

[< PREVIOUS](#)[NEXT >](#)

Chapter 2. Game Concepts

This chapter addresses the earliest stage of game design: getting and refining an idea for a game. Your goal at this point should be to write the high-concept document that we described in [Chapter 1](#), "What Is Game Design?." To do this, you don't have to have all the details worked out yet. But you do need a clear understanding of what your game is about, and you must answer certain essential questions. When you've answered those questions to your satisfaction and have written down the answers, you have turned your idea into a *game concept*. At the end of this chapter, we've included a worksheet to help you out.

[< PREVIOUS](#)[NEXT >](#)

Getting an Idea

Game ideas come from almost anywhere, but they don't walk up and introduce themselves. You can't sit around and wait for inspiration to strike. Creativity is an active, not a passive, process. You have to put yourself in an inquisitive frame of mind and then go out and look for game ideas. Look everywhere. Some of the most mundane things could be hiding a game idea. Even delivering newspapers provided the basis for a successful arcade game, *Paperboy*, though developers spiced up the job by letting the player break windows with the newspapers and making him dodge cars on his bicycle.

One idea isn't enough. It's a common misconception that a brilliant game idea will make you a fortune. In fact, this occurs extremely rarely. You might think you have the game idea of the century, but concentrating on it without bothering to think about other game ideas is a little like pinning all your hopes on a single lottery ticket and not bothering to get up for work while you wait to see if your numbers come up. Unlike lottery tickets, ideas are free, so think about new ones constantly. Make a note of them and go on. If one seems especially promising to you, then you can start to expand and refine it, but don't let that prevent you from thinking about other games as well. When thinking up game ideas, more is always better.

Dreaming the Dream

Many game ideas begin as dreams. Not real dreams, but daydreams, things you think about when you're staring out the window or watching the clouds on a summer afternoon—these are the thoughts that you have when you let your mind roam free.

Computers can make dreams real. This is the unique characteristic of interactive entertainment that sets it apart from all other forms. Interactive entertainment can take you away to a wonderful place and there let you do an amazing thing. Books and movies can't do that. They can take you away to a wonderful place, but they can't let you do an amazing thing. Books and movies can create fantastic worlds and show them to you, but they can't let you be a part of them. Computer games create worlds, and they can let you live inside of them as well.

A lot of computer games are light entertainment, designed to while away a few minutes with a puzzle or a simple challenge. But larger, richer games begin with a dream. If you've ever thought to yourself, "I wish I could..." or "Imagine what it would be like to...", then you've taken the first step on the road to creating a computer game. The computer has the power to simulate reality (with varying degrees of accuracy), but, more important, it has the power to simulate dreams. Computers can create almost any sort of experience you can imagine visually, even experiences that are physically impossible in the real world. The design of a computer game begins with the question, "What dream am I going to fulfill?".

Perhaps it's a dream of exploring a dungeon infested with monsters. Perhaps it's a dream of coaching a football team. Perhaps it's a dream of being a fashion designer. But before you do anything else, you must dream the dream. Understand it. Feel it. Know who else dreams it and why.

Game Ideas from Other Media

Books, movies, television, and other entertainment media are a great source of inspiration for game ideas. The game *Interstate '76* (see [Figure 2.1](#)) was inspired by 1970s cop shows. Movies such as the James Bond series often inspire games. Any story containing an exciting action with something important at stake can form the kernel of a game. Think over the books you've read and the movies you've seen, and ask yourself whether any of the scenes in them could serve as the basis for a game.

Figure 2.1. *Interstate '76* was a great game inspired by another medium.



You can't, of course, go stealing other people's intellectual property. Even if the Pirates of the Caribbean ride at Disneyland seems like the basis for a great game, you can't go ahead and make it without Disney's approval. But you can certainly make a lighthearted game about pirates—as LucasArts did with its *Monkey Island* series.

You should also look beyond the usual science fiction and fantasy genres and beyond the usual sources. How about poetry? Beowulf's epic battle with the monster Grendel and then his even more terrible battle with Grendel's mother in a cave at the bottom of a lake sounds like the basis for a game. "The Charge of the Light Brigade" might make you wonder about cavalry tactics. What are the advantages and disadvantages? Would a game based on cavalry warfare be interesting to anyone? It's worth thinking about. The smash-hit game *The Sims* was partly inspired by a nonfiction book called *A Pattern Language*, which is about the way people's lives are affected by the design of their houses.

Game ideas can crop up in all sorts of unlikely places. Just as great scientists look at even the most common things in the world and ask how they work, great game designers are always looking at the world and wondering if it can be made into a game. The trick is to develop a game designer's instincts, to look for the fun and challenge even in things that don't sound like games at all.

Game Ideas from Other Games

A great many people who play computer games want to design them as well. Something about playing games stirs up people's creative juices. When you play a lot of games, you develop a sense of how they work and what their good and bad points are. Playing games is a valuable experience for a game designer. It gives insight and lets you compare and contrast the features of different games.

Sometimes we get game ideas through frustration. Most of us have had the experience, at one time or another, of playing a game that wasn't quite right somehow. The user interface was awkward, the game was too difficult, or the payoff was boring. We think, "If I had designed this game, I would have...." We have in our minds an imaginary ideal game, the one that would fix all those problems and the one that we would make if we had the chance.

To learn from other games, you have to pay attention as you play. Don't just play them for fun; look at them seriously and think about how they work. Take notes especially of things that you like or don't like and of features that seem to work particularly well or particularly badly. How do resources flow into the game? How do they flow out? How much is luck, and how much is skill?

