WORCESTER POLYTECHNIC INSTITUTE

Master's Thesis

Compensating for Network Latency in Cloud-based Game Streaming using Control Assistance in a Top-down Game

By: Bharat Monavarti & Surya Srinivasan

A thesis submitted in fulfillment of the requirements for the degree of
Master of Science

In

Interactive Media and Game Development

April 2024

APPROVED:

____

Professor Mark Claypool, Thesis Advisor

____

Professor Charles Roberts, Reader

Abstract

Cloud game streaming shifts game processing to servers, enabling gameplay on low-powered thin clients. This innovation alleviates the need for high-end hardware by the player and streamlines game development for a unified cloud platform. However, cloud gaming faces a critical challenge: network latency. Traditional latency compensation techniques for multiplayer games are unsuitable for thin clients, sparking research for innovative server-based solutions. This research focuses on using control assistance techniques to mitigate network latency effects for cloud-based game streaming in top-down games from the client to the server and back. We designed and implemented 'Spectres,' an isometric bullet-hell game with simulated latency conditions to evaluate network latency effects in a controlled environment. We integrated multiple control assistance techniques to assess player Quality of Experience (QoE) when exposed to latency. The result showed that aim assistance significantly improved scoring performance, survivability, and shooting accuracy in cloud-based gaming environments, highlighting its efficacy in mitigating latency-induced delays. Additionally, movement assistance showed positive effects on survivability, particularly at moderate latency levels, underscoring the potential of control assistance techniques in enhancing the overall gameplay experience.

Table of contents

List of Tables

List of Figures

1. Introduction

Video games have evolved from arcade machines to cartridges to digital copies and now cloud gaming. With advancements in cloud computing technology of streaming game frames as video to the client, games can now be run on the server allowing the game to be played remotely through clients that are not required to be graphically or computationally powerful.

In recent years, the global cloud gaming market has witnessed remarkable growth, with projected revenue for the year 2023 to reach USD $4.3 billion, etc. This growth is expected to continue with a Compound Annual Growth Rate (CAGR) of 44% for the next 4 years. Moreover, large technology companies and game developers have made substantial investments in cloud gaming platforms, with examples including Microsoft's Xbox Cloud Gaming, NVIDIA's GeForce NOW, Sony’s PlayStation Plus and Amazon’s Luna.

The major idea in cloud game streaming is to offload the work-intensive part of rendering the graphics and game logic to the server while the user’s client registers input and receives video streaming data and displays it. This reduces the need for graphics-intensive hardware on the user side and also creates a single target platform for developers instead of multiple platforms.

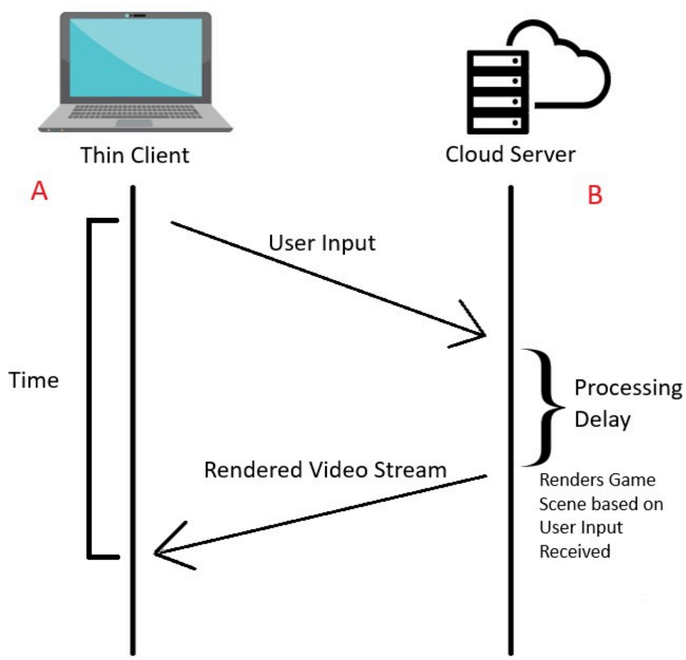

Figure 1: Cloud Game Streaming Architecture

Figure 1 depicts the basic cloud game streaming architecture where the cloud server receives user input details from the user (Thin Client). The Cloud Server then computes and renders the game scene according to the inputs received. Rendered frames are sent back to the Thin Client in the form of a video stream, which is decoded and displayed to the user.

