6 Foundations of Game Design

In This Chapter

- The Phases of Development
- Examining the Design Process
- Tight Design
- Elastic Design
- Extensive Design
- The Presentation Dilemma
- Conclusion

Game designers meet a lot of people who say they want to be game designers. Usually, they are players of video games who are drawn to the idea of getting to decide what goes into a game and believe that a game designer is the person who makes these decisions. The truth, however, is that game designers rarely decide what a game will be about, what genre it will be, who the central characters will be, or any of the other factors which frame the creation of a game. This situation has been the case since the end of the era of "bedroom programmers" (individuals who would design, code, and create graphics and sound for their games on early home computers) and is unlikely to change.

A game designer creates paperwork that specifies the design details of a game, but they do not usually get to decide what goes into the game. Those decisions are

made either at a higher level of the "chain of command" or made by the entire team as a group. The game designer might be charged with the task of producing a coherent design that ties together all of the elements required to make the game work and might create inventive mechanics to integrate the game elements, but they do not control the development of the game.

So if the game designer does not decide what the game is about, what does the game designer do? What are the basic game design skills that the designer uses to craft a design document, and why is game design as essential to the development process as programming or art? This chapter looks at the foundations of game design, the video game development process, some basic game design skills, and the core of the role of the game designer.

THE PHASES OF DEVELOPMENT

The complete process of video game design tends to consist of four discrete phases. The first of these, *concept*, involves the creation of the game idea and framework in the broadest strokes. The second phase, which can be called *initial design*, involves expanding the concept into a more complete design document. The third phase is the period of *expansion*, when the game design grows to accommodate all the desired features, and the final phase is *contraction*, when the more ambitious features are lost in a desperate attempt to meet the milestone schedule.

Phase One: Concept

Most games begin either with a developer pitching a concept to a publisher (usually supported by a tech demo), or with a publisher looking to produce a game for a license they have acquired (for example, a popular film license). As a result, the game designer's first task on any game is usually to create a concept document that encapsulates the desired high-level details and expresses an initial concept of gameplay.

Because of the developer-publisher relationship, the publisher is usually at the top of the development chain of command. Publishers want to create a product that will make as much money as possible and attempt to influence the development process towards this goal. In point of fact, the publisher very rarely has any way of knowing what will sell well and has to rely essentially on guesswork. Therefore, most publishers assess projects in terms of the risk of the project not being completed (or not being completed on time), and their chief role in the development process is to ensure that game milestones (development targets) are made on time.

As a result of this situation, the people who generally determine the high-level details about a game are at the management level of the developers, but they do so with the goal of interesting a publisher. Each developer has their own way of approaching this. Some have management staff who perform this and other duties; others allow the development teams themselves to determine the broad strokes of a game in a democratic manner. Even if the team is in charge of determining the shape of the game, they usually have to win approval for the project from someone in the developer's management staff.

All these factors—the developer's need to determine the overall shape of a game, and the need for publishers and developers to have some tangible basis for discussion—create the need for a concept document. These documents vary in size from one page to a dozen and serve to present a broad-strokes overview of a game. Concept documents contain the high-level details of a game and the basic gameplay mechanics—enough to give an idea of how the game might work. Depending on its size, it might cover the user interface, the core mechanics, the game structure, and the narrative component of the game, where appropriate.

Phase Two: Initial Design

Once a publisher has agreed to fund a project (and in some cases even before this point), the concept document becomes insufficient, and a design document is required. This transition can result from an expansion of the concept document—filling in the blanks that were only implied in the concept, for example—or the design document can be an entirely new item of documentation generated using the concept document as a general overview.

When developers make games without a dedicated game designer, the game design document generally gets put together by someone during the early stages of development. In these situations, the document is often being created only in order to get the game design approved by the publisher (or in some cases the platform licensor, if it is destined for a console); after this point, the document falls by the wayside as an artifact of development. This state of affairs suits only the least ambitious game projects, or those projects that are driven solely by technology.

