◂ PREVIOUS NEXT ▸

Chapter 1. What Is Game Design?

Humans have been devising and playing games for thousands of years. There is hot debate about which existing game is the oldest; some argue for Go (shown in Figure 1.1) and some for the African stone game Awari, but the first game ever devised is almost certainly lost to us today. The field of game design is as ancient as the human neocortex and is clearly related to our capacity to pretend, for pretending—that is, creating and playing in an artificial world—is at the heart of all games. Some, such as Go and chess, were perfected so long ago that their rules have not changed for centuries. Other games come on the market, enjoy a brief popularity, and fade away again. People are always interested in new games, so there is a constant demand for new game designs.

Figure 1.1. Go.

Game design is the process of:

- *Imagining* a game.
- *Defining* the way it works.
- *Describing* the elements that make up the game (conceptual, functional, artistic, and others).
- *Transmitting* that information to the team that will build the game.

A game designer's job includes all of these. In this chapter, we discuss what's involved in game design, why we do it, and what it takes to be a game designer.

In spite of the long history of game design, surprisingly little study has been devoted to the subject. Indeed, game design has been taken seriously as a subject worthy of academic interest only in the last few decades. But study requires funding; consequently, most game theory research has been directed toward economic principles and military applications. Little of this work applies to games intended for the consumer, and most of it requires an advanced knowledge of statistics to decipher. We concentrate on practical game design rather than formal game theory.

◂ PREVIOUS NEXT ▸

◀ PREVIOUS NEXT ▶

Art, Science, or Craft?

Many people consider game design to be an art, drawing on a mysterious wellspring of creativity possessed only by a talented few. They think of the "big names" in the game industry—Peter Molyneux, Brian Moriarty, Roberta Williams—as artists, and they admire the vision and originality that such people bring to their games. They imagine that game designers spend their time indulging in flights of imagination, and they ignore or are not even aware of the long and painstaking work that real design requires.

Other people who are more mathematically oriented see game design as a science. They concentrate on the methodology for determining the best rules of play, the intricate procedure of balancing a complex game. They think about equations, relationships, and the flow of resources. Game design to these people is a set of techniques, a process of thought.

We believe both of these views are wrong, or at least incomplete. Game design is not purely an art because it is not primarily a means of aesthetic expression. "Artistry" comes into envisioning the initial concepts and ideas, but once that is done, it's time for the real work of defining and refining how the game will function. Nor is game design purely a science; it doesn't posit hypotheses or seek truth. It's not bound by rigorous standards of logic or formal methods. The goal of a game is to entertain through play, and designing a game requires both creativity and careful planning.

Interactive entertainment is an art form, but like film, television, and theater, it is a collaborative art form, with no single person entitled to call himself the artist. In fact, most designers don't think of themselves as artists at all. Designing games is a *craft*, like cinematography or costume design in Hollywood. A game contains both artistic and functional elements: It must be aesthetically pleasing, but it also must work well and be enjoyable to play. The greatest games combine these attributes brilliantly, achieving a quality for which the only word is *elegance.* Elegance is the sign of craftsmanship of the highest order.

◀ PREVIOUS NEXT ▶

◀ PREVIOUS NEXT ▶

The Anatomy of Game Design

Game design cannot be reduced to a set of discrete instructions and processes. There is no formula that you can follow and produce a perfect game design, ready for your programming team to code into existence. However, you can take advantage of a set of common principles that apply to *all* successful games, and doing so can save you a fair amount of grunt work. Designing a successful game is not just random chance—although so many other factors besides design influence a game's success that it sometimes seems that way.

The Importance of Game Design

Game design (at least for the computer and console) is a very young field, and there is still much to be discovered. The movie industry and even the advertising industry know more about invoking atmosphere and mood than any game designer out there—and more important, they know how to apply their techniques effectively.

Take McDonalds, for example: Have you ever wondered why they use a predominantly red-and-yellow color scheme? Psychologists report that the color yellow influences the perceived hunger level of the subject, while red increases anxiety and the need to hurry. The result, according to the psychologists, is that you order more food, eat it quickly, and leave. The reasons for this appear to be deep in our ancestry. Red is the color of blood, signifying danger, and yellow is a predominant color in foodstuffs. In his book *ManWatching*, Desmond Morris goes into this and many other aspects of human behavior in a lot more detail. It makes for interesting reading, and we recommend it to any serious game designer. It's useful for the game designer to consider the ways that humans interact with each other and the subconscious stimuli that influence them. This and other kinds of understanding will take game design to a new level, resulting in richer, subtler, and ultimately better games.

Even though some people view game design as a mystical art, they can be peculiarly inconsistent in their attitudes: Nearly everybody who expresses an interest in the subject believes that he could be a game designer. The skill of game design is, so to speak, invisible. It looks as if anyone should be able to do it. After all, it's just a matter of writing a simple story, knocking up a design document or two, and telling the programmers what you want, right? No. That's about as realistic as expecting that anyone who can use a hammer and a saw should be able to produce a violin. Good game design is, as we said, a matter of craftsmanship.

