·46·

USER-CENTERED DESIGN IN GAMES

Randy J. Pagulayan, Kevin Keeker, Dennis Wixon, Ramon L. Romero, and Thomas Fuller
Microsoft Game Studios

INTRODUCTION

The intent of this chapter is to review principles and challenges in game design and evaluation and to discuss user-centered techniques that address those challenges First we present why games are important, followed by the definitions and differences between games and productivity software. In the next section, we discuss the principles and challenges that are unique to the design of games That discussion provides a framework for what we believe are the core variables that should be measured to aid in game design and evaluation The chapter concludes with some examples of how those variables can be operationalized in which we present the methods used by the Microsoft Game Studios User-testing Group.

WHY GAMES ARE IMPORTANT

Computers that appeared commercially in the 1950s created a technological barrier that was not easy to overcome for the greater population Only scientists, engineers and highly technical persons were able to use these machines (Preece et al 1994) As computers became less expensive more advanced, and more reliable the technology that was once only available to a small group of people permeated throughout the population and into everyday life To ease this transition, the need for a well-designed interface between the user and the technology became a necessity Computers games come from similar origins and potentially may head down a similar path Early attempts at making commercial video games have failed because of unnecessarily high levels of complexity The following quote from Nolan Bushnell (cofounder of Atari) states the issue quite succinctly: 'No one wants to read an encyclopedia to play a game' (Kent, 2000 p 28). In retrospect, some of the most successful video games were the ones that were indeed very simple

Entertainment in general is a field of great importance to everyone (Schell, chapter 43, this volume) Today the video game industry is one of the fastest growing forms of entertainment to date According to the Digital Interactive Software Association (IDSA), revenue from computer and console games practically doubled from $3 2 billion in 1994 to $6 02 billion in 2000, when more than 219 million games were sold (IDSA, 2000) To put this into perspective the number of people in the United States who played video games during 2000 was 5 times that of those who went to America's top five amusement parks combined, and 2 times as many as those who attended all Major League Baseball games (IDSA) Thirty to forty percent of homes in the United States own a console gaming system with another 10 to 20% renting or sharing consoles (Cassell & Jenkins 2000). These statistics do not even take into account the international importance of video games

One of the misconceptions about video games is that they are currently played only by a small segment of the population that is, younger boys This misconception may stem from a variety of things from controversial beliefs about boys and men possessing an innate mathematical superiority (e g Williams, 1987) to statements that the electronics industry has traditionally been marketed to boys (e g Chaika 1996) to the fact that one of the most successful companies in the gaming industry (i e , Nintendo) has indeed targeted the younger male population Nintendo's positioning has traditionally been targeted toward youth and adolescent age groups which have clearly been demonstrated by previous marketing campaigns and promotions on such things as cereal boxes (Herz, 1997; Provenzo, 1991). However, the average age of those who play video games is 28 (IDSA 2000) Also, there is a large movement in the gaming industry that targets the female market as well (Gorriz & Medina 2000) Purple Moon was founded in 1996, and Girl Games Inc. was founded in 1993, both as an effort to make games dedicated for girls (Cassell & Jenkins 2000) A recent survey done by Peter D Hart Research Associates claims that three in five Americans say they play computer or video games and that 43% of them are female gamers (IDSA)

Currently, many would argue that games play a large role in pushing research and technology forward This includes such areas hardware innovations (graphics cards processors sound cards; e g., Kroll 2000) research on the relationships between artificial intelligence and synthetic characters or computer graphics (e g , Funge, 2000; Laird, 2001) and physics (e g , Hecker 2000) These are all areas that previously were confined to engineering and computer science communities

The gaming industry is becoming as widespread and popular as computers and televisions and is also matching the same levels of complexity and advanced technology The difference though, is that user-centered design (UCD) principles have not reached the same level of usage or awareness in game design and development as it has in other electronic applications To understand the differences and unique challenges brought forth in games one must appreciate how games fit in relation to other software applications in which UCD principles are present such as in productivity applications

DEFINING GAMES

There are many ways to define a class of objects One popular approach is to define an object by stating the principles that differentiates one class of objects from another Another is to define a class of objects by enumerating the items in the class Here we do both First, we distinguish games from productivity applications then we list several types of games

Games Versus Productivity Applications

The distinction between games and productivity is often clear and straightforward in practice As with some other forms of art and entertainment we know a game when we see one However articulating a clear and succinct set of principles that capture the differences and similarities between a game and a productivity application is not as straightforward In other words the distinction is simple in practice but more difficult in principle However shedding some light on the difference between computer games and productivity may help readers understand both the choice of methods and differences used in

their application It may also deepen one s understanding of productivity applications For example, the goal of iterative usability testing on games is to reduce the obstacles to fun, rather than the obstacles to accomplishment (as in productivity applications) Many of the issues of concern in games are virtually identical to those used on other applications in which the obstacles to fun are often similar to obstacles in the way of productive work These include confusing screen layouts misleading button labeling or an inconsistent model of use (to mention a few) Thus, there are both principles and methods that can be successfully applied both to games and productivity applications At the same time there are fundamental differences between them which are sometimes easy to see and articulate and sometimes quite subtle Let us consider briefly some of these similarities and differences

Process Versus Results

The purpose of games is different from the purpose of productivity applications At their root productivity applications are tools Similar to tools the design intentions behind productivity applications are to make tasks easier and quicker to reduce the likelihood of errors to increase the quality of the result and to extend domains of work to larger and larger populations In this sense a word processor or spreadsheet differs only from a powered woodworking tool in terms of its complexity and domain of application Broadly speaking both the motivation for design and usability work on productivity applications such as a word processor is to make better documents—more quickly, more easily and with fewer errors—and to extend these capabilities to the widest number of people Similarly a power mitre box cuts wood with more precise angles more quickly and reliably than a hand mitre box, which in turn makes work more precise faster and more reliable than a hand saw The focus of design and usability is to produce an improved product or result

At their root games are different Games are intended to be a pleasure to play Ultimately games are like movies and literature They exist to stimulate thinking and feeling This is not to say that word processors or other tools cannot be a pleasure to use or that people do not think or feel when using them but that is not their design intention By and large the outcome goal of games (i e winning) serves to enhance the pleasure of participation or the process of playing Thus the goal of both design and usability when applied to games is creating a pleasurable process This fundamental difference leads us to devote more of our effort to collecting user evaluation of games (as opposed to strictly performance) than we would if we were working on productivity applications, where more of our work would measure accomplishing tasks (or productivity)

Defining Goals Versus Importing Goals

Games define their own goals The goals of productivity applications are defined by the external environment independent of the application itself

In some cases, the design goal of productivity applications is simple—complete the checkout process in buying a product For more general applications, such as word processors or spreadsheets the goals can be complex and variable In these situations one of the most difficult design and usability problems is mapping the interface to the users' goals When an application presents goals (e g , Microsoft Works), it presents them in a simple and straightforward way—as a choice of possible work products (e g Letter and Labels, Business and Address Cards Cover Letters; see Fig 46 1)

Games create an artificial world in which the objectives are set entirely within that world For example, in chess the objective is to capture the opponent s king The terms *opponent* and *king* have a different meaning in this situation than they do in other contexts *Capture* also has a specific meaning here, again different than other contexts In some sense this simplifies the problem of the game designer and of the usability practitioner User's goals are easily identified because they are designed in as the purpose of the game The resulting concern for game design is to convey the goal to the user in a clear and straightforward way This is one factor that increases the importance of tutorials and training simulations in games Often games simply present the user with a goal using an introductory and hopefully entertaining cinematic that establishes the background story and thereby explicitly or implicitly presents the user with a high-level goal In addition, many games will provide 'briefings' to describe a 'mission' that outlines immediate goals of this part of the game and may suggest some of the obstacles that a gamer may face

Few Alternatives Versus Many Alternatives

Whereas choices are limited for productivity applications competition in the gaming industry is fierce.

Pretend that you have to write a book chapter For that you have limited options: handwriting dictation a typewriter or various software applications such as Microsoft Word or Corel WordPerfect In practice however users may be even more constrained. Publishers may accept a book chapter only in a given format that excludes any handwritten chapters regardless of its merit.

Competition within the games industry is intense more intense than any other software domain It is conceivable that there are more games produced each year than versions of word processors produced ever Games also naturally compete with other forms of entertainment Take for example, the popular PC fantasy role-playing game *Diablo* (Vivendi, 2000) by Blizzard software *Diablo* must compete with other role-playing games with other kinds of games in general with other entertainment options and with more practical needs In a situation in which you want an engaging puzzle that will occupy your mind for a few hours and ultimately give you a sense of accomplishment you could consult the *New York Times* crossword puzzle argue politics with your best friend or watch a mystery film Or you could stock up with health potions and attack demons in *Diablo* Games must grab your attention and each game must be noticeably different than the last

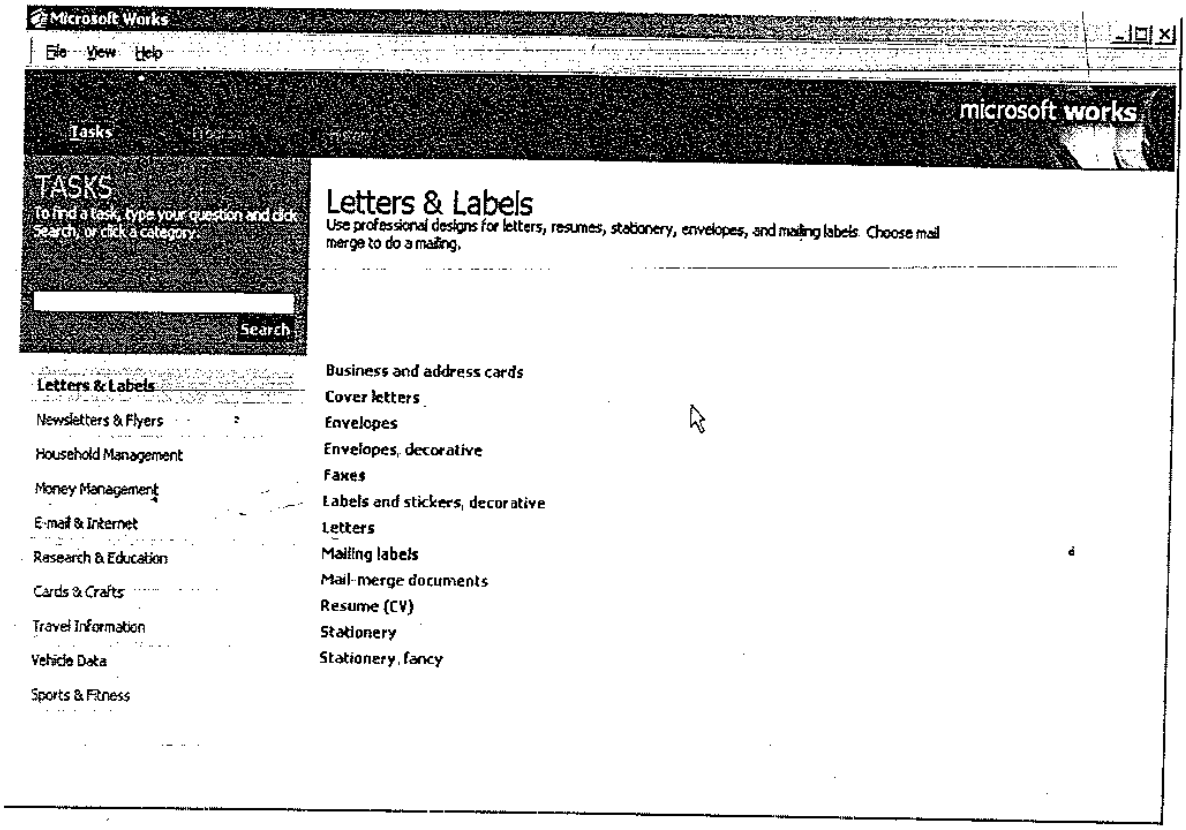

FIGURE 46.1. An example of how a productivity application (Microsoft Works) presents the user with goals.

Being Consistent Versus Generating Variety

Games must provide a variety of experience. In contrast, a user's experience throughout a productivity application strives for consistency.

Variations in users' experience with productivity applications come from differences in their goals. For example, when you need to produce a document with an index, you learn how to do indexing in a word processor. Designers have a goal of making these new functions work in a way that is consistent with the rest of the application while providing the functionality required by the task. In other words, technology can provide significant enhancements for users by presenting new functionality, such as automating routine and tedious tasks. However, an important goal of design is to generate consistency in the user experience.