Initial design without a game designer is usually a series of meetings between various development staff (or, in the worst case, all the staff). Everyone has ideas they want to incorporate, and a large number of these ideas pull in radically different directions, leading to the usual problems associated with committee-driven design. The final results can be worthwhile if the team is talented, has good communication skills, has sufficient patience, and if sufficient time is budgeted to warrant such an approach.

Conversely, a lone game designer (or, even better, a small design team) can complete the initial design in a relatively short period of time, without leaving out the views of the rest of the team. By initially discussing the game with the individuals involved in the project, the game designer or design team can learn the desired traits and then work to incorporate them into a coherent initial design document.

Phase Three: Expansion

Once an approved initial design is in place, programming and art creation can proceed, and the period of expansion begins. In practice, the initial design phase can sometimes be omitted (especially if the game is technology driven) or the expansion phase can occur as part of the initial design, with the programmers and artists starting work only when the paperwork is principally complete. More commonly, some programming and art work begin at the concept phase, because of the need for tech demos to convince publishers of the value of the project; in other words, by the expansion phase, a prototype already exists.

The game designer has more work to do during the expansion phase than at any other time. Every aspect of the concept must be expanded, and every point of expansion must be built on sound design principles, be realistic for the programming team to implement, and be reasonable for the artists to render (where applicable). When the process is working smoothly, each iteration of the design documents receives feedback from the programming, art, and production team until the design is complete.

For larger projects, the documentation expands to the point whereby it is no longer helpful to keep it in a single document. It can splinter into main design documents, level design overviews, narrative documents, agent reference lists, and art specifications—a vast variety of documentation.

Phase Four: Contraction

At a certain point during development, it is no longer tenable for the game to continue to expand, and the contraction process begins. Either management puts a freeze on the game, refusing to allow any new features to ensure that the game is delivered on time, or the project has already started to slip from its milestone schedule and the game design must shrink to make allowances for the delays. From hereon in, the game begins to contract.

This is when the most difficult part of the game design process generally begins (although shrewd design work in the earlier stages can relieve some of the difficulty). When it becomes apparent that not all the high-level features that the team (or management) wanted to get into the game are feasible, the game designer must determine what is essential to the design and what can be removed. They are often

unhappy with what must be lost in the contraction, but with good design skills, the game can be significantly improved by this process.

Contraction allows the game design to become more focused. The features that get thrown out first are often those that seemed like a good idea but do not support the core of the design, resulting in a tighter design. At its best, contraction improves the final game; at its worst, the process strips the game design of anything interesting. It is therefore a critical part of the design process and must always be taken into consideration.

EXAMINING THE DESIGN PROCESS

Game design is characterized by an early, abstract phase (concept); a period during which the game acquires its outline and core features (initial design); an extension phase that increases the level of detail and completeness (expansion); and a simplification phase that focuses and tightens the design (contraction). Programming and art creation can begin at any stage, but preferably tend towards starting in the expansion phase, not counting any materials put together for a tech demo.

The amount of freedom a game design has to move in the expansion and contraction phases can be referred to as the *elasticity* of the design. A design in which all of the features support the core concept of the game can be described as *tight*, whereas a design that contains an exhaustive collection of features can be described as *extensive*.

Each of these concepts—tightness, elasticity, and extensiveness—are worth closer examination, as each has a role in the process of game design.

TIGHT DESIGN

Although we tend to think of the role of game designer as referring solely to video games, the process of game design has been around for at least a hundred years. Many video game designers started as game designers in another medium. Substantial level design in *Doom* (id Software, 1993) and *Quake* (id Software, 1996) was carried out by Sandy Peterson, who also created the popular *Call of Cthulhu* (Chaosium, 1981) tabletop role-playing game; Julian Gollop's *Chaos* (1984) was based upon a board game he had created; and game design luminary Warren Specter produced tabletop role-playing games and board games for Steve Jackson Games prior to working on video games.

Board game designer Klaus Teuber has stated that when he redesigns his board games, his goal is to simplify the game without losing any of its essence [Gamewire04].

