PLAYER BALANCING FOR FIRST-PERSON SHOOTER GAMES

A Thesis Submitted to the College of

Graduate Studies and Research

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In the Department of Computer Science

University of Saskatchewan

Saskatoon, CANADA

By

Rodrigo Vicencio-Moreira

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

ABSTRACT

When player skill levels differ widely in a competitive First-Person Shooter (FPS) game, enjoyment suffers: weaker players become frustrated and stronger players become less engaged. *Player balancing* techniques attempt to assist the weaker player and make games more competitive, but these techniques have limitations for deployment when skill levels vary substantially. In this thesis, we developed new player balancing schemes to deal with a range of FPS skill difference, and tested these techniques in a series of five studies using a commercial-quality FPS game developed with the UDK engine. Our results showed that our balancing techniques (*Combo* and *Delay*) are extremely effective at balancing, even for players with large skill differences. These techniques also led to higher enjoyment of the game by players of all skill levels. Our studies are the first to show that player balancing can work well in realistic FPS games, providing developers with a way to increase the audience for this popular genre. In addition, our results demonstrate the idea that successful balancing is as much about the way the technique is applied as it is about the specific manipulation.

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisors Regan Mandryk and Carl Gutwin for their guidance and input throughout my graduate career at the University of Saskatchewan. I am sincerely thankful for everything they have done for me. A big thank you to the faculty and staff at the Department of Computer Science, and particularly to the students and staff in the HCI Lab for their help and their friendship.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

2D	Two-Dimensional
3D	Three-Dimensional
CD Ratio	Control-to-Display Ratio
DDA	Dynamic Difficulty Adjustment
FPS	First Person Shooter
UDK	Unreal Development kit

CHAPTER 1

INTRODUCTION

The video game industry is a booming business, with almost 23 billion dollars spent on video games in 2014 in the US [20]. The First Person Shooter (FPS) genre, which includes games such as the *Halo* [16] series shown in Figure 1.1, makes up a substantial part of those sales, with about 21% of total game units sold in 2014 being classified as FPS games [20]. FPS games require skill and quick thinking to navigate through complex 3D environments while dodging enemy fire and firing back. FPS games are popular due to their action oriented nature and the tough challenges that they present to players [24]. The social and multiplayer aspects available in most FPS games also add to their success; many people enjoy playing with friends or competing with others online [31], [42].

Figure 1.1: Screenshot of *Halo* [16].

While the multiplayer aspect of these games may be a major draw for playing first person shooters, the fact that not all players have the same level of skill is a problem. A player with greater skill can easily defeat the novice player, leading to a poor play experience for both players. Players whose skills are greater than their opponents become bored and unengaged with the game, and players who are not as skilled become frustrated because they have no chance of winning [15][31][33]. As a result, players have less fun and are less likely to continue playing the game. This is a major concern for game developers, because when a game isn't fun, players will not want to keep playing or buy these kinds of games in the future. This is an issue that is present in virtually all competitive skill based games [54]; however, in this thesis we focus on solving this problem of differential skill in FPS games as they are the second bestselling video game genre in the US [20].

The mechanism that will be used in this research to address the problem of different skill levels is player balancing. Balancing is the term used to describe the process of ensuring that the difficulty of the game is properly matched to player skill levels. Designers need to ensure that their games are balanced in order to make sure a player is not overwhelmed or underwhelmed by the difficulty presented in a game. Balancing is well established in single-player games, where game balancing involves changing in-game variables such as health points or the number of enemies [3], [15]. In multiplayer games, however, ensuring balanced gameplay is a more difficult undertaking.

In multiplayer games, the challenge a player faces is determined by an opponent's skill rather than by elements in the game itself. Because the source of challenge is another player, there is no way for game developers to ensure that when two or more players face off in a game they will experience the optimal level of challenge. Game developers cannot completely control the difficulty of the in-game challenges because the skill level of the opponent players will vary across matches. Player balancing is the term that refers to this type of balancing between players. There are a few different approaches to player balancing that are currently being used; however, they all have limitations at effectively creating balanced gameplay. For example, a common player balancing technique is to use a matchmaking system; however these kinds of systems are limited by the number of suitable players, which makes it difficult to find an appropriate opponent (and rules out situations where friends of different skill levels want to play with each other). Multiplayer game balancing is an important issue in order to retain player and have them enjoy the product, and it requires an effective solution that currently isn't available in FPS games.

In this thesis, we propose a new player balancing solution for multiplayer FPS games that allows players of almost any skill level to play together. Our solution provides assistance with several game mechanics in order to help weaker players compete. One main mechanism that we manipulate is aim (aiming assistance), because aiming is a major factor that sets experts apart from novices in FPS games. Aiming assistance improves the accuracy and speed of target acquisition by modifying factors such as the size of the target or assisting the user's aiming trajectory. The solution also incorporates other gameplay elements such as damage rates and knowledge of the map, to add to the effects of aiming assistance when necessary. The results of our studies show that the two best player balancing techniques developed, *Combo* and *Delay*, can be extremely effective at creating closer scoring matches without compromising the perceptions of fairness and leads to players enjoying the game more than without the techniques. These results indicating that our system should be considered by game designers who wish to retain players and engage a wider audience.

1.1 PROBLEM

The problem we address in this thesis is that there is currently no effective solution to balance gameplay in multiplayer First Person Shooter games when players have different skill levels.

Having a game that balances the difficulty level with the skill level of the player is important. If a game does not achieve this optimal balance, the player will feel discouraged and frustrated because the game is too hard, or will become bored and unengaged because the game is too easy [10], [65].

While there are many ways to ensure balance in single player games, it is more difficult to balance multiplayer games. In single player games, the game difficulty can be manipulated by changing values such as health, and by modifying the difficulty of the computer controlled opponents. In multiplayer games, the game challenge is defined by the opponent, whose skill level varies. It is important to achieve balanced gameplay even in multiplayer situations, because imbalanced games result in players not having fun. Additionally, the amount of skill needed to play a game might seem daunting for some players and discourage them from playing. If an effective balancing mechanism existed for multiplayer FPS games, game designers could attract a larger and more diverse audience to the game, which could positively affect sales as the game would be more appealing and fun.

The issue of unbalanced multiplayer games has led to several proposed solutions by game developers and game researchers. However, most of these solutions do not adequately solve the problem. The various shortcomings with these current solutions are described in Chapter 2. One solution, Skill Assistance, is a fairly new gameplay balancing mechanic that has shown promise in some genres [10][17]. Skill assistance helps players with the core mechanics that are needed to play a game. For example, skill assistance in racing games helps users stick to the middle of the track [17]. In FPS games, a major factor that differentiates expert players and novices is aiming accuracy. Therefore, we start with *aiming assistance* as a primary manipulation for balancing FPS games. Aiming assistance takes concepts developed in traditional 2D interface pointing [7] and applies them to video games. Aiming assistance helps users to acquire targets more quickly and more accurately by changing factors involved with aiming, such as the size of the target or the user's aiming trajectory. Aiming assistance has been shown to be effective at improving competition in 2D shooter games where players have uneven skills, using techniques developed from traditional 2D targeting assistance work [7], [10], [28].

While aiming assistance has been shown to be effective in 2D, little information is available as to how effective it could be in 3D. While some 3D targeting assistance has been used previously in some commercial games, there is little publicly available information that indicates their effectiveness. Therefore, it was unknown if the solution that worked in a 2D environment could even work in 3D.

1.2 SOLUTION

The solution presented in this thesis is to provide the first effective player balancing technique for multiplayer FPS games. We had three goals in developing our solution: first, the system should be able to handle a large range of skill differences; second, the system should be able to effectively balance gameplay in a real world FPS game; third, we wanted a system that would increase player enjoyment without being seen as unfair.

In order to determine if player balancing techniques could result in closer matches between players of differing skill levels, and to determine if player balancing is something players would enjoy, we developed an FPS game named *Mega Robot Shootout* with the Unreal Development Kit (UDK) [22] that is as close to a commercial game as possible. Our player balancing techniques were tested and refined through several studies to determine if they would be effective at balancing an FPS game, and to determine which ones would be the most effective at equalizing performance and increasing enjoyment. In the end, the results showed that two techniques (called *Combo* and *Delay*) were the most successful at achieving the goals of balanced gameplay and increased player enjoyment.

Figure 1.2: In-game screenshot of the game used in this thesis, *Mega Robot Shootout*. Minimap (top right) shows player location in white and enemies in red.

1.3 STEPS IN THE SOLUTION

In order to achieve balanced gameplay in multiplayer FPS games, several steps were completed during the research process.

1.3.1 Develop suitable aiming assistance techniques

Since aiming is an important part of FPS games, it was decided that the investigation into player balancing would start by helping improve the accuracy of the weaker player to bring them up to the level of their opponent. Several methods have already been developed to increase accuracy of targeting tasks in computer environments [6], [7], [11], [12], [18], [19], [27], [34], [40], [47], [73], [74]. However, these were all created to work in 2D interfaces. Additionally, some video games had already used some form of aiming assistance in a non-balancing context. Therefore, this first step consisted of deciding which of the 2D techniques could work in 3D and which techniques currently being used could possibly be useful for player balancing. This task was accomplished by conducting a literature review of 2D targeting assistance work, and by reviewing existing aiming assistance techniques in commercial video games. This step resulted in five different aiming assistance techniques that seemed suitable for player balancing in FPS games.

1.3.2 Determine aiming assistance performance in 3D

The second step was to determine the efficacy of the chosen aiming assistance techniques at increasing aiming performance in three dimensional environments, as there was no previously available information on the topic of aiming assistance performance in 3D. We carried out three user studies in order to examine the performance of five different aim assist techniques identified in the previous step. These three studies incrementally added different "real world" game elements to determine how they would interfere with aiming assistance effectiveness. First, the five techniques were tested in a simple 3D "shooting gallery" level with stationary targets (similar to how 2D techniques are evaluated). The second study involved testing the performance of the techniques in a full, "real world" FPS game. In the third study, we examined how different game elements changed the performance of the aiming assistance techniques.

In the three studies, the performance of five aiming assistance techniques were compared with each other and with a control condition where no assistance was applied. This stage provided initial insight into how game factors affect aim assistance, and identified techniques that show potential for player balancing. It also provided the first publically available data set of 3D aiming assistance performance. This step is described in Chapter Four.

1.3.3 Test aiming assistance in real FPS games

In the third step, we applied what we learned in the second step and using the recommended techniques in a multiplayer setting. Participants were split into novice and expert pairs and asked to play a one-on-one "deathmatch" game with the aiming assistance techniques being applied dynamically based on performance. The study consisted of five one minute-long rounds in which players would compete to get the highest score. This initial attempt showed that improving the aiming accuracy of novice players is only the first step in effectively balancing a game, and the participant feedback suggested that novices had more deficiencies than just aiming. This resulted in an iterative process in which we improved the techniques to account for the additional complexity present in the multiplayer situation. This step is described in Chapter Five.

1.3.4 Determine if new player balancing techniques are effective at balancing real FPS games

Finally, the improved techniques from step three were tested in a final study. From the results in step three, we realized that in order for assistance to be successful, we couldn't simply focus on a single element of gameplay. Novice players still had problems in other skill areas (such as finding their opponent and staying alive long enough to shoot back), so assisting players in aiming alone wasn't effective enough to balance them with the more experienced player. Therefore, in step four we developed new player balancing techniques based on three ideas: extending the strength of assistance, changing the way that the assistance is applied, and assisting players in other areas than just aiming. A study was conducted to determine if our new techniques were powerful enough to create balanced gameplay, as well as to determine if the enjoyment of participants increased with the techniques. This step resulted in the final recommendations about player balancing, and is covered in Chapter Six.

1.4 EVALUATION

In order to determine if our developed techniques effectively balance gameplay and increase enjoyment, we examined the different techniques in a multiplayer FPS game, in which we matched up players of largely different skill levels. Participants played through a FPS match with different player balancing techniques, and the results were compared against the control (no balancing) condition. We measured several kinds of performance metrics (Kills, Hit ratio, Score Differential, and number of Lead Reversals) to determine if the games resulted in closer score. We also had several subjective questions we asked of the participants (Competence, Autonomy, Relatedness, and Enjoyment) to determine their subjective experience of play after using each technique.

1.5 CONTRIBUTIONS

The primary contribution presented in this thesis is the first effective technique for balancing multiplayer FPS games. By creating and evaluating player balancing mechanics, we provide empirical data showing that our player balancing solution works, and should be considered by game designers as a method to increase the audience for this popular and competitive genre. Our results show that effective player balancing creates closer scoring matches and more outcome variability, which leads to greater enjoyment for both unassisted (expert) and assisted (novice) players. Our results also show that the two most effective techniques are *Combo* (which combines several skill assistance techniques) and *Delay* (which applies assistance longer).

There are also several secondary contributions:

1. We provide the first publically available empirical data on the performance of aiming assistance in 3D environments.
2. We show how different game factors affect aim assistance
3. We show that the targeting techniques that do not modify the aiming process of the user are more effective than the techniques that do modify the aiming process.

4. We show how different aiming assistance techniques react to certain in game situations, and provide recommendations for which technique would be optimal in which situations.
5. We show that combining different types of skill assistance is more effective than only assisting players with one game mechanic.
6. We show that despite significant assistance sometimes being provided, assisted players were not able to perceive that they were being assisted.
7. We show that game matches with assistance were perceived to be more fair by the weaker (assisted) player, but assistance did not reduce the fairness rating of the stronger player.

1.6 THESIS OUTLINE

Chapter Two presents a survey of related research and practices which form the foundation for the research presented in this thesis. The current solutions to multiplayer game balancing and their shortcomings are discussed. Following this, we cover the background of aiming assistance - mainly Fitts' Law and virtual pointing. This leads to an examination of aiming assistance solutions that are currently available on the video game market. Finally, previous research on player enjoyment of video games and how difficulty levels relate to player enjoyment is discussed.

Chapter Three describes the design and implementation of the aiming assistance techniques that were developed, based on traditional 2D techniques and current solutions in commercial games.

Chapter Four details the first set of studies that we conducted, which investigated how well aiming assistance would work in a 3D FPS game. To do this, we had expert and novice players play through a single player map several times, using a different aiming assistance technique each time. We then compared the results of each technique with the control condition. Three separate studies were required to completely determine aiming assistance effectiveness, and each one is presented in detail. The final results answer the questions of aiming assistance effectiveness in 3D, and indicates which of the aiming assistance techniques are promising for player balancing.

Chapter Five presents our initial investigation of the efficacy of aiming assistance for player balancing in multiplayer FPS games. Using the recommended aiming assistance techniques from Chapter Four, expert players were matched with novices and competed in a multiplayer FPS match. In some rounds, novices had aiming assistance techniques applied as they fell behind in the score. The findings indicated that targeting assistance alone was not enough to completely balance gameplay between players with largely varying skill levels.

Chapter Six details our third study that used what we learned in Chapter Five to come up with more effective balancing mechanisms. Three new player balancing techniques for FPS games are presented, which are better designed to overcome large skill differences than the techniques used in Chapter Five. The performance and subjective data are also presented. The findings indicated that the mechanism we developed created balanced gameplay and increased game enjoyment of all players.

Chapter Seven summarizes the research presented in this thesis, discussing the findings, contributions, and recommendations for aiming assistance and player balancing in FPS games. Future work as a result of this thesis is also discussed.

CHAPTER 2

RELATED WORK

In this chapter, we cover three main areas of research that serve as the background of our work. First, we look at the four current solutions to multiplayer game balancing and their strengths and weaknesses. Next, since aiming assistance is used as our initial solution, we examine the mechanics behind it - mainly Fitts' Law and virtual pointing. An examination of aiming assistance solutions that are currently available on the video game market is also covered. Finally, knowing what motivates players to play and enjoy games is important to determine if our balancing techniques are successful. Therefore our third section examines research in the area of psychology that attempts to understand what drives players to play video games, as well as how difficulty levels relate to player enjoyment.

2.1 PLAYER BALANCING

Flow describes a state in which people are engaged and interested in their tasks without being overly anxious [65]. In order to provide players with a state of flow, game designers try to achieve game balance [25]. A game is considered balanced when the challenge of the game matches the skill of the player [43], [65]. Unbalanced game elements may cause players to become frustrated or unengaged because they find the game too difficult or too easy [65]. At this optimal balance level, the player is not overly anxious or bored, meaning they are experiencing an optimal level of enjoyment.

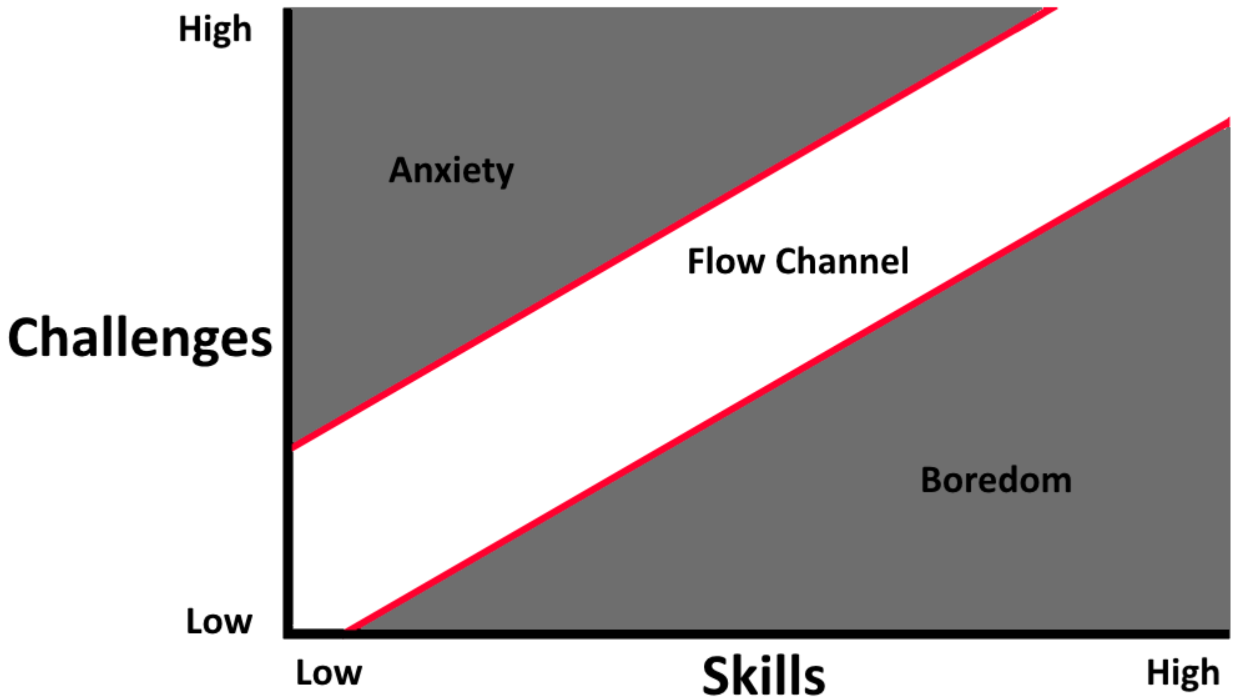


Figure 2.1: The flow diagram shows the optimal level of challenge given a level of skill provides optimal "flow"

This optimal level of balance between game difficulty and player skill is difficult to achieve because people's abilities vary, and are based on many factors including their experience, reaction times, and skill. While difficult, it is important to get balance right because unbalanced games lead to less enjoyable experiences due to players becoming frustrated or unengaged when the game is either too difficult or too easy. If players aren't enjoying themselves, they will not be willing to play the game [54][72]. For this reason, the game industry makes use of simulations and extensive play testing to discover unbalanced game elements [15].

Competition has been shown to be key for the enjoyment of video games [72]. But in competitive scenarios with other human players it can be even more difficult to keep the game balanced than in single player games. In multiplayer games, players can be at very different skill levels [4]. Keeping the definition of balance in mind, in order for a multiplayer game to be balanced, all players should have an equal opportunity to win [3]. This type of balancing

(balancing amongst multiple players of variable skill or ability) is *player balancing*. Player balancing is important in ways that are similar to game balancing – the weaker player can become frustrated if they are losing and the skilled player can become bored because they do not have a challenge. When the players are evenly matched, enjoyment is increased for both parties [10][31].

This behavior was demonstrated in a study of competition between siblings during gameplay, in which the younger child became upset or lost motivation to continue playing if they were never able to perform as well as the older sibling [33]. While this study was aimed at investigating the multiplayer habits of young children, the findings likely extend to interactions in other multiplayer situations.

The issues of balancing competitive multiplayer games have led to research into ways of achieving this balance. Previous game research to solve the gameplay balance problem can be categorized into four general categories: difficulty adjustment, matchmaking, asymmetric roles, and skill assistance.

2.1.1 Difficulty Adjustment

This is one of the most common ways of balancing, which adjusts the level of challenge in the game for different players. This type of balancing can be either static or dynamic.

Figure 2.2: Difficulty selection screen present in *Dishonored* [5]

The static approach allows the player to explicitly select a difficulty setting (like "easy", "intermediate", or "hard"), typically at the game start. This approach is mostly used in single player games. In multiplayer games, static difficulty tends to take the form of handicaps. This can involve manipulating the capabilities of units, access to resources, amount of health, and starting positions, to give each player an equal chance of winning. The problems with this method are that players need to know their skill level before the game starts, which can lead to situations in which the game difficulty does not match the player's ability [37]. For example, players do not know how hard "hard" is until they've played. Also, once a player chooses a difficulty they are generally locked to this level for the entire game, or the player may miss out on special rewards or unlockables if they change it mid-game. In general, some studies have found that static difficulty levels aren't ideal for providing a good level of challenge to keep players interested in playing a game from beginning to end [37][38].

Dynamic Difficulty Adjustment (DDA) changes game difficulty based on player performance [37]. Systems that use DDA adapt and respond to the abilities of a player during a game session. DDA is used in games such as *Mario Kart* [56] (where players that are farther behind are given better powerups) or *Left 4 Dead* [66] (where the AI Director controls things such as the number of enemies and map layout depending on player performance) [14], or even *Unreal Tournament*'s [21] big head mode (where the more successful players are given a bigger head to make them easier to hit). A huge benefit to dynamic systems is that it reduces the need for iterative refinement based on play testing because the system regulates the difficulty itself [3][37].

While this kind of approach may seem ideal, care needs to be taken in competitive scenarios to ensure that experienced players do not feel cheated of a victory that should have been theirs [54]. Additionally, it has been shown that the assisted player may feel like their performance was entirely due to the system helping them and may feel discouraged and as if their achievements are meaningless [31][33]. This will also mean that the assisted player will miss out on the enjoyment that comes from finally beating something difficult [42]. Another potential problem with these systems is that when difficulty adjustments are too noticeable, some players can exploit the system better than others [3][37][75]. Therefore, the common consensus has been that these systems should be unobtrusive and not noticeable in order to be effective [3]. Examples of obvious balancing mechanisms that may reduce enjoyment are ones that use “rubber band” adjustment, such as the system in the *Mario Kart* series [56] that gives trailing players better items and leading players have a greater chance of being hit with an item that will handicap them. It is too noticeable and makes other players feel cheated [31][37].

2.1.2 Matchmaking

Matchmaking systems are popular in more competitive video games. They have elaborate ranking systems that aim to group players together who have the same level of experience or ranking to ensure the level of competition is balanced [26]. These work in a manner similar to ladders for sports like squash or tennis at sports clubs. For example, *StarCraft 2* [13] matches players into different tiers (Bronze, Silver, Diamond, Master, etc.) based on performance and selects opponents who are in the same tier [45] . *Halo* [16] uses TrueSkill, a Bayesian ranking algorithm [35]. Based on the Elo ranking system used in Chess, players are assigned a ranking which serves as a conservative estimate of their skill. This ranking is updated after each loss or win, and the amount that the ranking changes depends on how surprising that outcome was.

The problem with matchmaking is that the system may not always be able to match up players of equal skill, and must compromise based on the available players. It also does not take into account certain conditions that may temporarily change an individual's performance, such as having a "bad day". In addition, matchmaking does not help when two individuals know that they want to play together, but are unevenly matched.

2.1.3 Asymmetric Roles

In team games, player balancing can occur naturally if players can choose different roles that are better suited for their level of expertise. For example, *Team Fortress 2* [67] allows players who do not have good shooting skills to choose a class such as the Medic, with which they can be a benefit to their team without needing to shoot accurately. This is analogous to real world sports teams, where players may have very different roles (e.g., offense or defense). These games must be carefully designed to ensure that every class contributes to the team and no class is considered unnecessary. The downside to this kind of player balancing is the fact that players may not like feeling forced into a certain role, and game balancing issues may arise if not every role is filled. It also does nothing to improve the skills in which players are lacking, because they do not get to practice them.

2.1.4 Skill Assistance and Aiming Assistance

Skill assistance techniques can differentially assist the core mechanics that are needed to play a multiplayer game – such as targeting in shooting games or steering in driving games. For example, one study by Cechanowicz et al. investigated dynamic skill assistance in multiplayer racing games [17]. The system manipulated speed, acceleration, and steering to provide closer games by helping the less skilled players and hindering the better players. The authors found that player balancing worked well at balancing, and increased competence ratings for novices (but enjoyment did not change for the novices or experts). This paper also noted that assisting players in more than one skill area is more effective than only assisting in one single game mechanic.

One kind of skill assistance, *aim assistance*, facilitates the basic skills used in games that involve shooting, by making it easier to select on-screen targets. Aiming assistance helps weaker players to hit their targets more often when they shoot. Using aiming assistance to balance gameplay has only recently been investigated as a way of solving the balance problem in games that use targeting as the primary game mechanic. Bateman et al. showed aiming assistance to be an effective mechanism at improving competition in 2D games where players had uneven skill using techniques developed from traditional 2D targeting assistance work [10]. Dynamic Difficulty Adjustment was also incorporated into this system, as the farther behind a player became, the more assistance they would receive. Players reported having more fun when the balancing was active. In addition, the techniques were not obvious to the expert or the novice, and did not make people feel that the game was unfair [10]. These techniques generally give players a feeling of autonomy and competence, which as noted by Ryan et al., results in increased satisfaction [62].

While aiming assistance has been tested in 2D and found to work effectively, there is little information about whether or not aim assistance techniques would be effective in a 3D environment such as FPS games. Research has shown that even simple game elements can affect how well aiming assistance techniques perform [46], which casts doubt as to its potential as a player balancing solution. Additionally, “Expertise” in a 3D first-person shooter is made up of much more than just aiming ability – there are substantial differences between experts and novices in the ways that players move to evade enemy fire, use cover in the environment, learn the map layout, navigate their avatar through the virtual 3D space, and make use of resources such as ammunition and health packs. This additional complexity may result in 3D aiming assistance not working as effectively. As part of this thesis, we set out to answer these unanswered questions.

2.2 VIRTUAL POINTING AND AIMING ASSISTANCE

Aiming assistance is based on *target assistance*, which are techniques that have been devised to allow people to select on-screen targets as quickly and accurately as possible in 2D interfaces. Most of the previous work in using these techniques has been focused on helping older or people with disabilities use a computer to select items (like buttons, other targets) in 2D interfaces. Aiming assistance, in contrast, is targeting assistance in a 3D space – in which the action is "aiming" in a 3D environment instead of "targeting" a 2D icon. Both kinds of assistance techniques are based on a basic understanding of aimed movements as modeled by Fitts' Law.

2.2.1 Virtual Pointing

The Fitts' Law equation is based on work that was done by Paul Fitts to model one-dimensional pointing in the physical world [28]. The Fitts' Law equation (Equation(2.1) and Equation (2.2)) predicts the amount of time (*MT*) and the amount of difficulty (*ID*) it takes to point to a target. In this equation, *a* and *b* are empirically determined constants that are attributes of the mouse used. *W* is the smallest of the width or height of the target, and *D* is the distance from the current location of the cursor to the target. The Index of Difficulty (*ID*) is a number given to an object to give an object a score of pointing difficulty based on the size and distance. This equation states that the smaller and farther away an object is, the longer it will take to target and the harder it is to target. While Fitts' Law was originally meant to model physical pointing in one dimension, it has been shown to also model virtual pointing in both 2D [7] and 3D [46] using devices such as mice or game pads, and has formed the backbone of all pointing work done in the field of Human Computer Interaction.

$$MT = a + b\,(ID) \quad \text{.......} \quad (2.1)$$

$$ID = \log_2\left(\frac{D}{W} + 1\right) \quad \text{.......} \quad (2.2)$$

In physical pointing, there is a one-to-one mapping between the visual space and the motor space. Virtual pointing has three spaces: the motor space (physical movements of the mouse), the visual/display space (the space on the monitor), and the control-to-display space that links both (the intermediary device that converts how much physical movement moves the object in visual space, such as a mouse and driver) [7]. The *control-to-display (CD) ratio* is a number that describes this third space and represents how much physical movement corresponds with how far the cursor moves on screen. This means that pointing in virtual spaces is not limited by the same things that limit physical pointing, because it is possible to manipulate the control-to-display space to allow systems to make pointing easier [12].

The underlying movements in virtual pointing can be described with the *optimized initial impulse model*. This model says that pointing consists of two parts, a ballistic phase and a corrective phase [7][27]. In the ballistic phase, coarse initial movement is made towards the target. If the pointer manages to land on the target, then no more work is needed because the pointing task is done. If it ends up outside the target, a secondary phase of fine movement is needed to correct and reach the right location on the target.

Aim assistance techniques work in digital environments to facilitate pointing tasks using a device like a mouse or game pad by artificially manipulating distance and width, which in turn reduces the amount of time and the index of difficulty of the pointing task [7].

With these ideas in mind, three basic categories for targeting assistance have been developed to facilitate pointing in 2D:

1. Reduce D: These solutions try to reduce the amount of distance between the cursor and the target. This reduces the Index of Difficulty, which reduces the total movement time in the Fitts' Law equation. These solutions modify the first, ballistic phase in the optimized initial impulse model.

Reducing distance to the target can be carried out in several ways. Existing techniques warp the cursor towards the target based on the user's initial movement (e.g., Delphian Desktop [6]), or create temporary proxies of targets that are closer to the user's staring point (e.g., Drag-and-Pop [11]).

