.

What Emergence Can Teach Us about Games

There is some disagreement about whether emergence is a property of a system (Holland 1998, 5) or simply situations in which a game surprises its designer (Johnson 2001, 179–180; Rouse 2001, 124–125), but as we are interested in the human experience of playing games, we certainly can not afford to leave out the psychological aspects of games. The original question was how game rules provide challenges for players, and the ability of a game system to surprise players is important for games. We can distinguish between different variations of emergence in games: emergence as *variation*, as *patterns*, as *irreducibility*, and as *novelty* or surprise.

1. *Emergence as variation* is the variety of possible states and game sessions that a game's rules allow. *Pong* is an instance of *variation* coming from the interaction between some very simple rules. This is *not* emergence as surprise: It should be obvious that a large number of different games can be played by having simple rules describing, for example, the movement of a ball and some bats.

2. *Emergence as patterns:* These are patterns that players cannot immediately deduce from the rules of the game.

- All game strategies. (Since a strategy requires regularity to work, strategies require some kind of pattern in the gameplay of a game.)
- The team play required in *Counter-Strike* or the advantage of working in groups in *EverQuest*. (Specific higher-level patterns.)

3. *Emergence as irreducibility*. In his article "Guidelines for Developing Successful Games," game designer Bruce Shelley emphasizes the importance of play-testing in game development: "Prototyping is not only useful from a technology standpoint, but is also critical for testing gameplay. Designers are usually left guessing until their games can be played. There are always surprises when a game is first played, some good and some bad. Prototyping for gameplay testing is especially useful for strategy and other empty map games that do not depend on pre-planned or linear story lines" (2001).

It is striking to compare Shelley's guideline for game design to what Stephen Wolfram has written on the complexity of cellular automata: "This complexity implies limitations of principle on analyses which can be made of such systems.... The behaviour of the system can thus be found effectively only by explicit simulation. No computational short cut is possible. The system must be considered 'computationally irreducible'" (1988). Most games are irreducible; there is no shortcut to actually playing the game. And the reason why Shelley puts less emphasis on play-testing for pre-planned or linear (progression) games is, of course, that they are not emergent games because their rules and game objects have very few potential ways of being combined.

If commercial games require play-testing to develop, where does this leave "folk" (non-commercial) games? Folk games are developed over long periods of time. Since most traditional games are strongly emergent, no one can predict from the rules how they will be played. Whenever changes are made to the rules of a folk game (by deliberate design or simply by misremembering), nobody can deduce whether this will lead to good game sessions is by playing the game. What happens then is that rule changes that lead to interesting game sessions survive by word of mouth, but the rule changes that lead to boring game sessions die out.

4. *Emergence as novelty or surprise:* This is in its simplest form when several rules or objects in a game are combined in a hitherto unseen way and surprise a human player or designer.

- In the game of *Quake III Arena*, this includes *rocket-jumping*, which is the tactic of jumping into the air, firing a rocket into the ground below, and flying on the shockwave of the blast. (This is a way of jumping further than you would otherwise be able.)
- Harvey Smith's example (2001) of proximity mine climbing in *Deus Ex* is also emergence as novelty.

Designing a game with many connections between different objects and rules certainly increases the likelihood that players will find unpredicted rule combinations.

Emergence and the Player

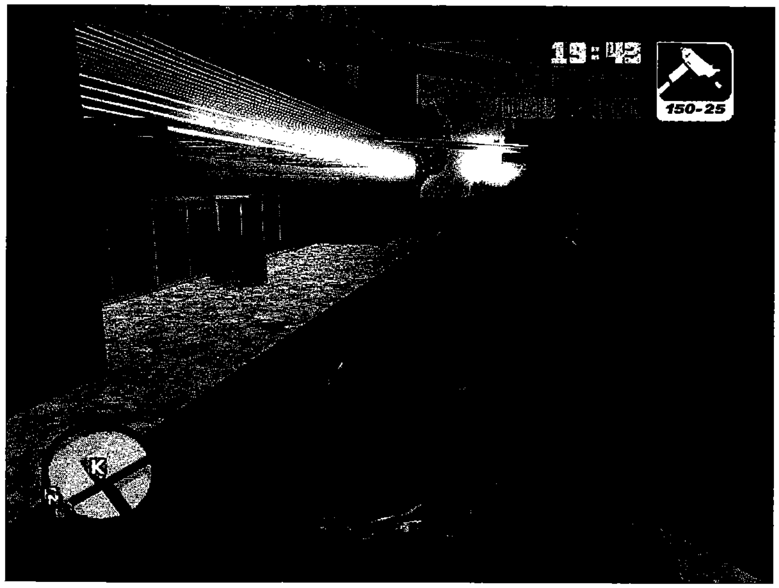
All emergent systems contain a high number of interactions between the different parts of the system. This observation provides a more precise way of differentiating between games of emergence and games of progression: In a game of emergence, a large percentage of the rules and objects in the game can potentially influence each other; in a game of progression, most rules and objects are localized. Strategy games are highly emergent and have a large degree of connectedness, since every move and unit can potentially matter to every other move and unit in the game. On the other hand, many games of exploration contain large areas that the player has to traverse, where the exact path the player follows is inconsequential—in a strategy game, the exact path of each piece always potentially matters to every other piece.

I mentioned the question of whether emergence is in the game itself or just in the mind of the player. Harvey Smith's examples (2001) describe situations where the game designer was *surprised* by what happened in the game, but he concludes that to promote emergent gameplay, games should be designed in a specific "systemic" way where the objects in the game can interact in many different ways. The experience of surprise occurs because the player and designer do not imagine the entire game tree and all possible game sessions. Emergence as novelty is therefore an interaction between the game system and human cognition. A game with many objects that interact according to well-defined rules can surprise a player in a way where the player can afterwards understand what *did* happen because the game proceeded according to clear rules.

Games between Emergence and Progression

Progression and emergence are the two extreme ways of creating games. In practice, most games fall somewhere between these poles.