Cloud game streaming adds network latency to all player interactions (A to B and back to A) in Figure 1. Player input (e.g shooting, walking) needs to be sent from the client to the server which must compute and return the video data to be displayed on the client.

Although network latency issues are present in traditional multiplayer games, they often can be mitigated using compensation techniques deployed on both the user and server side. In cloud game streaming however, many traditional latency compensation techniques are not suitable since the clients are thin, and hence cannot run computing-intensive tasks or modify the game. What is needed are techniques that can be implemented on the server alone [2].

Control assistance methods have historically been applied primarily in non-cloud gaming environments [4]. Some commercial games that implement some of these techniques (Bullet Magnetism) are Halo and Destiny. They serve the purpose of enhancing player accessibility or addressing limitations associated with joystick movements in conventional game controllers. These methodologies dynamically adjust player attributes in real-time, mitigating the precise control challenges posed by player actions. We propose to adapt these techniques to cloud game streaming.

For this research, we designed and implemented Spectres, a bullet-hell game deployed in a cloud gaming platform with built-in Control Assistance techniques to compensate for latency. We designed and conducted a user study where participants played short rounds of Spectres with different amounts of latency and different types of control assistance. We measured players' Quality of Experience (QoE) and performance when latency is induced between the client and server and compared with when the game is played without our latency compensation techniques.

In the subsequent chapters of this thesis, we delve into a comprehensive exploration of the key aspects surrounding cloud-based gaming and the development of latency compensation techniques. Chapter 2 reviews related work in the field, examining existing research and methodologies employed in addressing latency issues in gaming. Chapter 3 provides a detailed account of the implementation process, elucidating the design and development of "Spectres" and the integration of control assistance techniques. Following this, Chapter 4 presents the evaluation of our proposed techniques, detailing the testing methodologies employed, the data collected, and the analysis of the results obtained. Chapter 5 offers a conclusion, summarizing the findings of this research and discussing their implications. Lastly, Chapter 6 outlines potential avenues for future research and development in the field, identifying areas for further exploration and improvement.

2. Related Work

Claypool et al. describe how network delays significantly impact gameplay [5]. To address this, the paper introduces a classification system based on the player control and camera perspectives. Using the "precision" and "deadline" of player actions, the paper shows how different game genres have different sensitivity to latency in traditional multiplayer games.

The above is further expanded upon by Sabet et al. [1] introducing two new characteristics which are "predictability", the ability of the player to predict the upcoming events in game and the "number of required actions in the game". Using these characteristics, they concluded that Spatial Accuracy (precision), Temporal Accuracy (deadline), and predictability were the most effective in classifying the influence of delay on QoE.

Liu et al. provide a survey of research done in the area of latency compensation [2]. The paper provides a taxonomy, dividing the various techniques into 4 broad categories, namely: Feedback, Prediction, Time Manipulation and World Adjustment. They provide further details on how each works and how or where they can be implemented in a networked gaming architecture.

Based on the taxonomy above, we focus on techniques that can be deployed on the server side for cloud game streaming including World Adjustment. Ku et al. describe the usage of scaling different attributes within the game world that could potentially make the game easier counteracting the difficulty added by latency [3].

In [4],[6],[8], researchers explain different Control Assistance techniques for aiming. From these we infer that certain gameplay elements greatly affect how useful an aim-assistance technique can be. Techniques such as sticky targets and gravity were not popular with gamers due to the visual movement and stickiness of the reticle. Which hinders the player's intention while moving the reticle. Other techniques like target lock made the game too easy to play and could potentially degrade the player's experience with latency.

For this research we choose bullet magnetism, an approach where there is minimal visual feedback hindering the player's experience.