Games must be different every time because people get bored easily. An important component in this is challenge. Each time the game is played, the user should be invited to learn new rules, or try new strategies to achieve a goal. This learning and exploration is part of the game, which implies that users should have a different experience every time they play.

Imposing Constraints Versus Removing or Structuring Constraints

Games deliberately impose constraints, whereas productivity applications attempt to remove them. Constraints in productivity applications are almost never designed in to intentionally create difficulties. They are most often a reflection of an unsolved (and probably unanticipated) design problem, or a result of the domain in which one is working. A number of successful design approaches attempt to minimize or structure the constraints inherent in task demands. Wizards are a good example. To make a graph, one must select the data to be graphed, choose the type of graph, map the data to axes, and determine display appearance and labeling. This is done effectively by the wizards that structure the task and illustrate the choices to make at each point. The goals of design, usability, and technology are to remove unnecessary constraints because they stand between the user and the result to be obtained, or at least to minimize externally imposed constraints that arise from the interaction between user goals and the environment.

Games, however, often impose artificial constraints because they contribute to the fun of the game. To illustrate this difference, consider the following example. One of our games usability leads, Bill Fulton, has said that the easiest game to use would consist of one button labeled *Push*. When you push it, the display says "YOU WIN." The irony of this example is that this game would have few, if any, usability issues, but it clearly would not be much fun. At the same time, making a game difficult does not necessarily make it fun. Consider chess. The constraints consist of rules imposed on moving pieces. However, these constraints combine to yield a complex set of possible combinations of movement that vary through the opening, middle, and end portions of the game. Chess would not be "improved" by introducing more constraints, such as making it harder to distinguish

the pieces or by imposing a new set of bizarre and apparently inconsistent rules of movement such as 'after the 20th move, any bishop left of center but past the midpoint of the board will now be able to move seven spaces horizontally (diagonal movement is unchanged)' In short, games must impose consistent constraints on user behavior while making the process of attaining the defined goals interesting and challenging

Function Versus Mood

Although productivity applications use sound and graphics to convey function, games create environments through the use of sound and graphics

The use of sound and graphics in productivity applications is relatively small compared to games Buttons may have drop shadows to produce a three-dimensional-effect in an attempt to show that the button affords clicking Icons may be placed on buttons which afford software functions such as the disk icon to represent the save function A 'click' sound may be used to provide feedback that a certain function has occurred such as in selecting a drop-down menu Even at higher level, overall look of the interface may produce an identity (e g buttons that appear to be illuminated from within, or the 'lozenge' look of Macintosh OS X)

In contrast most games attempt to create an 'environment' through the use of sound and graphics which is integral to the game For example, House of the Dead II (1999) creates a dark spooky environment through the use of rain effects compelling depictions of an abandoned city populated by zombies and a few straggling civilians to be saved (or not) In addition music and sound effects contribute to the environment by creating a sense of tension

View of Outcome Versus View of World

Productivity applications rarely have a point of view or perspective In contrast, most games need to assume a point of view (first person third person Eye of God) or perspective

Productivity applications may offer alterative views (normal outline print, and Web) of the same data Two of these views (print and Web) are intended to help envision the final product The others provide ways to see more of the document (normal) or to manipulate the data in specialized ways (outline) It is also common for productivity applications to offer zoom or degrees of magnification allowing the user to see more or less detail in exchange for a narrower or wider field of view What is uncommon, is introducing the notion of perspective

Games often provide perspective or a point of view In some games it attempts to mimic perceived optical flow (i e first-person shooters) or center of outflow to specify direction of locomotion With the increased use of three-dimensional environments, issues often arise in games that are very similar to those experienced in a virtual environment community (e g Barfield & Williges 1998; Smart 2000) Game designers must appreciate the difference however, between trying to simulate reality and creating an environment that users perceive as reality (cf Stoffregen Bardy Smart & Pagulayan in press) Stanney (chapter 31 this volume) provided an in-depth discussion of many design and implementation strategies when dealing with three-dimensional environments

Organization As Buyer Versus Individual As Buyer

Many productivity applications are bought by organizations Games are almost always bought by individual users Some productivity applications are sold to home users who are interested in using the same tools both at work and home, but by and large businesses buy productivity applications In making these purchases, key influential purchasers often spend time evaluating learning about and prejustifying their decisions before ever purchasing a product for their business This changes the marketing of products to focus on direct sales relationships with the influential purchasers rather than consumer advertising

In contrast game makers must sell to consumers rather than influential purchasers because the majority of game purchases are made by consumers This market is large and diverse which makes it difficult to directly communicate with consumers Because of this, key features must not only be usable by relatively untrained people they must be visible explainable and appealing to purchasers using minimal media (i e , box covers and 30-second advertisements)

Form Follows Function Versus Function Follows Form

Users of productivity applications tend to be cautious about adopting innovation while game buyers tend to welcome innovation Users of productivity applications tend to be reluctant to embrace entirely new and innovative designs The primary interface of many productivity applications has been relatively stable for many years (take the graphical user interface for example found in current Macintosh and Windows operating systems) Although there are many proffered explanations for this, the most plausible one is that current designs suit user needs relatively well and none of the innovations offered over the years have appeared compelling to users Innovation does not necessarily equate with greater functionality or ease of use

Games are compelled by economics to push the technological envelope and to show off exciting features of the system for which they are made Like other forms of entertainment games incorporate new and novel features in the hope of attracting a wider audience and not losing their existing audience Although this is often very exciting for consumers it can also lead to problems This issue is discussed further later in the chapter

Standard Input Devices Versus Novel Input Devices

There is wider variation in game input devices than in productivity applications Most productivity applications use two input devices, a pointing device and a typing interface Use of pointing devices or touchscreens can be taught in short and simple tutorials Typing is a skill that can be learned independently of the applications In both cases there is not as much variability in input devices when using productivity applications

FIGURE 46 2 This is a sample of different input devices used in games from the traditional keyboard and mouse to different console gamepads

Games, however, often have unique input devices that are especially designed for a particular games genres, or platforms. Gamepads, joysticks, steering wheels, aircraft yolks simulated guns, and microphones all solve particular design problems but often end up creating problems For example one could argue that using a joystick should be more satisfying than using a keyboard for controlling an airplane flying through the air However, it can also be argued that using that same joystick is not very efficient for inputting text on the screen to save your high score Particular game genres and platforms (see section below) complement certain input devices to make games more or less fun but also create new issues not seen in productivity application input devices Figure 46 2 is a sample of different types of input devices

Summary

We believe the above ten characteristics summarize most of the ways that games differ from productivity applications Many of the characteristics mentioned are not solely unique to games however Many of these characteristics can also be found in areas such as virtual environments, Web applications, or home software applications such as home publishing software for example For purposes of this chapter however all these characteristics need to be taken into account and judged appropriately when discussing a game Later we discuss how these differences create particular challenges for both the design and evaluation of games

These principles may appear to be overly general but this is because as a class games are very diverse Now we review some common classification schemes for games and enumerate the various game types

TYPES OF GAMES

PC Versus Console

Personal computer (PC) versus console (Cassell & Jenkins 2000) is one of the simplest classifications that can be made. It differentiates those games played on a PC from those played on a console (e g Sony PlayStation Sega Dreamcast Nintendo GameCube Microsoft Xbox) There are many characteristics that are associated with each platform For example the PC offers different control mechanisms (keyboard mouse and other possible game controls) Games usually eschew a keyboard in favor of a controller that provides a series of buttons and joysticks.

Crawford (1982) divides games into even finer categories: arcade (coin-operated) console PC mainframe and handheld These are all important distinctions For the purposes of this chapter however we often refer to the PC versus console distinction

Game Genre Classification

The genres of interactive games available are as diverse as those who play them Some of the more consistent genres used in the games industry include Action Adventure/Role-Playing, Sports and Family Table 46 1 lists the best selling genres in 2001

TABLE 46.1 Best Selling Game Genres in 2001 for Console Games and PC Games (IDSA 2002)

Console Game Genres	PC Game Genres
Sports	Strategy
Action	Child
Strategy/RPG	Family
Racing	Action
Fighting	RPG
Other Shooters	Sports
Family	Driving
1st Person Shooters	Simulation
Child	Adventure

Note: RPG = role-playing game

Interactive Digital Software Association (2002). Essential facts about the computer and video game industry. Washington, DC: Interactive Digital Software Association.

PRINCIPLES AND CHALLENGES OF GAME DESIGN

Having differentiated games from other applications, we can look at some of the unique issues in game design and evaluation.

Identifying the Right Kind of Challenges

Games are supposed to be challenging. This requires a clear understanding of the difference between good challenges and frustrating usability problems. Most productivity tools struggle to find a balance between providing a powerful enough tool set for the expert and a gradual enough learning curve for the novice. However, no designer consciously chooses to make a productivity tool more challenging. The ideal tool enables users to experience challenge only in terms of expressing their own creativity. For games, learning the goals, strategies, and tactics to succeed is part of the fun.

Unfortunately, it is not always clear which tasks should be intuitive (i.e., easy to use) and which ones should be challenging. Take a driving game, for example. We are willing to argue that it is not fun to have to discover how to make your car go forward or turn, but that does not mean that learning to drive is not part of the fundamental challenge in the game. Although all cars should use the same basic mechanisms, it may be fun to vary the ways that certain cars respond under certain circumstances. It should be challenging to identify the best car to use on an oval, yet icy, racetrack as opposed to a rally racing track in the middle of the desert.

Learning and mastering these details may involve trial, error, insight, and logic. Furthermore, the challenge level in a game must gradually increase to maintain the interest of the player. If these challenge-level decisions are made correctly, players will have a great time learning to overcome failure. Defining where the basic skills should stop and the challenging skills should start is difficult, so input from users becomes necessary to distinguish good challenges from incomprehensible design.

Addressing Different Skill Levels

Different strategies can be used to regulate the challenge level in the game to support both skilled and unskilled players. Players of all skill levels must do more than simply learn the rules; each one must experience a satisfying degree of challenge. Given that both failure and success can become repetitive quickly, games must address the problem of meeting all players with the correct level of challenge. Although this is complicated in single-player games, it is an even more daunting task when players of different skills are pitted against one another. Tuning a game to the right challenge level is called game balancing.

There are many ways to balance the difficulty of the game. The most obvious is to let players choose the difficulty themselves. Many games offer the choice of an easy, medium, or hard difficulty level. Although this seems like a simple solution, it is not simple to identify exactly how easy the easiest level should be. Players want to win, but they do not want to be patronized. Too easy is boring; too hard is unfair. Either perception can make a person cease playing.

The environments, characters, and objects in a game provide another possibility for self-regulation. Most games will offer the player some choices regarding their identity, their opponents, and their environment. The better games will provide a variety of choices that allow users to regulate the difficulty of their first experiences. With learning in mind, it is not uncommon for the novice player to choose a champion football team to play against a weak opponent. As long as the player can distinguish the good teams from the bad ones and the teams are balanced appropriately, users will be able to manage their own challenge level.

Another obvious approach to varying skill levels is to require explicit instruction that helps all users become skilled in the game. You might imagine a tutorial in which a professional golfer starts by explaining how to hit the ball and ends by giving instruction on how to shoot out of a sand trap onto a difficult putting green. Instruction, however, need not be presented in a tutorial. It could be as direct as automatically selecting the appropriate golf club to use in a particular situation with no input from the user, similar to the notion of an adaptive interface where the interface provides the right information at the right time.

Some games take it even further by identifying the skill level of the player and regulating the game difficulty appropriately. In this situation, instruction can be tuned to the skill level of the player by associating it with key behavioral indicators that signifies that the player is having difficulty. If the game does not detect a problem, it does not have to waste the player's time with those instructions.

Productivity tools have implemented similar problem-identification features but often with mixed success because of the open nature of tasks in most productivity applications. Good game tutorials have succeeded by setting clear goals and completely constraining the environment. Doing so focuses the user on the specific skill and simplifies the detection of problem behavior. Other lessons from game tutorial design will be described in later sections of this chapter.

