Effects of Adaptive Time Delay in First Person Shooter Games

By: James Cannon, Saketh Dinasarapu, Ao Jiang, Hanzalah Qamar

Graduate Mentor: Samin Shahriar Tokey

Advisor: Mark Claypool

Abstract

Latency impacts responsiveness and fairness in first-person shooter (FPS) gaming, leading to an overall worse player experience. Time delay applies a fixed delay to players to adjust for latency, but has the drawback of always adding delay regardless of whether players are interacting or not. Adaptive time delay dynamically adds latency when players interact during the game. This study assesses user performance across various latency settings to see if adaptive time delay mitigates the cost of time delay for FPS gaming. We used a custom developed game called Zombiefield as a testing environment to collect player performance metrics. By examining player accuracy, quality of experience, and other parameters, results from our study showed benefits of adaptive time delay in overcoming latency in FPS gaming compared to time delay.

Table of Contents

1. Introduction

The excitement of First-Person Shooter (FPS) gaming heavily depends on the fast-paced action, precise aiming, and split-second decision-making, making latency a critical concern for both developers and players.

Latency, often referred to as the delay between a player's action and its corresponding response in-game, can significantly influence gameplay fairness, responsiveness, and overall player experience. In competitive FPS environments, even the slightest delay can disrupt a player's performance and experience. For example, Team Sentinels took home the first international Valorant tournament in VCT Stage Two Masters Reykjavík in May 2021. Since it was a LAN (local area network) event at Reykjavík, Iceland, the gaming server was set in the same building as where the tournament is held, the Internet latency was minimal compared to a home network. One of the Sentinel's players, Tyson "TenZ" Ngo, said Valorant feels amazing on a LAN because of the low latency. Figure 1 shows his tweet about the latency change at LAN events compared to his own home network (Dacanay, 2023). At the same time, LAN events can also ensure that players competing have similar network latencies, thereby preventing any unfairness due to different network latencies that players may have at their own home networks.

Figure 1: TenZ Tweet (Dacanay, 2023)

To address the challenges posed by latency, developers have devised various latency compensation techniques, aiming to minimize the effects of delay on unfairness. Adaptive time delay dynamically adjusts the delay added to one player based on player interactions with others to maintain the same delay across all players, which can potentially minimize the negative effects of latency such as unfairness. This approach differs from traditional static delay compensation methods, which always adds the same delay for players even if there is no interaction between players.

In this user study, we explore the effect of adaptive time delay within the context of Zombiefield, an FPS game developed by Samin Shariar Tokey. In a user study, participants played Zombiefield with various latency configurations. A total of 38 users participated in this study. Analysis of the results show that adaptive time delay leads to a better gaming experience, as users' accuracy and quality of experience show a consistent improvement at various latency configurations compared to the normal static time delay method.

In this report, the background chapter gives general information about latency in FPS games and various latency compensation techniques. The methodology chapter provides details on the user study. The analysis chapter shows the study results and the analysis. The conclusion chapter summarizes our findings. The future work chapter shows the potential continued research on adapted time delay.

2. Background

This section provides information important for understanding the concepts of our project. Included topics are latency in FPS (first-person shooter) games, techniques to deal with latency, time delay, adaptive time delay, and details on Zombiefield, the game used in our testing.

2.1 Latency in FPS Games

2.1.1: First Person Shooter Game

A first-person shooter (FPS) game is a genre of video game where the player is placed at the perspective of the game character and viewing the game via in-game character's eyes. In this type of video game, the primary focus or the goal is to eliminate enemies or achieve specific tasks by utilizing various weapons or character abilities.

FPS games typically emphasize fast-paced action, quick reflexes, and precise aiming, all of which place significant demands on the game's fairness and responsiveness to player input. A precise and rapid response to a player's actions can greatly enhance the overall experience of an FPS game. Additionally, FPS games often feature both single-player experiences and multiplayer modes, catering to casual and competitive players alike. Fairness is a critical aspect of FPS game development due to the competitive and dynamic nature of the genre.

In the gaming industry, numerous competitive FPS games utilize various latency compensation methods. Figure 2 shows 2 games, Overwatch by Blizzard Entertainment and Valorant by Riot Games, which are two examples of competitive FPS games that utilize various latency compensation methods. In this research, Zombiefield, developed by Samin Shahriar Tokey, is employed to assess the impact of adaptive time delay.

Figure 2: Overwatch & Valorant

2.1.2: What is Latency

In FPS games, latency refers to the delay between a player's actions (such as shooting an enemy) and the corresponding response on the screen (such as enemy getting shot). Sometimes, latency is also called “lag” or “ping”. This delay can significantly impact the gaming experience, especially in scenarios where split-second decisions are crucial. In the pursuit of minimizing time delay, FPS game developers implement a range of strategies aimed at ensuring a more responsive and immersive gaming experience, even when faced with latency challenges.

Latency plays a big role in maintaining the player’s experience of the game since it directly affects the player's experience of the game's responsiveness. In a competitive setting, the difference in two players' latencies also causes unfairness where the player with lower latency usually gains advantage. For example, when two players with different latency meet in game around a corner, the player with a lower latency will see the player with a higher latency first because of the shorter communication time with the game server, thus gaining an advantage. When a player engages in an FPS game in a high-latency environment, all player's inputs to the game, including movement and feedback from various actions, exhibit a noticeable delay.

2.1.3: Cause of Latency

Latency issues can have many causes, which can be divided into four categories. The first is network latency, often measured in milliseconds(ms). This delay is caused by the time it takes for data to travel between the player's device (client) and the game server. It is dependent on the physical distance between the player and the game server as well as the local data transfer devices (router, wires, etc) (Liu, 2022). Second, the latency may be caused by the hardware, for example the polling rate of the mouse and the keyboard, or the framerate of the monitor a player is using. Usually, the higher the polling rate of the input devices, the lower the hardware latency will be. Third, the server tick rate in an online FPS game refers to how frequently the game server updates the game world and processes player actions. A higher tick rate means more frequent updates but can also lead to higher server load. Games with higher tick rates often feel more responsive, but they require a robust server infrastructure. Lastly, the latency may also depend on the latency compensation methods used by the game. For example, if a game is utilizing a prediction method for their game, the input from players will be slightly delayed in achieving that prediction (Liu, 2022).

In this research, all the hardware used is set to be capable of running the test game Zombiefield at the best performance to keep the hardware latency as small and constant as possible. As the network latency and the server tick rate will be simulated, the adaptive time delay latency compensation method will be the only influence towards the testing latency environment.

2.2 Techniques to Deal With Latency

2.2.1: Prediction in FPS Games

In First Person Shooter (FPS) games, prediction can help to mitigate latency. It involves the game server anticipating the actions of players to reduce the perceived delay to the server. When a player makes a movement or shoots, the server predicts their next actions and immediately updates the game world accordingly. Additionally, self-prediction mechanisms are employed by the player's own client-side software. This means that not only does the game server predict the player's actions, but the player's own device predicts their future movements, as well. This predictive approach, both on the server and client side, helps maintain smooth gameplay, ensuring that actions like aiming, shooting, and dodging feel responsive even in the face of network latency. This synergy of predictive algorithms from both the server and the player's device can create a seamless gaming experience in fast-paced FPS games.

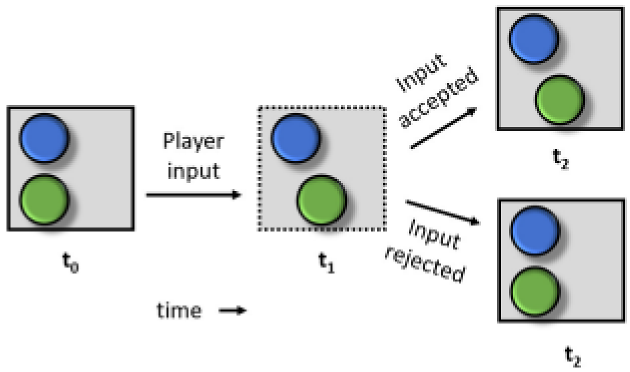


Figure 3: An example of self-prediction. (Liu, 2022)

In Figure 3, both the player and server share the same game world at t_0. At t_1, the player issues a command to move the green avatar to the right, and the client expects the server to allow it. The client displays the green avatar in the anticipated new position. When the server receives and responds to the player's input at t_2, two possible scenarios unfold. In the first case, the server accepts the input, and the new position of the green avatar is confirmed. The client's prediction was correct, so the game world remains unchanged. In the second case, the server rejects the player's input, possibly due to obstacles the client was unaware of. In this situation, the client adjusts the game world to match the server's decision.

2.2.2: Time Warp in FPS Games

Time Warp is another latency compensation technique in FPS games, particularly for online multiplayer scenarios. It allows the server to warp or adjust the game timeline to synchronize players' actions accurately. When one player experiences higher latency than others, time warp helps ensure fair and consistent gameplay. It lets the server adjust the position and actions of players to align with the most recent game state, reducing situations where a player's actions seem out of sync with the rest of the game world (Liu, 2022). This technique plays a crucial role in maintaining a level playing field and minimizing the frustration of high-latency players.

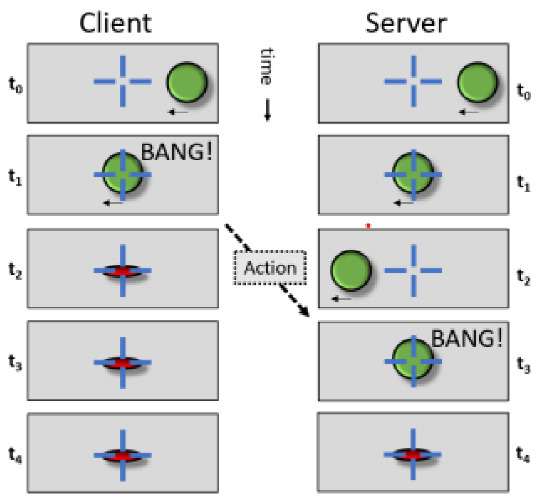


Figure 4: An example of time warp. (Liu, 2022)

In Figure 4, we have a shooter game where a player on the client is shooting at a moving green target. The player aims at the target with a reticle and fires the weapon at time t_1. This action is sent to the server a bit later, around time t_2. However, in the server's world, the green target has already moved past the reticle by time t_2.

To address this delay, the server essentially rewinds time back to when the player took the shot at time t_1 and applies the action to the game world as it was at that moment. This ensures that the shot hits the target correctly, even though there was a delay in transmitting the action to the server.

2.3 Time Delay - Understanding Latency Compensation

In multiplayer online gaming, the management of network latency is a concern for both game developers and enthusiasts. Time synchronization, particularly in the context of multiplayer games, involves orchestrating deliberate delays in various facets of gameplay to make the network responses equal and fair for players. This fundamental concept serves as the bedrock for ensuring that all participants enjoy an equitable and seamless experience, regardless of their diverse network conditions.