3. Methodology

In order to develop and test our latency compensation techniques, we have done the following:

1. Designed and implemented Spectres, a bullet-hell game
2. Implement control assistance for movement and shooting using
3. Used timers to induce latency to test the implemented techniques
4. Designed and executed a user study to assess performance and QoE.
5. Analyzed the results

3.1. Spectres

Spectres is an action game specifically part of the shoot-em up sub genre called bullet-hell, developed using Unreal Engine 5. Spectres features a top-down view of the level, with two dimensional movement for the player.

Created in Unreal Engine 5, it features assets taken from the Unreal Store, Audio/Music from free sources online.

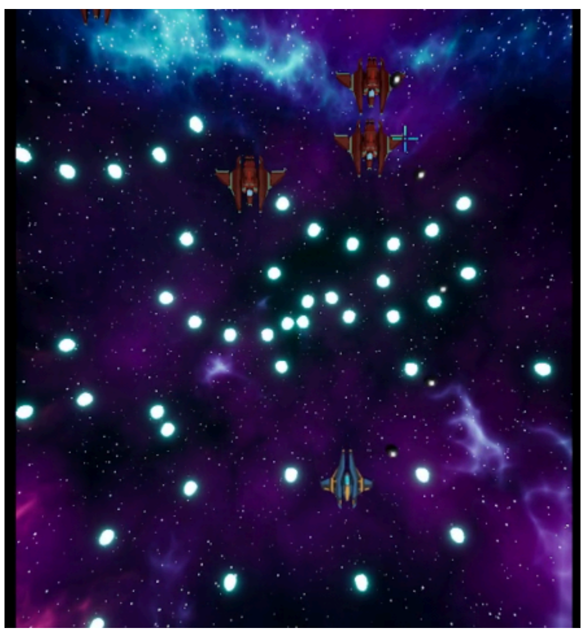

Figure 2: Spectres

The games main features are:

1. Movement:

   Players can move their ship across 2-dimensions using WASD keys, allowing for fast movement in a confined space, a requirement for dodging bullets.

2. Shooting:

   Players have to use the mouse to manually aim at the enemies to shoot, by clicking, in the direction of enemies.

3. Bullet-Hell

   The main focus area of Spectres is its bullet-hell mechanics. This includes various opponents with unique bullet patterns that require precise dodging movement from the player to maneuver through these attacks. Latency during these moments is expected to cause the players to lose health making for a poor experience.

   We developed a Bullet Hell Generator that can generate different patterns of bullets ranging like lines, circles, spirals and more. We attach this generator to different enemies to create unique patterns, allowing for tweaks to be made for different variables such as fire rate, Number of bullets, Lock on Player.

   With the focus on bullet hell, we implemented a movement assistance technique to help players dodge these bullets when in latency.

3.2. Control Assistance

Our method of latency compensation uses Control Assistance. Control Assistance is the implementation mechanism to aid players in controlling or interacting with the game more effectively. Examples include Aim-Assist [4] and Assisted Steering[2].

For this research, we have focused on two areas of bullet hell games that are the most affected during latency. They are:

1. Shooting

   Shooting is one of the most prominent actions in Bullet Hell games. With the player constantly having to aim and shoot at enemies, even small amounts of latency may lead to the player missing their shots.

   To overcome this we implemented Bullet Magnetism. Bullet Magnetism is a gameplay mechanic implemented in "Spectres" designed to enhance player accuracy and target acquisition. When a player fires a projectile, the Bullet Magnetism feature dynamically adjusts the trajectory of the bullet, causing it to gradually bend towards the nearest target to the reticle. This adjustment creates the illusion of the bullet being attracted to the target like a magnet, thereby increasing the likelihood of hitting the intended target even if the player's aim is slightly off.