If you want proof, just check any magazine that reviews games and see how often it marks down games for poor design. Games might be technically superb and look wonderful, but if the gameplay isn't there, it's not a game; it's a pretty demo. Even the original *Quake*, while lauded for its amazing technology, was slammed for the poor quality of its single-player gameplay (after the stunned awe had worn off). Sure, it looked great, but running around shooting poorly designed and badly placed enemies wore thin very quickly.

The creators of *Quake*, id software, acknowledged that the first *Quake* was just a technology demo. The gameplay was improved in *Quake II*, which presented a good single-player game, but with *Quake III*, we're back to the technology demo again. It was left to third parties (such as Valve, which developed *Half-Life* using the *Quake II* engine) to take the technology and make a game with it.

Given the amount of money routinely sunk into these technological powerhouse products, it seems amazing that more money is not spent on producing a decent game design. In many cases, the game design is an amalgamation of the "best" ideas of the development team. This works so rarely that when it *is* successful, the process is widely publicized and the publicity gives the impression that this is the best way to design a game. You might have heard the saying a camel is a horse designed by committee. Nowhere is this more applicable than the game industry.

Of course, for every rule, you can find a high-profile exception (otherwise, life would be a lot simpler—and duller). The designers of *Half-Life*, a truly excellent game, used a process like this, dubbed the *cabal process*. Valve was incredibly fortunate to have the right people in the right place at the right time. Most developers can't count on such luck. Valve can't even be certain that the cabal process will work again. History is on their side, but as they say in the financial industry, past performance is no guarantee of future success.

So what's the upshot of this? Although game design is a creative process requiring the ability to dream and imagine amazing worlds populated by strange and wonderful denizens, a great many practical principles also can be extracted and analyzed. When you thoroughly understand the techniques of game design, your imagination and intellect will be free to work together and concentrate on what's really important: great gameplay. Of course, in an ideal world, original and innovative gameplay would be equally important. Unfortunately, most publishers don't want games that are *too* innovative; they want something just like a recent hit, with a few more twists.

Seeking the Key Elements of Games

Our approach in teaching you how to design games centers on the idea that games are made up of certain key elements, and that the games in a given genre tend to have many of those elements in common. These elements include such things as the rules of the game, the player's role (pilot, athlete, general, spy, and so on), the challenges the player will face, and many others that we will introduce as we go along.

This doesn't mean that we think all games in a particular genre should be alike—far from it. Among cars, for example, all minivans (a genre of car, you might say) include seating for five or more people, have a rear door rather than a trunk, and stand up fairly high off the ground. Yet not all minivans look alike or have the same performance characteristics. The same is true of games. Two war games can include many elements in common and still have completely different settings, units, and strategies, and be balanced in different ways.

We encourage you to adjust our elements as you see fit. The last thing we want to see is more games with different graphics but identical play mechanics; there are too many of those already. During the heyday of the Sega Genesis and the Super Nintendo, we saw too many side-scrollers, games about running and jumping on platforms; nowadays store shelves overflow with too many first-person shooters in which players run through a 3D world firing at anything that moves. The game industry and our players don't need more games that look a little different but are essentially the same. That leads to creative stagnation among developers and eventual boredom and disinterest among customers.

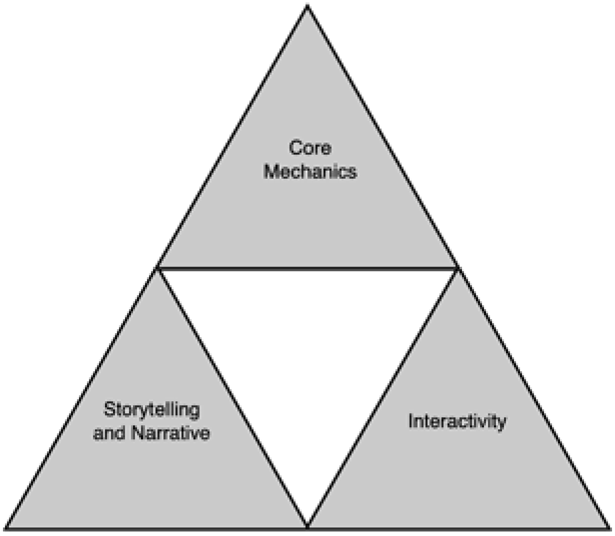
The purpose of identifying common elements in a genre is not to encourage the development of cookie-cutter games, but to make sure that when you design a game, you have covered all the basics—the essential components that a game in a particular genre should have and without which is incomplete.

Okay, that's far enough. Before we continue, we need to discuss exactly what we mean by game design. After all, one of the most confusing aspects of the study of game design is that there's no official definition agreed on by the whole industry—different game designers might have different ideas of what comprises game design.

Laying Down the Ground Rules

For the purposes of this book, we have broken down game design into three specific areas: core mechanics, storytelling, and interactivity. Each is a distinct, complementary element of a game, and each makes up one part of a larger whole, as shown in Figure 1.2.

Figure 1.2. Core mechanics, interactivity, and storytelling.



Core Mechanics

The rules that define the operation of the game world make up the *core mechanics* of the game, or the foundations of gameplay.

