

now entertaining users perceive the
 e more interested in a topic would
 ted information. The results did not
 appeal was not related to the duration
 with the level of interactivity by the
 estigation.

ilizing both small group design
 to be highly beneficial. In the small
 a large number of people about a
 e data and arrive at a design for
 me and labor intensive for the team
 ch and detailed design information
 hods are appropriate for different
 complementary information.

experiences for users, and there is
 inment on the web. Future research
 ing this topic through an in-depth
 of varying duration, content, with
 t, and physical settings.

ES

ing: Tight coupling of dynamic query
 ton MA, April 1994), ACM Press, 313-

California.: UCLA Center for

w the Geeks, the Suits, and the Ponytails
 g, New York, New York.

on-Wesley.

rrative in Cyberspace. New York, NY.

igital Hollywood. Available:

l.

). Seductive interfaces: Satisfying a mass
 pril 1994), ACM Press, 359-360.

terface tool. . *Proceedings of CHI '94*

, 2000]

., Riecken, D., and Podlaseck, M. (2001).

tered design of a multi-institutional web

Museum and the Web 2001: Selected

sum Informatics, 23-32.

CHI '99, ACM Press, 306-312.

ion, Cambridge University Press,

lia presentations. *The Data Base for*

RANDY J. PAGULAYAN, KEITH R. STEURY,
 BILL FULTON, AND RAMON L. ROMERO

CHAPTER 11

DESIGNING FOR FUN: USER-TESTING CASE STUDIES

1. INTRODUCTION

The goal of this chapter is to demonstrate that extending current usability methods and applying good research design based on psychological methods can result in improved entertainment experiences. This chapter will present several case studies where user-centered design methods were implemented on PC and Xbox games at Microsoft Game Studios. The examples were taken from several series of larger studies on Combat Flight Simulator (PC), MechWarrior 4: Vengeance (PC), Halo: Combat Evolved (Xbox), and RalliSport Challenge (Xbox). These examples were chosen to illustrate a variety of user-centered methods and to demonstrate the impact user-centered design principles can have on an entertainment product. Furthermore, these examples are presented in a way that illustrates a progression from addressing usability issues similar to those found in productivity applications, to extending usability methods to address more unique aspects of game design, to using survey methods to address issues related to fun for which standard usability methods do not suffice. For more detailed descriptions of Microsoft Game Studios User-testing methods and laboratory facilities, see Pagulayan, Keeker, Wixon, Romero, and Fuller (2003).

1.1 Methods and Games

The many similarities between productivity applications and games suggest that traditional discount usability methods would be suitable in the entertainment domain. Games have selection screens and menus just like other software applications. Task persistence, performance, ease of use, learnability, and all the potential obstacles to efficiency and productivity are found in games as well. However, it is possible to conceptualize usability in games as including other areas of game design, such as the comprehension of rules and objectives, control of characters, and manipulation of camera (view), to name a few. A game designer must script an experience within a game, so an extension of usability techniques from productivity applications to games becomes clearer. Usability testing in games

(and productivity applications for that matter) can be viewed as an experiment designed to assess whether users will interact with a given product in the way the designer intended.

Video games also differ from productivity applications in a variety of ways. For example, productivity applications represent tools as a means to an end, whereas games are designed to be pleasurable for the duration of gameplay. This distinction means that we must ensure that the user has an appropriate level of challenge and engagement while playing a game, rather than focusing on how efficiently they can achieve their goals. Goals are often defined externally in productivity applications, whereas games define their own goals. This implies that goals and objectives in games must not only be clear to the user at all times, but that they must be interesting as well. Another difference between games and productivity applications lies in the number of choices available. While there are relatively few productivity applications designed for a specific purpose, there are many games. Games must compete with other game titles for consumer attention, but they must also compete with different forms of entertainment, such as watching television or reading a book (Pagulayan et al., 2003). The implication is that games that are not immediately enjoyable will be dismissed quite easily in favor of other forms of entertainment.

It is our intent to show that the HCI field has access to methods drawn from current usability methodologies and experimental psychology that can be adapted to address issues in the entertainment industry. Extending current usability methods and utilizing good research design can result in improved entertainment experiences.

2. CASE STUDIES

Below are four examples of user-centered design methods that were used on video and computer games at Microsoft Game Studios. The first case study begins with more traditional usability methods. Subsequent case studies extend the use of current methodologies to situations that demonstrate how one can address some of the unique issues encountered with games.

2.1 *Combat Flight Simulator*

This case study is an example of a usability issue identified using limited-interactive prototypes. This method is not substantially different from standard usability methods. The focus of the usability test in this case study was on the game shell screens for the PC game, *Combat Flight Simulator (CFS)*. Game shell screens consist of all menus and screens (e.g., Main Menu, Options, etc.) which are used to set the particular desired gaming experience, but not encountered during actual gameplay. For example, in the game shell a user can often change the difficulty level of the game and set other gameplay parameters.

A usability test was performed early in the development cycle of CFS and used a limited-interactive prototype of the game shell screens (i.e., functional screen widgets including radio buttons, check boxes, and toggles). Techniques utilized

included user and task observation (Nielsen, 1993). Some of the tasks were performance based. In general, the test was conducted while thinking aloud.

In CFS, parameters users can adjust include selection, start location, time of day, number of enemies. During the exploratory test, several issues were detected related to terminology. Some terms commonly used in the video and computer games, such as non-human entities. For example, the term "AI" as well as the marines that control the vehicles. In situations the cars or marines are referred to as "AI" referred to the skill level of the computer. Participants in several different tests were presented to participants.