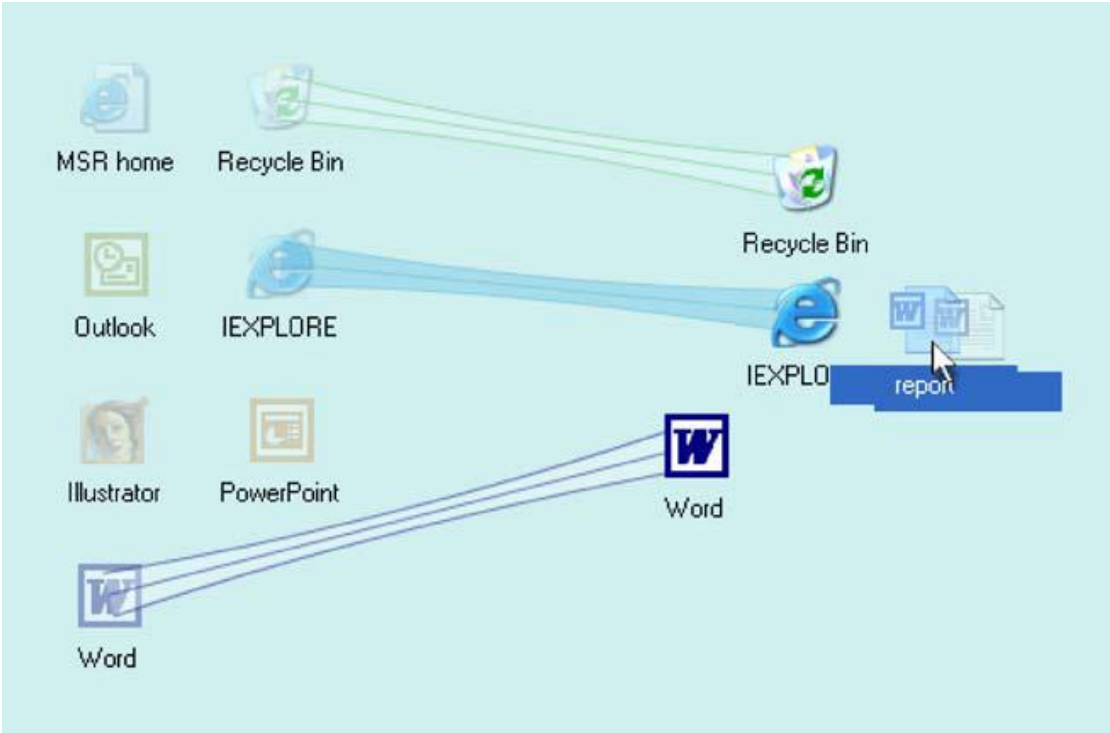


Figure 2.3: Drag and pop technique – "virtual proxies" of the icons are brought closer to the cursor

2. Increase W: These solutions try to increase the target size in either visual or motor space. These solutions target the second, corrective phase of targeting in the optimized initial impulse model. An example of this kind of solution is sticky targets, which reduces the control-to-display ratio of the cursor when it is over the target [47][74].

Another example is the fisheye method, implemented in systems such as Apple's dock. It is a visual-only expansion and involves dynamically expanding targets as the cursor nears a target. Visual-only expansions only increase the size of the object on the display. Motor expansions, however, increase the size of the object in motor space, meaning it takes more physical movement to cross the target. Research has shown that the visual-only expansions result in poor performance, and can even hinder the acquisition of targets because they lead to overshooting and confusion [7].

Figure 2.4: An example of Sticky Targets. The top image is the visual representation of the dialog, the bottom image is the dialog representation in motor space

3. Reduce D and Increase W: This category combines the first two approaches by reducing the distance to the target and also increasing the size of the target. An example is the Angle Mouse technique [73]. This system changes the control-to-display ratio depending on the current phase of pointing to simulate the effect of reducing D and increasing W. It lowers the control-to-display ratio when it detects the user is in the ballistic phase (assumed to be when straight linear movement occurs), reducing the distance because it takes less physical movement to move the cursor to the target. When small quick movement is detected, the system assumes the second corrective phase is active, so the control-to-display ratio is increased, making the object bigger because it takes more physical movement to move the cursor.

2.2.2 3D Virtual Pointing

While work in aim assistance has been focused on its use in traditional 2D GUI pointing, a recent study, by Looser et al., shows that the Fitts' Law equation holds in 3D virtual spaces as well

[46]. This suggests that the targeting assistance techniques that have been developed can also be applied to 3D games with good performance.

Despite their use in games for individual players, to our knowledge no commercial game has used aiming assistance for multiplayer game balancing.

2.2.3 Aim Assistance in Commercial Games

Commercial video games use aiming assistance mainly on console systems because of the lack of precision of the gamepads that are used as input [53]. Gamepads, on which thumbsticks are used for aiming, are not as precise as using mice [53]; to address aiming issues, several ideas from traditional target assistance have been adapted on console games that use game controllers as input. Our review identified three common approaches for targeting assistance in console video games: sticky targets, target lock on, and target gravity.

Sticky targets adjusts the CD ratio to slow movement when the cursor passes over the target. This technique gives targets a sticky effect when the cursor is on them, increasing the size of the object in motor space. This technique is already in use in FPS games such as *Call of Duty* [39] and *Halo* [16]. This assists during the fine correction phase of pointing. A problem that comes up in these types of assistance is overcorrection. A player may try to compensate for the slowdown and once the cursor leaves the slowdown area the cursor would swing rapidly and overshoot [7].

Target lock on isn't commonly used in FPS games but is popular in several other genres. Games like *Grand Theft Auto* [59], *Red Dead Redemption* [60], and the 3D games in the *Legend of Zelda* [55] series have used these techniques to improve gameplay. A problem with these sorts of techniques is that players may lock onto targets they didn't mean to target.

Target gravity is used in *Call of Duty* [39] in addition to sticky targets. Target gravity is like a passive version of target lock, which nudges the crosshair towards opponents [10]. The same problems that target lock on suffer from plague gravity as well. A player's crosshair may be moved to a target they were not trying to target.

There is little research into the efficacy of these techniques in these 3D game environments or how they perform when compared to each other. Despite the fact that targeting assistance is used in some games, aiming assistance techniques as a player balancing mechanic is not a central component of commercial games.

2.3 CHALLENGE, PLAYER ENJOYMENT, AND PLAYER MOTIVATION

The reason why people play video games and what makes people enjoy them has been subject to much research in the field of psychology. In this section, we cover current theories on player enjoyment and motivation, and examine the role challenge and suspense have in player engagement, in order to determine why player balancing techniques are so important.

2.3.1 Player enjoyment and motivation

A lot of work has been done to understand what motivates people to engage in a task or activity. There are two main types of motivation, intrinsic and extrinsic motivation [61]. Intrinsic motivation is described as doing an activity for personal satisfaction rather than due to some external consequence [61]. People do an action for the enjoyment of the action itself. Extrinsic motivation, on the other hand, is the kind of motivation that drives people to complete an action due to an external prod, pressure, or reward [61]. Intrinsic motivators are considered the stronger form of motivation, because they are more long lasting and self-sustaining. Furthermore, introducing external rewards and motivators can lower a person's intrinsic motivation. The general consensus is that the underlying motivation behind why people play games is intrinsic. People play games because they are intrinsically satisfying [62].

Researchers have started to deconstruct what it is about games that make them intrinsically motivating to play. One of the most popular theories is the Player Experience of Need Satisfaction (PENS) by Ryan et al [57] which is based on Self-Determination Theory (SDT) [62]. This theory states that games satisfy basic psychological needs and this in turn motivates sustained engagement from players. In order for something to be enjoyable or fun, it is necessary for that activity to fulfill a basic psychological need [57]. This theory has been shown to be an accurate measure for predicting if players would continue playing a game, and was shown to be an accurate motivation predictor regardless of genre or player type [58].

The basic psychological needs according to Self-Determination Theory are Competence, Autonomy, and Relatedness. Competence is defined as a "sense of efficacy" [57], or the need to feel effective in what we are doing [58]. Feeling competent or effective at something motivates further action, while feeling ineffective decreases motivation. Autonomy is defined as the amount of volition and personal agency a person experiences in their decisions and action [57]. Games try to fulfill autonomy needs by providing flexibility in goals, choice over strategies, and opportunities. Choices that are forced on players or gameplay that takes control away from the player, such as cut scenes or invisible walls that limit movement, reduce feelings of autonomy and reduce motivation. Relatedness is the need of a person to be connected with other people [57]. Games try to fulfill this need by providing opportunities for engaging in online interactions and communities. For example, FPS games allow players to form clans, and Massively Multiplayer Online games allow for players to experience adventure together in a party or to form guilds. The more a game fulfills these needs, the more enjoyable that game is rated [57].

The Competence construct is directly tied into the challenge and difficulty of the game. If the difficulty is too hard, it will lead to the player feeling less competent, and therefore they will be frustrated and less motivated to play the game. On the other hand, if a game is too easy, players will be underwhelmed and their competence needs will not be fulfilled. Ryan et al. argue that competence is among the most important construct provided by games [62]. Because competence is an important source of motivation, it is therefore important to have the level of difficulty be at an optimal setting in order to ensure players are motivated, engaged, and continue playing. Our research is concerned with trying to raise competence ratings in FPS games by giving players of all skill levels an optimal level of challenge.

The PENS model involves the Competence, Autonomy, and Relatedness and adds one more construct that is not in SDT, Mastery of Controls. Mastery of Controls is defined as "the ability to effortlessly perform intended actions in the game" [57]. The authors theorize that players need to be able to use the controls of a game in order to have an effective experience. This means that the easier a game is to control, the more effectively players will be able to fulfill their psychological needs. Normally, inexperienced FPS players don't have a great mastery of the controls, so it is our hope that aiming assistance will give them a greater mastery of controls, which will give them the ability to fulfill their psychological needs.

Other theories of player motivation have attempted to classify players into categories to understand what motivates them. Bartle proposed one such theory, in which players fall into one of four player types: Socializers, Explorers, Achievers, and Killers [9]. Socializers enjoy connecting and interacting with other players. Explorers are motivated to play a game for interacting with the virtual world rather than with people. Achievers are players who are motivated by points, competitive ladders, badges, and other measurements of success in a game. Killers are generally motivated by competition with other players. Killers enjoy fighting against other players for the sport and challenge, although some killers enjoy picking on weaker opponents.

Bartle hypothesizes that successful games must provide a way for each of the player types to meet their needs. Challenge is a major source of motivation in this system, as killers and achievers are generally motivated by competing and overcoming difficult challenges. Socializers and explorers are generally looking to use a game as a way to enjoy interacting with the environment or other players, and therefore aren't looking for difficult challenges as it gets in the way of their priorities.

Sherry et al. created a framework based on Uses and Gratification Theory to explain why people play video games [64]. The Uses and Gratifications Theory describes how media in general serve as solutions to everyday problems, such as boredom. This framework states that players use games for: competition (to experience of defeating others), challenge (to experience success following effort), diversion (to escape stress), fantasy (to experience novel or unrealistic situations), social interaction (to experience social experiences), and arousal (to experience positive emotions such as excitement).

There are several theories about why players are motivated to start playing a game, and what keeps them playing. We detailed several theories about what motivates players to play games and how to create an enjoyable game experience. All of them agree that challenge is an important factor in player enjoyment and retention. Therefore it is vital that players encounter an optimal level of challenge.

2.3.2 The role of challenge in enjoyment

Facing challenges and overcoming obstacles is one of the major draws of video games [24]. Game challenges can be classified into six groups: time challenges (in which players have a limited amount of time to complete a task), dexterity challenges (tasks require physical dexterity or quick thinking), endurance challenges (continuous streams of obstacles until the player fails), memory/knowledge challenges (require the knowledge of certain facts to be remembered), logic challenges (testing a player's intelligence) and resource control challenges (need to control usage of limited sources to achieve the goal) [26]. A major source of dissatisfaction with games is the inadequacy of games to accommodate to individual player skills and abilities, which give rise to player frustration [32].

The GameFlow model of player enjoyment directly investigates the role that challenge plays in the experience of players [65]. It is based on flow theory - flow describes a state in which people are engaged and interested in their tasks without being overly anxious, or experiences "so gratifying that people are willing to do it for its own sake, with little concern for what they will get out of it, even when it is difficult or dangerous". In order to experience the flow state, the challenges associated with the task must match a person's skill. In that regard, the GameFlow model states that games should: provide challenges that match a player's skill level, provide different levels of challenge for different players, provide challenge levels that increase as the player progresses through the game, and provide new challenges at an appropriate pace.

Bostan et al. include four more recommendations to the GameFlow model: challenges should be consistent with the storyline, dynamic adjustments should not restrain the sense of achievement, Dynamic Difficulty Adjustments should be believable, and difficulty curves should conform to the three skill acquisition phases (cognitive, associative and autonomous) [15].

2.3.3 The role of suspense in enjoyment

In many forms of media, outcome uncertainty and suspense is a major source of enjoyment. This is also the case for video games, in which the player experiences suspense from not knowing if they will succeed or fail [1]. Suspense is defined as thrilling and exciting situations that provide intense emotional arousal due to uncertainty about the outcome of an upcoming event [41][50]. Suspense is made up of positive feelings of hope and negative feelings of fear. In the context of games, studies of competitive games have suggested that player enjoyment increases the more suspenseful a game is (the closer the game scores are) [1][24]. One study by Klimmt et al. specifically investigated the role of suspense in an FPS game [41] and confirmed that higher suspense in the game means higher enjoyment.

This indicates that by creating matches that are closer in score, games become more suspenseful and more enjoyable. For experienced players, balancing will also allow them to get a greater challenge, and novice players will be able to fulfill their competency requirements.

CHAPTER 3

3-DIMENSIONAL AIMING ASSISTANCE TECHNIQUES DESIGN AND IMPLEMENTATION

Over the three studies conducted in this thesis, five different aiming assistance techniques were examined. Four of these techniques were based on targeting assistance research developed for use in two dimensions (2D), modified to work in three dimensions (3D). These techniques are *Target Lock*, *Bullet Magnetism*, *Area Cursor*, *Sticky Targets*, and *Target Gravity*. This chapter will describe the design decisions made to translate the classic 2D aiming assistance techniques to 3D, and describe the implementation of the five techniques.

In Chapters Five and Six, the amount of Aiming Assistance that is provided to players is determined by using a Dynamic Difficulty Adjustment (DDA) system [37]. This was implemented with a variable we call Levels: the score differential between two players determines a level between 0-10 that is used to determine when to apply assistance, and how strong the effect should be. For example if one player has 7 kills and the other player has 6, the player with 6 kills would be given an assistance level of 1 and the other would be given a level of 0. Larger score differentials (and thus higher levels) lead to increased assistance, with level 10 being the highest level of assistance possible. As the score difference decreases, the effect tails off until it disappears entirely when scores are equal. Level 0 corresponded to no assistance being given to a player.

The initial values used in the implementation of these techniques were determined through pilot studies in order to find values that would be suitable for improving performance for players of different skill levels. These values were refined through each study.

3.1 TARGET LOCK

The Target Lock method (Lock) moves the crosshairs of the player to the closest target's head when an activation key is pressed. In terms of Fitts' law, Lock reduces targeting time by reducing the distance between the pointer and the location of the target to essentially zero; however, our Lock implementation took a variable amount of time (depending on the score differential/Level) so locking onto a target was not instantaneous. Lock is activated when players pressed a button on the mouse. In our implementation, players could either use the 'q' key on the keyboard, or a function key available on the mouse. The key on mouse was used as our pilot studies indicated that some users had trouble using the 'q' key while navigating with the WASD movement scheme.

This technique manipulates the pitch and yaw of the player so that they face towards the closest opponent when lock is activated. When the Lock technique is first activated, the closest opponent is chosen and this target will only stop being the chosen target when it is no longer in view. Each frame updates the rotation towards the chosen target with the formula shown in Equation (3.1). In this equation, *CurrentYaw* is the yaw value that the player is currently facing. *YawToOpponent* is the desired yaw value for the player (the yaw value if they were facing the closest opponent). This equation shows only the calculations for the yaw component; however, the pitch was updated in the same way.

$$ChangeRotationBy = \frac{(CurrentYaw - YawToOpponent)}{LockSpeed} \quad (3.1)$$

We varied the $LockSpeed$ variable based on the current Level, so the time it took to move the crosshairs to a target's head would depend on their current performance. In Chapter Four, $LockSpeed$ = 13 - Level. Chapters Five and Six did not use Lock. At the lowest levels, it was possible for an opponent to outrun being locked on to. At the highest level, the crosshairs would be on the target virtually instantly. It took approximately 0.5s to lock on to a stationary target 90 degrees away at level 1 (roughly 7 degrees per game tick) and 0.15s at level 10 (roughly 12 degrees per game tick).

Target lock is not commonly used in FPS games but is popular in other genres. Games like *Grand Theft Auto* [59], *Red Dead Redemption* [60], and the 3D games in the *Legend of Zelda* [55] series have used locking. This method is based on the "object pointing" interaction technique, which tries to reduce empty space between targets by moving between selectable targets [34]. Lock may seem ideal for reducing targeting time but has drawbacks that may not make it viable in an FPS: distractor targets may compromise the performance of Lock; Lock requires explicit activation by the player; and Lock is obvious so players in competitive situations may not enjoy it [3].

3.2 BULLET MAGNETISM

The Bullet Magnetism (Bullet) technique "bends" the bullet towards the closest target if a target is within the activation range, essentially increasing the width of the targets making them easier to hit. Bullets in the UDK are instant shots and are therefore described by a vector. The vector is adjusted towards the first enemy that is within range of this vector when the player fires and this updated vector is used in the bullet collision logic. This effect can be seen in Figure 3.1, where the path of the bullet can be seen by the trail of white smoke. Bullet Magnetism is applied towards the body of the enemy if the crosshair is off the target and to the head of the enemy if the crosshair is already over a target. The higher the level of assistance, the farther away the effect begins and the more the vector is corrected. As the score difference decreases, the effect tails off until it disappears entirely when scores are equal. Additionally, the closer the player is to a target when shooting, the more the bullet is attracted to the target.

The values used in the calculations were refined over several pilots and the three studies in order to be effective in the multiplayer context. The Bullet Magnetism algorithm uses two variables, *HitDistance* and *MaxDistance*.

HitDistance is a variable that determines how close the original shot vector must be to the target in order for Bullet Magnetism to hit the target. The higher this number is, the farther away the original shot can be and still hit the target. For the study detailed in Chapter Four, this value was calculated to be (9*Level). In Chapter Five, we this value was calculated to be (12*Level). In Chapter Six, the final iteration, we found that when we calculated this value to be 12*(Level+7), the technique had enough power in multiplayer games. For any distance value that is less than or equal to *HitDistance*, Bullet Magnetism changes the shot vector so it directly points to the target's center (or head, if the original shot would already result in a hit).

If the distance between the original shot and the target was greater than *HitDistance*, but less than the value of the variable *MaxDistance*, our implementation still adjusted the flight path of the bullet. The shot wouldn't hit the target, but it would appear to have ended up closer than it normally would have been. If the distance between target and original shot was greater than *MaxDistance*, no Bullet Magnetism adjustment would be done. *MaxDistance* depends on the current Level to determine this value. In Chapter Four, this value was (100*Level)+60, in Chapter Five, this value was (100*Level)+100, and in Chapter Six it was (100*(Level+7))+100. When the distance is between *HitDistance* and *MaxDistance*, Bullet Magnetism finds the point between the target and the original shot vector and changes the shot vector so it points to that spot. Essentially, this is a cylindrical intersection test.

Both calculations return the result in UDK units, where each UDK unit is roughly 1cm in the game. The calculation for distance between target and original shot was done by finding the point on the original shot that was closest to the target and calculating the distance between these two points.

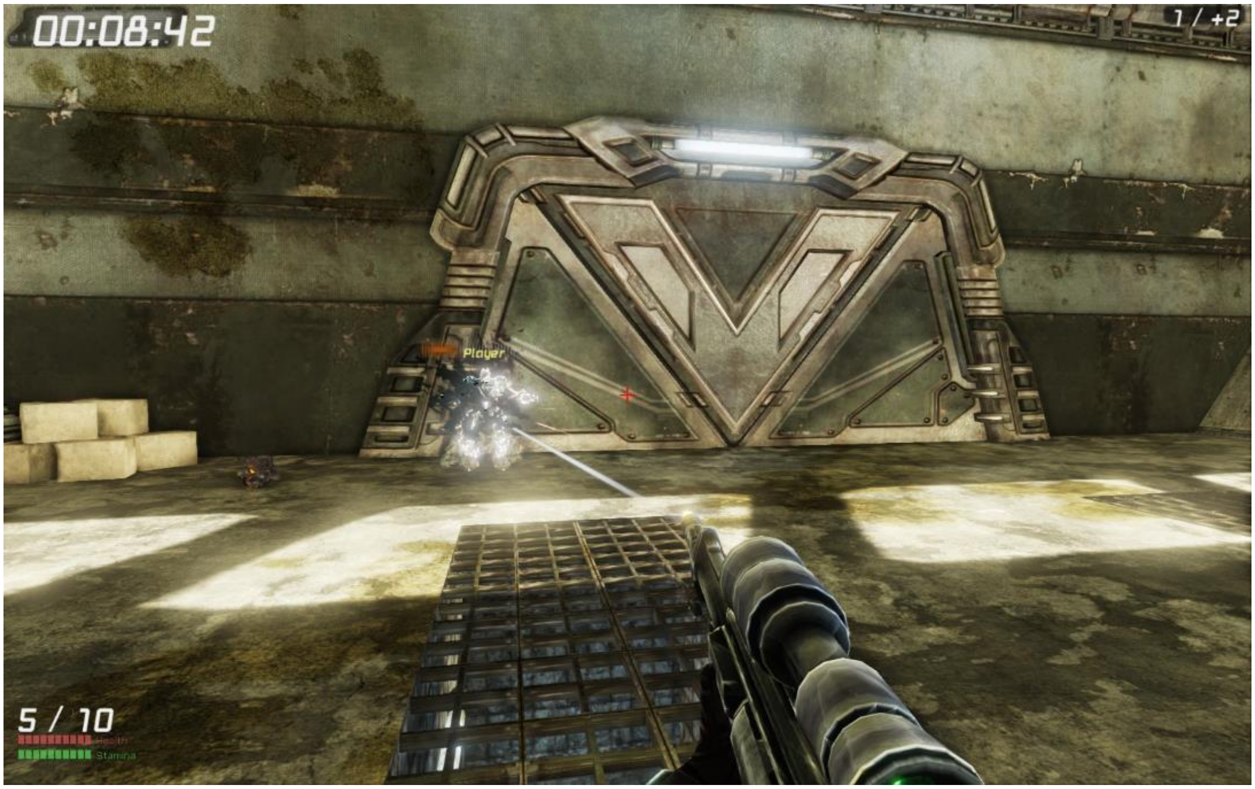


Figure 3.1: Bullet Magnetism adjusting the path of the bullet. Instead of going straight, the bullet bends towards the target

This end effect is similar to the area cursor method (described next) because it allows players to hit targets without perfect aiming, essentially increasing the target's width. However the visual feedback is different (fired shots bend instead of showing a bigger cursor). This method does not move the crosshair or change the control-to-display (CD) ratio so it may be less intrusive than other methods [69]. However, Bullet Magnetism may have issues if multiple targets are present, as the bullet may be attracted to the wrong opponent.

Bullet Magnetism is not based on any targeting assistance or Fitts' Law research. It is a technique that can be used in first person shooters because of the presence of bullets, which regular interface selections do not have. This technique is present in console games such as *Halo* and *Gears of War 3* [23] to make up for the reduced accuracy of controllers.

3.3 AREA CURSOR

The Area Cursor technique follows the original 2D implementation of an area cursor [27][40], but with modifications to work in a 3D environment. Normally when a shot is made in the implemented game, a zero extent trace is used to determine if a target has been hit by the bullet. The Area Cursor assistance technique uses a rectangle/non-zero extent trace to test intersection. This can be thought of as a huge bullet being fired. Intuitively it seems obvious that a bigger bullet will make it easier to hit a target, and is confirmed if this technique is thought of in terms of Fitts' Law. It has been shown that increasing the size of the activation area is the same as increasing the width of the target [40] meaning that bigger cursors lower the index of difficulty and movement time.

In our initial implementation, the size of the crosshair changes as the activation area changes, as seen in Figure 3.2. By the last study in Chapter Six, however, the size of the crosshair was not changed as it did not seem to provide additional value to do so. Normally the regular crosshair radius is 10px on the screen. Our calculation for the size of the crosshair/activation used in Chapter Four was 10px + (5px * Level). In Chapter Five, the size was 10px + (10px * Level). For the final iteration of Area Cursor used in Chapter 6, the size of the crosshair/activation area was 10px + (10px * (Level+6)) where $1 \leq Level \leq 10$. This reflects the size of the activation area/rectangle used for intersection. This means that less precision is needed as the assistance level is increased. In pilot studies where the crosshair would also grow, the growth of the crosshair was subtle enough that users tended not to notice it.

Figure 3.2: Area cursor at level 1 (a) and level 9 (b). Lines are thicker than in the actual game

This technique was chosen because of its performance in 3D environments [69] and because of the successful results at balancing achieved in 2D environments [10]. Area Cursor has also been shown to improve targeting performance for older adults [74] and users with motor impairments [27]. It also doesn't degrade performance in situations with moving targets like some of the other assistance techniques. However sloppy targeting behavior may appear with players who become accustomed to the extra activation area [18]. Distractor targets can also become a big issue if the cursor is too large [40]. The implementation handles the multiple targets issue by choosing the target closest to the center of the crosshairs. This method also does not directly help players get headshots, unlike Bullet Magnetism. The player will still need to center their crosshair over a target's head. Area Cursor also differs from Bullet Magnetism due to the shape of the intersection test being different (cylinder vs. square), different methods of calculating the effect sizes, and the difference in visual feedback.

3.4 STICKY TARGETS

The Sticky Targets method (Sticky) works by changing the Control-to-Display (CD) Ratio when the crosshairs are over a target. The lower CD Ratio results in a pseudohaptic effect of stickiness when over a target [18]. In our implementation in Chapter Four, we slowed down the cursor by dividing the amount of movement by (Level + 4), where $1 \leq \text{Level} \leq 10$. To determine if the crosshairs are over a target, a zero extent trace is done extending from the crosshair. If an opponent intersects with this trace, the stickiness effect kicks in. In terms of Fitts' law, this method increases the width of the target in motor space [12] by affecting the second phase of pointing where corrective actions are taken. Once the player gets to the target and the stickiness kicks in, they can position the cursor on the opponent more precisely due to the bigger target size in motor space.

Sticky Targets are used in console games like *Halo* [16], *Call of Duty: Modern Warfare 2* [39], and *Red Dead Redemption* [60] (called reticule magnetism). Stickiness is generally applied on console games to compensate for the lower precision in targeting, as compared to targeting with a mouse. We include Sticky because previous studies have shown that low levels of stickiness are difficult to perceive [47]. Informal studies have also suggested that subjects are more accurate in a 3D FPS environment with CD adaptation than without [19]. However, distractors could be an issue with Sticky, and overcompensation for the stickiness effect may lead to overshooting [47]. Additionally, studies have suggested that this method might not be useful for moving targets [10]. Games like *Call of Duty* tend to move the cursor along with a target to mitigate the moving target problem; however, we did not implement this feature to better mirror the 2D version used in prior research.

3.5 TARGET GRAVITY

The target gravity method (Gravity) gives each target an attractive force that results in a player's crosshairs being dragged towards targets. In terms of Fitts' Law, Gravity reduces the distance to the target. There are several different ways of implementing target gravity; however, we transferred the 2D algorithms [10] directly into the 3D environment. The gravity is calculated by first identifying all opponents that are visible to the player. For these *n* targets, let $p_1, p_2,..p_n$ be the position of the targets, p_0 be the position of the player in the 3D space, and p_w be the warped position. *Level* represents the strength of the assist, where 1 ≤ Level ≤ 10. For each target, the target weight is calculated with Equation (3.2). Then the warped position in 3D space is calculated with Equation (3.3).

$$w_i = \frac{Level}{|p_0 - p_i|^2 + 1} \quad (3.2)$$

$$p_w = \frac{\sum_{i=1}^{n} w_i p_i}{\sum_{i=1}^{n} w_i} \quad (3.3)$$

Equation (3.3) gives a position in the 3D environment that is the weighted average of all attractive forces. The crosshair is then moved towards this spot, with the strength of the movement depending on the Level. A lower level of assistance moves the crosshairs more slowly than a higher level. In our implementation, we only activate gravity if the mouse is moved toward the warped position. If the mouse is not moved, or is moved away from the warped position, no effect is applied on the crosshairs. In addition, we implemented gravity to be attracted to the head of enemy targets to assist players with headshots.

Gravity was included because of its strong performance in previous studies of 2D games [10]. Commercial games that use this technique include the console *Call of Duty* series. Like previous techniques, Target Gravity also has problems with distractor targets; the technique may move the crosshairs to the wrong opponent. Gravity is subject to distractors, potentially moving the crosshairs when a player is aiming at another target.

CHAPTER 4

EVALUATING THE PERFORMANCE OF AIMING ASSISTANCE IN 3D

Aiming is a critical component of FPS games, and aiming speed and accuracy is often a main differentiator between experts and novices. Since our goal is to move the weaker player in a pairing closer to the performance of the stronger player, *aim assistance* could be used to achieve this. Aim assistance improves the accuracy and speed of target acquisition by manipulating factors such as the size of the target in motor space. Aim assistance techniques have been shown to improve targeting in 2D shooting games and led to closer matches between pairs of differently skilled players [10]. Aim assistance was also shown to increase enjoyment, and the effects were not highly perceivable [10]. Therefore, 3D versions of these aim-assistance techniques hold promise for assisting novices in FPS games.