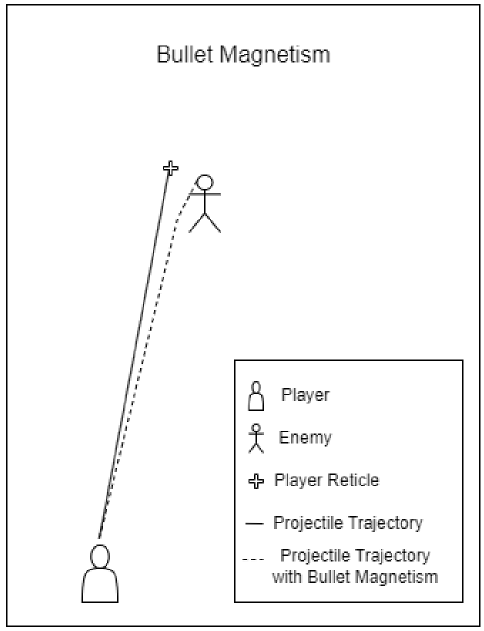

Figure 3: Bullet Magnetism

In Figure 3, the player's reticle is represented by the central crosshair. Surrounding the reticle are several enemy targets indicated by circles. As the player fires a projectile (represented by the straight line), the Bullet Magnetism feature influences the trajectory of the bullet, causing it to curve towards the nearest enemy target. This bending effect helps the player land shots more effectively, especially when aiming in the vicinity of multiple targets.

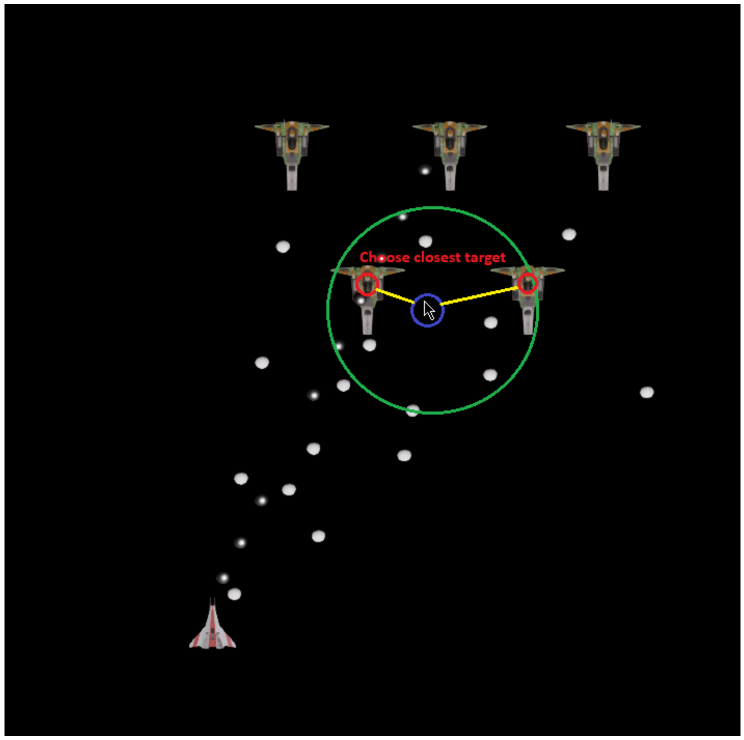

Figure 4: Bullet Magnetism in Spectres

2. Movement

Movement is a fundamental aspect of gameplay in Bullet Hell games where players engage in dynamic maneuvers to evade enemy fire. However, the presence of latency can introduce challenges, potentially resulting in players being hit by enemy bullets and sustaining damage. To mitigate this risk, we have implemented a technique known as Force-Based Assistance Steering.

Force-Based Assistance Steering functions by providing players with additional assistance during movement, particularly in scenarios where latency may impact responsiveness. This technique operates by applying a pushing force to guide the player's movement, directing them towards small, safe spaces within the game environment that are determined based on the proximity of incoming enemy bullets.

Inspired by the principles of the Boids flocking algorithm [9], our approach emulates the natural behavior of flocking entities by steering the player's ship away from nearby bullets, thus guiding them towards minuscule pockets of safe space. By leveraging this technique, players can navigate through challenging situations with greater precision and confidence, enhancing their overall gameplay experience and minimizing the detrimental effects of latency on movement responsiveness.