As creative people, our instinct is to devise totally new kinds of games that have never before been seen. Unfortunately, publishers want games that they are sure they can sell, and that usually means variants on existing genres, perhaps with a new twist that they can use in marketing. This is why we keep seeing sequels and thinly disguised copies of earlier games. As designers, we have to learn to balance the tension between our own desire to innovate and the publisher's need for the comfortably familiar. Leonardo da Vinci warned against persistent imitation, however, in his *Treatise on Painting*:

"The painter will produce pictures of little merit if he takes the work of others as his standard; but if he will apply himself to learn from the objects of nature he will produce good results. This we see was the case with the painters who came after the Romans, for they continually imitated each other, and from age to age their art steadily declined... It is safer to go directly to the works of nature than to those which have been imitated from her originals, with great deterioration and thereby to acquire a bad method, for he who has access to the fountain does not go to the water pot."

There is a downside to deriving game ideas from other games. It tends to result in games that look or work alike. It's an evolutionary, not a revolutionary, approach. Deriving game ideas from other games is an excellent way to learn about games and gameplay, but if pursued exclusively, it produces similarity and, ultimately, mediocrity. The greatest games break new ground. They're unlike anything seen on the store shelves before. To achieve that, you have to dream.

From Dream to Game

A dream or an idea alone is only a start; it is not enough to make a game. A dream is a fantasy that you have by yourself. You can make computer games purely for yourself if you like, but most of us don't have the money to do that. A computer game is something

that you make for someone else. You'll also discover after you've built a few games that playing a game that you worked on is a very different experience from playing a game that someone else has created. When you know what's on the inside and how it works, some of the fantasy is lost. Just as actors often don't watch their own movies, some game developers don't play their own games. For one thing, of course, if it's a single-player game, they already know how to beat it. But the experience, the dream, isn't quite the same when it's a game you built yourself. In your heart, you know it's an artificial simulation.

The chief purpose of a computer game is to entertain someone else. This means that you and your development team are the performers, the people who create the entertainment. An essential part of your job is communication, transmitting your dream to your audience, the players. If the game is in a well-known genre and setting (for example, a World War II flight simulator), you can be pretty certain that a number of people already share your dream. But if your game is in a new setting (a futuristic city of your imagination, for example)—and especially if it's in a new genre—you have to be very careful and thorough in communicating your dream to others. Some of the very first questions a publishing executive is going to ask you are "Why would anyone want to play this game?" and "What's going to make someone buy this game instead of another?"

But what does it mean to entertain someone? Many people think entertainment is synonymous with having fun, but even that isn't completely straightforward. People have fun in all kinds of ways. Some of those ways involve very hard work, such as gardening or building a new deck. Some of them involve frustration, such as solving a puzzle. Some, such as athletic competitions, even involve pain. One person's entertainment is another person's insufferable boredom. In building a computer game that entertains, it's important to understand *how* it entertains. And to do that, you have to know what a game really is.

[< PREVIOUS](#) [NEXT >](#)

The Elements of a Game

In this section, we give a formal definition of a game and describe the elements that comprise one. Remember that we're talking about any kind of game, computerized or not. We continue to use these terms throughout the book.

Games, Toys, and Puzzles

A *game* is a form of participatory, or interactive, entertainment. Watching television, reading, and going to the theater are all forms of passive entertainment. In those media, the entertainment is presented *to* you, and you're not expected to participate. In some plays, the audience itself has a role, but even then the actors are in control. The content of the entertainment is the drama, and the way you perceive it is by watching it. This mode is fundamentally passive: They act, you watch.

A game is a much more complicated thing. When people play a game, they are being entertained by actively participating. Although we use the term *interactive entertainment* to refer to computer and video games, any game is interactive if you're taking part in it, whether it's softball or postal chess. Active forms of entertainment are slowly gaining ground over passive forms; people are starting to play online role-playing games instead of watching TV. People love the feeling of involvement and empowerment that gaming gives.

A game takes place in an artificial universe that is governed by *rules*. The rules define the *actions* or *moves* that the players may make in the game, and also the actions that they may not make. In a computer game, most of these rules are hidden. Because you can interact with the game only through the machine's input devices, and the machine can ignore inappropriate inputs, you don't have to be told the rules explicitly. Computer games generally allow you to try anything you want; you can presume that anything you can do, you are allowed to do.

When you're playing a game, especially board games and computer games, you're often playing a role of some sort. Defining the player's role is a key part of game design, and we'll discuss it in more detail later.

Other kinds of interactive entertainment are not games. A *toy* is an object that you play with without rules. You can play a game with a toy if you make up some rules to play by, but ordinarily a toy does not come with rules.

Unlike a toy, a *puzzle* does have one definite rule: It has a correct solution that you are trying to find. Puzzles are normally worked by one person. Typically, they require problem-solving skills and the ability to think ahead, but they don't involve any role-playing and are not set in an artificial universe.

Challenges, Gameplay, and the Victory Condition

The rules also define the obstacles, or *challenges*, that the players must overcome to win the game. The challenges, together with the actions the players can take to meet them, make up the *gameplay*. Defining and tuning the gameplay is the largest and most difficult task in designing a game; this is discussed extensively in [Chapter 7](#), "Gameplay."