The core mechanics are the translation of the designer's vision into a consistent set of rules that can be interpreted by a computer—or, more accurately, rules that can be interpreted by the people who write the software that is interpreted by the computer. Defining the core mechanics is the "science" part of game design. It is important not to confuse this with the technology of computer gaming. Although the core mechanics are implemented by creation of a mathematical (or computational) model of the abstract game world, the core mechanics describe the way the game works, not the way the software operates. In a noncomputerized game, we would simply call these the rules, but the rules of computer games are far more complex than the rules of any board or card game.

This is the heart and soul of the game—if the core mechanics aren't sound, you end up with a poor game. Unfortunately, this happens fairly frequently. It seems that the core mechanics are often the least-developed area in modern games. The reasons for this are many and varied, but the three most significant are as follows:

- Designer ignorance of game mechanics
- Marketing pressure
- Conflicting demands for impressive technology and good gameplay

Of these three, we can help with only the first and the third. Designer ignorance is what this book is intended to address. The second, marketing pressure, is beyond our control, much to our despair on a number of occasions in the past. We will also address the third, technology versus gameplay, in this book, although it's also partly brought on by market conditions. The technology race is analogous to the Cold War arms race: Technology produces eye candy, eye candy grabs customers' attention, and, in the viciously competitive arena of the software store and magazine page, attention is the most valuable commodity. Games have limited development budgets, and the money tends to go toward areas that will produce the most visible results. If technology sells products, then products will focus on technology. The core mechanics are often given short shrift in consequence.

Storytelling and Narrative

All games tell a story. The complexity and depth of that story depends on the game. At one extreme, in adventure games such as *Grim Fandango*, the game *is* the story. At the other extreme, it's the player who tells the story by the act of playing. Even *Tetris* has a story—a story created by the player as she plays.

Narrative, as we use the term, means that part of the story that is told by you, the author and designer, to the player. Narrative is the noninteractive, presentational part of the story. *Tetris* has a story, but it contains no narrative. Because playing games is an active process and listening to a narrative is a passive one, there is an inherent tension between interactivity and narrative. We discuss this tension in much more detail later.

Without a story, or some way for a player to implicitly form his own story, the game simply will not interest the player. As you probably remember from high school English classes, all stories feature *dramatic tension:* an unresolved issue, problem, or conflict that keeps the reader's attention and makes him want to read on. This is no less true of games than it is of novels or short stories. In the case of games, the dramatic tension often, though not always, arises from a challenge in the game that the player has not yet overcome. Whether the player makes up his own "story" or whether he reads or watches a scripted narrative, it's the primary hook that keeps the player playing. Many games try to aim for a middle ground: They provide a back story and let the player fill in the details—quite often by completing missions that drive the plot along.

Narratives in games are often quite linear, unaffected by the player's actions and unchanging from one playing to the next. Many designers see this as a limitation, impeding the freedom of the player. In recent years, there has been a lot of discussion about nonlinearity in computer games and the improvements it would bring. As with many "new and improved" features of games, a certain amount of smoke and mirrors is involved in bringing nonlinearity to nontrivial narrative-based games. Automated storytelling is the holy grail of computer game design. Chris Crawford, designer of *Balance of Power*, has performed some work in this area with his *Erasmatazz* project (www.erasmatazz.com).

There is still a long way to go before we have (or can provide) fully dynamic computer-generated stories. In the meantime, we'll have to continue to use the old-fashioned methods and let humans write the stories. Unfortunately, the effort involved in creating a nonlinear story grows exponentially with the number of story threads and typically weakens the story's impact. Hence, games with strong storylines are usually fairly linear. In fact, some genres suffer from *too* much scripting. Examples are flight simulators and strategy games that attempt to give the player an illusion of freedom beyond the bounds set by the scripting and consequently fail when the player does something unexpected that breaks the linearity.

A linear story does not have to be a disadvantage. The *Final Fantasy* series of games allows little room for variation from the main story—the only way to complete the game is to allow the story to carry you along, but they are still good games. Certain scripted events that are completely unalterable by the player push the story along—and the main challenge for the player is to fit in with these events, to become part of the story. Although this makes the story feel a bit unrealistic, it is nonetheless convincing enough to enthrall the player. We don't yet have the technology to create engaging and fully self-consistent nonlinear stories. We'll probably start to see real generic story-telling systems at about the same time we can say, "Good morning, computer," and expect it to understand and answer us!

Interactivity

For our purposes, *interactivity* is the way that the player sees, hears, and acts within the game's world—in short, the way the player plays the game. This covers a lot of diverse topics: graphics, sounds, user interface—everything that comes together to present the gaming experience. As a game designer, you will not create all these items yourself, but you will specify how the interactivity will work. It's up to the specialized members of your team to make it happen.

Interactivity has been a buzzword connected with computers and games for some time. It's often overused or used inappropriately. A few years back, some people thought that they could create a new kind of product by prefixing the word *interactive* to just about anything: interactive movies, interactive television, and so on. Don't let that turn you off: Interactivity is not just a buzzword in game design; it's the term used for making the game playable.

Poor interactive design ruins many products. For example, a badly designed user interface severely compromised *Warwind*, an otherwise promising *Warcraft* derivative. *Warwind* required players to negotiate an impossibly nested series of icon-based menus using only right-clicks. We're sure you can come up with your own list of similar games in which you've found critical design flaws such as these.