Figure 1. Portion of Options screen from *Combat Flight Simulator*.

The term "AI Level" was not well understood by the participants. Seven participants did not understand the intended meaning of the term. The term was presented for AI Level (Low, Med, High). The main problem was with the term "AI". For many developers, it is unfamiliar to many participants who had experience with computer games but not with flight games. A poor choice of

can be viewed as an experiment in a given product in the way the

cations in a variety of ways. For as a means to an end, whereas on of gameplay. This distinction appropriate level of challenge and on how efficiently they can ally in productivity applications, es that goals and objectives in times, but that they must be es and productivity applications are relatively few productivity are many games. Games must on, but they must also compete ng television or reading a book ames that are not immediately her forms of entertainment.

ccess to methods drawn from chology that can be adapted to ling current usability methods ved entertainment experiences.

hods that were used on video e first case study begins with idies extend the use of current ne can address some of the

ified using limited-interactive ent from standard usability study was on the game shell (CFS). Game shell screens ions, etc.) which are used to t encountered during actual en change the difficulty level

ent cycle of CFS and used a ens (i.e., functional screen ggles). Techniques utilized

included user and task observations, scenarios, and thinking aloud protocols (e.g., Nielsen, 1993). Some of the tasks presented to users were exploratory and not performance based. In general, the users' tasks were to explore a particular game shell screens while thinking aloud.

In CFS, parameters users can modify using the game shell include aircraft selection, start location, time of day, weather, number of enemies, and difficulty of enemies. During the exploratory task for the Options screen, a usability problem was detected related to terminology. "AI" is an acronym for artificial intelligence commonly used in the video and computer games industry to refer to the behavior of non-human entities. For example, the cars players drive against could be referred to as "AI" as well as the marines that fight along side of players, providing that in both situations the cars or marines are not actively controlled by a human player. In CFS, "AI" referred to the skill level of computer-controlled enemy pilots, which confused participants in several different ways. Figure 1 represents a portion of an Options screen presented to participants.

Difficulty Presets

	☒	☐	☐	☐
	Easy	Medium	Hard	Custom
AI Level	☒	☐	☐	
	Low	Med	High	
Flight Model	☒	☐	☐	
	Easy	Med	Hard	
Weapons Strength	☐	☐	☒	
	Real	Stronger	Strongest	
Tactical	☒	☐	☐	
	Easy	Med	Hard	

Figure 1. Portion of Options screen presented to participants

The term "AI Level" was not well understood by most—in fact, just two of seven understood the intended meaning of the term. In addition, the options presented for AI Level (Low, Med, High) failed to provide helpful context cues. The main problem was with the term "AI". While it is a jargon term for game developers, it is unfamiliar to many gamers. It is worth noting that all participants had experience with computer games, and the majority of them also had experience with flight games. A poor choice of fonts further compounded the problem. The font

used was sans serif resulting in participants confusing the uppercase "I" with a lowercase "l", or the numeral "1". This introduced incorrect cues that some participants used as a basis for guessing (e.g., "is it ALtitude"?). These quotes from participants were representative of the problems they encountered.

"A-One-level? I don't know what that means. I have no idea..."

"A. I. uh...this uh...AL level? A.I. level? I don't understand this terminology."

"A-One-level, I'm not sure what that means, or AL level."

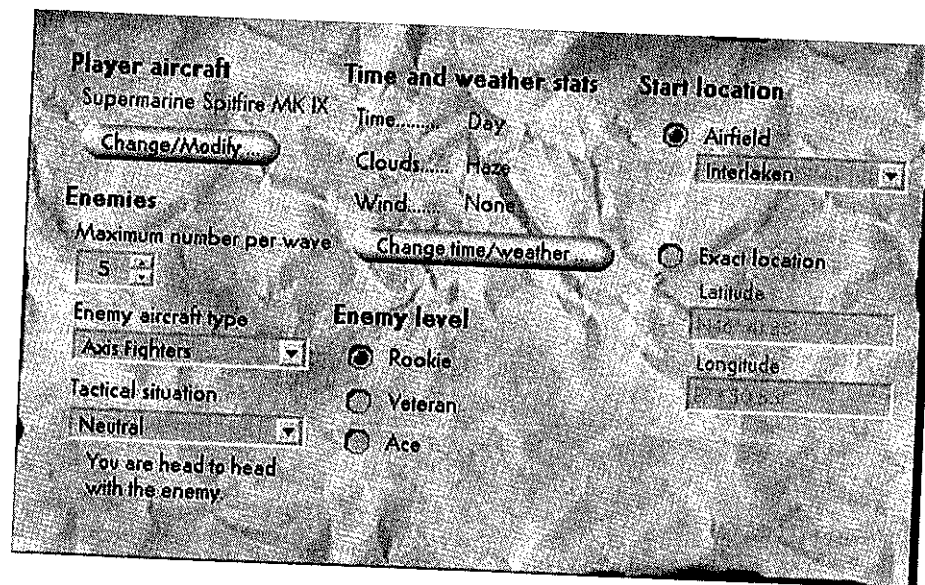


Figure 2. Final interface for Options screen.

Figure 2 represents the final solution. The term "AI Level" was replaced with "Enemy Level". The options that were originally "Low," "Medium," and "High" were replaced with "Rookie," "Veteran," and "Ace". Subsequent testing detected no difficulties with these terms, and the options are relatively clear in the context of a WWII fighter pilot game. This terminology has remained consistent through subsequent releases of the Combat Flight Simulator series.

In this example, information obtained from users through current usability practices was used to improve upon the usability of the game shell. This is a case where the user goals for the options screen did not substantially differ from user goals in productivity applications; the tools were a means to an end. Users should be able to set their preferences with maximum learnability and minimum errors.