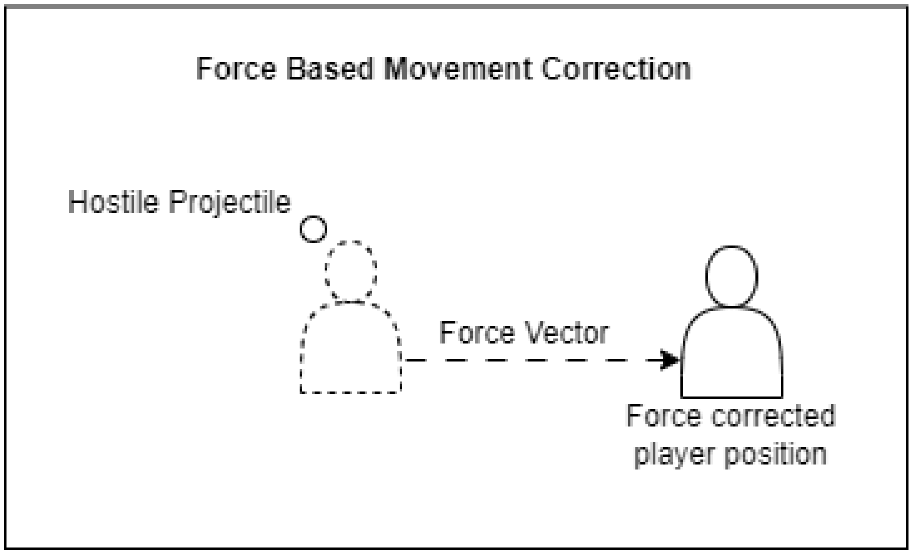

Figure 5: Movement Assistance

In the diagram, the player's ship is represented by the dotted lines. A projectile approaches from the left, posing a threat to the player's current position. To assist the player in evading the projectile, a force vector (indicated by an arrow) is applied to the right, directing the player towards a corrected position, denoted by the straight line. This corrective movement helps the player avoid potential damage from the incoming projectile while maintaining their momentum and strategic positioning within the game environment.

Scoring Formula:

The scoring mechanic in "Spectres" was designed to provide players with a comprehensive evaluation of their performance in each round. The final score was computed using the following formula:

Final Score = *Base Score* × *Multiplier* + *Additional Points*

Where,

Base Score: The base score is calculated as the sum of various components:

100×Elapsed Time (seconds): 100 points added for each second elapsed.

1000×Number of Enemies Killed: 1000 points added for every enemy killed.

5000: Additional 5000 points if the boss enemy was killed.

Multiplier: The multiplier is determined based on how quickly the player completes the round and is calculated using the formula:

$$\text{Multiplier} = \frac{4}{10^{\frac{t}{50}}} + 1$$

Where,

t is the time taken by the player to complete the round.

Additional Points: Additional points are added based on specific actions and performance metrics:

−2000×Number of Deaths: Deduct 2000 points for each death.

5×Shots Hit: Multiply the number of shots hit by 5.

5000×Shots Fired/Shots Hit : Multiply the ratio of shots fired to shots hit by 5000 (Shot accuracy).

The final score obtained provides a comprehensive assessment of the player's performance, considering factors such as time efficiency, accuracy, and overall effectiveness in combat.

3.3. Data Collection

During the testing phase, data collection was facilitated by a logger integrated into the game, which recorded various aspects of player performance in each round. This logger outputs the collected data to a .csv file for further analysis.

The data logged included:

- Score: The cumulative score achieved by the player throughout the round.
- Health: The remaining health points of the player at the end of the round.
- Death: Whether the player died during the round (binary).
- Distance Traveled: The total distance covered by the player's ship during the round.
- Time Taken: The time taken by the player to defeat each phase of the round.
- Shots Fired: The total number of shots fired by the player during the round.
- Shots Hit: The number of shots successfully hitting enemy targets during the round.
- Questionnaire Answers: Responses provided by the player in the questionnaire administered at the end of each round to gauge their overall experience.