Interactivity (in the nonbuzz sense of the word) starts at the user interface. The user interface defines the "flavor" of your game. For example, the gameplay of *Mario 64* on the Nintendo 64 console is virtually identical to that of a 2D platform game such as the *Sonic the Hedgehog* games on the Sega Genesis (or Megadrive, to those of you in Europe). You run, jump, collect goodies, and avoid the baddies. The user interface—the buttons you push to play the game—made *Mario 64* stand out from its peers.

Taking this to the extreme, there is nothing in *Mario 64* that couldn't be implemented as a text-based game with type-in commands such as "run left," "jump right," and "take coin." Of course, it would be ridiculous to do this (even if the Nintendo 64 console had a keyboard)—the game would have sold close to zero copies. But the example shows that even though every other aspect of the game might be well designed (as it was in *Mario 64*), a good game requires a good user interface. Derivative games that arrived on the scene after the release of *Mario 64* didn't duplicate *Mario 64*'s excellent user interface and control system. Although they were similar in concept, they didn't reach that golden standard.

A game's graphics are also important parts of the interactive component, although designers disagree about how important they are. Unfortunately, the prevailing commercial pressure compels many designers to concentrate on appearances. In general, far too much time is spent on getting the look of a game right at the expense of tuning its gameplay. A lot of nostalgic gamers have called for a return to the values of the "golden age" of gaming—the 1980s, when hardware limitations forced developers to concentrate on gameplay. Nowadays, however, the capabilities of new machines require that more effort be spent on presentation. The more effort spent on presentation, the more the player expects to see in the next game, resulting in a vicious positive-feedback cycle of player expectations boosting emphasis on presentation, which, in turn, boosts player expectations, and so on.

We've all had the experience of playing an action game really well, getting into a sort of "groove" in which your eyes and hands seem to meld with the machine. The best user interfaces allow you to immerse yourself in the game so deeply that you are no longer aware of the user interface at all—the infamous *Tetris Trance*. That's what well-designed interactivity does for a game.

◂ PREVIOUS NEXT ▸

Documenting the Design

As part of their job, game designers produce a series of documents to tell others about their game design. Exactly what documents they produce and what the documents are for varies from designer to designer and project to project—but they usually follow a common thread.

Why Do We Need Documents?

Beginning programmers, especially those who want to get into the game industry, often make the mistake of thinking up a game and then diving in and starting to program it right away. Programming is an immensely rewarding activity because you get to see the results of your work within seconds, and those programmers are seeking that reward as soon as possible. They can't wait to see at least some portion of their game up on the screen.

Back in the days when a single person designed and wrote an entire computer game, there wasn't really anything wrong with this. The programs were so small that even if the idea changed radically in the course of development, the programmer could modify the code within a few days. The games themselves were simple enough to be described in a page or two, so developers didn't feel any need for a formal design process.

In modern commercial game development, however, this kind of *ad hoc* approach is disastrous. Development teams often consist of 20 to 50 people. Millions of dollars are at stake. Critical release dates must be met to get the game on the shelves at a particular time—usually right before the Christmas shopping season. You can't build a game consisting of hundreds of megabytes of software, artwork, animations, movies, and sound files with a "let's try it and see" approach; a job of this scale calls for some sort of methodology. Different developers require different degrees of formality, but all serious game companies now insist on having some kind of written design before they start work.

As we said before, a key part of game design is transmitting the design to other members of the team. In practice, a lot of that communication takes place not through the documents themselves, but through team meetings, bull sessions, and conversations over lunch. That doesn't mean that there's no point in writing design documents, however. The documents record decisions made and agreed upon orally; they create a paper trail. More important, writing a document is a process of turning vague, unformed ideas into explicit plans. Even if no one reads it at all, an idea written down is a decision made, a conclusion reached. If a feature of a game is *not* described in writing, there's a good chance that it has been overlooked and that someone will have to make it up on the fly—or, worse, that each part of the team will work toward a different goal. It's far easier and cheaper to correct a design error before any code is written or artwork is created. Depending on the size of the game, wise developers will allot anywhere from one to six months for pure design work before starting on development, usually in combination with some throwaway prototype for testing out gameplay ideas.

Idea Versus Design Decision

Here's an idea: "Basilisks should protect their eggs."

Here's a design decision: "Whenever they have eggs in their nests, female basilisks will not move beyond visual range from the nest. If an enemy approaches within 50 meters of the nest, the basilisk will abandon any other activity and return to the nest to defend the eggs. She will not leave as long as a living enemy threatens the eggs, and will even defend the eggs to her own death."

See the difference? This is what creating design documents is about.

The Types of Design Documents

This section is a short introduction to the various types of documents a game designer might be asked to create. In chronological order, they are:

- High concept (2–4 pages)
- Game treatment (10–20 pages)
- Game script (50–200 pages)

The following sections briefly discuss these documents. Appendix A, "Sample Design Documents," contains samples (or pointers to samples) of each one and discusses their contents and formatting in much more detail.