Little information is available, however, about the use of these techniques in 3D FPS environments. Commercial games such as Halo appear to implement various forms of aim assist and previous work has shown that Fitts' law holds in 3D aiming tasks [46], so it is likely that applying aim assist techniques should help to improve performance in a 3D FPS. However, previous research has also shown that adding even simple game elements to a 2D aiming task can significantly change target acquisition time [29]. The complexity of a 3D FPS game environment (e.g., moving while aiming, moving targets, multiple targets, targets that shoot back, etc.) may interfere with the potential benefits of aim assist techniques, ultimately reducing their efficacy. In order to determine if aiming assistance alone could be used for a player

---

Portions of this chapter have appeared in "The effectiveness (or lack thereof) of aim-assist techniques in First Person Shooter games" [69].

balancing context in FPS games, we first needed to determine if it could even increase an individual player's performance. We also needed to know if the techniques would be perceptible to players, because this was one of our requirements for player balancing.

To investigate whether aim assist techniques could be effective in 3D FPS games, we implemented five techniques (Target Lock, Bullet Magnetism, Area Cursor, Sticky Targets, and Target Gravity) in a custom game environment and carried out three performance studies. In the first study (S1), we set up a simple 3D shooting range that was similar to the 2D games used in past evaluations of aim assistance. The next two studies began to investigate how adding in real game elements affected the efficacy of aiming assistance. In the second study (S2), we created a realistic game level where players moved through a map with a number of computer-controlled players (bots). To begin teasing out the effects of realistic game elements, we carried out a third study that systematically varied two specific factors from S2: the effect of distractor targets, by removing friendlies from the map (S3A), and the effect of having more precise weapons, by switching to a semi-automatic sniper rifle (S3B).

These sets of studies were the first to explore the complexities of aim assistance in 3D environments, and the first to provide an analysis of real-game factors that can affect assistance. The results of this study also identified two successful candidate techniques that show potential for use in player balancing for FPS games.

4.1 AIM ASSISTANCE TECHNIQUES

We chose to investigate the five aim assist techniques detailed in the previous chapter (Target Lock, Bullet Magnetism, Area Cursor, Sticky Targets, Target Gravity) in the three studies. These were chosen from their previous use in console FPS games and 2D targeting assistance work. Study 1 and Study 2 used all five techniques. Due to the fact that Target Lock was too noticeable (failing one of the requirements we were looking for), and is often associated with "auto aim hacks", we did not investigate Target Lock in Study 3.

As defined in the previous chapter, the strength of the aiming assistance techniques are determined by *levels*. The system has 10 levels, where 1 is the lowest and 10 is the highest amount of assistance possible. The amount of effect of each level of aim assistance was determined by pilot studies.

4.2 STUDY DESCRIPTIONS

Below, we cover the three studies that investigated aiming assistance effectiveness in a 3D FPS game.

This study used the *Mega Robot Shootout* FPS game that was custom built using the Unreal Development Kit (UDK) for this study. This game implemented the five aiming assistance techniques and was used to test their performance. The game was developed in the UnrealScript language, using Visual Studio 2010 with the Nfringe add-on by PixelMine. The three studies ran on a Windows 7, Intel Core 2 Quad machine and an Acer 24-inch, 1920x1080 LCD monitor with a 60 Hertz refresh rate. Each participant set a comfortable sensitivity level for the Logitech G5 gaming mouse during the training round, and could not change it once the training rounds were complete. Logging of in-game actions was done on a Microsoft Sql Server 2008 R2 database.

As in most FPS games, targets were "shot" by placing the center of the crosshairs on the target and left-clicking the mouse. The targets in Study 1 were stationary. In Study 2, the enemies moved in a random pattern when the player was in the enemy's line of sight. A random spot was chosen (within 300 UDK units from the current enemy position) and the enemy began to move to that location while shooting at the player. Once the spot was reached, a new random location was chosen; this was repeated until the enemy was dispatched. Basically, the AI used a simple wander algorithm. The movement was limited so the enemies could not wander out of their rooms or move to the location of another enemy.

In Study 2 and Study 3, the player movement was controlled with a standard WASD scheme. Aim assistance levels were determined through pilot tests. The walkthrough level used in Study

2 and Study 3 was split into a hospital-themed area and a warehouse area. Both areas were linear so that the player could not explore, affecting the time measure. In the hospital area, enemies were placed in rooms along with friendlies. The warehouse area was darker and the enemies and friendlies were encountered as the player moved up. The locations of the enemies can be seen on the map in (Figure 4.1). The shooting gallery level was a simple outdoor environment. The player was confined to a square and the targets appeared in front of the player in several waves.

Participants were told that they would be testing several implementations of aiming assistance and that some rounds would have assistance, whereas some would not. Participants were unique to each study; the behavioral research ethics board at the University of Saskatchewan approved all studies.

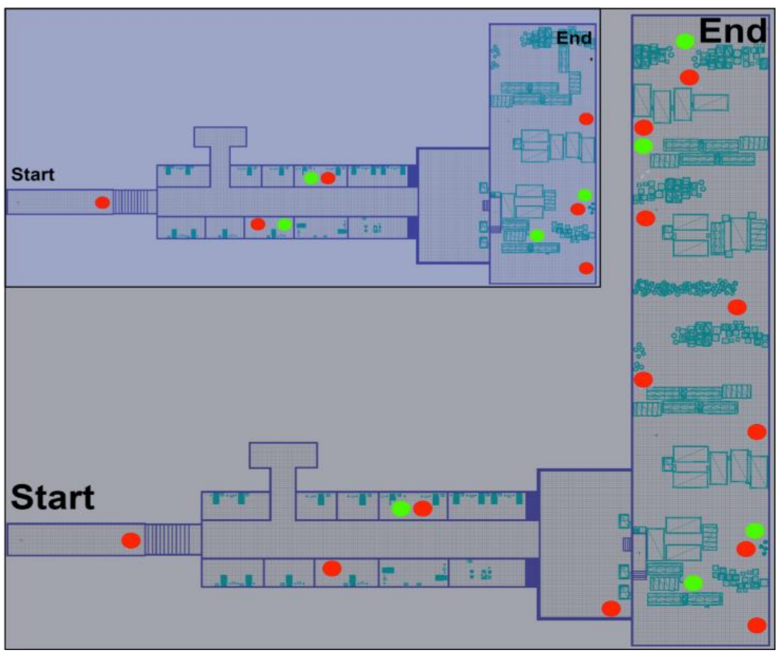


Figure 4.1: Walkthrough Maps: Grey Background–Full (S2); Inset Blue Background–Modified (S3). Red dots represent enemies; green dots represent friendly targets.

4.2.1 Study 1 – Shooting Gallery

Study 1 (S1) was the first step in determining how aiming assistance would perform in 3D. It extended previous work on modeling Fitts' Law in 3D FPS environments by Looser et al. [46].

This study tried to translate a simple 2D Fitts' task to 3D, in order to prove that Fitts' Law could hold in a 3D environment. It also builds on work that quantified targeting assistance in simple 2D shooting games, where players would be tasked with shooting 2D targets [10]. In order to be consistent with these previous works, we created a shooting gallery-style level (seen in Figure 4.2) where targets would appear and the participant would need to "shoot" them. User movements and enemy actions were constrained and the task focused on control rather than realism.

Figure 4.2: Player view of the shooting gallery

4.2.1.1 Task

Study 1 (S1) was a shooting-range-style level that was modeled after the level in the FPS Fitts' Law study by Looser et al. [46]. Participants used the sniper rifle (scope disabled), which took two body shots or one head shot to "kill" a target. The study consisted of one training round (no assistance), followed by the six technique rounds. In each round, there were seven waves of enemies. Each wave consisted of six targets; players tried to shoot as many as possible in 10 seconds. Waves 1 and 4 were close to the player; 2 and 5 were at a middle distance; 3 and 6 were farthest. In wave 7, enemies were placed in varying locations at the three distances, as shown in Figure 4.3.

In each of the six rounds, one type of aim assist was used: Bullet Magnetism, Area Cursor, Sticky Targets, Gravity, and Lock, all with the high level of assistance (Level 8), and no assist (Control); a Latin Square balanced the presentation order. The experiment took 20 minutes to complete.

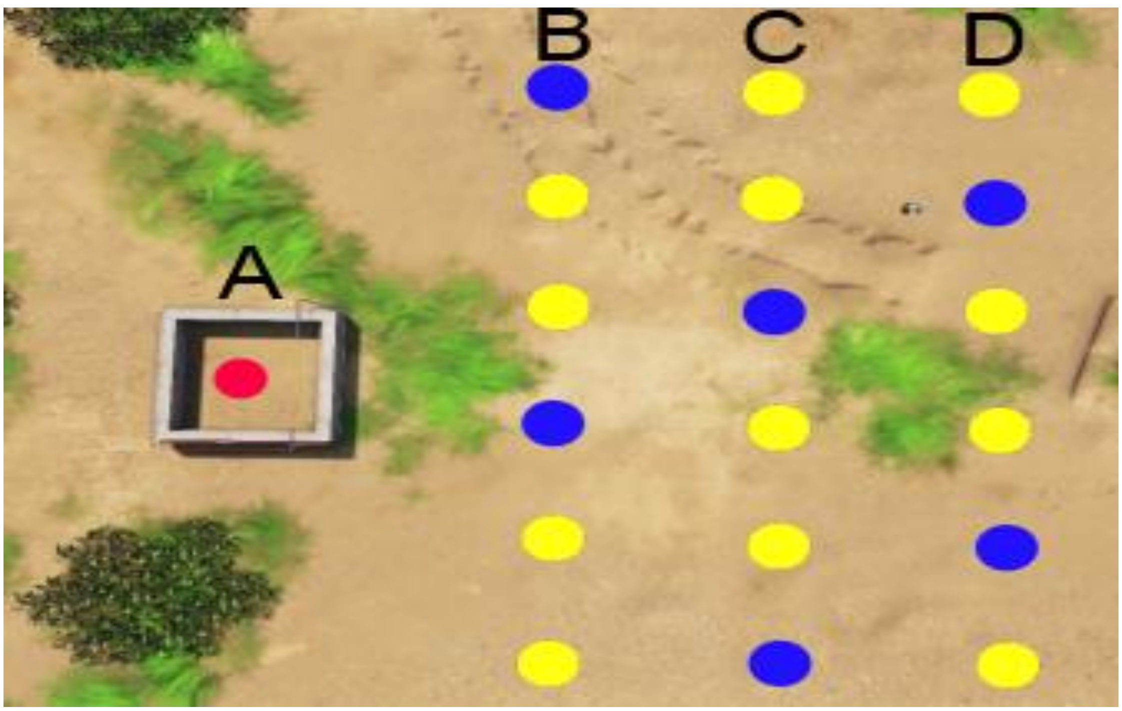


Figure 4.3: Player locations (red), enemy locations (B, C, and D), and location of wave 7 (blue)

4.2.1.2 Participants

We recruited 12 players (11 male) to participate in Study 1. Three participants identified themselves as novices at FPS games, five as intermediate players, and four as experts.

Participants ranged in age from 19-26 (mean 22.3). After providing informed consent, participants completed the experimental task. Participants were compensated with $5.

4.2.2 Study 2 – Full Walkthrough

Study 2 (S2) investigated the same 5 aim assist techniques in a realistic walkthrough task, complete with enemy targets who fired at the player and friendly targets that were to be avoided. Common features in FPS games (e.g., rifle scope) were implemented as we focused on game realism over control. The purpose of the study was to determine if the techniques that worked in the simple 3D environment would hold in a more complex, realistic "game-like" environment.

4.2.2.1 Task

S2 was a full game-like walkthrough level set in an abandoned warehouse. The level contained 12 computer-controlled opponents and 8 friendly turtles that acted as distractor targets. Distractor targets are often used in 2D targeting assistance tasks, because distractors are a major limitation for some aim assistance techniques. Therefore we decided to model distractor targets with our 8 friendly turtles.

Figure 4.4: a) The computer controlled opponents b) The friendly distractor turtles

Participants were equipped with an assault rifle and a pistol. The assault rifle had a higher rate of fire and did less damage than the sniper rifle used in S1. It took roughly 10 body shots or 5 head shots to kill a target with the assault rifle. The pistol had a slower rate of fire than the assault rifle, but did more damage per bullet (roughly five body shots or three head shots to kill a target).

The level was complete with visual effects (e.g., lighting effects, glass shattering) to look and feel like a realistic FPS game.

Participants played the custom level 14 times. The first two rounds were used as training to familiarize players with the controls and locations of enemies and friendlies. The remaining 12 rounds consisted of one round with each of the six aim methods at low (Level 2) and high (Level 8) assistance and an additional two control rounds (no assistance). At the end of each round, players completed a survey. Participants were instructed to avoid shooting the friendly turtles.

4.2.2.2 Participants

We recruited 16 players (10 male) to participate in S2. Eight participants identified themselves as FPS novices and eight as experts. Participants ranged in age from 18-37 (mean 25.1). The same procedure as for S1 was used, except that due to the longer experiment time (between 80 min and 2 hours), participants were compensated with $15.

4.2.3 Study 3 – Full Walkthrough 2

Study 3 systematically varied two of the factors present in Study 2 – presence of friendlies (S3A) and weapon type (S3B) – to quantify their relative effects on performance.

4.2.3.1 Task

Participants played two versions of the same map as S2 that was reduced in distance by half and contained only six enemies (see Figure 4.1). The first version (S3A) used an assault rifle and had no friendly distractor targets. The second version (S3B) used a sniper rifle (scope disabled) and four friendly turtles. S3A and S3B omitted the Lock method and the techniques also used only high (Level 8) assistance.

Participants went through the level 12 times. The first two rounds were training; the first was training for S3A and the second for S3B. The remaining ten rounds consisted of running through the control, bullet magnetism, area cursor, sticky targets, and gravity rounds in S3A and then repeating the process in S3B. The order of presentation of assistance types was balanced with a Latin Square. In addition, half of the participants started with S3A and half started with S3B.

4.2.3.2 Participants

We recruited 15 players (14 male) to participate in S3. Six participants identified themselves as FPS novices, five as intermediates, and four as experts. Participants ranged in age from 19-31 (mean 23.4). S3 used the same procedure as S2; participants were compensated with $10.

4.2.4 Dependent Measures

The same dependent measures were collected for each of the three studies, except there are no survey data for S1. We looked at several dependent measures across four categories.

Accuracy: *Hit Ratio* is the number of shots that hit a target out of all shots fired, representing targeting efficiency. *Headshot Ratio* presents the number of headshots over the total number of hits, representing targeting quality. *Subjective Accuracy* rating was provided by participants on a 21-pt scale (-10=least accurate to 10=most accurate, including 0=neutral).

Speed: *Time* measured the elapsed time to complete the task. *Subjective Speed* rating was provided by players on a 21-pt scale (-10=slowest to 10=fastest, including 0=neutral).

Outcome: *Deaths* is the number of times a player was killed by an enemy. In S1, there were no enemies, thus *Kills* (the number of enemies killed) is used to reflect outcome (in S2 and S3, the player had to kill all enemies to proceed, thus there was no variance in kills for those studies). *Subjective Performance* rating was provided by players on a 21-pt scale (-10=worst to 10=best, including 0=neutral).

Perception: The *Perceptibility* of the assist technique was rated by players on an 11-pt scale (0=lowest, 10=highest).

4.2.5 Data Analyses

For each of the four data sets, we conducted RM-ANOVAs with Assist Technique as a within-subjects factor on all dependent measures. In S1 and S2 there were six levels of Aim Assist Technique (Control, Lock, Bullet Magnetism, Sticky Target, and Gravity); in S3A and S3B, there were five levels (no Lock). In S2, S3A, and S3B, we also included Expertise (novice or expert) as a between-subjects factor). Finally, S2 had an additional within-subject factor: level of Technique Strength (low or high).

Data violating the sphericity assumption had the degrees of freedom adjusted using the Huynh-Feldt method. Type 1 error was prevented by using the Bonferroni adjustment on all pairwise comparisons, which divides α by the number of comparisons. α was set at 0.05 for all effects.

4.3 RESULTS

For all studies, statistical test results are shown in Table 4.1 and means and variances are shown in Figure 4.5, Figure 4.6, Figure 4.7, Figure 4.8, and Figure 4.9. To improve readability, we summarize only main results here.

4.3.1 Study 1 – Shooting Gallery

Accuracy. All of the techniques improved the Hit Ratio over Control, except Gravity. Lock and Bullet also improved Headshot Ratio.

Outcome. Lock, Bullet, and Area improved Kills over Control (and Gravity).

4.3.1.1 Summary of Study 1 Results

Gravity did not improve over Control on any measure. Sticky improved on Hit Ratio; Area improved on both Hit Ratio and Kills. Lock and Bullet improved on Hit Ratio, Headshot Ratio, and Kills. There were differences between the techniques as well, with Bullet and Lock beating the other techniques for Headshot Ratio and Kills.

Table 4.1: RM-ANOVA main effects for aim assist technique across all four data sets (columns) and measures (rows). Significant pairwise comparisons are shown (C=Control, B=Bullet Magnetism, A=Area Cursor, S=Sticky Targets, G=Gravity, L=Lock).

	Measure	S1 – Target Range	S2 – Realistic Level	S3A – No Friendlies	S3B – Sniper Rifle
Accuracy	Hit Ratio	$F_{5,55}$=23.3, p≈.000, η^2=.68 L, B, A, S > $C^‡$, $G^‡$	$F_{5,70}$=7.7, p≈.000, η^2=.35 L > C^*, G^* A > G^*	$F_{4,52}$=14.5, p≈.000, η^2=.53 B > $C^‡$, $S^†$, $G^†$ A > $C^‡$, S^*	$F_{4,52}$=9.4, p≈.000, η^2=.42 A > $C^‡$, $G^‡$
	Headshot Ratio	$F_{5,55}$=31.1, p≈.000, η^2=.74 L, B > $A^‡$, $C^‡$, $S^‡$, $G^‡$	$F_{5,70}$=53.5, p≈.000, η^2=.79 L, B > $A^‡$, $C^‡$, $S^‡$, $G^‡$ G > $A^†$	$F_{4,52}$=60.0, p≈.000, η^2=.82 B > $A^‡$, $C^‡$, $S^‡$, $G^‡$ G > $A^†$, C^*	$F_{4,52}$=64.6, p≈.000, η^2=.83 B > $A^‡$, $C^‡$, $S^‡$, $G^‡$ G > $S^†$
	Accuracy Rating	N/A	$F_{5,70}$=7.8, p≈.000, η^2=.36 L > C^*, S^*, G^* B > G^*	$F_{4,52}$=3.27, p=.019, η^2=.20 No pairwise differences	$F_{4,52}$=4.8, p=.002, η^2=.27 B > $S^†$
Speed	Time	$F_{5,55}$≈0.0, p≈1.00 (Controlled)	$F_{5,70}$=7.8, p≈.000, η^2=.36 L < C^*, G^*	$F_{4,52}$=9.48, p≈.000, η^2=.42 B < $C^†$, $A^†$, S^*, G^*	$F_{4,52}$=6.1, p≈.000, η^2=.32 B < C^* A < C^*, S^*, G^*
	Speed Rating	N/A	$F_{5,70}$=3.7, p=.007, η^2=.21 L > S^*	$F_{4,52}$=3.23, p=.019, η^2=.20 B > C^*	$F_{4,52}$=7.7, p≈.000, η^2=.37 B > A^*, $S^†$, $G^‡$
Performance	Deaths (Kills-S1)	$F_{5,55}$=84.7, p≈.000, η^2=.89 L, B > $A^‡$, $C^‡$, $S^‡$, $G^‡$ A > C^*, $G^‡$; C > G^*	$F_{5,70}$=7.9, p≈.000, η^2=.36 B < $C^†$, $S^†$, G^* L < C^*	$F_{4,52}$=10.3, p≈.000, η^2=.44 B < $C^†$, A^*, $S^‡$, $G^†$	$F_{4,52}$=8.7, p≈.000, η^2=.40 B < S^*, G^* A < $S^‡$, G^*
	Performance Rating	N/A	$F_{5,70}$=5.3, p=.003, η^2=.28 L > $C^†$, $S^†$	$F_{4,52}$=4.6, p=.003, η^2=.26 B > C^*, $S^†$	$F_{4,52}$=4.9, p=.002, η^2=.27 No pairwise differences
Perception	Perception Rating		$F_{5,70}$=8.1, p≈.000, η^2=.37 L > B^*, $A^†$, $C^‡$, $S^‡$	$F_{4,52}$=4.8, p=.002, η^2=.27 $A^†$, S^* > C	$F_{4,52}$=2.7, p=.041, η^2=.17 B > C^*

$^*p<.05$, $^†p<.01$, $^‡p<.001$

4.3.2 Study 2 – Full Walkthrough

In S2, there were main effects of Technique Strength (level); the higher level resulted in a higher Headshot Ratio ($F_{1,14}$=14.9, p=.002, η^2=.52) and greater Perception ($F_{1,14}$=6.8, p=.021, η^2=.33). Because level produced few systematic differences, and did not interact with the assistance type, we present results for the higher level of aim assist only in S2 to correspond to the high level that was used in S1 and S3 (also in Table 4.1 and Figure 4.5, Figure 4.6, Figure 4.7, Figure 4.8, and Figure 4.9).

4.3.2.1 Differences in Aim Assist Techniques

Accuracy. Lock and Area had higher Hit Ratios than Gravity. Lock and Bullet had higher Headshot Ratios than others.

Outcome. Bullet and Lock resulted in fewer Deaths than Control.

Perception. Lock was most noticeable.

There were Expertise effects: novices took more Time ($F_{1,14}$=28.2, p≈.000, η^2=.67), had more Deaths ($F_{1,14}$=10.3, p=.006, η^2=.42), and a lower Hit ($F_{1,14}$=6.3, p=.025, η^2=.31) and Headshot Ratio ($F_{1,14}$=31.1, p≈.000, η^2=.69); however, there were no differences in technique based on expertise.

4.3.2.2 Summary of Study 2 Results

In general, players noticed the Lock method, and perceived that it aided their speed, accuracy, and overall performance. In terms of objective differences, only Lock showed systematic improvements over Control in improved Time, Hit Ratio, and Headshot Accuracy, and fewer Deaths. Bullet magnetism also improved Headshot Ratio and Deaths over control, but there were no other improvements offered by the aim assist techniques over Control and differences between the assist techniques were minimal.

4.3.3 Study 3A – Assault Rifle, No Friendlies

There were no main effects of Expertise for any of the measures and no systematic differences between the aim assist techniques depending on expertise.

Accuracy. Bullet and Area had higher Hit Ratios than Control or Sticky (Bullet higher than Gravity). Bullet and Gravity improved Headshot Ratio over Control and Area (Bullet higher than Sticky, Gravity).

Speed. Bullet reduced Time over others.

Outcome. Bullet reduced Deaths over others.

Perception. Area and Sticky most perceptible.

4.3.3.1 Summary of S3A Results

When friendlies were removed, Bullet is the clear winner in terms of improved Hit Ratio, Headshot Ratio, Time, and Deaths. Interestingly, players did not notice Bullet, making it both effective and subtle. Other techniques also improved aspects: Area improved Hit Ratio but was perceived; Gravity improved Headshots and wasn't perceived; Sticky had no effect on performance, but was perceived.

4.3.4 Study S3B – Sniper Rifle

There were no main effects of Expertise for any of the measures; however, an interaction of Expertise and Aim Assist on Time ($F_{4,52}$=3.0, p=.027, η^2=.19) shows that experts took significantly less time than novices in the Bullet (p=.019) and Area (p=.001) conditions.

Accuracy. Area increased Hit Ratio over Control and Gravity. Bullet increased Headshot Ratio over all others.

Speed. Area reduced Time over Control, Sticky, and Gravity. Bullet reduced Time over Control.

Outcome. Bullet and Area reduced Deaths over Sticky and Gravity.

Perception. Area and Sticky were more perceptible.

4.3.4.1 Summary of S3B Results

When the sniper rifle was used, the techniques improved aiming differently. Area improved the Hit Ratio, whereas Bullet improved the headshot ratio. Both Bullet and Area reduced Deaths and the Time taken over the other approaches. In terms of noticeability, Area was perceived, whereas

Bullet was not. Gravity had an elevated Headshot Ratio but no other differences, and Sticky resulted in no improvements, but was perceived.

4.3.5 Comparison of Techniques across Studies

The techniques differentially improved performance over Control under the various experimental conditions. **Gravity** underperformed across the board, and was the only approach that did not improve hit ratio in S1. **Sticky** helped with Hit Ratio in S2 but did not improve performance on any measure in the walkthrough tasks. Sticky only helps with targeting when the avatar and the enemies are both standing still. **Area cursor** improved Hit Ratio in S1, S3A, and S3B, but not S2. Area also improved kills/deaths in S1 and S3B, but did not work as well when the assault rifle was primarily in play (S2, S3A). **Bullet magnetism** worked well on all measures in S1, S3A, and S3B, but only worked for Headshot Ratio and Deaths in S2. **Lock** was only used in S2, and improved over Control on all objective measures.

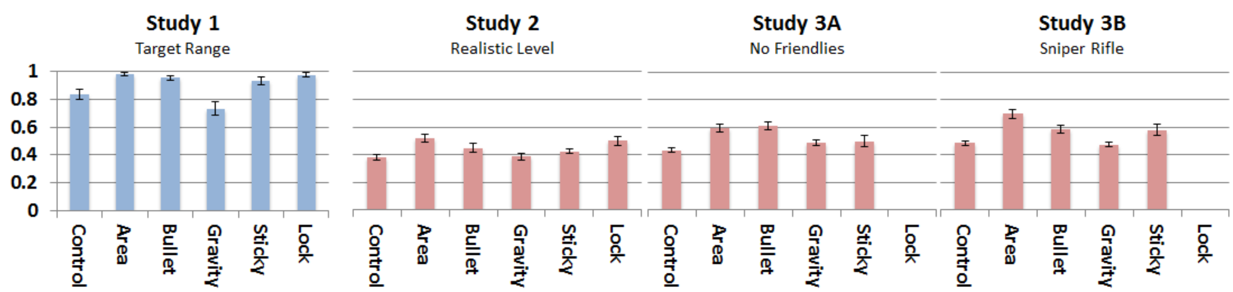


Figure 4.5: Hit Ratio means (± Standard Error) from Studies 1 - 3.

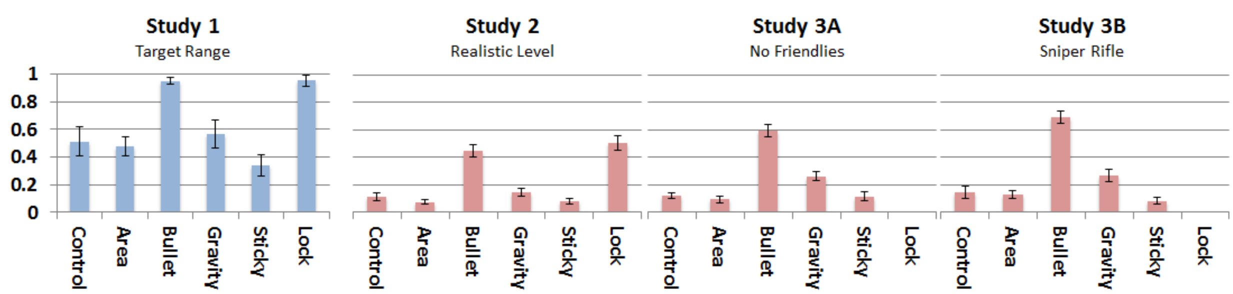


Figure 4.6: Headshot Ratio means (± Standard Error) from Studies 1 - 3.

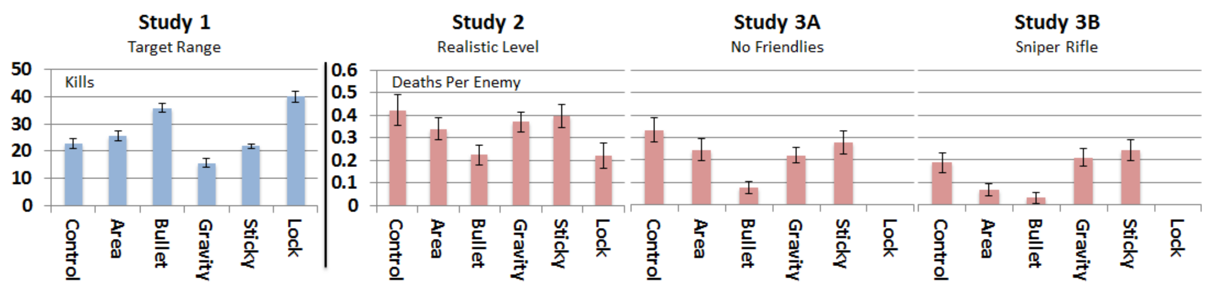


Figure 4.7: Kills/Deaths Per Enemy means (± Standard Error) from Studies 1 - 3. Vertical separators used where comparison between studies is not possible.

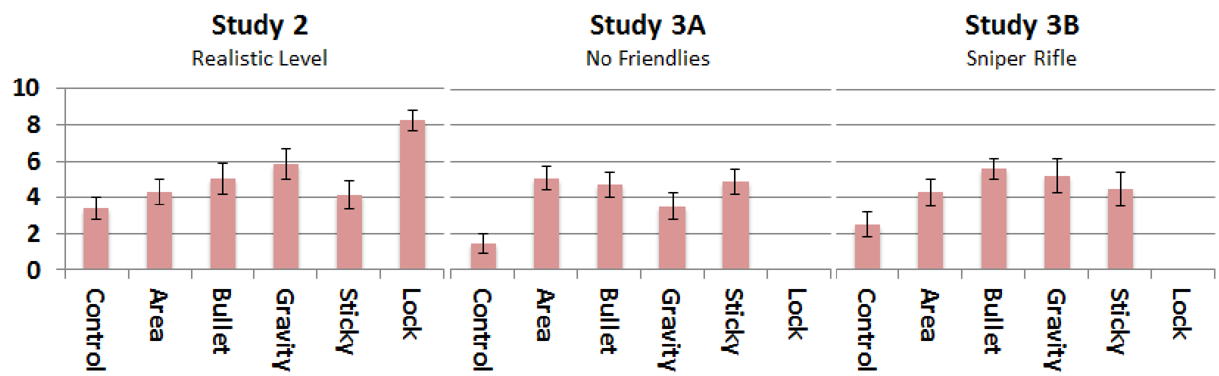


Figure 4.8: Perception means (± Standard Error) from Studies 2 - 3.

