## Chapter 7

# Gameplay

Any game designer should agree that gameplay is the core of the game. Given an ideal world, designers would probably claim that gameplay should be put above all other considerations. And in a lot of cases, were it not for external pressures, these same game designers would attempt to treat the gameplay with the level of importance that it deserves. There's just one problem with this: There is no universally accepted definition of gameplay. Gameplay is an important, if nebulous, concept. Many times during discussions of games, we have heard comments such as, "This has great gameplay," followed by a detailed description of the particular aspect of the game. However, if instead you were to ask the question, "What is gameplay?", most answers would attempt to explain by example. Indeed, explanation by example can be helpful, but it requires that you infer a definition of gameplay by induction. Describing gameplay without using self-reference is similar to trying to explain the concept of red without reference to color. It is difficult to conceive, but not impossible.

There is a reason for this difficulty: The concept of gameplay is extremely difficult to define. Each designer has his or her own personal definition of gameplay, formed from exposure to many examples over the course of a career.

Gameplay is so difficult to define because there is no single entity that we can point to and say, "There! That's the gameplay." Gameplay is the result of a large number of contributing elements. The presence, or lack thereof, of gameplay can be deduced by examining a particular game for *indications* and *contraindications* of these elements. (These terms are borrowed from medical terminology: An indication is a positive sign that implies the existence of gameplay, and a contraindication is a negative sign that implies that gameplay does not exist.)

## Use of Language

In other fields, such as engineering, architecture, and mathematics, the spread of ideas is facilitated by the use of a common language. Each engineer or mathematician knows how to express ideas—even brand-new ideas—in the given language of the craft.

The vocabulary and mechanism for expressing ideas is already there, formalized and developed over many years of practical use and theoretical study. As game designers, we do not have that luxury. Although there has been talk of defining a universal frame of reference for game designers, no such lexicon has been attempted in earnest. Any attempts that have so far been made have not gained major acceptance, and there is no real coordinated effort or cooperation between alternate factions (to the best of our knowledge).

This chapter attempts to define gameplay without reference to itself or reliance on examples of itself for definition. That doesn't mean that we won't give examples, but those examples will not serve as definitions. Instead, they will be used in their traditional role to illustrate the definitions previously laid out. This will give us the beginnings of our lexicon of game design. This might or might not become a standard, but it is at least a starting point that we can use to explain our ideas in this book.

## Defining Gameplay

Although we briefly discussed (and loosely defined) gameplay in Chapter 2, "Game Concepts," we did so in terms of the player experience. To continue, we examine gameplay independent of the player experience. We examine the core concepts of gameplay, which are invariant with the player. To do this, we need to state a player-independent definition of gameplay. Sid Meier once defined gameplay as "a series of interesting choices." This is an excellent starting point and forms the basis of our definition of gameplay. We take this statement one step further with our formal definition of gameplay:

One or more causally linked series of challenges in a simulated environment.

On the surface, this does not seem that far removed from Sid Meier's original definition (although it's not quite as good of a sound bite). However, our statement is more precise and rigorous. To be fair, it's unlikely that Mr. Meier expected his original definition to be used for anything more than the off-the-cuff comment it was probably intended to be—a statement designed to challenge and spur further thinking on the subject. If this was the case, it certainly had its intended effect and served as an excellent starting point for our definition.

In the original statement, the use of the word *series* implies a number of sequential events. Although these events follow one another chronologically, there is no implication that they can be linked. For example, lightning strikes tend to come in a rapid succession of bolts, but there is no evidence to suggest that the strike order is anything other than chance. Hence, we need to define specifically that our gameplay events are linked by causality. Note that we do not say anything about whether the multiple series are required to be interlinked. In most cases, they are—for example, the multiple plot threads in an adventure game—but this is not a specific requirement.

The second half of the original definition uses the words "interesting choices." Although this is true, we feel that this is too broad of a definition. Choosing to visit the cinema, deciding what movie to watch, and thinking about whether to have caramel popcorn or salted popcorn is an example of a series of interesting choices, but it isn't an example of gameplay. So we replace this with "challenges in a simulated environment." The reason for the further restriction to a simulated environment should be self-evident: We stop playing when we quit the game.

Why are we using *challenges* in place of *choices*? Again, we feel that the word *choices* is too broad to be particularly useful. For example, we can make a decision to attempt to shoot the attacking robot, to avoid it, or to quit the game and play something else. All three of these are available choices, but only the first two are gameplay decisions. Consequently, we have chosen to use the word *challenges* because it more accurately describes the type of event that the player is subjected to.

Another example of a choice that is not directly a part of the gameplay is the prevalence of user-defined "skins" in games such as the *Quake* series and *Half-Life*. The player can choose any appearance, but it is purely a cosmetic choice and normally has little effect on gameplay (except when unscrupulous players use this to their advantage, either by deliberately choosing a skin that camouflages them too well—for example, in the extreme case, a moving, shooting crate—or by forcing all the opposing players to take on skins that make them more visible, such as pure white).

Odysseus faced many challenges on his 20-year voyage to return home to his wife, Penelope, in Homer's *Odyssey*. Gordon Freeman (and, by proxy, the player) faces many challenges on his quest to escape from the Black Mesa Research Laboratory in Valve's *Half-Life*. *Tetris* players face challenges in their attempts to attain a higher score. Even Pac-Man faces challenges in his attempts to eat all the pellets in the maze while avoiding the evil ghosts bent on his destruction.

The use of *challenges* is not perfect, but it'll do. An alternative to the use of the word *challenges* that we discussed in the past was *ordeals*, but this was found to be arguably too restrictive. Ideally, we'd like to use a word that indicates a concept somewhere between the two.

## Pure Challenges

*Pure challenges* are the archetypal form of gameplay challenges. They are not often found in the wild in this form, but they form the basis for most, if not all, actual gameplay challenges. We first discuss the possible forms that pure challenges can take, and then we discuss how these can be applied to real gameplay situations.

Challenges come in many shapes and forms. Even within a genre, a good game presents a range of challenge types. The narrower the genre definition is, the narrower the range is, but this is usually not a problem. Game players who buy within genres tend to know what to expect. In fact, unless it is particularly well done and appropriate, they generally reject new forms of challenge as inappropriate to the genre in question.

An example is the inclusion of a fast-action, reflex-based arcade sequence in a traditional adventure game such as *Escape from Monkey Island* (see Figure 7.1). Handled properly, this can enhance the gameplay, giving a welcome break from the usual action. Handled badly, it can break the player's suspension of disbelief and effectively ruin the game.

**Figure 7.1** *Escape from Monkey Island*.

A more concrete example of this phenomenon is found in Valve's *Half-Life* (see Figure 7.2), an excellent game that has rightly won many awards for its original and innovative gameplay and story line. (However, I also need to point out that the story line is excellent only when compared to other games within the same genre; it wouldn't be a best-selling novel or a blockbuster movie.) For the most part, playing *Half-Life* is a joy. In the first two thirds of the game, the sense of immersion and of actually being there as Gordon Freeman is unparalleled. You can imagine yourself squeezing through the confining corridors of the Black Mesa Research Laboratory out in the middle of the desert, avoiding the unwanted attentions of both vicious aliens out for blood and hostile government troops sent in to clean up the transdimensional mess. Then, as the story reaches the first climax point, you are catapulted into the alien dimension to take the battle into their territory.

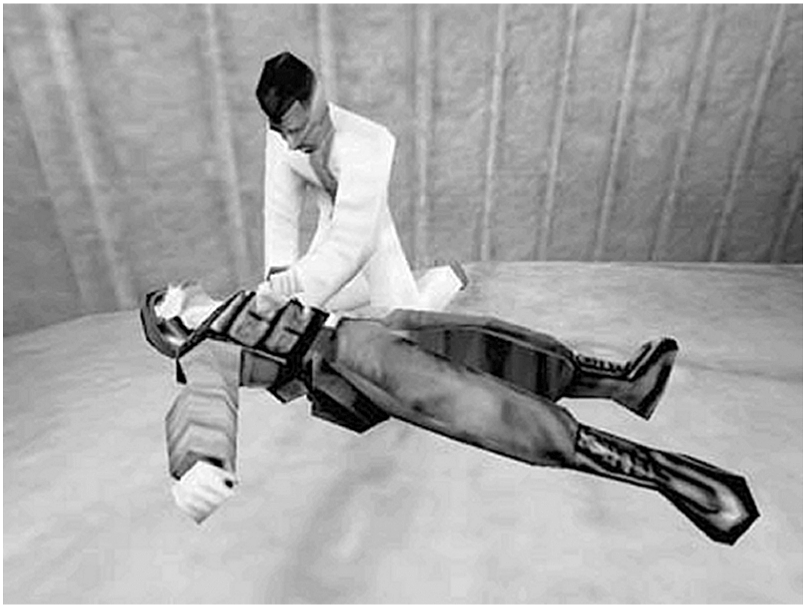

**Figure 7.2** *Half-Life*.

In the alien dimension, things take a turn for the worse—at least, in terms of gameplay. Although it's not a game breaker by any means, the story line experiences a significant lull here. Initially, Gordon is required to jump from platform to platform in a sub-Mario platform game style. This abrupt change in the gameplay is a showstopper as far as the suspension of disbelief, which the designers had worked so hard to cultivate, is concerned. And as if that didn't deal enough of a blow, the subsequent levels are practically straight out of the original *Quake*, culminating in a showdown with the big, bad, end-of-game boss. Now, we don't mean to be unduly harsh on an otherwise excellent game, but the last third of the game is a real letdown in gameplay terms. All of the innovative and exciting features of the Black Mesa levels were replaced with a standard first-person jump'n'shooter. Even with the benefit of the intriguing and imaginative end sequence, the damage is done by this point: The suspension of disbelief is shattered and the player is left feeling somehow cheated. The inclusion of the platform-based level followed by the standard first-person fare is a classic nonsequitur that affects the enjoyment of the game.