2.2 MechWarrior 4: Vengeance

The popular MechWarrior series is fought by elite warriors who pilot Battlemechs. One of the defining features of the series is its action-oriented nature. However, the series attempts to model the complexity of the fun for those who enjoy this type of game. The weapons were designed with limited ammunition, differential firing rates for the Battlemech to overheat and use weapons. This level of complexity is a simplistic solution to this problem because some of the game's fun stems from novice users sufficiently mastering the mechanics of the game. In the following case study, user-testing of novices who experienced difficulties with MechWarrior 4: Vengeance (MW4) was used to define the MechWarrior series.

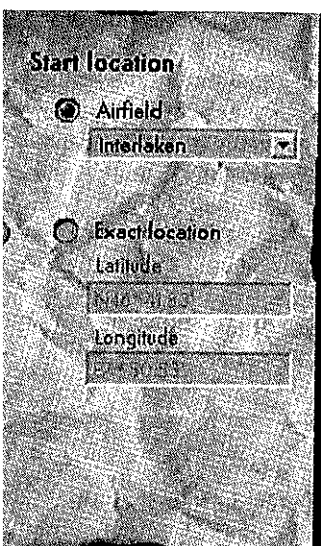
The development team established guidelines to make the game as accessible as possible for novice users. In particular, movement and weapon controls were simplified because Battlemechs are complex because Battlemechs are where the top part of the Battlemech controls its movement trajectory, similar to a tank. The bottom part of the torso is placed in the upper part of the torso and aims weapons while playing the game. On-screen aids were included as part of the game.

Figure 3 is a screenshot of the Battlemech's line of sight. The items on the screen that were created by the Battlemech are within the circular radar placed at the user's line of sight, or field-of-view. The enlarged version of the Battlemech is at the bottom portion of the Battlemech's line of sight would be to the right of the

ing the uppercase "T" with a
ed incorrect cues that some
Latitude"?). These quotes from
ncountered.

1.."

his terminology."



is screen.

AI Level" was replaced with
ow," "Medium," and "High"
bsequent testing detected no
vely clear in the context of a
ained consistent through
es.

s through current usability
ie game shell. This is a case
ubstantially differ from user
as to an end. Users should be
and minimum errors.

2.2 MechWarrior 4: Vengeance

The popular MechWarrior series is based in a science fiction future where wars are fought by elite warriors who pilot giant destructive walking tanks called Battlemechs. One of the defining features in MechWarrior games is that they are action-oriented. However, they are also designed as simulations, which are games that attempt to model the complexity of the real world. Mastering the complexity is part of the fun for those who enjoy this type of game. For example, in MechWarrior, all weapons were designed with limitations that could exist in the real world, such as limited ammunition, differential firing rates for different weapons, and the potential for the Battlemech to overheat and shut itself down with the overuse of some weapons. This level of complexity can be quite intimidating for novice users. A simplistic solution to this problem would be to make the game simpler. However, because some of the game's fun stemmed from its complexity, the goal was to help novice users sufficiently master the intricacies of the game in an enjoyable way. In the following case study, user-testing methods helped identify areas of assistance for novices who experienced difficulties controlling their Battlemechs when playing MechWarrior 4: Vengeance (MW4), without sacrificing the complexity that helps define the MechWarrior series.

The development team established a goal of making MW4 as approachable as possible for novice users. In particular, novices needed to be able to use the basic movement and weapon controls with minimal difficulty. The controls in MW4 are complex because Battlemechs are capable of a movement called torso-twisting, where the top part of the Battlemech can rotate to look in a direction different from its movement trajectory, similar to a gun turret on a modern tank. Weapons are placed in the upper part of the torso, so rotating the upper torso is fundamental to aiming weapons while playing the game. To help users coordinate these movements, on-screen aids were included as part of the standard interface.

Figure 3 is a screenshot of the MW4 interface during gameplay. There are two items on the screen that were created to aid users. The green cone-shaped item within the circular radar placed at the bottom-center of the screen represents the user's line of sight, or field-of-view. It is marked Field of Vision. Figure 4 is an enlarged version. In this situation, the torso is rotated slightly to the left relative to the bottom portion of the Battlemech. If the cone funneled out toward the right, the line of sight would be to the right of the Battlemech.