This is the purpose of tight design—to use the minimum quantity of elements required to support the desired gameplay. Teuber's most famous creation, *Die Siedler von Catan/Settlers of Catan* (Franckh-Kosmos Verlags, 1995), demonstrates the benefits of tight design with its simple set of mechanics that supports diverse gameplay, and the board game has sold more than three million copies around the world. Game designers often cite it as an exemplary case of elegant design.

The epitome of tight game design comes with card games, and creating a card game for an arbitrary audience is an exemplary exercise in tight design. What makes card games so useful in this area is that the creative freedom is limited by the available materials. You have only fifty-four cards to use—four suits, thirteen values, plus jokers (potentially), plus a few peripheral options such as score or use of a table. Because the "hardware" we have available is so simplistic, successful game design is almost guaranteed to be tight.

Designing Card Games

What are the design components of a card game? The players, the hand, the deck, the discards, the cards in play, the turn sequence, the rules, and the meta-rules. Decisions in each of these areas define the nature of the card game.

A card game can have one, two, four, or many players. Although in principle a card game can be designed for three players, in practice most new card games are designed for either solo play or two or more players. Designing a game for exactly three players would dramatically restrict the chances to play it, so this would be an unusual choice for any game. Some examples of different player choices include *Solitaire* (1 player), *Beggar-Your-Neighbor* (2 player), *Poker* (2 or more players), and *Bridge* (4 players).

The hand denotes cards the players keep to themselves. Not every card game has a concept of a hand. For example, *Memory* begins by all the cards being dealt onto the table, and players take turns trying to find cards of matching value out of the cards in play. In general, however, each game has rules about the hand. *Beggar-Your-Neighbor* begins with a hand equal to half the cards in the deck, and *Bridge* begins with one quarter of the cards in the deck. Each of the *Poker* variants, on the other hand, has a certain hand size—generally from two to seven.

The deck and the discards define the flow of cards in the game (though as with hands, not every game has a deck and discards). In most games, new cards come from the deck and go to the discards. In general, discards are out-of-play. However, games such as the many *Rummy* variants allow new cards to also come from the discards (and indeed, the prime difference between most variants of *Rummy* is in the degree of access to the discards that the player is allowed, for example, top card only, all discards, etc.).

"Cards in play" can mean cards in front of a particular player that apply only to that player, or cards on the table that affect every player. As with other elements, not every card game requires cards in play, whereas other games (such as *Solitaire*) get almost all of their gameplay from the cards in play.

The turn sequence dictates both the order of play and the allowable actions. The most common system is that each player takes a turn in a clockwise sequence, but many other options are available. Some *Poker* variants, for example, vary the order of play according to the nature of the betting, and *Bridge* and many *Whist* variants have a turn sequence dictated by victory in the previous turn (known generally as a "hand"). The allowable actions state how the hand and cards in play can change during the course of a player's turn.

Finally, the rules and the meta-rules dictate the nature of all remaining interactions. Rules dictate the meaning of individual cards, sets of cards, and the state of play, as well as how the game is won or lost. The rules of *Poker*, for example, dictate that a straight flush is the strongest hand, followed by four of a kind, flush, straight, three of a kind, etc., and (in most variants) they dictate that the strongest hand wins. If a card game has a scoring mechanism, the rules dictate this.

Meta-rules are modifiers to the general rules. When *Poker* players decide that, for example, deuces are wild, they have added a meta-rule to the rules. The distinction between a rule and a meta-rule is largely one of interpretation, but in general a meta-rule is something that can change according to the desires of the players, while a rule defines the nature of the game. What people call "house rules" are functionally equivalent to meta-rules.

Between these eight factors, a card game results. The simplest games, like *Beggar-Your Neighbor* (also known as *Beggar-My-Neighbor* or *Strip Jack Naked*) are deterministic and suitable for entertaining only children, because the rules mean that the player has no choice in the outcome; there is as much skill in such a game as there is in rolling dice and determining who has the highest score (although such aleatory play becomes more meaningful to adults when money is at stake). The most accessible games strike a balance between skill and luck (most *Rummy* variants fall into this group), and the most enduring games are simple to learn but support a wide variety of play styles (as is true of most *Poker* variants) [Hoyle46], [Penguin79].

