Using Aim Assistance to Balance First Person Shooters

Rodrigo Vicencio-Moriera¹, Regan L. Mandryk¹, Carl Gutwin¹, Scott Bateman²

¹University of Saskatchewan
Saskatoon, SK, S7N 5C9
firstname.lastname@usask.ca

²University of Prince Edward Island
Charlottetown, PEI, C1A 4P3
sbateman@upei.ca

ABSTRACT
Previous work has used targeting assistance as a game balancing mechanic in a 2D environment successfully, but little information is available about how well game balancing with targeting assistance would work in 3D games, such as first person shooter (FPS) games. We tested the effectiveness of five aim assistance techniques in three studies and found that the Bullet Magnetism and Area Cursor showed the most promise for 3D FPS game balancing. Based on these results, we propose an environment and method to show that games that have aiming assistance will result in closer scores and better performance than those without, and that these closer scores will translate into a better player experience. This project will be the first to investigate the use of targeting assistance to balance competitive gameplay in an FPS.

Keywords
Aim assistance, first person shooter games, game balancing

1. INTRODUCTION
Video games are a multi-billion dollar industry, and First Person Shooter (FPS) games are one of the biggest genres, making up 21% of total sales [6]. FPS games require a lot of skill and quick thinking to navigate through 3D worlds while dodging enemy fire and firing back. These complex skills lead to fierce competition in FPS because they drive people to test their skills to be the best. However, competition brings in complications for players of different skill levels who wish to play together. When an expert plays a match against a novice, the game becomes imbalanced, and when games are imbalanced, players do not have fun [15]. The player with more skill (the expert player) gets bored due to the lack of challenge, and the person with less skill (the novice) gets frustrated because they can’t perform as well. This skill imbalance also affects the game’s demographics since it limits who can play the game and enjoy it. When novices lack the skills to compete against experts, they may feel alienated and will have no desire to play the game, leading to fewer sales.

One solution is to use aiming assistance to help weaker players. This approach has proven successful in 2D shooters [3]; however, no work has been done to see how well this approach would work in 3D. Recently, targeting assistance in a 3D environment has been tested [16], and while results may not be as clear cut as in 2D situations, the use of 3D targeting assistance looks promising.

To investigate how effective targeting assistance is at balancing gameplay in first person shooters, we developed an FPS game using the Unreal Development Kit. First, we conducted a series of three studies to test the effectiveness of targeting assistance in a 3D game environment. Next, we propose to study whether including aiming assistance in 2-player matches results in more balanced games. Our goal is to show that targeting assistance can be used in an FPS to balance gameplay, thus increasing enjoyment and maintaining fairness, and as a result, giving game developers tools to expand games to a wider audience to increase sales.

2. BACKGROUND
2.1 Game Balancing
A player’s willingness to play a game depends on their enjoyment of it [15]. Unbalanced games lead to less enjoyable experiences due to players becoming frustrated or unengaged because the game is either too difficult or too easy. Therefore it is important for game developers to thoroughly play test to ensure the game is balanced. Competition is part of the appeal of video games, but in competitive scenarios with other human players it is difficult to keep the game balanced if the players do not have equal skill.

A game is considered balanced when the challenge of the game matches the skill of the player [3]. A game that is too hard or too easy is not fun for the player. With this definition in mind, keeping a multiplayer game balanced means that all players should have an equal opportunity to win [1]. When the players are evenly matched, enjoyment is increased for both parties [3].

This game balancing issue has resulted in work aimed at trying to balance competitive multiplayer games. The general approaches game developers and researchers have developed to solve the gameplay balance problem can be put into three general categories:

1. Difficulty Adjustment: This is one of the most common ways of balancing which adjusts the level of challenge in the game. This type of balancing can be either static or dynamic. The static approach allows the player to explicitly select a difficulty setting (like “easy”, “intermediate”, or “hard”) and is mostly used in single player games. Dynamic difficulty adjustment is used in games such as Mario Kart (where players that are farther behind are given better powerups) or Left 4 Dead. These systems adapt and respond to the abilities of a player during a game session [10].

While this kind of approach may seem ideal, care needs to be taken in competitive scenarios to ensure experienced players don’t feel cheated because their opponents only won when they were helped [15]. Therefore the system must be unobtrusive and noticeable in order to be effective [12]. A huge benefit to dynamic systems is that it reduces the need for iterative refinement based on play testing since the system regulates the difficulty itself [12].

2. Matchmaking: Matchmaking systems are popular in more “hardcore” games. For example, Dota 2, StarCraft 2, and Halo try to match players together who have the same level of experience or ranking to ensure the level of competition is balanced [15].