3.4. Evaluation

To evaluate the gameplay experience of "Spectres" under various latency conditions and control assistance settings, we are testing with a user study. The testing was done with IRB approval and participants signed a consent from before beginning the testing. Participants were compensated with playtesting credits for some courses and were entered into a raffle for a chance to win a $25 Amazon gift card. Each study lasted roughly 25 min and participants could stop any time or could take breaks if needed.

The configuration for testing included input lags of 60, 120, and 180 milliseconds with aim assistance and movement assistance toggled both on and off, resulting in a total of twelve distinct testing scenarios shown in Table 1. Each scenario was tested over 25 rounds, comprising 2 rounds of each variation with the three latencies, one round with 0 latency and no assistance, and one practice round at the beginning to familiarize participants with the game mechanics. The rounds are shuffled

Latency(ms)	Aim Assistance	Movement Assistance
0	OFF	OFF
60,120,180	OFF	OFF
	OFF	ON
	ON	OFF
	ON	ON

Table 1: Testing Configurations

Each participant engaged in gameplay sessions conducted on an NVIDIA 4070 PC in a controlled laboratory environment equipped with a 500Hz, 1080p monitor. During each 45-second round, players encountered two phases of small enemies followed by a challenging boss enemy with complex bullet patterns. After completing each round, participants were prompted to provide feedback by responding to a set of four questions designed to gauge their overall quality of experience.

The Questions asked were:

1. How difficult was the round?
2. How did ship movement feel?
3. How did aiming feel?
4. How was the lag?

Each question could be answered using a slider that ranged from 1.0 (Bad, Hard) to 5.0 (Good, Easy).

4. Results

The results of the user study are presented as graphs that compare different metrics with no assist, only aim assist, only movement assist and both assists turned on. We analyze the effectiveness of the implemented control assistance techniques.

4.1. Participant Demographics

A cohort of 20 university students with diverse gaming backgrounds participated in the evaluation. All participants ranged between the ages of 18 and 34. They all possessed moderate to extensive experience playing video games and were comfortable using keyboard and mouse controls. Additionally, most participants had moderate experience playing bullet hell-type games or shoot-'em-ups, or games of similar genres.

Figure 6 shows the age distribution of the demographics of our participants. With most participants ranging between the ages of 19 - 25.

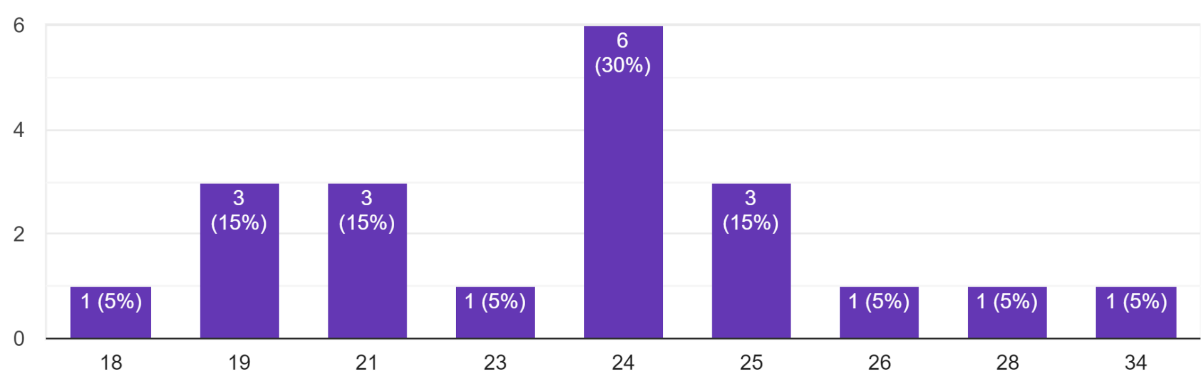

Figure 6: Age of Participants

When asked about experience with bullet hell and similar games, as seen in Figure 7, most players had occasionally played similar games and a few participants were well acclaimed with games of the genre.