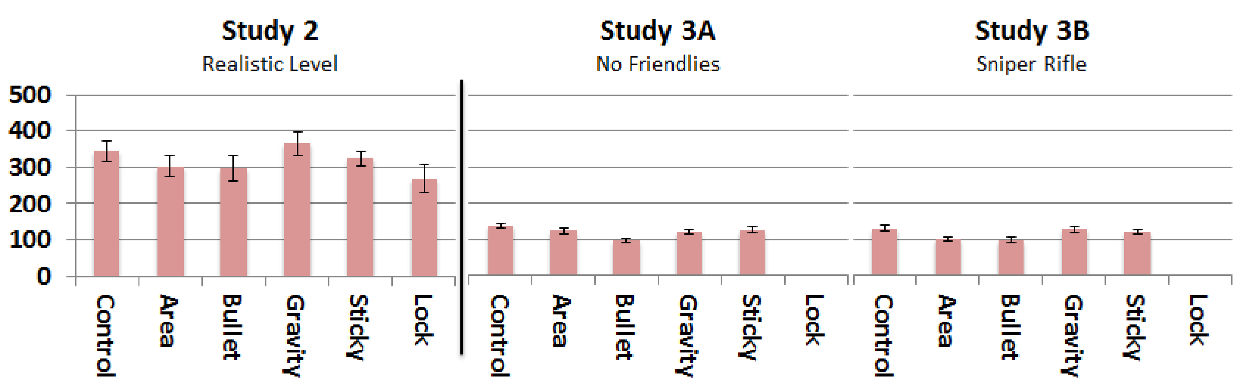


Figure 4.9: Time (in seconds) means (± Standard Error) from Studies 2 - 3. Vertical separators used where comparison between studies is not possible.

4.4 DISCUSSION

The main findings from our studies are:

- Realistic game elements such as moving and aggressive targets, distractor friendlies, multiple-shot weapons, and lighting effects dramatically altered which aim assist techniques worked well in the 3D FPS;
- In our least-constrained game environment (S2), only target lock consistently outperformed the control condition (but was highly perceptible); bullet magnetism worked well for deaths and headshots;
- In the single-issue manipulations of study 3, bullet magnetism and area cursor performed well; gravity and sticky targets performed poorly.

4.4.1 Why did bullet magnetism and area cursor work well?

The bullet magnetism technique attracts a fired bullet, by a certain amount, to a point on the nearest target's head (if shooting at them) or body (if not shooting at them). Using the head as an attraction point is a clear explanation for this technique's success on the headshot measure; even when targeting the body, bullet magnetism could still bend the bullet's trajectory upwards to the enemy's head. Bullet magnetism also worked well both with a higher-precision weapon and with fewer distractor targets. These results are likely due to the technique's limited manipulation of trajectory: if the player can get the targeting reticule close to the intended enemy then the assist will improve targeting performance without causing errors.

Area cursor also performed well for several measures; this technique essentially shoots a larger bullet, increasing the target size. This effect worked particularly well with the higher-precision rifle – because there is no alteration to the aiming action, higher precision appeared to allow players to make faster use of the larger targets. Area cursor did not do well on the headshots measure, likely because of our implementation – unlike Bullet, if a user targeted the body with Area, the shot would hit the body, even if the area cursor's rectangle intersected the enemy's head.

It is interesting that both bullet magnetism and area cursor alter the targeting process only *after* the player has carried out their aiming action. These techniques do not change the user's control over moving their view, and do not attempt to 'decide' which target the user is aiming for. It is possible that magnetism and area cursor worked well because they caused less conflict with the user's control actions.

4.4.2 Why did gravity and sticky targets perform poorly?

Gravity and Sticky performed badly on almost all measures in all studies. Observations and player comments suggest that these techniques had problems because they change the movement of the reticule *while* a player is aiming, and this may cause conflicts with the user's intentions.

The gravity technique could drag the player's crosshairs to the wrong target (e.g., a friendly, or a different enemy than the user intended), and this would result in the user fighting against the assist. The gravity algorithm is attracted to the nearest target, but this may sometimes be at odds with where a user is trying to aim. This behavior may be specific to FPS games where a player plans to hit targets in a certain order – in contrast, in the 2D shooting gallery game where gravity worked well, targets were equally desirable and it mattered little if the technique hit the wrong target; the user could simply target the desired target on the next shot.

Sticky targets encountered similar problems – player ability to control an aiming movement was hampered by the assist because it manipulates the 'user movement phase' of shooting rather than the 'bullet travel' phase.

4.4.3 How can the effectiveness of techniques be judged?

In previous work in aim assistance for 2D pointing, and in applying aim assistance in a 2D game [10], hit ratio (i.e., a reduced error rate) was a primary metric for assessing success. This was the approach we initially used in judging our techniques in S1. However, with S2, it became clear that Bullet and Lock were the most effective techniques – participants died fewer times while completing the level, even though the hit ratio of Bullet was worse than other techniques that performed poorly in terms of outcome. Furthermore, Bullet was effective at increasing the number of headshots; because headshots are so important in Unreal, scoring a few headshots has more impact than scoring many body shots. When considering assistance techniques, designers should not focus only on hit ratio, but also be aware of key game mechanics, such as the importance of where hits register. While we expected that hit ratio would be the most important metric, it turned out to be only part of the larger picture when evaluating a complex game.

4.4.4 Practical Significance and Limitations

Our study looks at only how players respond when first using aim assistance (i.e., the early effects) and does not address what might happen after long-term repeated use (i.e., the long term effects). However, we feel that these early effects are more important to understand because the goal is for the techniques to be applied in a multiplayer environment to help weaker players compete. Balancing play actually *requires* early effects – the techniques have to work well for people who have not used them before. In addition, the weaker player will in all likelihood have less experience, and thus is it critical that the techniques work in unfamiliar environments with imperfect handling of the controls; realistically, novice players cannot anticipate enemies and may not be familiar with a certain game map.

The results from this study shows that the performance of aiming assistance varies depending on the game elements present. For instance, Bullet Magnetism tends to perform better with fewer distractor targets, and Area Cursor and Sticky Targets tend to result in higher hit ratios with a slow-firing powerful weapon such as the sniper rifle than the assault rifle. Lock is the technique that is least affected by game elements, however it is very noticeable.

The long-term effects are also important, which even our 2-hour study cannot assess. However, we have some guidance because half of our participants were FPS experts (and thus very familiar with the type of control and environment); the other half were novices. There were no systematic differences in the efficacy of the techniques depending on expertise. This is important because regardless of whether a novice is playing against an expert or an expert is playing against a professional, certain techniques will be more effective. So although we cannot predict the effects of long-term usage of the techniques themselves, we can assume that long-term usage of the general controls and environment do not affect technique efficacy.

The application of our results is limited to aiming in 3D games – our work is the first to investigate aim assistance in this realistic scenario, and generalizing to all 3D aiming is premature. However, this is not a niche topic – these games have huge player numbers and represent a massive industry that historically has not been guided by HCI research.

There are also aspects of our work that may apply beyond 3D FPS. The trend that post-aim techniques work better might be interesting to designers of other realistic and time-constrained systems (e.g., safety-critical or medical), whereas the controlled nature of S1 means that our results are of interest to others who study 3D aiming (e.g., VR).

4.5 CONCLUSION

The purpose of this study was to identify whether aiming assistance could effectively be used to increase player performance in aiming and shooting tasks. We also wanted to determine if the assistance would be noticeable by players (as detailed in Chapter 2, previous work has indicated that assistance should not be highly noticeable). We found that most of the techniques do increase player performance and remain relatively imperceptible. We identified two techniques that best fit our requirements, Bullet Magnetism and Area Cursor. An interesting finding was how these techniques share the common trait of not modifying the pointing process. Our results also showed that game elements (such as weapon type and distractor targets) do affect how well these techniques perform, which could end up being a problem when we add in multiplayer elements.

CHAPTER 5

AN INITIAL ATTEMPT TO USE AIMING ASSISTANCE IN A REALISTIC GAME SETTING

In the previous chapter, we considered whether aiming assistance could be effective at increasing targeting performance in a 3D FPS environment. The results indicated that most techniques allowed users to perform significantly better. In this chapter, we will detail the study that was run to investigate how effective these techniques would be when used to balance "real world" gameplay in multiplayer FPS games. Once again we used the custom built UDK game, *Mega Robot Shootout*, to conduct this study involving pairs of players competing in a deathmatch game.

In this study, we used the two aiming assistance techniques (Area Cursor and Bullet Magnetism) that we found to be effective in our previous studies. The aiming assistance was given to the weaker player dynamically according to their performance (as defined by the score differential, detailed in Chapter 3). Our results indicated that while the performance of novices increased when they used aiming assistance, the actual game score was not affected by the inclusion of aiming assistance. Therefore, our results suggest that targeting assistance techniques alone are not effective in a First Person Shooter game to balance gameplay or increase game enjoyment (at least between pairs of novices and experts). Our results indicate that balancing performance in 3D FPS games is more complex than simply helping weaker players' aim, and we suggest possible reasons that warrant further investigation.

Portions of this chapter have appeared in "Balancing Multiplayer First-Person Shooter Games using Aiming Assistance" [70].

5.1 TECHNIQUES CHOSEN

The Bullet Magnetism and Area Cursor aim assistance techniques were chosen based on the previous studies about the effectiveness of different techniques in 3D environments, in which we found that Bullet Magnetism and Area Cursor tended to perform the best and had minimal noticeability. It is important to note that the previous study was only focused on measuring aiming assistance performance. This study aimed to apply aiming assistance for player balancing.

This study uses Dynamic Difficulty Adjustment to decide how much assistance to provide. Like in the previous chapter, the strength of the aiming assistance is determined by the *level*. The dynamic difficulty system works by setting the level for the weaker player (defined as the player with less kills) based on how far behind they are to the other player. The system has 10 levels, where 1 is the lowest and 10 is the highest amount of assistance possible. The level of assistance is calculated by subtracting the number of kills of the weaker player from the number kills of the player in the lead. For example, if the current leader has 5 kills and the other player has 1 kill, the level of assistance for the player with 1 kill will be 4. The player with 5 kills has the level of 0, or no assistance. The amount of effect of each level of aim assistance technique was increased from the previous study, as we noted that more game like elements would reduce efficacy. Multiplayer games add in a lot more of these potential problems (such as unpredictable opponent behavior) so we would need to increase the aiming assistance power to compensate. We changed the assistance techniques in the following ways:

5.1.1 Bullet Magnetism

Bullet Magnetism was adjusted to be stronger, to be (12*Level) for *HitDistance* and (100*Level)+100 for *MaxDistance* (refer to Section 3.2).

5.1.2 Area Cursor

The effect size of the updated Area Cursor was 10px + (5px * Level) where $1 \leq \text{Level} \leq 10$.

5.2 STUDY DESCRIPTION

Participants were matched with a single opponent with a different skill level in the First Person Shooter PC game *Mega Robot Shootout*, again played with a keyboard and mouse. The goal of the experiment was to test the Area Cursor and Bullet Magnetism targeting assistance techniques to see if they would result in closer scoring games and improved player experience than in games without assistance. Participants played one round with each of the two techniques with an additional round as the control (with no assistance).

5.2.1 Apparatus

Mega Robot Shootout, the game we developed with the UDK game engine, was once again used for this study to test the balancing capabilities of aiming assistance. It was developed in the UnrealScript language and modifications to the original game were made for this version. Visual Studio 2010 was used as the IDE with the Nfringe add on created by PixelMine. All participants played on machines with similar specifications to eliminate any external factors. Participants used Windows 7 machines with Intel Core i7-2600 machine using standard Dell mice, and Lenovo 22-inch LCD monitors with 60 Hertz refresh rates at the same resolution. Each participant was allowed to set custom mouse sensitivity due to the fact that people have different preferences and forcing some participants to play at a specified sensitivity might put them at an unfair disadvantage. Logging was done to a Microsoft SQL Server 2008 R2 database.

5.2.2 Participants and Procedure

Ten novice-expert pairs of participants were recruited from a University of Saskatchewan class and given course credit for their participation. Participants ranged in age from 18-29 (mean 21.5). Prior to the study, participants were asked to fill out a questionnaire about their gaming habits and FPS experience to sort them into either the “novice” or “expert” category. These experience ratings were verified by watching the participants during the training round. After providing informed consent, participants completed the experimental task.

5.2.3 Task

Novice participants were matched with an expert. At each session, the participant pair was told that they would be testing different game balancing techniques and that some of the rounds they played may have some sort of game balancing enabled. Then, the two players were instructed to join the server and play 4 five-minute rounds of a 1 versus 1 deathmatch game, where the first round served as training. The map used was a default map that came packaged with UDK called "deck". As this map was too big (players would spend most of the rounds wandering around trying to find each other in the pilot study), this map was customized so several floors and sections were inaccessible.

The participants started the game with an assault rifle gun and a pistol. There were two points on the level that spawned sniper rifles that the players could pick up. The assault rifle had a higher rate of fire than the sniper rifle but did less damage with each hit. Assault rifles needed 10 body hits or 5 head hits to kill a target and the sniper rifle needs one head shot or two body shots. The pistol was slower and more powerful than the assault rifle, and not as powerful but faster than the sniper rifle.

The study consisted of one training round with no assistance followed by rounds of Bullet Magnetism, Area Cursor, and no assistance. The ordering was balanced using Latin Square ordering, and each round lasted five minutes. At the end of each round subjective questions were presented to the participants with questions such as how fair they felt the game was and if they noticed anything helping them/anything strange going on, as well as how they thought they performed and how they thought their opponent performed. Each session lasted around 30 minutes when accounting for survey completion time.

The game continued to use standard controls, so the player could control their view with the mouse, and opponents were shot by moving the center of the crosshairs onto the target and left clicking the mouse. Player movement was controlled with a standard WASD control scheme. During the training round, a full explanation of the controls was given to both participants.

5.2.4 Dependent Measures

To see if the performance of the novices became better with aiming assistance, we looked at the following categories for both the novices and experts:

Accuracy: *Hit Ratio* is the number of shots that hit a target out of the total number of shots that a player took, representing targeting efficiency. *Headshots* are the number of headshots a player made, representing targeting quality.

Performance: *Kills* is the number of kills a player obtained. *Score Difference* is the average score difference between novices and experts, representing how close matches were.

Experience: The *Fairness* of a match is provided as a yes/no answer. Participants were also asked who the *Stronger Player* of the match was. *Competence*, *Autonomy*, and *Relatedness* subscales from the PENS scale of player experience [62] and *Interest-Enjoyment* from the IMI scale of player motivation [48] were also used to gauge user experience.

5.2.5 Data Analysis

For the data sets, we conducted RM-ANOVAs with Assist Technique (Control, Bullet Magnetism, Area Cursor) as a within-subjects factor and Expertise (Novice, Expert) as a between-subjects factor on the dependent measures of Kills, Deaths, Hit Ratio, Headshot Ratio, and the experience measures of Competence, Autonomy, Relatedness, and Interest. An RM-ANOVA was also conducted on the score difference with Assist Technique as a within-subjects factor.

Type 1 error was prevented by using the Bonferroni adjustment on all pairwise comparisons, which divides α by the number of comparisons. α was set at 0.05 for all effects.

5.3 RESULTS

Means and Standard Errors for all measures are shown in Figure 5.1, Figure 5.2, and Figure 5.3; our analysis of each dependent variable is given below.

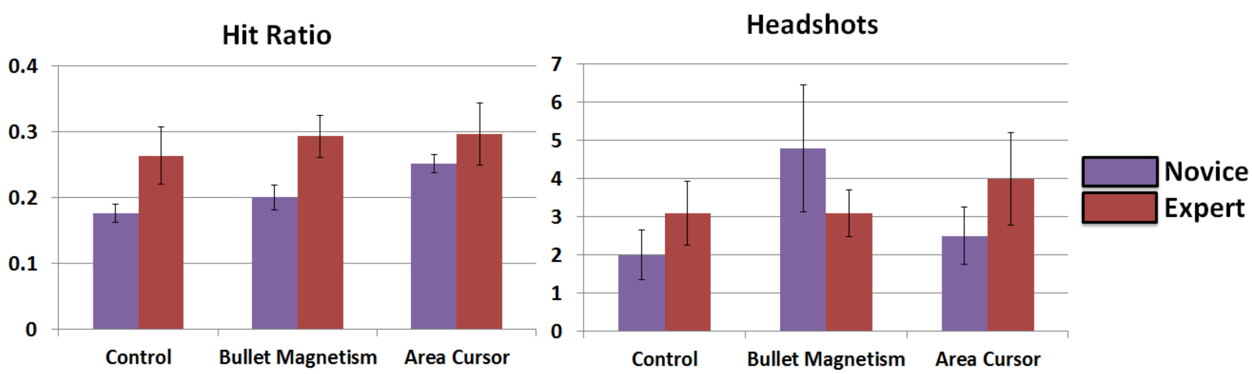


Figure 5.1: Means (+- Standard Error) for accuracy measures.

Accuracy: There was a significant effect of Assist Technique on Hit Ratio ($F_{2,36}$=4.2, p=.024, η^2=.19); pairwise comparisons showed that Area Cursor yielded a better hit ratio than Control (p=.018). Although Bullet Magnetism yielded a better hit ratio than Control, this failed to reach significance (p=.115); there was no difference between Area Cursor and Bullet Magnetism (p=.170). There was no effect of expertise on Hit Ratio ($F_{1,18}$=.961, p=.391) or any interaction between Expertise and Assist Technique ($F_{2,36}$=.3.3, p=.085).

There were no main effects of Assist Technique ($F_{2,36}$=.969, p=.389) or Expertise ($F_{1,18}$=.093, p=.764) on Headshot Ratio.

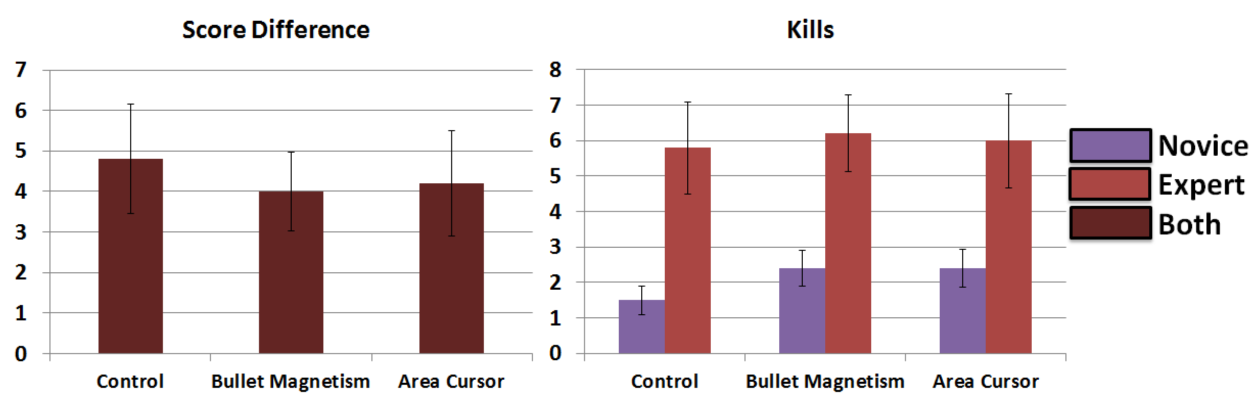


Figure 5.2: Means (+- Standard Error) for performance measures.

Performance: Experts had significantly more kills than novices ($F1,18=8.66$, $p=.009$, $\eta^2=.33$); however, there was no effect of Assist Technique on Kills ($F2,36=1.62$, $p=.212$) or interaction of Expertise and Assist Technique ($F2,36=.43$, $p=.654$). Correspondingly, novices had significantly more Deaths than experts ($F1,18=8.87$, $p=.008$, $\eta^2=.33$); however, there was no effect of Assist Technique on Deaths ($F2,36=2.3$, $p=.117$) or interaction of Expertise and Assist Technique ($F2,36=.43$, $p=.654$).

There was no effect of Assist Technique on Score Difference ($F2,36=.105$, $p=.901$, $\eta^2=.008$).

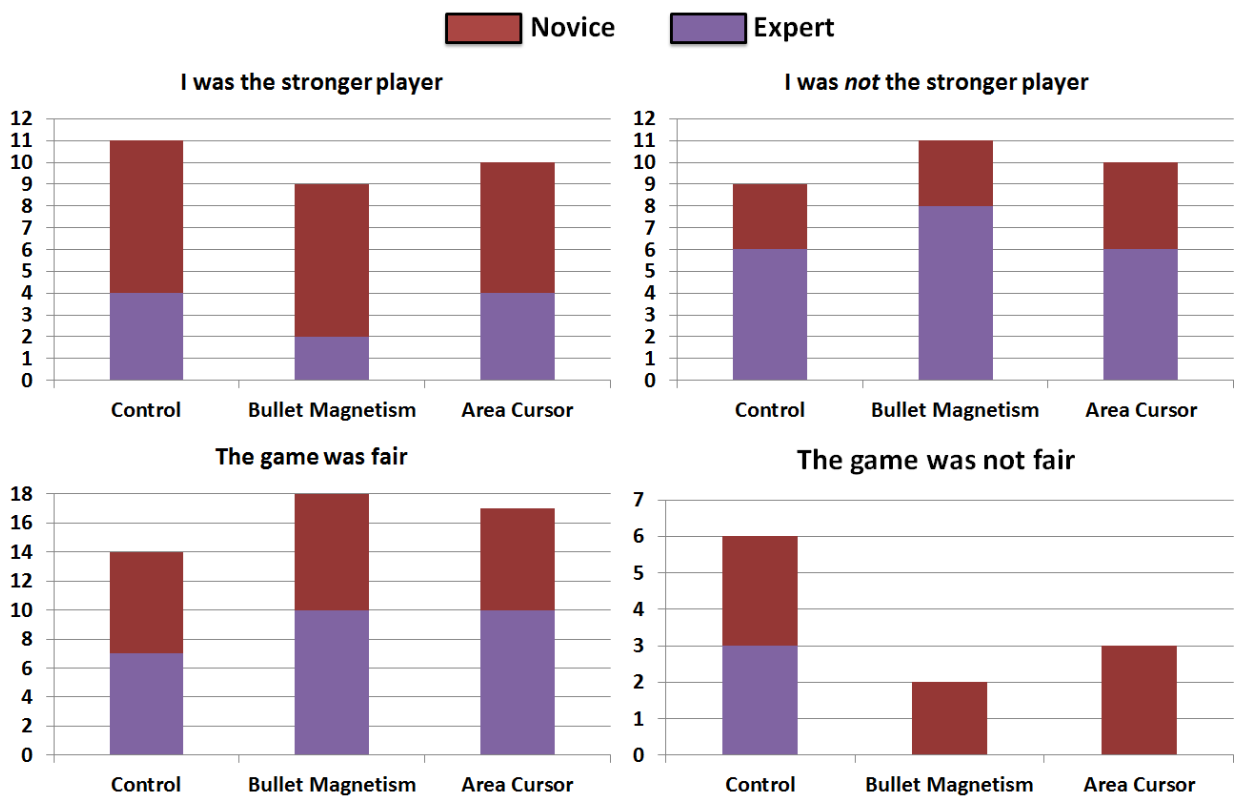


Figure 5.3: Frequencies of responses for subjective questions.

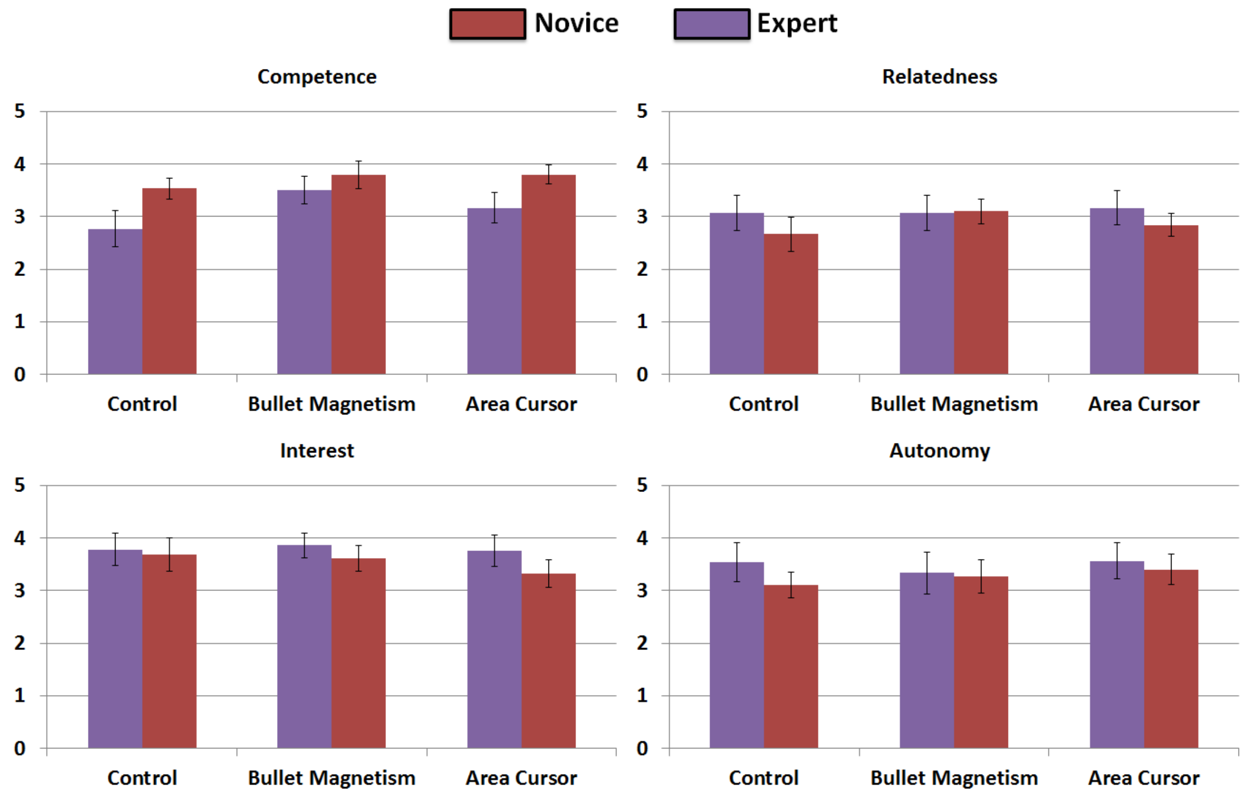


Figure 5.4: Means (+- Standard Error) for experience measures.

Experience: There was a main effect of Assist Technique on perceived Competence ($F_{2,36}$=8.66, p=.009, η^2=.33); pairwise comparisons show that participants felt more competent with Bullet Magnetism than Control (p=.018). Although Area Cursor was perceived as yielding more competence than Control, this result failed to reach significance (p=.070); there was no difference between Area Cursor and Bullet Magnetism (p=.390). There were no main effects of Assist Technique on Relatedness ($F_{2,36}$=.652, p=.527), Autonomy ($F_{2,36}$=1.2, p=.310), or Interest ($F_{2,36}$=.859, p=.432). There were also no effects of Expertise on Competence ($F_{1,18}$=2.9, p=.101), Relatedness ($F_{1,18}$=.35, p=.560), Autonomy ($F_{1,18}$=.22, p=.642), or Interest ($F_{1,18}$=50, p=.489).

10/10 (100%) novices stated that the game was fair when Area Cursor and Bullet Magnetism were used; 7/10 (70%) thought games without aiming assistance were fair. 7/10 (70%) experts considered the game to be fair when Area Cursor or no assistance was used; 8/10 (80%) thought that it was fair when Bullet Magnetism was used.

4/10 (40%) novices felt that they were the stronger player in games played with Area Cursor and no assistance and 2/10 thought that they were the stronger player when using Bullet Magnetism. 7/10 (70%) experts felt that they were the stronger player when Bullet Magnetism or no assistance was used; 6/10 (60%) experts felt they were the stronger player when Area Cursor was used.

Table 5.1: Frequency of responses for "Was this round fair?" (Yes responses shown) and "Who was stronger?" (Me responses shown).

	Was this round fair (Yes)			Who was stronger (Me)		
	Total	Novices	Experts	Total	Novices	Experts
No Assistance	14	7	7	11	4	7
Bullet Magnetism	18	10	8	9	2	7
Area Cursor	17	10	7	10	4	6

5.3.1 Summary of Results

Aiming assistance helped the novices to improve their accuracy (although only Area Cursor reached significance) to the extent that differences were not observed between novices and experts on the accuracy measures; however, this did not translate into significant improvements in performance (in terms of kills or deaths). Aiming assistance also improved how competent players felt, although only Bullet Magnetism reached significance; no other experience measures showed significant differences.

The fairness ratings indicate that the majority of players found the games to be fairer when aiming assistance was present, but novices and experts mostly agreed that the expert was the stronger player.

5.4 DISCUSSION

The main findings from our study are:

- aiming assistance techniques led to an increase in accuracy for novice players (e.g., Area Cursor had a significantly higher overall hit rate for novices);
- however, this increased accuracy did not lead to significantly closer scores in the game (score differences were slightly reduced with assistance, but not significantly so).

In the following sections, we discuss reasons why the assistance techniques did not work as well as expected, and outline how our results could be used to develop more effective player balancing techniques.

5.4.1 Why did Bullet Magnetism and Area Cursor not result in closer matches?

Although Bullet Magnetism and Area Cursor helped users improve their accuracy, it did not translate into an increase in score for the novice player. From our observations of the matches, we see four possible reasons for this result – relating both to the assistance techniques themselves, and also to broader game issues that limited the amount of assistance that the techniques could provide.

1. *Inexperienced Novices* – Several of the participants had little previous experience playing FPS games on PCs – that is, they had played only a handful of times casually, or had only played FPS games with a game controller on consoles. These participants tended to be very unfamiliar with the game, leading to in-game behaviors that could not be aided by the assistance techniques. For example, if players are unfamiliar with the controls and spend the majority of their time controlling movement rather than taking shots at the other player, no amount of aim assistance will help. This shows that a balancing technique that only affects one aspect of gameplay (i.e., shooting) can be unsuccessful if that aspect is overshadowed by other activities (e.g., player movement).