Tight Card Games

To produce a tightly designed card game, you need to make a decision in at least three of the eight aspects of card game design. (You must have either a hand or cards in play, there must be a turn sequence of some manner, and there must be

rules to dictate the end of play.) You can be certain that the game is tight by keeping each of the given aspects as terse as possible, but what most game designers find when they experiment with creating card games is that a very basic set of design components produces ambiguities or problems that require clarification or refinement.

When the volume of additions to the design components exceeds the initial set of components, the game design has gotten out of control. At this stage, the game might possibly be fun to play, but it is difficult to learn and almost impossible to spread to a wider audience. It might be a card game, but it is no longer a tight card game, and as such its "shelf life" is limited. We can say that the mechanics have become slack.

Tight game design is not only elegant, but it is easier for arbitrary players to learn to play. This accessibility is the main commercial advantage of tight design in video games, but tight design also means that little development time is wasted developing secondary components that the game does not actually need. Many video games suffer from slack design, often as a result of a committee-driven design process—the aspiration to accommodate all the desires of a development team can be disastrous if no one is looking for the subset of ideas that lead to a tight design.

Anyone involved in (or wanting to be involved in) video game design can learn much about tight game design by experimenting with card games. With such a small set of options to play with, these games can be created quickly, and the distinctions between a tight card game design and a confused and cluttered card game design are easy to identify. Working in such a constrained state space discourages slack design, and every game should work towards tight mechanics if it is ever to reach a wide audience.

Goals of Tight Design

Tightness is a concept we have defined as the property of having all (or the majority of) features in a game's design support the core concept. It is a desirable property for all game designs, because a tight set of mechanics is easy to learn, but it is especially desirable in video games where the cost of implementing game mechanics is high and therefore you do not want to commit to developing more than is necessary.

We have seen that it is easier to produce a tight design for a card game than for other types of games, because the number of design components is limited, which forces a certain economy. You can produce card game designs that are not tight but still fun, just as you can produce a video game that is fun to play, but has a giant learning curve. The fact that a game is fun to play does not necessarily justify the game in a commercial perspective, though. A fun, complex card game costs nothing to design or implement—a complex video game is very expensive to develop and must reach a large audience to justify that expense.

By keeping to the goal of tight game mechanics, the video game designer reduces the cost of development, improves the accessibility of the game to a wider audience, and ensures that the game play feels elegant and interconnected. Because tightness cannot be measured, it is a subjective concept, and you must acquire an aesthetic to judge its presence and absence. We find few better ways to acquire this perspective than by experimenting with card games.

ELASTIC DESIGN

The form that best expresses the concept of elastic design is that of the card-based board game, characterized by games such as *Fluxx* (ICE, 1998), *Illuminati* (Steve Jackson Games, 1982), and trading card games like *Magic: The Gathering* (Wizards of the Coast, 1993). These games are primarily comprised of a set of cards sometimes (but not always) used in conjunction with a set of counters. The basic game design principles are almost identical to card games, the chief difference being that each card has its own unique identity carrying with it the capacity to alter, expand, or change the rules of the game.

Fluxx is the epitome of the form, consisting as it does of four clearly identifiable types of cards. Keepers are privately owned cards in play, New Rules are publicly owned cards in play that alter the state of the rules, Goals are publicly owned cards in play that specify the winning conditions, and Actions are cards that have an immediate effect when played. The game offers an entertainingly variable play experience as the changing state of the rules makes each game unique—for example, changing the number of cards drawn or played, forcing players to play a card randomly, causing people to exchange their hands, and so forth.

Designing Card-Based Board Games

When creating a game in this form, the card mix is the main design problem (because the core game mechanics usually write themselves). For the game to be fun to play, players must be able to draw interesting cards throughout the game, they must frequently have meaningful choices to make, and (most importantly) no card should upset the balance of the game play. In many games of this style, certain key card types must occur with a certain frequency or else the game mechanics break down (for example, in *Magic: The Gathering* the land type must appear approximately one third of the time if a deck is to be playable).