Included with the rules of most (but not all) games is a special rule that defines the *victory condition*: a state of affairs in which one or more players are said to be the winner. Usually, the first player to achieve the victory condition is the winner and the others are the losers. These kinds of games are said to be *competitive*—that is, the players are trying to achieve the victory condition for themselves while preventing the others from doing so. Some games define the rules in such a way that all players must work together to achieve the victory condition; these are called *cooperative* games. If groups of players work together against other groups of players, the game is a *team* game. Most noncomputerized games—and many computerized ones as well—are *multi-player* games; if a game is played by one player alone, it is a *single-player* or *solitaire* game. Many video games sold at retail offer several different play modes: single-player, two-player competitive, two-player cooperative, and so on. Sports games are particularly well suited to these kinds of variations.

In addition to a victory condition, many games have one or more *loss conditions*. Sometimes the loss condition is implicit: If you aren't the first to achieve the victory condition, you lose. But in others, the loss conditions are explicit: Your character has died, for example, or you have run out of some vital resource.

Some computer games have no victory condition. These are often construction and management simulations such as *Sim City*. Rather than trying to "win" by achieving the victory condition, you can set your own goals for what you're trying to achieve. Or, you can just play around with the game without trying to achieve anything in particular. (*Sim City* also has some scenarios with victory conditions.) These kinds of games are sometimes referred to as *software toys*.

Many single-player arcade games, such as *Tetris*, are peculiar in that they have no victory condition. They are effectively unwinnable; the game just gets harder and harder until eventually you must lose. You do, however, get a score based on your ability, and your goal is to be listed among the top players. In effect, the victory condition is to do better than other people who play the game, but you can never actually win the game itself.

Setting, Interaction Model, and Perspective

A game also takes place in a *setting* or world. A sport such as football, for example, takes place on a field with defined boundaries. A board game takes place on and around the board. Even simple games with cards, dice, or dominoes take place on a tabletop, and the players implicitly agree that the table is the only legitimate playing surface. Because a computer screen has the capacity to display anything, a computer game can take place in any setting imaginable. We talk more about game settings and worlds in [Chapter 3](#), "Game

Settings and Worlds." For now, it's enough to know that a game's setting is part of its concept.

The way that the player interacts with the game world—takes actions to overcome the challenge—is called the game's *interaction model*. There are many different interaction models; some games have different models at different points in the game or allow the player to choose from a selection of models.

Two interaction models are particularly common in computer gaming. If the player plays in the game world by controlling a single character or piece that represents him, and if that character exists in a single place and can influence only the local area around him, that character is called his *avatar*. In Nintendo's *Mario* games, Mario is the player's avatar. On the other hand, if the player has the ability to view different parts of the game world, taking actions in many different places, the player is said to be *omnipresent* (even if parts of the world are hidden from him at times). This applies even if the player can act only by directing characters or units that belong to him; if he has the power to give directions to all of them independently, he is omnipresent. A good example of this kind of game is chess. The player can move any of his own pieces on the board, no matter where they are.

Perspective describes how the player actually sees the world on the screen. In war games, players usually have an aerial perspective above the battlefield. If the view is from directly above, like looking at a map, this is called a *top-down* perspective. Most old war games, as well as the original *Sim City*, used this perspective. Beginning with *Populous*, however, many aerial-perspective games adopted a more dramatic viewpoint. In this the view is at a 30- or 45-degree angle from the vertical, looking slightly across the landscape and not just straight down. The landscape is also rotated 45 degrees with respect to the bottom of the screen, so the player is always looking at one corner of rectangular objects such as buildings. This is called an *isometric* perspective. This viewpoint makes the player feel closer and more involved with events than the top-down perspective. It has the disadvantage that, if it can be rotated in 90-degree steps, the art team has to draw four different versions of everything that can appear on the ground, from each of the four angles.

Many games offer different perspectives or camera angles that the user can change. Not all camera angles suit all games; perspectives that are visually stunning to watch from can turn out to be impractical during actual gameplay. *Madden NFL Football*, for example, normally uses a perspective in which the camera is above and behind the ball carrier, looking somewhat downfield toward the goal. But it also provides a sideline camera, showing the field as it would appear to someone sitting in the stands. Although this is a very familiar perspective and is great for taking screenshots that look like real football on television, the game is quite difficult to play from this angle.

You're probably familiar with many common perspectives: *first person*, in which you look through the eyes of your avatar (most commonly found in first-person shooters such as *Half-Life*); *third person*, in which you follow behind your avatar in a three-dimensional world (*Tomb Raider* is one of the best-known examples); and *side-scrolling*, commonly seen in older video games such as *Sonic the Hedgehog*. With the introduction of 3D display engines, games can use any number of perspectives at no additional cost in art development time.

The Player's Role

When you're playing a game, especially board games and computer games, you're often playing a *role* of some sort. In *Monopoly*, you're playing a real estate tycoon. In *Goldeneye*, you're playing James Bond. Defining the player's *role* in the game world is a key part of defining your game's concept. If the player's role is difficult to describe, it might be difficult for the player to grasp as well, and it might indicate that there is a conceptual problem with the game. This doesn't mean that the role always has to be simple. In *Madden NFL Football*, for example, the player can be either a football player, a coach, or both. As the coach, he calls the plays that his team will run and makes player substitutions. As a player, he takes the snap, passes the ball, and runs with it. In fact, he's not controlling any single player; he's controlling whichever one happens to be in possession of the ball. On defense, he can switch from player to player to control whichever one has the best chance of tackling the ball carrier.