The High Concept

Writing the *high-concept document* is the first step after scribbling down the initial idea. Its aim is to express the fundamental spirit of the game. Just as the purpose of a resumé is to get you a job interview, the purpose of a high-concept document is to get you a hearing from someone, a producer or publishing executive. It puts your key ideas down on paper in a bite-size chunk that he can read in a few minutes. Like a resumé, it should be short—not more than two to four pages long. The high-concept document should take, at most, a week to create, of which four days are spent thinking and one is spent writing.

If possible, try to begin the document with a single, punchy sentence—the high concept itself—that describes the game in a nutshell. Unfortunately, publishing executives have notoriously short attention spans and a great many calls on their time. You need to grab them as quickly as possible. The high concept for *Interstate '76* might have been, "Automotive vigilantes defend America's oil supply with heavily armed 1970s muscle cars in a high-octane, 3D action game."

The high-concept document covers these details:

- The premise of the game
- Its intended audience
- Its genre (if it belongs to one)
- Its unique selling points
- The target platform(s)
- The overall storyline

It must also describe the gameplay—what the player is supposed to do, what type of environment or scenarios he will encounter, and a general overview of the game flow. You might also want to include a description of any special technologies that will be used to build the game and any special hardware it might require.

If you plan to use the high-concept document as part of a sales pitch to a publisher, you might want to include a section containing short profiles of the development team members, with details of relevant past experience. You'll need to explain why they are the right team to build your game. You might or might not also want to include a budget estimate, depending on who will see it and what your relationship is with them.

The high-concept document need not be a sales tool; it's also worthwhile to write one for yourself, just to record an idea that you might want to work on in the future.

The Game Treatment

The purpose of the *game treatment* is to present the game in broad outline to someone who's already interested in it and wants to hear more about it. The treatment is designed both to satisfy initial curiosity and to stimulate real enthusiasm for the game. When you give a presentation about your game to a publisher, you should hand him the game treatment at the end so he'll have something to take away and look at, something that will float around his office and remind him of your game. Your goal at this point is to get funding of some sort, either to create a more thorough design or a prototype, or (preferably!) to develop the entire game.

You shouldn't try to cover all aspects of the game in rigorous detail. This isn't the game's design script. It can be a tool for selling the game to a potential publisher or investor; if you're assembling a team of developers, it's a good way to explain it to potential candidates. The treatment should fill in a few of the gaps and answer some of the questions left by the high-concept document. This is the place for mocked-up screen shots, background on the key characters, a brief description of the overall story arc, and anything else that's crucial to understanding what the game will look and feel like to play. You should also include an analysis of the competition and indicate the ways in which your game will be different—and better.

The initial treatment is still a simple document—almost a brochure that sums up the basic ideas in the game. A good way of picturing what to write in a treatment is to imagine that you are making a web site to help sell your game; then throw in some business and development details for good measure.

The Game Script

The *game script* (or "bible") is the largest and the last in this series. It's not a sales tool; it's much too large and comprehensive for that. It's intended to document design decisions, not to persuade anyone of anything. The game script is the definitive reference for all matters relating to the structure and organization of the game, what the player does and sees—the gameplay. It should also cover the game storyline, characters, user interface, and rules of play. It should answer all possible questions (except for technical ones) about the game.

The game script does not include the technical design. It documents the creative, conceptual, and functional aspects of the game, and it should include technical specifications where necessary. However, it does not address how the game is built or implemented in software. The technical design document, if there is one, is usually based on the game script and is written by the lead programmer or technical director for the game. Technical design is beyond the scope of this book. If you want to know more about technical design, read *Game Architecture and Design* (New Riders Publishing, 2004).

As a good rule of thumb, the game script should enable you to "play" the game. That is, it should specify the rules of play in enough detail that you could, in theory, play the game without the use of computer—maybe as a (complicated) board game or table-top role-playing game. This doesn't mean you should actually sit down and play it as such, but it should theoretically be possible to do so, just based on the game script document. Sitting down and playing paper versions of game ideas is a very inexpensive way of getting valuable feedback on your game design. For designers without huge teams and equally huge budgets, we heartily encourage paper-play testing.

◂ PREVIOUS NEXT ▸

PREVIOUS NEXT

Anatomy of a Game Designer

Many of the people working in game development aspire to be game designers. It's a highly sought-after position, and for good reason. Of all the jobs on the development team, game design is the one that offers the greatest scope for creative expression. In most cases, it is also the designer who gets the credit for a successful game and who gets the media attention. And if the designer's name becomes well enough known, fame can lead to fortune. Sid Meier, the designer of *Civilization* and many other titles, is now so famous that his name alone can increase the sales of a game. *Sid Meier's Alpha Centauri* (Firaxis Software) was actually designed by an ex-employee of Firaxis, Brian Reynolds, yet the publisher put Sid Meier's name in the title.

Like all crafts, game design requires both talent and skill. Talent is innate, but skill is learned. Skilled craftsmen know everything about the requirements of their job, the tools of their trade, the material with which they work, and the result they are trying to produce. For an effective game designer, a wide base of skills is required.