2. *Novices got lost* – Novices also seemed to spend much of their time wandering around the map, and ended up getting lost frequently. They seemed to have difficulty building a spatial map

of the level: during the sessions, several comments were made by the novices about being lost or about having difficulty figuring out where they needed to go. Experts, in contrast, seemed to build their spatial map immediately and then used that knowledge to efficiently move around the map. This result shows that there are several different elements in First-Person Shooters that differentiate experts and novices. Knowledge of the map, movement patterns, and other factors are an important part of being successful in FPS games, none of which are aided by aim assistance.

3. *Insufficient levels of assistance* – although the levels of assistance worked well during pilot testing and in previous studies [24], the levels in the dynamic system did not appear to be strong enough to balance real world situations. One of the problems is that the dynamic adjustment system took time to come into effect, and by the time it started to provide a substantial level of assistance, novices were often so far behind they couldn't catch up. This suggests that more aggressive adaptation is needed in dynamic balancing, at least for short games.

4. *Health packs* – the map had several health packs that allowed players to repair damage done by the other player. These packs respawned fairly quickly, so it was possible for a player to memorize their locations and use them to substantially extend their endurance in a firefight. This strategy was employed by most experts: after an intense firefight, experts would make sure to heal up to full health before reengaging the novice. Novices, in contrast, would not actively look for health packs after a firefight, and therefore would be at a disadvantage when they next encountered the (fully healed) expert. This result shows again that there is more to expertise in real-world FPS games than just shooting accuracy.

Overall, our results show that a player-balancing technique for a real-world First-Person Shooter game will need to take into account more of the game elements that differentiate novices and experts – including, for example, development of spatial knowledge, movement abilities, and strategies for using resources in the game world. A balancing technique can only assist performance when certain actions are occurring, and if these do not occur often enough, the technique may be prevented from being successful.

CHAPTER 6

A SECOND EVALUATION OF AIMING ASSISTANCE IN A REALISTIC GAME

In the previous chapter, we detailed our initial attempt to use several aim assistance techniques in a multiplayer environment for the purposes of player balancing. We found that it was very difficult to bring the performance of novice players up to the level of experts. We found that since there are several factors that determine expertise in an FPS game in addition to accurate aiming (e.g., avatar movement, learning and remembering maps, management of resources), and that these other factors made it difficult for a single mechanism to equalize performance. The fact that experts are so much better than novices in many different aspects of the game suggests that our current approach to balancing realistic FPS games are unlikely to work. However, our findings gave us several suggestions to solve the balance issue that warranted further study.

Using what we learned, we developed three general approaches for balancing FPS games that are better designed to overcome the problems encountered in the previous study. First, we refined the existing aim assistance techniques (bullet magnetism and area cursor) that showed promise in previous work by increasing the size of the techniques' effects even more. Second, we incorporated a new way of dynamically applying the aiming assistance that maintains the effect for a longer time period. Third, we developed a new assistance technique that manipulates two additional elements of FPS games in addition to aim assistance: a map view that shows the location of the expert to the novice, and different rates of damage for stronger and weaker players.

Portions of this chapter have appeared in "Now You Can Compete With Anyone: Balancing Players of Different Skill Levels in a First- Person Shooter Game" [71].

Our goal in developing these techniques was to handle a large range of skill differences, and be able to balance players in FPS scenarios that have all of the complexity of real games – but without making the techniques seem obvious and unfair to players. To test these new techniques, *Mega Robot Shooter* was used in a study in which novice, intermediate, and expert players played a series of one-on-one deathmatches. These new techniques were given to the weaker player dynamically according to their performance (as defined by the relative number of kills). Our test environment was similar in look, feel, and performance to a commercial FPS game, making the players feel like they were competing in an off-the-shelf game. We recorded performance (shots, hits, and kills), player experience (ratings of enjoyment, autonomy, competence, and relatedness), and perception of fairness (ratings of how much assistance had been provided to player, and direct ratings of how fair the game was).

Our results show that the balancing techniques worked extremely well. The study provides seven main findings:

- The techniques all showed significant improvements in minimizing score differential compared to the control condition with no assists provided.
- The techniques worked for pairings with very large skill differences (i.e., novice-expert pairings).
- There were significant differences between the balancing techniques in terms of efficacy – the standard aim assists were least effective, and the multi-factor technique (that combined aim assist with location indicators and differential damage) was the most effective.
- Despite the large degree of assistance sometimes provided by the techniques, perceptibility of the assistance was very low for players who were assisted, and stronger players only noticed the assist for the multi-factor technique.
- The balancing techniques actually improved the perception of fairness for the weaker player without detracting from the perceived fairness of the stronger player.

- The new balancing techniques improved the weaker player's perception of their own competence (in line with their improvements in accuracy), but did not affect the stronger player's feelings of their own competence.
- Finally, although the techniques affected only the weaker player's performance and perceived competence, experienced enjoyment increased for both the stronger and weaker players in when matches were balanced.

6.1 TECHNIQUES CHOSEN

We developed four skill assistance techniques for our realistic evaluation. The techniques vary in three ways:

1. The game mechanic that is manipulated (e.g., aiming)
2. The manipulation that changes the game mechanic (e.g. aiming can be changed using assistance algorithms that increase the effective width of the target)
3. The application method that controls the manipulation, which includes elements such as the game statistic that determines when balancing is needed, the start and end of balancing, and the strength of the manipulation.

Our four techniques cover a range of possibilities for these factors, including different game mechanics, different manipulations, and different application methods. As in our previous studies, all our techniques make use of a basic application method that we call *levels*: the score differential between two players determines a level between 1-10 that is used to determine when to apply assistance, and how strong the effect should be. Larger score differentials (and thus higher levels) lead to increased assistance to the weaker player. As the score difference decreases, the effect tails off until it disappears entirely when scores are equal.

6.1.1 Area Cursor with Levels

Area Cursor balancing (*Area*) is the same as the previous Area Cursor technique, where the 3D area cursor algorithm is the manipulation. With Area Cursors, players essentially fire a larger bullet (effectively making all targets larger). Normally, zero-extent traces are used to determine if the shot has resulted in collision with any enemies; area cursor instead uses a rectangular trace for collision detection.

In this version, the effect is much stronger than previous versions. The size of the rectangular activation area is 10px + (10px * (Level+7)). As the level increases, less precision is needed in targeting. If there are multiple targets within the activation area, the system chooses the enemy closest to the center of the crosshair. This version does not change the size of the targeting crosshair, but still changes the size of the bullet trace based on the level.

6.1.2 Bullet Magnetism with Levels

Bullet Magnetism (*Bullet*) uses the same bullet magnetism algorithm from the previous studies to adjust and improve aiming for weaker players. Bullet magnetism bends the path of a bullet towards any opponents that are within a certain angle of the initial shot, so that some shots hit that would normally have missed. Bullet magnetism essentially increases the width of the targets, making them easier to hit.

This implementation does the same adjustment of the shot vector towards the first enemy that is within a certain range from the normal bullet path, but is stronger than previous versions ((12*(Level+7)) for *HitDistance* and (100*(Level+7))+100 for *MaxDistance*) to compensate for the increased complexity in multiplayer. The higher the level, the farther away the effect begins, and the more correction is applied. The algorithm corrects the shot towards the body of the enemy; if the aim is already on the body, the bullet is corrected towards the head.

6.1.3 Bullet Magnetism with Levels and Delay

Delay balancing (*Delay*) uses the Bullet technique as the manipulation, but changes the application method from Levels to Levels plus Delay. Score differential is still used to determine the initiation and strength of the bullet-magnetism manipulation, but extends the time that balancing is active. Whenever the level of assistance becomes zero (indicating that the scores are now equal), the Delay method retains a level 1 assistance for 30 extra seconds. The Bullet Magnetism uses the same values as in the previous implementation. We developed this condition because during pilot tests we noticed that the novice could often get close to the expert's score, but could not get ahead because the aim assist had been removed. We hoped that this technique would allow for more variability of leads in the game.

6.1.4 Bullet Magnetism with Levels, Damage modification, and Location awareness assistance

This hybrid balancing technique (*Combo*) differs from the other techniques in that it adds two new game mechanics to the technique in addition to bullet magnetism-based aim assistance. First, Combo gives both players an indicator of the opponent's location, as seen in Figure 6.1 and Figure 6.2. This indicator can be seen behind walls and can indicate when the opponent is behind the player. Second, Combo modifies the damage done by a hit – both reducing incoming damage and increasing outgoing damage. For example, a player with level 10 assistance only needed two body shots to kill a player, and could withstand 50 shots before being killed. The Bullet Magnetism uses the same values as in the previous implementation. This technique was developed to investigate whether a combination of skill assistance techniques could be more effective than aim assistance alone.

Figure 6.1: Opponent location icon in Combo

Figure 6.2: Opponent location icon in Combo (behind obstacle)

6.2 STUDY DESCRIPTION

As in the previous study, this current study involves matching participants with an opponent in a multiplayer FPS game, where participants of differing expertise were matched up and told their goal was to score more kills than the opponent. The game was played on a PC using standard keyboard and mouse input. The goal was to see if the addition of our new player balancing techniques would increase enjoyment and experience of both parties. Participants played one round with each technique with an additional round as the control (with no assistance). Participants were given an extra round at the start with no assistance for training.

6.2.1 Apparatus

The participants once again used *Mega Robot Shootout* for this study, which was developed using the Unreal Development Kit (UDK). This UDK game was developed in the UnrealScript language, using Visual Studio 2010 and the NFringe add-on as the IDE. All sessions were played on 64-bit Windows 7 machines with comparable Intel processors and Nvidia graphics cards, Razor Imperator mice, and 22-inch LCD monitors with 60 Hertz refresh rates. Each participant was allowed to set custom mouse sensitivity. Logging was done to a Microsoft SQL Server 2008 R2 database.

6.2.2 Participants

We recruited fifteen pairs of participants, and they were compensated with $10. Participants ranged in age from 18-40 (mean 24.7). Participants were asked to fill out a questionnaire about their gaming habits and FPS experience to gauge their experience level and sort them into a "novice", "intermediate", or "expert" category. This was verified by watching their performance during the training round. The fifteen pairs consisted of five expert-novice, five expert-intermediate, and five intermediate-novice pairs.

6.2.3 Task

Each session involved one pair of participants (novice-expert, expert-intermediate, and intermediate-novice). Each pair was told that they would be testing different game balancing techniques and that some of the rounds they played may have game balancing enabled. The two players were then instructed to join the server and play a 1-on-1 deathmatch game. At the end of each round participants filled out a questionnaire for subjective measures. The map was the default UDK "deck" map, which was customized to be smaller to accommodate only two players.

Each session consisted of one training round with no assistance followed by rounds of Bullet Magnetism, Area Cursor, Delay, Combo, and no assistance. The ordering was balanced across participant pairs using a Latin Square, and each round lasted 5 minutes except for the training round which lasted 10. Each session lasted around 45 minutes when accounting for survey completion time. After each round, subjective questions were presented to the participants (detailed below).

Players could control their view and aim at opponents using the mouse. Shooting was controlled with the left mouse button and player movement was controlled with the standard WASD control scheme. During the training round, a full explanation of the controls was given to both players.

Both players had access to a minimap at the top right corner of the screen that showed them the map of the level, their location (in white), and the location of their opponent (in red). There was no way for the players to heal themselves.

The players started each round with an assault rifle and a pistol. There were two points that spawned sniper rifles the players could pick up. The assault rifle had a higher rate of fire than the sniper rifle but did less damage. The pistol was slower and more powerful than the assault rifle, and not as powerful, but faster than the sniper rifle.

6.2.4 Dependent Measures

To see if gameplay with aiming assistance is more balanced than without, we looked at several dependent measures across three categories.

Performance: *Hit Ratio* is the number of shots that hit a target out of the total number of shots. *Kills* is the number of times the player killed their opponent in the match. *Score Differential* is the difference in the number of kills between the two players. *Reversals* are the number of times in a match that there were changes in which player was in the lead. *Outcome* refers to which player won the match.

Perception: Participants were asked to rate the amount of *My Assistance*, and the amount of *Opponent Assistance*.

Experience: The *Fairness* of a match was recorded as a simple fair/not fair answer on a questionnaire. The *Competence*, *Autonomy*, and *Relatedness* subscales from the Player Experience of Needs Satisfaction (PENS) scale of player experience [62] and *Interest-Enjoyment* from the Intrinsic Motivation Inventory (IMI) scale of player motivation [48] were used to gauge user experience.

6.2.5 Data Analysis

We conducted a RM-MANOVA with Balancing Technique (Control, Bullet Magnetism, Area Cursor, Delay, Combo) as a within-subjects factor and Pairing (Novice-Expert, Novice-Intermediate, Intermediate-Expert) as a between-subjects factor on the dependent measures of Score Differential (with each dyad treated as a single case) and Reversals. We also conducted a RM-MANOVA with Balancing Technique as a within-subjects factor and Pairing and Expertise (whether the participant was the weaker or stronger player) as between-subjects factors on the dependent measures of Performance (Kills, Hit Ratio), Perception (My Assistance, Opponent's Assistance), and Experience (Competence, Autonomy, Relatedness, Enjoyment), treating each participant as a single case. Type 1 error was prevented by using the Holm-Bonferroni adjustment on all pairwise comparisons with α set to 0.05; because the Holm-Bonferroni method was used to correct for familywise Type 1 error, individual p values for pairwise comparisons are not presented, but are less than 0.05 after the correction is applied. Sphericity violations were corrected using the Huynh-Feldt method of adjusting the degrees of freedom.

We also conducted chi-squared tests on the true/false question of "In general, did you find this round to be fair?" and on Outcome for each Balancing Technique separately.

Our sample was of a reasonable size (N=30) to draw conclusions in a repeated-measures design; however, we report eta-squared values to give an indication of the amount of variance explained by each significant effect.

6.3 RESULTS

We present our results by answering questions about player performance, player perception, and player experience.

6.3.1 Performance

6.3.1.1 Were the games closer with balancing applied?

The RM-MANOVA for score difference showed a main effect of Pairing ($F_{2,12}$=18.4, p≈.000, η^2=.75), in which the novice-expert pairing had a greater Score Differential than the novice-intermediate (p≈.000) or intermediate-expert (p≈.000) pairings. The was also main effect of Technique ($F_{4,96}$=27.7, p≈.000, η^2=.70). Pairwise comparisons using the Holm-Bonferroni correction with α=.05 showed that there was a higher score differential in the Control and Area conditions than with the Bullet, Delay, or Combo techniques. Combo was also resulted in lower score differentials than Bullet. However, the significant interaction between Technique and Pairing ($F_{8,48}$=5.2, p≈.000, η^2=.46) revealed that the differences of Technique were significant for the novice-expert pairings, but not the intermediate/expert or novice-intermediate pairings as calculated with the Holm-Bonferroni adjustment (Figure 6.3).

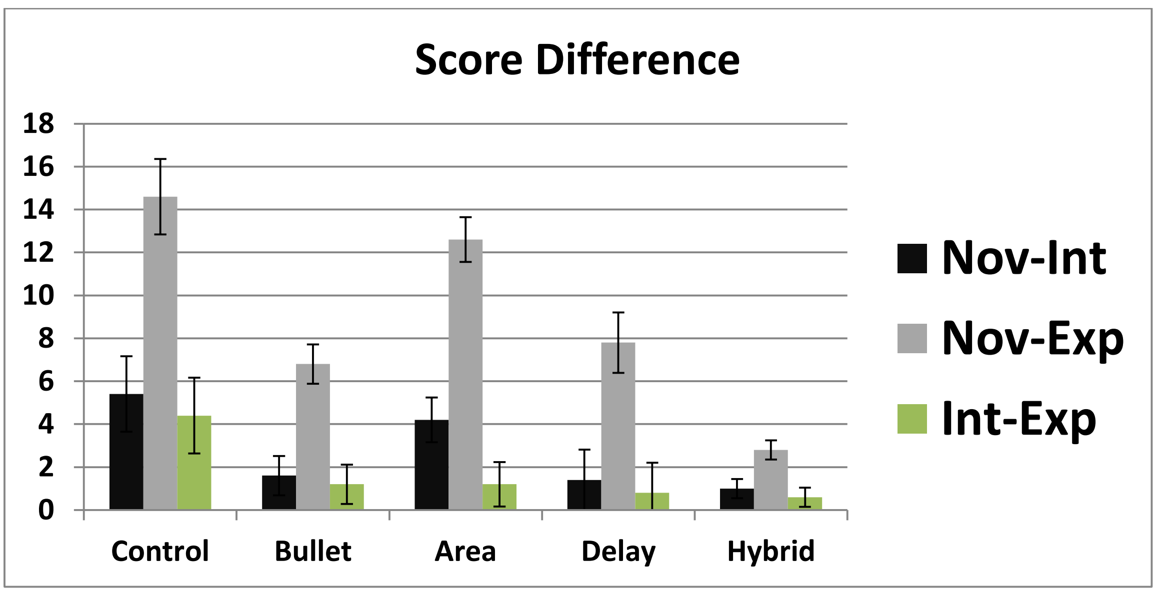


Figure 6.3: Means +/- Standard Error for Score Difference

6.3.1.2 Did game outcome change as a result of balancing?

Although the balancing techniques helped to balance games through score differential, there was not a significant effect on outcome in terms of who won the match (determined by the relative number of kills when the game ended). Chi-squared tests of who won the game showed significant differences for each condition, indicating that the stronger player won significantly more matches, regardless of balancing technique (all $p<.02$); see Table 6.1.

However, there was a significant effect of Balancing Technique on the number of lead reversals that occurred throughout the match, i.e., when one player took the lead ($F_{2.4,28.8}=4.63$, $p=.014$, $\eta^2=.28$). Pairwise comparisons showed that Combo provided the most lead reversals, but that Bullet and Delay also improved over Control. Although there was also a significant effect of pairing on the number of lead reversals ($F_{2,12}=5.4$, $p=.021$, $\eta^2=.48$) – in which the intermediate-expert group had more lead reversals than the novice-intermediate or novice-expert group – there was no interaction between Technique and Pairing.

Table 6.1: Count of who won the game in each condition and mean (SE) of lead reversals by balancing technique

	Control	**Area**	**Bullet**	**Delay**	**Combo**
Weaker	0	1	2	3	4
Stronger	15	13	13	12	10
Tie	0	1	0	0	0
X^2	n/a	19.2	8.1	5.4	8.4
p	**n/a**	**.000**	**.005**	**.020**	**.015**
Reversals	0.2 (0.2)	0.3 (0.2)	0.5 (0.3)	0.6 (0.2)	1.2 (0.3)

6.3.1.3 Was accuracy affected by the balancing technique?

To measure accuracy, we looked at the hit ratio and the number of kills. Hit ratio is an individual metric, whereas kills also depends on the evasive abilities of the other player. As expected, there were main effects of Expertise on both measures, with the stronger player having more kills ($F_{1,24}=24.1$, $p\approx.000$, $\eta^2=.50$) and a higher hit ratio ($F_{1,24}=9.2$, $p=.001$, $\eta^2=.38$) than the weaker players. There were also main effects of Balancing Technique on both kills ($F_{3.9,94.7}$=13.5, p≈.000, η^2=.36) and hit ratio ($F_{4,96}$=27.7, p≈.000, η^2=.70); however, the significant interactions between Balancing Technique and Expertise on both kills ($F_{4,96}$=22.8, p≈.000, η^2=.49) and hit ratio ($F_{4,96}$=11.9, p≈.000, η^2=.33) show that the differences exist mainly for the weaker player who was provided assistance. Specifically, as can be seen in Figure 6.4, weaker players had a higher hit ratio with Delay than all other conditions and a lower hit ratio with Control than all other conditions, whereas there were no differences for the stronger players between any conditions. In addition, weaker players had fewer kills with Control than with any other type and more kills with Combo than any other technique (also more with Delay than Area), whereas the stronger players killed less often with Bullet than with Control or Area.

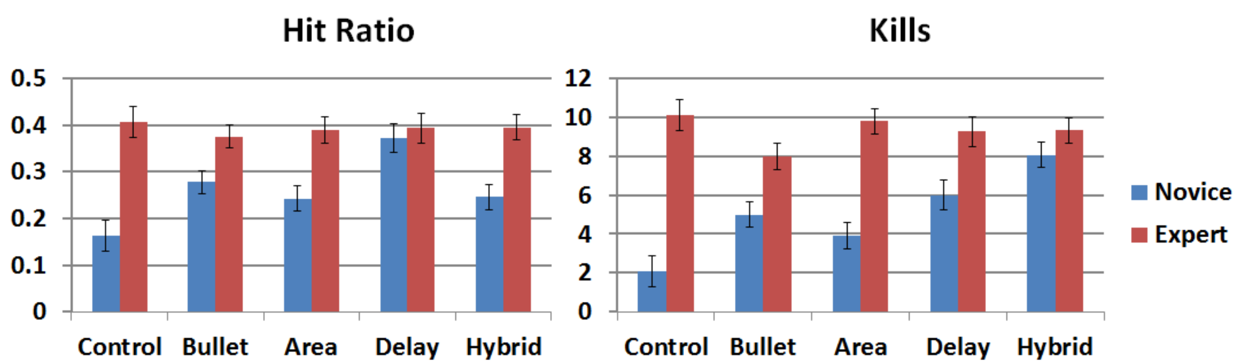


Figure 6.4: Means +/- Standard Error for Hit Ratio and Kills, split by expertise

6.3.2 Perception

6.3.2.1 Did players notice the balancing techniques?

We asked players to rate the level of assistance that they were provided with on an 11-point scale, where 0 is none and 10 is high. Figure 6.5 shows that perception was generally rated quite low (between 2 and 4 points on the 11-point scale). Stronger players perceived balancing less than weaker players ($F_{1,24}$=4.8, p=.038, η^2=.17), which makes sense given that they would have rarely received assistance. In addition, there were differences in the perceptibility of Balancing Techniques ($F_{4,96}$=6.7, p≈.000, η^2=.22), in which players perceived the Combo technique more than the other approaches, with no other differences between the techniques. This is also not surprising as the Combo technique provided the visual location indicator that was not present in the other techniques (see Figure 6.1 and Figure 6.2). There were no effects of Pairing or other significant interactions.

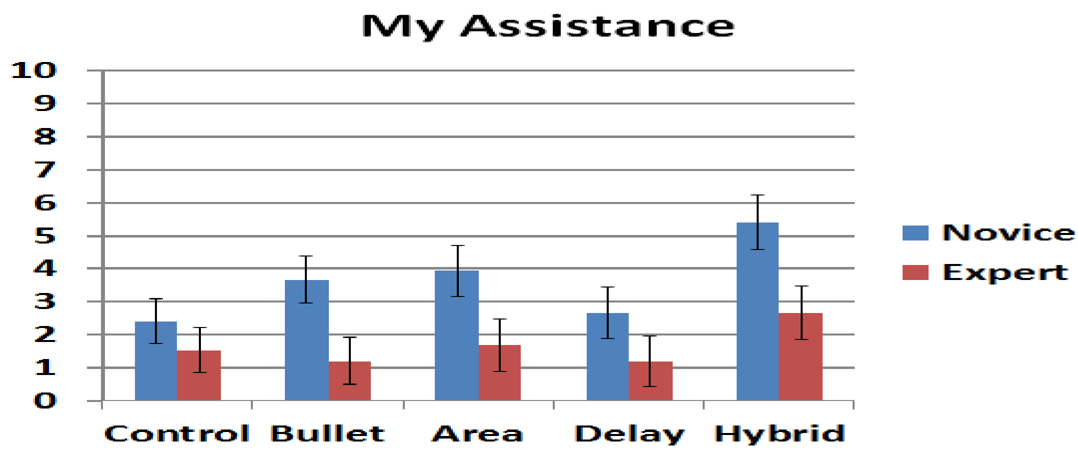


Figure 6.5: Means +/- Standard Error for My Assistance, split by expertise

We also asked players to rate the level of assistance that they thought their opponent was provided with on the same 11-point scale. A significant three-way interaction between Balancing Technique, Pairing, and Expertise ($F_{8,96}$=2.6, p=.014, η^2=.18) shows that for novice-expert pairings only, the stronger player perceived the assistance given to the weaker player more in Combo than in Bullet, Area, or Control and more in Delay than in Area or Control. In the intermediate-expert pairings, the stronger player perceived their opponent's assistance more in Combo than Control (see Figure 6.6).

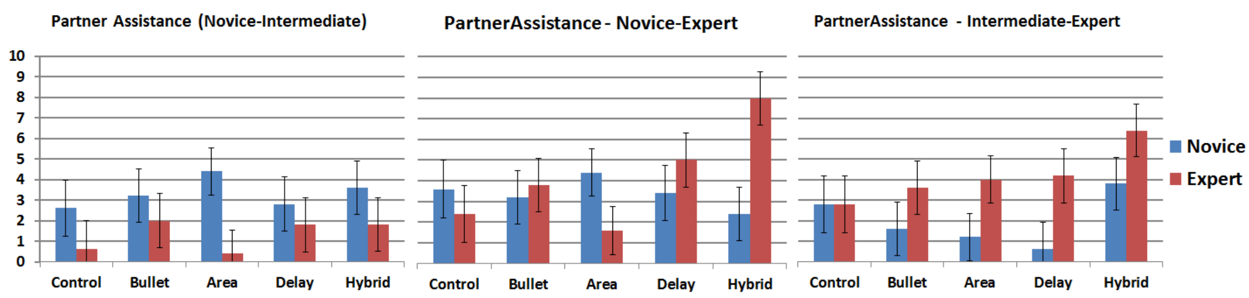


Figure 6.6: Means +/- Standard Error for Partner Assistance, split by pairing

6.3.2.2 Were the games perceived as fair?

We asked players to agree or disagree with the statement, "In general, did you find this round to be fair?" by answering either true or false. A chi-squared test on the frequency of responses split by whether the player was the stronger or weaker player in the dyad showed that the weaker players felt that games were fair only when assistance was applied, whereas the stronger players were divided on whether games were fair, regardless of provided assistance (Table 6.2).

Table 6.2: Count of responses of "FAIR' to the question of whether the game was fair, split by expertise.

	Control	Area	Bullet	Delay	Combo
Weaker	8/15	11/15	12/15	13/15	13/15
X^2	.067	3.3	5.4	8.1	8.1
p	.796	.071	.020	.005	.005
Stronger	11/15	9/15	11/15	8/15	10/15
X^2	3.3	0.6	3.3	0.67	1.7
p	.071	.439	.071	.796	.197

6.3.3 Experience

6.3.3.1 Did people experience greater competence with balancing?

As expected, there was a main effect of Expertise on perceived competence ($F_{1,24}$=5.5, p=.027, η^2=.19), in which the stronger player perceived himself or herself as more competent than the weaker player. The RM-MANOVA also showed a main effect of Technique on competence ($F_{4,96}$=3.8, p=.006, η^2=.14). Figure 6.3 and Figure 6.7 shows how the measure of perceived competence reflects the inverse of score differential. However, the significant interaction of Technique with Expertise ($F_{4,96}$=6.6, p≈.000, η^2=.22) shows that the differences in perceived competence were only changing for the weaker player (see Figure 6.7). Specifically, the pairwise corrections showed that the weaker player felt more competent after using Combo than any other approach, and more competent after using Bullet than Control, whereas there was no difference in perceived competence for the stronger member of the dyad. There were no effects of pairings or interactions with pairing.

6.3.3.2 Did people experience greater satisfaction of autonomy and relatedness with balancing?

There were no effects of Technique, Expertise, or Pairing on relatedness or autonomy. There were also no interactions between factors on either measure.

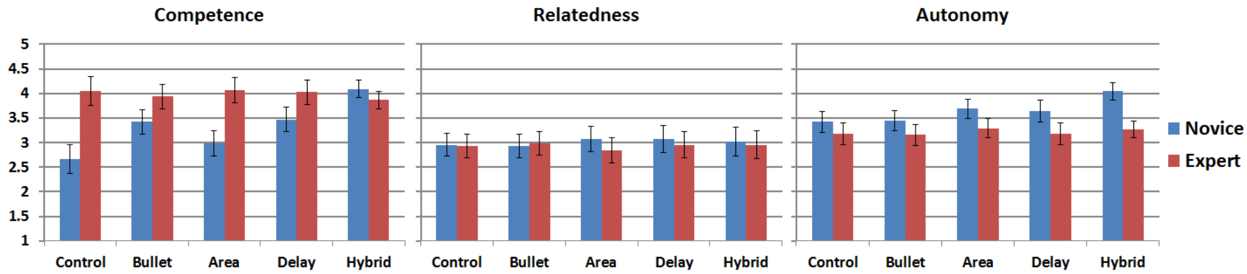


Figure 6.7: Means +/- Standard Error for Competence, Relatedness, and Autonomy split by expertise

6.3.3.3 Did the effects from balancing translate into enjoyment?

There were no effects of Pairing or Expertise on enjoyment. The RM-MANOVA showed a main effect of Technique on enjoyment ($F_{4,48}$=3.3, p=.015, η^2=.12), in which the pairwise comparisons showed that the Combo and Delay techniques were considered more enjoyable than Control (Combo was also more enjoyable than Bullet). There was no interaction of Technique with Pairing or Expertise, suggesting that all players experienced similar patterns of enjoyment regardless of their expertise within the dyad or the relative expertise of their opponent, i.e., enjoyment increased for *both* members of the dyad in conditions where assistance was most successful at balancing the score (Combo and Delay).