Many types of challenges can be included in a game. In the majority of cases, these challenges are purely mental. In a few games, there is some degree of physical challenge, but this is usually understated—a simple test of reflexes or hand-eye coordination. In any case, they are localized to the hands and wrists.

**NOTE**
A high-profile exception is the recent spate of Japanese dancing games, such as *Dance Dance Revolution* (see Figure 7.3), which provide the player with a pressure-sensitive mat. The mat allows the player to dance in time to the music and dancers onscreen—an interesting gameplay innovation, but one that is hardly likely to amount to anything other than an amusing diversion.

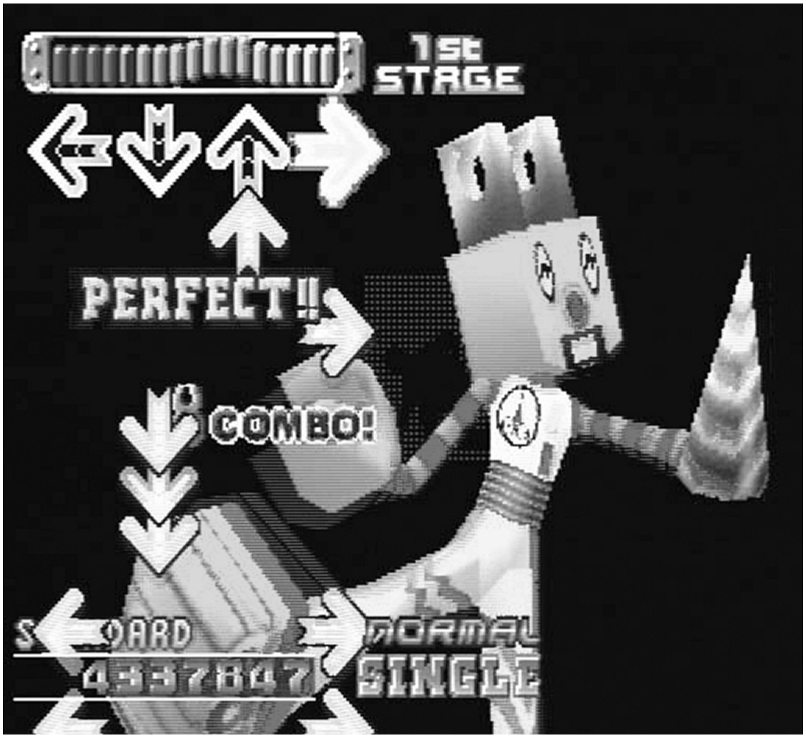


**Figure 7.3** *Dance Dance Revolution.*

A game can contain many challenges of each different type. To save us the insanity of trying to analyze the challenge content of a whole slew of games and concluding that they all have all the challenge types, it will serve our purpose to define two classes of challenge: *implicit* and *explicit*.

An *explicit* challenge is an intentional challenge specifically designed by the game designer. An example is the exact timing required to dodge the swinging pendulums in *Quake III Arena* (see Figure 7.4). This kind of challenge tends to be more immediate and intense than an implicit challenge.

**Figure 7.4** *Quake III Arena*.

An *implicit* challenge is one that is not specifically designed in; in other words, it is an emergent feature of the game design. An example of an implicit challenge is figuring out the most efficient way to distribute items among your group in a traditional computer role-playing game (CRPG) such as Black Isle's *Baldur's Gate*. Implicit challenges tend to be more drawn out and less focused than explicit challenges.

Having stated that the challenges present in games are mostly mental, let us take a closer look at the many forms these challenges can take. It's important to note that the following sections describe pure archetypal challenges; that is, they can be categorized as a simple challenge type, such as logic-based or reaction time-based. Not all challenges can be categorized so easily: The "challenge space" is not populated by a set of discrete points representing the archetypal pure challenge types, but instead is a smoothly varying continuum. Challenges can be hybridized (for example, a logic-based puzzle requiring a fast reaction time) and rarely—if at all—appear in their pure form.

### Logic and Inference Challenges

*Logic and inference challenges* test the ability of the player to assimilate information and use that information to decide upon the best course of action.

Logic is primarily used when the player is presented with perfect information, as in chess. In classical game theory, there are two broad classes of game: those of *perfect information*, with the complete state of play known to each player at all times, and those of *imperfect information*, with each player knowing only a fraction of the state of play (and not necessarily the same fraction for each player). For example, chess is a game of perfect information because the player is at all times aware of the state of the board and the position of all the pieces—both his own and his opponent's. Theoretically, given enough time and processing power, it is possible to analyze the game of chess to produce a perfect strategy. A perfect strategy is one that yields the maximum benefit to the player at all times. In the case of chess, this means that a user of that strategy would never lose. Of course, with the number of possible permutations of the chessboard and the move sequences, it would be beyond any human to blindly commit that strategy to memory, just as it is currently beyond any computer to calculate it.

When played in its puzzle mode, *Chu Chu Rocket* (see Figure 7.5), by Sega, is an example of a game of perfect information. The player is given a clearly defined win condition, a known playing field, and a known set of pieces to lay on that playing field. Hence, the player has perfect information. Knowing the rules governing the cat and mouse movement allows the player to predict (a *pattern-recognition* challenge) the paths of the cat and the mice and to place the playing pieces accordingly. Then the game is started and the results can be seen. If the win condition is not met, the player can replay the level.

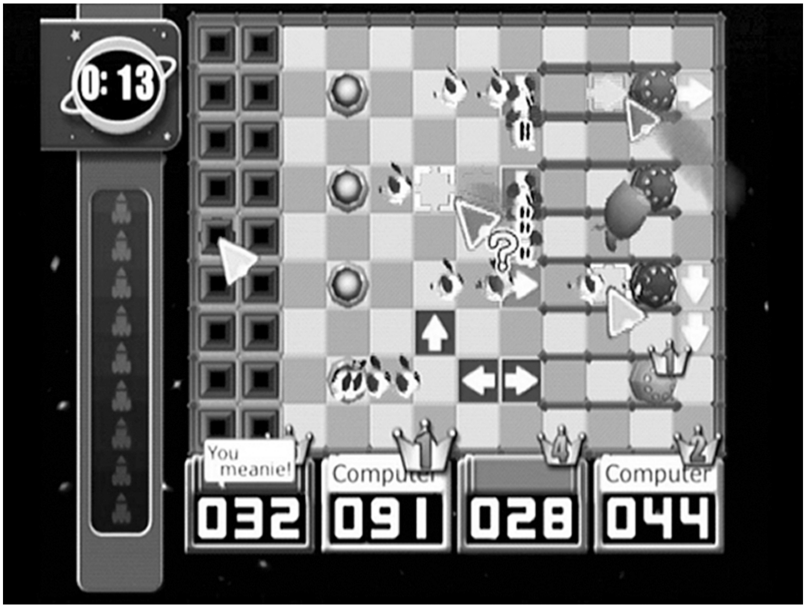


**Figure 7.5** *Chu Chu Rocket*.

In games of imperfect information, logic is not sufficient. Logic cannot fully operate given an incomplete knowledge of the state of play. In these cases, the gaps in that knowledge must be filled using *inference*. In this context, inference is the ability to surmise, or guess, the incomplete knowledge based on extrapolation of the existing facts.

*Microsoft Hearts* (see Figure 7.6) is an example of a game of imperfect information. Initially, you do not know the contents of the hands of the other players, but a skilled player can work them out to a reasonable degree of certainty by using the information revealed by which cards are passed and what tricks are laid during the course of the game.

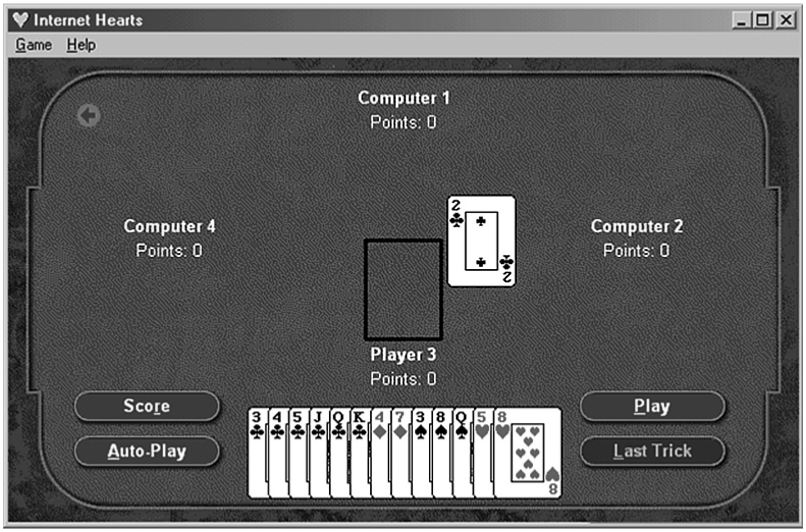


**Figure 7.6** *Microsoft Hearts.*

Bridge is another classic example of a game of imperfect information. A player does not know the contents of his partner's or his opponents' hands. He must use his knowledge of the game to calculate the best estimate during the course of the game.