To give a high-profile example, in *Grand Theft Auto III* the player is free to drive around the city and to take on the missions that various dubious characters offer. "Deal Steal" is a typical mission that takes place about one-third of the way through the game (figure 3.13–3.19). *Grand Theft Auto III* is structured in two different ways: In the big picture, the game is a linear sequence that the player has to complete, from being betrayed in the beginning of the game to finally getting revenge. There are a few optional missions and a few missions that can be completed in different order, but overall *Grand Theft Auto III* is a game of progression. Nonetheless, "Deal Steal" shows how the goals do not specify how they are to be achieved. It is up to the player to complete the mission in the way he or she wants. In diagram form, this is illustrated in figure 3.20. The advantage of structuring a game like this is that the player experiences a predefined story by completing the missions, *while* having freedom



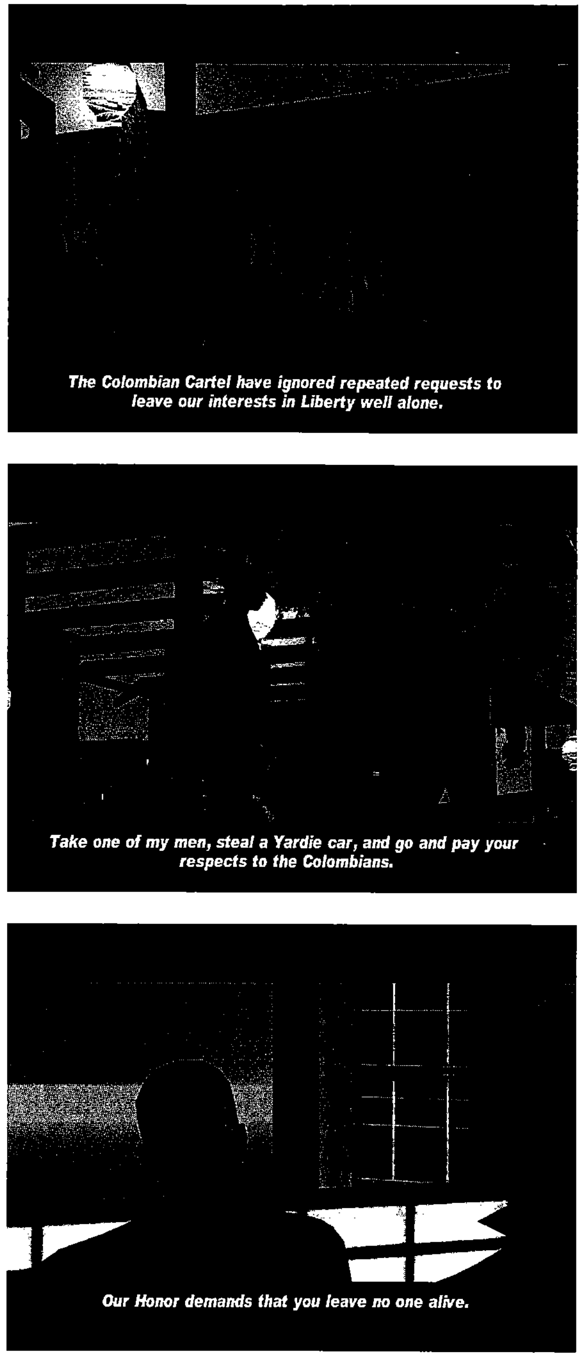
| **Figure 3.13** |
Grand Theft Auto III (Rockstar Games 2001). An indicator on the map (lower left corner) shows where to receive a mission.

to solve the tasks in different ways. Even though the player is in principle free to ignore the missions, most players will try to complete them *because they want to*, because it is more interesting to undertake the missions than not to. So even figure 3.20 does not quite express the flexibility of the game.

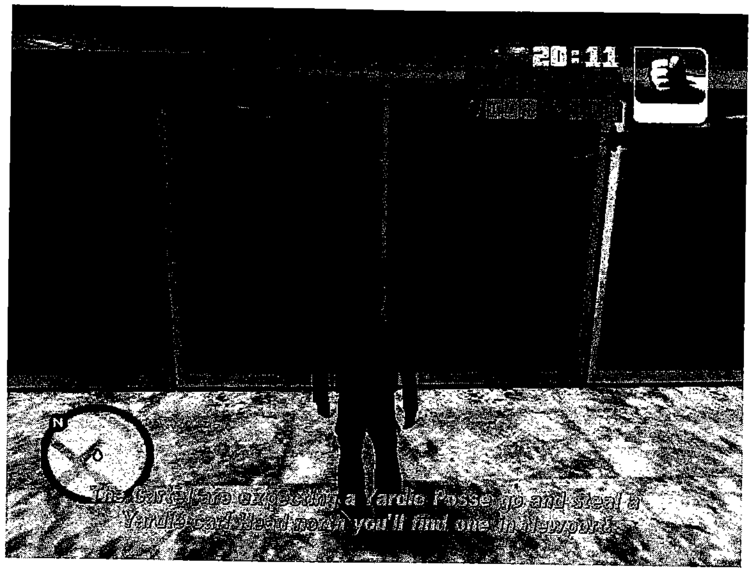
Gameplay: Rules in Action

While we can, in principle, list all the rules that govern any game and then proceed to draw the game tree of the game, this does not tell us how the game will be played. The term *gameplay* is commonly used to describe this dynamic aspect of a game. It is important to understand that the gameplay is not the rules themselves, the game tree, or the game's fiction, but the way the game is actually played. Richard Rouse's discussion of gameplay focuses primarily on gameplay as a property of the game:

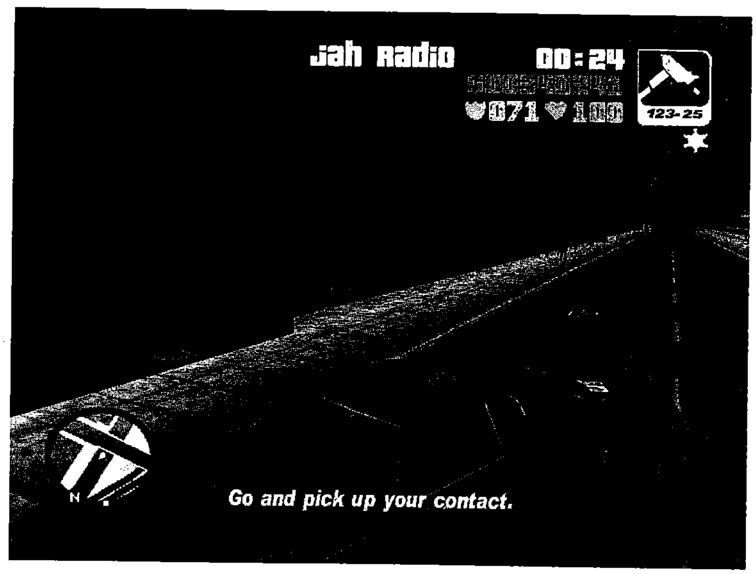
(a)
(b)
(c)



| Figure 3.14 |
Kenji explains that we should help him by getting rid of a gang.



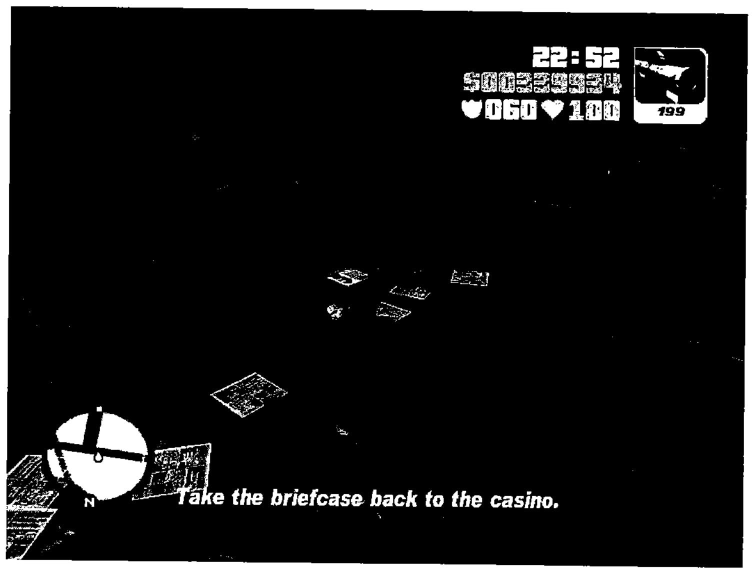
| **Figure 3.15** |
Leaving Kenji, the mission is explained in detail: The player has to steal a specific car, a yardie.



| **Figure 3.16** |
Once in the car, the map indicates where to drive to.

| **Figure 3.17** |

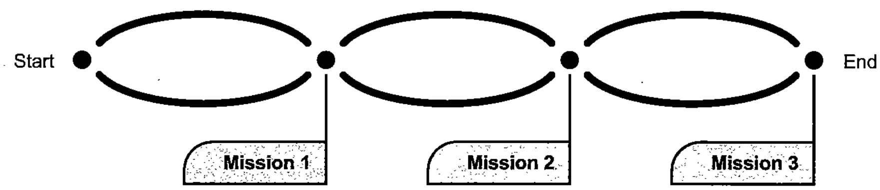
It turns out to be an ambush.



| **Figure 3.18** |

Having disposed of all the enemies, pick up a suitcase and return it to Kenji's casino.

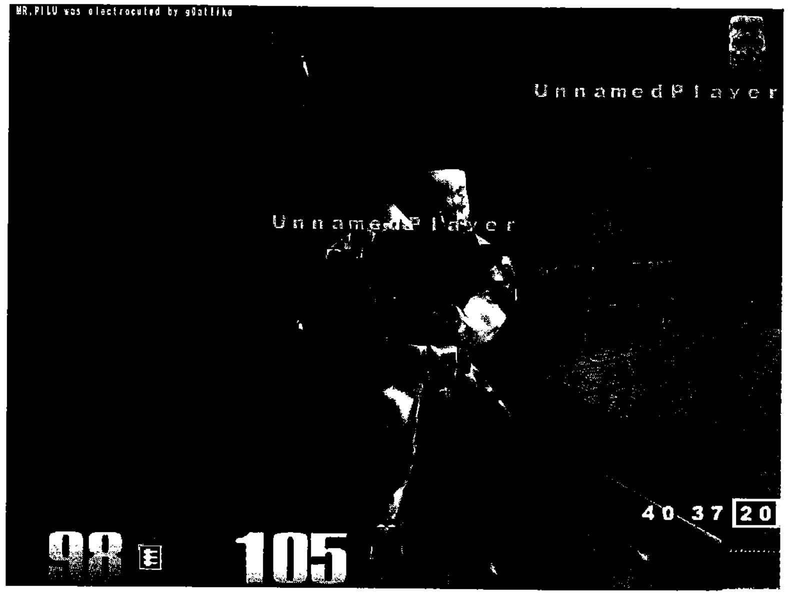
| **Figure 3.19** |
Fulfilling the mission, a $25,000 reward.



| **Figure 3.20** |
Grand Theft Auto III is a series of missions, each of which can be solved in many ways.