Another in-game approach to regulating the difficulty level requires adjusting the actual challenge level of the opponents

during the game Evaluating the success of the player and adjusting the opponent difficulty during the game is often called rubber-banding This tactic is frequently used in racing games These games monitor the progress of the player and modify the skill of the opponents to ensure that each race is a close one If the player is not a great driver the game can detect that and slow down the computer opponents

Rubber-banding works the same way in multiplayer games However, restricting players can feel unfair to the restricted player and patronizing to the unskilled player The popular football game NFL Blitz (Midway Games 1997) uses this technique in both single and multiplayer modes to keep the challenge level continuously high Even though this may seem like a good solution, there can be a downside When players perform skillfully, their performance is moderated by computer-generated bad luck and enhanced opponent attributes The likelihood of fumbling throwing an interception, or being sacked increases as a player increases their lead over their opponent Most people would prefer to play a competitive game (and win) than to constantly trounce a less skilled opponent But, at the same time, overdeveloped rubber-banding can cheat a skilled player out of the crucial feeling of mastery over the game

The key for the game designer is to think of ways to maintain challenge reward and progress for the unskilled player without severely hampering the skilled player A final approach focuses on providing tools that maximize the ability of the trailing player to catch up with the leading player One interesting and explicit example of this is found in Diddy Kong Racing (1997) In this game the racer can collect bananas along the roadway Each banana increases the top speed of your car The player can also collect missiles to fire forward at the leading cars Each time you hit a car with a missile it not only slows the car's progress but it jars loose several bananas that one (as the trailer) can pick up Thus trailing players have tools that they can use to catch the leaders even if the leader is not making any driving mistakes The chief distinction between this and rubber-banding is that the game is not modifying skills based on success Instead, the rules of the game provide the trailing player with known advantages over the leader

Players Must Be Rewarded Appropriately

Explicit or slow reinforcement schedules may cause users to lose motivation and quit playing a game Because playing a game is voluntary games need to quickly grab the user's attention and keep them motivated to come back again and again. One way to accomplish this is to reward players for continued play Theories of positive reinforcement suggest behaviors that lead to positive consequences tend to be repeated Thus, it makes sense that positive reinforcement can be closely tied to one s motivation to continue playing a game. However, it is less clear which types of reinforcement schedules are most effective

Although it s not necessarily the model that should be used for all games research suggests that continuous reinforcement schedules can establish desired behaviors in the quickest amount of time (Steers & Porter, 1991) Unfortunately once continuous reinforcement is removed, desired behaviors extinguish very quickly Use of partial reinforcement schedules take longer to extinguish desired behaviors but may take too long to capture the interest of gamers Research suggests that variable ratio schedules are the most effective in sustaining desired behaviors (Jablonsky & DeVries, 1972) This kind of schedule is a staple of casino gambling games in which a reward is presented after a variable number of desired responses Overall, there is no clear answer Creating a game that establishes immediate and continued motivation to continue playing over long periods of time is a complex issue

Another facet of reinforcement systems that may impact enjoyment of a game is whether the player attributes the fact that they have been playing a game to extrinsic or intrinsic motivations Intrinsic explanations for behavior postulate that the motivators to perform the behavior come from the personal needs and desires of the person performing the behavior Whereas extrinsically motivated behaviors are those that people perform to gain a reward from or please other people In research on children s self-perceptions and motivations Lepper Greene and Nisbett (1973) discovered that children who were given extrinsic rewards for drawing were less likely to continue drawing than those who had only an intrinsic desire to draw The conclusion that they drew is that children perceived their motivation to draw as coming from extrinsic sources and thus discounted their self-perception that they liked to draw

The same may be true of reward systems in games (Lepper & Malone 1987; Malone 1981) To a certain degree all reinforcement systems in games are extrinsic because they are created or enabled by game developers but some reward systems are more obviously extrinsic than others For instance, imagine the following rewards that could be associated with combat in a fantasy role-playing game (RPG) The player who slays a dragon with the perfect combination of spell casting and sword play may acquire the golden treasure that the dragon was hoarding In this situation, the personal satisfaction comes from being powerful enough to win and smart enough to choose the correct tactics The gold is an extrinsic motivator The satisfaction is intrinsic By analogy from Lepper et al s (1973) research feelings of being powerful and smart (intrinsic motivators) are more likely to keep people playing than extrinsic rewards

Collecting and Completing

The chief goal of many games is to acquire all of the available items rewards, or knowledge contained in the game It may seem obvious that all gamers like to win Most racing games allow players to acquire the ability to race new cars and new tracks by winning the basic tracks These games progress through a series of challenges unlocking new challenges and opportunities along the way The ultimate goal is to race with all of the cars and tracks that have been included until new options are exhausted and the game has finally been beaten In one sense, the goal of the game is to experience everything that is available until all options have been completely identified or uncovered used, and mastered

This motivation is even more fully developed in games such as Pokemon Crystal (2000), where the central challenge is to acquire as many of the Pokemon characters as you can and learn all of their skills well enough to outsmart your opponent at selecting the right characters for a head-to-head competition. Not coincidentally the catch phrase for the Pokemon Crystal game is 'Gotta catch 'em all!'

Beyond Interactivity

Game design is not just about adding interactive elements to a narrative. We purport that game designers must use the tools of their medium to channel the thoughts and feelings of their players. Whereas observers are given the freedom of interpretation through their own unique perspective, games offer users more opportunities to change the actual elements and sequences in a story. There have been many great games that are actually linear in structure. In fact, most games are linear in structure, but great games hide this fact from the player. The player feels like they are in control of what happens next when in fact they are not.

Although most games involve some sort of interactivity, there are as many flavors of these elements as there are game genres. In some cases, authors have suggested that games are inherently more immersive or involving than other media because of the interactive elements (Au, 2001; Lieberman, 1998; Stevenson, 2001). The implication is that button pressing requires or enables a different degree of mental involvement than that required of moviegoers sitting passively in their seats. We propose that button presses and other mechanisms for providing overt interaction are really just physical manifestations of the same types of mental processes that occur during the experience of any work of entertainment or art. Every director or author knows that the audience interacts with their works in a very real way. *The Usual Suspects*, *Casablanca*, and *Psycho* are clear examples of movies that are scripted specifically to involve the user in conscious problem solving by creating expectations and then violating those expectations.

Thinking of interactivity as a wholly unique phenomenon to games can interfere with effective game design if it causes the designer to think about actions and sequences rather than empathizing with the way that their user will be thinking and feeling as they progress through the game. In most cases, it is not enough to merely keep the player busy at pressing buttons. The designer has to think about the causes and consequences of the player's actions, because that is where the player's attention will be focused. Games give you the opportunity for increased identification, distraction, and physical action. But the most important part of interactivity is not clicking a mouse or moving a joystick. The task of the game designer is to empathize with the viewer and carefully construct an experience that causes them to think clever thoughts and feel profound emotions.

Should There Be a Story?

A story line serves to provide meaning and importance that increases significance, tension, and motivation to succeed in the action sequences. There are those in the games industry that propose that many games neither have nor need a story. It is our contention that all games do have a story and that game designers should approach story line in a similar fashion to the way that authors approach it. The key to understanding narrative in games is to realize that story lines may be embedded in the game, or they may emerge in the course of playing a game.

Most traditional narratives (i.e., books, movies, plays) have a three-act structure. The first act is used to introduce the context and central conflicts. Its purpose is to grab the user's attention and hook them into caring about the characters and their problems. It should make the viewer an active participant and problem solver. The second act is a struggle for success in the primary conflict. At the end of the second act, the conflict changes in some fashion. Success is snatched from the protagonist's hands. Then, the final act often allows the protagonist to regain success. Engagement in the story line is required to get the viewer to suspend their disbelief and become emotionally involved in the trials of the main characters. This immersion is tested each time there is a transition or new element introduced. At each change point, there is a risk to lose viewers, but an opportunity to deepen their involvement in the story.

When most consumers think about story, they think about embedded story lines (Levine, 2001). In games, embedded story lines are often communicated by stopping the game play to watch computer-generated movies. In some games, such as Final Fantasy IX (2000), these embedded stories form a central part of the appeal of the game. In others, they are nearly an afterthought. Embedded narrative need not be presented as computer-generated cut scenes. There may be elements of the story communicated by the action sequences, interactions with characters, or the rewards of the action sequences. This is the easiest form of stories for consumers to recognize and evaluate.

But judgments about this type of narrative may not reflect all of the actual story elements in the game. Ken Levine (2001) pointed out that much of the narrative in a game emerges in the interaction between players throughout the course of the game. This is especially true of a multiplayer game. The story in a multiplayer game often comes not from the game, but from the struggle of multiple opponents. As Levine described it, the story is generated by what happens when people replay an abstract narrative structure and a strict set of rules and possibilities within a novel set of circumstances. In this way, a game of chess or baseball can have the same basic elements every time, but still create opportunities for the players to create epic tales and battles in their minds. The challenge for the game designer is to create a set of rules and opportunities that maximize the ability of users to create unique strategies and anticipate the strategies of others. In this way, the gamers themselves can become the protagonists.

Technological Innovation Drives Design

There is a great deal of pressure on designers to use new technologies that may break old interaction models. As mentioned

earlier many would argue that games play a large role in pushing research and technology forward in areas that were previously were confined to engineering and computer science communities On the other side of the coin, in addition to appealing to consumers, games act as showcases for the hardware technology on which they are run Although this is often exciting for consumers it can also lead to problems In the rush to innovate technologically, game makers may not make it a priority to create usable interfaces In some cases, there may even be strong incentives for game makers to 'spruce up' familiar interfaces and break rules of consistency

Stating that great games are the only thing required to sell the console systems on which they are played would be a dubious claim The success of a console system depends entirely on whether the games that are played on a particular console are noticeably different from alternative technologies Because of the rapid pace of innovation the costs of producing game hardware are quite high Unfortunately you cannot sell any games if users do not have access to that video game system This creates a somewhat dangerous 'chicken-and-egg' situation It would cost the consumer an enormous amount to buy the latest video game hardware to cover production costs But no games will sell if consumers do not have that hardware So Nintendo Sony Microsoft, and a handful of other major game system creators sell their hardware at a loss for much of its production run They make up the loss by collecting royalties from third parties and by publishing games under their own brand

In his book about the success of the Sony PlayStation, Asakura (2000) noted a series of technological and business decisions that affected the competition between Sony and Nintendo One example from Asakura's book illustrates how a business decision may influence both customer appeal and usability Asakura noted that the decision to use relatively low cost CD-ROM technology allowed PlayStation games to be cheaper by reducing the total cost of goods and reducing significant planning and distribution problems Previous Nintendo games were produced on cartridges with circuit boards This enabled Nintendo games to minimize loading times to transfer information from the game to the system It also allowed Nintendo games to save information directly on to the cartridge Both of these characteristics are positives for consumers because they reduced delays and hassles that were unrelated to playing the game itself Ultimately Asakura argued that these consumer benefits were overshadowed by the bottom line that it cost less for consumers to acquire games, and developers had a higher profit margin on each sale So this suggests that usability advantages alone are not sufficient for success

Perceptual-Motor Skill Requirements

The way that functions are mapped onto available input devices can determine the success or failure of a game A crucial part of the fun in many games comes from performing complex perceptual-motor tasks Although today's arcade-style games include increasingly more sophisticated strategic elements a core element of many games is providing the ability to perform extremely dexterous yet satisfying physical behaviors These behaviors are usually quick and well-timed responses to changes and threats in the environment If the controls are simple enough to master and the challenges increase at a reasonable difficulty these mostly physical responses can be extremely satisfying (Csiksentmihalyi 1990)

Problems can arise when games require unfamiliar input devices This is a common complication in console game usability research because new console systems usually introduce new input device designs unique to that system (see Fig 46 2) Furthermore game designers often experiment with new methods for mapping the features of their games to the unique technical requirements and opportunities of new input devices Unfortunately this does not always result in a better gameplay experience

IMPORTANT FACTORS IN GAME EVALUATION

Typical usability outcome measures such as task times and errors can be used to evaluate certain aspects of games but it is also necessary to measure users subjective experiences and attitudes about games This section explores some of the subjective attributes that are common to many games

Overall Quality (Also Known As Fun)

Most game genres are subtly different in the experiences that they provoke It may seem obvious that the point of game design is making a fun game Some games are so fun that people will travel thousands of miles and spend enormous amounts of money to participate in gaming events However we would like to propose a potentially controversial assertion: The fundamental appeal of some games lies in their ability to challenge to teach to bring people together or to simply experience unusual phenomena Likewise the definition of fun may be different for every person When you play simulations games, your ultimate reward may be a combination of learning and mastery. When you play something like the MTV Music Generator (1999) your ultimate reward is the creation of something new When you go online to play card games with your uncle in Texas you get to feel connected Flight Simulator (2000) lets people understand and simulate experiences that they always wished they could have Although these may be subcomponents of fun in many cases there may be times when using 'fun' as a synonym for overall quality will lead to underestimations of the quality of a game A fun game is often synonymous with a good game, but researchers are warned to wisely consider which measures best suit the evaluation of each game that they evaluate