The classic real-time strategy game staple, the "fog of war" shown in Figure 7.7, is a way of graphically representing imperfect information of a battlefield. The player can see only enemy units that are within the line of sight of any of his units. When an enemy unit goes into the fog of war (usually represented by a grayed-out area as the terrain was last seen, or a black area where the terrain has never been seen), the player can estimate where his enemies are and, based on his knowledge of the battlefield, attempt to draw conclusions about their intentions and plan his counterattack against them.


**Figure 7.7** The fog of war (right side of screen) in *Warcraft II*.

Games of imperfect knowledge are much more common than games of perfect knowledge. This is because one of the key elements of gameplay is challenging the player to hypothesize about the game worlds, forming her own internal picture. The degree to which this picture matches the real thing depends very much on the logic and inference skills of the player. It is much harder to design a good game without the element of mystery. Only a few designers can achieve this with any degree of success. Mystery can be viewed as the easy way out. There is no better way to hook a player than to get her involved in a compelling mystery story. Human curiosity is a very strong attractor, and any game that successfully taps into this provides a strong gameplay element. *Half-Life* did this extremely well, putting the player in the role of a new scientist trying to escape after a hideous cross-dimensional experimental error at his first day of work.

One problem with games of perfect information is that, because of the difficulty of designing an engaging playing experience without hiding anything from the player, they tend to be very simple. Usually, they are implemented as computer board games or simple arcade games. *Archon* (see Figure 7.8) is an excellent example of a computerized board game that was popular in the 1980s.

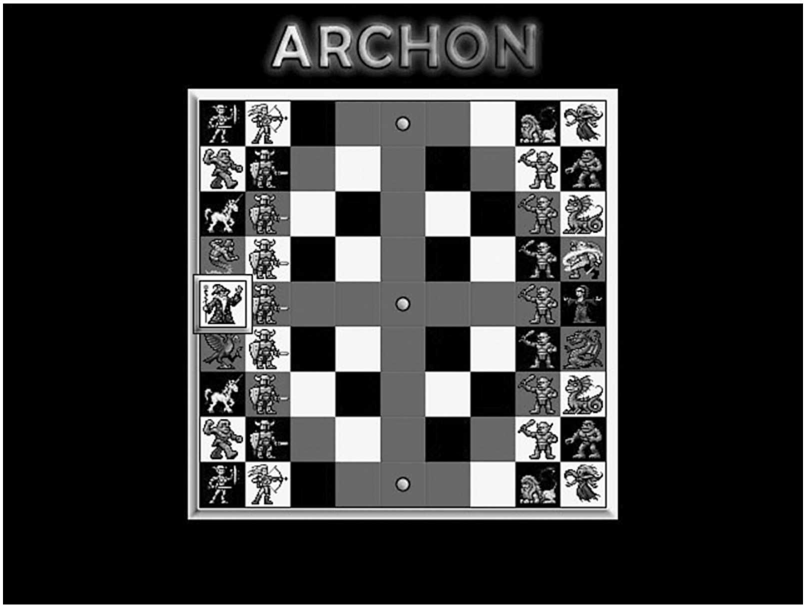


**Figure 7.8** *Archon.*

In *Archon* (and its sequels), the whole board was visible on the screen, and both players had full knowledge of the game state. In many ways, *Archon* was a computer-age successor to chess, combining elements of board-game strategy and arcade action in a single game.

## Lateral-Thinking Challenges

In some ways, *lateral-thinking challenges* are an extension of inference challenges. Certainly, they draw on the same core skills, but taken to the extreme. A lateral-thinking challenge tasks the player to draw on her previous experience and knowledge and combine them in a new and unexpected way.

This knowledge can be *intrinsic* or *extrinsic*. Intrinsic means that the knowledge was gained from within the game world—for example, figuring out a new combination of runes to cast a previously unknown spell, as was the case with the "flux cage" in FTL's *Dungeon Master* (see Figure 7.9). If the player figured out the meaning of the runes, it was possible to figure out roughly what purpose the unknown spell had, and the player needed to do that to win the game. No knowledge gained outside the game would have helped to figure out that particular problem (unless the player looked up the answer in a game magazine or on the Internet, but that's cheating).


**Figure 7.9** *Dungeon Master.*

The converse of intrinsic knowledge is extrinsic knowledge. This means knowledge that was gained outside the game world, perhaps in real life. For example, a player could use his knowledge that wood floats to retrieve a key attached to a wooden block just beyond his reach at the bottom of a narrow container by filling the container with water. Or, for an example from a published role-playing game written many years ago by Dave Morris (co-author of *Game Architecture and Design* by New Riders Publishing, 2004), the player could use her knowledge that repeated rapid heating and cooling of a metal object causes it to become brittle. This was the required technique to break through a metal door, otherwise impervious to both weapons and magic. Of course, the player wasn't dropped into this situation unprepared. There were clues to guide the player toward this solution.

*Half-Life* made great use of extrinsic knowledge-based lateral-thinking problems. In one particularly memorable sequence, the player had to figure out that the giant tentacled monster was sensitive to sound and then could use that as a detection mechanism, necessitating extreme stealth or noisy diversionary tactics in its presence. Not only that, but the player also had to make the mental connection between the oxygen and fuel pipes running throughout the level and the ominous rocket poised directly over the seemingly invincible tentacle. There are many other such puzzles in *Half-Life*, but these are particularly notable (and ingenious) examples.

### Memory Challenges

*Memory challenges* tax the player's memory of recent (and sometimes not so recent) game events. They are also purely intrinsic. That is to say, they rely specifically on the player's memory of events that have happened in the context of the game and do not rely on, for example, the player's memory of what he had for dinner a week ago.

Probably the best-known and most obvious example of a game based around a memory challenge is Milton Bradley's *Simon* (see Figure 7.10), a simplified electronic version of the classic children's game Simon Says. This game was very popular back in the 1980s. It had four buttons, colored red, yellow, green, and blue. When the player started a game, the computer flashed the buttons in a random sequence, although usually the game started with a single flash. After each sequence, the player had to repeat the sequence. If successful, the computer repeated the sequence again, adding one flash each time. The game was lost if the player made a mistake remembering the sequence. Many games—in particular, adventure games, role-playing games, and first-person shooters—make use of this particular memory-based challenge.

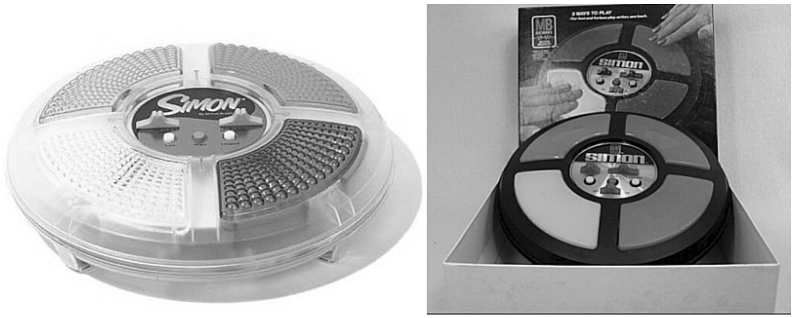

**Figure 7.10** *Simon.*

Nowadays, memory-based challenges are commonly seen in children's software, and even then they are usually hybridized with other types of challenge.

In fact, at the most basic level, it could be said that memory challenges are present in virtually every game; for example, remembering the layout of the complex tunnels onboard the Borg cube in Raven's *Voyager: Elite Force* is an example of an implicit memory challenge.

### Intelligence-Based Challenges

*Intelligence-based challenges* rely purely on the intelligence quotient of the player. This is extremely difficult to quantify and define, and, as far as we can tell, intelligence-based challenges do not exist "in the wild" in their pure form—at least, not in games.

In fact, the only place where this form of challenge exists in pure form is in official intelligence quotient (IQ) tests, such as those administered by Mensa, the organization for extremely intelligent people.

An example of an intelligence-based challenge, similar to those used by Mensa, is, given a sequence of similar shapes, to predict the next shape in the sequence from a choice of answers.

Intelligence-based challenges are included here as an archetype because they often form part of other challenges. Usually a more intelligent player will do better when playing a game using the more cerebral challenges.

### Knowledge-Based Challenges

*Knowledge-based challenges* rely on the knowledge of the player. As we have already touched upon, there are two types of knowledge to consider: intrinsic and extrinsic. Intrinsic challenges rely on knowledge from within the game world. Extrinsic challenges rely on knowledge external to the game world.

In the case of knowledge-based challenges, the ultimate real-world example is *Trivial Pursuit* (see Figure 7.11). This board game, which most people are familiar with, relies on general knowledge to win. A player's progress is determined by his answers to a set of questions in various categories, the vast majority of which are simple and straightforward—provided that the player knows the answer. Of course, in some cases the player can attempt to answer questions that he isn't sure of by listening for the clue in the question—crossing over into the territory of a lateral-thinking challenge. Clearly, this is an example of a game relying on extrinsic knowledge-based challenges to provide the gameplay. *Trivial Pursuit* has also been released in computer versions for various platforms since its debut in the mid-1980s.