A game's gameplay is the degree and nature of the interactivity that the game includes, i.e., how the player is able to interact with the game-world and how that game-world reacts to the choices the player makes. (Rouse 2001, xviii)

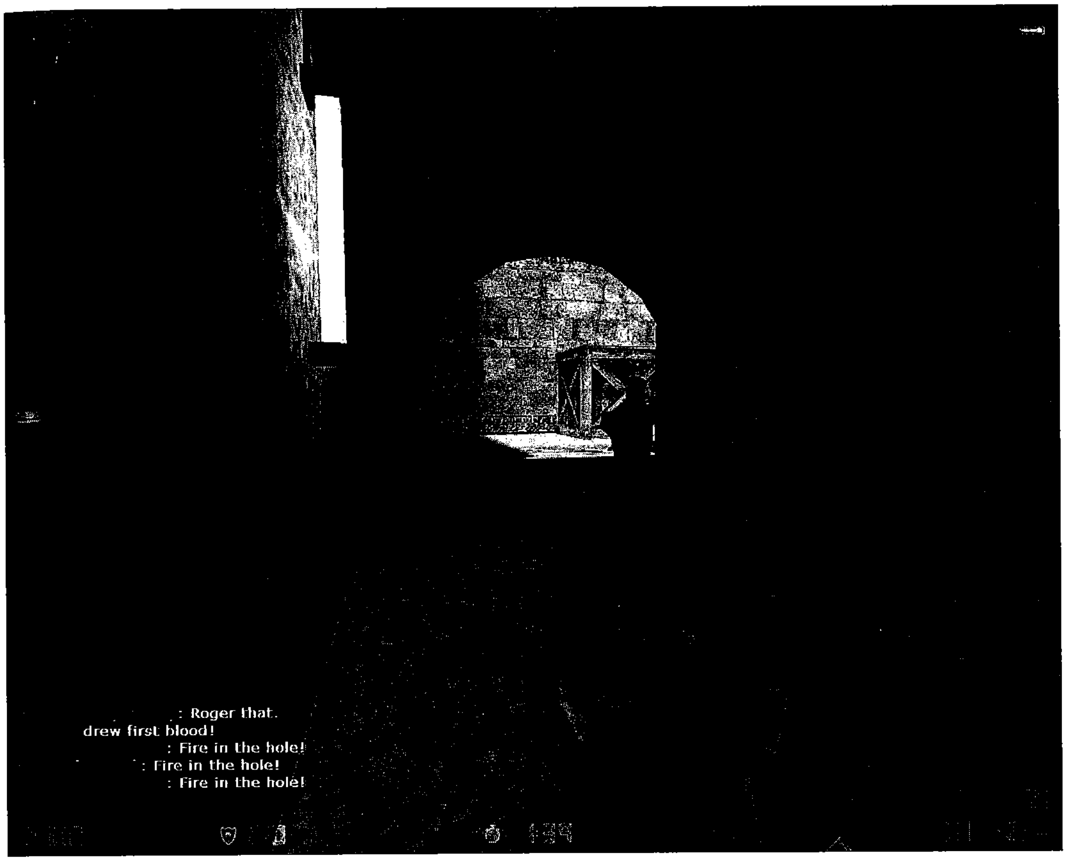
All the glitz and glitter poured into games these days, such as expensive art, animation, real actors, or the best musicians, cannot cover up for poor gameplay. (Saltzman 1999, 16)



| **Figure 3.21** |
Quake III Arena (ID Software 1999)

The Rouse quotation describes gameplay as the purely dynamic aspect of games: the interactivity, the way the player can interact with the game world, and the way in which the game world reacts. The Saltzman passage is the archetypal statement about the *importance* of gameplay—gameplay as the fun factor of games, the secret ingredient that makes them worth playing.[15]

Where does gameplay come from? I believe that gameplay is not a mirror of the rules of a game, but a consequence of the game rules and the dispositions of the game players. For example, the two games of *Quake III Arena* (figure 3.21) and *Counter-Strike* (figure 3.22) have mostly similar rules, but are considered very different gameplay experiences: "*Counter-Strike* is a game of kill or be killed. But unlike *Quake III Arena* or *Unreal Tournament*, it's not a mindless action game that involves nothing more than twitch-shooting" (Ajami and Campanaro 2001). It is a general trait of emergent systems that a small change can have big ramifications throughout the system, and this is also the case with games: *Quake III*



| **Figure 3.22** |
Counter-Strike (The Counter-Strike Team 2000)

Arena is a fast-paced first-person shooter. When players die, they respawn within a few seconds. Even when playing multiplayer team games, the game tends to be fairly individual.

Unlike *Quake III Arena*, *Counter-Strike* is famous for its team-oriented gameplay, but since there are no rules in *Counter-Strike* that tell the players to "play team-oriented," the question is, what makes *Counter-Strike* a team-oriented game? *Counter-Strike* only adds a few variations on the team-based modes of *Quake III Arena:* Players do not respawn during a round, there are goals that can win an entire round for a team, and players move more slowly and are much more vulnerable. As it turns out, these variations completely change the game to be more oriented toward team play. Since the player has only one life per round, death becomes something to be avoided at all costs. This makes it very important for players to work together. In even a simple skirmish, being in a group is much better than being alone. Having your back covered becomes important. Communication therefore becomes important. In this way, very simple rule changes can completely change the gameplay of a game. This is how *Counter-Strike* can be team-oriented even though it does not say so anywhere in its rules.

As has now been discussed, emergence allows for variation and improvisation that was not anticipated by the game designer, variation that is not easily derivable from the rules of the game. However, this does not mean that players are free to do what they like or that their behavior is devoid of patterns or regularity. Even in an emergent system, some events can still be determined or at least be very likely to happen. This can be a property of the system—some games tend to drift toward certain outcomes no matter what the players do—as well as a psychological effect. The psychological effect is straightforward: Especially in multiplayer games, players tend to accept the rules and agree to pursue the game goal. This means that players will tend to do certain things. Since players pursue the game goal, they will search for a good strategy. If the game allows for a good strategy that leads to interesting interaction, it is a good game. If the optimal strategy for playing the game leads to dull game sessions, the game will be considered uninteresting.

Games of *Counter-Strike* usually lead to skirmishes between the two teams. Neither the *Counter-Strike* instructions nor the *Counter-Strike* programming state that fights *will* take place, but they take place because the players try to win, and because winning is best achieved by subduing the other team. To give a non-electronic example, a game of *Monopoly* usually ends with a player going bankrupt. There is no rule in *Monopoly* stating, "a player will go bankrupt," but this nevertheless almost always happens as a *result* of the rules and the players' desire to win. Viewed as a game tree, Monopoly ends with a player going bankrupt because the players explore specific parts of the game tree in order to win. Gameplay therefore results from the interaction between three different things:

1. The rules of the game.
2. The player(s)' pursuit of the goal. The player seeks strategies that work due to the emergent properties of the game.
3. The player's competence and repertoire of strategies and playing methods.