> *"One of the most difficult tasks people can perform, however much others may despise it, is the invention of good games...."*
>
> —**C.G. Jung**

Perhaps because the ideal skill set for an effective game designer is *so* extensive, diverse, and poorly defined, the requirements for the position of game designer are frequently underestimated. Many development groups treat game design as just one more task for a technical project lead, or as a communal process performed by the entire team. In practice, the skills required for effective game design are much more wide-ranging than you'll find in your average technical lead. This doesn't mean that a programmer can't become a game designer; it just means that you need far more than programming skills to be one.

The following sections discuss some of the skills that are most useful for the professional game designer. Don't be discouraged if you don't possess all of them. That does not mean that game design is not for you. It's a wish list—the characteristics we would like to see in a hypothetical (but certainly nonexistent!) "perfect designer."

Imagination

A game exists in an artificial universe, a make-believe place governed by make-believe rules. Imagination is essential to creating this place. Fortunately, this is not a problem for most people. Even if you feel that your imagination isn't your strongest point, it's possible to develop and improve it. Of course, this isn't a new concept. It's been around for hundreds of years, as can be seen in the following extract:

> *"Thou shouldst regard various walls which are covered with all manner of spots, or stone of different composition. If thou hast any capacity for discovery, thou mayest behold there things which resemble various landscapes decked with mountains, rivers, cliffs, trees, large plains, hills and valley of many sort. Thou canst also behold all manner of battles, life-like positions of strange, unfamiliar figures, expressions of face, costumes, and numberless things which thou mayest put into good and perfect form....*
>
> *Do not despise this opinion of mine when I counsel thee sometimes not to let it appear burdensome to thee to pause and look at the spots on walls, or the ashes in the fire, or the clouds, or mud, or other such places; thou wilt make very wonderful discoveries in them, if thou observest them rightly.... For through confused and undefined things the mind is awakened to new discoveries. But take heed, first, that thou understandest how to shape well all the members of the things that thou wishest to represent, for instance, the limbs of living beings, as also the parts of a landscape, namely the stones, trees, and the like."*
>
> —**Leonardo da Vinci**, *Treatise on Painting*

The imagination is like a muscle; with regular exercise, it grows stronger and easier to flex. Imagination comes in various forms:

- *Visual and auditory* imagination enables you to think of new buildings, trees, animals, creatures, clothing, and people—how they sound and their strange ways of speaking.
- *Dramatic imagination* is required for the development of good characters, plots, scenes, motivation, emotions, climaxes, and outcomes.
- *Conceptual imagination* is about relationships between ideas, their interactions and dependencies.
- *Lateral thinking* is the process of looking for alternative answers, taking an unexpected route to solve a problem.

A great example of a product that demonstrates all of these forms of imagination in abundance is Infogrames's *Outcast*.

Deduction isn't ordinarily thought of as imagination, but it can lead to new and unexpected ideas. *Deduction* is the process of reasoning from a creative decision you've made to its possible consequences. For example, if you design a preindustrial farming community with no beasts of burden, the size of the fields and the productivity of the agriculture must necessarily be limited. Horses and oxen can pull plows and carry loads much farther than humans can. The limited food supply, in turn, affects the population growth rate and so on. These details are arrived at by deduction.

Nor does imagination consist only of the ability to think of things that are completely new. It's also valuable to be able to look at an old idea and apply a new spin to it, to breathe new life into it with a fresh approach. Too many people in the game industry see something old and familiar and dismiss it. Instead they should think, "How can this be made better? How might this be more interesting?" J.K. Rowling does this brilliantly in her *Harry Potter* novels. They're full of imaginative twists on old ideas about witches and wizards. She still has witches flying on broomsticks, but she invented the sport of Quidditch, which is played while flying on them.

Technical Awareness

Technical awareness is a general understanding of how computer programs, particularly games, actually work. It isn't necessary to be a software engineer, but it is extremely valuable to have had a little experience programming, even if only in Blitz Basic (www.blitzbasic.com) or some other simple language. A computer game designer's delivery medium is the computer, so it behooves you to know how computers do what they do and also to have a general idea of what they're not good at doing. A designer with entirely unrealistic expectations of what a computer can do isn't going to get very far. For example, computers do not understand English well. If your game design requires that the machine be able to interpret complex sentences typed on the keyboard, your programmers are likely to throw things at you.

You must also have a basic knowledge of the technical capabilities of your target platform. At a minimum, you need to know whether you're designing a product for a home console, desktop computer, handheld platform, or more than one of these. Every feature that you specify must be possible on the machine that you're designing for. If in doubt, ask the programmers. Knowing the limitations imposed by the selected platform will result in an achievable design.

Analytical Competence

No game design is perfect from the start. Game design is a process of iterative refinement, which progresses from a rough draft to the finished *meisterwerk*. Consequently, you must be able to recognize the good and bad parts of a design for what they are. This requires a keen logical and analytical mind, and the ability to manipulate nebulous concepts with a high level of mental agility and critical analysis.

It is very difficult to effectively criticize your own work. You can be excessively hard on yourself and become convinced that all your work is worthless, or you can be blinded by familiarity and unable to analyze your own work in an unbiased fashion. Inexperienced designers often err in both directions, swinging wildly from one to the other.