This works very well in a football game, where these roles, although different from one another in real life, are well understood by the game's audience and are exactly the ones they want to perform. But if a game takes place in a less familiar world with less familiar objectives, it's important for the roles to be clear. If the player's role changes from time to time—especially involuntarily—it's essential that the player know why it changed and be able to adapt quickly to the new circumstances.

The player's role also helps the player to understand what he's trying to achieve and what rules he's playing by. In Sierra Online's *Police Quest* series, for example, the player takes on the role of a real police officer. Real police officers can't just shoot anything that moves; they have to obey strict rules about when and how to use their guns. Tactical combat simulations such as *Rainbow Six* and *Counter-Strike* also implement these sorts of rules, placing the player in the shoes of a real Special Forces soldier. By telling the player what role he'll be playing, the player knows that his actions will have to be more cautious than in the usual frenetic shooter.

Modes and Structure

Some games, such as checkers, work the same way from beginning to end. The player is always trying to accomplish the same thing in the same way. Other games have distinct *modes*, in which the nature of the gameplay changes significantly from one mode to the next. War games, for example, might have a strategic mode in which you plan which battles you intend to fight and a tactical mode in which you actually fight the battles. Many things often change when the game switches modes: the interaction model, the perspective, the player's role, and some of the challenges.

In addition, many games have noninteractive modes that are interspersed with the interactive ones. War games, again, often have a briefing that occurs before the battle to explain to you what resources you have and what your objectives are, and a debriefing that occurs afterward to summarize what happened and tell you whether you won or lost.

The relationships between the modes and the rules determining when and why the game switches among them collectively form the *structure* of the game. The best way to document the structure of a game is with a *flowchart*, a series of boxes representing the modes with arrows from one to the next showing how the transitions occur. At the concept stage, this doesn't have to be detailed, but if you're designing a game with multiple modes, you should have at least a general idea of what modes you want and why.

Realism

When a game depicts a world, even an imaginary one, in such a way that the principles of real-world logic and common sense apply, we say that the game is *realistic*. *Microsoft Flight Simulator*, which tries to model accurately the behavior of real aircraft, is a good example. On the other hand, when a game has quite arbitrary rules and you cannot count on real-world common sense, as in *Pac-Man*, for example, we say that the game is *abstract*. We discuss realism in more detail in [Chapter 3](#).

At the concept stage, you don't have to decide exactly how realistic your game is going to be. For example, in designing a war game, you might defer decisions about whether to allow friendly fire to harm your own troops. In real life, the military goes to great lengths to prevent friendly fire incidents, but tragically, they still do happen. In a war game, in which the player rarely has precise control over how each weapon is aimed, it might be placing too much of a burden on the player to handle the friendly fire problem. On the other hand, this might be an issue that you want to focus on. Play-testing can tell you whether it's manageable, but an experienced designer can usually predict gameplay problems before the game gets that far. At the concept stage, it's not that important to know whether your simulation will include the effects of friendly fire. It is useful, however, to have a general idea of whether your game is going to be abstract or realistic.

A Word About Story

Computer games lie in a peculiar middle ground between the purely passive, narrative media such as film and television and active, non-narrative games such as poker or dominoes. Some computer games, such as *Tetris*, have no narrative; others, such as the *Gabriel Knight* series, have a great deal. Some games tell linear, noninteractive stories interspersed with interactive episodes, others implement storylines that branch as the player makes decisions, and still others put their players in a setting where storylike adventures can take place and let them have at it. The relationship between the game and the story has been debated many times in the game industry.

It is our contention that, whether a game contains a great deal of narrative content or none at all, the player must ultimately live his own story through playing the game. The fundamental principle of computer gaming, its *raison d'être*, is interactivity: providing the player with something entertaining to *do*. It is not the business of the game designer to tell stories, but to create worlds in which stories take place around an active player.

A common error made by beginning game designers is to concern themselves with the story too early in the design process. The first concern of the designer is not "What am I going to show or tell my player?" but "What is my player going to do?". When conceptualizing your game, you don't have to know exactly what narrative content you want to include in your game. If you are working on it now, you are focusing your attention in the wrong place. All you need to know is whether you want a story and, if so, what its overall direction will be. You should be able to summarize it in a sentence or two, for example: "Jack Jones, leader of a secret DEA task force, will conduct a series of raids against the drug barons, ending in an apocalyptic battle in the cocaine fields of Colombia. Along the way, some of the people he encounters will not be quite what they seem." Remember that errors in the storyline are much easier to correct than errors in the gameplay, and gamers will forgive them more easily as well. Make sure you understand your game first; then build your story into it.

We discuss storytelling and narrative at length in [Chapter 4](#), "Storytelling and Narrative."

[< PREVIOUS](#)
[NEXT >](#)

Understanding Your Audience

In the previous section, we defined the elements of a game. But games don't exist in a vacuum; they're intended to be played by people. A common misconception among game designers is that all players enjoy the same things that the designer enjoys, so the designer only has to examine his own experience to know how to make a game entertaining. This is dangerous hubris. The reason for making a computer game is to entertain an audience. You have to think about who those people are and what they like.

Generalities are always risky, and misguided assumptions about players can result in games that no one will buy. Still, unless the game you're designing has been commissioned by a single individual, you are making it for a class of people, not for one person, and that class will be defined by common characteristics. Another important question a publisher will ask you is "Who will buy this game?". Think carefully about the answer. What things do they have in common? What things set them apart from other gamers? What challenges do they enjoy? More important, what challenges do they *not* enjoy? What interests them, bores them, frustrates them, excites them, frightens them, and offends them? Answer these questions, and keep the answers close at hand as you design your game.