As such, game design is about designing rules so that the actual strategies used by the players are enjoyable to execute.

Since gameplay is in many ways unpredictable, the actual playing of a game may reveal unanticipated dominant or plainly uninteresting strategies. When game companies issue patches, they often address not only technical bugs or incompatibilities with specific hardware, but also modify the rules of a game in order to prevent uninteresting gameplay.[16]

Gameplay and the Social

In the *Counter-Strike* example, the small modification of the rules compared to *Quake III Arena* changed the best strategies from being mostly individual to being mostly team-oriented. In the case of multiplayer games, rules are generally designed to make sure that the best strategies require interaction between the players. It would be very easy to design a multiplayer game where the optimal strategy was to avoid other players at all time. This would, however, not be a very interesting game. In the board game of Ludo/Parcheesi, the rule for capturing makes it pay off to seek out other players and interact with them (since it sends their pieces back to square one). The rule could be reversed so that the player who lands on an opponent's piece is the one that is sent back to square one. In this case, it would not pay off to seek out other players on the board, and player-to-player interaction would generally be *discouraged* by the game. *Counter-Strike* is more than just team play for a single game session: *Counter-Strike* has a large community of players who form clans, meet for tournaments to compete, and discuss strategy tips on web sites. The fact that communication and strategic planning is important for victory in *Counter-Strike* is an important incentive to build community—being part of a community will make you a better player. Similarly *EverQuest* promotes playing in groups and guilds simply because this is the best strategy (and in many cases the only working strategy) for fighting higher-level monsters.

This explains Johan Huizinga's observation that play/games can produce social groupings (1950, 3). The rules, the player's skills and the resultant gameplay can encourage community-building around a given game. The gameplay of a game is the basis for the building of player-driven communities.[17]

Enjoyable Rules: Interesting Choices and Aesthetics of Mind

If games are enjoyable, and enjoyment partially comes from the gameplay of the game, the question is then, what is quality gameplay? The most famous one-line description of game quality describes it as hinging on challenging choices: "A game is a series of interesting choices" (Sid Meier, in Rollings and Morris 2000, 38).

What is an interesting choice? Elsewhere, Sid Meier has described three criteria for interesting choices:

1. No single option should be the best.
2. The options should not be equally good.
3. The player must be able to make an informed choice. (Rouse 2001, 27–28)

In Sid Meier's description, an "interesting choice" is one that is mentally challenging (strategic rather than skill-oriented). A simple game such as rock-paper-scissors does meet the first criterion since there is no single option that is the best—scissors beat paper, paper beats rock, rock beats scissors. If we played an alternative version where the scissors were made of kryptonite and could cut through rock, all players would choose scissors and the game would cease to hold interest. Rock-paper-scissors fails the second criterion—it does not really matter which one we choose. According to the third criterion, it is not sufficient for a game to contain a choice that *is* slightly better than the other choices if the player does not understand it.

Our example of the tic-tac-toe example in chapter 2 can also be viewed in this new perspective: Tic-tac-toe ceases to be enjoyable over time. Once you figure out a complete strategy, the game ceases to provide you with any interesting choices—and tic-tac-toe remains a children's game.

Marcel Danesi's book on puzzles, *The Puzzle Instinct* (2002), provides another starting point for considering the quality of gameplay. Though puzzles are just a small subset of games, being usually considered the kind of single-solution tasks that constitutes a step in an adventure game, Danesi has some relevant opinions on the quality of a challenge. In Danesi's view,

Puzzles are pleasurable in themselves. The suspense that accompanies an attempt to find a solution to a challenging puzzle, or the anxiety that develops from not finding one right away, is a significant part of what makes the puzzle so fascinating and engaging.... The peculiar kind of pleasure that puzzles produce can be called an *aesthetics of mind*.... Poetry and music, for instance, evoke a cathartic response that imparts a sense of meaningfulness to existence. This can be called, more specifically, an *aesthetics of emotion*....

Needless to say, some puzzles are more intellectually pleasurable than others are. The *aesthetic index* of a puzzle, as it may be called, seems to be inversely proportional to the complexity of its solution or to the obviousness of the pattern, trap, or trick it hides. (Danesi 2002, 226–227)

From a game perspective, puzzles hold a unique place in that they usually conceal their own solution or even lead the player down the wrong path in the quest for a solution: "A man was watching his son pick apples, noticing that the number of apples in his basket doubled every minute and that it was full at precisely 12 noon. At what time was the basket half full?" (Danesi 2002, 29).

The reader should consider this puzzle before continuing. The solution is here.[18] This puzzle, as is common, tries to trick the reader into believing that the solution is much more complex than it is.

An emergent game will generally not present challenges that have been *designed* to be misleading, but the player may in actuality be lead to attempt to solve a challenge in a wrong way. What is shared between puzzles and most games is the mode of reasoning needed in order to play the game: Danesi distinguishes between puzzles that can be solved using straightforward *reckoning* and puzzles that require *insight thinking*. This is the difference between challenges that can be solved using a simple routine and therefore do not require any ingenuity, and the more interesting (or actual) puzzles that require some thinking "outside the box."

According to an anecdote from Danesi, John von Neumann, the co-creator of economic game theory, was present at a cocktail party where the following riddle was told:

Two children, a boy and a girl, were out riding their bikes yesterday, coming at each other from opposite directions. When they were exactly 20 miles apart, they began racing toward each other. The instant they started, a fly on the handlebar of the girl's bike also started flying toward the boy. As soon as it reached the handlebar of his bike, it turned and started back toward the girl. The fly flew back and forth in this way, from handlebar to handlebar, until the two bicycles met. Each bike moved at a constant speed of 10 miles an hour, and the swifter fly flew at a constant speed of 15 miles an hour. How much distance did the fly cover? (Danesi 2002, 33–34)

Please try to solve the riddle before proceeding.