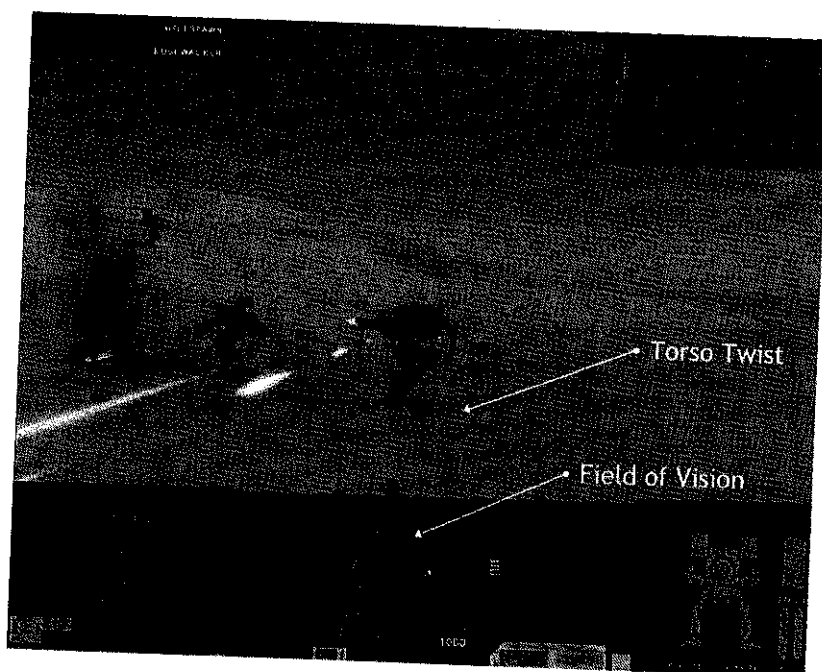


Figure 3. User interface for MechWarrior 4 during gameplay

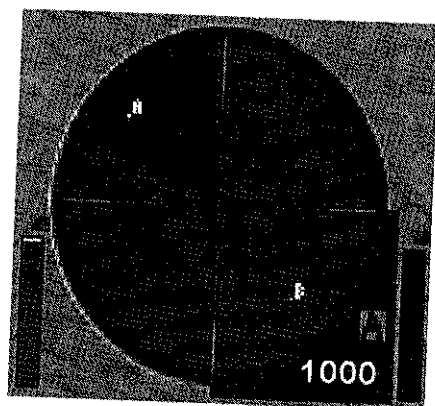


Figure 4. Field of Vision indicator and Radar

The second item is a green horizontal scale placed in the center of the screen beneath the targeting reticle where a horizontal bar represents the amount of torso rotation (see Figure 3). This is marked as Torso Twist. This provided users with a graphical cue for torso rotation on the x-axis relative to the bottom part of the Battlemech. For example, if a Battlemech is facing straight forward (i.e., the torso is

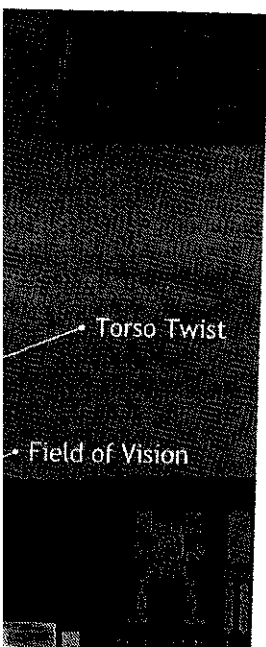
perfectly aligned with the lower reticle. The length of the bar grows as the Battlemech is rotated much further



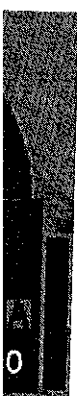
Figure 5. Torso Twist indicator

Usability testing of these indicators showed that most participants used the counting techniques. During this testing, the basic controls of the game were familiar to the participants. The Battlemech's torso, frequently rotated in the direction in which the Battlemech was facing. Of seven participants, only one did not notice the torso twist indicator, and no participants noticed or used the indicator. This demonstrated a skillful use of the indicator in steering their Battlemech.

It became apparent that torso twisting was a learning curve. Based on the feedback, the development team chose to create a training mission to educate users about the visual cues. The training mission decreased the number of torso-twisting errors. In the follow-up study, all participants correctly avoided torso-twisting problems. The critical difference



during gameplay



and Radar

ed in the center of the screen
represents the amount of torso
st. This provided users with a
ve to the bottom part of the
ight forward (i.e., the torso is

perfectly aligned with the lower portion of the Battlemech) there is no horizontal bar beneath the reticle. The length of the horizontal green bar is a function of the distance the torso moves away from the center in either direction. In other words, the length of the bar grows as the Battlemech rotates further from perfect alignment of the upper (torso) and lower portions of the Battlemech. In Figure 5, the green bar represents the Battlemech rotated slightly to the left in panel A. In panel B, the Battlemech is rotated much further away from the center toward the left.

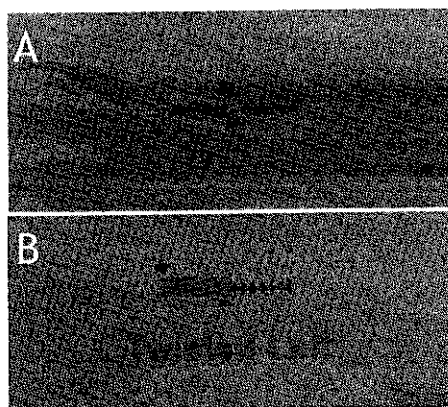


Figure 5. Torso Twist indicator. A) slightly turned to the left, B) severely turned to the left

Usability testing of these interface items included common pass/fail and error counting techniques. During this test it was shown that novice users struggled with the basic controls of the game. In particular, they struggled to control the Battlemech's torso, frequently confusing the direction of the upper torso for the direction in which the Battlemech was moving. The fundamental problem was that few people were using the on-screen cues created to avoid this exact problem. Four of seven participants did not notice or use the field of view reference (green cone), and no participants noticed or used the green rotational bar. Only two participants demonstrated a skillful use of torso twisting, but all demonstrated difficulties steering their Battlemech.

It became apparent that torso-twisting with Battlemechs is a skill with a gradual learning curve. Based on these results and other convergent evidence, the development team chose to create a training mission with the primary goal of educating users about the visual cues in the interface while giving them a relatively safe environment to practice the use of torso-twisting. Follow-up testing showed that the training mission decreased the torso-twist problem, but did not eliminate it. In a follow-up study, all participants knew how to use the on-screen visual cues to correct or avoid torso-twisting problems, though they still struggled to control their Battlemechs. The critical difference between the game prior to the addition of the

training mission and after is that although users still experienced difficulties learning the controls, they were empowered to fix the problems themselves after completing the training mission.