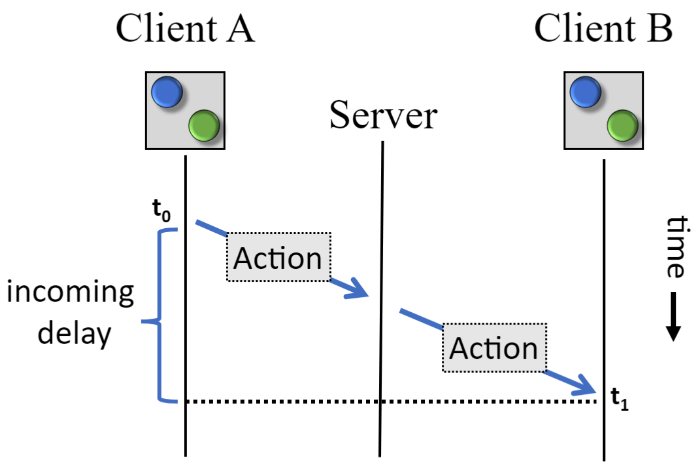


Figure 5: Incoming Time Delay (Liu, 2022)

2.3.1: Incoming Time Delay

Incoming Time Delay revolves around the deliberate introduction of time delays during the processing of incoming player actions. The objective is to achieve synchrony in actions across all connected clients, thus contributing to a level playing field and a consistent multiplayer gaming experience. From the perspective of the player's client, this process often involves temporarily storing player actions before executing them. To illustrate this, imagine a scenario where a player's action is intentionally held back to ensure its simultaneous execution on all clients.

2.3.1.1: Exploring Techniques in Incoming Time Delay

Researchers and game developers have explored techniques for achieving incoming time synchronization. For instance, Mauve contributed insights to formalizing this concept, shedding light on its practical applications (Mauve, 2000). Diot and Gautier introduced a unique notion called "bucket synchronization incoming delay", which entails storing and deferring game state updates received by a client until their designated time intervals (Diot, 1999) . Lin took the concept of incoming time synchronization a step further by implementing it at both client and server levels, a technique they aptly term "sync-in and sync-out" (Lin, 2003). Savery, on the other hand, proposed the utilization of incoming time synchronization at the server side, taking into account variations in client latencies and making adjustments based on client proximity (Savery, 2014). Additionally, Chen merged self-prediction with incoming time synchronization to tackle state inconsistency issues (Chen, 2007) while Zhang employed incoming time synchronization alongside extrapolation to enhance gameplay consistency (Zhang, 2006).

While incoming time synchronization has advantages, it is not without its tradeoffs. Simulated evaluations have illustrated its potential to eliminate state inconsistency in games, especially in the presence of up to 500 ms of network latency. However, this pursuit of consistency may come at the cost of reduced responsiveness, particularly in fast-paced games such as first-person shooters (FPS). For instance, introducing a 200 ms delay can potentially result in a 50% reduction in player performance within FPS games (Zander, 2005). Striking the delicate balance between responsiveness and consistency remains an ongoing challenge. Yet, for network latencies hovering around the 250 ms mark, fine-tuning incoming time synchronization techniques can enhance player performance by up to 30% with negligible impacts on Quality of Experience (QoE).

2.3.2: Outgoing Time Delay

In the intricate landscape of multiplayer online gaming, the management of network latency is an ever-present concern, and one critical aspect of addressing this challenge is outgoing time synchronization. This mechanism serves as the keystone for aligning the delivery of critical game state updates and ensuring fairness among players. Outgoing time synchronization entails the introduction of strategic delays before transmitting crucial game state information to all connected clients, with the aim of achieving uniform latencies across the player base. One common strategy employed in implementing outgoing time synchronization is introducing the delay before sending updates to players, as illustrated in Figure 2. Additionally, this delay can be dynamically adjusted based on variations in client latencies, further bolstering the synchronization efforts.

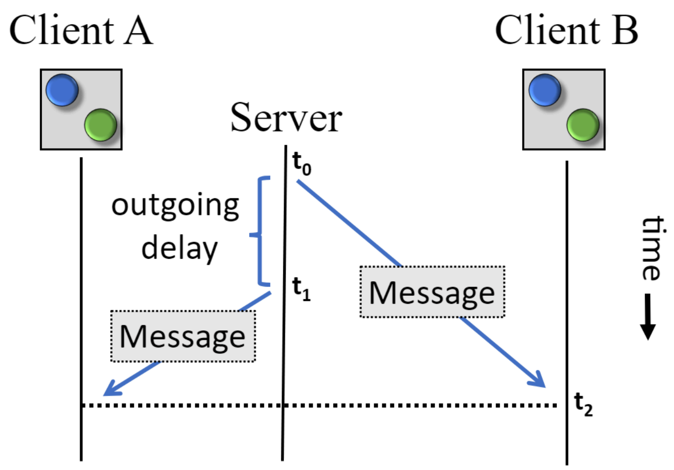


Figure 6: Outgoing Time Delay (Liu, 2022)

2.3.2.1: Exploring Strategies in Outgoing Time Synchronization

Zander has implemented outgoing time synchronization within game servers to ensure that latencies among players are balanced, with adjustments tailored to the specific game type (Zander, 2005). Brun has proposed the adoption of outgoing time synchronization within servers to uphold fairness among clients, particularly when contending with varying latencies (Wikstrand, 2005). In the context of peer-to-peer architectures, Wikstrand have introduced an "input buffering" scheme that leverages outgoing time synchronization to harmonize the arrival times of messages among players (Wikstrand, 2005). Kaiser has taken an approach where they aggregate game update messages on the client side into a single, larger packet, deliberately delaying its transmission for a predefined duration (Kaiser, 2010). Similarly, Li has introduced outgoing time synchronization in a cloud-based game platform, where the degree of delay is inversely proportional to the latency of each client (Li, 2018). They have also put forth maximum thresholds to prevent exceedingly high latencies.

2.3.2.2: Potential for Enhanced Fairness and Satisfaction

While incoming time synchronization has received more extensive evaluation, outgoing time synchronization holds significant promise in enhancing fairness among players, particularly within the context of FPS games. It has the potential to reduce performance disparities by roughly 30% even when facing up to 400 ms of latency (Zander, 2005). Moreover, outgoing time synchronization can positively impact player Quality of Experience (QoE) by enhancing perceived fairness, even with relatively minor delays, such as 80 ms (Li, 2018).

In summary, outgoing time synchronization is a potential component in multiplayer game design, offering a viable means to mitigate the complexities of network latency. By judiciously introducing delays in the transmission of game state updates, this technique strives to promote fairness and synchronization among players. Although it has undergone less extensive scrutiny compared to incoming time synchronization, it holds promise in enhancing the multiplayer gaming experience, particularly in scenarios where disparate latencies among players can lead to frustration and inequity.

2.4 Adaptive Time Delay

Adaptive time delay is a currently proposed alternative to time delay found exclusively in our testing game, Zombiefield. Due to this, there is no prior research to draw on and we must do extensive testing ourselves. The idea was first proposed by Tokey as a way to enhance fairness in FPS games (Samin, 2023), though he was unable to gather testing data. However, similar tests have been done on message synchronization in MMOFPS games (Paik, 2008) and with prediction in a haptic air hockey game (Yusuke, 2012). It is effectively the same as outgoing time delay, however the delay is dynamically adjusted according to the network latency and interactions of all players on a server. When it is activated, it will adjust the players with high latency to the latency of the lowest latency player that they are interacting with.

2.4.1: How Adaptive Time Delay is Applied

In general, adaptive time delay is applied whenever two or more players can interact with each other. In FPS games specifically, interaction is usually defined by players having an unobstructed view of each other. Zombiefield does adaptive time delay this way, with Figure 7 showing a simple depiction of how it works in game. Paik et al. defines interaction as the likelihood of players to interact within a circular disk known as an area of interest (AOI) around a player (Paik, 2008). Interaction could also happen when an important object needs to be on both players' screens in the same place at the same time, but there remains to be any application of this.

Figure 7: Adaptive time delay active (left) and inactive (right).

Adaptive time delay also factors in local delay and could alternatively use incoming delay as well, though both incoming and outgoing delay mitigate unfairness. Additionally, time delay can be applied either gradually or instantly when activated. A smooth, gradual increase avoids abrupt and disturbing changes in latency while instant activation is marginally more fair. We opted to use gradual activation and deactivation in our testing.

2.4.2: Possible Advantages Over Time Delay

Adaptive time delay should enhance fairness in FPS games as much, if not more than, standard time delay. By not always being active, adaptive time delay is also speculated to have better effects on QoE (quality of experience) for players. High latency players can still have a fair experience when it matters, and low latency players can enjoy responsive gameplay outside of crucial moments.

2.5 Zombiefield

Figure 8: Zombiefield menu screen.

The game used in our testing is Zombiefield, an FPS with realistic graphics built in Unreal Engine by Samin Shahriar Tokey (Samin, 2023). Figure 8 shows the menu screen, including the buttons that let people play and quit the game. The game features a dark, enclosed map with many obstacles, including walls and gravestones. Armed with an assault rifle, you must shoot the incoming horde of zombies to survive for as long as possible. The player can also sprint, jump, and aim down sights, all with the goal of earning a high score. A gameplay screenshot can be seen in Figure 9, showing the core elements of the game. There are also shooting and running challenges that get progressively harder each time they are completed.

Figure 9: Zombiefield gameplay, showing a player aiming at two zombies.

2.5.1: How Adaptive Time Delay is Implemented

The game draws invisible rays between players and enemies that will activate time delay as long as they are not blocked by obstacles or terrain. Since Zombiefield unfortunately does not currently support multiplayer servers, only zombies are able to activate time delay. Artificial delay is added to inputs while certain zombies have line of sight with the player, simulating the effect that would be achieved with other, higher latency players. This effect is gradually applied and also gradually declines when line of sight is broken with those zombies. An example is shown in Figure 10, showing an early version that still displayed the rays and the current latency on screen.

Figure 10: A special zombie that activates time delay. Notice the latency in the top right.

2.5.2: Modified Version

The version of Zombiefield used in our testing is slightly modified to better collect player data. On top of the already implemented points system, this version features set length rounds with different, configurable parameters for each. We did not use the shooting or running challenges as part of our testing. Important parameters include number of rounds, applied latency, and how long it takes to increase to that latency. These configurations are explained in more detail in our methodology and testing sections.

3. Methodology

This chapter describes our user study setup and procedure used to collect and analyze data on the impact of adaptive time delay in Zombiefield.

3.1 Hypothesis

H1. When adaptive time delay is enabled, user accuracy improves at high latency settings when compared to regular time delay.
H2. When adaptive time delay is enabled, user accuracy remains unchanged at low latency settings.
H3. When adaptive time delay is enabled, user quality of experience (QoE) increases at higher latency compared to regular time delay.

3.2 Pilot Studies and Game Adjustment

Participants started testing by signing a waiver, filling out a demographic survey, and taking a reflex test. Participants then underwent 16 rounds of gameplay, each designed to incorporate distinct latency values and compensation techniques. Engaging in discussions around parameters with Tokey led to the quick implementations of game features like a delay indicator for initial debugging and important bug fixes like stopping users from being able to climb onto walls. These enhancements collectively aimed to elevate the overall gaming experience for our participants and improve the quality of data gathered.

We also worked with Tokey on a configuration file to fine-tune various aspects of the gaming environment. Parameters included the number of rounds a user played, alongside the incorporation of distinct latency compensation types including "No Time Delay," "Adaptive Time Delay," and "Regular Time Delay.”

To gather data, a user interface was integrated into the study. After each round, participants were presented with a slider and a binary question, prompting them to rate the round on a scale of 1-5 and provide feedback on the overall acceptability of their experience (yes/no).

In order to assess participant abilities, a reflex test and demographic survey were incorporated into the study design. These additions aimed to capture user data, contributing to the user demographics and opinions.