Ease of Use

The ease of use of a game's controls and interface is closely related to fun ratings for that game Think of this factor as a gatekeeper on the fun of the game If users must struggle or

cannot adequately translate their intentions into in-game behaviors they will become frustrated This frustration can lead users to perceive the game as being unfair or simply inaccessible (or simply not fun) Thus, it becomes clear why usability is important in games Ease of use should be evaluated with both usability and attitude-measurement methodologies

Starting a Game. Starting the kind of game that the user wants is an easily definable task with visible criteria for success This is something we measure in our usability laboratories Although designers often take game shell (the interface used to start the game) design for granted a difficult or confusing game shell can limit users discovery of features and impede their progress toward enjoying the game The most immediate concern for users can be starting the kind of game that they want to play Games often provide several modes of play When the game shell is difficult to navigate users may become frustrated before they have even begun the game For example we have found that many users are unable to use one of the common methods that sports console games use to assign a game controller to a particular team This has resulted in many users mistakenly starting a computer versus computer game Depending on the feedback in the in-game interface users may think that they are playing when in fact the computer is playing against itself! In these cases users may even press buttons, develop incorrect theories about how to play the game, and become increasingly confused and frustrated with the game controls The most effective way to avoid these problems is to identify key user tasks and usability test them

Tutorials or Instructional Missions As mentioned earlier tutorials are sometimes necessary to introduce basic skills to play the game In this situation, instructional goals are easily translated into the usability labs with comprehension tasks and error rates

One of the risks of not testing tutorials or instructional missions is inappropriate pacing, which can often result from an ill-conceived learning curve at the start of the game Many games simply start out at too difficult a challenge level This is an easy and predictable trap for designers and development teams to fall into because when designers spend months (or even years) developing a game they risk losing track of the skill level of the new player. A level that is challenging to the development team is likely to be daunting to the beginner.

Unfortunately the reverse can also be troubling to the new user Faced with the task of addressing new players, designers may resort to lengthy explanations. Frequently developers will not budget time to build a ramped learning process into their game Instead they may realize late in the development cycle that they need to provide instruction. If this is done too abruptly, the learning process can end up being mostly explanation and, to be frank, explanation is boring The last thing that you want to do is to bore your user with a long-winded explanation of what they are supposed to do when they get into your game It is best to learn in context and at a measured pace or users may just quit the game

A positive example is the first level of *Banjo Kazooie* (2000) At the start the player is forced to encounter a helpful tutor and listen to a few basic objectives Then they must complete some basic objectives that teach some of the basic character abilities Much of the tutorial dialogue may be skipped but the skills necessary to continue must be demonstrated In this way the game teaches new skills but never requires tedious instruction The player learns primarily by doing All of this is done in the shadow of a visible path onto the rest of the game so users never loses sight of where they need to go

In-Game Interfaces In-game interfaces are used primarily to deliver necessary status feedback and to perform less frequent functions We measure effectiveness with more traditional lab usability testing techniques and desirability with attitude measurements such as surveys (see next section for example)

Some PC games make extensive use of in-game interfaces to control the game For example, simulation and real-time strategy (RTS) games can be controlled by keyboard and mouse presses on interface elements in the game Usability improvements in these interfaces can broaden the audience for a game by making controls more intuitive and reducing tedious aspects of managing the game play In-game tutorial feedback can make the difference between confusion and quick progression in learning the basic mechanisms for playing In this situation, iterative usability evaluations become a key methodology for identifying problems and testing their effectiveness (see next section for example)

Many complex PC and console video games make frequent use of in-game feedback and heads-up displays (HUD) to display unit capabilities and status For example, most flight combat games provide vital feedback about weapons systems and navigation via in-game displays Without this feedback, it can be difficult to determine distance and progress toward objectives unit health and attack success This feedback is crucial for player learning and satisfaction with the game With increasing game complexity and three-dimensional movement capabilities, these displays have become a crucial part of many game genres Usability testing is required to establish whether users can detect and correctly identify these feedback systems Attitude measurement is required to assess the utility of these features and gauge whether users have a satisfying amount of status feedback

Mapping Input Devices to Functions A learnable mapping of buttons keys or other input mechanisms to functions is crucial for enjoying games We measure effectiveness with usability techniques and desirability with attitude measurements Without learnable and intuitive controls the user will make frequent mistakes translating their desires into onscreen actions We have seen consistently that these kinds of mistakes are enormously frustrating to users because learning to communicate one's desires through an eight-button input device is not fun The selection of keys buttons, and other input mechanisms to activate particular features is often called control mapping. Players tend to feel that learning the control-mapping is the most basic part of learning the game It is a stepping stone to getting to the fun tasks of avoiding obstacles developing strategies and blowing things up

In contrast with other ease-of-use issues evaluating the control mapping may involve as much subjective measurement as behavioral observation. Button presses are fast, frequent and hard to collect automatically in many circumstances. Furthermore, problems with control mappings may not manifest themselves as visible impediments to progress, performance or task time. Instead, they may directly influence perceptions of enjoyment, control, confidence or comfort. Differences in experience levels and preferences between participants may create significant variation in attitudes about how to map the controls.

Dissatisfaction with the controller design can also be a central factor that limits enjoyment of all games on a system. For example, the results of one whole set of studies on the games for a particular console game system were heavily influenced by complaints about the system's controller. Grasping the controller firmly was difficult because users' fingers were bunched up and wrists were angled uncomfortably during game play. Ratings of the overall quality of the games were heavily influenced by the controller, rather than the quality of the game itself.

Because of these concerns and the importance of optimizing control mappings, we recommend testing them with both usability and attitude assessment methodologies.

Challenge

Challenge is distinct from ease of use and is measured almost exclusively with attitude assessment methodologies. This can be a critical factor to the enjoyment of a game, can be highly individualized, and is rightly considered subjective.

Consumers may have difficulties distinguishing the 'appropriate' kinds of challenge that result from calculated level and obstacle design from the difficulty that is imposed by inscrutable interface elements or poor communication of objectives. In either case, the result is the same. If not designed properly, the player's experience will be poor. Thus, it is up to the user-testing professional to make measurement instruments that evaluate the appropriateness of the challenge level independent of usability concerns.

Pace

We define pace as the rate at which players experience new challenges and novel game details. We measure this with attitude measurement methodologies.

Most designers recognize that appropriate pacing is required to maintain appropriate levels of challenge and tension throughout the game. You might think of this as the sequence of obstacles and rewards that are presented from the start of the game to the end. One group at Microsoft uses a critical juncture analogy to describe pacing. As a metaphor, they suggest that the designer must attend to keeping the user's attention at 10 s, 10 min, 10 hr, and 100 hr, recognizing that the player can always put down the game and play another one. One must think creatively about giving the user a great experience at these critical junctures. Some games excel at certain points but not others. For example, the massively multiplayer game may have the user's rapt attention at 10 s and 10 min. And the fact that hundreds of thousands pay $10 a month to continue playing indicates that these games are rewarding at the 100 hr mark. But anyone who has played one of these games can tell you that they are extremely difficult and not too fun to play at the 10 hr mark. At this point, you are still getting 'killed' repeatedly. This is no fun at all. Arcade games obviously take this approach very seriously. Although they may not scale to 100 hr, good arcade games attract you to drop a quarter and keep you playing for long enough to make you want to spend another quarter to continue.

Over the course of a game, many games take a similar approach to that taken by old-time movie serials. Realizing that users will not complete the game in one sitting, the game becomes a series of punctuated experiences. Each level or section of the game might be expected to build from a casual level of challenge that gradually teaches the user a set of skills that they must master before facing a penultimate challenge. Game magazines often refer to this penultimate challenge or opponent as the 'Boss' of a level. This usually manifests itself in a larger or more powerful opponent that must be vanquished before moving on to the next level or set of challenges.

Pacing may also be expressed as a set of interwoven objectives much like the subplots of a movie. Again, *Banjo Kazooie* provides an excellent example of good pacing. Each level in *Banjo Kazooie* contains the tools necessary to complete the major objectives. Finding the tools is an important part of the game. New abilities, objectives, skills, and insights are gradually introduced as the player matures. While progressing toward the ultimate goal (of vanquishing the evil witch and saving the protagonist's sister), the player learns to collect environmental objects that enable them to fly, shoot, become invincible, change shape, gain stamina, add extra lives, and unlock new levels. This interlocking set of objectives keeps the game interesting and rewarding. Even if one is unable to achieve a particular goal, there are always sets of subgoals to work on, some of which may provide cues about how to achieve the major goal.

Summary

Attitude methodologies are better apt to measure factors such as overall fun, graphics and sound, challenge, and pace. The typical iterative usability test is an exploratory exercise designed to uncover problem areas in which the designer's intentions don't match the user's expectations; as a result, we typically choose to not use usability test to assess 'fun' or challenge. When attempting to assess attitudinal issues as 'overall fun' and 'challenge' we make use of a survey technique that affords testing larger samples. Internally, we have adopted the term Playtest or sometimes Consumer Playtest for this technique. At the same time, we use typical iterative usability methods to determine design elements which contribute to or detract from the experience of fun.

USER RESEARCH IN GAMES

Introduction to Methods at Microsoft Game Studios—Principles in Practice

Thus far we have introduced a number of variables that we think are important for game development and evaluation Identifying these issues alone will not result in a better gaming experience however To improve the gaming experience, one must be able to accurately measure and improve on many of the aforementioned issues by involving users In this section, we propose various methodologies and techniques that attempt to do just that Examples are taken from techniques used by the Microsoft Game Studios User-Testing Group

Our testing methods can be organized by the type of data being measured At the most basic level we categorize our data into two types: *behavioral* and *attitudinal* Behavioral refers to observable data based on performance or particular actions performed by a participant that one can measure This is similar to typical measures taken in usability tests (e g , time it takes to complete a task, number of attempts it takes to successfully complete a task task completion, etc) Attitudinal refers to data that represent participant opinions or views, such as subjective ratings from questionnaires or surveys These are often used to quantify user experiences Selection of a particular method will depend on what variables are being measured and what questions need to be answered

Another distinction that is typically made is between *formative* and *summative* evaluations which we apply to our testing methods as well Formative refers to testing done on our own products in development Summative evaluations are benchmark evaluations, either done on our own products or on competitor products It can be a useful tool for defining metrics or measurable attributes in planning usability tests (Nielsen 1993) or to evaluate strengths and weaknesses in competitor products for later comparison (Dumas & Redish 1999)

Facilities

Usability Laboratories. The usability labs at Microsoft are fairly similar to other industry and university usability labs such as those found at the American Institutes for Research Lotus Development Corporation the University of Washington (Dumas & Redish 1999), Motorola and Oracle Corporation (R M Pagulayan personal communication June 26 2001) Our basic setup includes a participant side and an observer side divided by a one-way mirror The observer side contains video-recording equipment a PC and a large monitor for observers The participant side contains two cameras a PC and a television for console use

Playtest Laboratories The playtest labs at Microsoft were built to allow us to run up to 17 participants at once while minimizing the participants ability to interact with one another Our basic setup for each station includes a questionnaire PC a game PC a 17″ monitor and a switchbox to toggle between the two machines as necessary Two machines are necessary because we often test with early versions of PC games before they are considered stable and do not want to lose data as a result of a crash Each station also includes a small 13″ stereo television that we use for testing with video game consoles All participants are given stereo headsets so as to not disturb others around them Partitions are also set up at each station so they cannot see what is going on at surrounding stations Data are gathered via online questionnaire data capture

Usability Techniques. At a certain level games can possess similar behavioral goals to productivity applications For example gamers should be able to navigate through main menu options to change their gamepad settings as easily as a user can navigate a menu structure to change their document settings in their word processor Gamers should also be able to interact easily with an input device whether it is a gamepad or a traditional keyboard and mouse From this perspective games are the same as other software applications in that they must be easy to use and interact with