During the design process, the game designers are usually working towards a set card limit, often dictated by the production method (which generally involves printing onto cardboard sheets that are then die-cut into individual cards, with a fixed number of cards per sheet) and the cost of manufacture. This target number

of cards is usually fixed, but might change throughout the development process. The problem this target number represents is that designers always have many more possible cards than the target; they must throw out a certain proportion of the cards in development.

This is the principle of elasticity in action: having more resources than you need with the express purpose of whittling the set of components down to the best minimal set that the production constraints allow. In practice, some of the initial set of cards will prove unbalanced or unworkable and will have to be thrown out, and sometimes play testing suggests new cards that would greatly improve the play mix. In an elastic design situation, this freedom to make changes must be anticipated.

Types of Elasticity

/ When elasticity is expressed by a freedom to throw out components, we can call it *contractile elasticity*; when elasticity is expressed by a freedom to add new components, we can call it *expansile elasticity*. When both these properties exist in unison, the game designers have total freedom to redesign the game during the development process, a situation that is extremely common in board game design, but that is both rare and expensive in video games. /

Limiting elasticity generally implies that the game will not be as good as it could be, whether the game is a board game or a video game. The elasticity of design can be preserved in a card-based board game because the cost of each iteration of the design is trivial compared to the cost of manufacture. Each iterative print of the prototype cards costs very little, and the largest cost in this situation is the time and resources required for adequate play testing.

In video games, this is rarely the case. This kind of flexibility in video games is significantly more expensive, although some designs can benefit from cheaper elasticity through the use of paper prototypes. If a subset of the game mechanics can be expressed in a board game of some kind, this allows for cheap testing of the mechanics involved, although this option is generally limited to highly abstracted games. Eric Wujcik, creator of many tabletop role-playing games including *The Palladium Role-Playing Game* (Palladium Books, 1982) and *Teenage Mutant Ninja Turtles and Other Strangeness* (Palladium Books, 1985), observed that reality is too complex to simulate completely, although computers narrow the gap with every passing year [Schick91].

Expansile elasticity in video games is generally the purview of the most expensive products, those that are destined to be considered AAA. Only AAA class games can justify the budget requirements attached to the freedom to add new details and

features during the development process. Every video game project would benefit from this freedom, but in reality only the big budget games can afford it.

Contractile elasticity, on the other hand, is essential in all video game design. Game designs must have the freedom to contract because the actual time it takes the art team and, in particular, the programming team to implement a particular design cannot be calculated, only estimated. Unexpected difficulties, delays, bugs, and other problems all contribute to slippage (a delay in the release of the game) unless the design has contractile elasticity built into it.

In the worst case, the contraction of a game design means the loss of design features, which can invalidate the core gameplay. This is a situation all game designers want to avoid, and with this in mind, all video game designs must include a degree of contractile elasticity. They must be designed to allow for certain elements to fall out as the contraction period of development takes its toll on the project.

Goals of Elastic Design

We have defined elasticity as the freedom to make changes during the design process and have identified distinctions between expansile elasticity—the freedom to expand the game design—and contractile elasticity—the ability for the game design to shrink without developing problems as a result of the loss of design components. Contractile elasticity is desirable in all video game projects to allow for inevitable changes in the development cycle.

The development of card-based board games is inherently elastic, as game balance is a product of maintaining an effective card mix, and it is both simple and inexpensive to change this card mix for that particular style of game. Video game design should always aim to include elements of contractile elasticity, and this inclusion is a non-trivial design process. Remove too many elements of a design, and the remaining features no longer form a coherent whole. Conversely, expansile elasticity is easy to carry out, but generally applies only to the big budget, AAA games.

Because we have already demonstrated the value of tight design, it follows that the vast majority of video game designs need to be tight-elastic. Most games are not AAA, and most games have a strictly limited budget for development. If they can't be delivered inside that budget, they might not be completed at all. Tightness is desirable because it ensures that the game design is constructed around a complementary core of supporting game mechanics, but at the same time making the core mechanics tight removes a certain freedom for contraction, effectively limiting elasticity. Finding a way to balance tightness with elasticity is a useful skill for game designers to develop.