Pilot studies were conducted with IQP team members as both participants and proctors, as well as individuals unfamiliar with the study as participants. These pilot studies helped us fine-tune our settings, procedures, and overall duration of the study. The initially predicted time of 45 minutes with two proctors was reduced to around 30 minutes with only one proctor. This still gave users plenty of time to play the game while going through earlier parts of the study at their own pace.

3.3 Final Parameters and Data Recorded

Figure 11 outlines the specific latency values assigned to different rounds, as well as values affecting how adaptive time delay is implemented.

	Rounds / 75s	Before TD Latencies	After TD Latencies	Adt kick in delays	Adt let off delays	Td stay off inertia delays	Td stay on inertia delays
Base Group	1	0	0	0	0	0	0
Group 1	2	0	50	0	0	0	0
	3	0	100	0	0	0	0
	4	0	150	0	0	0	0
Group 2	5	50	50	0	0	0	0
	6	100	100	0	0	0	0
	7	150	150	0	0	0	0
Group 3	8	25	50	0	0	0	0
	9	50	100	0	0	0	0
	10	75	150	0	0	0	0
Group 4	11	0	50	2	2	0	0
	12	0	100	2	2	0	0
	13	0	150	2	2	0	0
Group 5	14	0	50	0	0	2	2
	15	0	100	0	0	2	2
	16	0	150	0	0	2	2

Figure 11: Round Setup for User Study

Each one of the 16 rounds lasted 75 seconds. The "BeforeTDLatencies (ms)" column signifies latency values before the introduction of Adaptive Time Delay (Adt), while "AfterTDLatencies (ms)" represents the latency values post-application of Adaptive Time Delay. The parameters "Adt kick in delays" and "Adt let off delays" detail the time taken (in seconds) for latency to transition between the pre-Adt (Regular Time Delay) and post-Adt (Adaptive Time Delay) states, and vice versa. Additionally, the "Td stay off inertia delays" and "Td stay on inertia delays" parameters indicate on the minimum time (in seconds) that Time Delay (TD) remains inactive or active before transitioning.

In our study, participants were exposed to different latency configurations to analyze their performance within the Zombiefield game. The Base group provided a baseline for assessment, devoid of any delay, allowing us to understand performance without external latency factors. Group 1 was Adaptive Time Delay (Adt), where latency spiked upon encountering a networked enemy, or an enemy that will trigger adaptive time delay, and dissipated upon their departure. Group 2 maintained a constant delay throughout the entire round, enabling an examination of performance under consistent latency conditions. Group 3 involved both Regular Time Delay and Adaptive Time Delay, with the base delay set at half when the user faced a networked enemy. Groups 4 and 5 delved into the intricacies of Adaptive Time Delay, scrutinizing the impact of Adt kick in, let off, and Time Delay (TD) stay on/off inertia delays on user experience.

In our study, we focused on latency as the primary independent variable, deliberately adjusting it to gauge its impact on player performance and experience. Conversely, the dependent variables recorded included accuracy, quality of experience per round, shots hit, deaths, score, and movement—metrics for evaluating participant performance and obtaining subjective feedback. By systematically manipulating these parameters, our goal was to uncover relationships between experimental conditions and observed outcomes. This approach aimed to yield an understanding of the factors influencing user performance and experience within the context of our study.

3.4 User Recruitment

For user recruitment, we sought out students from various majors including computer science (CS), robotics engineering (RBE), electrical and computer engineering (ECE), and interactive media and game development (IMGD). A Calendly link was distributed to allow users to schedule their preferred times for participation. As an incentive, compensation was offered in the form of IMGD playtesting credit and an opportunity to win a $10 gift card through either achieving a high score in the study or participating in a raffle. To confirm attendance, participants received reminders on the day of their appointments.

3.5 User Data Collection

The data collection process included academic and personal information, gaming profile, platform and connection type, the human benchmark test, and a voluntary raffle entry. By collecting this data, we gathered information about participants' backgrounds, gaming behaviors, and reflexes, helping us understand the experience and performance of each user.

3.5.1: Academic and Personal Information

Participants's academic level and age were collected in the demographic survey as it helped identify participants' background on gaming. Since participants were recruited from the WPI student and instructor body, academic level and age were both gathered separately. Gender and years of experience in FPS games was also gathered.

3.5.2: Gaming Profile

Participants reported their experience with specific FPS games, including games they have played and their ranks in these games. Participants also indicated whether their playstyle leaned towards competitive or casual gaming. Self-assessment of skill levels in FPS games was also obtained from participants.

3.5.3: Platform and Connection Type

Participants were asked whether they primarily played FPS games on personal computers (PC), consoles, or other platforms. Participants also gave information regarding the type of Internet connection they have set up for their gaming environment, between an Ethernet or Wi-Fi connection.

3.5.4: Human Benchmark Test

As part of the study, participants went through a ten-round reaction time test online. Reaction time data from each participant's test rounds were recorded and analyzed to provide additional insights into participants' performance in FPS games.

3.5.5: Raffle Entry

Participants could participate in a raffle with a $10 gift card for the study as long as they provided their email addresses. Along with the raffle, the participant with the highest score won a $10 gift card. The same participant could not win both gift cards. Participants' emails were guaranteed not to be shared.

3.6 Procedures

To maintain consistency, a training video (Jiang, 2023) was created to provide detailed instructions for participants on what to do during the playtesting session and how to play the game. The playtesting session started with participants agreeing with terms of an Institutional Review Board (IRB) consent form. Following this, participants completed a demographic survey to gather information about their gaming background. Then, participants engaged in a human benchmark test, which involved clicking on a screen when the background changed to assess reflexes. After that, participants played a 4-minute practice round to understand the game mechanics until comfortable. As for official rounds, participants completed 16 testing rounds of gameplay. Each round involved playing the game for 75 seconds, followed by rating the experience on a scale of 1 to 5 and indicating whether or not the conditions of that particular round was acceptable. The total playing time for each session was 20 minutes. Additionally, participants were prompted to ask questions and give feedback after each session to gain an understanding of their experience and insights for improvement.

3.7 Computer Setup

The experimental setup had a high-performance computer equipped with an RTX 2080 graphics card, an Intel Core i9 11900K processor, 32 GB of Corsair Vengeance DDR4 RAM, a Samsung 970 EVO SSD, and an ROG Strix Z590E motherboard, a Logitech gaming keyboard and mouse, along with a Lenovo legion 240 Hz monitor.

4. Analysis

This chapter presents an analysis of data collected from the user study to validate the hypotheses. We firstly present the user demographic data, then the reaction time test data, followed by performance statistics including kills, score, and accuracy. We then dive into the post-round evaluation question data which includes data about quality of experience and acceptability of each round.

4.1 User Demographics

For this study, 38 participants signed up for the game testing, and 3 data sets were discarded due to the incompleteness of the testing play or inactivity during the testing play. In total, there were 35 sets of game testing data for this study.

	Mean	Median	Mode	Standard Deviation
Age (Years)	20.23	19	19	3.82
FPS Experience (Years)	7.36	7	10	5.08
General Game Skill Level (1-5)	3.65	4	4	0.90
FPS Game Skill level (1-5)	2.89	3	3	1.06
Reaction Time (ms)	195.55	192.2	211	24.10

Figure 12: FPS Experience Demographic / Reaction Time Test Data

Gender	Count	Percentage
Male	36	94.74%
Female	2	5.26%
Total	38	100%

Figure 13: Gender Breakdown Data

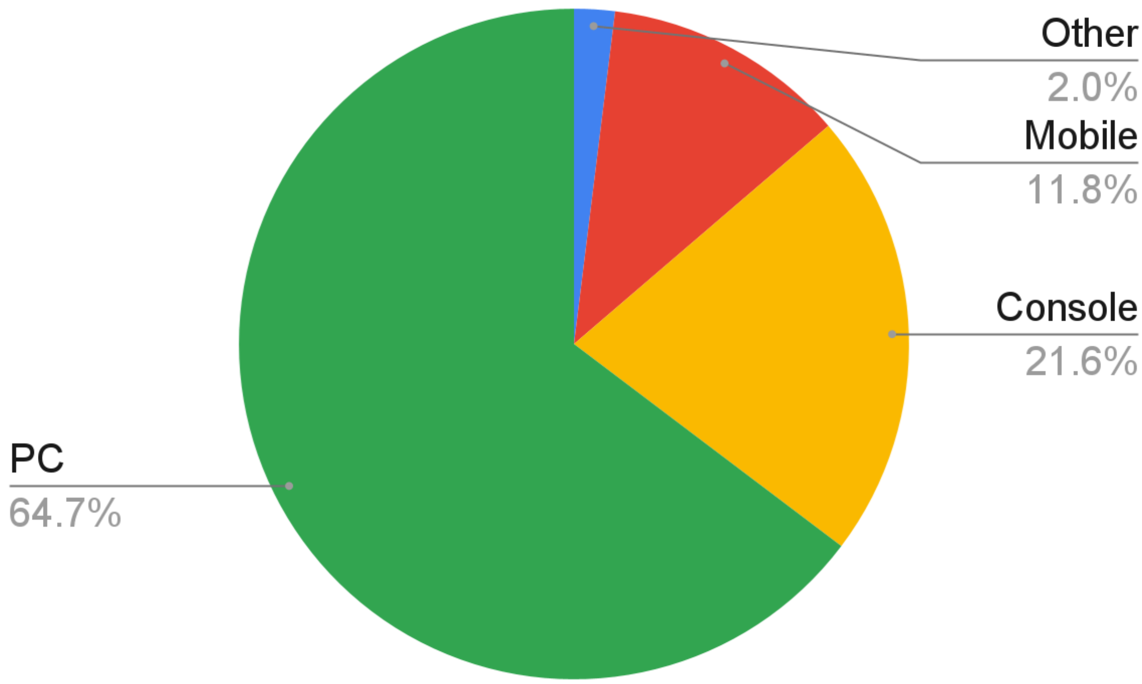


Figure 14: Platform Demographic Data

In Figure 12, we present statistics that highlight the diverse gaming experience and skills of our participants. The average age of our users was 20.24 years, with most participants being around 19 years old. This relatively young demographic is reflected in their gaming experience, with an average of 7.36 years in first-person shooter (FPS) games, indicating a mix of novices and veterans. In Figure 13, it is seen that the majority of the users (36 out of 38) that participated in the study identified their gender as male, and only two identified their gender as female.

The Figure 14 data also reveals that the general game skill level and FPS game skill level of our participants are above average, with mean scores of 3.66 and 2.89 out of 5, respectively. This suggests that while our participants consider themselves competent in general gaming, they perceive their skills in FPS games to be slightly lower. We also found that 89.5% of the participants have at least 2 years of FPS gaming experience, indicating that the users selected have a healthy amount of experience for the study. Additionally, the average reaction time of 195.55 milliseconds indicates a range of proficiency levels among our participants, which could influence their performance in FPS games. In Figure 14, we see that the majority of users play on P.C (64.7%) and console (21.6%). The figure also shows that a small number of users have experience in mobile gaming and other platforms.