Educating novice players through training makes the game more accessible to a larger population without alienating the current population of MechWarrior gamers. The training mode gives more time to novice users to adapt to the game's complexity, and for more experienced users, the training mode is optional.

2.3 *Halo: Combat Evolved*

A skilled game designer can create an experience that people consistently enjoy. However, as with any product, a user may not use it in the way the designer intended. In the case of video games, this may lead to an experience that is less fun for the user. This case study presents an example from the development of *Halo: Combat Evolved* (Xbox). It presents one of the biggest challenges in game design: making the user feel like they are making interesting decisions while ensuring that they are playing the game in a way that leads to the most fun. This example illustrates how discount usability methods enabled the designers to see how users approached their game, to understand barriers that were preventing users from experiencing the game as intended, and to refine the game so that it drew the user into the optimal experience.

One of the greatest difficulties involved in game design is perfecting the balance between scripting an experience intended by the designer without making the user feel restricted in the choices they can make. For example, a designer may want the user to move the character to a particular location in the virtual world, where an exciting battle awaits them. This could be accomplished by creating a narrow pathway, or by placing barriers in the virtual world restricting the user to only one path, with no other choices. However, such restrictions may be unsatisfying to users because they feel like they have no control over their decision-making. Alternatively, the designer could implicitly control the user's behavior by giving him or her free range of movement, but placing something attractive (e.g., a new weapon) at intermediate points between the user's current location, and the designer's desired location (i.e., the location of the exciting battle). This may draw the user down a certain path, while making him or her feel in control. While tactics such as this are common, there are a number of different variables encountered in game design that interact with one another, making it very difficult to predict exactly how users will play the game.

Different mechanisms affect a user's experience of combat in games. Because *Halo* is a game about combat, it was very important to the team that combat be compelling. The development team spent a lot of time refining the aiming controls, enemy behavior, weapon variety, and the layout of the environments to make combat as enjoyable as possible. In addition, the designers created a wide variety of combat situations and areas where users could engage in different tactics to succeed. Certain situations were created where a particular strategy would yield the most rewarding and fun experience. The assumption was that if users approached the

combat situations as the designers did.

The *Halo* team wanted to see if users would approach the game, and if not, what issues prevented them. For information, eight participants were recruited to play the game in a usability test.

Participants were presented with the game in a way that they would at home. For this test, training was not used. Instead, the focus was on the game's difficulty level in comparison with the 'ideal' level present at all sessions. This was done to see if users wanted players to respond to the cues users were not seeing.

The designers had created a level that gave players freedom to approach situations in a way that the environments afforded players. Testing revealed that participants did not approach the environments away then the designers intended. Participants were shooting at the enemy as soon as they saw them, rather than the players to first move closer to the enemy. This was not able to successfully complete the mission. The level was designed to be most enjoyable when the player was at a distance. At greater distances, the experience was more enjoyable. The experience was not designed to be effective when the player was close, complaining that their weapons were not effective.

From observing the usability issues, the designers found solutions. First, they revised the aiming reticle. The aiming reticle refers to a graphic that represents where the user is aiming. When the user is far from the enemy, the reticle would be small. As the user moved closer, the reticle would grow larger. The user that their weapon was aimed at. The designers changed the reticle behavior so that it turned on when the user was at the combat distance intended by the designers. When the enemy was from a long distance, the reticle would be small. The cue that they were not at an optimal distance. The designers also increased the distance at greater distances. Second, the team changed the way the user shot at the enemy from a much greater distance. The user would fire, but would dodge the shot or move closer. When the user reached the enemy, the user would step out of cover and engage the enemy. The user would approach the player to the enemy.

Subsequent usability testing showed that users were more of users to engage in combat from a distance. The designers found more of the interesting components of the game, and the performance of their weapons, and the

still experienced difficulties learning problems themselves after completing

makes the game more accessible to a population of MechWarrior gamers. Novice users to adapt to the game's training mode is optional.

ence that people consistently enjoy. not use it in the way the designer lead to an experience that is less fun. The development of Halo: the biggest challenges in game design: interesting decisions while ensuring that adds to the most fun. This example enabled the designers to see how users' that were preventing users from the game so that it drew the user

game design is perfecting the balance the designer without making the user. For example, a designer may want the situation in the virtual world, where an accomplished by creating a narrow world restricting the user to only one. Restrictions may be unsatisfying to users. Control over their decision-making. Control the user's behavior by giving something attractive (e.g., a new user's current location, and the the exciting battle). This may draw or her feel in control. While tactics of different variables encountered in making it very difficult to predict exactly

ence of combat in games. Because important to the team that combat be of time refining the aiming controls, out of the environments to make designers created a wide variety of engage in different tactics to succeed. A similar strategy would yield the most. It was that if users approached the

combat situations as the designers intended, users would have as much fun as the designers did.

The Halo team wanted to see whether the typical user would have fun playing the game, and if not, what issues were blocking them from doing so. To provide this information, eight participants who liked to play console action games were recruited to play the game in a usability test format.

Participants were presented with a mission and asked to play the game as they would at home. For this test, traditional error counts and pass/fail criteria were not used. Instead, the focus was on the tactics that participants used to complete the level in comparison with the 'ideal' tactics created by the designers. Designers were present at all sessions. This was critical because they best understood how they wanted players to respond to their game, and they could also see, first hand, what cues users were not seeing.