In the story, Neumann rushed into an adjacent room, and the teller of the riddle explained the simple solution to the audience: Since the two children are 20 miles apart and bike at 10 miles per hour, they will meet in one hour. Moreover, since the fly flies at 15 miles per hour, the fly will cover 15 miles during that time. Neumann then rejoined the party and exclaimed the solution to be "15 miles!" The teller of the riddle commented that this was interesting since most mathematicians usually failed to see the simple solution and instead tried to solve the problem as the sum of an infinite series of ever-smaller numbers, to which Neumann exclaimed, "Well, that's how I solved it!" (Danesi 2002, 33–34).

We can extend Danesi's ideas to focus on the player of games and puzzles: (1) Different people have different tools available for the solving of problems. (2) For Neumann, a brilliant mathematician, the preceding riddle did not work as a riddle since he had the tools to solve it using straightforward reckoning. In general terms, *a given task will not be equally challenging to all players*. Tic-tac-toe is only challenging up to a certain point in life after which it becomes trivial. This also means that there is no guarantee that a puzzle works as a puzzle: The player may simply be too smart, and the tools for solving, for example, a mathematical puzzle can improve over time, rendering it too easy for the general public.

Improving with Practice: The Player Repertoire

If games are challenging, they are also challenging in a way that players often learn to surmount. To play a game is essentially a learning experience where the player acquires the skills needed to overcome the challenges of the game. Cognitive scientists Allen Newell and Paul S. Rosenbloom have noted that—for any task—practice *almost always* brings improvement in performance: "*Practice makes perfect.* Correcting the overstatement of a maxim: Almost, always, practice brings improvement, and more practice brings more improvement" (1981, 1). In other words, players improve their skills at playing a game over time. If a game has to be challenging to be enjoyable, it means that the game must match the current skill level of the player.

But how do players improve their playing? In a 1996 article, Hilde Haider and Peter Frensch list four types of theories of skill acquisition: "Existing theories of skill acquisition generally assume that the effects of practice on task performance are due to either (a) qualitative changes in the effective task structure... (b) an increased efficiency of performing *individual* task components... (c) an increased efficiency in performing *sequences* of task components, or (d) some combination of these mechanisms" (1996, 305). For our purposes, we need not settle on one specific theory. We can focus on the larger picture, that players improve their skills over time, and on two theories of skill acquisition that are undoubtedly not exhaustive but can tell us something about game playing. In their article, Haider and Frensch present a theory of type (a), according to which performance improvement stems from learning to process only information that is relevant to completing a task:

> We argue that people learn, over the course of practice, to separate task-relevant from task-redundant information and to limit their processing to relevant aspects of the task. Thus, the information processed early in skill acquisition may be qualitatively different from the information processed late in skill acquisition. [Performance improvement] may at least partially reflect systematic reductions in the amount of information that is processed, rather than changes in the efficiency with which task components can be performed. (Haider and Frensch 1996, 305)

This theory will especially be useful in chapter 4, for understanding how the fictional aspect of a game can lose importance over time as the player learns to ignore elements of the fictional world that are not implemented in the rules.

For a general understanding of challenges, Newell and Rosenbloom present the better-known theory of *chunking* (a theory of type (c) according to Haider and Frensch), which states that people improve at processing information (and completing tasks) by combining a number of primitive elements of the environment into high-level chunks, which are then processed faster than it would be to process every primitive element: "The performance program of the system is coded in terms of high-level chunks, with the time to process a chunk being less than that time to process its constituent chunks" (1981, 42). For example, a master chess player can memorize and understand chess positions at great speed because he or she has a large collection of chunks.

According to Newell and Rosenbloom, "The master has acquired an immense memory for chess positions, organized as a collection of chunks. His ability for immediate perception and short-term memory of chess positions depends directly on how many chunks are used to encode a position.... By implication, master players must spend an immense amount of time with the game, in order to acquire the large number of chunks; this seems to be well supported by historical data" (1981, 50).

The theory of chunking is just one of many theories on skill acquisition, but it can serve as a starting point. Let us think about games not specifically in terms of chunking, but in terms of *methods:* a game will demand a specific repertoire of methods (or skills) that the player has to master in order to overcome its challenges. Having mastered or completed a game, the player will have expanded his or her repertoire to include the repertoire demanded by the game. This is, I think, a quite overlooked aspect of playing games, that *a game changes the player that plays it.* The negotiation of the challenges of a game can be described in general terms:

- A player will, at any given time, have a repertoire of methods to use for playing a game. Improving skills at playing a game involves expanding and refining the repertoire.
- A quality game must present the player with challenges, continually letting the player develop a better repertoire for methods for playing the