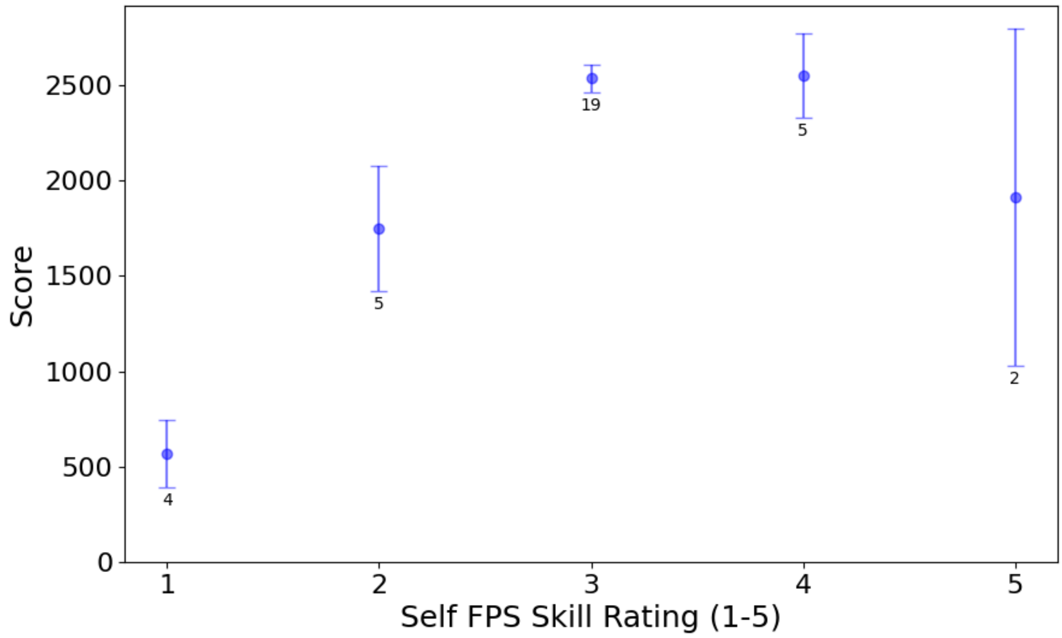


Figure 15: Score vs Self Score Rating

Figure 15 presents a comparison between the total user scores obtained over 16 rounds and their self-rated first-person shooter (FPS) skills from the demographic survey. The points on the graph are averages, the bars represent a 95% confidence interval, and the numbers between each set of data represent the count of user's that rated at the corresponding skill rating. Our analysis reveals a general alignment between the objective self-ratings and the scores achieved by participants in their respective user tests. Notably, the most common self-rating was 3 out of 5, which corresponded to the highest average score value of 2351. This observation suggests that participants' self-assessments of their FPS skills are relatively accurate and reflective of their actual performance in the tests.

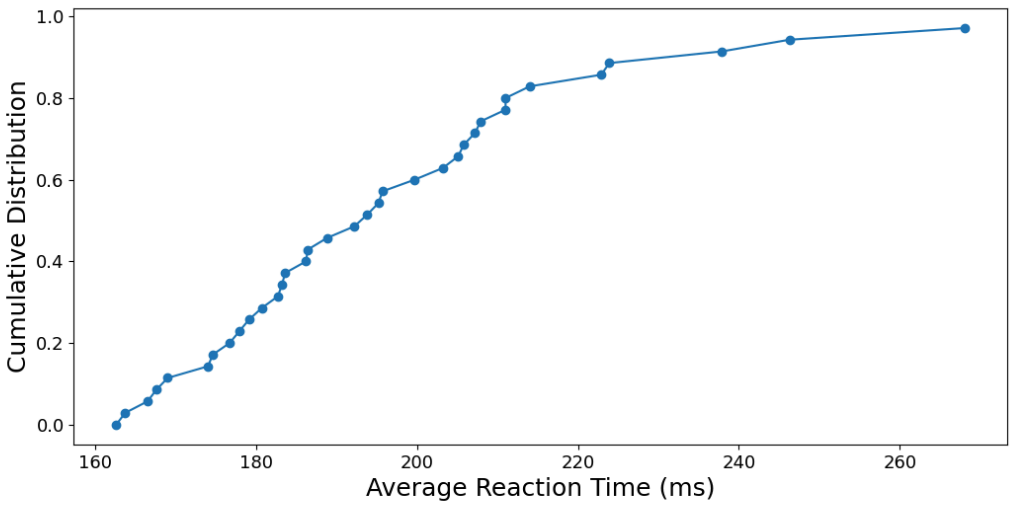


Figure 16: Average Reaction Time

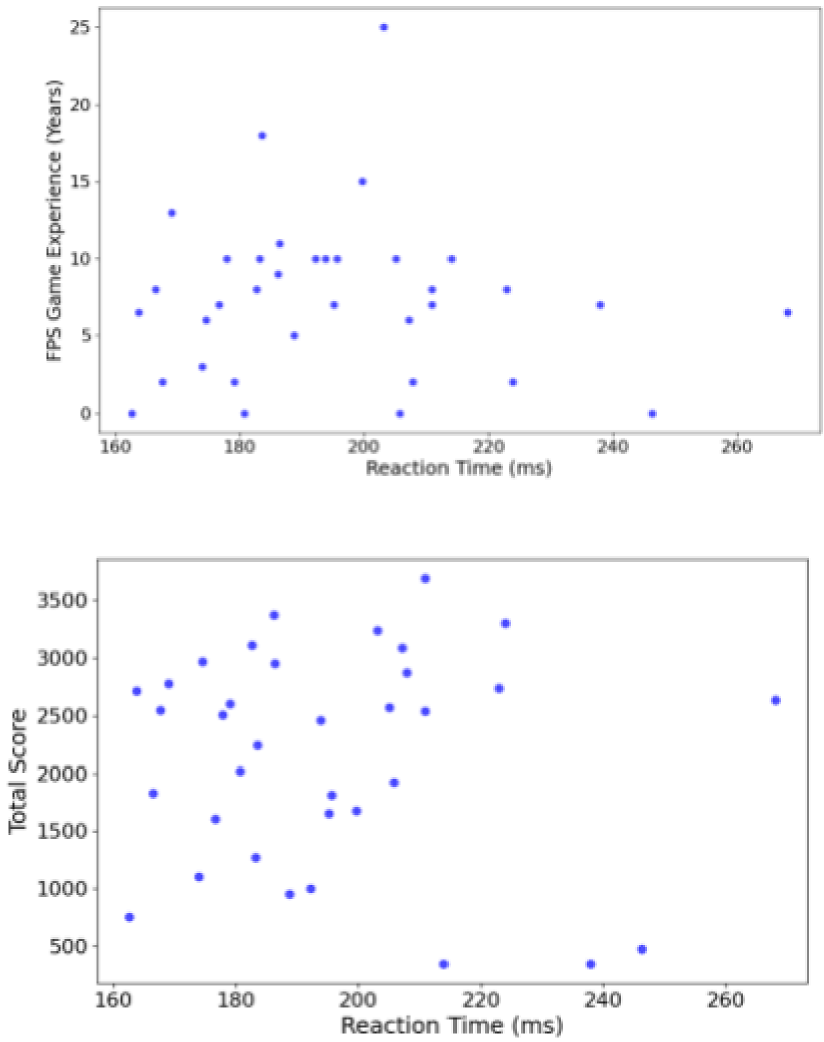


Figure 17: Reaction Time vs. FPS Experience and Score Scatter Plots

In Figure 16, we present the distribution of average reaction time test results for all users, based on 10 tests each. These results indicate that our users generally have a lower reaction time compared to the average human reaction time distribution (Jain, 2015). When calculating the results for the users, it is crucial to recognize that we did not subtract the latency of the system . This observation suggests that our user group may possess above-average reflexes or may be more accustomed to tasks requiring quick responses, which is a common characteristic in gaming populations. In Figure 17, we present scatter plots of the reaction time versus the FPS gaming experience of the user, and also the reaction time versus the total score of each user. Each dot represents one user.

4.2 Experimental Data and Observations

In this section, we present the experimental data and observations from the Zombiefield game testing. We took the data gathered from the game and presented graphs related to accuracy, score, kills, and other dependent variables against the independent variable: Latency.

4.2.1: Player Accuracy vs. Latency

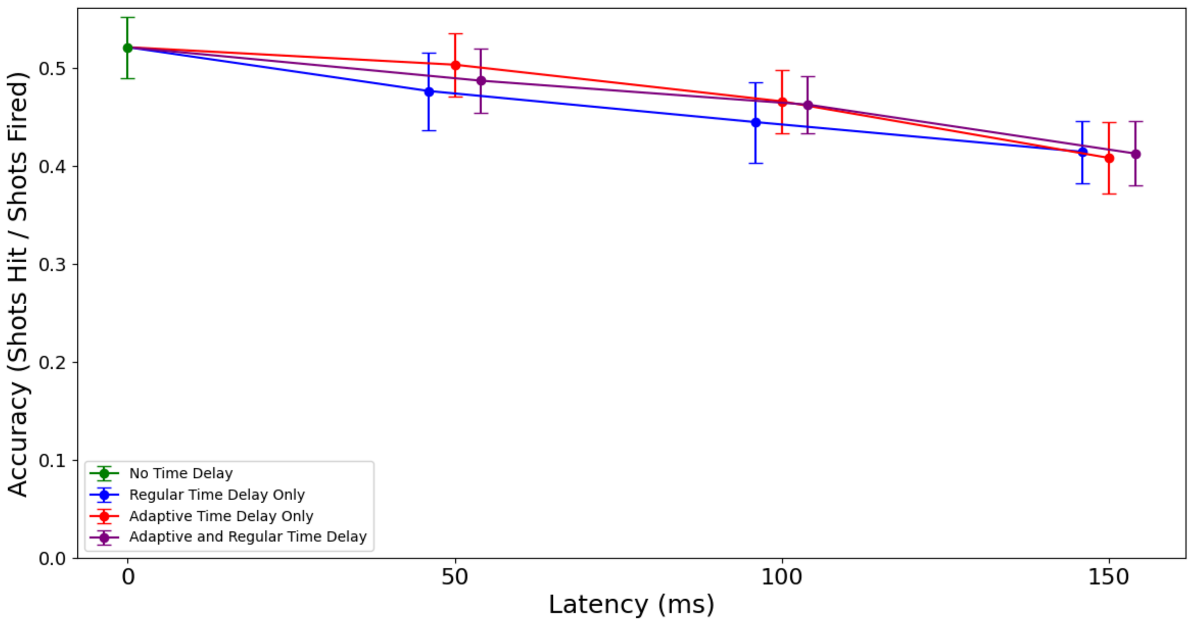


Figure 18: Latency Compensation Methods versus Accuracy

Figure 18 shows the differences in average accuracy values across all users, with each colored line representing a different latency compensation technique. The points represent averages on the graph, and the bars represent a 95% confidence interval of the data. In the proceeding graphs throughout this section, the reader can assume that the bars always represent a 95% confidence interval unless otherwise stated. The green line indicates rounds with zero latency throughout, serving as a baseline for comparison. The blue line represents rounds with consistent latency, without any adaptive time delay (TD) adjustments. The red line corresponds to rounds where latency dynamically switches from 0 to the value indicated on the x-axis upon encountering a networked enemy, showing the impact of adaptive TD. The purple line represents rounds with a combination of regular time delay for the entire round and an additional adaptive delay activated during encounters with enemies. In the proceeding graphs in this section, the colors all represent the same latency compensation technique for simplicity. In this case, the regular delay is set at half the value of the adaptive delay; for instance, at an x-axis value of 100 ms, the base latency is 50 ms, which then increases to 100 ms when the player encounters an enemy.

The data indicates that accuracy generally improves as latency decreases, aligning with expectations. More notably, the accuracy is consistently higher with both adaptive TD implementations compared to regular TD implementations. This suggests that adaptive TD is more effective for maintaining accuracy in gaming environments.