The designers had created large, outdoor environments that gave players more freedom to approach situations in a variety of ways. However, the large environments afforded players the ability to see enemies from a long distance. Testing revealed that participants were trying to fight enemies from much further away than the designers intended. During usability sessions, novice players began shooting at the enemy as soon as they could see them, but the designers had wanted the players to first move closer to the enemies, and then fight. While all users were able to successfully complete the missions using these tactics, the game was designed to be most enjoyable when combat was engaged at a much closer distance. At greater distances, the experience was less satisfying. For example, the weapons were not designed to be effective at these longer distances resulting in participants complaining that their weapons were not accurate enough.

From observing the usability sessions, designers quickly identified several solutions. First, they revised subtle cues that they had placed in the aiming reticle. The aiming reticle refers to a graphical indicator in the center of the screen that represents where the user is aiming a weapon. In the original design, at any distance from the enemy, the reticle would turn red when placed over an enemy, informing the user that their weapon was aimed properly at the enemy. The designers modified the reticle behavior so that it turned red when the enemy was targeted, but only at a combat distance intended by the designers. Therefore, when the player aimed at the enemy from a long distance, the reticle would not change color, providing a subtle cue that they were not at an optimal distance, thus encouraging users to move closer. The designers also increased the diameter of the reticle to emphasize its inaccuracy at greater distances. Second, the team modified the enemy behavior. When the user shot at the enemy from a much greater distance, the enemy now would not return fire, but would dodge the shot or move behind a rock or tree, requiring the user to move closer. When the user reached the optimal engagement point, the enemy would step out of cover and engage in battle with the player. At other times, the enemy would approach the player to maintain the intended combat distance.

Subsequent usability testing showed that these changes encouraged the majority of users to engage in combat from the intended distance. Users were able to see more of the interesting components of the game, were more satisfied with the performance of their weapons, and as predicted, enjoyed the game more.

In standard usability practices, the goal of the test is to ensure that the design of product effectively supports what the user is trying to accomplish. In this case study, the perspective of usability test goals has been shifted away from the end user, and toward the designer. In the Halo example, the goal was to ensure that the user experience matched designer intention. These types of situations often require less structure than more common usability methods.

Once we are confident that users are playing the game in the way that designers intended, we can begin to get feedback about subjective preferences on the game's design. Although this may not require direct observation, a larger number of users must be utilized to ensure reliability of participant response, as the next case study illustrates.

2.4 RalliSport Challenge

RalliSport Challenge (RSC) is a racing game on the Xbox console. As in many popular racing titles, the Career Mode is one of the core pieces of this game. In order to understand how user-centered design methods were able to influence the design of this game, one must first understand how the Career Mode functions in RSC.

The Career Mode starts with a limited set of cars available to the user, and a limited set of driving events they can participate in. Depending on the users' performance, new cars and more challenging events are made available to users as they progress through their careers.

Each event consists of several stages (or races). After each stage, users gain points based on a number of factors; top speed, what place they finished (e.g., 1st, 2nd, etc.), lap times, and amount of car damage. Once an event is over, all of the points gained from each stage are added up to obtain the user's total points. The user's goal is to accrue a specified number of points to gain access to the next set of cars and events that are currently unavailable to them. Reaching the specified point total is the only way to progress through the game. This is often referred to as "unlocking" cars and events.

This design is not unique in racing games. It is common to specify a certain performance criterion in order to progress through the game. However, keeping in mind that playing games is a choice (i.e., users are not forced to play), the manner in which this design is implemented could cause frustration for many users. In RSC, the issue revolved around the users' ability to re-race a stage within an event to earn more points.

To better illustrate this issue, consider a scenario where a user has three racing events that are currently available to them, each comprising several stages: Event A (stages A-1 through A-4), Event B (stages B-1 through B-5), and Event C (stages C-1 through C-6). To unlock the next set of events and cars, the user must accumulate a minimum set of points by racing through most (and sometimes all) of the stages in Events A, B, and C. If a user races all Events, but does not earn enough points to unlock new cars and events, what options are they left with? According to the developer's design, their only option would be to re-race an entire Event, including all stages within that Event. If the user selected Event C, all of the previous points

earned from each stage (C-1 through stages C-1 through C-6).

The developers preferred the easy otherwise, and also because rally racing. This is a valid argument the appropriate level of challenge potentially puts users in the situation are already happy with. If the user C-6, but performed very poorly, could result in extreme frustration.

To address this issue, a series of situations may be to users. The goal as possible a situation where users but not accrue enough points. Suffice; users had to experience usability study sample size would preference. Therefore, a survey size and not require direct observation.

Working with designers, a model Events, each with a minimum of population of users was created. In racing console games were based standard experimental research put up so that responses from each interaction between users.

Participants were instructed to were at home. Once they finished earn enough points to unlock the next were given the following options: unlocking the next set of events:

- Re-race any of the previous 3 stages. However, you must race ALL of them.
- Re-race CERTAIN STAGES in order to reach your total score;

Then participants were asked to unlock the next event.

- Re-racing an entire event (including all stages).
- Re-racing any stage of my own choice.

Lastly, users were asked the same question. The Events consisted of nine stages in the actual game, although they did not know this.

The data gathered provided feedback on re-race the entire Event. The majority of users were able to re-race a single stage as opposed to the entire event. This became even stronger when presented with the percentage of users preferred to be able to re-race a single stage.

s to ensure that the design of
accomplish. In this case study,
away from the end user, and
was to ensure that the user
situations often require less

ne in the way that designers
e preferences on the game's
on, a larger number of users
onse, as the next case study

Xbox console. As in many
pieces of this game. In order
able to influence the design
mode functions in RSC.

available to the user, and a
Depending on the users'
made available to users as

fter each stage, users gain
ace they finished (e.g., 1st,
an event is over, all of the
he user's total points. The
in access to the next set of
eaching the specified point
his is often referred to as

mon to specify a certain
ame. However, keeping in
ced to play), the manner in
1 for many users. In RSC,
age within an event to earn

re a user has three racing
ng several stages: Event A
5), and Event C (stages C-
the user must accumulate
etimes all) of the stages in
not earn enough points to
t with? According to the
an entire Event, including
all of the previous points

earned from each stage (C-1 through C-6) are erased, and the user must then re-race stages C-1 through C-6.