Traditional usability techniques can be used to address a portion of the variables identified as important for game design Although usability techniques have been around in industry for some time, they are currently evolving to meet many of today s needs (Dumas & Redish 1999). In addition to measuring performance we use many standard usability techniques to answer "how" and "why" process-oriented questions For example, how do users perform an attack or why are controls so difficult to learn? The following are different user-centered techniques we use to address these issues: (a) structured usability tests (b) rapid iterative testing and evaluation (RITE; Medlock, Wixon Terrano Romero & Fulton 2002) and (c) other variations and techniques including open-ended usability tasks paper prototypes and empirical guideline documents In any given usability test it is common to combine two or more of any of the aforementioned techniques

Although these methods are useful they do not allow us to address issues with extended gameplay We cannot address any issues that may arise after playing the game for a couple of days This is problematic because one of the key challenges in game design is longevity With the competition, the shelf life of a game becomes limited Usability techniques can only provide information approximately within the first hour of playing the game

For clarity of presentation each technique will be discussed separately followed by a case study Each case study will only contain information pertinent to a specific technique thus examples may be taken from a larger usability test

Structured Usability Test. A structured usability test maintains all the characteristics that Dumas and Redish (1999) proposed as common to all usability tests: (a) goal is to improve usability of the product (b) participants represent real users; (c) participants do real tasks (however, these tasks may only represent a subset of the capabilities in the game); (d) participant behavior and verbal comments are observed and recorded; and (e) data are analyzed problems are diagnosed and changes are

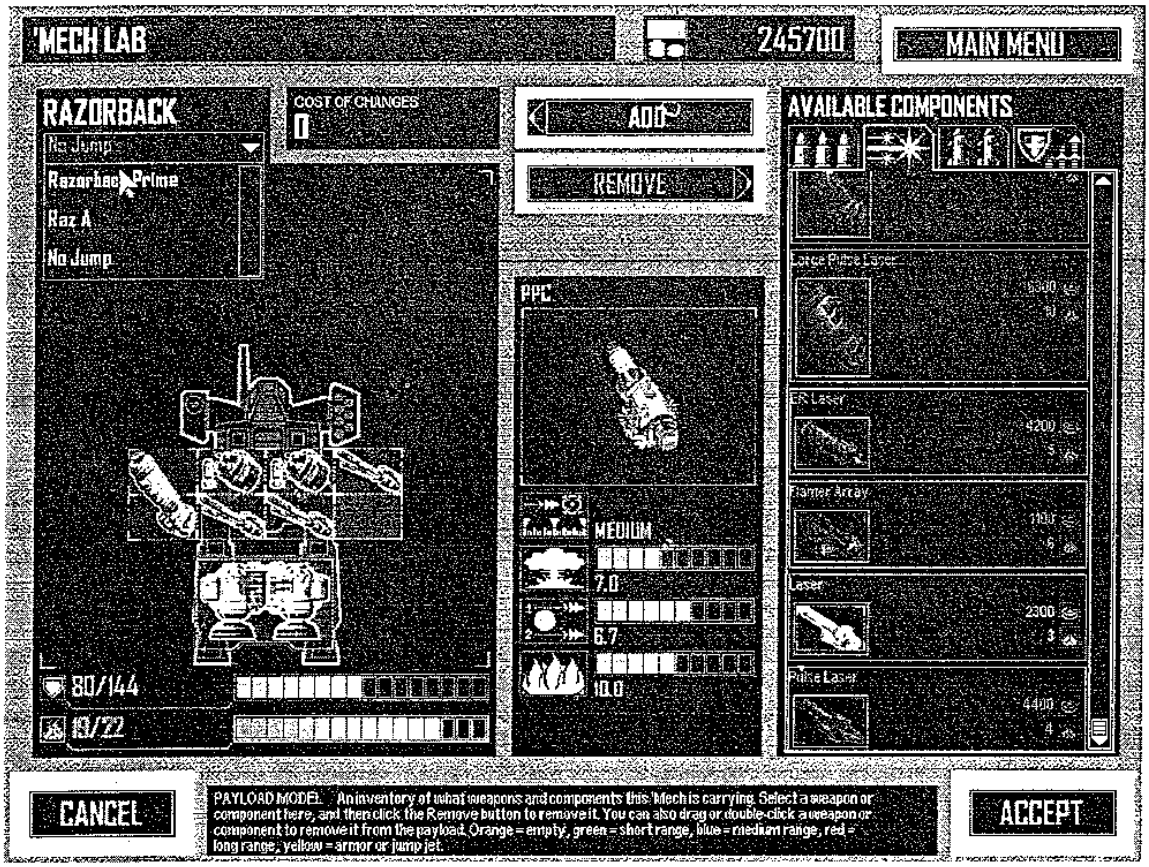

FIGURE 46 3 Screenshot of the Mech Lab in MechCommander 2

recommended We have found that issues relating to expectancies efficiency, and performance interaction are well suited for this type of testing Some common areas of focus for structured usability testing are in game shell screens or control schemes The game shell can be defined as the interface where a gamer can determine and or modify particular elements of the game This may include main menus and options screens (i e audio graphics, controllers, etc)

An example which uses this method is in the MechCommander 2 (2001) usability test Portions of the method and content have been omitted

Case Study: MechCommander 2 Usability Test MechCommander 2 (MC2) is a PC RTS game in which the gamer takes control of a unit of mechs (i e large mechanical robots) One area of focus for this test was on the Mech Lab a game shell screen where mechs can be customized (see Fig 46 3) Gamers are able to modify weaponry armor, and other similar features and are limited by constraints such as heat money and available slots.

The first step in approaching this test was to define the higher order goals Overall, the game shell screens had to be easy to navigate understand and manipulate not only for those familiar with mechs and the mech universe but also for RTS gamers who are not familiar with the mech universe Our goal was for gamers to be able to modify or customize mechs in the Mech Lab

As mentioned one of the most important steps in this procedure is defining the participant profile(s) Getting the appropriate users for testing is vital to the success and validation of the data since games are subject to much scrutiny and criticism from its gamers To reiterate playing games is a choice For MC2, we defined two participant profiles that represented all of the variables we wanted to cover The characteristics of interest included those who were familiar with RTS games (experienced gamers) and those who were not RTS gamers (novice gamers) We also wanted gamers that were familiar with the mech genre or the mech universe Overall we needed a landscape of gamers that had some connection or interest that would make them a potential consumer for this title whether through RTS experience or mech knowledge

Tasks and task scenarios were created to simulate situations that a gamer may encounter when playing the game. Most important tasks were created to address the predefined higher order goals Participants were instructed to talk aloud and performance metrics were recorded (i e task completion time) The following are examples from the usability task list

1 Give the SHOOTIST jumping capabilities This task allowed us to analyze participant expectations To succeed in this task one had to select a CHANGE WEAPONS button from a different game shell screen which brought them into the Mech Lab If the task was to change a weapon on a mech the terminology would probably have been fine Thus this task had uncovered two main issues: (a) could users get to the Mech Lab where you modify the mech and (b) were they able to discover

the process of modifying the mech It was accurately predicted that gamers would have difficulties with the button terminology Thus that button was changed to "MODIFY MECH

To change the components (i e., add the jump jets) participants could either select the item and drag it off the mech or select the item, and press the REMOVE button (see Fig 46 3) One unexpected issue that arose was that participants unknowingly removed items because the distance required for removing an item was too small The critical boundary that was implemented was too strict In addition participants had difficulties adding items by dragging and dropping because the distance required for adding an item was too large (i e the item would not stay on the mech unless it was placed exactly on top of the appropriate location). Appropriate recommendations were made and implemented

2 Replace the MG Array with the Flamer One of the constraints presented for modifying a mech was heat limit Each weapon had a particular heat rating For example, if the heat limit for a mech is 35 and the current heat rating is 32 only weapons with a rating of 3 or fewer could be added In this task, the "Flamer" had a heat rating much larger than the "MG Array" thus making impossible to accomplish this task without removing more items The issues here were the usability of the heat indicator heat icons and the discoverability of heat limit concept None of the participants figured this out Recommendations included changing the functionality of the Heat Limit Meter and to add better visual cues to weapons that exceed the heat limit Both of these changes were implemented

Rapid Iterative Testing and Evaluation Method (RITE) Medlock et al. (2001) have documented another common usability method used by the Microsoft Game User-Testing Group which they refer to as the RITE method In this method fewer participants are used before implementing changes but more cycles of iteration are performed With RITE it is possible to run almost 2 to 3 times the total sample size of a standard usability test However, only 1 to 3 participants are used per iteration with changes to the prototype immediately implemented before the next iteration (or group of one to three participants) This method has been used by us most commonly for game tutorials although it is not limited to just tutorials

The goal of the RITE method is to be able to address as many issues and fixes as possible in a short amount of time in hopes of improving the gamer s experience and satisfaction with the product However the utility of this method is entirely dependent on achieving a combination of factors (Medlock et al 2001) The situation must include (a) a working prototype (b) the identification of three types of behaviors (critical success behaviors, important but not vital behaviors and less important behaviors), (c) commitment from the development team to attend tests and immediately review results (d) time and commitment from development team to implement changes before next round, and (e) the ability to schedule or run new participants as soon as the product has been iterated Aside from these unique requirements, planning the usability test is similar to more traditional structured usability tests

It is helpful to categorize potential usability issues into four categories: (a) clear solution quick implementation; (b) clear solution slow implementation; (c) no clear solution; and (d) minor issues Each category has implications for how to address each issue In the first category, fixes should be implemented immediately and should be ready for the next iteration of testing In the second category fixes should be started, in the hope that it can be tested by later rounds of testing For the third and fourth category more data should be collected

The advantage of using the RITE method is that is allows for immediate evaluation and feedback of recommended fixes that were implemented Changes are agreed on and made directly to the product If done correctly, the RITE method affords more fixes in a shorter period of time In addition by running multiple iterations over time we are potentially able to watch the number of usability issues decrease It provides a nice easily understandable accessible measure In general the more iterations of testing the better This method is not without its disadvantages however In this situation we lose the ability to uncover unmet user needs or work practices, we are unable to develop a deeper understanding of gamer behaviors and we are unable to produce a thorough understanding of user behavior in the context of a given system (Medlock et al 2002)

The following example demonstrates how the RITE method was used in designing the Age of Empires II: The Age of Kings (1999) tutorial Again portions of the method and content have been omitted See Medlock et al (2002) for more details

Case Study: Age of Empires II: The Age of Kings Tutorial Age of Empires II: The Age of Kings (AoE2) is an RTS game for the PC in which the gamer takes control of a civilization spanning over a thousand years from the Dark Ages through the late medieval period In this case study, a working prototype of the tutorial was available, and critical concepts and behaviors were defined Also, the development team was committed to attending each of the sessions and they were committed to quickly implementing agreed on changes Finally, the resources for scheduling were available The key element in this situation for success was the commitment from the development team to work in conjunction with us

In the AoE2 tutorial there were four main sections: (a) marching and fighting (movement actions unit selection the "fog of war"[1]) (b) feeding the army (resources how to gather where to find); (c) training the troops (use of minimap advancing through ages build and repair buildings relationship between housing and population unit creation logic); and (d) research and technology (upgrading through technologies queuing units advancing through ages). Each of these sections dealt with particular skills necessary for playing the game In essence the tutorial had the full task list built in In the previous case study this was not the case

At a more abstract level the goals of the tutorial had to be collectively defined (with the development team). In more

[1] The *fog of war* refers to the black covering on a minimap or radar that has not been explored yet by the gamer The fog of war "lifts" once that area has been explored Use of the fog of war is most commonly seen in RTS games

TABLE 46 2 Age of Empires Example Concepts and Behaviors Categorized Into Three Concepts Using the RITE Method (Medlock, Wixon, Terrano, Romero, & Fulton, 2002)

Essential Concepts/Behaviors	Important Concepts/Behaviors	Concepts/Behaviors Of Lesser Interest
• Movement	• Queuing up units	• Using hotkeys
• Multiselection of units	• Setting gathering points	• Using minimap modes
• 'Fog of war"	• Garrisoning units	• Using trading
• Scrolling main screen via mouse	• Upgrading units through technology	• Understanding sound effects

Note RITE = rapid iterative testing and evaluation

concrete terms, specific behaviors and concepts that a gamer should be able to perform after using the tutorial were identified then categorized into the three levels of importance: (a) essential behaviors that users must be able to perform without exception (b) behaviors that are important but not vital to product success, and (c) behaviors that were of lesser interest Table 46 2 lists some examples of concepts and behaviors from each of the three categories This is an important step because it indirectly sets up a structure for decision rules to be used when deciding what issues should be addressed immediately, and what issues can wait The general procedure for each participant was similar to other usability tests we perform If participants did not go to the tutorial on their own they were instructed to do so by the user-testing engineer