Interestingly, the accuracy values converge at the highest latency, by the data point at 150 ms latency. This trend suggests that at higher levels of latency, the effectiveness of different compensation techniques becomes less distinct. This observation is expected to be consistent across other performance metrics, such as kills and shots fired, indicating that at higher latencies, the distinction between compensation techniques becomes less significant.

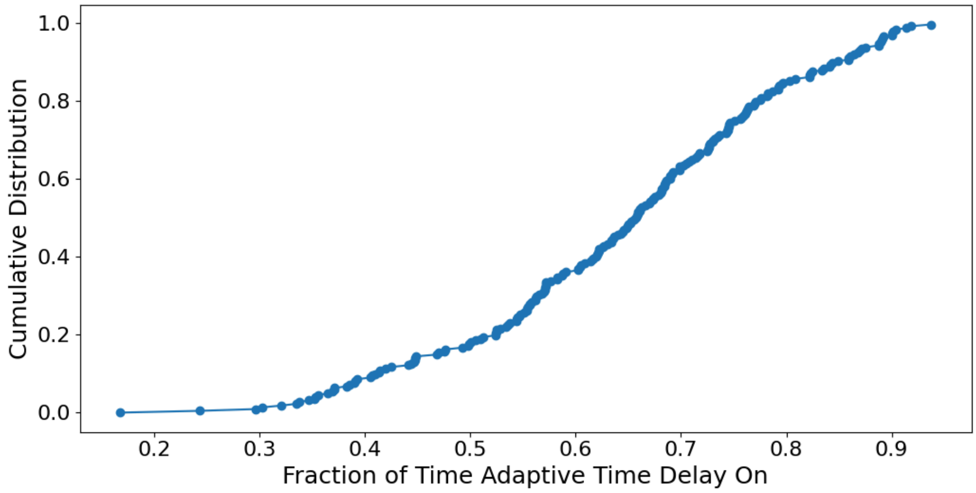


Figure 19: Fraction of Time Adaptive Time Delay On CDF

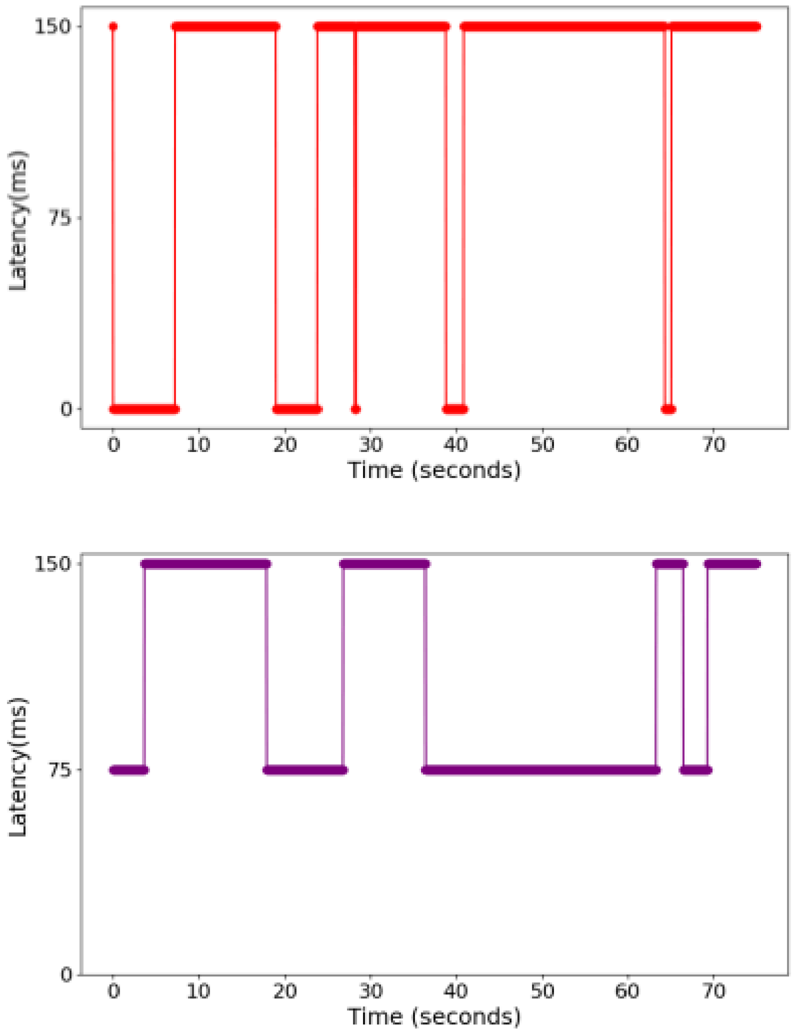


Figure 20: Adaptive Time Delay Vs. Time

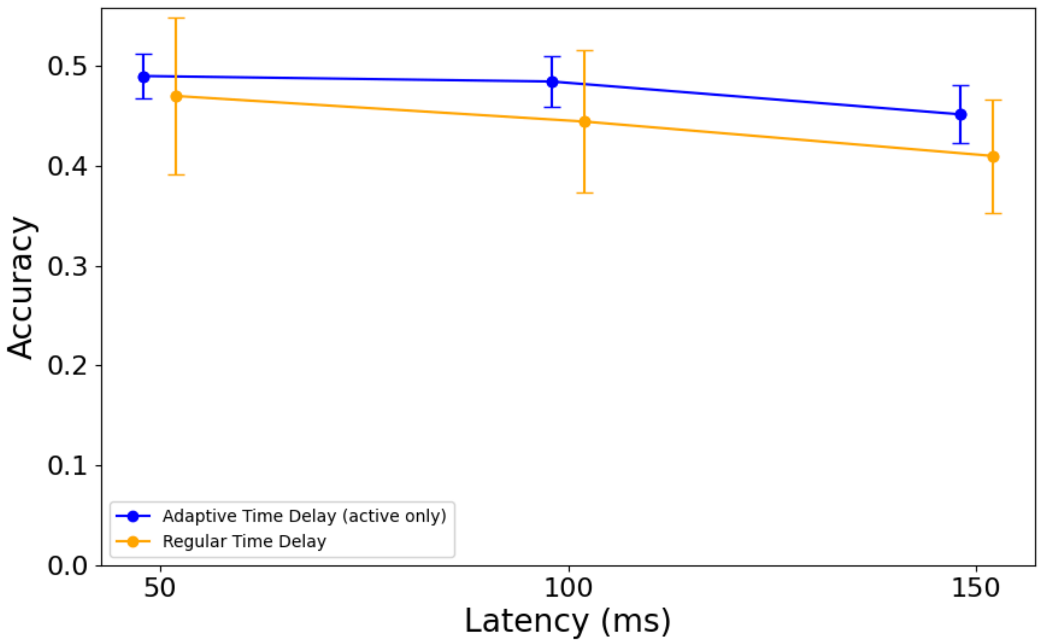


Figure 21: Adaptive TD (Active Only) vs Regular TD Accuracy Graph

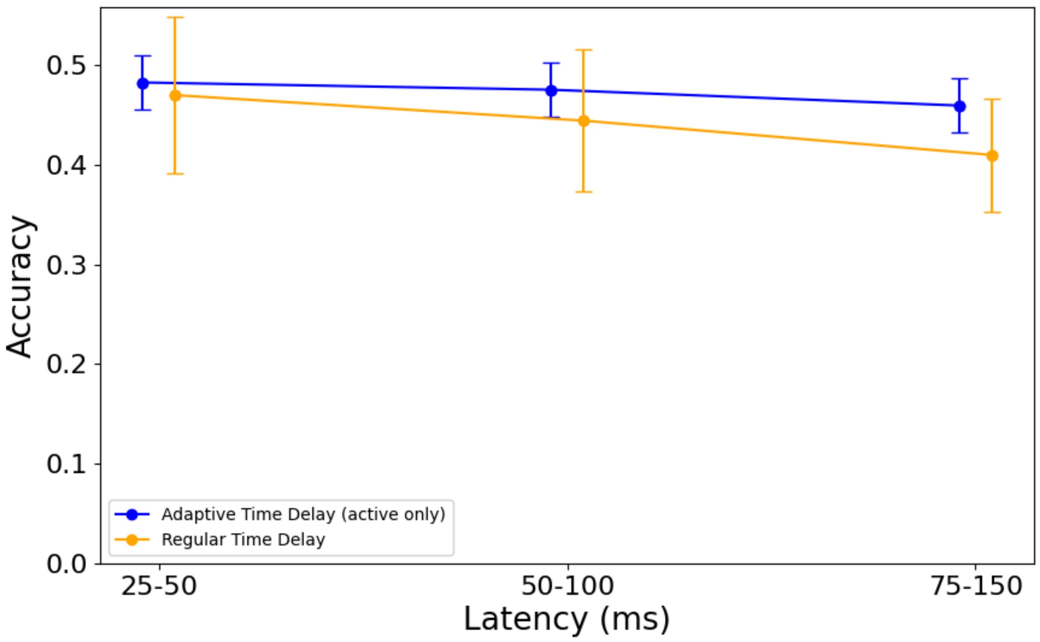


Figure 22: Base Regular TD + Adaptive TD (Active Only) vs Regular TD Accuracy Graph

Figure 19 shows the cumulative distribution of the ratio of adaptive time delay being active for adaptive time delay rounds. The figure shows that most of the round is actually played with adaptive time delay on. In Figure 20, we show two examples of how latency changes throughout an adaptive TD round. The red graph in Figure 20 represents a round where adaptive time delay switches from 0 to 150, and the purple graph shows a round where there is a base regular TD of 75, and it shoots up to 150 when adaptive TD is triggered. Figures 21 and 22 illustrate the advantages of adaptive time delay (TD) over regular time delay in terms of accuracy. In Figure 20, the blue line represents accuracy data from rounds where there was a switch from 0 latency to the value indicated on the x-axis. However, the accuracy data was collected only during instances when adaptive TD was active. On the other hand, the yellow line indicates rounds with regular TD, where latency remained constant throughout the round.

In Figure 22, the comparison is extended further. The blue line now represents accuracy values from rounds that had a base regular latency set at half the value of the adaptive latency. Again, accuracy was calculated only for instances when adaptive TD was active. These figures collectively demonstrate that the implementation of adaptive TD leads to consistently higher accuracy compared to regular TD.