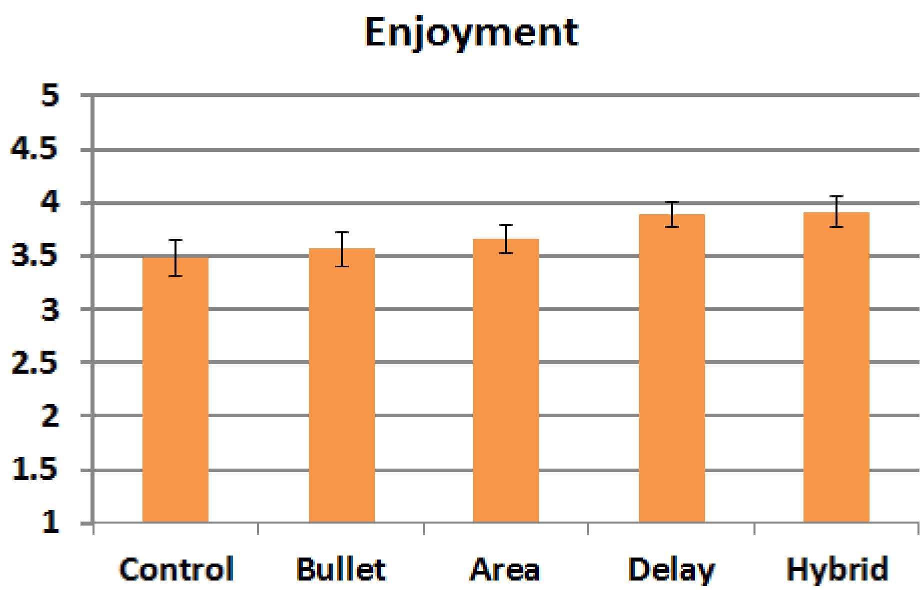


Figure 6.8: Means +/- Standard Error for Enjoyment, expertise and pairing combined

6.3.4 Summary of Results

Our results show that there are differences in the performance, perception, and experience of the various balancing techniques and that these differences often interact with whether the player was the weaker or stronger member of the dyad.

Specifically, although all assist techniques provided value, we can summarize our results with the following findings:

- Combo was best for balancing the games in terms of score differential and the number of lead reversals, although the stronger player still generally won the game;
- Combo and Delay were best at improving accuracy for the weaker player, with Delay improving the hit ratio of players and Combo improving the number of kills;
- Combo was the most perceptible approach, although mean ratings for perceptibility of all techniques were low;

- Combo was also most perceptible in terms of the stronger player sensing that their opponent received assistance, but only for expert-novice and expert-intermediate pairings;
- Games were perceived as more fair with assistance by the weaker player, Combo and Delay were rated as most fair by the weaker player;
- Combo produced the highest perceived competence in weaker players, with Bullet and Delay following; there were no differences for stronger players;
- Combo and Delay were the most enjoyable approaches for both the stronger and weaker players.

6.4 CONCLUSION

This chapter makes five main contributions. We show:

1. That player balancing can be extremely effective (and therefore should be considered by game designers who wish to engage a wider audience).
2. That the way in which balancing is applied is as important as the game mechanics that are being manipulated.
3. That combining game mechanics in a balancing technique can improve effectiveness.
4. That effective player balancing does not compromise perceptions of fairness for either player.
5. That although effective player balancing improves only the performance of the weaker player, it leads to greater enjoyment for both, suggesting the importance of the roles of suspense and uncertain outcome in generating positive play experiences.

Our overall goal was to develop and create a player balancing technique that is able to balance players of all skill levels in order to increase player enjoyment. In this step, we ended up at our desired goal. The Combo technique was able to balance players whose skills greatly differed, as well as players whose skills were similar. It also managed to increase enjoyment for all players, and was deemed as more fair by the weaker players in a pairing.

CHAPTER 7

CONCLUSION

This chapter elaborates on the findings of our studies investigating player balancing in First Person Shooters. We begin by outlining the contributions of this work, and then discuss the explanations for our results, limitations to our work, topics for future research, and finally present concluding remarks.

7.1 CONTRIBUTIONS

The primary contribution presented in this thesis is the first effective technique for balancing multiplayer FPS games. By creating and evaluating player balancing mechanics, we provide empirical data showing that our player balancing solution works, and should be considered by game designers as a method to increase the audience for this popular and competitive genre. Our results show that effective player balancing creates closer scoring matches and more outcome variability, which leads to greater enjoyment for both unassisted (expert) and assisted (novice) players. Our results also show that the two most effective techniques are *Combo* (which combines several skill assistance techniques) and *Delay* (which applies assistance longer).

There are also several secondary contributions:

1. We provide the first publically available empirical data on the performance of aiming assistance in 3D environments.
2. We show how different game factors affect aim assistance
3. We show that the targeting techniques that do not modify the aiming process of the user are more effective than the techniques that do modify the aiming process.

4. We show how different aiming assistance techniques react to certain in game situations, and provide recommendations for which technique would be optimal in which situations.
5. We show that combining different types of skill assistance is more effective than only assisting players with one game mechanic.
6. We show that despite significant assistance sometimes being provided, assisted players were not able to perceive that they were being assisted.
7. We show that game matches with assistance were perceived to be more fair by the weaker (assisted) player, but assistance did not reduce the fairness rating of the stronger player.

7.2 GENERAL DISCUSSION

In the following section we discuss our results and explain our findings, focusing primarily on the results from Chapter 6. Next, we cover the practical uses for our results, as well as limitations and future work that is not covered by this thesis.

7.2.1 Explanation of Results

Our main result was that all of our final four balancing techniques (Area, Bullet, Delay, Combo) worked, with the exception of Area. Some (Combo and Delay) worked extremely well at balancing and increasing player enjoyment. There are three underlying reasons for this success that can be valuable for designers.

First, our work shows that player balancing is not just about the method of adjusting a single game mechanic – the way in which the adjustment gets applied is also important. Using the 30-second-delay application method made a substantial difference compared to the Bullet technique alone. This method allowed us to deal with the problem of the weaker player getting near but never reaching the stronger player.

Second, the Combo technique demonstrates the value of combining multiple mechanics in one balancing technique, which agrees with player balancing research using driving games [17].

Multiple techniques appear to be particularly important in FPS games, where there are several aspects to a player's expertise [71]. The addition of several player balancing mechanics in addition to aim assistance allowed the Combo technique to perform best in balancing – although this also may have led to it being the most perceptible technique.

Third, one of our basic techniques (Bullet) performed better than suggested by our second study – this technique did well especially for the large skill differences of novice-expert pairs. The improvement can be explained by the fact that we recognized the magnitude of difference between weak and strong players, and made it stronger than in prior work.

Finally, it was interesting that although Bullet performed well, Area Cursor was not as successful. One reason is that the 3D implementation of Area was a direct transfer of the 2D implementation to work in 3D. This means that at greater distances, the area-cursor manipulation is less effective (i.e., the bullet is not as large as it seems when leaving the gun). In the final study, the large map meant more long-range engagements.

The increase in enjoyment from these techniques can be explained by players' increased perception of competence when using a player balancing technique. As noted by Ryan et al., feelings of increased competence result in increased satisfaction and enjoyment [62].

Our implementation of player balancing has the interesting quality that it scales with the expertise of the player. As novice players improve, the amount of assistance naturally declines, keeping games fair and balanced. Players can learn mastery of the skills needed to succeed in FPS play while the balancing technique naturally adapts to their skill.

7.2.1.1 Both individual and shared experience contribute meaningfully to experienced enjoyment

As expected, weaker players showed differences in aiming accuracy (hit ratio) with the various balancing techniques, whereas stronger players did not. This was also reflected in the rating of perceived competence for the players, which suggests that weaker players attributed the boost in their performance to their improved abilities. On the other hand, because the stronger players did not vary in their accuracy, their ratings of competence also did not change, suggesting that this measure reflects their own abilities, not their ability in relation to another player.

However, the ratings for enjoyment seem to reflect the experience of the dyad, rather than the experience of each individual within the dyad – for both weaker and stronger players, the ratings for enjoyment reflect the score differential between the two players rather than the individual abilities of the player providing the rating.

This distinction is interesting, because Self-Determination Theory specifies that the perceived competence of the player should translate into experienced intrinsic motivation (as measured by the interest-enjoyment subscale) [48]. For weaker players, this is true – the aiming assistance helps the weaker player feel more competent, which translates into increased enjoyment. However, for stronger players, perceived competence does not vary with balancing technique, as they were not provided with help and their accuracy did not improve. Although competence does not change for stronger players, their ratings for enjoyment do change, which is also not explained by changes in autonomy or relatedness. In particular, the ratings of enjoyment for stronger players reflect the convergence in score that results from the effective balancing techniques.

Our results reveal how in our multiplayer FPS game, competence ratings reflect the individual's performance (mirroring accuracy results), rather than the player's performance relative to their opponent; however, in contrast, enjoyment ratings reflect the level of balance in the dyad (mirroring the score convergence), rather than the performance of the individual player. Therefore the increased competition and increased uncertainty of the outcome of games in which the weaker player is assisted are experienced as more enjoyable for stronger players too.

7.2.1.2 Uncertainty in Game Outcome is Enjoyable

To further understand the role of score differential in multiplayer game enjoyment, we can look to previous literature. In a study of internet chess players, Abuhamdeh and Csikszentmihalyi [1] found that enjoyment peaked when players held a small performance advantage over their opponent (slightly smaller than the value of a pawn). Although unexplained by self-determination theory, the authors hypothesized that the differences could be explained by the suspense of an uncertain outcome.

To investigate the role of suspense, they conducted further experiments with single-player games in which the player thought that they were competing against a computer (but the opponent was actually a confederate researcher) [2]. The authors showed that peak enjoyment occurred when players were slightly beating the opponent, that peak suspense occurred when players were slightly trailing the opponent, and that perceived competence increased with the difference in score. In addition, they showed that when given the choice to play one more round of a game in which they had either won by a wide margin or a slim margin, 69% of participants chose to play the game that they had won by the slim margin, reflecting higher intrinsic motivation for games with uncertainty in the outcome [2].

Our results mirror these effects in the case of multiplayer FPS scenarios. Our participants experienced greatest enjoyment with the balancing technique that made the games closest, i.e., where the outcome was most uncertain. In addition, our stronger participants sensed that their weaker opponent was assisted, yet still felt the most enjoyment when outcome was uncertain.

7.2.2 Practical Significance and Limitations

One of our main goals in this research was to carry out balancing in realistic game scenarios. We succeeded – we used a real map and real weapons in a real game engine (UDK), and a type of competition that is used in realistic FPS play (deathmatch). Players were allowed to use whatever strategy they wanted, and we did not constrain player movement, evasive tactics, or other hallmarks of expert behavior. This degree of external validity suggests that our player-balancing techniques can be taken up by game developers, and can successfully improve the range of players who can enjoy the game.

However, there were also some limitations in our study setup that require further consideration:

1. More than two players. Both of our application methods use score differential in the two-player game to determine the onset and strength of balancing. In some FPS scenarios, using different game metrics may be necessary. For example, a metric such as kill-to-death ratio can indicate each player's relative expertise, regardless of the size of the group. Additionally, team matches (which aren't free-for-alls) may require the dynamic assistance calculations to be applied on a per team level rather than a per user, or may require different metrics all together.

2. Larger maps with more resources. Our UDK map was small, in order to provide more opportunity for conflict in the study sessions. In addition, our study map contained few spawning resources (e.g., health packs or powerups). In larger maps, expert behaviors such as the ability to memorize the map, or the ability to remember and utilize periodic resources could have a larger effect on score and outcome than what we saw in our study.

3. Strategy vs. game mechanic. There are also aspects of FPS games that clearly differ between experts and novices, but that would be difficult to add to a balancing technique. For example, higher-level strategies such as deployment of multiple players or selecting the best weapon for a particular situation clearly arise through experience, and the decision processes that go into these skills are not easily amenable to balancing. In addition, some strategies that could be part of a balancing technique might be better left out. For example, experts are much better at knowing when to reload – and although novice players could feasibly be given an auto-reload assist, it may be that this would make the game feel too much like the computer is controlling all aspects of the gameplay.

4. Longer-term effects. This thesis looks at only how players respond when first using aim assistance (i.e., the early effects) and does not address what might happen after long-term repeated use (i.e., the long term effects). It is possible that our techniques will perform differently over a longer term. For example, stronger players who take great pride in their abilities may begin to dislike the fact that novices get an advantage; or, it may be that experts find exploits that allow them to outwit or negate the balancing strategies. Our participants appeared to like the fact that there was strong competition, however, and so we do not expect either of these situations to occur for most players. Further work is also needed to examine how aiming assistance may affect skill development. It is not clear if aiming assistance would lead to novices relying on it too heavily and being unable to advance their skills. Further work over longer time periods is clearly needed.

5. Comparisons between aiming assistance techniques: Because the effect strength and values of each aiming assistance technique was determined independently of each another, and some of the effects are very different, it isn't possible to compare the techniques against each other. For instance, the intersection traces Area Cursor and Bullet Magnetism essentially only differ in the shape that is projected. However, the sizes of the traces are very different. Lock and Sticky Targets vary greatly, as one moves the crosshair towards a target, and the other slows down the crosshairs once the target is reached by the player. Therefore, it's not valid to compare the techniques against each other.

7.2.3 Future Work

There are three ways in which the studies presented in this thesis can be extended.

First, future work should investigate the performance of aiming assistance with controllers. Most publically available work in 3-dimensional aiming assistance has only investigated how well the techniques work with mice. Gamepads and mice require different expertise, which may have effects on the efficacy of balancing with aim assistance. Additionally, since aiming assistance is used extensively on console FPS games, having publically available data to determine which techniques are most effective would be valuable. Given this use of aiming assistance in console games already, it should be simple to include it as an option for multiplayer balancing if it was found to be as effective as with mice.

Second, the findings in this thesis indicate that incorporating player balancing techniques improves game experience for players. Given the success in FPS, it follows that other competitive genres can incorporate dynamic skill assistance to make the games more enjoyable. For example, fighting games could provide weaker players stronger attacks or extra health depending on their performance.

Finally, investigating how noticeable and effective the aiming assistance is with non-instant hit weapons would be valuable, as most games use projectile weapons and may result in modifications to the assistance techniques to get them to work.

7.3 CONCLUSIONS

7.3.1 Summary of Thesis

First Person Shooter games are a very popular video game genre. Competition is a big part of First Person Shooter games, but competition brings complications when trying to achieve game balance. Weaker players get frustrated and strong players get bored with the lack of challenge if the skill levels of the players differ. Several approaches (like matchmaking) have been developed to try and achieve this game balance; however, none of them have been shown to work well.

A recent approach to solve the skill gap problem is aiming assistance, which has been used in 2D environments to balance gameplay. This suggested that it could be a good mechanism to use in 3D environments as well in order to increase enjoyment for all parties. We used aiming assistance as a starting point to solve the player balancing issue in competitive First Person Shooters. We eventually determined that helping only one area of skill was not enough to create player balance, and created two player balancing techniques (*Combo* and *Delay*) that were effective at increasing player enjoyment and creating balanced games.

Three stages were necessary in order to solve the skill gap problem in competitive FPS games.

The first stage involved determining the efficacy of aiming assistance techniques at increasing aiming performance in 3D FPS environments, as there was no previously available information on this topic. To do this, we ran three user studies to examine the performance of five different aim assist techniques, most of which are used in video games for reasons other than balancing. These three studies incrementally added different "real world" game elements to determine how they would interfere with aiming assistance effectiveness. Although the assistance techniques worked well in the first target-range study that was similar to previous 2D Fitts' tasks, their performance was reduced when the "real game" elements were introduced. This stage provided initial insight into how game factors affect aim assistance, and identified techniques that show potential for player balancing. We found that two techniques – *Bullet Magnetism* and *Area Cursor* – fulfilled our criteria of working well in a wide variety of situations without being too perceptible. We also found that some techniques that were successful in 2D environments (*Gravity* and *Sticky Targets*), worked poorly throughout those initial 3D tests. Because this stage showed that aiming assistance could be used to improve performance of aiming tasks in 3D FPS games, it suggested that aiming assistance could be a good mechanism to use for player balancing.

In the second stage, we applied what we learned in the first stage to create an initial attempt at solving the skill gap problem in competitive FPS games. To do this, we investigated Area Cursor and Bullet Magnetism in a competitive two-player FPS game. Participants were split into novice and expert pairs and were made to play a deathmatch mode with the aiming assistance techniques being applied dynamically based on performance. We found that the techniques managed to increase accuracy and competence ratings of the novices, but they had no effect on performance game outcome or performance measures, and also had no effect on the enjoyment of the game as a result. These initial results showed that improving the aiming accuracy of novice players is only the first step in effectively balancing a game, and the participant feedback suggested that novices had more deficiencies than just aiming.

From the results in stage two, we realized that in order for assistance to be successful, we couldn't simply focus on a single element of gameplay. Therefore, in stage three we developed new player balancing techniques based on three ideas: extending the range of assistance, changing the way that a manipulation is applied, and adding more game mechanics to the technique. These four techniques were Area Cursor (*Area*), Bullet Magnetism (*Bullet*), Bullet Magnetism with Delay (*Delay*), and Bullet Magnetism with Location Awareness Assistance and Damage Modification (*Combo*). We tested these techniques in a controlled study again using one-on-one deathmatches in a realistic FPS game. Our results showed that our new balancing techniques were extremely effective – they were able to balance even amongst players with large skill differences. Combo, which combined several kinds of skill assistance, was most effective at balancing. Additionally, the techniques that were most effective at balancing were also rated as most enjoyable by both players.

7.3.2 Concluding Remarks

The problem addressed in this thesis was that in competitive FPS games, players with different skill levels will not have an enjoyable experience when playing against each other. Our solution was to apply skill assistance dynamically to bring the weaker player to the level of the stronger player. While our initial attempts used only aiming assistance, our findings showed that an assistance mechanism that targets only a single gameplay element was not effective enough to balance players with large skill differences. We used this finding to create our final player balancing techniques *Combo* and *Delay*. Combo combined one of the better performing aiming assistance techniques (Bullet Magnetism) along with location awareness assistance and variable damage, and Delay extended the amount of time aiming assistance was available once the player scores were equalized.

Our results showed that the new balancing techniques were extremely effective – they were able to create the most balanced matches even amongst players with large skill differences. The techniques that were most effective at balancing also led to an increase in the enjoyment ratings of the game for all players. These findings indicate that the FPS player balancing techniques developed are extremely effective, and should be used by game designers who wish to engage a wider audience and create more satisfied players.

REFERENCES

[1] Abuhamdeh, S., & Csikszentmihalyi, M. 2012. The importance of challenge for the enjoyment of intrinsically motivated, goal-directed activities. In Personality & Social. Psychology Bulletin, 38 (3), 317-330.

[2] Abuhamdeh, S., Csikszentmihalyi, M., & Jalal, B. 2014. Enjoying the possibility of defeat: Outcome uncertainty, suspense, and intrinsic motivation. Motivation & Emotion, 1-10. Published online.

[3] Adams, E. Difficulty Modes and Dynamic Difficulty Adjustment. Gamasutra: The Art & Business of Making Games Blog. Published May 2008. Retrieved April, 2015 from gamasutra.com/view/feature/3660/the_designers_notebook_.php

[4] Adams, E. Fundamentals of Game Design. New Riders, 2010.

[5] Arkane Studios. 2012. Dishonored. Bethesda Softworks.

[6] Asano, T., Sharlin, E., Kitamura, Y., Takashima, K., & Kishino, F. 2005. Predictive interaction using the Delphian desktop. In User Interface and Software and Technology Symposium (UIST) 2005, 133-141.

[7] Balakrishnan, R. "Beating" Fitt's law: virtual enhancements for pointing facilitation. In International Journal of Human-Computer Studies (IJHCS) 61, 857-874.

[8] Baldwin, A., Johnson, D., & Wyeth, P. 2014. The effect of multiplayer dynamic difficulty adjustment on the player experience of video games. In CHI'14 Extended Abstracts on Human Factors in Computing Systems (CHI EA), 1489-1494.

[9] Bartle, R. 1996. Hearts, clubs, diamons, spades: Players who suit MUDs. In Hournal of MUD Research, 1 (1). Retrieved April, 2015 from http://mud.co.uk/richard/hcds.htm

[10] Bateman, S., Mandryk, R.L., Stach, T. & Gutwin, C. 2011. Target assistance for subtly balancing competitive play. In Proceedings of the 2011 annual conference on Human factors in computing systems (CHI '11), 2355-2364.

[11] Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P. Bederson, B., & Zierlinger, A. 2003. Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content. In Interact 2003, 57-64.

[12] Blanch, R., Guiard Y., & Beaudouin-Lafon, M. 2004. Semantic Pointing: Improving Target Acquisition with Control-Display Ratio Adaptation. In Proceedings of the 2004 annual conference on Human factors in computing systems (CHI '04), 24-29.

[13] Blizzard Entertainment. 2010. Starcraft 2. Blizzard Entertainment.

[14] Booth, M. 2009. The AI systems of left 4 dead. In Keynote, Fifth Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE ‘09).

[15] Bostan, B., & Öğüt, S. 2009. Game challenges and difficulty levels: lessons learned From RPGs. In International Simulation and Gaming Association Conference.

[16] Bungie, 343 Industries. 2001-2014. Halo series. Microsoft Studios.

[17] Cechanowicz, J., Gutwin, C., Bateman, S., Mandryk, R.L., & Stavness, I. 2014. Improving Player Balancing in Racing Games. In Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play (CHI Play), 47-56.

[18] Cockburn, A, & Firth, A. 2003. Improving the Acquisition of Small Targets. In People and Computers XVII—Designing for Society, 181-196.

[19] Elmqvist, N., & Fekete, J. 2008. Semantic Pointing for Object Picking in Complex 3D Environments. In Proceedings of Graphics Interface (GI) 2008, 243-250.

[20] Entertainment Software Association. 2015 Essential Facts About the Computer and Video Game Industry. http://www.theesa.com/wp-content/uploads/2015/04/ESA-Essential-Facts-2015.pdf

[21] Epic Games. 1999-2015. Unreal Tournament series. Epic Games & GT Interactive Software.

[22] Epic Games. 2009. Unreal Development Kit. http://www.unrealengine.com/udk/

[23] Epic Games. 2011. Gears of War 3. Microsoft Studios.

[24] Ermi, L., & Mäyrä, F. 2005. Fundamental components of the gameplay experience: Analysing immersion. In Worlds in play: International perspectives on digital games research 37.

[25] Falstein, N. Understanding fun – the Theory of Natural Funativity. 2005. In Introduction to Game Development, Rabin, S. (ed.), 71-98.

[26] Feil, J., & Scattergood, M. 2005. Beginning game level design. Thomson Course Technology.

[27] Findlater, L, Jansen, A., Shinohara, K., Dixon, M., Kamb, P., Rakita. J., & Wobbrock, J.O. Enhanced Area Cursors Reducing Fine Pointing Demands for People with Motor Impairments. In User Interface and Software and Technology Symposium (UIST) 2010, 153-162.

[28] Fitts, P.M. 1954. The information capacity of the human motor system in controlling the amplitude of movement. In Journal of experimental psychology 47, 381-391.

[29] Flatla, D., Gutwin, C., Nacke, L.E., Bateman, S., & Mandryk, R.L. 2011. Calibration Games: Making Calibration Tasks Enjoyable by Adding Motivating Game Elements. In User Interface and Software and Technology Symposium (UIST) 2011, 403-412.

[30] Fullerton, T. 2014. Game design workshop: a playcentric approach to creating innovative games. CRC Press.

[31] Gerling, K.M., Miller, M.K., Mandryk, R.L., Birk, M., & Smeddinck, J.D. 2014. Effects of Balancing for Physical Abilities on Player Performance, Experience and Self-Esteem in Exergames. In Proceedings of the 2014 annual conference on Human factors in computing systems (CHI '14), 2201-2210.

[32] Gilleade, K., Dix, A., & Allanson, J. 2005. Affective videogames and modes of affective gaming: Assist me, challenge me, emote me. In Proceedings of DIGRA 2005.

[33] Go, J., Ballagas, R., & Spasojevic, M. 2012. Brothers and sisters at play: exploring game play with siblings. In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work. 739-748.

[34] Guiard, Y., Blanch, R., & Beaudouin-Lafon, M. 2004. Object pointing: a complement to bitmap pointing in GUIs. In Proceedings of Graphics Interface (GI) 2004, 9-16.

[35] Herbrich, R., Minka, T., & Graepel, T. 2007. TrueSkill(TM): A Bayesian skill rating system. In Advances in Neural Information Processing Systems 20, 569-576.

[36] Hernandez, H. A., Graham, T. C., Fehlings, D., Switzer, L., Ye, Z., Bellay, Q., Hamza, M. A., Savery, C., & Stach, T. 2012. Design of an Exergaming station for Children with Cerebral Palsy. In Proceeding of the 2012 annual conference on human factors in computing systems (CHI ’12). 2619-2628.

[37] Hunicke, R. 2005. The case for dynamic difficulty adjustment in games. In Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology, 429-433.

[38] Hunicke, R., & Chapman, V. 2004. AI for Dynamic Difficulty Adjustment in Games. In Challenges in Game Artificial Intelligence AAAI Workshop, 91-96.

[39] Infinity Ward, Treyarch, & Sledgehammer Games. 2003-2014. Call of Duty series. Activision.

[40] Kabbash, P. & Buxton, W. 1995. The "Prince" Technique: Fitts’ Law and Selection Using Area Cursors. In Proceedings of the 1995 annual conference on Human factors in computing systems (CHI '95), 273-279.

[41] Klimmt, C., Rizzo, A., Vorderer, P., Kock, J., & Fischer, T. 2009. Experimental evidence for suspense as determinant of video game enjoyment. In CyberPsychology & Behaviour,

12 (1), 29-31.

[42] Klimmt, C., Blake, C., Hefner, D., Vorderer, P., & Roth, C. 2009. Player Performance, Satisfaction, and Video Game Enjoyment. In Proceedings of the 2009 international conference on Entertainment computing (ICEC ’09). 1-12.

[43] Koster, R. 2004. Theory of Fun for Game Design. O'Reilly Media, Inc.

[44] Laurens, P., Paige, R. F., Brooke, P. J., & Chivers, H. 2007. A novel approach to the detection of cheating in multiplayer online games. In Engineering Complex Computer Systems 2007. 97-106.

[45] Liquipedia . Battle.net Leagues. http://wiki.teamliquid.net/starcraft2/Battle.net_Leagues

[46] Looser, J., Cockburn, A. & Savage, J. 2005. On the Validity of Using First-Person Shooters for Fitts' Law Studies. In People and Computers XIX 2, 33-36.

[47] Mandryk, R.L., & Gutwin, C. 2008. Perceptibility and Utility of Sticky Targets. In Proceedings of Graphics Interface (GI) 2008, 65-72.

[48] McAuley, E., Duncan, T., & Tammen, V. V. 1989. Psychometric properties of the Intrinsic Motivation Inventory in a competitive sport setting: In Research quarterly for exercise and sport 60 (1), 48-58.

[49] Meyer, D., Smith, J., Kornblum, S., Abrams, R., & Wright, C. 1988. Optimality in human motor performance: ideal control of rapid aimed movements. In Psychological review 95, 340-370.

[50] Moulard, J. G., Kroff, M. W., & Folse, J. A. G. 2012. Unraveling consumer suspense: The role of hope, fear, and probability fluctuations. In Journal of Business Research 65 (3), 340-346.

[51] Nacke, L. E., Bateman, C., & Mandryk, R. L. 2014. BrainHex: a neurobiological gamer typology survey. In Entertainment computing 5 (1), 55-62.

[52] Nardi, B., & Harris, J. 2006. Strangers and friends: Collaborative play in World of Warcraft. In Proc. Computer Supported Cooperative Work, 149-158.

[53] Natapov, D., Castellucci, S. J., & MacKenzie, 2009. I. S. ISO 9241-9 evaluation of video game controllers. In Proceedings of Graphics Interface (GI) 2009. 223-230.

[54] Newheiser, Mark. Playing Fair: A Look at Competition in Gaming. Strange Horizons. Published March 2009. Retrieved April 2015 from www.strangehorizons.com/2009/20090309/newheiser-a.shtml

[55] Nintendo. 1986-2015. Legend of Zelda series. Nintendo.

[56] Nintendo. 1992-2014. Mario Kart Series. Nintendo.

[57] Przybylski, A. K., Rigby, C. S. 2010. A motivational model of video game engagement. In Review of General Psychology 14 (2), 154-166

[58] Rigby, S., Ryan, R. Rethinking Carrots: A New Method For Measuring What Players Find Most Rewarding and Motivating About Your Game. Gamasutra: The Art & Business of Making Games Blog. Published January 2007. Retrieved April, 2015 from gamasutra.com/view/feature/130155/rethinking_carrots_a_new_method_.php

[59] Rockstar Games. 1997-2015. Grand Theft Auto. Rockstar Games.

[60] Rockstar San Diego. 2010. Red Dead Redemption. Rockstar Games.

[61] Ryan, R. M., Deci, E. L., 2000. Intrinsic and extrinsic motivations: Classic definitions and new directions. In Contemporary Educational psychology 25, 54-67.

[62] Ryan, R. M., Rigby, C. S., & Przybylski, A. K. 2006. The motivational pull of video games: A self-determination theory approach. In Motivation and Emotion 30 (4), 344-360.

[63] Salen, K., & Zimmerman, E. 2004. Rules of play: Game design fundamentals. MIT press.

[64] Sherry, J., & Lucas, K., Greenberg, B. S., Lachlan, K. A. 2003. Video game uses and gratifications as predictors of use and game preference. In Playing Video Games: Motives, Responses, and Consequences, 213-224

[65] Sweetser, P. & Wyeth, P. 205. GameFlow: a model for evaluating player enjoyment in games. In Computers in Entertainment (CIE) 3 (3), 3-3.

[66] Turtle Rock Studios. 2008. Left 4 Dead. Valve Corporation.

[67] Valve Corporation. 2007. Team Fortress 2. Valve Corporation.

[68] Vaughan, Graham Michael, & Michael A. Hogg. 2007. Introduction to Social Psychology. Pearson Education Australia.

[69] Vicencio-Moreira, R., Mandryk, R. L., & Gutwin, C., & Bateman, S. 2014. The effectiveness (or lack thereof) of aim-assist techniques in First Person Shooter games. 2014. In Proceedings of the 2014 annual conference on Human factors in computing systems (CHI '14), 937-946.

[70] Vicencio-Moreira, R., Mandryk, R.L., & Gutwin, C. 2014. Balancing Multiplayer First-Person Shooter Games using Aiming Assistance. In IEEE Games, Entertrainment, and Media 2014 (IEEE GEM '14), 77-84.