During the session errors and failures were recorded In this situation, an error was defined as anything that caused confusion A failure was a considered an obstacle that prevented participants from being able to continue After each session a discussion ensued among the engineer and the development team to determine what issues (if any) warranted an immediate change at that time

To do this successfully, certain things had to be considered For example, how can one gauge how serious an issue is? In typical usability tests the proportion of participants experiencing the error is a way to estimate its severity Because changes are made rapidly here, the criteria must change to the intuitively estimated likelihood that users will continue to experience the error Another thing to consider is clarity of the issue, which was assessed by determining if there is a clear solution We have often found that if issues do not have an obvious solution then the problem not fully understood And finally, what errors or failures were essential, important, or of lesser interest Efforts of the development team should be focused on issues related to the essential category when possible

At this point we broke down the issues into three groups The first group included issues with a solution that could be quickly implemented Every issue in this group was indeed quickly implemented before the next participant was run The second group consisted of issues with a solution that could not be quickly implemented The development team began working on these in the hope that they could be implemented for later iterations of the test Finally, there were issues with no clear solutions These issues were left untouched because more data were needed to assess the problem at a deeper level (i e , more participants needed to be run) Any fixes implemented in the builds were kept as each participant was brought in Thus, it was possible that many of the participants experienced a different version of the tutorial over the duration of testing

Overall, seven different iterations were used across 16 participants Figure 46 4 represents the number of errors and failures recorded over time The number of errors and failures gradually decreased across participants as new iterations of the build were introduced By the seventh and final iteration of the build the errors and failures were reliably reduced to zero

Although we feel that the *AoE2* tutorial was an enormous success, largely because we used the RITE method the method does have its disadvantages and should be used with caution Making changes when issues or solutions are unclear may result in not solving the problem at all and in creating newer usability problems in the interface. We experienced this phenomena a couple of times in the AoE2 study

Also making too many changes at once may introduce too many sources of variability and create new problems for users Deducing specifically the source of the new problem becomes difficult A related issue is not following up changes with

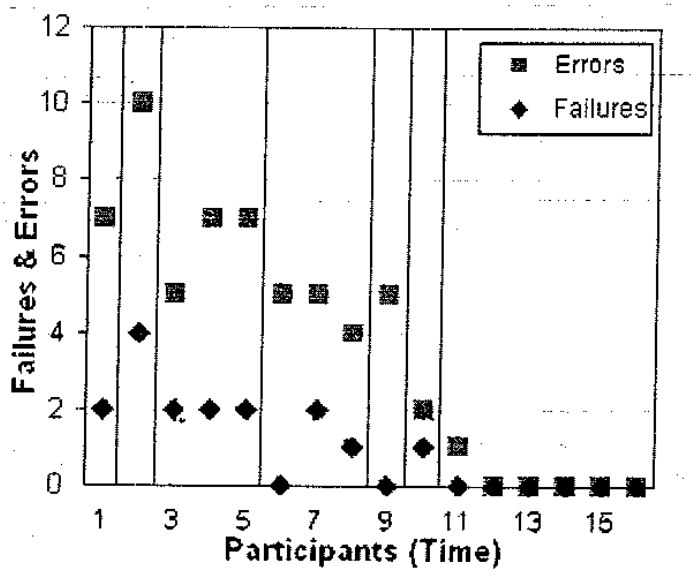

FIGURE 46 4. A record of failures and errors over time for the Age of Empires Tutorial when using the RITE method (Medlock, Wixon, Terrano, Romero, & Fulton, 2002) RITE = rapid iterative testing and evaluation

enough participants to assess whether or not the solution really addressed the problem. Without this follow-up there is little evidence supporting that the implementations made were appropriate (which is a problem with traditional usability methods as well). The last thing to consider is that other important usability issues that may surface less frequently are likely to be missed. Using such small samples between iterations allows for the possibility that those less occurring issues may not be detected.

Variations on Usability Methods Now that we have presented two general types of usability testing, it is worthy to mention some variations on these methods: (a) open-ended tasks, (b) paper prototyping, and (c) empirical guideline documents.

In general, it is often recommended that tasks in usability tests be small, with a specified outcome (e.g., Nielsen, 1993). However, we have found situations where the inclusion of an open-ended task yields important data as well. In many usability studies, we often include an open-ended task where participants are not instructed to perform or achieve anything in particular. In other words, there is no specified outcome to the participant. These tasks can be used to analyze how gamers prioritize certain tasks or goals in a nonlinear environment. These tasks are also useful in situations in which structured tasks may confound the participant experience, or situations where we are interested in elements of discovery. An example of an open-ended task is as follows:

1. *Play the game as if you were at home.* The moderator will tell you when to stop. This example was taken from a usability test on an Xbox sports game. In console sports titles, control schemes are often used repeatedly across different versions of the game. For example, Triple Play Baseball (2001) for PlayStation 2 uses the same control scheme (or very similar) for pitching and batting as many of the Triple Play predecessors (i.e., Triple Play '97 through 2001). Furthermore, similarities often exist across different titles from different developers. To select a pitch in Triple Play Baseball, a given pitch type is mapped to a given button. For example, to select a fast ball, one may have to press the button with the square icon on it. In All-Star Baseball 2002 (2001), the mechanic is the same. To select a fast ball, one may have to press a different button, but the idea is the same. Pitch selection (curveball, fastball, sinker, etc.) occurs via different buttons. The result is that gamers that play baseball console titles begin to expect certain functionalities to work across all baseball console titles.

One of the current sports titles we are working on has introduced a novel control technique that has not been done before. Thus, the reason we introduced the open-ended task was to find out if gamers could discover this new method, assess how they would attack the problem, and assess whether they liked it. At this point, introducing specific tasks would not allow us to make these types of initial discovery conclusions. This task allowed us to see how gamers would attack novel problems that did not match their current expectations. For example, novice gamers may behave differently than more experienced gamers, which has implications for how to introduce novel methods. Again, the attention span of gamers tends to be short. Thus, it is imperative that we ensure that novel functionality is immediately accessible.

Prototyping, heuristic evaluations, and empirical guideline documents are other techniques we use when more time-consuming testing cannot be done. In practice, these techniques do not differ when used on games. Nielsen (1993) categorized prototyping and heuristic evaluations as "discount usability engineering," and we would agree. We also tend to view empirical guideline documents in a similar manner.

Empirical guideline documents are essentially lists of usability principles for particular content areas based on our collective experience doing user-testing research. Examples of some of these content areas include console game shell design, PC game shell design, PC tutorial design principles, movement, aiming, and camera issues in first/third person shooter games, and online multiplayer interfaces.

Survey Techniques The use of surveys has been explored in great depth (e.g., Bradburn & Sudman, 1988; Couper, 2000; Labaw, 1981; Payne, 1979; Root & Draper, 1983; Sudman, Bradburn, & Schwarz, 1996) and is considered a valid approach for creating an attitudinal data set as long as you ask questions that users are truly capable of answering (see Root & Draper, 1983). One of the biggest complaints about a survey approach has to do with response rates to mail-based surveys (Armstrong & Lusk, 1988). We avoid this issue by bringing users on site. This also gives us the opportunity to question their attitudes while monitoring some simple behaviors. Even better, it allows us to take more control of the testing situations.

There are a number of advantages to using a technique such as this. We acquire so-called hard numbers that can be used for comparison from time one to time two (the formative approach) and to compare progress against other similar games (the summative approach). The numbers can also be used to give weight to issues. Other advantages include the factors of time and speed. Finally, the data can be presented in a purely descriptive format, meaning that trends of interest can come to light quite easily, even for lay people.

On the other hand, there are a few notable disadvantages. First and foremost, the engineer preparing the test must be intimately aware of what needs to be assessed to include questions in the survey that will answer the development team's questions. In other words, the questionnaire needs to allow for any potential useful user attitudes about the experience. Although there are a number of things we have done to address this (e.g., standardization of general questions to assure coverage of key gameplay perceptions, a question database to refer to), the ability to do this well is mostly a factor of the experience of the engineer and their familiarity with that game as well as the game genre. A high level of game domain knowledge is required. Second, the data is often difficult to interpret (70% believe an aspect of the game is "good" whereas 30% believe it is "bad") without some context. To limit this problem, we have built up a comparison database that we can use to answer exactly this kind of question. A third item to consider is that despite the existence of other data sets and reasonable foresight in questionnaire creation, there are still many unpredictable issues that may

arise Follow-up studies are sometimes necessary to parse out these issues Finally the approachable nature of the data is ripe for misinterpretation by the lay person Careful and consistent presentation formats are called for It has been rightfully asserted that usability tests do not necessarily require a formal report upon completion (Nielsen 1993) We believe that that may be less true for using this kind of technique

Although the use of a large sample can be used for a number of different things there are a few types of playtest studies that we conduct more consistently than others There are essentially two types of formative studies: the critical facet test and the initial experience test We also frequently run what we call benchmark (or summative) tests to build our comparison database, and, finally tests that do not easily fit into any of these categories including subtle variations on the above a few cases of large sample observationally based studies and studies that focus on games from a more conceptual level In nearly all instances, questioning follows the pattern of a forced choice question (sometimes a set of forced-choice questions) followed by a more open-ended question encouraging the participants to state the reasons behind their response

As in the usability section, for clarity of presentation each technique is discussed separately followed by a case study Each case study contains only information pertinent to a specific technique thus examples may be taken from a larger playtest

Critical Facet Playtest Games are often a collection of experiences that revolve around particular events In a baseball game although there are numerous aspects to the game such as playing the role of manager, controlling fielders and controlling runners on the base path, the game revolves around the critical game facet of pitcher versus batter Similarly, in an air combat game there may be several interesting choices to make with regard to which plane to choose for which mission perhaps even which weapons to outfit the plane with but the core gameplay the critical facets of the game are (a) being able to control the plane and (b) being able to control a plane well enough to win an air combat battle In both of these instances and across genres we have typically conducted tests that focus on critical facets such as these Although these kinds of game facets can often be assessed in a usability test as well as a playtest, the key is that attitudes about the experience are just as important as the usability (i e , learnability performance)

The following example demonstrates how a critical facet test was run in order to determine the optimal point of view behavior (often referred to as "the camera" in the gaming community) for Oddworld: Munch's Oddysee (2001), an Xbox game Munch s Oddysee is a platform/adventure game that allows you to switch back and forth between the two main characters as they proceed through the increasingly difficult dangers of Oddworld on a quest to save Munch's species from extinction Its core gameplay includes basic maneuvers including making the characters run jump and swim or making use of their special powers to proceed through the realm The camera behavior interacts with all components of the game

Case Study: Munch's Oddysee, Camera Previous usability testing with the *Munch's Oddysee* had determined that whereas some users indicated dissatisfaction with the behavior of the camera other participants chose not to mention it at all while engaged in the open-ended usability tasks The camera s behavior was programmed so as to create maximal cinematic effects (i e , sometimes zooming out to show the size of an area) and also attempt to enhance gameplay The camera would often show a specific view with the intent of showing the user what was behind "the next door" or on the other side of the wall while still keeping the main character in view The most common user complaint was that although they often liked the look, style and behavior of the camera they wanted more control over the behavior of the camera as other games in this genre have done Indeed some participants would actively say things such as "That looks really cool right now [after the camera had done something visually interesting] but I want the camera back pointing this way now" The nature of the data from the usability lab was not fully satisfying to the development team There was feedback related to attitudes of participants that indicated both positive and negative perceptions of the camera This was weighed against the fact that changing the behavior of the camera represented a major cost to the game in terms of development time and perhaps in the redesign of certain already completed areas of the game

A critical facet playtest was conducted in order to shed more light on the attitudes related to the camera behavior in Munch s Oddysee We targeted having at least 25 participants whose previous behavior (i e , their gaming experience) indicated that they fell within a general definition of the target market for this game

After having played the game for an hour participants were asked first for general perceptions of the game followed by more specific questions related to other areas of interest Questions related to the camera were asked in the latter portion of the questionnaire because previous experience in the usability lab had shown that merely mentioning the camera as part of a task would often cause participants previously silent on the subject to vociferously criticize aspects of the camera's behavior With the knowledge that we wanted to factor out any priming-related effects two analyses were conducted