4.2.2: Kills

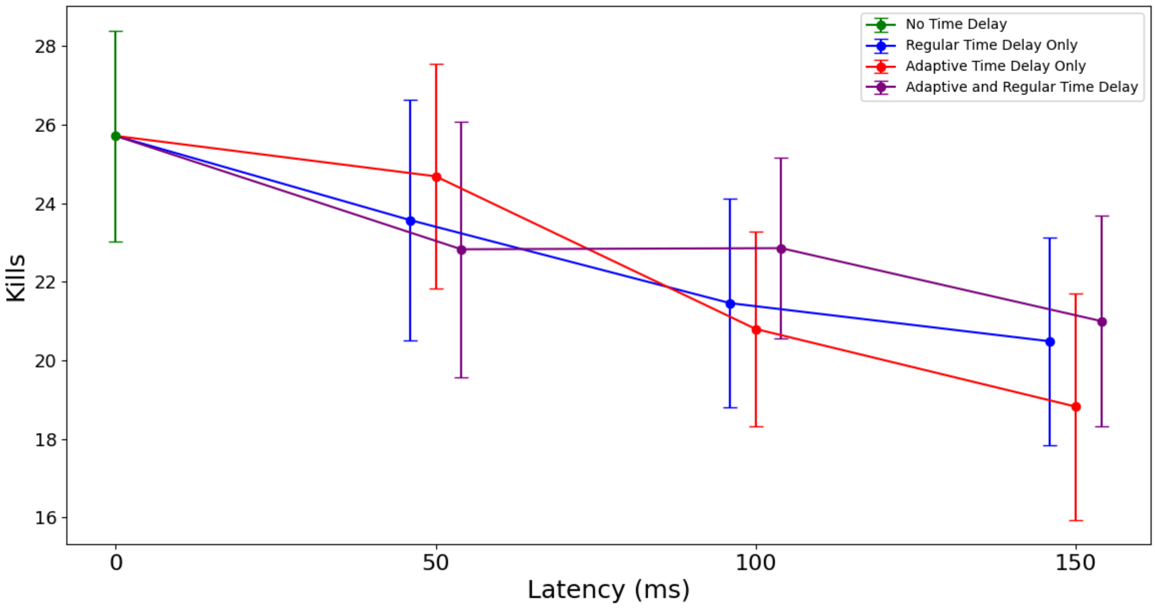


Figure 23: Kills vs Compensation Technique Graph

In Figure 23, the data shows the number of kills achieved by players in relation to the compensation technique used and the corresponding latency levels. At a 50 ms delay, the adaptive time delay with no base delay resulted in the highest average number of kills, with 25 kills per round.

For higher latency values, the adaptive delay with a base regular delay of half the latency value indicated on the graph exhibited higher average kills compared to other techniques. Specifically, at 100 ms and 150 ms latency values, this compensation technique achieved an average of around 23 and 22 kills per round, respectively.

4.2.3: Score

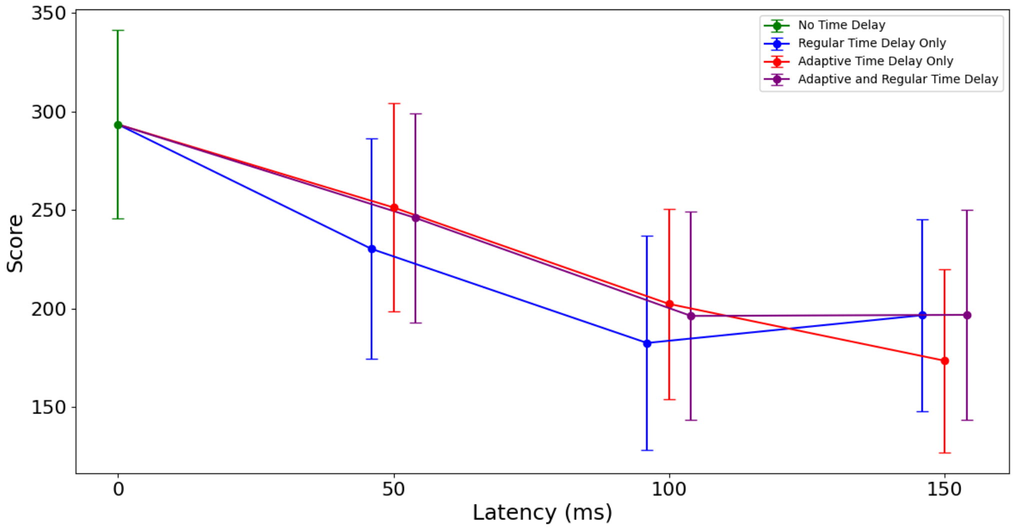


Figure 24: Score vs Compensation Technique Graph

Figure 24 shows score values versus compensation techniques with trends consistent with other user performance statistics observed in the study. The score was calculated in-game, with the value calculation involving several different variables, such as the types of zombies killed, deaths, and time alive. The implementation of regular time delay results in a lower average score compared to other techniques. Adaptive time delay (TD) alone stands as the next tier, with a higher average score. The highest average score is achieved by the combination of adaptive TD with a base regular TD.

The score statistics for both adaptive TD implementations align closely, yet the combination of adaptive TD with a base regular TD yields superior performance at higher latencies. This trend mirrors the patterns observed in the accuracy graph and the kills graph, indicating that the addition of a base regular delay to adaptive TD enhances performance under conditions of increased latency.

4.2.4: Quality of Experience

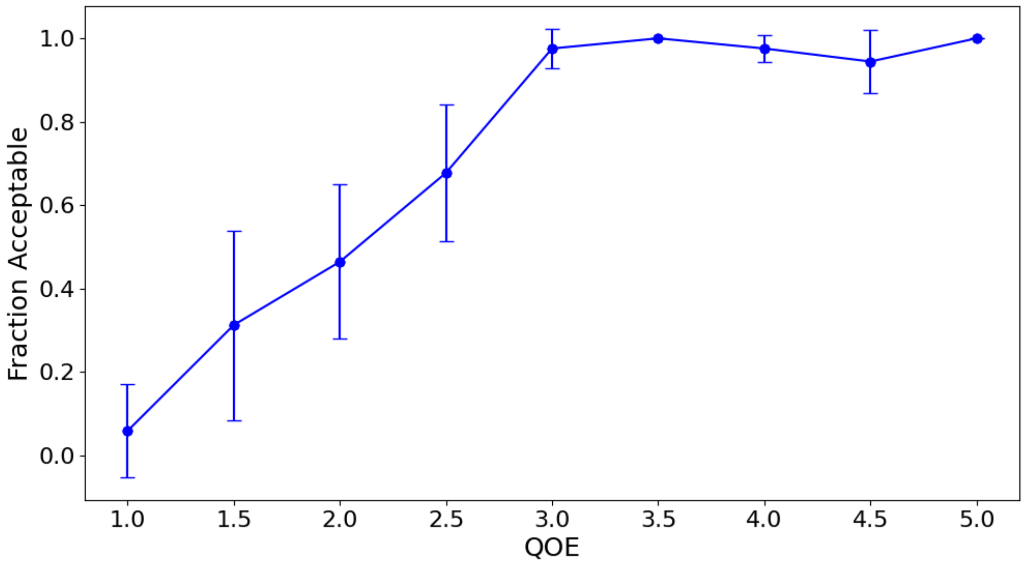


Figure 25: Acceptable Fraction vs QOE Graph

Figure 25 illustrates the relationship between the fraction of users who found each round acceptable and the quality of experience (QOE) ratings. Both of these questions were given after each round, where the user rated the experience from 1-5 on a slider and the user input yes or no for whether they thought the round was acceptable. Overall, there is more variation in acceptability on the lower end of the QOE values (1-2.5). For QOE values ranging from 1 to 2.5, there is a steady increase in the fraction of users who deemed the rounds acceptable, climbing from 0 to 0.6. This indicates that as the QOE improves from lower to mid-range values, a greater proportion of users find the gaming experience acceptable.

However, for QOE values from 3 to 5, the rounds are predominantly considered acceptable by the users. The acceptability percentages are also a lot more grouped together, forming a solid consensus for these upper values. This suggests that once the QOE reaches a certain threshold (around 3), the majority of users perceive the rounds as satisfactory, indicating a positive correlation between higher QOE ratings and user acceptance of the gaming experience.

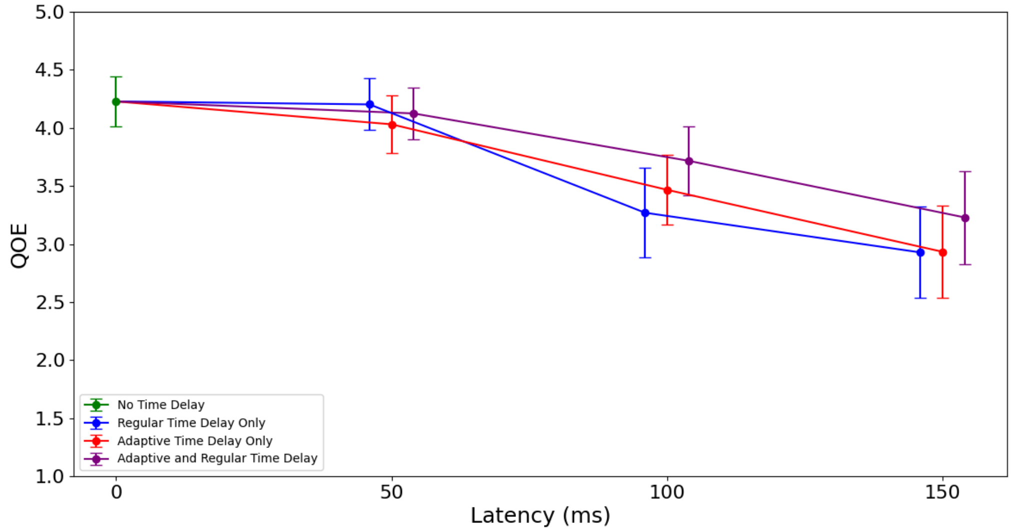


Figure 26: QOE vs Compensation Technique Graph

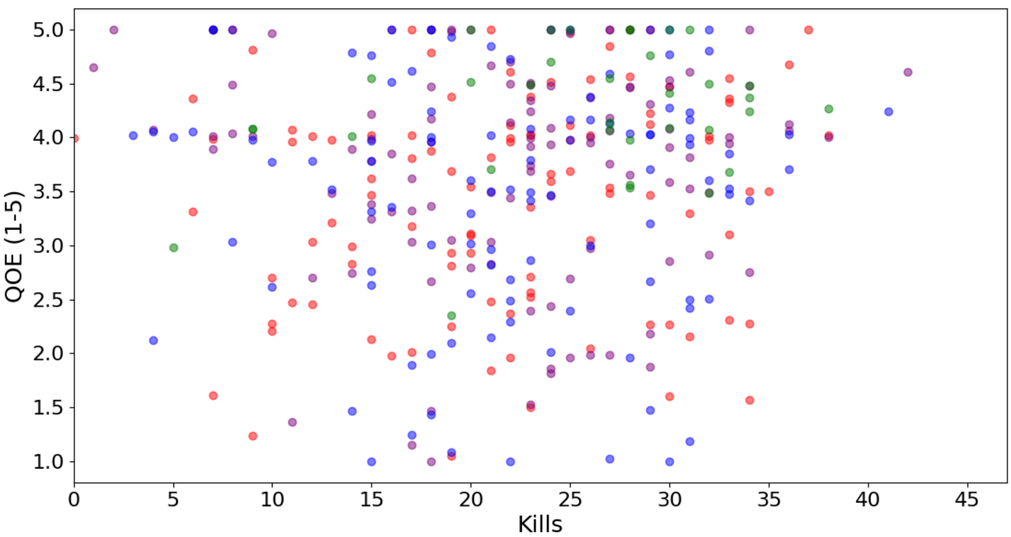


Figure 27: QOE vs Kills Scatterplot

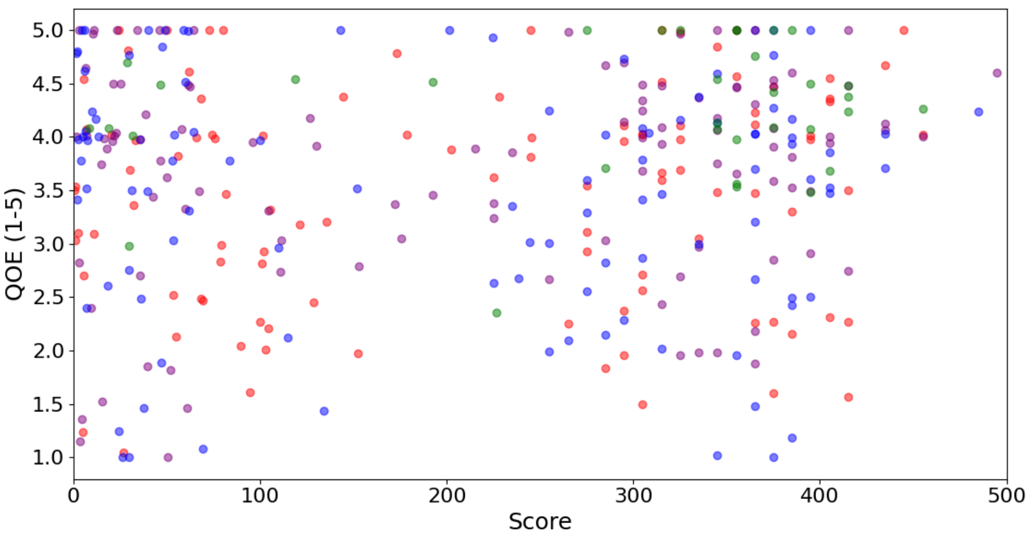


Figure 28: QOE vs Score Scatterplot

Figure 26 presents the Quality of Experience (QOE) values for different latency compensation techniques. The graph indicates that the QOE generally decreases as latency increases, which is consistent with expectations. Overall, users reported a better experience with adaptive time delay (TD) techniques compared to regular TD, with the combination of a base delay and adaptive time delay being the most preferred option.