EXTENSIVE DESIGN

Extensiveness is a property that is sometimes mistaken for good design. Any game with many different features, many activities supported, and a wide selection of choices in the player's core actions can be described as having an extensive design. In the sense that extensiveness gives the players options, greater freedom, and a reduced feeling of being constrained, it is a desirable property. Realistically, however, these benefits can be costly to develop, and if the game does not also feature the properties of tightness and elasticity, extensiveness can be the downfall of the entire project.

Extensiveness is not an antonym for tightness; any extensive game can be simultaneously tight, provided a coherent logic to the confluence of the game mechanics exists. However, games that lack extensiveness tend towards tightness; we can describe such designs as compact or concise, and concise design is inherently tight. Tightness is a property that you start with in abundance and that diminishes as new game mechanics are added that don't support a consistent core set; extensiveness is a property that starts in total absence but always increases as new mechanics are added—irrespective of the nature, tightness, or quality of the mechanics that are added.

The concept of extensive design is epitomized by the tabletop role-playing game. Although creating a complete role-playing game is a tremendous effort as merely an exercise, every video game designer would do well to look at several role-playing games to get a feeling for the concept of extensive design. It is not that the quality of game mechanics in role-playing games is particular good (it is generally quite patchy); rather, it is the quality of extensiveness that can be found in relative abundance.

Tabletop Role-Playing Games

The basis of all role-playing games is captured in *Contract* (Discordia Incorporated/Infamy Games, 1998), which was created as an attempt to boil the nature of this game form to its minimal form. The minimal form of the tabletop role-playing game (hereafter, abbreviated to RPG) is a social contract between the referee (often known as a games master) and the players; the players implicitly agree to let the referee decide how their actions will develop the story, and in return the referee implicitly agrees to provide either entertainment, or impartiality, in their role as mediator and chief storyteller.

Contract embodies the observation attributed to E. Gary Gygax that if RPG players ever realized that they could do everything by themselves, the game designers would be out of a job. The fact of the matter is, however, that most games masters require some mechanics to help them run the game world, and most players

feel more comfortable trusting the games master's role as referee when some random-driven mechanics provide a framework. The aleatory appeal of dice rolling is also part of the appeal and ritual of tabletop RPGs. Most RPGs are therefore concerned with providing a set of mechanics to simulate the game world and most (but by no means all) provide a set of mechanics for use with dice to simulate events.

The archetypal RPG provides its mechanics in a set way, and this method has been inherited almost wholesale by the computer RPG (cRPG). This form is constructed by creating a character role for each player, which is expressed in terms of attributes and/or skills—usually in terms of numbers attributed to qualities of personality, physical prowess, or talent. The mechanics provide a number of rules and/or tables for relating these attributes and skills to various game tasks, most specifically combat, which is usually the most detailed part of any RPG rulebook.

Eric Wujcik suggested that the trickiest balance in RPG design was combat realism versus playability, and this holds true in video games. In one of his essays he states: "Real-life combat is a model that can't be applied directly, simply because it's too deadly or (if accurately simulated) too time consuming" [Schick91].

RPG Templates

The most famous model of extensiveness in RPGs is E. Gary Gygax's *Advanced Dungeons & Dragons* (TSR, 1978). In this, so many tables, mechanics, and gameplay details were provided that the rules spread out across three different volumes comprising more than 460 pages (and later gaining more volumes adding ever-more detail). Ironically, the combat mechanics at the core of *AD&D* are largely biased towards playability over realism. The bulk of the mechanics are concerned with providing tools for the games master to create adventures, and trying to compensate for the inadequacies of the core mechanics at expressing much beyond combat.

AD&D, like *Dungeons & Dragons* (TSR, 1974) before it, is considered a "class and level" system. These systems are the basis of the majority of cRPGs because "class and level" systems are easily accessible (and also because the exponential experience system behind most level systems can be fiendishly addictive to some players, especially those with a preference for Type 1 play). The other most prevalent style of RPG is the skill-based system, which is usually excellent at simulating non-combat situations, because such systems have a general skill mechanic for resolving tasks by comparing character skills with a die roll by some method.