The first analysis was based on the response to the questions related to the behavior of the camera itself Nearly half of the participants (46%) indicated that the camera did not give them "enough flexibility" of control The second analysis was to go back through individual responses and determine if before we brought up the subject of the camera any participants (a) mentioned the camera and (b) if possible the comments related to the camera were subcategorized as a negative or positive statement Forty-three percent of the participants were found to have mentioned the camera in a negative fashion before the camera questions Based on these data and other anecdotal evidence, the development team chose to modify the behavior of the camera in this game to give the players more flexibility of control The result was more frequent use of a camera behavior that we have termed a third-person follow camera

The behavior of this camera had the double advantage of being more easily controlled by users and of conforming to a set of behaviors more often expected by users such as maintaining

focus on the main character without major adjustments to point of view (i e to 'look over a wall' or 'behind a door') Other camera behaviors (i e still camera behaviors that pan with the character) are still a part of the game but have been localized to areas where these alternative camera behaviors can only create an advantage for the user

Initial Experience Playtest (Formative) As with many things first impressions are a key component of overall satisfaction with a game Given that many games are experienced in a certain order (i e Mission 1, Mission 2 etc.) there is a lot of value in obtaining attitudinal data related to the first hour(s) of gameplay because the later portions of the game will clearly never be experienced unless the first portions of the game are enjoyed

As mentioned already games are entirely a voluntary experience and at any moment a person may become frustrated or bored and put the game down In a perfect world we would like to be able to identify these moments as they occur to be able to address them The realities of the manner in which games are created (often times individual portions of a game are completed at different times and in a different order than consumers will actually experience them) and the realities of the time constraints related to testing with users prevent this from happening If we want to know how users react in the 30th hr of gameplay then we need to conduct a study where the user would experience the first 29 hr of the game Those 29 hr of gameplay need to be ready and completed before running this 30-hr test Clearly this would be difficult expensive and time-consuming As a result we often focus on the first hour of gameplay because it is easily accessible to us from a research perspective and the lessons learned can in many cases be applied throughout the game

The following example explains how a set of formative initial experience tests were run for MechCommander 2 the RTS game described earlier in the chapter The earlier usability test focused on the choices that users could make before starting a mission whereas this test focused on in-game components such as user's ability to take control of their squad and lead them through battles, while experiencing satisfaction and motivation to continue playing

Case study: MechCommander 2, Initial Missions A sample of 25 participants who through previous behavior were identified as representative of the potential target market were included in the study The participants were brought on site and asked to begin playing from the first mission of the game After each mission was completed the participants were asked to stop playing and fill out a set of questions focusing on their impressions of the 'fun' the 'excitement' and the 'clarity' of objectives as well as respond to measures designed to assess their 'comfort level' with learning the basics of controlling the game The participants were able to play through as many as three missions before the session ended For purposes of brevity this case study will focus on the results related to the first mission

At the time of this test the first mission was a quick experience designed to introduce users to some basic control concepts to allow them to participate in some combat win that combat and then move on to the next mission Also notable was that at the time, the first few missions of the game were intended to act as the tutorial for the game as well as being fun missions As it happens this experience was generally not satisfying to these participants, causing a poor initial impression with the game Approximately one third of the participants felt the 'challenge' was 'too easy' and that the mission was 'not exciting.' Furthermore there were some problems related to clarity of goals where some participants were observed to move their units into an area where they were not intended to go, disrupting their experience and limiting their ability to proceed quickly Responses to more open-ended questions indicated that some of the core gameplay components and unique features did not come across to users because they complained about the 'standard' nature of the game Finally several participants complained about the fact that they were 'being taught' everything and wanted to 'turn the tutorial off'

From the data the development team, working with feedback from the user-testing engineer decided to take a number of actions First it was decided that a separate set of tutorial missions would be created to give those who wanted to be taught a place to go and learn the basics of controlling the game separate from the actual missions of the game Second the scope of the first mission was expanded so that users would have more time in their initial experience with the game Third the clarity of objectives was improved via minor interface changes Fourth addressing the same issue the design of the 'map' on which the mission took place was revamped such that users were required to take a more linear approach to this mission and be unable to 'get lost' Fifth the amount and challenge of combat was increased Finally one of the unique components of the game, the ability to 'call in support' from off-map facilities was introduced to the player demonstrating how they could choose to augment their force on a case by case basis Despite all these changes the mission was still targeted to be completed within approximately 10–15 min

A follow-up playtest was run to verify that the changes we introduced were paying off in terms of generating more user satisfaction with the initial mission of the game Again 25 participants (not the same people) representative of the target market were included in the study Results from this test indicated that the design changes had created a number of payoffs Far fewer participants felt the mission was 'too easy' (13% vs 33%) only 3% indicated that the mission was 'not exciting' measures of 'clarity of objectives' improved and, surprisingly, there was no dropoff on ratings of 'comfort' with basic controls as a result of the tutorial aspects being improved Results were not a total success, because some participants were now rating the mission as 'too hard' whereas others in response to open-ended questions complained about the 'overly linear' nature of the mission (i e there was only one way to proceed and few interesting decisions related to how to proceed through the mission) In response to these results the development team decided to 'retune' the first mission to make it a little easier but not to address the comments related to the linearity of the mission The second mission of the game was far less linear so it was hoped that, with the major constraints to enjoyment removed most participants would proceed to the second mission and find the

mission goals to be more engaging The data from testing with Mission 2 was rated far more "fun" than Mission 1 validating their assumption

Benchmark Playtests As indicated earlier the nature of the data that is supplied using this survey technique is not always clear and easy to interpret. If 10% indicate that learning how to drive a car is "too hard" then is this something we should act on? The question is not easily answered without context To supply context benchmark playtests are conducted on every completed Microsoft game, as well as on many competitors The data are used entirely for comparison purposes. Every benchmark playtest is run using the same protocol All participants are directed to play as if they are at home, and we stop them to ask a set of questions to gather initial impressions and then let them play for the rest of the session followed by a set of final questions Both of those question sets are general and concentrate on many of the core gameplay facets (i e , "fun" "challenge" "pace" etc) highlighted in previous sections.

The data gathered in these instances are used as comparison for initial experience playtests Typically, the general questions are followed by a set of more specific questions designed to highlight user impressions about critical facets of a game The data gathered in response to these questions are often used in comparison to critical facet playtests Although the choice of comparing data gathered from a benchmark playtest with that from a critical facet playtest (where the experience was likely very different) is certainly controversial, the only goal of the data is to shed light and provide some context, not always to force action based on these comparisons The decision on how strongly to weight the comparison is entirely a judgment call on the part of the user-testing engineer based on his or her knowledge of how closely the experience in a critical facet playtest models the intended initial experience with the game As in all good usability testing, the choice of areas to focus on in these critical facet questions is done in close consultation with the development teams.

Qualitative Group Methods. In addition to usability and survey techniques, we employ other UCD methods we categorize as qualitative group methods; deep gameplay and focus groups

Deep Gameplay. Thus far the techniques we presented have focused primarily on evaluating the initial experience with a product or one segmented hour of game play and are often experienced out of context Because of the nature of this field testing portions of game play out of sequence occurs quite often but can be somewhat problematic Although the initial experience with a game is important the success or failure of a game often depends on the entire experience In an attempt to provide game developers with user-centered feedback beyond the first hour of game play we established the deep gameplay program

Deep gameplay involves bringing in cohorts of users in to play a product repeatedly throughout the development cycle and collecting qualitative data from them The goal of deep gameplay is to provide fairly structured user-centered feedback on gameplay that occurs after the first hour of experience by supplying each team with its own dedicated gameplay team Deep gameplay teams can be used to (a) evaluate content beyond the first hour in a linear or nonlinear fashion (b) expose the same group of users to iterations of the same content, (c) test linear based modes of extended play (e g , a career mode in a snowboarding game or multiple levels of an adventure game) or (d) expose users to a prerelease video game so that they have the experience necessary to evaluate advanced features or features that appear later in the game

Deep gameplay requires commitment from participants to attend gameplay sessions on a consistent basis (potentially one to three times a month) otherwise the utility of deep gameplay is lost Participants are exposed to features and game content in various ways Generally the order in which they encounter content depends greatly on the development schedule, the type or genre of the game or the team s specific needs and goals Commonly, the participants test each level as it becomes available or is significantly improved The group can also return to previously tested sections that have been iterated to see if it has improved from before Ideally participants would be exposed to the game in a linear fashion progressing from the beginning to the end However this is only possible to do later in the development cycle because games are usually not developed from start to finish Generally when this occurs the team either takes the group back to the beginning of the game or starts a second group from the beginning

Although there is no single report format for compiling data from these sessions the report generally includes (a) a list of top issues that were brought up by the participants along with recommendations; (b) a summarization of the responses to the written questions; (c) and a list of remaining issues from previous sessions

The purpose of deep gameplay overlaps with other testing methods (viz usability and playtest) but it has been designed wholly as a supplement not a replacement Each of these testing methods has its own unique set of strengths and weaknesses

There are other methods that exist that attempt to get feedback from potential consumers These methods are often referred to in the industry as "beta" and "recon" Beta testing usually involves sending a beta or prerelease version of a game to a large number of potential users Feedback is collected on issues including configuration problems, technical bugs and gameplay problems Data collection is often done via newsgroups or electronic bug reports Beta testing has been successful in getting some useful feedback from a larger number of users but it does not ensure the amount focus or depth of the feedback that our methods provide Recon s purpose is similar to deep gameplay The difference is that recon uses professional gamers which does not truly represent consumer feedback

Focus Groups Focus groups provide an additional source of user-centered feedback for members of the project team including usability practitioners Focus groups give development

team members a rare opportunity to hear users speak candidly about the game in question. In our group, the interviews are especially useful because the focus group participants often come from a playtest session which gives them the opportunity to use the product before discussing it. Additionally, the focus group interviews offer a qualitative method of data collection that is used to support and supplement data produced using quantitative survey methods.

In this setting, participants can elaborate on their experience with the product and expectations for the finished version of that product. The interviews allow participants to talk about their experiences and share impressions that may not have been elicited by the playtest survey. Allowing users to talk about their experiences can sometimes result in more detailed responses to help validate the quantitative data. This provides the team with a method of delving deeper into gameplay issues and strengths.

By using the qualitative and quantitative methods together, we are sometimes able to use information from focus groups to improve on survey techniques. The focus group data may identify important topics that should be added to the survey, as well as clarify topics that were poorly understood by users. For example, a focus group discussion regarding a sports game revealed that users had varying opinions about what it meant for a game to be a simulation vs. an arcade-style game. Additionally, they had many different ideas as to what game features made up each style of game. From this conversation, we learned that certain playtest survey questions regarding this topic were not reliable and needed revision or exclusion from future surveys.

Besides the general limitations inherent in all focus group studies (Greenbaum, 1988; Krueger, 1994), the focus groups described here include limitations associated with their deviation from validated focus group methods. Generally, focus group studies comprise of a series of 4 to 12 focus groups (Greenbaum, 1988; Krueger, 1994); however, we are only able to run one to three focus groups per topic. This makes it more difficult to confidently identify "real" trends and topics in the target population. Additionally, fewer group interviews make the study more vulnerable to idiosyncrasies of individual groups.

Because all focus group participants have recently completed a survey in playtest, it is also possible that the survey topics confound the issues raised in the general discussion in the focus group. The focus groups are commonly used in conjunction with the survey data, so this is not a serious limitation if used properly.

Because of these limitations, we take many precautions to ensure the quality of the data produced. The data that are produced in focus groups are closely monitored and scrutinized appropriately. Despite these limitations, we have found that focus groups have offered a useful and face-valid method of supplementing user-centered feedback that we provide to game developers.

A Combined Approach. We have provided a number of different techniques that we use in game development and evaluation. The presentation has been structured in such a way to clearly demonstrate the differences between the techniques we use. The important thing to realize is that no one method exists independent of other methods. All of our techniques are used in conjunction with one another to fully address as many of the game evaluation factors presented earlier in the chapter. The following is a brief case study on an Xbox game called Blood Wake (2001) that demonstrates how some of these techniques can be used together.

Case Study: Blood Wake. Blood Wake is a vehicular combat console game using boats and a combination of powerful weapons. In a game such as this, there are certain game facets that must be done correctly for it to be successful. In Blood Wake, our initial main concern was the aiming mechanism and the controls. Users must also be able to destroy the enemy, and users must be able to easily control their boat. If these two factors were not addressed, the likelihood of a successful game significantly decreases. The next concern was related to the initial missions and gameplay and the general look and feel of the game overall. Users must be able to know what their objectives are, and it has to be fun.