At 50 ms of latency, regular time delay was just barely the most popular option out of all the compensation techniques, with a QOE rating of 4.3. However, at higher latencies of 100 and 150 ms, the adaptive TD with a regular TD base of half the latency value (as indicated on the x-axis) received the highest QOE ratings, with averages of 3.7 and 3.2 respectively. This suggests that while regular TD may be more favorable at lower latencies, adaptive TD techniques become more advantageous as latency increases, providing a better overall quality of experience.

Figures 27 and 28 display points of data representing rounds and their corresponding kills, quality of experience rating, and score values. In each graph, the green points represent no delay rounds, the blue points represent static regular TD rounds, the red points represent adaptive TD only rounds, and the purple points represent the rounds where there is a base regular TD and an adaptive TD of twice the base TD. Both of these graphs show no apparent visual trends at first glance. However, in the score scatterplot, it seems that there are a lot of low score rounds across all latency compensation techniques, while the grouping of data in the kills scatter plot seems to be more towards the middle of the graph.

5. Future Work

Follow-up studies could investigate the long-term effects of adaptive time delay on the FPS gaming experience. By logging player data over long periods of time, adaptive time delay can give information on how it can impact player skill development and overall experience with FPS games.

Although our study analyzed player accuracy, responsiveness, round score, and quality of experience, follow-up studies could involve other metrics to establish a better analysis on the impact of adaptive time delay. For example, other metrics could include time to kill and kill to death ratio.

Our testing environment, Zombiefield, was a single player experience, however future studies could look to develop a multiplayer mode. This would help to assess adaptive time delay's effectiveness on fairness and responsiveness while multiple players are present with different latencies.

Another factor that could help to further our understanding of the effect of adaptive time delay in FPS games is using multiple testing environments. Due to time constraints, we were only able to work with one game. However, investigating across different games would allow us to understand adaptive time delay's capabilities across different domains.

6. Conclusion

In first-person shooter (FPS) gaming, latency is a core issue that affects a player's overall experience. It impacts the responsiveness and fairness of the game when players with different network latencies meet. To address the issue, there are a lot of latency compensation techniques. For example, regular time delay applies a constant latency without taking player activity into account, while adaptive time delay applies latency only when players interact during the game.

Our study's purpose was to improve upon regular time delay by incorporating and observing the impact of adaptive time delay as a latency compensation technique. Through experimentation and analysis within Zombiefield, a custom developed game, the impact of adaptive time delay on player accuracy, responsiveness, and the quality of experience across latencies was evaluated.

For our study, we logged various player metrics while manipulating latency parameters. Experimental results show that adaptive time delay leads to improvements in player accuracy for high latency settings during gameplay. Other adaptive time delay performance statistics like kills and round scores on higher latency values proved to be better than regular time delay conditions. Furthermore, a positive correlation was established between the use of adaptive time delay and player satisfaction when compared to regular time delay. Participants reported a better and more acceptable gaming experience when adaptive time delay was applied via post-round surveys.

In conclusion, our findings in this study indicate that adaptive time delay may help improve player performance and QoE when compared to regular time delay in FPS gaming. Despite the fact that adaptive time delay was implemented only on ZombieField, which is currently a PVE (Player vs. Environment) game, it may improve the fairness of games that are PVP (Player vs. Player) with players experiencing different latencies. As FPS gaming continues to evolve, further research and experimentation could open the horizon for more ways to optimize player experiences.

7. Sources

1. Chen, L., Chen, G.-C., Chen, H., March, J., Benford, S., & Pan, Z.-G. (2007, March). An HCI method to improve the human performance reduced by local-lag mechanism. IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/8149513
2. Dacanay, R. (2023, March 28). 'Everything feels different': TenZ explains how VALORANT plays on LAN. Dot Esports. Retrieved February 20, 2024, from https://dotesports.com/valorant/news/everything-feels-different-tenz-explains-how-valorant-plays-on-lan
3. Diot, C., Profile, V., Gautier, L., & Metrics, O. M. A. (1999, July 1). A Distributed Architecture for Multiplayer Interactive Applications on The Internet. IEEE Network: The Magazine of Global Internetworking. https://dl.acm.org/doi/10.1109/65.777437#sec-cit
4. Jain, A., Bansal, R., Kumar, A., & Singh, K. D. (2015). A Comparative Study of Visual and Auditory Reaction Times on The Basis of Gender and Physical Activity Levels of Medical First Year Students. International Journal of Applied & Basic Medical Research, 5(2), 124–127. https://doi.org/10.4103/2229-516X.157168
5. Jiang, A., & Qamar, H. (2023, November). IQP testing video. YouTube. https://youtu.be/ouSVDV2TYQk?si=4b9GIHB-DZUeitqs
6. Kaiser, A., Achir, N., & Boussetta, K. (2010, July). Improving Energy Efficiency and Gameplay Fairness for Time-Sensitive Multiplayers Games in MANET. IEEE Xplore. https://ieeexplore.ieee.org/document/5503904/authors#authors
7. Li, Z., Melvin, H., Bruzgiene, R., Pocta, P., Skorin-Kapov, L., & Zgank, A. (2018, May). Lag Compensation For First-Person Shooter Games in Cloud Gaming. SpringerLink. https://link.springer.com/chapter/10.1007/978-3-319-90415-3_5
8. Lin, Y.-J., Guo, K., & Paul, S. (2003, February). Sync-MS: Synchronized Messaging Service For Real-Time Multi-player Distributed Games. IEEE Xplore. https://ieeexplore.ieee.org/document/1181396/authors
9. Liu, S., Xu, X, & Claypool, M. (2022, September 1). A Survey and Taxonomy of Latency Compensation Techniques for Network Computer Games. ACM Computing Surveys. https://dl.acm.org/doi/abs/10.1145/3519023

10. Mauve, M., Vogel, J., Hilt, V., & Effelsberg, W. (2004, February). Local-lag and Timewarp: Providing Consistency for Replicated Continuous Applications. IEEE Xplore. https://ieeexplore.ieee.org/document/1261886
11. Mauve, Martin. (2000, December 1). Consistency in Replicated Continuous Interactive Media. ACM Conferences. https://dl.acm.org/doi/10.1145/358916.358989
12. Paik, D., Yun, C.-H., & Hwang, J. (2008, May 15). Effective Message Synchronization Methods For Multiplayer Online Games with Maps. Computers in Human Behavior. https://www.sciencedirect.com/science/article/abs/pii/S074756320800054X?via%3Dihub
13. Samin, T. S. (2023, March). Enhancing Fairness using Adaptive Time-Delay For a First-Person Shooter Game. Master Thesis Proposal. Worcester Polytechnic Institute.
14. Savery, C., Graham, N., Gutwin, C., & Brown, M. (2014, February 1). The Effects of Consistency Maintenance Methods on Player Experience and Performance in Networked Games. ACM Conferences. https://dl.acm.org/doi/10.1145/2531602.2531616
15. Stuckel, D., & Gutwin, C. (2008, November 1). The Effects of Local Lag on Tightly-Coupled Interaction in Distributed Groupware. ACM Conferences. https://dl.acm.org/doi/abs/10.1145/1460563.1460635
16. Wikstrand, G. (2005, August 30). High and Low Ping and The Game of Pong. Academia.edu. https://www.academia.edu/31035728/High_and_Low_Ping_and_the_Game_of_Pong
17. Yusuke , H., Shinji , S., Norishige , F., & Yutaka , I. (2012, November). Adaptive Delta-Causality Control with Prediction in Networked. IEEE Xplore. https://ieeexplore.ieee.org/document/6388192
18. Zander, S., Leeder, I., & Armitage, G. (2005, June 1). Achieving Fairness in Multiplayer Network Games Through Automated Latency Balancing. ACM Conferences. https://dl.acm.org/doi/10.1145/1178477.1178493
19. Zhang, Y., Chen, L., & Chen, G. (2006, October 1). Globally Synchronized Dead-Reckoning with Local Lag for Continuous Distributed Multiplayer Games. ACM Conferences. https://dl.acm.org/doi/abs/10.1145/1230040.1230071

8. Appendices

8.1 Demographic Survey

1) What is Your Academic Level?

A) High School
B) Undergraduate
C) Graduate
D) Other

2) Gender (select other if you prefer to self-identify)?

A) Male
B) Female
C) Non-Binary
D) Prefer not to answer
E) Other

3) How old are you?

4) How many years have you been playing FPS games?

5) Which games have you played?

A) CS:GO
B) Overwatch
C) Valorant
D) Apex Legends
E) Call of Duty
F) Rainbow Six Siege
G) None of the above
H) Other

6) What rank(s) are you in the game(s) you selected

In the format:

(game name : rank)

(game2 name : rank) …

7) Are you a competitive or casual player?

A) Competitive

B) Casual

8) What is your general gaming skill level (1-5)?

9) What is your FPS gaming skill level (1-5) ?

10) What platform(s) do you primarily use for playing FPS games?

A) PC

B) PC with Controller

C) Console (e.g., Xbox, PlayStation)

D) Mobile

E) Other

11) What type of internet connection do you primarily use for online gaming?

A) Wired (Ethernet)

B) Wi-Fi

8.2: Post Round Survey

1) On a scale from 1-5, how was your experience?

1 2 3 4 5

I——————————————————————I

2) Was the round acceptable?

Yes No

I——————————————————————I

8.3: IRB Approval Letter

WORCESTER POLYTECHNIC INSTITUTE

100 INSTITUTE ROAD, WORCESTER MA 01609 USA

Institutional Review Board

FWA #00030698 - HHS #00007374

Notification of IRB Approval

Date:	20-Nov-2023
PI:	Mark L Claypool
Protocol Number:	IRB-24-0195
Protocol Title:	First-Person Science - Exploring FPS Games and Latency
Approved Study Personnel:	Jiang, Ao~Cannon, James M~Claypool, Mark L~Qamar, Hanzalah~Dinasarapu, Saketh R~
Effective Date:	20-Nov-2023
Exemption Category:	3
Sponsor*:	

The WPI Institutional Review Board (IRB) has reviewed the materials submitted with regard to the above-mentioned protocol.We have determined that this research is exempt from further IRB review under 45 CFR § 46.104 (d). For a detailed description of the categories of exempt research, please refer to the IRB website.