There is no easy solution to these problems. Good self-analysis skills come with practice. As a reality check, you can try peer reviews (which are always a good idea anyway). See what your colleagues and co-workers think of your design. Try to choose people who are likely to give you an unbiased opinion. Friends and family are usually not good choices. They will be either too easy on you—after all, they are close to you personally—or too hard on you because they don't want to be seen as biased and, consequently, overcompensate.

A specific example of when analytical skills are particularly useful is in detecting dominant (that is, unbeatable or nearly unbeatable) strategies at the design phase and weeding them out before they get into the code, as in the infamous *Red Alert* "tank rush." The tank effectiveness in *Red Alert* was so out of balance that an experienced player could dedicate all production to cranking out a few tanks and then immediately storm the opposition base before the enemy had a chance to get a production line set up.

Mathematical Competence

Designers must have basic math skills (particularly statistical math skills) because balancing a game is largely a matter of math and can be a difficult task. In the case of a real-time strategy game or a war game, the apparently simple problem of ensuring that there are no dominant strategies or fighting units to unbalance the game is actually quite math-intensive.

That's not to say that you need a Ph.D. in mathematics—you're not going to be deriving complex equations from scratch. The only thing you'll need above basic math is some knowledge of statistical methods. Even so, in some cases, the math can be handled computationally. It's entirely possible that you can handle the math requirement simply by being good with a spreadsheet program such as Microsoft Excel.

Aesthetic Competence

Although you need not be an artist, you should have a general aesthetic competence and some sense of style. Far too many games are visual clones of one another, depending on stereotypes and clichés rather than real imagination. It's up to you (along with your lead artist) to set the visual tone of the game and to create a consistent, harmonious look.

Suppose you're designing a clever and calculating female assassin with nerves of steel. Many designers would clothe her in skin-tight black leather and give her a big, shiny automatic pistol; in other words, they'd opt for the stereotype, the easy way out. A slightly more imaginative designer might give her a miniskirt and a crossbow—still male-fantasy material, but marginally more interesting. A really good designer would realize that an assassin needs to blend in to her surroundings, to look like anybody else, yet still be visually identifiable to the player. He would think about her personality and create a unique visual style for her that is distinctive yet unobtrusive. Lara Croft is a good example of this kind of thing. Despite her ludicrous proportions, she's dressed appropriately but quite identifiably for her role as an explorer: shorts, tank top, hiking boots. More important, her shirt is a particular color that nobody else in the *Tomb Raider* games wears, which makes her stand out on the screen. If you're seeing that color, you're seeing Lara.

We encourage you to expand your aesthetic horizons as much as you can. Learn a little about the fundamentals of art: the principles of composition, and which colors coordinate and which clash. Check out classic art books and visit art museums. Find out about famous art movements—Art Nouveau, Surrealism, Impressionism—and how they changed the way we see things. Watch movies that are famous for their visual style, such as *Metropolis*. Then move on to the more practical arts: architecture, interior decoration, industrial design. All these things can contribute to the look of your game. The more aesthetic experience you have, the more likely you are to produce an artistically innovative product.

General Knowledge

Some of the best game designers tend to be ex-programmers or people with at least some formal scientific training—as long as they have broadened their sphere of knowledge outside of their narrow field of specialist expertise. A base level of general knowledge is valuable for a game designer, as is the ability to research what you might not know. It helps to be well-versed in mathematics, logic, history, literature, art, science, and current affairs. The more source material a game designer can assimilate, the better the final game design will be.

Make sure you watch a lot of movies and documentaries (The History Channel and The Discovery Channel are excellent resources). Read books, too, both for direct research and background material. The encyclopedia is a good place to start for any given subject. The information it contains might be too general for any detailed design, but it will provide a launching point. From there, you can increase your knowledge of a particular area by investing in other, more specialized books. They don't have to be particularly advanced—unless you're aiming for something such as a historically accurate simulation, you can often use books aimed at children to bone up on the background. After all, you don't want to risk alienating your audience. A game that is too true-to-life (and, hence, is likely to require specialist knowledge) will disappoint its players.

Let's take an example: Most people's idea of pirates comes from swashbuckling Errol Flynn movies, J.M. Barrie's (or even Disney's) *Peter Pan*, and the Muppets' *Treasure Island*.

If you're designing a game based on piracy on the high seas, then exact historical detail isn't going to be what they expect—a game based on venereal disease, scurvy, maggot-infested food that has to be eaten in the dark so the crew can't see what they're eating, starvation, brutality, and the "consequences" of dark and lonely nights on a female-free boat won't make a particularly interesting (or even tasteful) game.

On the other hand, if you have a game design featuring colorful, eye-patch wearing, wooden-legged pirates, gold, running deck battles, the Black Spot, the Jolly Roger, and buried doubloons secreted away on remote desert islands, then you probably have the seed of an entertaining game. Though, of course, if you'd stuck to the historically accurate literature, you wouldn't use the Jolly Roger (pure mythology) or the Black Spot (invented by Robert Louis Stevenson, author of *Treasure Island*).

Books aren't the only source of knowledge, of course. A game designer's first research stop is often the Internet, with its wealth of free information. It takes only a small amount of searching using a quality search engine, such as www.google.com, to unearth some useful information.