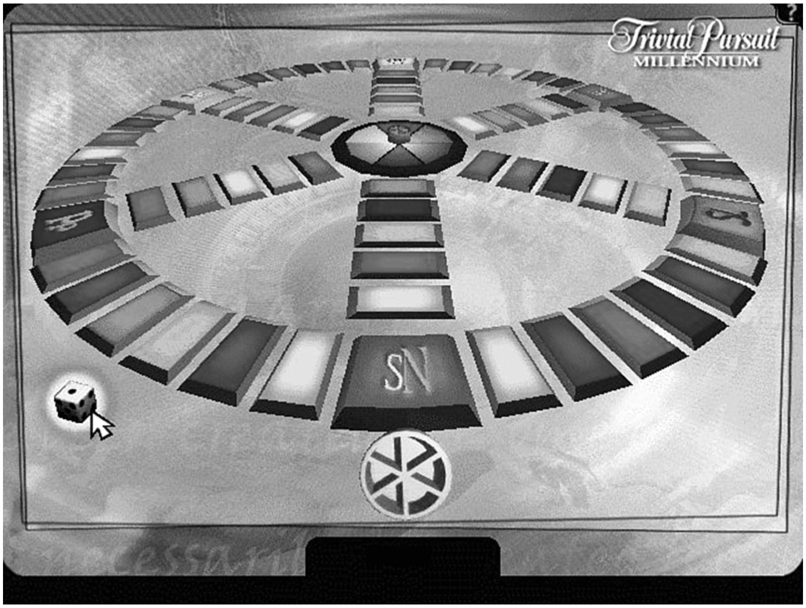


**Figure 7.11** *Trivial Pursuit Millennium Edition.*

More recently, *You Don't Know Jack* (see Figure 7.12) tests general (hence, extrinsic) knowledge in a quiz game format. However, this is an example that does not use knowledge-based challenges in their pure form. Instead, the questions are mostly phrased as a humorous lateral thinking problem and are set to a time limit so that players can—in most cases—figure out the answer with some (admittedly rapid) careful thought. In a lot of cases, knowledge-based challenges are inextricably linked with lateral thinking–based challenges. Except in certain rarified environments such as quiz games, knowledge-based challenges rarely appear in their archetypal form.

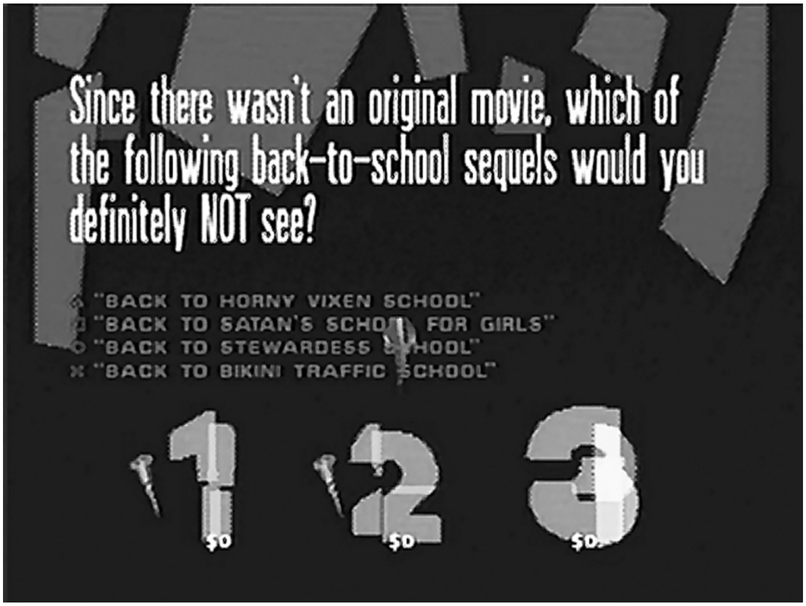


**Figure 7.12** *You Don't Know Jack.*

Intrinsic knowledge–based challenges are found in practically all games. However, explicit, intrinsic knowledge–based challenges are more often found in role-playing or adventure games. Here, a good knowledge of the game world and the background story and characters is essential to progress in the game. In real terms, this means that if you were to start a new game of, for example, Warren Spector's *Deus Ex* by loading a saved game provided by someone else, and it started you halfway through the game, you would have a much harder time trying to progress through the game than you would if you had started from the beginning.

## Pattern-Recognition Challenges

According to the theorists, the impressive abilities demonstrated by the human brain mainly stem from one basic ability: *pattern recognition*. In essence, our brain is a generalized pattern-recognition machine; our brain implicitly forms archetypes of objects and events and compares new experiences with these archetypes to recognize which category they fall under. For example, there are many different shapes and forms for tables, but somehow we always implicitly recognize a table when we see one, even if we have never seen that particular table.

According to some theories on learning, all types of learning are a form of pattern recognition and classification. When learning to speak, we are required to recognize and classify the sounds we hear as babies. In fact, to deal with everyday life, we are constantly recognizing patterns in events and using these to classify what is happening so that we can act according to past similar experiences. You know not to walk into a road without looking because you recognize the archetypal road, the archetypal event of walking across a road, and the possibility of the archetypal car or truck colliding with you and smearing you along several hundred yards of archetypal highway.

In this particular case, the human brain's ability to recognize patterns is sometimes overeager (for the technically minded, it uses a greedy algorithm) and can recognize patterns where there (arguably) are none. The name for this phenomenon is *pareidolia*, a type of illusion or misperception involving a vague or obscure stimulus being perceived as something clear and distinct. Human history is littered with examples of this: the constellations of stars in the night sky, the man in the moon, the whole field of astrology, and the articles that appear regularly in the *National Enquirer* proudly displaying the face of Jesus in a sesame seed bun. In fact, the Rorschach test, first published by Herman Rorschach in 1921, relies on the brain's overactive capacity for pattern recognition to attempt psychometric evaluation of the patient.

You can see this effect for yourself: Stare up at the clouds and see what they resemble (as an imaginative game designer, you should have no problem with this). For a slightly less subjective test, stare at the static on a television set for a minute or two, and you should begin to see imaginary structures pinwheeling about the screen. This is the brain attempting to find patterns where there are none.

A Google search on "nature versus nurture" and "pareidolia" will turn up lots of useful links on these subjects.

Figure 7.13 is a collection of common optical illusions. These illusions work primarily because of the way the brain's pattern recognition ability works. The top-left image is merely a set of straight lines with right angles, but we perceive it as an octagon with a square in the center. The top-right image could be taken from a Pac-Man conference, but we also see a phantom white triangle. The bottom-left image conjures up ghostly gray spots at the intersections. The bottom-right image appears to spin in different directions as you focus on the black dot in the center and move the page toward you.

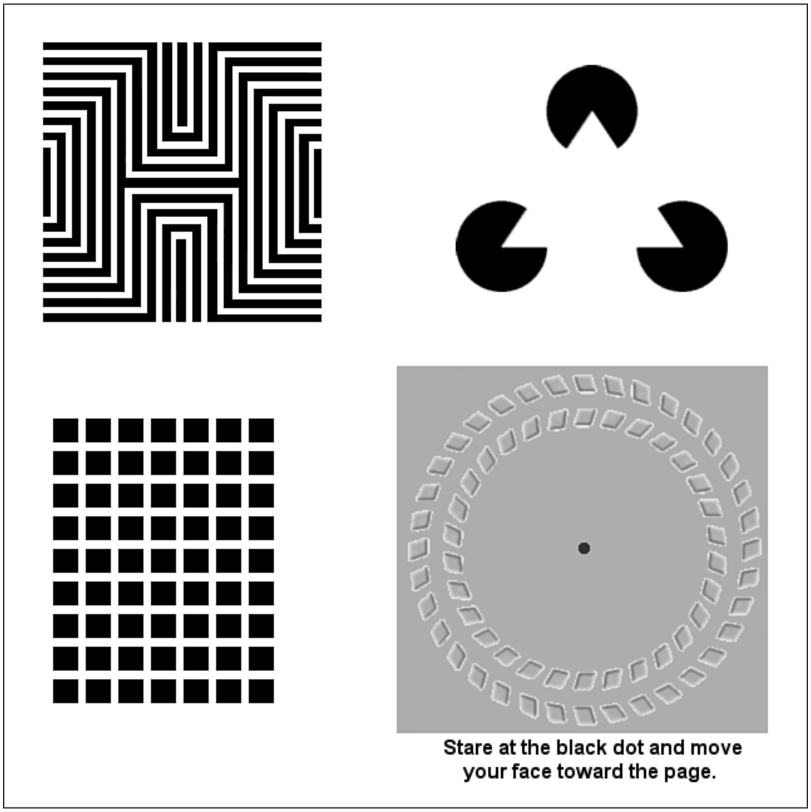


**Figure 7.13** Pattern recognition: There is no spoon!

In some fairly unique games, the brain's ability to recognize patterns can be tuned into, to enhance the ability of the player. An example of this is *Tetris*. *Tetris* can be played consciously, examining each block as it falls and actively deciding where to stack the block for best effect. However, the best players don't seem to play like this, especially at the later levels, where blocks fall too fast to be able to make any conscious decision where to put them. Instead, these players seem to tune into the game at an almost subconscious level and enter what we call the "*Tetris* trance," a Zen-like state in which the players seem to lose all track of time and don't concentrate on the specifics of the game board. Instead the players defocus and appear to process the entire playing area as a whole, without considering the individual elements. In fact, if these players were in the *Star Wars* universe, the Force would be strong in them.