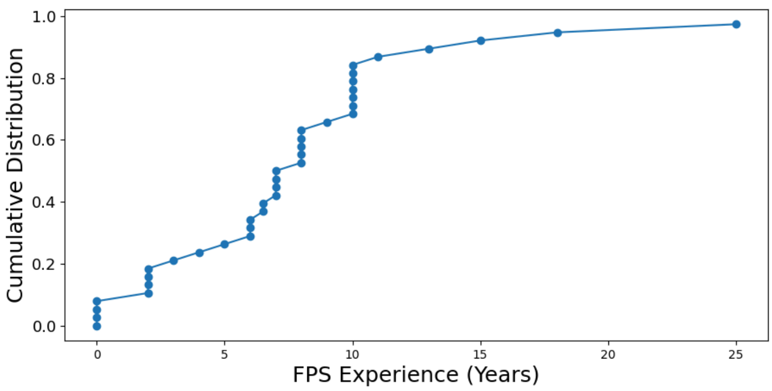
The study is approved indefinitely unless terminated sooner (in writing) by yourself or the WPI IRB. Amendments or changes to the research that might alter this specific approval must be submitted to the WPI IRB for review and may require a full IRB application in order for the research to continue. You are also required to report any adverse events with regard to your study subjects or their data.

Changes to the research which might affect its exempt status must be submitted to the WPI IRB for review and approval before such changes are put into practice. A full IRB application may be required in order for the research to continue.

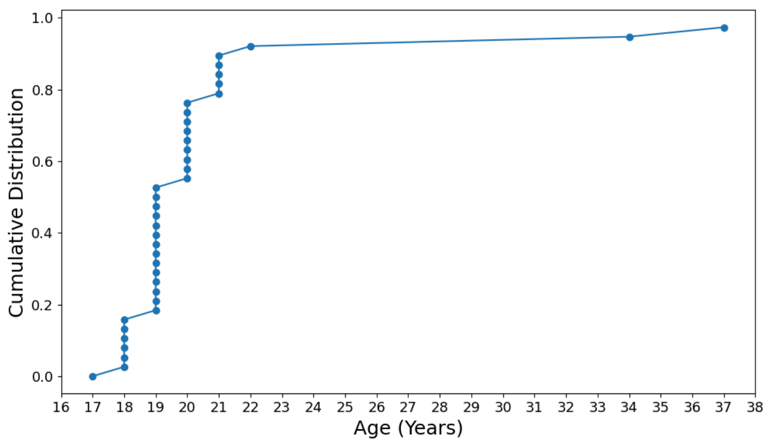
Please contact the IRB at irb@wpi.edu if you have any questions.

*if blank, the IRB has not reviewed any funding proposal for this protocol

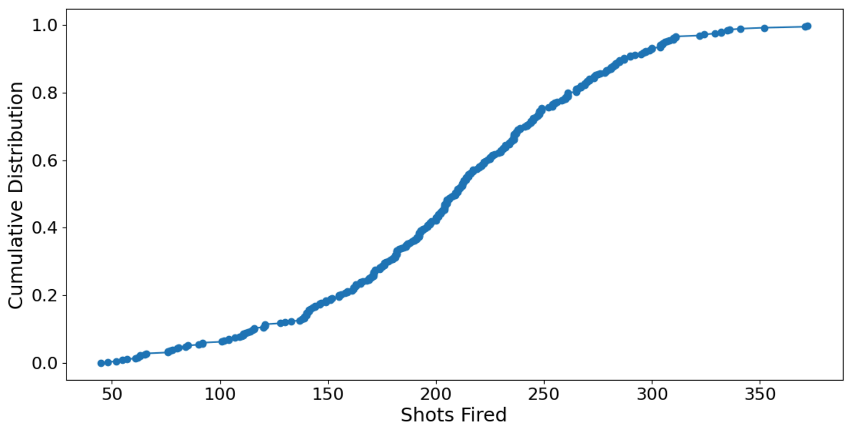
8.4: Extra Graphs



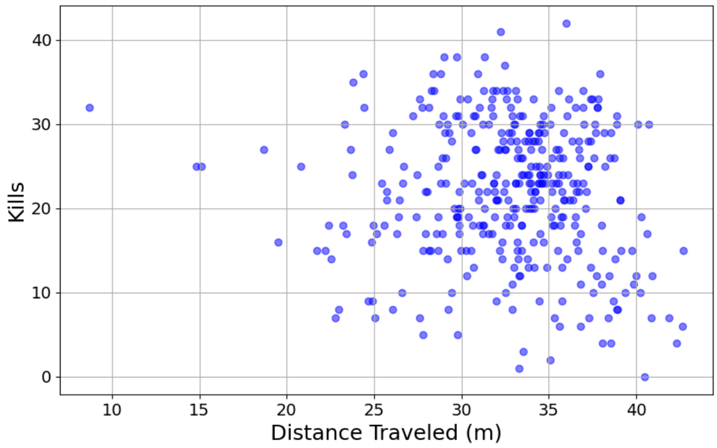
Years of FPS Experience CDF



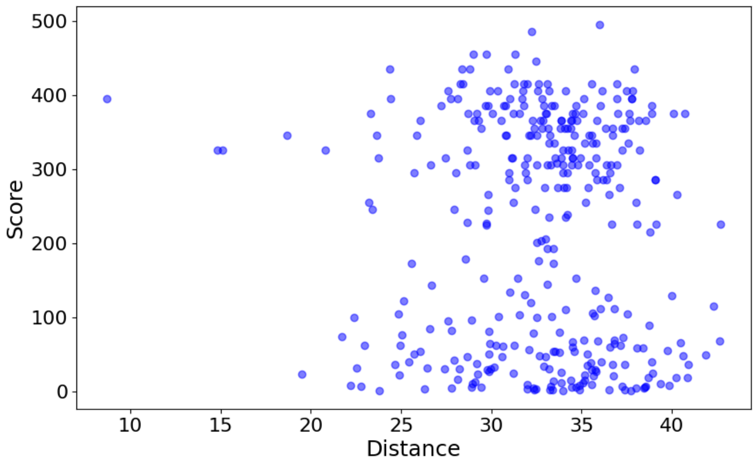
Age CDF



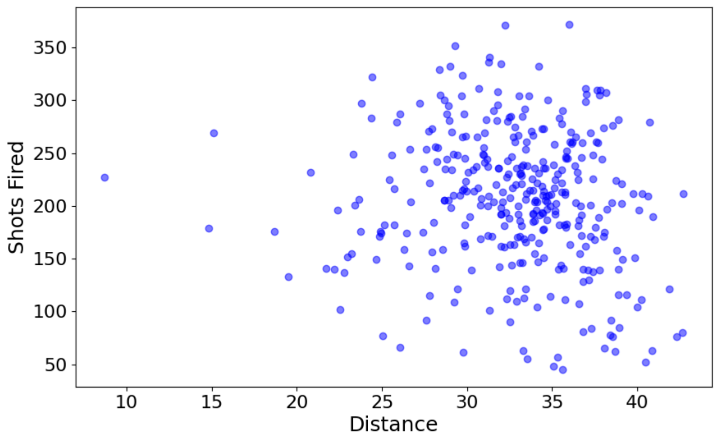
Shots Fired CDF



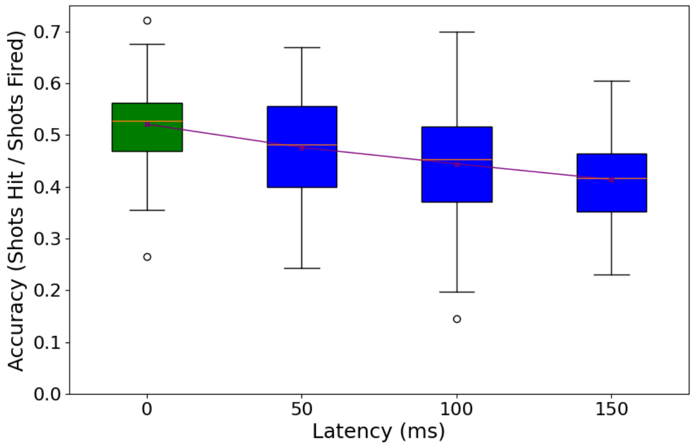
Kills Vs Distance Traveled Scatterplot



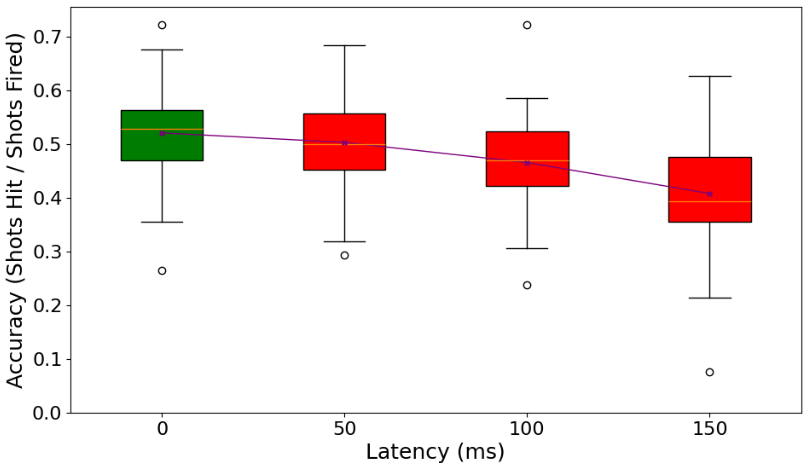
Score Vs Distance Traveled Scatterplot



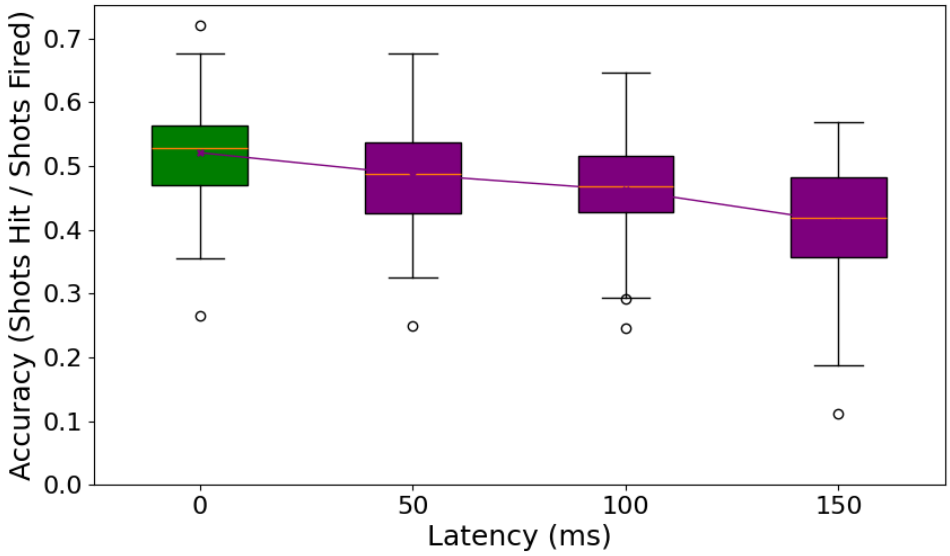
Shots Fired vs Distance Traveled



Accuracy Boxplot Regular TD



Accuracy Boxplot Adaptive Only



Accuracy Boxplot Adaptive + Base TD