Core Versus Casual

In our opinion, the most significant distinction among player types is not between console-game players and computer-game players, not between men and women, and not even between children and adults. The most significant distinction is between *core* gamers and *casual* gamers.

Core gamers play a lot of games. Games are more than light entertainment to them; they're a hobby that demands time and money. Core gamers subscribe to game magazines, chat on game bulletin boards, and build fan web sites about their favorite games. Above all, core gamers play for the exhilaration of defeating the game. They tolerate frustration well because of the charge they get out of finally winning. The greater the obstacle, the greater the sense of achievement. Core gamers thrive on competition. They don't like games that are easy; they like games that are challenging.

By comparison, *casual gamers* play for the sheer enjoyment of playing the game. If the game stops being enjoyable or becomes frustrating, the casual gamer will stop playing. For the casual gamer, playing a game must be entertaining, whether it's competitive or not. A casual gamer is simply not willing to spend hours learning complex controls or getting killed again and again until he finds the one weak point in an otherwise invincible enemy. To design a game for casual gamers, you have to challenge their minds at least as much as their motor skills.

In reality, of course, there are as many types of gamer as there are gamers; everyone has their own reasons for playing computer games. But the casual/core distinction is a very powerful one. If you design a game specifically for one group, you almost certainly won't have a lot of sales to the other group. A few very well-designed games manage to appeal to both: *Goldeneye*, for example, could be played happily by both core and casual gamers. Core gamers could set the game at the highest difficulty level and drive themselves crazy trying to cut 15 seconds off the last time it took to play a mission. Casual gamers could set the game at the easiest level and blast away, enjoying the game's smooth controls and visual detail.

Even at the concept level, you must have some understanding of who will play your game and what they will enjoy about it. A game concept is not complete without a statement describing its intended audience.

The Genres of Interactive Entertainment

As we said in the introduction, each genre of interactive entertainment displays a common pattern of challenges. In later chapters, we'll look at these genres in detail, examining each to see what can be learned from it:

- *Action games* normally include physical challenges, puzzles, races, and a variety of conflict challenges, mostly at the personal level. They can also contain simple economic challenges, usually involving collecting objects. They seldom include strategic or conceptual challenges.
- *Strategy games* tend to include strategic (naturally), tactical, and logistical challenges, in addition to the occasional economic ones. Once in a while, they have a personal conflict challenge thrown in for spice, but this often annoys strategically minded players.
- Most *role-playing games* involve tactical, logistical, and exploration challenges. They also include economic challenges because the games usually involve collecting loot and trading it in for better weapons. They sometimes include puzzles and conceptual challenges, but rarely physical ones.
- *Real-world simulations* include sports games and vehicle simulations, including military vehicles. They involve mostly physical and tactical challenges, but not exploration, economic, or conceptual ones.
- *Construction and management games* such as *Roller Coaster Tycoon* are primarily about economic and conceptual challenges. Only rarely do they involve conflict or exploration, and they almost never include physical challenges.
- *Adventure games* are chiefly about exploration and puzzle-solving. They sometimes contain conceptual challenges as well. These may include a physical challenge also, but only rarely.
- *Puzzle games* tend to be variations on a theme of some kind. *Sokoban* is about moving blocks around in a constricted space; *The Incredible Machine* is about building Rube Goldberg contraptions to accomplish particular tasks. The challenges are almost entirely logical, although occasionally there's time pressure or an action element.

Some games cross genres for some reason, combining elements that are not typically found together. The adventure game *Heart of China*, for example, included a small 3D tank simulator at one point. This is occasionally a design compromise between two people on the team who want the game to go in different directions. It's also sometimes an effort to appeal to a larger audience by including elements that both will like.

Although it can add flavor and interest to a game, crossing genres is a risky move. Rather than appealing to two groups, you might end up appealing to neither. Many players (and game reviewers) prefer particular genres and don't want to be confronted by challenges of a kind that they normally avoid. The wholesale buyers, who are planning to purchase a certain number of games from each genre for their stores, might not know which pigeonhole to put the game into and might shy away from it entirely.

However, you should not allow these genre descriptions to circumscribe your creativity—especially at the concept stage. If you have a wholly new, never-before-seen type of game in mind, design it as you see it in your vision; don't try to shoehorn it into a genre for the wrong reason. A game needs to be true to itself. But don't mix up genres purely for its own sake. A game should cross genres only if it genuinely needs to as part of the gameplay. A flight simulator with a logic puzzle in the middle of it, just to be different from other flight simulators, will only annoy flight sim fans.

The Types of Game Machines

Most game concepts should be fairly independent of the target platform. If you start thinking too early about the machine and its capabilities, you run the risk of designing a "[technology-driven game](#)," a notion that is discussed a little later in the chapter. Still, some genres of game are better suited to one kind of machine than another. It's valuable to know the strengths and weaknesses of the different types of machines and, even more important, how they are used by their owners.

Home Game Consoles

A *home game console* is usually set up in the living room or a child's bedroom. The player sits on a couch holding a dedicated controller in both hands, 3 to 6 feet away from a relatively low-resolution display, the television. This means that games designed for the home console machine cannot be as intricate as the typical PC game. The graphics have to be simpler and bolder, and the control method and user interface must be manageable with the provided controller. The kind of precision pointing that's possible with a mouse is much more difficult with most controllers, even those with analog joysticks. However, you are guaranteed that every machine will ship with a standardized controller; you don't have to cope with the huge variety of controllers and joysticks that are available for the PC.