In reality, however, it appears that these players are tapping into their brain's subconscious pattern-recognition ability to improve their game. *Tetris* is not the only game in which this occurs. Pretty much any game that uses pattern-recognition challenges as the primary gameplay mechanism can be played in such a way, although we certainly believe that it helps if those games have a clear and simple presentation. Maybe that is because the area of the brain dealing with pattern recognition is quite primal and, to process information quickly at that level, needs the information to be presented clearly so that minimal preprocessing is required. Of course, this is pure speculation on our part, but it is no coincidence that many of the older games that are now considered classics are those that can be played in this fashion. The one thing that all of these games have in common (apart from their reliance on pattern-recognition challenges) is their simple presentation. Classic games such as *Robotron*, *Defender*, and *Sinistar* all exhibit this feature.

So, if the brain's primary cognitive function is to recognize patterns, what does this mean in terms of gameplay? Pattern-recognition challenges can make or break a game, depending on how they are used. If in an entirely deterministic game one or more of the players can determine the pattern of play, this allows them to make 100% accurate predictions about game world events before they actually occur. Although they should be commended on their acumen, this does not make the game fun for the other players. This could rapidly degenerate to the situation in which it is almost as if the predicting player is a god of the game world and the other players are mere pawns, with no free will of their own.

**NOTE**

Note that the opposing players can be either humans with limited play options or a computer opponent that has been programmed to respond in certain ways to specific inputs. We heard a story once about a game with an adaptive computer opponent; the opponent's skill level depended on the perceived level of skill of the player. Soon players discovered that the easiest way to progress past difficult levels in the game was to deliberately do badly in the levels immediately preceding the difficult level, whereupon the computer immediately eased up on the player, making the difficult level slightly easier. Although this is an ingenious and valid approach, it is probably not what the designer intended, even from emergent behavior. No battle in the field has ever been won by the enemy commander sympathizing with his opponents' lack of ability and "going easy on them." Note that in the context of gameplay, adaptive difficulty is a useful tool. Just don't make it so recognizable to the player that she can exploit it to progress in the game. This is one pattern that you do not want the player to recognize.

Plenty of basic pattern-recognition games exist. A simple example that combines pattern-recognition challenges with reflex/reaction time-based challenges is the card game Snap. In this game, the players take turns laying a card from their hands face up on the discard pile, making sure that it is unseen by *any* player until the last possible moment. When the card is turned face up, the players check to see if it matches the card underneath (and by *match*, we mean it is of the same face value). That's the pattern-recognition challenge. If there is a match, the first player to shout "Snap!" wins all the cards in the discard pile. That's the reflex/reaction time challenge. If any players run out of cards, they are out of the game. The winner is the last player remaining with any cards in his hand.

In the early days of computer games, patterns were a lot more prevalent (or, at least, more obvious) in games than they are today. There could be any number of reasons for this. Maybe patterns were the most efficient way to code for an interesting game, given the limited processing power of the target platform. Another option is that the patterns are always there in games, but in the older games they stood out in stark relief against the simplicity of the gameplay. Games such as *Space Invaders* and *Galaxians* made heavy use of patterns. In many cases, playing effectively was simply a matter of memorizing the patterns and reacting accordingly. This play method persisted through most of the shoot 'em-ups that were produced until recently. However, even *Iridion 3D* released on the Game Boy Advance is a shoot 'em-up that defines attack wave patterns that can be learned and dealt with accordingly. This is a very transparent use of patterns and temporal pattern recognition, and it would be considered a bit simplistic and naive for unmodified use in a game design today. However, it is certainly a useful starting point for the inclusion of pattern-recognition challenges in your own game designs.

Slightly more advanced use of pattern recognition is evident in many games that involved exploration. For example, in *Doom*, secret doorways could be found by searching for an area of wall that looked slightly different from the norm. Also, games such as the previously mentioned *Dungeon Master* relied on pattern-recognition challenges for the player to decipher the complex systems of runes governing spells and spell casting.

Platform games, such as the *Mario* series of games, often rely on pattern-recognition challenges quite heavily. Not only are the levels carefully scripted to be a repeatable (hence, learnable) experience, but the end-of-level bosses also tend to behave according to a certain pattern. Thus, in *Super Mario Advance*, you can defeat one of the end-of-level baddies by carefully counting how many flaming spit wads she ejects and then attacking in the interim. In this case, the pattern-recognition challenge is used to make the game more manageable. It is difficult enough to manipulate the player avatar on the platforms (an example of coordination, spatial awareness, and reflex/reaction time challenges), but trying to handle unpredictable enemies on top of this would detract from the gameplay. This is an example in which two distinct challenge types work together synergistically to improve the gameplay potential. The whole is more than the sum of the parts.

### Moral Challenges

A *moral challenge* is a high-level challenge that can operate at several levels. Without delving too deeply into the field of metaethics, we can define these levels as universal, cultural, tribal, and personal. These levels are ordered from the all-encompassing to the specific. Each successive level affects a smaller moral area than the previous one. Usually, the lower levels have precedence, but that is not always the case.

Let us assume that there are no absolutes in morality. This implies that it is fundamentally incorrect to say that there is a definite right or wrong answer to a moral challenge; so much depends on context, emotional state, and past experience that an answer that might be correct for one individual would be totally wrong for another. An example: It is wrong to steal. But is it wrong to steal food if the only alternative is to starve? The answer to this depends on the individual.

But how does this example apply to games? In many games, the player is asked to make such choices. Raven's *Voyager: Elite Force* presents such a moral challenge early in the game: Should you save your teammate from the Borg and go against the captain's orders, jeopardizing the success of the mission?

We examine examples of the various forms that moral challenges can take in more detail. Before we can do this, however, we need to further define our various levels of moral challenge. Note that this is subjective: Exactly what defines the differences among universal, cultural, and tribal designations depends on context and the personal views of the observer. In the case of game design, it means that our definitions directly depend on the scope of the game. For example, a game set in America (with no mention of the rest of the world) would treat the whole of America as the universe. From here, the divisions of cultural and tribal entities would depend entirely on the game designers. They are under no compulsion to stick to reality—after all, it is their game.

A universal moral challenge is invariant no matter what the context is. By this, we mean that the correct moral outcome is independent of the entity making the choice. It would not matter if you were a human or a Zlerg from the planet Zlumpf—the correct choice would be the same. Universal moral challenges are concerned with the good of the universe as a whole. In the real world, they are most likely only a theoretical construct—a null container or superset for all the lower moral levels. They are extremely difficult to define and, as such, are a fairly rare form of challenge. In the limited context of a computer game, however, the cultural and universal morality levels are usually one and the same. (Often you will get a cultural moral challenge masquerading as a universal challenge; this is usually due to the game designer's inability to look outside her own backyard. This used to be a staple error in old sci-fi movies. Whenever the world was under threat, you'd see only America invaded—it was as if the rest of the world simply didn't exist.)

One of the main difficulties in defining a universal moral challenge is to define the limits. Do you say that the population of the world defines the moral universe, or is there life elsewhere in the universe governed by these morals? These are difficult metaphysical questions to answer, and the fact that games are set in a simulated universe does not make it any easier. Moral challenges are unusual in that they explicitly rely on the players' real-world experiences to provide their gameplay value. Hence, our views on the world directly affect our playing experience. For our purposes, we define the universal challenge as pertaining to all living beings in existence, within the confines of the game's simulated universe. With this definition in mind, we can infer that universal moral challenges are, at best, likely to be overly grandiose and, at worst, clichéd. As an example, imagine that the player is given a choice to go back in time to just before the birth of the universe and prevent it from happening. To simplify the choice, let's assume that the player's avatar is given amnesty from the effects of his choice: He would still exist and be able to live a (paradoxically) normal life, whatever the outcome. Given sufficient reasons for and against this would be a difficult moral choice to make. Should the player destroy all existence before it even comes into being, or should he allow things to happen as normal? (Obviously, you'd need a pretty good set of reasons for and against to make this into a difficult choice, but let's assume that the game designer has done a good job of setting that up for us.)

At a lower level than the universal challenge is the cultural challenge. Here we define a culture as a loosely affiliated collection of individuals all living by roughly the same standards; they do not necessarily have to be affiliated in any way other than their living standards and general lifestyle. For example, the Western world could be loosely viewed as a culture. If we wanted to take it down to a slightly finer grain, we could consider America as a culture. We could go further still and define Native American culture, Southern culture, Californian culture, and others.

Consequently, our definition of a cultural moral challenge is one that deals with the good of that culture as a whole. An example of dealing with the consequences of a moral challenge at the cultural level was provided in the 1988 film *Alien Nation*, directed by Graham Baker. In the opening scenes of this film, America (specifically, Los Angeles) is faced by a request for asylum from an escaped race of aliens genetically bred for slavery. The moral choice is whether to welcome the aliens into society, risking the dilution or destruction of human culture, or to turn the aliens away.