The developers preferred this model because they felt the game would be too easy otherwise, and also because this system is much closer to the actual sport of rally racing. This is a valid argument. Games should not be too easy and must ensure the appropriate level of challenge and success. On the other hand, this strategy potentially puts users in the situation where they lose points on several stages they are already happy with. If the user performed very well on C-1, C-2, C-3, C-5, and C-6, but performed very poorly on C-4, they would lose points on all stages, which could result in extreme frustration.

To address this issue, a series of tests was designed to assess how frustrating this situation may be to users. The goal in designing these tests was to simulate as close as possible a situation where users would actually play through a series of events, but not accrue enough points. Simply asking users about this situation would not suffice; users had to experience it. In addition, data collected from a traditional usability study sample size would not be sufficiently reliable to measure a subjective preference. Therefore, a survey methodology that would allow for a larger sample size and not require direct observation was used.

Working with designers, a mock career mode was created that consisted of three Events, each with a minimum of four stages. Secondly, a profile that described the population of users was created. In this case, nearly thirty users who frequently play racing console games were brought in. To prevent contamination of the data, standard experimental research protocols were implemented. The laboratory was set up so that responses from each user could be collected independently, with no interaction between users.

Participants were instructed to play through the mock career mode as if they were at home. Once they finished the events, they were informed that they did not earn enough points to unlock the next set of cars and events. At this time participants were given the following options, and asked which option they would use for unlocking the next set of events:

- a) Re-race any of the previous 3 EVENTS to improve your total point score;
However, you must race ALL stages within the EVENT(S) you choose;
- b) Re-race CERTAIN STAGES in any EVENT of your own choosing to improve your total score;

Then participants were asked how they felt about the following methods for unlocking the next event.

- a) Re-racing an entire event (including all tracks within the event);
- b) Re-racing any stage of my own choosing within an event;

Lastly, users were asked the same question, but presented with a situation where the Events consisted of nine stages (which was representative of some Events in the actual game, although they did not experience this).

The data gathered provided fairly convincing evidence against forcing users to re-race the entire Event. The majority of participants reported a preference for being able to re-race a single stage as opposed to an entire event and this preference became even stronger when presented with the nine stage event. A smaller percentage of users preferred to be able to re-race only entire events, suggesting that

they agreed with the developers, who felt that allowing users to re-race stages made the game too easy.

Based on these survey data, the recommendation was to implement multiple difficulty levels, one of which allowed the re-race option (Normal difficulty), but the more advanced levels that did not (Difficult or Advanced difficulty). This solution met the needs of both user preferences, those who want to be able to re-race any stage at any time, and "hard-core" users who prefer more of a challenge and penalty when playing through their career. This was presented to the developer, who in turn implemented the functionality into the design of the final retail product--Normal (no re-race) and Easy (re-race). The developer felt the no re-race option was truer to the sport of rally racing, which is why it was termed Normal. More importantly, the decision to implement multiple difficulty levels was also validated by some reviews from popular gaming magazines and gaming sites after the game was released.

...punishingly difficult and repetitive, depending on which of the two available skill levels you choose when you start the game....With no "re-do" button available in Normal mode to simply restart a race if you screw up, some gamers will find themselves angered and frustrated with having to start over from the beginning of the series, while others will embrace the degree of challenge with open arms - Mahood, 2002

We were also ready to slap the game down for being too hard, until we tried it on the easy difficulty setting. Normal difficulty, as well as giving you some stiff times to beat, doesn't allow you to retry a stage - if you mess up, you have to restart the whole event. Even racing veterans would be well advised to play Easy mode to get to grips with the tracks at first, as this allows you to retry. - Smith, 2002

In this example, the question being addressed (to allow re-racing or not) did not have a clear answer before data were collected, and did not have a specified task-based performance criterion. Finding a solution for this design question was not necessarily related to the learnability of the product, or the efficiency of use, or the number of errors a user would make. The solution was based on what was most fun for users.

3. CONCLUSION

There are a variety of approaches that can be taken from existing HCI-related methods to address the unique aspects of user-centered design issues found in games. Although the case studies discussed above focus on specific issues, they should also be viewed as a broad representation of some methods currently being utilized by the User-Testing Group at Microsoft Game Studios. Each of these examples is part of a larger series of tests performed throughout the development cycle of each of the games. It is not sufficient to run a single test for one game. As in any HCI-related field, to be maximally beneficial, user-testing must be integrated into the development process from the very beginning.

Each of the methods presented in the case studies has limitations, and should only be used when appropriate. A good user-testing engineer must understand the points in the development cycle at which certain methods should be used to address

particular problems, and conv collected through user testing.

In general, standard usability subjective preferences, due to studies. With fewer participants. However, usability methods provide rich information about provide a designer with greater their game.

Larger sample methods are are limited because they rely on a user's actions, are better in traditional usability testing, to design a game. Users can not"). From there, it is up to the to define the root causes of issues.

Unfortunately, there is no variability in the process through the methods used for addressing

Furthermore, there are many the user experience when playing utilized in testing adults proved from children on child-oriented a game (when two or more per difficulties. Interactions between observations are independent, the results (i.e., subjective ratings within the game, or pre-existing

Finally, most of our research Information based on this stage and developers are so immersed experience is like for someone development teams would also occurs beyond the initial experience has many practical constraints. provide cost-efficient user-testing the user experience.