Because the television is designed to be seen by several people at once, and because the console usually allows for at least two controllers, console machines are excellent for multiplayer games in which all the players look at the same screen. This means that every player can see what every other player is doing on the screen, which is a consideration in the design of some games. On the other hand, until recently, home consoles had no hard disk drives, so there was little space in which to store data between games. Games designed for consoles weren't very customizable and couldn't save complex states.

Home consoles tend to have very powerful graphics-display chips but slower central processing units and less RAM than personal computers. Because they sell for \$200–\$300, the manufacturer has to cut the hardware design to the bone to keep the cost down. This means that as computing devices, they are less powerful than personal computers and more difficult to program. On the other hand, their low price means that there are far more of them around, and a larger market for their games.

Personal Computers

A *personal computer* is usually set up away from the communal living space, on a computer desk. In this case, the player has a keyboard, a mouse, possibly a joystick, and (more rarely) a dedicated game controller like those on console machines. The player sits 12 to 18 inches away from a relatively small (compared to the television) high-resolution display. The high resolution means that the game can have subtle, detailed graphics. The mouse allows precision pointing and a more complex user interface. The keyboard enables the player to enter text conveniently and send messages to other players over a network, something that is nearly impossible with console machines.

The personal computer is quite awkward for more than one person to use. The controls of a PC are all designed for one individual, and even the furniture it usually sits on—a desk—is intended for a solitary use. PC games are rarely designed for more than one person to play on a single machine. On the other hand, a PC is very likely to be connected to the Internet, while consoles are just now beginning to get this capability. The PC is still the machine of choice for multiplayer networked games, but this could change soon.

The great boon of PC development is that anyone can program one; you don't have to get a license from the manufacturer or buy an expensive development station. Consequently, PCs are at the cutting edge of innovation in computer gaming. They're the platform of choice for small-scale, low-demand projects; interactive art; and other experimental forms of interactive entertainment.

The great bane of PC development is that no two machines are alike; because they're customizable, there are millions of possible configurations. In the early days of the game industry, this was a real nightmare for programmers. Fortunately, the Windows and Macintosh operating systems have solved many of these problems by isolating the programs from the hardware. Still, games tend to "push" the machine a lot harder than other applications, and configuration conflicts still occur.

Handheld Game Machines

Handheld game machines are a hugely popular and very inexpensive form of entertainment, mainly used by children. A given model is absolutely standard; there's no room for customization at all. These machines normally have a very limited number of controls and a very small LCD screen. They have little or no capacity to store data between games. Their CPUs are weak and slow by modern standards.

Many cheap handheld machines offer a fixed set of games that are built in, but the more versatile ones, such as the Game Boy Advance, accept games stored on ROM cartridges. Cartridges store far, far less data than the CD-ROMs or DVD discs that home consoles and computers use. Designing for a cartridge machine places severe limits on the amount of video, audio, graphics, and animation that you can include in the game. Because they're solid-state electronics, though, the data on a cartridge is available instantly. There's no delay for it to load the way there is with optical media devices.

The handheld game market is very lucrative, but creating a game for one will severely test your skills as a designer. With no room for fancy graphics or movies, you must rely on pure gameplay alone to provide the entertainment.

Other Devices

Games are showing up on all sorts of other devices these days. The more specialized the device is, the more important it is to have a clear understanding of its technical limitations and its audience. Text-based messaging on cellular telephones might breathe new life into a niche genre, the text MUD (multiuser dungeon or domain). Airlines are starting to build video games into their seats. Personal digital assistants (PDAs) are a great new platform for small, simple games. Video gambling machines are an entire industry unto themselves, one in which the random-number algorithms you use are closely monitored by state regulators. And, of course, there are arcade machines. Arcade games are subject to strange design limitations not seen on other devices. They have to maximize what the operators call "coin drop"—the amount of money that people put in through the front. Arcade operators care little for richness, depth, and the aesthetic qualities of a game as long as it makes a lot of money for them. This requires some fine balancing. If a game is too hard, people will abandon it in disgust, but if it is too easy, they will be able to play for a long time without putting any more money in.

Because these devices occupy niche markets with peculiar restrictions, we won't be addressing them in detail. This is a book about game design in general, so we concentrate on games for all-purpose game machines: home consoles and personal computers.

[< PREVIOUS](#)[NEXT >](#)

Motivations That Influence Design

Why build a computer game? The answer probably seems obvious: because it's a fun thing to do, to sell the game for money, or both. That's why most people get into the game industry—and probably why you are reading this book. But, in fact, there are a variety of reasons for building a game, and motivation has a powerful effect on the way a game is designed. Sometimes a game is built for several different reasons at once, and they're not always compatible with one another. Different motivations tend to pull the design of the game in different directions, which requires someone (usually the designer or the producer) to make a decision about which reason is most important.

Market-Driven Games

Most of the computer games in the world are built for sale. The publishing company wants to sell as many copies as it can. However, no one knows exactly what makes a game a hit, and unexpected hits such as *The Sims* continue to prove a lot of the conventional wisdom dead wrong. Still, certain genres and elements of games are generally thought to be popular with particular markets. Scantly clad women, big guns, and spectacular explosions, for example, are considered popular with teenage boys. Games with interesting characters, rich plots, and clever puzzles are thought to be popular with girls and women. These are, of course, stereotypes, but they are commonly held in the game industry. When a company chooses to build a game specifically for a particular market and to include certain elements in its design specifically to increase sales within that market, that game is said to be *market-driven*.