[71] Vicencio-Moreira, R., Mandryk, R.L., & Gutwin, C. 2015. Now You Can Compete With Anyone: Balancing Players of Different Skill Levels in a First- Person Shooter Game. In Proceedings of the 2015 annual conference on Human factors in computing systems (CHI '15), 2255-2264.

[72] Vorderer, P., Hartmann, T., & Klimmt, C. 2003. Explaining the enjoyment of playing video games: the role of competition. In Proceedings of the second international conference on Entertainment computing (ICEC ’03), 1-9.

[73] Wobbrock, J.O., Fogarty, J., Liu, S., Kimuro, S., & Harada, S. 2009. The Angle Mouse: Target-Agnostic Dynamic Gain Adjustment Based on Angular Deviation. In In Proceedings of the 2009 annual conference on Human factors in computing systems (CHI '09), 1401-1410.

[74] Worden A, Walker N, Bharat K, & Hudson, S. 1997. Making Computers Easier for Older Adults to Use: Area Cursors and Sticky Icons. In In Proceedings of the 1997 annual conference on Human factors in computing systems (CHI '97), 266-271.

[75] Youssef, A. & Cossell, S. 2009. Thoughts on adjusting perceived difficulty in games. In Proceedings of the Sixth Australasian Conference on Interactive Entertainment (IE ’09), 14.

APPENDIX A

STUDY 1 FORMS

7.4 CONSENT FORMS

7.4.1 S1 (Shooting Range)

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF SASKATCHEWAN
INFORMED CONSENT FORM

Research Project:	**Targeting Assistance in First Person Shooters**
Investigators:	Dr. Regan Mandryk, Department of Computer Science (966-4888)
	Dr. Carl Gutwin, Department of Computer Science (966-8646)
	Rodrigo Vicencio, Department of Computer Science

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

This study is concerned with **measuring targeting performance in a 3d game with a first person perspective.**

The goal of the research is to **determine the effectiveness of different aim assist methods in increasing targeting performance in a 3d first person computer game.**

The session will require **30** minutes, during which you will be asked **play a 3d first person game where you will be playing in a "shooting range" type map while shooting stationary targets.**

At the end of the session, you will be given more information about the purpose and goals of the study, and there will be time for you to ask questions about the research. As a way of thanking you for your participation and to help compensate you for your time and any travel costs you may have incurred, you will receive a $**5** honorarium at the end of the session.

The data collected from this study will be used in articles for publication in journals and conference proceedings.

As one way of thanking you for your time, we will be pleased to make available to you a summary of the results of this study once they have been compiled (usually within two months). This summary will outline the research and discuss our findings and recommendations. If you would like to receive a copy of this summary, please write down your email address here.

Contact email address:____

All personal and identifying data will be kept confidential. If explicit consent has been given, textual excerpts, photographs, or video recordings may be used in the dissemination of research results in scholarly journals or at scholarly conferences. Confidentiality will be preserved by using pseudonyms in any presentation of textual data in journals or at conferences. The informed consent form and all research data will be kept in a secure location under confidentiality in accordance with University policy for 5 years post publication. Do you have any questions about this aspect of the study?

You are free to withdraw from the study at any time without penalty and without losing any advertised benefits. Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed. Your right to withdraw data from the study will apply until results have been disseminated, data has been pooled, etc. After this, it is possible that some form of research dissemination will have already occurred and it may not be possible to withdraw your data.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact:

- Dr. Regan Mandryk, Associate Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. If you have further questions about this study or your rights as a participant, please contact:

- Dr. Regan Mandryk, Associate Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca
- Research Ethics Office, University of Saskatchewan, (306) 966-2975 or toll free at 888-966-2975.

Participant's signature:____

Date:____________________

Investigator's signature:____

Date:____________________

A copy of this consent form has been given to you to keep for your records and reference. This research has the ethical approval of the Research Ethics Office at the University of Saskatchewan.

7.4.2 S2 (Full Walkthrough)

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF SASKATCHEWAN
INFORMED CONSENT FORM

Research Project:	**Targeting Assistance in First Person Shooters**
Investigators:	Dr. Regan Mandryk, Department of Computer Science (966-4888)
	Dr. Carl Gutwin, Department of Computer Science (966-8646)
	Rodrigo Vicencio, Department of Computer Science

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

This study is concerned with **measuring targeting performance in a 3d game with a first person perspective.**

The goal of the research is to **determine the effectiveness of different aim assist methods in increasing targeting performance in a 3d first person computer game.**

The session will require **90** minutes, during which you will be asked **play a 3d first person game where you will be playing a basic level several times while shooting moving targets**. **You will be asked to fill out a questionnaire at the end of each level.**

At the end of the session, you will be given more information about the purpose and goals of the study, and there will be time for you to ask questions about the research. As a way of thanking you for your participation and to help compensate you for your time and any travel costs you may have incurred, you will receive a $**15** honorarium at the end of the session.

The data collected from this study will be used in articles for publication in journals and conference proceedings.

As one way of thanking you for your time, we will be pleased to make available to you a summary of the results of this study once they have been compiled (usually within two months). This summary will outline the research and discuss our findings and recommendations. If you would like to receive a copy of this summary, please write down your email address here.

Contact email address:____

All personal and identifying data will be kept confidential. If explicit consent has been given, textual excerpts, photographs, or video recordings may be used in the dissemination of research results in scholarly journals or at scholarly conferences. Confidentiality will be preserved by using pseudonyms in any presentation of textual data in journals or at conferences. The informed consent form and all research data will be kept in a secure location under confidentiality in accordance with University policy for 5 years post publication. Do you have any questions about this aspect of the study?

You are free to withdraw from the study at any time without penalty and without losing any advertised benefits. Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed. Your right to withdraw data from the study will apply until results have been disseminated, data has been pooled, etc. After this, it is possible that some form of research dissemination will have already occurred and it may not be possible to withdraw your data.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact:

- Dr. Regan Mandryk, Associate Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. If you have further questions about this study or your rights as a participant, please contact:

- Dr. Regan Mandryk, Associate Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca
- Research Ethics Office, University of Saskatchewan, (306) 966-2975 or toll free at 888-966-2975.

Participant's signature:____

Date:____________________

Investigator's signature:____

Date:____________________

A copy of this consent form has been given to you to keep for your records and reference. This research has the ethical approval of the Research Ethics Office at the University of Saskatchewan.

7.4.3 S3 (Full Walkthrough 2)

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF SASKATCHEWAN
INFORMED CONSENT FORM

Research Project:	**Targeting Assistance in First Person Shooters**
Investigators:	Dr. Regan Mandryk, Department of Computer Science (966-4888)
	Dr. Carl Gutwin, Department of Computer Science (966-8646)
	Rodrigo Vicencio, Department of Computer Science

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

This study is concerned with **measuring targeting performance in a 3d game with a first person perspective.**

The goal of the research is to **determine the effectiveness of different aim assist methods in increasing targeting performance in a 3d first person computer game.**

The session will require **60** minutes, during which you will be asked **play a 3d first person game where you will be playing a basic level several times while shooting moving targets**. **You will be asked to fill out a questionnaire at the end of each level.**

At the end of the session, you will be given more information about the purpose and goals of the study, and there will be time for you to ask questions about the research. As a way of thanking you for your participation and to help compensate you for your time and any travel costs you may have incurred, you will receive a $**10** honorarium at the end of the session.

The data collected from this study will be used in articles for publication in journals and conference proceedings.

As one way of thanking you for your time, we will be pleased to make available to you a summary of the results of this study once they have been compiled (usually within two months). This summary will outline the research and discuss our findings and recommendations. If you would like to receive a copy of this summary, please write down your email address here.

Contact email address:____

All personal and identifying data will be kept confidential. If explicit consent has been given, textual excerpts, photographs, or video recordings may be used in the dissemination of research results in scholarly journals or at scholarly conferences. Confidentiality will be preserved by using pseudonyms in any presentation of textual data in journals or at conferences. The informed consent form and all research data will be kept in a secure location under confidentiality in accordance with University policy for 5 years post publication. Do you have any questions about this aspect of the study?

You are free to withdraw from the study at any time without penalty and without losing any advertised benefits. Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed. Your right to withdraw data from the study will apply until results have been disseminated, data has been pooled, etc. After this, it is possible that some form of research dissemination will have already occurred and it may not be possible to withdraw your data.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact:

- Dr. Regan Mandryk, Associate Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. If you have further questions about this study or your rights as a participant, please contact:

- Dr. Regan Mandryk, Associate Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca
- Research Ethics Office, University of Saskatchewan, (306) 966-2975 or toll free at 888-966-2975.

Participant’s signature:____

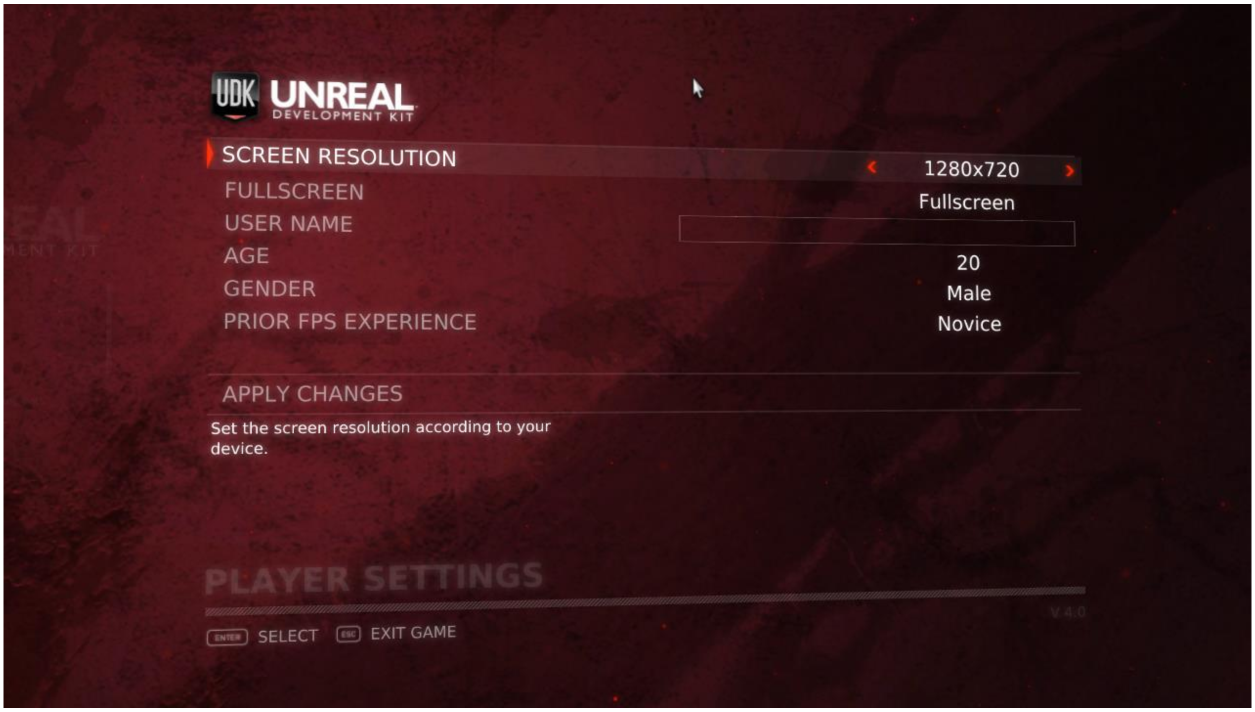
Date:____________________

Investigator’s signature:____

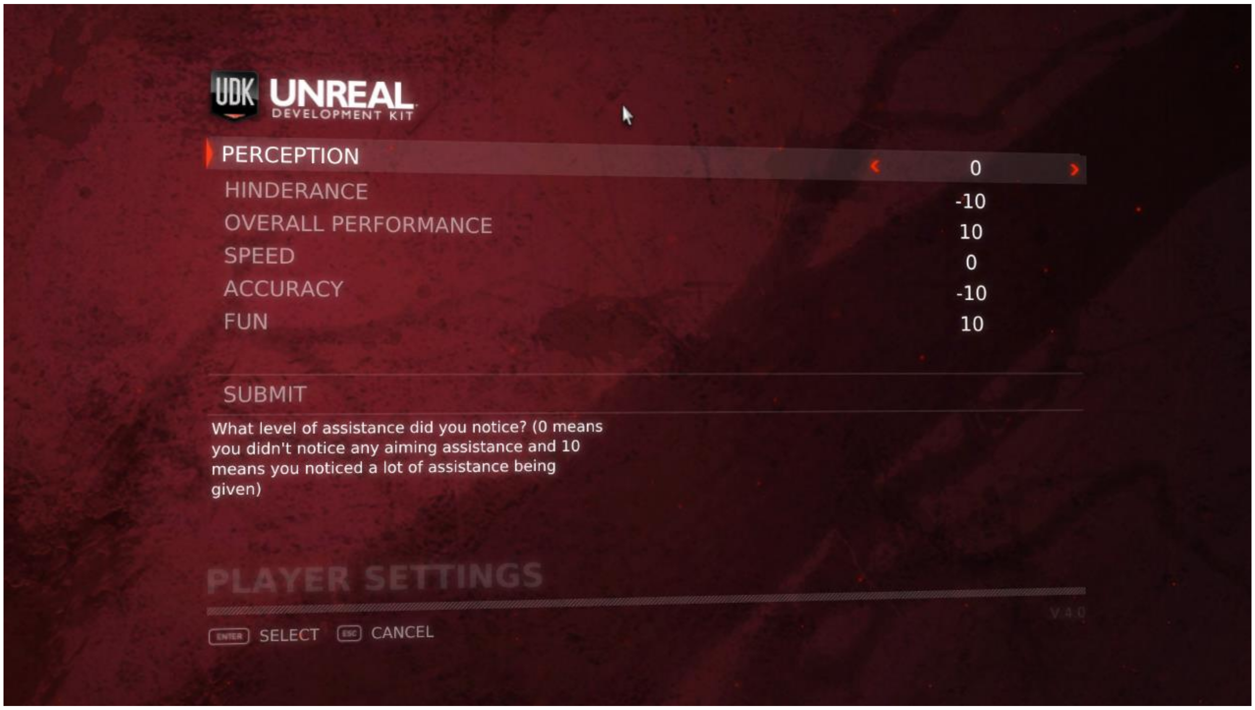
Date:____________________

A copy of this consent form has been given to you to keep for your records and reference. This research has the ethical approval of the Research Ethics Office at the University of Saskatchewan.

8.1 DEMOGRAPHIC FORM



8.2 POST GAME QUESTIONNAIRE



1. **Perception:** What level of assistance did you notice? (0 means you didn't notice any aiming assistance and 10 means you noticed a lot of assistance being given)
2. **Hindrance:** Did this method hinder you in aiming? (-10 means you couldn't hit the targets you meant to hit, 0 means it was neutral and 10 means you couldn't miss the targets you meant to hit)
3. **Overall Performance:** Overall, how do you think you did? (-10 means the worst you've ever done, 10 means the best you've ever done)
4. **Speed:** How fast were you? (-10 means the slowest you've ever gone through, 10 means the fastest you've ever gone through)
5. **Accuracy:** Overall, how was your accuracy? (-10 means the least accurate you've ever been, 10 means the most accurate you've ever been)
6. **Fun:** Did this method increase fun? (-10 means you had no fun at all, 0 means it was neutral and 10 means the most fun you've had)

8.3 EXPERIMENT SCRIPT

8.3.1 S1 (Shooting Range)

Setup

- Have the FPS game installed on a computer. Most recent version is at http://hcigame.usask.ca/
- Make sure the logging server is up
- Have the webpage http://hcigame.usask.ca/ open, which contains information on controls
- Have notebook and pen ready to record observations from gameplay, important comments, etc
- Delete " \UDK\AimAssistStudy\UDKGame\Config\UDKUDNGame.ini" so the user will be prompted for user details next time they start the game.
- Open " \UDK\AimAssistStudy\UDKGame\Config\DefaultUDNGame.ini" and under [UDNGame.UDNUIDataStore_StringList]increment DefaultValueIndex of the ShootRangeStartAimMode item.

Introduction

- Give consent form
  - "You will get $5 for participating, and are free to withdraw from the study at any time, no questions asked and you can still collect the money. Your data will also be removed from any analysis"
  - Collected data will be used in academic publications and other publicly available information: we will not use any identifying information
- Explain: "We are testing how effective different types of aiming assistance are. We are trying to determine which method gives the best performance increase in a FPS. You will be playing on a shooting range, and you'll be playing for 6 rounds plus a training round at the start. Each round has 7 waves of enemies at different distances, each wave lasts for 10

seconds. While playing, try to 'kill' as many targets as possible in these 10 seconds. This should take about 20 minutes. "

- Show the participant the webpage with the controls, http://hcigame.usask.ca/
- "Everything is pretty straightforward, all you need to know is the left mouse button is to shoot. One important thing to note is that in one of the rounds, it will prompt you to use the Q button to activate locking. Make sure you take note of this and use it."
- Any questions? More details about the implementations will have to wait until after the experiment
- Start the game. The first screen should prompt them to enter their information.
- Once finished, select "Shooting Range" from the main menu.
- Let them know they can change the sensitivity with the [and] keys, and change the volume with + and – while in game

8.3.2 S2 (Full Walkthrough)

Setup

- Have the FPS game installed on a computer. Most recent version is at https://dl.dropboxusercontent.com/u/4255072/UDKInstall-AimAssist.exe
- Make sure the logging server is up
- Have the webpage http://hcigame.usask.ca/ open, which contains information on controls
- Have notebook and pen ready to record observations from gameplay, important comments, and signs of frustration or fun.
- Delete " \UDK\AimAssistStudy\UDKGame\Config\UDKUDNGame.ini" so the user will be prompted for user details next time they start the game

Introduction

- Give consent form
    - "You will get $15 for participating, and are free to withdraw from the study at any time, no questions asked and you can still collect the money. Your data will also be removed from any analysis"
    - Collected data will be used in academic publications and other publicly available information: we will not use any identifying information
- Explain: "We are testing how effective different types of aiming assistance are. We are trying to determine which method gives the best performance increase in a FPS. You will be testing several different implementations, and each one will be played at multiple levels of assistance. There will also be multiple rounds you will play where you are not being assisted"
- Show the participant the webpage with the controls, http://hcigame.usask.ca/
- Any questions? More details about the implementations will have to wait until after the experiment
- Start the game. The first screen should prompt them to enter their information.

Warm-Up

- Explain: “Now we’ll play through the level a couple times to get you familiar with the controls. The goal is to kill the evil robots while making sure you don’t hit the friendly turtles. Keep in mind that this game was designed to be very difficult, so don’t be discouraged if you find yourself dying a lot. These first two training rounds will seem to be very hard, but after you will be assistance and will be more used to the game, so it’ll never be this bad again. This study is also pretty long, so if you’d like to take a break in between rounds just let me know.”
- Select “Instant Action” in the main menu
- Select “Custom Deathmatch” as the mode(Should be default) and “Shooting Gallery” as the map
- Once the first game has started: “Let’s run through the controls. You can move with the wasd keys, and use the mouse to aim. The left mouse button shoots. The mouse wheel is used to switch weapons. You have a pistol and an assault rifle in your inventory. R is to reload. You can sprint with shift and if it gets too dark use F to toggle your flashlight. Also, headshots do 2x damage. Looking at the HUD, the bottom left part of the screen indicates how much ammo you have left in your current clip, the next number over shows how much reserve ammo you have. The two bars below the ammo indicator show your health and sprint stamina.”
- Let them know they can change the sensitivity with the [and] keys, and change the volume with + and – while in game
- Have the participant play 1-2 times through the map until they are comfortable with the controls and layout of the map.

Experiment

- Select the “Aim Mode” option in the main menu
- Make sure “Shooting Gallery” is the selected map
- Select one of the aim modes 1-12– in the menu it is referred to as “Condition”
- If Condition 3 or 4, remind the participant the Q key(or the special function key on the mouse) is used to lock on. “Keep in mind, this targets friendlies as well as enemies”

- Let them play the game to the end
- Once the map has been finished, a post-game questionnaire will pop up. Have them fill this out: “Now I’ll have you fill out this questionnaire. At the bottom you can see more details about the question you’re currently on. The first option is a scale from 0 to 10. The rest range from -10 to 10“
- Ask the participant if they have any extra comments about this method or what they noticed – write them down
- Repeat until all 12 options have been played

8.3.3 S3 (Full Walkthrough 2)

Setup

- Have the FPS game installed on a computer. Most recent version is at https://dl.dropboxusercontent.com/u/4255072/UDKInstall-AimAssist.exe
- Make sure the logging server is up
- Have the webpage http://hcigame.usask.ca/ open, which contains information on controls
- Have notebook and pen ready to record observations from gameplay, important comments, and signs of frustration or fun.
- Delete " \UDK\AimAssistStudy\UDKGame\Config\UDKUDNGame.ini" so the user will be prompted for user details next time they start the game

Introduction

- Give consent form
  - "You will get $10 for participating, and are free to withdraw from the study at any time, no questions asked and you can still collect the money. Your data will also be removed from any analysis"
  - Collected data will be used in academic publications and other publicly available information: we will not use any identifying information
- Explain: "We are testing how effective different types of aiming assistance are. We are trying to determine which method gives the best performance increase in a FPS. You will be testing several different implementations, and there will also be multiple rounds you will play where you are not being assisted. "
- Show the participant the webpage with the controls, http://hcigame.usask.ca/
- Any questions? More details about the implementations will have to wait until after the experiment
- Start the game. The first screen should prompt them to enter their information.

Warm-Up

- Explain: “Now we’ll play through the two versions of the level a couple times to get you familiar with the controls. The goal is to kill the evil robots. In the second version, you will also have to make sure you don’t hit the friendly turtles. In this version you will also have a different gun, a sniper rifle. Keep in mind that this game was designed to be very difficult, so don’t be discouraged if you find yourself dying a lot. These first two training rounds will seem to be very hard, but after you will be assisted and will be more used to the game, so it’ll never be this bad again. This study is also pretty long, so if you’d like to take a break in between rounds just let me know.”
- Select “Instant Action” in the main menu
- Select “Custom Deathmatch” as the mode(Should be default) and “Shooting Gallery (No Friendly/Sniper)” as the map
- Once the first game has started: “Let’s run through the controls. You can move with the wasd keys, and use the mouse to aim. The left mouse button shoots. The mouse wheel is used to switch weapons. You have a pistol and an assault rifle in your inventory. R is to reload. You can sprint with shift and if it gets too dark use F to toggle your flashlight. Also, headshots do 2x damage. Looking at the HUD, the bottom left part of the screen indicates how much ammo you have left in your current clip, the next number over shows how much reserve ammo you have. The two bars below the ammo indicator show your health and sprint stamina.”
- Let them know they can change the sensitivity with the [and] keys, and change the volume with + and – while in game
- Have the participant play 1 time through the No Friendlies map and 1 time through the Sniper map until they are comfortable with the controls and layout of the map.

Experiment

- Select the “Aim Mode” option in the main menu
- Make sure “Shooting Gallery No Friendlies” is the selected map
- Select one of the aim modes 1-12– in the menu it is referred to as “Condition”
- Let them play the game to the end

- Once the map has been finished, a post-game questionnaire will pop up. Have them fill this out: “Now I’ll have you fill out this questionnaire. At the bottom you can see more details about the question you’re currently on. The first option is a scale from 0 to 10. The rest range from -10 to 10“
- Ask the participant if they have any extra comments about this method or what they noticed – write them down
- Repeat until all 5 options have been played
- Refresh the aim modes
- Repeat with the “Shooting Gallery Sniper” map

APPENDIX B

STUDY 2 FORMS

8.4 CONSENT FORM

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF SASKATCHEWAN
INFORMED CONSENT FORM

University of Saskatchewan

Research Project: **Game Balancing in First Person Shooters**

Investigators: Dr. Regan Mandryk, Department of Computer Science (966-4888)

Dr. Carl Gutwin, Department of Computer Science (966-8646)

Rodrigo Vicencio, Department of Computer Science

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

This study is concerned with **measuring game balancing performance in a 3d game with a first person perspective.**

The goal of the research is to **determine the effectiveness of different game balancing methods in increasing targeting performance and balancing gameplay in a 3d first person computer game.**

The session will require **45-60** minutes, during which you will be asked **play a 3d first person game against an opponent where the goal is to get a higher score by having more "kills"**. **You will be asked to fill out a questionnaire at the end of each round.**

At the end of the session, you will be given more information about the purpose and goals of the study, and there will be time for you to ask questions about the research. As a way of thanking you for your participation and to help compensate you for your time and any travel costs you may have incurred, you will receive **$10** at the end of the session.

The data collected from this study will be used in articles for publication in journals and conference proceedings.

As one way of thanking you for your time, we will be pleased to make available to you a summary of the results of this study once they have been compiled (usually within two months). This summary will outline the research and discuss our findings and recommendations. This summary will be available on the HCI lab's website: **http://www.hci.usask.ca/**

All personal and identifying data will be kept confidential. If explicit consent has been given, textual excerpts, photographs, or video recordings may be used in the dissemination of research results in scholarly journals or at scholarly conferences. Confidentiality will be preserved by using pseudonyms in any presentation of textual data in journals or at conferences. The informed consent form and all research data will be kept in a secure location under confidentiality in accordance with University policy for 5 years post publication. Do you have any questions about this aspect of the study?

You are free to withdraw from the study at any time without penalty and without losing any advertised benefits. Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed. Your right to withdraw data from the study will apply until results have been disseminated, data has been pooled, etc. After this, it is possible that some form of research dissemination will have already occurred and it may not be possible to withdraw your data.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact:

- Dr. Regan Mandryk, Associate Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. If you have further questions about this study or your rights as a participant, please contact:

- Dr. Regan Mandryk, Associate Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca
- Research Ethics Office, University of Saskatchewan, (306) 966-2975 or toll free at 888-966-2975.

Participant's signature:____

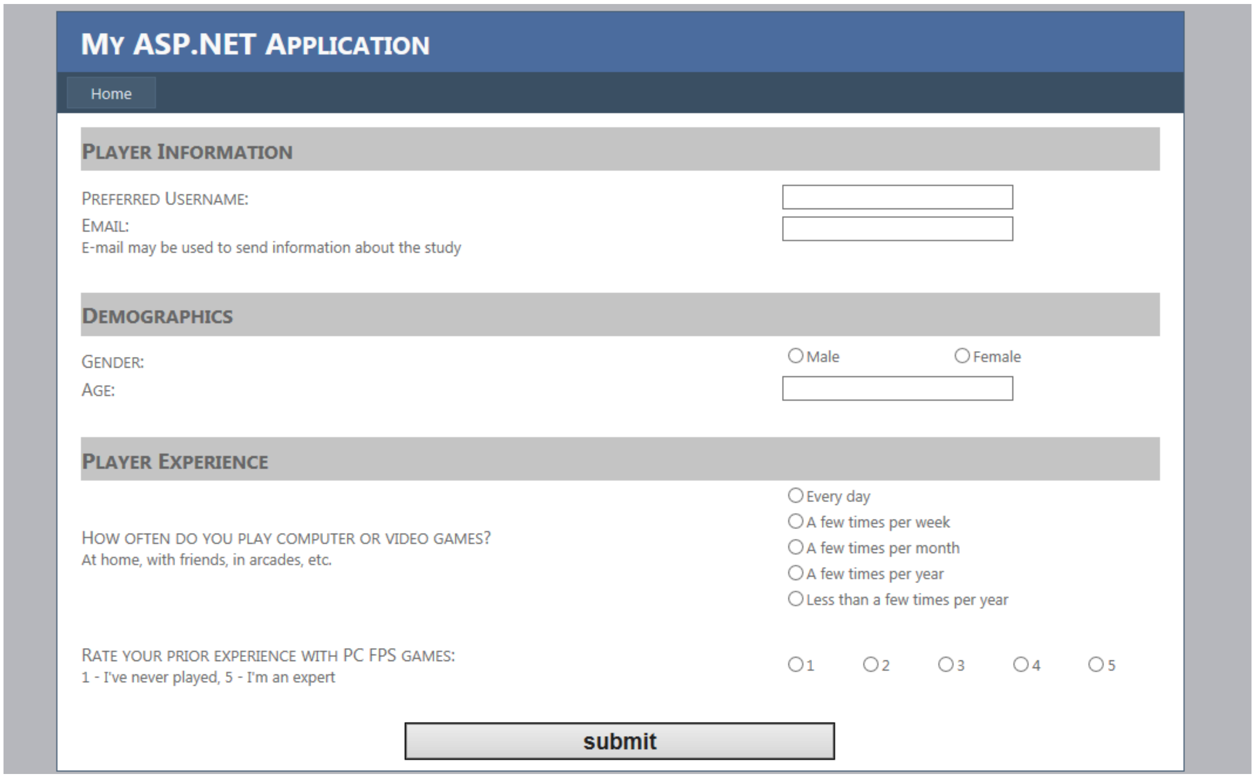
Date:______________________

Investigator's signature:____

Date:______________________

A copy of this consent form has been given to you to keep for your records and reference. This research has the ethical approval of the Research Ethics Office at the University of Saskatchewan.

8.5 DEMOGRAPHIC FORM



8.6 POST GAME QUESTIONNAIRE

USERNAME:
Ensure this is correct. If it is not, alert your study coordinator.

QUESTION SET 1

WHO WAS THE STRONGER PLAYER? ○ Me ○ My Partner

WHAT LEVEL OF ASSISTANCE WAS PROVIDED TO ME? [0 ▾]
0 = None, My performance ws entirely due to my skill,
10 = A lot, my performance/all my kills was due to the help I was given

WHAT LEVEL OF ASSISTANCE WAS PROVIDED TO MY PARTNER? [0 ▾]
0 = None, Their performance was entirely due to their skill,
10 = A lot, their performance/all their kills was due to the help they were given

IN GENERAL, DID YOU FIND THIS ROUND TO BE FAIR? ○ Fair ○ Not Fair

QUESTION SET 2

FOR THE FOLLOWING QUESTIONS: 1=STRONGLY DISAGREE, 2=DISAGREE, 3=NEUTRAL, 4=AGREE, 5=STRONGLY AGREE.