Chaosium pioneered arguably the most famous of the early skill-based systems, which they called Basic Role-Playing—the backbone of their successful *RuneQuest* (Chaosium, 1978), *Call of Cthulhu* (Chaosium, 1981), and several other RPG systems. In these, the mechanics are simple—skills are represented as

percentages, and success or failure is determined by a die roll. The simplicity is enviable, but the problem with this mechanic is that most die rolls end in failure, which makes storytelling somewhat fatalistic unless the games master is generous with what the players can achieve without having to roll dice.

Percentage-based skill systems are often extensive by virtue of the need to interpret the meaning of the percentage levels of the various skills. Many games systems that use this approach provide supplemental mechanics for each and every skill, providing the games master with potential assistance with problem situations at the cost of greater reliance on mechanics. These systems are of very little use in video games where percentage systems seem somewhat capricious. If the player reaches a locked door, for example, the game cannot make a single random determination as to whether the player can break into the door unless other guaranteed means of entry exist—and if reliable methods for getting past the door do exist, the random chance of breaking in becomes both incongruous and superfluous.

Converting Mechanics

The inapplicability of most RPG mechanics to video games stems from the key distinction between the two forms. An RPG uses simple mechanics because it (in general) uses dice as its random source, and because it works best when the random mechanics are used only when dramatically appropriate. A video game uses comparatively complex mechanics because (a) they can and (b) video games must model all interactions in the game world, not just those that are dramatically interesting.

Despite this, RPGs are tremendous examples of extensive design, frequently attempting to cover almost every situation that might present itself with some form of mechanics. Like an extensive video game design, they end up focusing on those situations likely to occur, or those that support the core expected game activities. In a game where combat is a key activity (true of most RPGs, perhaps unfortunately), even the variety of combat situations are covered extensively.

An excellent exercise for any would-be video game designer is to take an arbitrary RPG and convert it to a concept document. This forces the subject to consider many useful questions: What are the core game activities? How much expressiveness can we support in the game actions available to the player? Which mechanics will convert well to a video game setting? Because one is dealing with source materials that already have mechanics specified, the resulting process presents more game design challenges than simply coming up with a concept document from scratch.

The Cost of Extensiveness

The reason RPGs display such a high degree of extensiveness is that it is a property that is cheap in that form of game. They are generally printed as books, and the cost of adding another page of rules material is trivial compared to the total cost of each print run. The same page of additional game mechanics added to a video game design could add another man-month of programming, art, and QA, adding at least an additional $20 thousand to the cost of the project.

In fact, a video game must have substantial development resources (both in terms of budget and of team size) if it is to express any degree of extensiveness, and this can make it a highly undesirable property for many projects. This is not to say that the audience does not desire extensiveness—to a certain extent it is expected that all AAA games deliver a high level of extensiveness—but the acquisition of extensiveness might cost tightness. Therefore, it is a property to be approached with some caution.

No video game can ever be as extensive as a tabletop RPG, at least until AI can accurately simulate human intelligence (which is unlikely to happen during the lifetime of any reader of this chapter). Despite this gap, studying RPGs can show how to achieve a high degree of expressiveness using only a simple core of central mechanics, and this is the goal of extensive design: to support a considerable number of player activities using the same core resources.

Dangers of Extensive Design

We have defined extensiveness as tending towards a wide variety of features and activities expressed in the design. Although extensive designs can easily lose tightness, you can produce tight-extensive designs provided the game designer always keeps in mind the principles of tight design when adding in new features. Extensiveness is a luxury that in general can be afforded only by AAA designs, because only these big budget games can expect to implement more than a core set of mechanics. The presence of extensiveness usually implies contractile elasticity, which is always desirable in development.

Tabletop RPG's are replete with extensiveness, because of the cheap cost of including many options in a game form that is printed as text. By comparison, extensive video game design is always expensive to develop, and as such, game designers must be careful when they have the freedom to produce extensive designs. Analyzing RPG design can be a useful step towards identifying both the benefits and pitfalls of extensive design.