3. Asymmetric Roles: These types of games allow for several different roles for people who have different skills. For example, Team Fortress 2 allows players who are skilled shooters to choose a class such as the sniper. Otherwise, the player can choose a class such as the medic whose task isn’t focused on shooting, but healing and supporting other players. These games are carefully designed to ensure that every class contributes to the team and no one class is considered unnecessary.
2.2 Aim Assistance and Virtual Pointing

The foundations of virtual pointing can be traced to work done by Paul Fitts to model one-dimensional pointing in the physical world [8]. The Fitts’ Law equation basically states that the smaller and farther away an object is, the longer it will take to target. While Fitts’ Law was originally meant to model physical pointing, it has been shown that it also models virtual pointing and is the backbone of all pointing work done in HCI.

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) [2]. This means that pointing in virtual spaces isn’t limited by the same things that limit physical pointing, since it is possible to manipulate the control to display space, allowing systems to make pointing easier [4].

With these ideas in mind, three basic approaches have been developed to facilitate pointing:
1. Reduce Distance: These solutions try to reduce the amount of distance between the cursor and the target (e.g., change a right-click activated linear list to a pie menu where each item is the same distance away from the cursor).
2. Increase Width: These solutions try to increase the target size in motor space (e.g., ‘sticky’ targets that require more movement to cross over the target).
3. Reduce Distance and Increase Width: This category combines the first two approaches by reducing the distance to the target and increasing the size of the target (e.g., the Angle Mouse technique [17] that changes the control to display ratio depending on the current phase of pointing to simulate the effect of reducing distance and increasing width).

2.3 Aim Assistance in Games

Previous targeting assistance research mostly uses these approaches for helping older or disabled people use a computer to select items in 2D interfaces [7] [18]. 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. Target assistance has been shown to be effective at improving competition in 2D games where players have uneven skill using techniques [3]. The assistance was not noticeable and participants noted they had increased enjoyment in games where assistance was present.

The use of targeting assistance techniques in commercial games are mostly confined to console games that use gamepads as input because gamepads are not as precise as mice [14]. Despite the fact that targeting assistance is used in some games, we have not found a commercial game that uses it as a balancing mechanic.

Recent work has been done to show that the Fitts’ Law equation holds in 3D virtual spaces [13]. This suggests that the targeting assistance techniques that have been developed could be applied to 3D games with good performance. However, previous research has also shown that adding even simple game elements to a 2D aiming task can significantly change target acquisition time [9]. The complexity of a 3D FPS game environment (e.g., moving while aiming in 3D, moving targets, multiple targets, targets that shoot back, etc.) may interfere with the potential benefits of aim assist techniques, ultimately reducing their efficacy.

3. SINGLE-PLAYER STUDIES

To investigate targeting assistance in 3D FPS games, we first conducted a series of three studies to test the efficacy of various techniques in a single-player scenario (see [16] for details). We implemented five techniques (target lock, bullet magnetism, area cursor, sticky targets, and gravity) in a custom game environment and carried out three performance studies.

3.1 Techniques

We chose to investigate five techniques (see [16]). Target Lock moves the crosshairs of the player to the closest target’s head and was activated using a key press. Bullet Magnetism “bends” bullets towards the closest target if a target is within the activation range. Area Cursor increases the size of fired bullets. The Sticky Targets method works by changing the control-to-display (CD) ratio when the crosshairs are over a target. And the Target Gravity method gives each target an attractive force that results in a player’s crosshairs being dragged towards targets.

3.2 Apparatus

We developed an FPS game including a walkthrough level and a shooting gallery level using the Unreal Development Kit (UDK) to test the performance of the aim assistance methods. The game was developed in UnrealScript, using Visual Studio 2010 with the Nfringe add-on by PixelMine. Enemy targets were shot by placing the center of the crosshairs on the target and left-clicking the mouse. The targets in S1 were stationary. In S2, the enemies moved in a random (but limited) pattern when the player was in the enemy’s line of sight. In both studies, the player movement was controlled with a standard WASD scheme.

Aim assistance levels were determined through pilot tests, and were custom-made. The walkthrough level was split into a hospital-themed area and a warehouse area. In the hospital area, enemies were placed in rooms along with friendlies (distractors that don’t shoot back). The warehouse area was darker and the enemies and friendlies were encountered as the player moved. 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.

3.3 Study Summaries

In the first study (S1), we used the simple 3D shooting range so that it would be similar to the 2D games used in past evaluations of aim 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) that were both targets (enemies) and distractors (friendlies). 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).