Fortunately for us, the smaller the scale of the moral choice is, the easier it is to define and give examples. Tribal moral choices are much smaller in scope. Note that the use of the word *tribal* is not intended to imply tribes in the full sense of the word; we use it here to mean any group of closely affiliated individuals. In a sense, a family unit can be considered a tribe, as can a role-playing adventure group and an American football team. Tribal moral choices are those that affect the well-being of the tribe. An example is the classic clichéd group decision in which all the group members have to decide which of them is going to have to perform some difficult—and, quite often, fatal—task to save the others. In fictional works, drawing lots usually solves this particular situation: a nonideal solution that avoids the difficult moral choice by abdicating the decision to the whims of chance.

Easiest of all to define, and perhaps the most familiar, is the personal moral choice. This is a moral choice made by an individual that has a direct outcome on that individual's own well-being and state of mind. There are no repercussions other than at the personal level for the player making the choice.

For example, in Will Wright's *The Sims*, the characters can earn money in a number of ways. A character can get a job and earn money the hard way, or he can become a professional widow: marry other characters and then kill them for the inheritance. The onus of this moral choice is really on the individual player. There are no lasting repercussions in the game world for murdering your husband or wife, and so (apart from the individual morals of the player) these are both equally valid methods of making money. This also depends on the player's level of involvement. It could be rendered more effective if there were unavoidable consequences within the game world. (The ghost of the dead Sim does not count; it can be removed by selling the tombstone.)

Moral dilemmas do not have to reside fully within one level. In fact, dynamically altering the priorities of these levels to force the player to decide between solving a moral dilemma within each fork in a different level can often lead to interesting and challenging gameplay. For example, we could posit a moral choice around the validity of the statement "The needs of the many outweigh the needs of the few."

So now we need some examples of real games that use moral dilemmas—but there is a problem. Until now, games have not sufficiently explored this area. Dealing with moral dilemmas has not traditionally been an area in which games excel. Morality in games has barely been considered at any level above simple "black and white" (no pun intended) playground morality. One reason for this is the difficulty of involving the player in difficult emotional situations; the willing suspension of disbelief required for the player to actively participate and believe in difficult emotional decisions is greater than that required for simpler choices. Hence, games that have employed moral decisions as a gameplay factor have relied on the simple "this is good, that is bad" approach. More recently, a game that has attempted (to some success) to deal with moral decisions in a more adult fashion is Lionhead's *Black and White*. Despite the title, the game attempts to deal with a moral spectrum. The player takes on the role of a god tending to the needs of her people. Aiding in the quest is a familiar, taking the form of a giant creature that can be trained to follow orders. The player is free to become any kind of god that she wants: from sickeningly good to terribly evil and anywhere in between. The nature of the god is reflected in the creature and the appearance of the land. How well this works in practice is open to discussion. So far, players have tended to gravitate directly toward total evil or total goodness. Although it cannot be strictly classed as a weakness or flaw, the cartoonlike nature of the game does undermine the seriousness of the moral decisions involved. This could be a good thing, of course—after all, you don't necessarily want your player to be racked by guilt for days after performing a questionable act. That would be going too far (if, indeed, it was possible).

## Spatial-Awareness Challenges

*Spatial-awareness challenges* are usually implicit. Only a handful of games have relied on explicit spatial-awareness challenges, and, in most cases, they were 2D games, such as *Tron* (the light-cycles game) and *Snakes*. A 3D version on the Sinclair Spectrum (Sinclair Timex in the United States) was entitled *Knot in 3D* (shown in Figure 7.14) and was a 3D extension of the classic *Tron*-based game. A recent update of *Tron* is shown in Figure 7.15. Spatial-awareness challenges are a specialized hybrid of a memory challenge and an inference challenge.

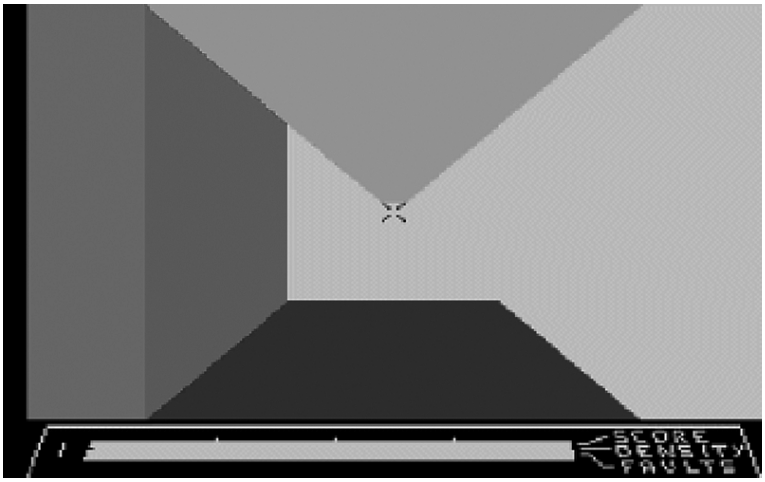


**Figure 7.14** *Knot in 3D.*

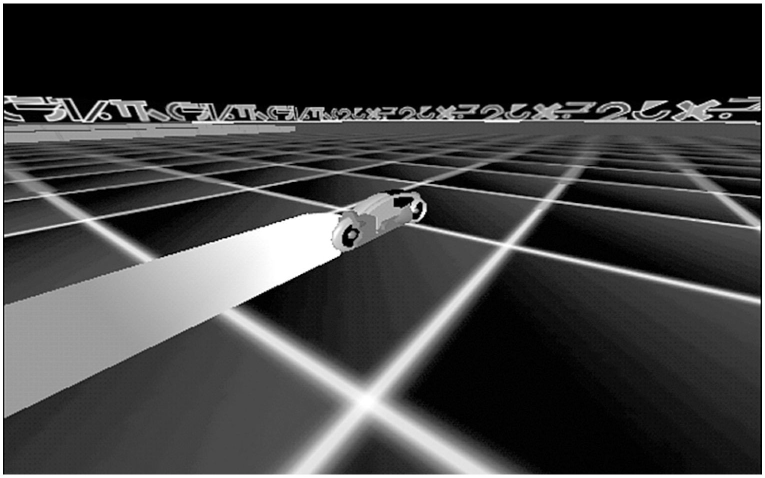

**Figure 7.15** *glTron.*

Games that rely on spatial awareness are usually 3D games. The challenge of representing a 3D world on a 2D surface, and the challenge to the player to make sense of that representation form the bulk of the spatial awareness problem. In many cases, the player receives aid in the form of a computer-generated map, but in other cases, such as *Quake III*, the player is left to his own devices to find his way around the world.

The types of games that usually rely heavily on spatial-awareness challenges are flight simulators, space-flying games, and 3D combat games (particularly *Quake III* and *Unreal Tournament*). To a lesser extent, 2D games that involve large playing areas, such as *Age of Kings*, also use spatial-awareness challenges.

### Coordination Challenges

Pretty much any game uses coordination challenges. Coordination challenges basically test the ability of the player to perform many simultaneous actions. They are almost always found in combination with reflex/reaction time challenges and are usually tightly coupled with them.

In its pure form, a coordination challenge is not dependent on any time constraints, but it isn't often found in the pure form. An example of a game (and there are many) that uses the coordination challenge to good effect (in combination with reflex/reaction time challenges) is *Super Mario*. Here, the player is expected to finely time jumps across wide chasms while avoiding circling enemies, requiring a plethora of accurately timed button presses from the player.

Shooting games of various sorts pose a challenge of accuracy: lining up a shot at a target, when the player or the target or both might be moving. Steering also requires accuracy. Flight simulators that properly model the behavior of aircraft, or racing simulators that accurately model the behavior of racing cars, require a high degree of precision. Airplanes, in particular, usually respond rather slowly to their controls. A player expecting an instant response will tend to overcompensate, pushing farther and farther forward on the joystick when the plane's nose doesn't drop right away, and then yanking it back in panic when it finally drops much farther than he intended in the first place.

Some games are forgiving about precision, allowing the player to be sloppy; others demand a delicate touch. Back before racing cars had airfoils to help hold them on the pavement, they flipped over very easily and required a much higher degree of skill from their drivers to keep them on the road. Papyrus Design Group accurately modeled this challenge in the game *Grand Prix Legends*.

Timing is the ability to overcome an obstacle by coordinating player moves with something else that is happening onscreen. Many video games present a weakness in an opponent's defenses for a limited period of time that, with practice, a player can learn to anticipate. Ducking under a constantly rotating hazard, for example, involves timing. Running and jumping across a chasm by pressing the Jump button at the last second is also an example of timing. It's related to reaction time, but instead of trying to do something as fast as possible, the player is trying to do something at exactly the right moment.

Many fighting games require complex sequences of joystick moves and button presses that, once mastered, will allow a "special move"—a particularly devastating attack, for example. These take a long time to learn and require very good motor coordination to achieve consistently. This sort of challenge is best suited to a player who can tolerate a high degree of frustration, or to a game that gives ample reward for this kind of persistence. Games that rely heavily on such techniques are difficult to balance. It is difficult to balance games that are based purely on physical dexterity. What one player might find easy, a different player might find impossible.

### Reflex/Reaction Time Challenges

*Reflex/reaction time challenges* test the timing abilities of the player. The simplest example of a reaction time challenge (which we previously mentioned) is the children's card game Snap.