When tackling extensive design, the game designer must keep in mind the principle of tightness to ensure coherence in the resulting project, but often need not

worry about elasticity, which is usually inherent to extensiveness. The expansion phase of video game development tends towards extensiveness, and the extent to which this growth is allowed to proceed must be in proportion with the budget for the game. The game designer might desire to focus on extensiveness over tightness, but to do so is to be negligent in the designer's duty to the game.

THE PRESENTATION DILEMMA

One quandary facing many video game designers working on typical products is that their design work generally includes some features that are outside the core mechanics but that potentially make for interesting new gameplay. The problem with these unnecessary new ideas is that the full implication of the mechanics might not be clear until much later in the development process, making their inclusion in the game somewhat risky. Because they do not want to lose these desirable features, they expend some effort in making these points essential to the game design—with a corresponding loss of elasticity.

Many of these features are doomed to be cut, and the honest way to present them would be to mark them as optional or discretionary design components. However, designers are reluctant to do this because they know from experience that cuts have to be made at some point and optional components are the first to go. They might just as well cut them at the initial design phase, because in the extremely unlikely event that additional time is left at the end of development, the time can always be used for spit and polish, and tweaking time can be more valuable to a project than new design features.

The presentation dilemma is precisely this: should you include new, interesting, and innovative design features if you know they are going to be cut later in the project? Perhaps the best answer to this question is the following: "If you know they can be cut, the features cannot be part of the core design." In an extensive design, you might have the luxury to include such features and see what happens, but in a typical tight-elastic design, you don't. Cut everything you can live without.

Instead of gambling on getting additional design features into place, the most prudent approach is to allow for contractile elasticity in the peripheral design elements (for example, the level structure). In a game organized on a level basis, figure in some levels that can drop out of the design without the need for excessive restructuring. These components can afford to be optional because publishers expect a certain minimum play time for every product and therefore cuts that affect game length (play time) are less likely to be implemented than cuts that effect core design features.

CONCLUSION

The game design process begins with a tight concept, for which expansile elasticity is present in abundance because the concept is largely incomplete. The initial design phase begins the growth of extensiveness, which must be curtailed in most game projects but is allowable in AAA games. Focusing on tight mechanics in the initial design phase makes it easier to remain tight throughout the development process by identifying the core elements of the design conception. During expansion, extensiveness must be controlled in proportion to the development budget, and the designer must ensure that contractile elasticity is present or else the inevitable contraction phase will be unable to adjust to delays and problems resulting in development.

As espoused in the philosophy of Zen Game Design, the game designer has a duty to design responsibly; they must balance the needs of the game with the desires of the development team, while simultaneously attempting to anticipate the needs of the game's audience. Striving for tight game designs, innovation should be focused at the micro-level with a goal of producing consistent, integrated core mechanics that provide a high degree of expressiveness. Elasticity is essential, tightness is desirable, and extensiveness is a dangerous property that is needed in the most expensive game projects, but must be secondary to maintaining tight mechanics.

Every game designer wants the opportunity to be inventive, but such unbounded creativity can be the most expensive part of any video game project. AAA products might have the luxury of inventiveness, but most games are not AAA and must emphasize reliability of design over ingenuity. Restraint is one of the greatest talents any game designer can possess; when it is coupled with instincts for tightness and elasticity, it is the height of the game design art.

Resourcefulness is still of value in game design, and every design, no matter how modest, has room for creativity. If designers sharpen their skills in tight-elastic game design, they can be certain that when they earn the freedom to carry out extensive designs, they will be able to design in a manner that demonstrates all three traits: tightness, extensiveness, and elasticity.

Returning to Eric Wujcik for one last piece of game design wisdom: "Add only those new elements that are necessary for the game. Aim for the minimum of innovation" [Schick91]. The foundation of game design is in providing the right design work for the project, not in trying to dazzle with unfettered creativity. All game designers have a duty to the project that must always take precedence over their desires as players of games, and that duty is the core of the game designer's role.