Writing Skills

The professional game designer *must* have good writing skills. This means being clear, concise, accurate, unambiguous, and, above all, readable. Apart from having to write several detailed documents for each design, you might be expected to produce the story narrative or dialogue—especially if the budget won't stretch to a scriptwriter.

Writing comes in several forms, and we briefly discuss some of these here. (We go into more detail in Chapter 4, "Storytelling and Narrative," and the appendixes.)

- *Technical writing* is the process of documenting the design in preparation for development. All possible questions on the game have to be answered unambiguously and precisely. For example, if a farm with one peasant produces one unit of food per week, does a farm with five peasants produce five units, or do the additional peasants consume additional food? Is the peasants' own food factored in? What is the maximum number of peasants that can work on a given farm? If all the peasants are killed in a raid, what happens to the farm? If it is not immediately destroyed, can the empty buildings be reoccupied by new peasants? How do farms get built anyway? Where do peasants come from anyway? At what rate? Does the player have to manage the farming process, or do the peasants just get on with it? Can peasants fight? And so on and so on.
- *Fiction writing* (narrative) is useful for the manual, background material; character descriptions; introductory, transitional, and finale material; as well as other bodies of text, such as mission briefings.
- *Dialogue writing* (drama) is needed for audio voiceovers and cinematic material. Unfortunately, in most games the dialogue is even cornier than 1970s television shows, and the acting is as bad or worse. Try to avoid clichés, and use grammar and vocabulary that match each character's personality. If you have an ear for regional dialects, it can add real variety to your game, but, again, avoid stereotypes such as "dumb rednecks" or "brassy New Yorkers." The game *Starcraft* made excellent use of the subtle variety of southern American dialects, including proud aristocrats, gruff old generals, obnoxious bikers, and cheerful mechanics.

The greatest game design acumen in the world will be useless to you unless you can effectively communicate your ideas. As a designer, you will often need to evangelize your design at several different levels. In the professional setting, you will initially need to present your design to a publisher. Following that, you need to obtain buy-in from the team that will develop the game. This can be done in person or in writing, but the result is the same. You will need to be able to transmit your enthusiasm for the game as well as go into mind-boggling detail of its finer points to allow the development team to be able to create a game from your vision.

Drawing Skills

Some skill at basic drawing and sketching is highly valuable, although not absolutely required for a designer if you have a concept artist to work with. The saying "A picture is worth a thousand words" is never more true than when you are trying to impart a game idea. The vast majority of computer games rely heavily on visual content, and drawings are essential when pitching a product to a third party.

Game-publishing executives will be interested in a hot concept, a hot market, or a hot license, but only pictures get them excited. The images are the hook on which the executives will hang everything else that they hear. Otherwise, you can talk about concepts forever and they won't remember a thing when you leave the room because they don't have a visual hook to hang it on. The images will remain in their memories long after they forget the details.

The Ability to Compromise

Probably the most important skill for a professional game designer is the ability to compromise on details and integrate a variety of opinions while preserving a consistent, holistic vision of the game. In an ideal world, we would be able to design whatever suited us and never have to worry about the demands of a publisher or the interests of the customers. Unfortunately, these external needs and interests do have to be considered, and, more often than not, the game designer is constrained by genre or license.

Different people on the development team and at the publishing company will have concerns about their own areas of expertise (programming, art, music, and so on), and their opinions will pull and push the design in different directions. As the designer, it's tempting to seek sole ownership of the vision, to argue that things must be exactly as you imagined them and to ignore other considerations. After all, there are plenty of other voices in your company advancing points of view about the other areas. But you must resist the temptation to do that, for two reasons:

- First, you must allow your team some ownership of the vision as well, or its members won't have any motivation or enthusiasm for the project. No one builds computer games solely for the money; we're all here so that we can contribute creatively.
- Second, a designer who gets a reputation as a visionary but who can't deliver a buildable, marketable game doesn't stay employed for long. It's your job to deliver an integrated design.

In many cases, you'll be given a brief that limits you to designing a genre clone or a heavily restricted licensed property. Being able to work within these limits to the satisfaction of the customer, whether your customer is the publisher, license holder, or end user, is essential. Unless you are one of the famous game designers who can guarantee success with your name alone, you are unlikely to be given a completely free rein. You may have designed the best text adventure game since *Colossal Cave*, but if the style of game is out of favor with the public, you're probably not going to get your game made—let alone published.

◀ PREVIOUS NEXT ▶

◂ PREVIOUS NEXT ▸

Putting It Together

This chapter puts forward the view that game design is not an arcane art, but rather a craft, just like any other, that can be learned with application.

Games are not works of art created by a mysterious, hit-or-miss process. Instead, they are a discrete set of rules and a presentation layer for those rules. We believe that by examining and documenting the fundamental similarities and differences between games of all types—computer and board, mental and physical—we can identify a valuable set of common elements that can be used to understand and, consequently, improve how great games are constructed.

As we said before, no one person is likely to have all these skills. We are talking about a hypothetical ideal game designer—the Superman of game design. It's no coincidence, however, that the best professional game designers possess a high degree of competence in many of these skills. They weren't born with them; they educated themselves and worked to develop them. We encourage you to do the same.

◂ PREVIOUS NEXT ▸