However, reflex/reaction time challenges are usually not used in isolation in games and are often found in combination with coordination challenges. The types of games that most commonly exhibit this type of challenge are platform games, fast shoot 'em-ups, first-person shooters, and pure arcade games such as *Tetris* and *Centipede*.

This type of challenge is a factor of most action games. Only turn-based games, adventures, and role-playing games tend not to rely on reflex/reaction time challenges.

In an action game, the speed at which you operate the controls often maps directly to the speed at which your avatar reacts. This is not always exactly true because your avatar might be displayed by animations that require a certain length of time to execute, but in general, the faster a player can move and the better his reaction time is, the greater advantage he has. Good speed and reaction time are particularly valuable in fighting games.

### Physical Challenges

*Physical challenges* are extremely rare in games. The input methods available for computer games do not lend themselves to physical activity—at least, not without the purchase of specialized hardware.

Games such as *Samba De Amigo* and *Dance Dance Revolution* provide custom controller hardware, such as a special dance pad that enables the player to control the game by dancing on the pad. Others, such as Konami's *Hypersport*, don't use specialized hardware, relying on a standard joystick and, consequently, focusing the physical challenge to the hand and lower arm of the player.

Physical challenges are not often found in their pure form, and because of the expense and difficulty of including them in games, they are not often found at all.

## Applied Challenges

You will recall from Chapter 2 that gameplay consists of the challenges the player faces, plus the actions she can take to overcome them. As we said previously, designing the gameplay is one of your most important design tasks. To some extent, the nature of the challenge suggests the nature of the player's response. The best games, however, allow the player to think creatively and use unconventional actions to meet the challenges.

At the concept stage, you don't have to define precisely what challenges the player will face, but it's good to have an idea of what *kinds* of challenges you want in the game. *Applied challenges* are the application and use of the pure challenge forms we have discussed thus far. An applied challenge is a combination of one or more pure challenge forms applied to a given gameplay situation or style.

### Races

A *race* is an attempt to accomplish something before someone else does. It doesn't have to be a physical race through space; it can also be a race to construct something, to accumulate something, or to do practically anything else. Normally we think of races as peaceful, involving competition without conflict, but, of course, they can be combined with conflict as well. Because races put time pressure on the player, they discourage careful strategic thought and instead encourage direct, brute-force solutions. If the player has only 15 seconds to get through a host of enemies and disarm a bomb, he's not going to pick them off one by one with sniping shots; he's going to mow them down and charge through the gap, even if it means taking a lot of damage.

### Puzzles

Far too many kinds of puzzles exist to list here, but a *puzzle* is primarily a mental challenge. Often a puzzle is presented as a sort of lock that, when solved, opens another part of the game. The player is presented with a series of objects—often objects that are related in ways that are not directly obvious—and he must manipulate them into a certain configuration to solve the puzzle. To solve the puzzle, it's necessary to understand the relationship among the objects, usually by trial and error and close observation.

Players normally get all the time they need to solve puzzles. Because different people have differing amounts of brainpower, requiring that a puzzle be solved within a time limit might make the game impossible for some players.

A few games offer puzzles whose correct solution is not made clear at the outset. The player not only has to understand how the puzzle works, but also has to guess at the solution she is trying to achieve. We consider this a case of bad game design: It forces the player to solve the puzzle by trial and error alone because there's no way to tell when she's on the right track. *Infidel* was one such game. In the final puzzle at the end of the game, to open a stone sarcophagus, the player had to find 1 of 24 possible combinations of objects. There were no hints about which combination was correct; the player simply had to try them all.

## Exploration

*Exploration* is a key element of many games and is often its own reward. Players enjoy moving into new areas and seeing new things, but exploration cannot be free of challenge or it will just become "sightseeing." Sightseers can exhaust the entertainment of your game in such a short time that they won't perceive the value in the game; it will fail to entertain them for long. To prevent this, we design obstacles that make the players work for their freedom to explore.

The simplest sort of obstacle to exploration is the locked door. We don't literally mean a door with a lock in it, but any device that prevents the player from going on until he has done something to unlock it. You can require the player to do an infinite number of things: find a key elsewhere and bring it to the door; find and manipulate a hidden control (usually unmarked) that opens the door; solve a puzzle that is built into the door; discover a magic word; defeat the doorkeeper in a test of skill, either physical or mental; and so on. The trick is to make the challenge interesting and fresh.

Another common obstacle is the trap. A *trap* is a device that somehow harms the player's avatar when triggered—possibly killing her or causing damage—and, in any case, discouraging her from coming that way or using that move again. A trap is like a locked door with higher stakes: It poses an actual threat to the player. Traps can take a variety of forms:

- Some fire off once and then are harmless.
- Others fire but require a certain recycle time before they can fire again.
- Still others respond to particular conditions but not to others, like a metal detector at an airport.

A player might simply withstand some traps if they don't do too much damage; other traps can be disarmed or circumvented in some way. If a player has no way of detecting a trap and can find it only by falling into it, it's really just the designer's way of slowing the player down. It's not much fun for the player. For players, the real fun comes in outwitting traps: finding and disabling them without getting caught in them. This gives players a pleasurable feeling of having outfoxed you, the designer, even as you were trying to outfox them.

Yet another example is the maze. A *maze* is an area where every place looks alike, or mostly alike, and the player has to discover how the places are related to get out, usually by wandering around. Good mazes are implemented as a sort of puzzle, in which the player can deduce the organization of the maze from clues found in the rooms. Poor mazes simply put the player in an area and let her find the way out by trial and error.

Illogical spaces are a variant on the maze theme. In old text adventure games, it was not uncommon that going north from area A took you to area B, but going south from area B did not take you back to area A. The relationships among the spaces were illogical. This challenge requires the player to keep a map, because he can't rely on his common sense to learn his way around. In modern games with 3D engines, illogical spaces are more difficult to implement than they were in text adventures. Illogical spaces are now considered an outdated technique, but they still crop up from time to time. If you're going to use them, do so sparingly, and only in places where there's an explanation for it: "Beware! There is a rip in the fabric of space-time!" or some similar excuse—although preferably more original than this one.

Teleporters are the modern equivalent of illogical spaces. A *teleporter* is any mechanism that suddenly transports the player from where she is to someplace else. Teleporters are often hidden, which means that players trying to explore an area get caught in them and moved elsewhere without warning. If there are many hidden teleporters in an area, they can make it very difficult to explore. Teleporters can further complicate matters by not always working the same way, teleporting the player to one place the first time they are used, but to somewhere else the second time, and so on. They can also be one-way or two-way, teleporting players somewhere with no way to get back, or allowing them to teleport back again.

## Conflict

*Conflict* is a central element of a great many games because it seems almost inherent in the notion of winning and losing. To win a game, you have to beat the other players. The question is how you beat them. If you beat them by attacking them directly in some way, the game is about conflict. This doesn't necessarily mean combat or violence; checkers is a completely bloodless game, but it's still about conflict.

The challenges associated with conflict depend on the following:

- The scale of the action (from individuals to whole armies)
- The speed at which the conflict takes place (from turn-based, allowing all the time you want, to frenetic activity)
- The complexity of the victory conditions (from simple survival to complex missions with goals and subgoals)

*Strategy* is the mental act of planning: taking advantage of your situation and resources, anticipating your opponent's moves, knowing and minimizing your weaknesses. A strategic challenge is one in which the player must look carefully at the game and devise a plan of action. In a strategic game, the player's chance of winning depends greatly on the quality of her plan. Chance (luck) and missing information interfere with strategy. Chess is the classic strategy game because it contains no element of chance and offers complete information to both players. Nine Men's Morris and Tic-Tac-Toe are also pure, if simple, strategy games. Backgammon is a game with some strategy, but it also depends a great deal on luck.

Pure strategy games favor the player with a certain type of talent, and they appeal most to the kinds of people who have that talent. Because computer games are usually aimed at a broader audience, relatively few offer pure strategy games. They tend to include elements of chance and missing information as well.

*Tactics* involve putting a plan into *execution*, the process of accomplishing the goals that strategy calls for. Tactics are also about responding to unexpected events or conditions, which can include new information or bad luck. Even chess has tactics: The unknown quantity is your opponent, and she might make moves that you did not anticipate. Responding to them requires tactical skill.

It's possible to design a purely tactical game with no strategy. A small-squad combat game in which the soldiers are always moving into unknown territory contains no opportunities for strategy—you can't plan if you don't know where you're going or what you're up against—but many for tactics, such as keeping your soldiers covered, taking advantage of their particular skills, and so on.

The business of supporting troops in the field and bringing fresh troops to the front lines is called logistics. Most war games don't bother with logistical challenges such as transporting food and fuel to where they're needed. These activities are generally considered boring and distracting from the main purpose of the game, which is combat. Real armies have whole teams of people responsible for logistics and could never win without this support; computer games have only the player to handle everything, so it stands to reason that he should be concentrating on more exciting tasks such as attack and defense.

However, modern real-time strategy (RTS) games have introduced one important logistical challenge: weapons production. Unlike board war games, in which the player commonly starts with a fixed number of troops, RTSs now require the player to produce weapons and to research new ones from a limited amount of available raw material. The production facilities themselves must be constructed and then defended. This has changed the entire face of war-gaming, adding a new logistical challenge to what was formerly a purely combat-oriented genre.