You might think that if publishers want to maximize sales, any game made for sale should be market-driven. Experience shows, however, that most market-driven games aren't very good. They do make a fair amount of money, but seldom enough to qualify as blockbuster hits. For one thing, the best games are expressions of the designer's vision and carry the designer's personal stamp, which makes them stand out from other games. Market-driven games do not reflect anyone's vision; instead, they reflect what the publishing company thinks the market wants. "Everybody knows" that boys like big explosions, so the publisher insists that the game include big explosions, whether they're really part of the designer's vision or not. As a result, market-driven games all tend to look alike because they're all designed according to the same stereotypes about the market. This results in a row of clone games on the store shelves. Because there isn't any particular reason to choose one over another, none of them becomes a blockbuster hit.

Another reason most market-driven games aren't very good is that they lack harmony, a concept that we discuss in more detail in [Chapter 3](#). You can't make a brilliant game by simply throwing in all the popular elements you can think of. If you try, you get a game that doesn't feel as if it's about anything in particular. It doesn't hang together; it's merely a collection of pieces. Games don't sell only because they contain the "right" pieces; they also sell because the pieces fit together synergistically to make a coherent whole.

Finally, because market-driven games are designed to appeal to a stereotypical kind of gamer, they won't appeal to anyone who doesn't match the stereotype. If you make a game that is explicitly intended for teenage boys, you will not only turn off most girls who might otherwise like to play your game, but you'll also turn off any teenage boys who don't fit your stereotype. And if your stereotype happens to be wrong, you could end up making a game that appeals to nobody at all.

As a designer, you will, of course, make decisions with your players in mind. You will try to include elements that they will enjoy and avoid elements that they won't. That's entirely appropriate, and generally it's what your publisher wants you to do. But those elements must be consistent with your own vision of the game. If you include extraneous elements purely to boost sales but that don't really feel as if they belong there, you could end up doing your game more harm than good.

Designer-Driven Games

The opposite of a market-driven game is a *designer-driven* game. In designer-driven games, the designer retains all creative control and takes a personal role in every creative decision, no matter how small. Usually he does this because he's absolutely certain that his own creative instincts are the only right ones for the game and that his vision must rule supreme.

Relatively few games are designed this way. Few major publishers will grant a designer this much power unless he can claim a strong track record; there's too much money at stake to allow a single person to dictate everything. In practice, the design of most games is a collaborative process. The design includes not only the work of the lead designer, but also the input of others on the team. Producers, level designers, programmers, artists, and sometimes testers play a role in shaping the game. On the other hand, a good many small self-published games are designer-driven. Designer-driven games tend to be rather idiosyncratic. This might be harmful or it might be helpful; it depends on whether the designer's instincts are good ones.

The main problem with designer-driven games is that they're usually designed according to the designer's own notions of what constitutes a fun game, regardless of what play-testing shows. The designer invests too much of his own ego in his creation to allow other people to change it, even if changes would be an improvement. Also, when the designer insists on personal control over every decision, he often becomes a bottleneck in the development process. Other people on the team end up waiting around for the designer to make decisions for them, which wastes time and money.

A very small number of game designers—Will Wright and Sid Meier among them—have years of experience and proven track records, and can sell games on the strength of their names alone. They frequently turn out blockbusters even if the marketing department doesn't like or understand the idea. But even they are usually good at delegating. Micromanaging control freaks seldom make great designers. They're too busy cultivating one tree while the rest of the forest dies around them.

License Exploitation

Another reason that people make games is to exploit a particular intellectual property, a license. These games, often tie-ins with movies or books that have a highly recognizable brand name, can be enormously lucrative. Which do you think will sell better, a game about a suave, well-dressed British spy, or a game about James Bond? A game about a brilliant young star pilot, or a game about Luke Skywalker?

Working on a licensed game can be a lot of fun. As a designer, you will get to work creatively with characters and a world that you might already know and particularly like, and you'll be making a contribution to the canon of materials in that world. You might even get to meet some of the famous writers, directors, or actors who have brought that world to life in other media. When you say, "I designed a *Star Wars* game," people will be a lot more impressed than if you say, "I designed a game about a brilliant young star pilot."

One downside of designing licensed games is that you don't have as much creative freedom as you do designing a game entirely from your own imagination. You have to use the characters and settings provided by the license—but, more important, you have to conform to certain rules laid down by the original owners. They're usually very anxious to make sure their intellectual property isn't used in ways that conflict with their own marketing strategy or that would present their property in a negative light. The owners will almost certainly insist on the right to approve your game before it ships and to demand that you change things they don't like. If you had a license from the Walt Disney Corporation to make a Winnie the Pooh game, for example, Disney would never allow you to give Winnie the Pooh a machine gun. It would be a dumb thing to do, but it would also completely violate Disney's notions of what Winnie the Pooh is about—not to mention all the expectations buyers have when they see Winnie the Pooh on the box. Some intellectual property owners are lenient and don't really care what you do; most, however, are extremely strict and specify details right down to the precise vocabulary that you may use.