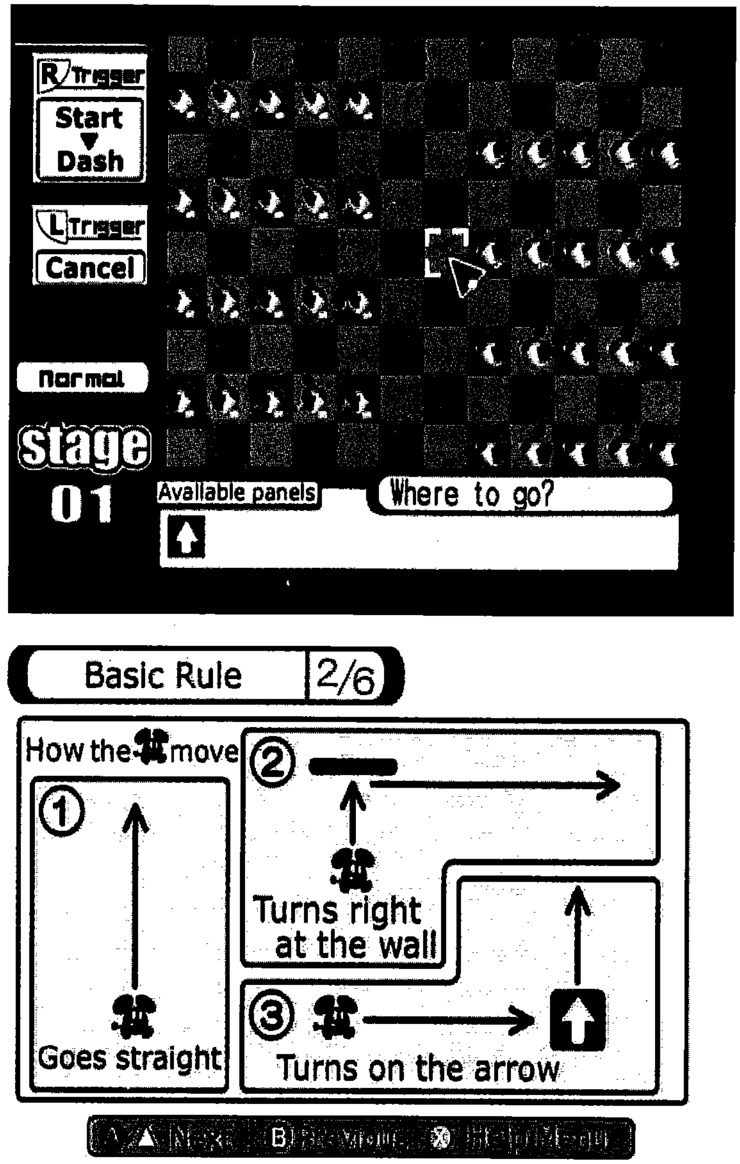
game, while continually preventing the player from playing the game just using a well defined routine.

Progression: Working with the Player's Repertoire

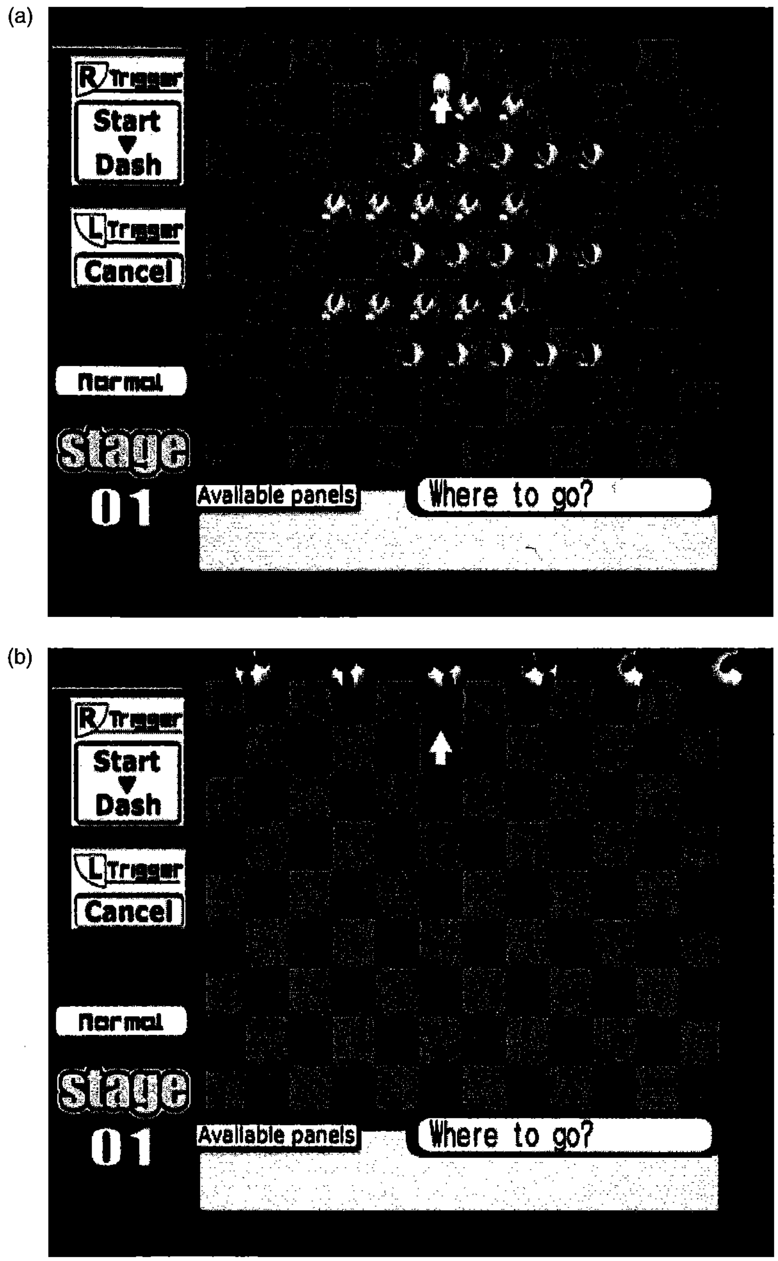
Different games provide challenges in different ways. This is apparent in the distinction between games of progression that are only completed once, and games of emergence that can be played to their conclusion many times. In the progression game, the challenges presented can be explicitly designed on a case-by-case basis, and the designer can work with the player's current expectations and current repertoire. In the emergence game, the rules of the game must keep producing new challenges by way of their design.