Our results showed that although all techniques worked well in S1, the presence of realistic game elements compromised the efficacy of most techniques – sticky targets and target gravity in particular. However, bullet magnetism and area cursor worked well in several settings, and were not highly perceptible. These two techniques are interesting in that they both change the targeting process after the user has completed their aiming motion.
and pressed the trigger; in contrast, the two worst techniques (gravity and sticky targets) both adjust the user’s crosshairs during the aiming action. We speculate that there is value in letting the user plan and execute their own aiming motions, and only adjusting the shot during the flight of the bullet.

4. MULTI-PLAYER STUDY
Having established that aim assistance techniques are valuable in 3D FPS play, we propose to balance play in a multi-player FPS.

4.1 Chosen Techniques
The Bullet Magnetism and Area Cursor assistance methods have been chosen for this study because these two methods tended to increase performance under a range of conditions with low noticeability [16]. In our previous work, we used static levels of assistance; however, in the multi-player study, we will use dynamic difficulty adjustment to decide how much assistance to provide. The system has 10 levels, where 1 is the lowest amount of assistance and 10 is the highest. To calculate the level of assistance given, the player with the most kills will be compared with the weaker player’s kill and the difference will be compared for each assistance technique. For example, if the current leader has 5 kills and the other player has 1, the level of assistance for the player with 1 kill will be 4. The two techniques are described further.

4.1.1 Bullet Magnetism
The Bullet Magnetism technique “bends” the bullet towards the closest target if a target is within the activation range. The end effect can be seen in Figure 1. Bullets in the developed UDK game are instant shots and are therefore described by a vector. The vector is adjusted towards the first enemy that is within range (160 UDK units * Level, where 1 ≤ Level ≤ 10) of this vector when the player fires and before the bullet collision logic. 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 more the vector is corrected. Additionally, the closer the player is to a target when shooting, the more the bullet is attracted to 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. This method does not move the crosshair or change the CD ratio so it may be less intrusive than other methods. However, our single-player results suggest that Bullet Magnetism may have issues if multiple targets are present as it may be attracted to the wrong opponent.

4.1.2 Area Cursor
The Area Cursor technique follows the original 2D implementation [11] 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 [11] meaning bigger cursors lower the movement time. In the game implementation, the size of the crosshair changes as the activation area changes, as
seen in Figure 1. Normally the regular crosshair radius is 10px on the screen. When Area Cursor is used, the size of the crosshair radius is 10px + (5px * Level) where 1 ≤ Level ≤ 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, the growth of the cursor was subtle enough that users tended not to notice it growing.

Area Cursor has also been shown to improve targeting performance for older adults [18] and users with motor impairments [7]. It also doesn’t degrade 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 [5]. Distractor targets can also become a big issue if the cursor is too large [11]. The implementation handles this multiple targets issue by choosing the target closest to the center of the crosshairs.

4.2 Study Description

In the multi-player study, we will use our game developed with the Unreal Development Kit for a combat game with a single opponent. Participants will be asked to fill out a questionnaire about their gaming habits and FPS experience so that novice participants can be matched with an expert. Participants will be told that they will be testing novel game balancing techniques and in some rounds of play their guns may behave differently. Then, these two players will join a server and play a one-on-one deathmatch game. The level will be a modified deathmatch map that comes with UDKit. It contains three floors and an open middle area. Because this map is too big for two players, we will block off sections so players don’t simply wander around the whole match looking for each other.

Participants will first be given a round with no assistance for training. Then, three rounds (Bullet Magnetism, Area Cursor, and no assistance) will be presented (counterbalanced using Latin Square ordering). Each round will last five minutes. At the end of each round, survey questions will be presented about how fair players felt the game was and if they noticed anything helping them/anything strange going on, as well as how they thought they did and how they thought their opponent did.

5. Conclusion

Ensuring balanced play is an important part of game development. In competitive multiplayer situations, this is difficult to achieve because of varying skill levels. In FPS games, targeting assistance is a promising approach to achieving this. This paper outlines a series of preliminary single-player studies and a proposed multiplayer study that will attempt to provide a new way of balancing competitive multiplayer games in first person shooters by using two targeting assistance techniques, Bullet Magnetism and Area Cursor. Our goal is to show that assistance results in games that are closer in score than when no assistance is present, and that as a result, players prefer games with assistance. The results of this study will be a valuable resource for game developers to increase their game’s appeal to novice players, thereby boosting sales and allowing for a better player experience.

6. ACKNOWLEDGMENTS

We thank our Grand reviewers, the Interaction Lab, NSERC, and Grand for funding.

7. REFERENCES