Licensed sports games have their own peculiarities. Usually, such games will have a license from the major league of the sport they are simulating—for example, Major League Baseball or the National Football League. These licenses will allow you to use the names and logos of teams in the league; they will also require you to display them properly and in the correct colors. The personal publicity rights of the players further complicate matters; some leagues do not have the right to license the players' names and images. We discuss these issues in more detail in [Chapter 12](#), "Sports Games."

A great license doesn't guarantee success, however. A bad game with a famous license is still a bad game. The word will get around, and the players won't buy it. The most infamous example of this is the ill-fated *E.T.* game produced for the Atari 2600 machine during the summer of 1982. Atari manufactured millions and millions of cartridges in anticipation of huge sales, intending to take advantage of the success of the movie *E.T. The Extraterrestrial*. Unfortunately, it was a poor game, few people bought it, and, in the end Atari—which had paid an unheard-of \$22 million for the license—ended up burying most of the cartridges in a landfill to take a tax write-off.

In short, designing a game with a license is a somewhat different experience from designing a game from scratch. You'll have both the original owners of the intellectual property and your own marketing department looking closely over your shoulder. Your publisher will probably put a lot of money into development and, having spent a lot to get the license, will want to make sure the game is good enough to earn it back. Of course, this means there's more at risk. If you do well and the brand is a popular one, your company stands to make a fortune. If you do a poor job and turn out a bad game, the owner of the license might refuse to renew it and will give it to some other developer instead. You will have cost your company millions in potential future revenue and put that money in your competitor's pockets.

Technology-Driven Games

A *technology-driven* game is one that is designed to show off a particular technological achievement, most often something to do with graphics. The original *Quake* was a technology-driven game. There wasn't much game there, but it helped to sell the *Quake* game engine to other developers. Sometimes the achievement is a piece of hardware instead of an algorithm. Console manufacturers often write technology-driven games when they release a new platform, to show everyone the features of their hardware.

Technology-driven games tend to sell well to hard-core gamers because those gamers are often technology-oriented themselves. Casual gamers can still recognize a big jump in image quality even though they're less impressed by performance statistics, and that will encourage them to buy a technology-driven game, too.

The main risk in designing a technology-driven game is that you'll spend too much time concentrating on the technology and not enough on making sure your game is really enjoyable. As with a hot license, a hot technology alone is not enough to guarantee a hit. If you're the first to market with the new technique and it really is spectacular, you could well have a meteoric success in the first few months. To last longer than that, your game has to be really fun to play as well. After the novelty of the technology has worn off, the gameplay will continue to sell the game.

Art-Driven Games

Art-driven games are comparatively rare. Just as a technology-driven game exists to show off a technical feature or achievement, an art-driven game exists to show off someone's artwork. These games are often designed by artists who have a strong visual sense but are new to the game industry. Although such games are visually innovative, they're seldom very good because the designer has spent more time thinking about ways to show off his artwork than about the player's experience of the game. If you want to design games to showcase your artwork, you must be aware that you need enjoyable gameplay as well as great visuals. *Myst* is an example of a game that got this right; it is an art-driven game with strong gameplay.

Entertainment and Integration

As we have shown, when a particular factor drives the development of a game, the result is often a substandard product. A good designer seeks not to optimize one characteristic at the expense of others, but to integrate them all in support of a higher goal: entertainment.

- A game must sell well, so the designer must consider the audience's preferences.
- A game must present an imaginative, coherent experience, so the designer must have a vision.
- A game with a license must pay back the license's cost, so the designer must understand what benefits it brings and exploit them to his best advantage.
- A game must offer an intelligent challenge and a smooth, seamless experience, so the designer must understand the technology.
- A game must be attractive, so the designer must think about the aesthetic style.

In designing an entertaining game, every element and every feature are tested against the standard: Does this contribute to the player's enjoyment? Does it entertain him? If so, it stays; if not, it should be looked at closely. There are reasons for including features that don't directly entertain: They might be necessary to make other parts of the game work or might be required by the licensor. But you should regard them with great suspicion and do your best to minimize their impact on the player.

Game Concept Worksheet

To turn your game idea into a fully fledged game concept, you need to think about and answer for yourself the following questions. You don't have to be precise or detailed, but you should have a general answer for all of them.

1. What is the nature of the gameplay? That is, what challenges will the player face? What actions will the player take to overcome them?
2. What is the victory condition for the game, if any? What is the player trying to achieve?
3. What is the player's role? Is the player pretending to be someone or something, and if so, what? How does the player's role help to define the gameplay?
4. What is the game's setting? Where does it take place?
5. What is the player's interaction model? Omnipresent? Through an avatar? Something else? Some combination?
6. What is the game's primary perspective? How will the player view the game's world on the screen? Will there be more than one perspective?
7. What is the general structure of the game? What is going on in each mode, and what function does each mode fulfill?
8. Is the game competitive, cooperative, team-based, or single-player? If multiple players are allowed, are they using the same machine with separate controls or different machines over a network?
9. Does the game have a narrative or story as it goes along? Summarize the plot in a sentence or two.
10. Does the game fall into an existing genre? If so, which one?
11. Why would anyone want to play this game? What sort of people would be attracted to this game?

[< PREVIOUS](#)
[NEXT >](#)

[< PREVIOUS](#)[NEXT >](#)

Putting It Together

Now your idea is no longer just an idea. You've devised the essential elements of a game, and you know the answers to the vital questions that a publisher will ask. You have a game concept down on paper, and you're ready to show it to someone else for an opinion. The skeleton is assembled. Now it's time to put meat on the bones.

[< PREVIOUS](#)[NEXT >](#)