In role-playing games, the limited size of the characters' inventories presents another logistical challenge. The player must frequently decide what to carry and what to leave behind. Equipping and balancing a party of heterogeneous characters with all that they need to face a dangerous adventure occupies a significant amount of the player's time. Of course, sometimes this is the fault of a badly designed inventory system, in which an apple takes up the same amount of space as a single coin.

On a smaller scale, personal conflict, as a one-on-one or one-on-many challenge, is a key feature of many action games. The player controls an avatar who battles directly against one or more opponents, often at very high speeds. The challenge of personal combat is immediate, exciting, and visceral.

The fundamental challenge in any game based on conflict is survival. If characters can be removed from the field of play by death or any other means, it is essential to preserve their lives or effective playing time, or you cannot achieve the victory condition. In a few games, survival is itself the victory condition and no other achievements are required, but in most, survival is necessary but not sufficient to win.

Survival is about defending one's self, but many games require that the player defend other things as well, especially things that cannot defend themselves. In chess, this is, of course, the king. This challenge requires that the player know not only the capabilities and vulnerabilities of his units, but also those of the thing he is protecting. He must be prepared to sacrifice valuable units to protect the vital item. *Lemmings* was an excellent game about sacrificing some units to preserve others.

Another important gaming challenge, first used extensively in *Thief: The Dark Project*, is stealth—the ability to move undetected. This is an extremely valuable capacity in almost any kind of conflict, especially if the player is the underdog. War games occasionally pose challenges in which the victory condition cannot be achieved through combat but must be achieved through stealth. *Thief* was designed entirely around this premise. Players had to achieve their missions by stealth as much as possible and had to avoid discovery or combat if they could.

The element of stealth introduces considerable complexity into the design and gameplay of war games. The simplest war games are traditionally games of "perfect information," in which both players know everything about one another. Imagine how difficult chess would be if there were an invisible piece somewhere on the board that could be discovered only by accident.

## Economies

An *economy* is a system in which resources move around, either physically from place to place, or conceptually from owner to owner. This doesn't necessarily mean money; any sort of resource that can be created, moved, stored, earned, exchanged, or destroyed can be involved. Most games contain an economy of some sort. Even a first-person shooter has a simple economy: Ammunition is obtained by finding it or taking it from dead opponents, and it is consumed by firing your weapons. Health points are consumed by being hit and are restored by medical kits. The designer can make the game easier or harder by adjusting the amounts of ammunition and medical kits available, and a player who is running short must meet the challenge of obtaining more somehow.

Economic challenges are defined in terms of the flow of resources. Some games, such as *Theme Park*, consist only of economic challenges; others, such as first-person shooters, combine both economic and conflict challenges.

In many games, the challenge is simply to accumulate something: wealth, points, or anything else of intrinsic value. The object of the game might be to accumulate more money, plutonium, or widgets than your opponents. This is the basis of *Monopoly*, of course, and many other games. The game challenges the player to understand the mechanisms by which wealth is created and to optimize them to his own advantage. In the case of *Monopoly*, it's helpful to mortgage low-rent properties and use the cash to purchase high-rent ones because high-rent properties are the real source of wealth toward the end of the game. Players who understand this are at an advantage over those who don't.

Requiring your players to achieve balance in an economy gives them a more interesting challenge than simply accumulating points, especially if you give them many different kinds of resources to manage. *The Settlers* is a series of games involving complex interactions among resources: Wheat goes to the mill to become flour, which goes to the bakery to become bread. Bread feeds miners who dig coal and iron ore, which goes to the smelter to become iron bars, which then go to the blacksmith to become weapons, and so on. All of these resources have to be produced and transported to establish a balanced economy. Produce too little of a vital item, and the whole economy grinds to a halt; produce too much, and it piles up, taking up space and wasting time and resources that could be better used elsewhere.

A peculiar sort of economic challenge involves looking after a person or creature, or a small number of them, as in *The Sims* and *Creatures*. Unlike a large-scale simulation such as *Caesar*, in which the player must build and manage an entire town, these smaller-scale simulations focus on individuals. The player must meet the needs of each individual and take into account the unique characteristics that differentiate each one from other individuals. The challenge is to make sure their needs are met and perhaps to improve their growth in various ways. The creatures often behave unpredictably, which adds both to the challenge and to the charm.

### Conceptual Challenges

*Conceptual challenges* are those that require the player to understand something new. To the game designer, conceptual challenges are the richest and most interesting to design because they offer the broadest scope for innovation. They can also be difficult to design and even more difficult to program. Conceptual challenges often occur in construction and management simulations, in which the game is simulating processes that the player must come to understand. In *Sim City*, for example, there is a direct relationship between an efficient transportation system and economic prosperity. The player who does not deduce this will have difficulty with the game. *Sim City* challenges the player to comprehend this and many other relationships involved in town planning.

Another sort of conceptual challenge occurs in mystery or detective games, in which the object is not merely to accomplish certain feats, but also to examine the evidence and deduce who committed the crime and how. The game *Eagle Eye Mysteries* is an excellent example of this: Players follow clues, ignore red herrings, and arrive at a theory of the crime, assembling the relevant evidence to demonstrate proof. *Planescape: Torment* also offered significant conceptual challenges and had several different endings, depending on how the player interpreted a complex and bizarre series of events.

### Gameplay Worksheet

1. What types of challenges do you want to include in your game? Do you want to challenge the player's physical abilities, his mental abilities, or both?
2. Game genres are defined in part by the nature of the challenges they offer. Have you selected a genre in advance, and if so, what does that imply for the gameplay? Do you intend to include any cross-genre elements, challenges that are not normally found in your chosen genre?
3. Does the game include implicit challenges (those that emerge from the design), as well as explicit challenges (those that you specify)?
4. If the game has a story, how does the story influence the gameplay, and vice versa? Do they operate in tandem, or are they effectively separate pieces?
5. If the player has an avatar, how does the gameplay influence the avatar's appearance and capabilities?
6. Is the game's collection of challenges a related group, or is it a compilation of unrelated elements? If the latter, does that have any effect on the player's suspension of disbelief?
7. Given that not all players enjoy the same kinds of challenges, how does the game's target audience influence the challenges it includes? What challenges will you deliberately exclude?
8. Will the player be required to face more than one challenge at a time? Which ones?

## Putting It Together

As we have discussed in this chapter, there is no single aspect of any game that we can point to and identify as the gameplay. That is because gameplay is not a singular entity. It is a combination of many elements, a synergy that emerges from the inclusion of certain factors. If all of those elements are present in the correct proportion and style, we can be fairly sure that the potential for good gameplay is there; consequently, we can presume (but not be certain) that we have a good game. The gameplay emerges from the interaction among these elements, much in the same way as complex automata emerge from the simple rules of *Conway's Game of Life*.

There is a particular paradox known as the *Sorites Paradox* or *Heap Paradox*. It concerns a pile of sand. An observer is asked whether sand is a pile, and the answer is yes. Then a grain of sand is taken away. The question is repeated, and the answer is still in the affirmative. This process continues, and then at some point, the observer will say that it is no longer a pile. The question then posed is to ask why one grain of sand makes a difference between a pile and a nonpile. Can the observer state a specific number of grains of sand that define a pile? It's back to the familiar "argument of the beard": Why is the observer's definition any better than another observer's definition?

The same applies to gameplay, although on a smaller and coarser-grained scale. In a *gedanken experiment*, we can look at a game and take away an element (or part thereof) of gameplay. (For example, we could disable Mario's ability to turn left in *Mario 64*.) We can then pose the question "Does it still have gameplay?" We can continue to remove elements or sub-elements and pose the same question. At some point, the game will be sufficiently crippled for the observer to say that it no longer has gameplay. This point will be different for every observer. Whose opinion is best? That's a question for the philosophers. In short, we cannot define exactly how many gameplay elements are required to make a game. We cannot even state with certainty which are required and which are superfluous. We can only state that, to have gameplay, we need some or all of these elements; to have a pile of sand, we need some or all of these grains.

Much the same way that we can expect to find elements indicating gameplay, we can expect to find opposing elements that indicate the absence of gameplay. By this, we mean that the inclusion of the particular element could be detrimental to the gameplay or, more rarely, that gameplay is not present at all. The game in question might have included all of the elements expected to indicate good gameplay, but it might have also muddied the mix by including extra unwelcome elements that detracted from the positive effects of the good. We have all played games that were almost perfect, apart from one or two annoying flaws: Maybe the difficulty level ramped too quickly, maybe the controls were unwieldy, or maybe the collision detection was slightly suspect. Whatever the cause, it has the overall effect of taking a potentially superb game and knocking it down a peg or two, reducing it to the rank of failed contender. This determines the difference between the excellent and the merely good.

It would seem fairly obvious to the game designer that she is including some suspect elements to the gameplay and, therefore, would make efforts to eliminate them from the design. This has happened. A particular case of note is Blizzard Entertainment's *StarCraft*. This game was continually tweaked right up until the point of release, to ensure that the gameplay and unit/unit balance was as good as possible. Even so, they didn't quite get it right, and so the expansion pack, *Brood War*, made further changes to the unit/unit balance—the most notable being an increase in usefulness of the Terran marine and an overhaul of the air-air and air-ground combat units.

The presence or absence of these elements of gameplay can often be inferred only by the existence of their indications or contraindications. We examine these in more detail in the genre-specific chapters.