I FELT COMPETENT IN THIS ROUND:	○1 ○2 ○3 ○4 ○5
I FELT VERY CAPABLE AND EFFECTIVE WHEN PLAYING:	○1 ○2 ○3 ○4 ○5
MY ABILITY TO PLAY THIS ROUND IS WELL MATCHED WITH THIS ROUND'S CHALLENGES:	○1 ○2 ○3 ○4 ○5

QUESTION SET 3

FOR THE FOLLOWING QUESTIONS: 1=STRONGLY DISAGREE, 2=DISAGREE, 3=NEUTRAL, 4=AGREE, 5=STRONGLY AGREE.

I FOUND THE RELATIONSHIPS I FORMED IN THIS ROUND FULFILLING:	○1 ○2 ○3 ○4 ○5
I FOUND THE RELATIONSHIPS I FORMED IN THIS ROUND IMPORTANT:	○1 ○2 ○3 ○4 ○5
I DIDN'T FEEL CLOSE TO THE OTHER PLAYER(S):	○1 ○2 ○3 ○4 ○5

QUESTION SET 4

FOR THE FOLLOWING QUESTIONS: 1=STRONGLY DISAGREE, 2=DISAGREE, 3=NEUTRAL, 4=AGREE, 5=STRONGLY AGREE.

I ENJOYED THIS ROUND VERY MUCH:	○1 ○2 ○3 ○4 ○5
PLAYING THIS ROUND WAS FUN:	○1 ○2 ○3 ○4 ○5
I WOULD DESCRIBE THIS ROUND AS VERY INTERESTING:	○1 ○2 ○3 ○4 ○5
WHILE PLAYING THIS ROUND, I WAS THINKING ABOUT HOW MUCH I ENJOYED IT:	○1 ○2 ○3 ○4 ○5
THIS ROUND DID NOT HOLD MY ATTENTION:	○1 ○2 ○3 ○4 ○5

QUESTION SET 5

FOR THE FOLLOWING QUESTIONS: 1=STRONGLY DISAGREE, 2=DISAGREE, 3=NEUTRAL, 4=AGREE, 5=STRONGLY AGREE.

THIS ROUND PROVIDED ME WITH INTERESTING OPTIONS AND CHOICES:	○1 ○2 ○3 ○4 ○5
THIS ROUND LETS YOU DO INTERESTING THINGS:	○1 ○2 ○3 ○4 ○5
I EXPERIENCED A LOT OF FREEDOM IN THE ROUND:	○1 ○2 ○3 ○4 ○5

[submit]

APPENDIX C

STUDY 3 FORMS

8.7 CONSENT FORM

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF SASKATCHEWAN
INFORMED CONSENT FORM

Research Project:	**Game Balancing in First Person Shooters**
Investigators:	Dr. Regan Mandryk, Department of Computer Science (966-4888)
	Dr. Carl Gutwin, Department of Computer Science (966-8646)
	Rodrigo Vicencio, Department of Computer Science

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

This study is concerned with **measuring game balancing performance in a 3d game with a first person perspective.**

The goal of the research is to **determine the effectiveness of different game balancing methods in increasing targeting performance and balancing gameplay in a 3d first person computer game.**

The session will require **45-60** minutes, during which you will be asked **play a 3d first person game against an opponent where the goal is to get a higher score by having more "kills"**. **You will be asked to fill out a questionnaire at the end of each round.**

At the end of the session, you will be given more information about the purpose and goals of the study, and there will be time for you to ask questions about the research. As a way of thanking you for your participation and to help compensate you for your time and any travel costs you may have incurred, you will receive **$10** at the end of the session.

The data collected from this study will be used in articles for publication in journals and conference proceedings.

As one way of thanking you for your time, we will be pleased to make available to you a summary of the results of this study once they have been compiled (usually within two months). This summary will outline the research and discuss our findings and recommendations. This summary will be available on the HCI lab's website: **http://www.hci.usask.ca/**

All personal and identifying data will be kept confidential. If explicit consent has been given, textual excerpts, photographs, or video recordings may be used in the dissemination of research results in scholarly journals or at scholarly conferences. Confidentiality will be preserved by using pseudonyms in any presentation of textual data in journals or at conferences. The informed consent form and all research data will be kept in a secure location under confidentiality in accordance with University policy for 5 years post publication. Do you have any questions about this aspect of the study?

You are free to withdraw from the study at any time without penalty and without losing any advertised benefits. Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed. Your right to withdraw data from the study will apply until results have been disseminated, data has been pooled, etc. After this, it is possible that some form of research dissemination will have already occurred and it may not be possible to withdraw your data.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact:

- Dr. Regan Mandryk, Associate Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. If you have further questions about this study or your rights as a participant, please contact:

- Dr. Regan Mandryk, Associate Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca
- Research Ethics Office, University of Saskatchewan, (306) 966-2975 or toll free at 888-966-2975.

Participant's signature:____

Date:______________________

Investigator's signature:____

Date:______________________

A copy of this consent form has been given to you to keep for your records and reference. This research has the ethical approval of the Research Ethics Office at the University of Saskatchewan.

8.8 DEMOGRAPHIC FORM

FPS STUDY

Home

PLAYER INFORMATION

PREFERRED USERNAME: [ ]

EMAIL: [ ]
Make sure this field is correct

DEMOGRAPHICS

GENDER: ○ Male ○ Female

AGE: [ ]

PLAYER EXPERIENCE

WHICH SYSTEM DO YOU PRIMARILY PLAY VIDEO GAMES ON:
At home, with friends, etc.

- ○ Console (Xbox, playstation, nintendo, etc)
- ○ Computer/PC

HOW OFTEN DO YOU PLAY VIDEO GAMES ON CONSOLES OR COMPUTER?
At home, with friends, etc.

- ○ Every day
- ○ A few times per week
- ○ A few times per month
- ○ A few times per year
- ○ Less than a few times per year

HOW OFTEN DO YOU PLAY COMPUTER (PC) GAMES?
At home, with friends, etc.

- ○ Every day
- ○ A few times per week
- ○ A few times per month
- ○ A few times per year
- ○ Less than a few times per year

RATE YOUR PRIOR EXPERIENCE USINNG KEYBOARD + MOUSE:
1 - I've never used them, 7 - I'm an expert

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

RATE YOUR EXPERIENCE WITH PC FIRST PERSON SHOOTER GAMES:
1 - I've never played, 7 - I'm an expert

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

HOW MANY HOURS PER WEEK DO YOU PLAY PC FPS GAMES? [ ]

WHICH FPS GAMES DO YOU TYPICALLY PLAY?

- ☐ Call of Duty
- ☐ Counter Strike
- ☐ Battlefield
- ☐ Borderlands
- ☐ Other

If Other, please specify: [ ]

submit

8.9 POST GAME QUESTIONNAIRE

Username:
Ensure this is correct. If it is not, alert your study coordinator.

Question Set 1

Who was the stronger player?	○ Me ○ My Partner
What level of assistance was provided to me? 0 = None, My performance ws entirely due to my skill, 10 = A lot, my performance/all my kills was due to the help I was given	0 ⌄
What level of assistance was provided to my partner? 0 = None, Their performance was entirely due to their skill, 10 = A lot, their performance/all their kills was due to the help they were given	0 ⌄
In general, did you find this round to be fair?	○ Fair ○ Not Fair

Question Set 2

For the following questions: 1=strongly disagree, 2=disagree, 3=neutral, 4=agree, 5=strongly agree.

I felt competent in this round:	○1 ○2 ○3 ○4 ○5
I felt very capable and effective when playing:	○1 ○2 ○3 ○4 ○5
My ability to play this round is well matched with this round's challenges:	○1 ○2 ○3 ○4 ○5

Question Set 3

For the following questions: 1=strongly disagree, 2=disagree, 3=neutral, 4=agree, 5=strongly agree.

I found the relationships I formed in this round fulfilling:	○1 ○2 ○3 ○4 ○5
I found the relationships I formed in this round important:	○1 ○2 ○3 ○4 ○5
I didn't feel close to the other player(s):	○1 ○2 ○3 ○4 ○5

Question Set 4

For the following questions: 1=strongly disagree, 2=disagree, 3=neutral, 4=agree, 5=strongly agree.

I enjoyed this round very much:	○1 ○2 ○3 ○4 ○5
Playing this round was fun:	○1 ○2 ○3 ○4 ○5
I would describe this round as very interesting:	○1 ○2 ○3 ○4 ○5
While playing this round, I was thinking about how much I enjoyed it:	○1 ○2 ○3 ○4 ○5
This round did not hold my attention:	○1 ○2 ○3 ○4 ○5

Question Set 5

For the following questions: 1=strongly disagree, 2=disagree, 3=neutral, 4=agree, 5=strongly agree.

This round provided me with interesting options and choices:	○1 ○2 ○3 ○4 ○5
This round lets you do interesting things:	○1 ○2 ○3 ○4 ○5
I experienced a lot of freedom in the round:	○1 ○2 ○3 ○4 ○5

submit

8.10 EXPERIMENT SCRIPT

Setup

- Have the FPS game installed on three computers. Most recent version is at http://hcigame.usask.ca/Updates/AimAssist.exe
- For researcher computer:
    - Remote into server (start -> type “mstsc”)
        - Computer: **HCIGame**
    - Start multiplayer server. Use the MultiplayerLauncher in C:\AimAssist\Binaries.
        - Map Cycle: **CDMM-Deck2** AdminPassword: **password** GameType: **UDNGame.UDNGame** NumBots: **0** TimeLimit: **5(10)**. Press “Start Server”
    - Open “SQL Server Management Studio” to ensure logs are being written successfully.
        - ServerName: **localhost\SQLEXPRESS** Authentication: **Windows**. Hit connect, open the required tables under Databases > UDKLogDevMulti2 > Tables. Remember you’ll need to refresh to see new data
- For both participant computers:
    - Have the webpage http://hcigame.usask.ca/ open, which contains information on controls (optional)
    - Delete “ C:\AimAssist\UDKGame\Config\UDKUDNGame.ini” so the user will be prompted for user details next time they start the game
    - Have the game open, on the login screen, set resolution to **1680x1050**
- Have notebook and pen ready to record observations from gameplay, important comments, and signs of frustration or fun.

Introduction

- Give consent form

    - “You will get $10 for participating, and are free to withdraw from the study at any time, no questions asked and you can still collect the money. Your data will also be removed from any analysis. Collected data will be used in academic publications and other publicly available information: we will not use any identifying information”
- Explain: “You will be testing how effective different types of gameplay balancing are. We are trying to determine which method is the best. You will be playing six rounds together, some with a game balancing algorithm and some without, each round lasts 5 minutes, and at the end of each round there will be a questionnaire to fill out.“
- Any questions? More details about the implementations will have to wait until after the experiment
- Start the game. The first screen should prompt them to enter their information. Have them login, and set the appropriate resolution.

Warm-Up

- Explain: “This first round is the training round, just to get you familiar with the game ”
- Have both participants select “Multiplayer” in the main menu (assuming server is up)
- Once the first game has started, for each participant: “Let’s run through the controls. You can move with the wasd keys, and use the mouse to aim. The left mouse button shoots. The mouse wheel is used to switch weapons. You have a pistol and an assault rifle in your inventory, and there are two spots on the map you can pick up a sniper rifle. R is to reload. Looking at the HUD, the bottom left part of the screen indicates how much ammo you have left in your current clip, the next number over shows how much reserve ammo you have. The two bars below the ammo indicator show your health and sprint stamina. You can sprint with shift . Also, headshots do 2x damage.” Let them know they can change the sensitivity with the [and] keys
- “Now we’re going to play the real game. Don’t feel bad for killing/dying, be ruthless”

Experiment

- Have both participants select “Multiplayer” in the main menu

- Enter the server as a spectator. Press ~ then type:
    - **open hcigame.usask.ca?spectatoronly=1**
- Change the aim mode (if needed) -> Bullet = 2, Area = 3, Delay = 2 delay. Press ~ then type:
    - **AdminLogin**
    - **AdminSwitchAimMode (0/2/3/2 delay)**
    - For “Other” do instead: **AdminToggleOtherBalancing**
- Once the map has been finished, a post-game questionnaire will pop up. Have them fill this out: “Now I’ll have you fill out this questionnaire. In the second section onward, answer the questions on the 1-5 scale as described“
- **If Player Not Found**, have them fill out the form and take screenshots. Make sure you save the AimLogBackups (see post experiment)
- **If you get spammed with post-game questionnaire popups**, try to close the server and game
- Ask the participant if they have any extra comments about this method or what they noticed – write them down
- Repeat until all 5 matches + training have been played

Post Experiment

- Check “C:\AimAssist\Binaries\Win32\UserCode\Logs” for ErrorLog.txt.
- Save newly written files from “C:\AsimAssist\AimLogBackups\”

APPENDIX D

MEGA ROBOT SHOOTOUT VIDEO LINKS

8.11 YOUTUBE LINKS

Mega Robot Friendship playlist: https://youtu.be/ol6mvABeDh0?list=PLvMyOVzYnDtv57m-Pnozahzo-J_c1Aycn

CHI 2014 Video: https://youtu.be/ol6mvABeDh0

CHI 2015 Video: https://youtu.be/4045gjzGfXc

Mega Robot Shootout Gameplay: https://youtu.be/6CQxdNU8RBk

APPENDIX E

AIMING ASSISTANCE CODE

8.12 UDNPLAYERCONTROLLER.UC

```
/**************************************
 **************************************
        Level System
 **************************************
 **************************************/

reliable server function ClientSetLevel(int L)
{
    Level = L;
}

reliable server function DynamicChangeLevel()
{
    local Pawn P;
    local int HighestScore, previousLevel;

    previousLevel = Level;
    if(UDNGameSingle(WorldInfo.Game) != None)
    {
        Level = UDNGameSingle(WorldInfo.Game).SinglePlayerAimLevel;
        if(Level > 10) Level = 10;
        else if(Level < 0) Level = 0;
        ClientSetLevel(Level);
    }
    else if(bUseStaticLevelling)
    {
        HighestScore = -1;
        foreach WorldInfo.AllPawns(class'Pawn', P)
            if(P != None && P.PlayerReplicationInfo != None && P != Pawn && P.PlayerReplicationInfo.Kills >  HighestScore)
                HighestScore = P.PlayerReplicationInfo.Kills;
        if(PlayerReplicationInfo == None) Level = 0;
        else if(PlayerReplicationInfo.Kills >= HighestScore) Level = 0;
        else Level = 8;

        if(Level > 10) Level = 10;
        else if(Level < 0) Level = 0;
        ClientSetLevel(Level);
    }
    else if(bUseDynamicLevelling)
    {
        HighestScore = -1;
        foreach WorldInfo.AllPawns(class'Pawn', P)
            if(P != None && P.PlayerReplicationInfo != None && P != Pawn && P.PlayerReplicationInfo.Kills >  HighestScore)
                HighestScore = P.PlayerReplicationInfo.Kills;
        if(PlayerReplicationInfo == None) Level = 0;
        else if(PlayerReplicationInfo.Kills >= HighestScore) Level = 0;
        else Level = (HighestScore - PlayerReplicationInfo.Kills);

        if(Level > 10) Level = 10;
        else if(Level < 0) Level = 0;

        if(bUseDelay && previousLevel > 0 && Level == 0)
        {
            if(startDelayTime == 0) startDelayTime = WorldInfo.RealTimeSeconds;
            if(WorldInfo.RealTimeSeconds - startDelayTIme >= 30){
                Level = 0;
                startDelayTime = 0;
            }
            else Level = 1;
        }

        ClientSetLevel(Level);
    }
}
```

```
/***************************************
 ***************************************
       Aim Mode Switching
 ***************************************
***************************************/
exec function SwitchAimMode(int AimModeSwitch)
{
    if(WorldInfo.NetMode == NM_Standalone && CanSwitchAimMode)
    {
        Self.AimMode = AimModeSwitch;
        if(UDNGame(WorldInfo.Game) != None) UDNGame(WorldInfo.Game).GlobalAimMode = AimModeSwitch;
        LastTarget = None;
    }
    else ClientMessage("Unable to switch aim mode");
}

reliable client function ClientSwitchAimMode(int AimModeSwitch)
{
    Self.AimMode = AimModeSwitch;
    LastTarget = None;
}

reliable server function ServerSwitchAimMode(int AimModeSwitch, string args)
{
    Self.AimMode = AimModeSwitch;
    LastTarget = None;
    startDelayTime = 0;
    if(args == "delay") bUseDelay = true;
    else bUseDelay = false;
    ClientSwitchAimMode(AimModeSwitch);
}

/***************************************
 ***************************************
       Helpers
 ***************************************
***************************************/

function float GetYaw(Vector from, vector to)
{
    local float degrees;
    degrees = Atan2(to.Y - from.Y, to.X - from.X)*(180/PI);
    Debug = Debug$"Yaw - Degrees"@degrees@"Adjusted"@(degrees*(8192/45)$"\n");
    return degrees * (8192/45);///PI*180+180;
}

function float GetPitch(Vector from, vector to)
{
    local float degrees, dist;
    dist = VSize(from - to);
    degrees = Asin((to.Z - from.Z)/dist)*(180/PI);
    Debug = Debug$"Pitch - Degrees"@degrees@"Adjusted"@(degrees*(8192/45));
    return degrees * (8192/45);///PI*180+180;
}

function OutputIntersect()
{
    local Pawn p;
    local vector ret;
    Debug $= "\nCursor Size:"@CrosshairSize;;
    ret.X=CrosshairSize; ret.Y=CrosshairSize; ret.Z=CrosshairSize;
    if(Pawn != None && UDNWeapon(Pawn.Weapon) != None) p = UDNWeapon(Pawn.Weapon).PlayerInCrosshair(ret);
    if(p != None) Debug = Debug@"You are on:"@p.Name;
}

exec function ChangeTargetSelectionMode(int newSM)
{
    TargetSelectionMode = newSM;
}
```

```
simulated function Pawn GetBestTarget(bool LOSOnly)
{
    local UTPawn P;
    local Pawn tTarget;
    local float bestDist, tDist;
    local vector StartTrace, To;

    tTarget = None;
    bestDist = 99999999.0;
    Debug $= "These are the visible players:";

    if(TargetSelectionMode == 1)
    {
        StartTrace = UTWeapon(self.Pawn.Weapon).InstantFireStartTrace();
        To = UTWeapon(self.Pawn.Weapon).InstantFireEndTrace(StartTrace);
    }
    else
    {
        if(LastTarget != NONE && LastTarget.IsAliveAndWell() && CanSee(LastTarget))
            return LastTarget;
    }

    foreach WorldInfo.AllPawns(class'UTPawn', P)
    {
        if(P != NONE && P.IsAliveAndWell() && Self.Pawn != P && !P.IsSameTeam(Self.Pawn))
        {
            if(LOSOnly && CanSee(P))//********not completely LOS -> fix MyIsVisible
            {
                if(TargetSelectionMode == 1) tDist = PointDistToLine(P.Location, To - StartTrace, StartTrace);
                else tDist = VSize(P.Location - Self.Pawn.Location);
                //The VSize() function is used to get the distance from the player's position to the Controller's position.
                Debug = Debug@P.GetHumanReadableName()@" Dist:"@tDist$",";
                if(Tdist < 0) ClientMessage("Uh oh tdist shouldnt be smaller than 0************************");
                if(tDist < bestDist)
                {
                    tTarget = P;
                    bestDist = tDist;
                }
            }
            else if(!LOSOnly && FastTrace(Self.Pawn.Location, P.Location, , true))
            {
                if(TargetSelectionMode == 1) tDist = PointDistToLine(P.Location, To - StartTrace, StartTrace);
                else tDist = VSize(P.Location - Self.Pawn.Location);
                //The VSize() function is used to get the distance from the player's position to the Controller's position.
                Debug = Debug@P.GetHumanReadableName()@" Dist:"@tDist$",";
                if(Tdist < 0) ClientMessage("Uh oh tdist shouldnt be smaller than 0************************");
                if(tDist < bestDist)
                {
                    tTarget = P;
                    bestDist = tDist;
                }
            }
        }
    }
    LastTarget = tTarget;
    return LastTarget;
}
```

```
/****************************************
 ***************************************
        Aim Mode Implementation
 ***************************************
****************************************/

function UpdateRotation( float DeltaTime )
{
    local int tempLevel;
    //rotation unit -> Unreal rotation units 4096 each. [0, 65536] 65536 = 0 = 360
    SetCrosshairSize(10);
    DynamicChangeLevel();
    if(AimMode == AM_LOCK) Lock(DeltaTime);
    else if(AimMode == AM_AREACURSOR)
    {
        tempLevel = Level;
        if(Level > 0)
        {
            tempLevel = tempLevel + 6;
            if(bMegaDoubleLevelling) tempLevel = (level*2) + 6;
        }
        SetCrosshairSize(10 + (10 * tempLevel));/**40**/ if(ShowDebug) OutputIntersect(); super.UpdateRotation(DeltaTime);
    }
    else if(AimMode == AM_STICKY) CalculateSticky(DeltaTime);
    else if(AimMode == AM_GRAVITY) CalculateGravity(DeltaTime);
    else super.UpdateRotation(DeltaTime);
}

reliable server function SetCrosshairSize(int CSize)
{
    CrosshairSize = CSize;
}

function Vector CalculateAreaCursor()
{
    local vector ret;
    ret = vect(1,1,0);
    if(AimMode == AM_AREACURSOR && Level > 0)
    {
        ret.X=CrosshairSize;
        ret.Y=CrosshairSize;
        ret.Z=CrosshairSize;
    }
    return ret;
}

function CalculateSticky(float DeltaTime)
{
    local Pawn p;
    local int Sticky;
    local vector extent;
    Sticky = (Level == 0 ? 1 : (Level + 4));
    if(UDNWeapon(Pawn.Weapon) != None)
    {
        extent.X=0; extent.Y=0; extent.Z=0;
        p = UDNWeapon(Pawn.Weapon).PlayerInCrosshair(extent);
        if(p != None)
        {
            Debug $= "You are on:"@p.Name$"\n";
            PlayerInput.aTurn /= Sticky;
            PlayerInput.aLookUp /= Sticky;
        }
    }
    Super.UpdateRotation(DeltaTime);
}
```

```
function Lock(float DeltaTime)
{
    Local Pawn target;
    local Rotator newRot;
    local float pitch, yaw;
    local int AimSpeedFactor;

    newRot = Rotation;//get the rot to be from 0 to 360
    if(newRot.Yaw > 65536)
        while(newRot.Yaw > 65536) newRot.Yaw -= 65536;
    else if(newRot.Yaw < 0)
        while(newRot.Yaw < 0) newRot.Yaw += 65536;
    SetRotation(newRot);
    if(AimMode != AM_LOCK || Pawn == None || Self.Pawn.Health <= 0 || bLock != 1)//IsAliveAndWell?
    {
            Super.UpdateRotation(DeltaTime);
            return;
    }
    target = GetBestTarget(true);
    if(target != None)
    {
        AimSpeedFactor = 10 - Level + (Level == 0 ? 15 : 3);//10 && : was 1
        yaw = GetYaw(Pawn.Location, target.Location);
        pitch = GetPitch(Pawn.Location, target.Location);//target.Pawn.Location

        if(yaw < 0) yaw += 65536;
        if(yaw  > 65536) yaw -= 65536;
        //Imagine a circle, T is the angle from current rotation left to target, t is the angle right.
        //If the left angle is smaller, go left.

        //handle yaw
        //first if: get distance going left, second: get distance going right
        if(newRot.Yaw - (yaw > newRot.Yaw ? yaw - 65536 : yaw) < (yaw < newRot.Yaw ? yaw + 65536 : yaw) - newRot.Yaw)
            PlayerInput.aTurn  = -((newRot.Yaw - (yaw > newRot.Yaw ? yaw - 65536 : yaw)) / AimSpeedFactor);
        else PlayerInput.aTurn  = (((yaw < newRot.Yaw ? yaw + 65536 : yaw) - newRot.Yaw) / AimSpeedFactor);

        //handle pitch
        if(newRot.Pitch > Pawn.ViewPitchMax) { newRot.Pitch -= 65536; SetRotation(newRot); }
        if(newRot.Pitch > pitch) PlayerInput.aLookUp  = -((Rotation.Pitch - pitch) / AimSpeedFactor);
        else if(Rotation.Pitch < pitch) PlayerInput.aLookUp  = ((pitch - Rotation.Pitch) / AimSpeedFactor);

        Super.UpdateRotation(DeltaTime);
        //snap on if close enough
        if(abs(abs(yaw) - abs(Rotation.Yaw)) < AIMLOCK && (abs(abs(pitch) - abs(Rotation.Pitch)) < AIMLOCK ||
            abs(abs(pitch + 65536) - abs(Rotation.Pitch)) < AIMLOCK))
        { newRot.Yaw = yaw; newRot.Pitch = pitch; SetRotation(newRot); }
    }
    else
        Super.UpdateRotation(DeltaTime);
}
```

```
simulated function Vector CalculateBulletMagnetism(vector To)
{
    local int tempLevel;
    local vector loc, ClosestPoint;
    Local Pawn target;
    local float AimMagFactor, AimHelpDist, TargetDist, BulletMagEffectRadius, PerfectAimAllowance;

    if(AimMode == AM_BULLETMAG)
    {
        tempLevel = (level == 0 ? 0 : level + 7);
        if(bMegaDoubleLevelling) tempLevel = (level == 0 ? 0 : (level*2) + 7);
        //higher means the effect starts farther away
        BulletMagEffectRadius = (100 * tempLevel) + (tempLevel == 0 ? 0 : 100);
        //higher means you have to be this close for perfect aim -> dont have to be as precise
        PerfectAimAllowance = (12 * tempLevel) + (tempLevel == 0 ? 0 : 0);
        target = GetBestTarget(true);

        if(target != None)
        {
            loc = target.Location;
            //if intersects or higher than eyeheight + Z
            if(loc.Z + target.EyeHeight <= To.Z || UDNWeapon(Pawn.Weapon).PlayerInCrosshair(vect(1,1,0)) != None)
                loc.Z += target.EyeHeight;
            //if under head and above feet, then use the same height
            else if(loc.Z - (target.CylinderComponent.CollisionHeight / 2) <= To.Z) loc.Z = To.Z;

            //decide if it's too far to bother
            TargetDist = PointDistToLine(loc, To - Pawn.Location, Pawn.Location, ClosestPoint);
            AimHelpDist = BulletMagEffectRadius * class'UTProjectile'.Default.GlobalCheckRadiusTweak;

            if ( (abs(ClosestPoint.Z - loc.Z) < target.CylinderComponent.CollisionHeight + AimHelpDist) &&
                (VSize2D(ClosestPoint - loc) < target.CylinderComponent.CollisionRadius + AimHelpDist) )
            {
                AimMagFactor = 1 - ((FClamp(TargetDist - PerfectAimAllowance, 0, TargetDist))/BulletMagEffectRadius);
                if(AimMagFactor <= 0) AimMagFactor = 0;
                To = ((loc - To) * AimMagFactor) + To;
            }
        }
    }
    return To;
}
```

```
function CalculateGravity(float DeltaTime)
{
    local UTPawn P;
    local vector Top, Bottom, TempPos;
    local int AimSpeedFactor;
    local float w, pitch, yaw;
    local Rotator newRot;
    if(AimMode != AM_Gravity) return;
    if(PlayerInput.aTurn == 0 && PlayerInput.aLookUp == 0) return;

    AimSpeedFactor = (9 - Level) + 8;
    Gravity = Level;
    newRot = Rotation;//get the rot to be from 0 to 360
    if(newRot.Yaw > 65536)
        while(newRot.Yaw > 65536) newRot.Yaw -= 65536;
    else if(newRot.Yaw < 0)
        while(newRot.Yaw < 0) newRot.Yaw += 65536;
    SetRotation(newRot);

    foreach WorldInfo.AllPawns(class'UTPawn', P)
    {
        if(P != NONE && P.IsAliveAndWell() && Self.Pawn != P && !P.IsSameTeam(Self.Pawn) && CanSee(P))
        {
            w = (Gravity / (abs((Pawn.Location.X - P.Location.X) ** 2.0f) + 1));
            Top.X += (w * P.Location.X);
            Bottom.X += w;
            w = (Gravity / (abs((Pawn.Location.Y - P.Location.Y) ** 2.0f) + 1));
            Top.Y += (w * P.Location.Y);
            Bottom.Y += w;
            w = (Gravity / (abs((Pawn.Location.Z - P.Location.Z) ** 2.0f) + 1));
            Top.Z += (w * P.Location.Z);
            Bottom.Z += w;
        }
    }
    //If already aiming at the head, no need to apply gravity
    if(Bottom.X != 0 && Bottom.Y != 0 && Bottom.Z != 0)
    {
        TempPos.X = Top.X / Bottom.X; TempPos.Y = Top.Y / Bottom.Y; TempPos.Z = Top.Z / Bottom.Z;
        yaw = GetYaw(Pawn.Location, TempPos);
        pitch = GetPitch(Pawn.Location, TempPos);

        if(yaw < 0) yaw += 65536;
        if(yaw  > 65536) yaw -= 65536;

        //handle yaw //first part: get distance going left, second: get distance going right
        if(newRot.Yaw - (yaw > newRot.Yaw ? yaw - 65536 : yaw) <
            (yaw < newRot.Yaw ? yaw + 65536 : yaw) - newRot.Yaw && PlayerInput.aTurn < 0)
            PlayerInput.aTurn  -= ((newRot.Yaw - (yaw > newRot.Yaw ? yaw - 65536 : yaw)) / AimSpeedFactor);
        else if(PlayerInput.aTurn > 0)
        {
            if(yaw < newRot.Yaw) PlayerInput.aTurn  += 0;
            else PlayerInput.aTurn  += ((yaw - newRot.Yaw) / AimSpeedFactor);
        }

        //handle pitch
        if(newRot.Pitch > Pawn.ViewPitchMax) { newRot.Pitch -= 65536; SetRotation(newRot); }
        if(newRot.Pitch > pitch) PlayerInput.aLookUp  -= ((Rotation.Pitch - pitch) / AimSpeedFactor);
        else if(Rotation.Pitch < pitch) PlayerInput.aLookUp  += ((pitch - Rotation.Pitch) / AimSpeedFactor);
    }
    Super.UpdateRotation(DeltaTime);
}
```