In the elegant puzzle game, *ChuChu Rocket*, the player must guide a number of mice to a number of spaceships. The player cannot directly control the mice, but only place a limited number of directional arrows and then the mice start. Mice change direction according to the arrows they encounter. If the player has placed the arrows correctly, the mice eventually end up in the spaceships. On some stages, the mice are threatened by cats that need to be avoided. The first stage is, as in any good game, very simple (figure 3.23). The mice are simple creatures and move according to the very simple rules stipulated here. On the first stage, the player has one "up" arrow to use, and has to bring the mice to the six spaceships on the top of the screen. This is easily done by placing the arrow anywhere in the path of the mice (figure 3.24). The game makes the player build a repertoire of methods for moving the mice. The first method is the straightforward one: Place an arrow pointing in the direction of the spaceships. Soon, however the player finds that this does not suffice: on stage 5, only an up arrow is available, but since the only spaceship is placed on the bottom row, the first method will not solve the problem (figure 3.25). It turns out that the up arrow can be used to send the mice on a major detour that will eventually bring them to the spaceship. The player's understanding of the game is challenged because the goal at this level cannot be attained using the method learned at level 1, and the player must now expand his or her repertoire to include how walls can be used to control mice indirectly. Additionally, the player may have picked up another method, the pattern in figure 3.26. This is a "staircase" pattern: sending the mice up in the lower right corner leads them to the

(a) (b)

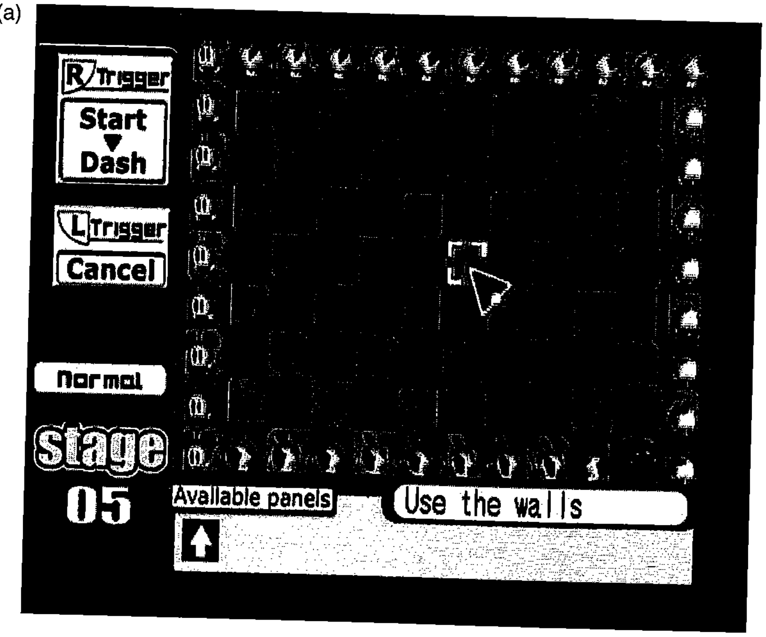


| **Figure 3.23** |
ChuChu Rocket (Sonic Team 2000). Basic rules for mouse movement.

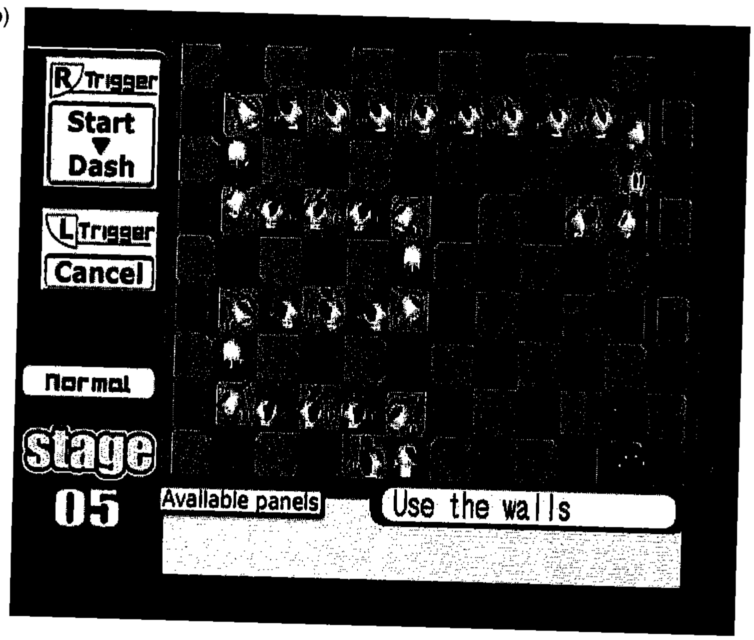


| Figure 3.24 |
The mice follow the arrow to a spaceship. The spaceships take off.

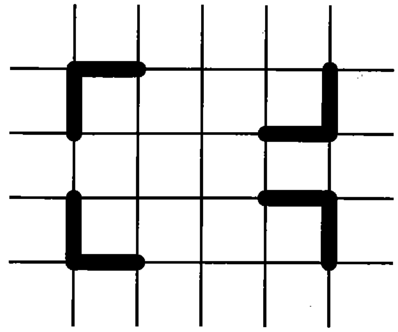
(a)



(b)



| Figure 3.25 |
Only an up arrow is provided, but the mouse needs to go down. However, the up arrow can be used indirectly. .



| **Figure 3.26** |
Staircase pattern in *ChuChu Rocket.*

lower left corner, upper left corner, and upper right corner. Any number of these can be stacked and mirrored; it is a good strategy to look for such patterns on the screen on the later stages. Stage 5 may therefore have expanded the player's repertoire with two additional methods. The staircase pattern is also a *chunk* as described by Newel and Rosenbloom previously.

The skilled puzzle designer can thus work with the player by foregrounding a specific kind of method that then turns out not to work. For example, in stage 10 of the "special" puzzles, the puzzle is set up to divert the player from the solution (figure 3.27). Stage 10 is divided into four squares. I solved the top right one first: Here I had to set an arrow to make the mouse go to the spaceship without colliding with the cat. I then worked for a while to solve the other three challenges with success except for the bottom left square. No matter how I placed the arrow, I could not make the mouse avoid the cat. The puzzle design foregrounded a specific method of solving the problem, that of controlling the mouse. The solution turned out to be to control the cat rather than the mouse (figures 3.28–3.29). According to Danesi, a puzzle tries to conceal its own answer, and the foregrounding of the wrong solution—only controlling the mouse—is an instance of that. It helps explain the attraction of *ChuChu Rocket:* The game also works well because it has cats *and* mice; this lets the game cue the players into foregrounding either mice or cats, thereby playing with their expectations and tricking them into focusing on the wrong parts of their repertoire.