For more in-depth case studies 2, Oddworld: Munch's Oddysee of Empires II).

owing users to re-race stages made

ation was to implement multiple option (Normal difficulty), but the dvanced difficulty). This solution io want to be able to re-race any rs more of a challenge and penalty nted to the developer, who in turn e final retail product--Normal (no no re-race option was truer to the d Normal. More importantly, the as also validated by some reviews after the game was released.

which of the two available skill no "re-do" button available in some gamers will find themselves he beginning of the series, while rms - Mahood, 2002

too hard, until we tried it on the ring you some stiff times to beat, a have to restart the whole event. isy mode to get to grips with the

to allow re-racing or not) did not d did not have a specified task- or this design question was not t, or the efficiency of use, or the was based on what was most fun

aken from existing HCI-related entered design issues found in e focus on specific issues, they f some methods currently being Game Studios. Each of these ed throughout the development a single test for one game. As in user-testing must be integrated

g. lies has limitations, and should g engineer must understand the thods should be used to address

particular problems, and convince development teams of the general value of data collected through user testing.

In general, standard usability methods are not appropriate when assessing subjective preferences, due to the small sample size that accompany typical usability studies. With fewer participants, reliability of the results comes into question. However, usability methods used in conjunction with more open-ended tasks provide rich information about how a user interacts with a product which can provide a designer with greater understanding of how a user approaches and plays their game.

Larger sample methods are good for assessing subjective preferences, but they are limited because they rely on self-report. Usability studies, which typically focus on a user's actions, are better for obtaining in-depth, behavioral information. Just as in traditional usability testing, it is also impossible to look to the user to tell you how to design a game. Users can only evaluate aspects of the game ("this is fun", "that is not"). From there, it is up to the user-testing engineer and designer to work together to define the root causes of issues and create compelling solutions.

Unfortunately, there is no set formula for how the pieces fit together. The variability in the process through which games are developed must be reflected in the methods used for addressing user-testing issues.

Furthermore, there are many challenges with obtaining feedback on all areas of the user experience when playing games. For example, the survey methods that we utilized in testing adults proved to be problematic when trying to obtain feedback from children on child-oriented games. Additionally, testing multiplayer portions of a game (when two or more people play the same game together) presents further difficulties. Interactions between participants do not allow us to assume that the observations are independent, which makes it difficult to know the extent to which the results (i.e., subjective ratings) are influenced by the players' social interactions within the game, or pre-existing social relationships between them.

Finally, most of our research focuses on the initial experience of a game. Information based on this stage of the user's experience is critical because designers and developers are so immersed in the game that they often forget what the initial experience is like for someone who has never played their game. While many development teams would also like to obtain detailed feedback about gameplay that occurs beyond the initial experience, doing so with current development practices has many practical constraints. As such, these issues need to be explored further to provide cost-efficient user-testing methods to obtain feedback on all these areas of the user experience.

For more in-depth case studies, see Pagulayan et al., (2003) (MechCommander 2, Oddworld: Munch's Oddysee, and Blood Wake) and Medlock et al., (2002) (Age of Empires II).

4. ACKNOWLEDGMENTS

Special thanks to the development teams on CFS, MW4, Halo, and RSC and Digital Illusions. In particular, we'd like to express our gratitude to Christina Chen, TJ Wagner, David Luehmann, Jaime Griesemer, John Howard, Hamilton Chu, and Peter Wong for permitting us to use the actual games in our examples. We'd also like to thank the Microsoft Game Studios User-Testing Group, in particular, Lance Davis, Kyle Drexel, Kevin Goebel, David Quiroz, Bruce Phillips, John Davis, and Kevin Keeker.

5. REFERENCES

- Mahood, A. (2002). RalliSport Challenge. *Official Xbox Magazine*, April 2002. Brisbane, CA: Imagine Media, Inc.
- Medlock, M. A., Wixon, D., Terrano, M., Romero, R. L., & Fulton, B. (2002). Using the RITE Method to improve products; a definition and a case study. *Humanizing Design, Usability Professionals' Association 2002 Annual Conference*, 2002 July 8 - July 12; Orlando, Florida: Chicago, IL: Usability Professionals' Association, 2002.
- Nielsen, J. (1993). *Usability Engineering*. New York: Morgan Kaufmann.
- Pagulayan, R. J., Keeker, K., Wixon, D., Romero, R. L., Fuller, T. (2003). User-Centered Design in Games. In J. Jacko & A. Sears (Eds.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications* (pp. 883-906). Mahwah, NJ: Lawrence Erlbaum Associates.
- Smith, M. (2002). Reviews: RalliSport Challenge. *British Telecommunications plc*.

KRISTINA AND

PLAYING GA

The nature of user studies is c relevant to study and *how* to st progressively moved from the people's actions but the social behaviour. Now, with the so-ca and pervading people's persona the user; in order for a produ satisfying and pleasurable i methodological point of view t application is pleasurable for a the personal sphere and useful emotions? How do we interpret

The work of Gaver et al. (1 or creative techniques can be s are both provocative and aes experiences is a possible way revealing data. We believe tha preliminary studies in the design produce are not scientific in a manner. While designing for p be measurable and must be inter

This paper presents the app aim of the project was to exp communication between peop perception of distance; an addit which to access and study peopl investigations, we propose cor products we design and the me

Mark A. Blythe, Andrew F. Mo
Funology: From Usability to En
© 2003 Kluwer Academic Publish