The first problem we addressed was the aiming mechanism. We were presented with two different types of aiming schemes: auto aim and manual stick aim. In the auto aim scheme, the target reticle automatically 'snapped' to an enemy that was within a reasonable distance allowing the player to maneuver their boat while shooting the enemy. In the manual stick aim scheme, the user had to manually aim the reticle using the right stick on the game pad with no assistance. In previous usability studies, we have found that many users encounter difficulties with a manual aiming mode. However, this method is used often in popular games, thus the general belief was that it was better or preferred. In addition, many felt the auto aim method would be too easy.

To solve this problem, we decided that a playtest would be most suitable. A usability test would probably allow us to demonstrate errors based on tasks and performance but would not really allow us to make any conclusions about which scheme users preferred. Thus, we ran a within-subjects design playtest to assess which scheme users liked better. Our results showed that 73% of participants liked the auto aim, whereas only 38% liked the manual aim. This was proof enough to convince the developers to focus their efforts on the auto aim method.

Although we solved the aiming problem, we still were not in any position to get feedback on the fun or challenge of the game. The reason for this was that the controls (i.e., handling) of the boat was not really optimized yet. Any conclusions about fun or challenge of the game at this point in development would be a dubious enterprise. The reason for this was because we would not have the ability to separate poor attitudinal ratings due to the gameplay from the poor attitudinal ratings deriving from the poorly tweaked 'feel' of the boat or uncomfortable button mappings. To optimize these controls, we headed to the usability labs.

In the usability test, we presented participants with two control schemes or button mappings. In one scheme, throttle

and steering were both mapped to the left analog stick (push forward/back for throttle, push left/right for steering). This means that a user would be able to drive and steer with one finger. In another scheme the throttle was mapped to the right trigger (which exists under the game pad), and steering was on the left analog stick. Previous data suggest that neither scheme was superior so the goal of this test was to optimize the controls for both schemes

Participants were presented with a series of tasks that were created to try and simulate common maneuvers performed during the game. The following are examples taken from the usability task list

1 Once the game starts you should see a bunch of boxes that form a path in the distance. As fast as you can try to follow the trail of items and collect as many boxes as you can without turning around if you get off course
2 Somewhere on this map there is a straight line of boxes. Find the straight line of boxes then try to slalom (like a skier) through the line of the boxes. Try to do this as fast as you can without collecting any of the boxes

As you can see from the two examples the tasks were very performance-based. Participants were instructed to think aloud while we measured task performance and task time. By the end of the study, we were able to provide detailed information to the team about how to optimize the controls. Some of the issues that were addressed included the sensitivity of the steering controlling boat speed and difficulties moving the boat in reverse to just name a few

At this point changes were implemented to the controls and the aiming method. The next step would be to validate these changes but not from a performance-based perspective. We needed to determine if users liked how the controls felt, and we needed to begin getting feedback on missions and game play. Thus multiple playtests were run focusing on the game play of five of the initial missions and the controls. Our data on the controls from the playtest validated our earlier recommendations and allowed us to tweak the controls even more

In addition, by running multiple playtests, we were finally able to get data and feedback on gameplay. After each mission we elicited feedback on the perceived difficulty, clarity of objectives, attitudes about the environment, and general fun ratings. This allowed us to provide specific details to the development team on where exactly participants were running into problems (such as not knowing where to go next, not understanding how to complete objectives, etc.). This feedback led to dramatic changes design of the missions. For example, after the third mission, participants began to express boredom with the 'look' of the environments. They did not know that by Mission 5, more interesting things would be introduced (such as differing the time of day, the weather, and the color of the water). This led to the introduction of more varied environments within the first three missions which helped increase fun ratings.

CONCLUSION

The need for user-centered design methods in video games has indeed arrived. Games drive new technologies and affect an enormous amount of people. In addition, games represent a rich space for research areas involving technology, communication, attention, perceptual-motor skills, social behaviors, virtual environments, to name a few. It is our position that video games will eventually develop an intellectual and critical discipline like films, which would result in continually evolving theories and methodologies of game design. The result will be an increasing influence on interface design and evaluation. This relationship between theories of game design and traditional HCI evaluation methods has yet to be defined but definitely yields an exciting future.

As mentioned earlier, UCD methods have yet to find their way into the video game industry at least to the same extent as it has in other fields. However, this is not to say that video games are qualitatively different from other computer applications. There are just as many similarities as differences to other computer fields that have already benefit from current UCD methods. It makes sense to use these methods when applicable but also to adapt methods to the unique requirements that we have identified in video games.

In this chapter, we emphasized the difference between games and productivity applications to illustrate the similarities and differences between these two types of software applications. We also chose to reference many different video games in hopes that these examples would resonate with a number of different readers. Case studies were included to try and demonstrate in practice how we tackle some of the issues and challenges mentioned earlier in the chapter. It is our intention that practitioners in industry, as well as researchers in academia, should be able to take portions of this chapter and adapt them to their particular needs when appropriate, similar to what we have done in creating the actual methods mentioned in this chapter. That said, we acknowledge that most, if not all, of our user-testing methods are not completely novel. They have been structured and refined based on a combination of our applied industry experience, backgrounds in experimental research, and, of course, a passion for video games. This allows us to elicit and use the types of information needed for one simple goal: to make the best video games we can for as many people as possible.

ACKNOWLEDGMENTS

We would like to thank the Microsoft Game Studios User-testing Group. In addition, we would like to express our gratitude to Paolo Malabuyo and Michael Medlock for their insights and input on the creation of this chapter. Kathleen Farrell and Rally Pagulayan for their editing assistance and reviewing of early drafts, and Ed Fries and all of Microsoft Game Studios for their support. The views and opinions contained in this chapter are those of the authors and do not necessarily represent any official views of Microsoft Corporation.

References

Age of Empires II: Age of Kings [Computer software] (1999) Redmond WA: Microsoft.

All-Star Baseball 2002 [Computer software] (2001) Glen Cove NY: Acclaim Entertainment.

Armstrong, J. S. & Lusk E. J. (1988). Return postage in mail surveys *Public Opinion Quarterly, 51*, 233–248

Asakura R. (2000) *Revolutionaries at Sony*. New York: McGraw-Hill

Au W. J. (2001 April) Playing God *Salon* Retrieved from http://www.salon.com/tech/review/2001/04/10/black_and_white/index.html

Banjo Kazooie [Computer software] (2000) Redmond WA: Nintendo of America

Barfield W. & Williges R. C. (1998) Virtual environments: Models methodology and empirical studies [Special issue] *Human Factors 40*(3)

Blood Wake [Computer software]. (2001) Redmond WA: Microsoft

Bradburn, N. M. & Sudman S. (1988) *Polls and surveys. Understanding what they tell us*. San Francisco: Jossey-Bass

Cassell J. & Jenkins H. (2000) *From Barbie to Mortal Kombat Gender and computer games*. Cambridge MA: The MIT Press.

Chaika M. (1996). Computer games marketing bias *ACM Crossroads* Retrieved from: www.acm.org/crossroads/xrds3-2/girlgame.html

Couper M. P. (2000) Web surveys: A review of issues and approaches *Public Opinion Quarterly, 64* 464–494

Crawford C. (1982) *The art of computer game design* Berkeley CA: Osborne/McGraw-Hill

Csikszentmihalyi M. (1990) *Flow—The psychology of optimal experience* New York: Harper & Row

Diablo [Computer software] (2000) Paris. Vivendi Universal

Diddy Kong Racing [Computer software] (1997) Redmond WA: Nintendo of America

Dumas J. S. & Redish J. C. (1999) *A practical guide to usability testing* (Rev. ed.) Portland OR: Intellect Books

Final Fantasy IX [Computer software] (2000). Redmond WA: Electronic Arts

Flight Simulator [Computer software] (2000) Redmond WA: Microsoft

Funge J. (2000) Cognitive modeling for games and animation *Communications of the ACM 43*(7), 40–48

Gorriz C. M. & Medina C. (2000) Engaging girls with computers through software games *Communications of the ACM 43*(1) 42–49

Greenbaum T. L. (1988) *The practical handbook and guide to focus group research* Lexington MA: D C Heath

Hecker C. (2000) Physics in computer games *Communications of the ACM, 43*(7) 35–39

Herz J. C. (1997) *Joystick nation. How videogames ate our quarters, won our hearts and rewired our minds* New York: Little Brown

House of the Dead II [Computer software] (1999) San Francisco: Sega.

Interactive Digital Software Association (2000) State of the industry report (2000–2001) Washington DC: Author.

Jablonsky S. & DeVries D. (1972) Operant conditioning principles extrapolated to the theory of management *Organizational Behavior and Human Performance, 7* 340–358

Kent S. L. (2000) *The first quarter: A 25-year history of video games* Bothell WA: BWD Press

Kroll, K. (2000) Games we play: The new and the old *Linux Journal 73es* Retrieved from www.acm.org/pubs/articles/journals/linux/2000-2000-73es/a26-kroll/a26-kroll.html

Krueger R. A. (1994) *Focus groups: A practical guide for applied research* Thousand Oaks CA: Sage

Labaw P. (1981) *Advanced questionnaire design* Cambridge MA: Abt Books

Laird J. E. (2001) *It knows what you're going to do: Adding anticipation to a Quakebot Proceedings of the Fifth International Conference on Autonomous Agent* (pp. 385–392) Canada: CSREA Press

Lepper M. Greene D. & Nisbett R. (1973) Undermining children's intrinsic interest with extrinsic rewards *Journal of Personality and Social Psychology, 28* 129–137

Lepper M. R. & Malone T. W. (1987) Intrinsic motivation and instructional effectiveness in computer-based education In R. E. Snow & M. J. Farr (Eds.) *Aptitude learning and instruction III Conative and affective process analyses* Hillsdale NJ: Lawrence Erlbaum Associates

Levine K. (2001 May) *New opportunities for Storytelling* Paper presented at the Electronic Entertainment Exposition Los Angeles CA

Lieberman D. A. (1998 May) *Health education video games for children and adolescents. Theory, design and research findings* Paper presented at the annual meeting of the International Communication Association Jerusalem

Malone T. W. (1981). Towards a theory of intrinsic motivation *Cognitive Science, 4* 333–369

MechCommander 2 [Computer software] (2001) Redmond WA: Microsoft

Medlock M. C., Wixon D. Terrano M. Romero R. L. & Fulton B. Using the RITE Method to improve products: a definition and a case study *Humanizing Design, Usability Professional's Association 2002 Annual Conference* Orlando FL USA July 8–12, 2002.

MTV Music Generator [Computer software] (1999) Warwickshire United Kingdom: Codemasters Software

NFL Blitz [Computer software] (1997) Corsicana TX: Midway Games

Nielsen, J. (1993) *Usability engineering* San Francisco: Morgan Kaufmann

Oddworld: Munch's Odyssey [Computer software]. (2001) Redmond WA: Microsoft.

Payne S. L. (1979) *The art of asking questions* Princeton NJ: Princeton University Press

Pokemon Crystal [Computer software] (2000) Tokyo: Nintendo Japan

Preece J. Rogers Y. Sharp H. Benyon D. Holland S. & Carey T. (1994) *Human-computer interaction* Reading MA: Addison-Wesley

Provenzo E. F. Jr. (1991). *Video kids. Making sense of Nintendo* Cambridge MA: Harvard University Press

Root R. W. & Draper, S. (1983 December) *Questionnaires as a software evaluation tool* Paper presented at the ACM Conference on Human Factors in Computing Systems Boston MA.

Smart L. J. (2000) *A comparative analysis of visually induced motion sickness*. Unpublished doctoral dissertation University of Cincinnati OH

Steers R. M. & Porter L. W. (1991). *Motivation and work behavior* (5th ed.) New York: McGraw-Hill Inc.

Stevenson S. (2001 April) Why are video games for adults

so juvenile? *Slate* Retrieved from http://slate msn com/Culture-Box/entries/01-04-19_104657 asp

Stoffregen T A Bardy B G Smart L J & Pagulayan, R J (in press) On the nature and evaluation of fidelity in virtual environments In L J Hettinger & M W Haas (Eds), *Psychological issues in the design and use of virtual environments* Mahwah NJ: Lawrence Erlbaum Associates

Sudman S Bradburn, N M & Schwarz N (1996) *Thinking about answers The application of cognitive processes to survey methodology* San Francisco: Jossey-Bass

Triple Play Baseball [Computer software] (2001) Redwood City CA: Electronic Arts Inc

Williams J H (1987) *Psychology of women Behavior in a biosocial context* (3rd ed) New York: W W Norton