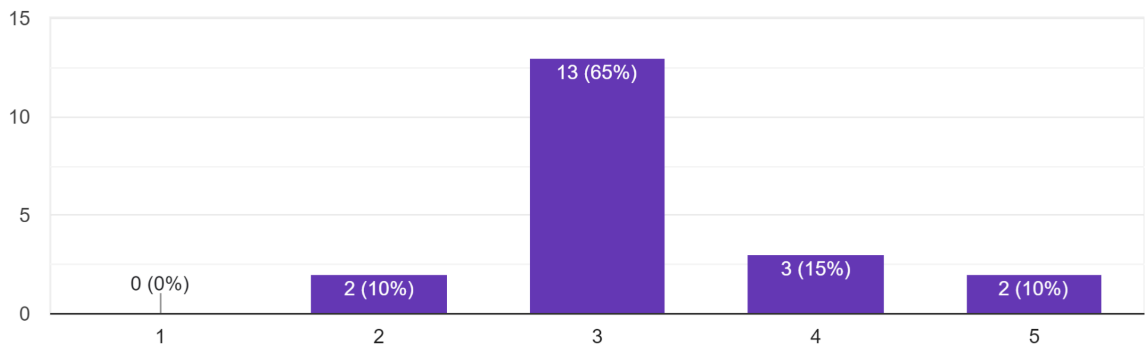

Figure 7: Experience of Participants with Similar Games

4.2. Score

Figure 8 illustrates the cumulative distribution of the scores across all latencies. Scores are earned by defeating the enemies that appear in waves of 5 each. A bonus score is added based on the accuracy of the player shots that are targeted at the enemies. If the player finishes the 2 waves and the final boss encounter before the round timer is up, a round end bonus is also added to the score. Aim assistance performed the best in scoring with both assists turned on also performing relatively well. The movement assist on the other hand did not show much improvement to the score. We could attribute this to the fact that our scoring methodology is heavily skewed towards precision of player targeting.

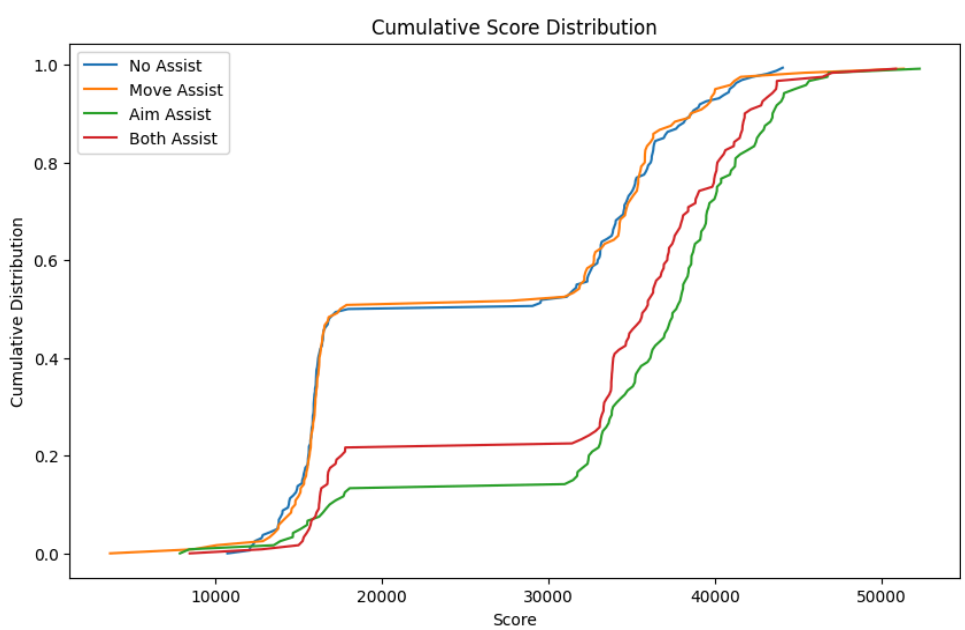

Figure 8: Distributions of Player Score

4.3. Player Deaths

Figure 9 illustrates how the assistance techniques help the players to avoid damage and in turn avoid death (when player health reaches 0). The mean deaths are calculated at different latencies bounded by 95% confidence intervals. Movement assistance helps players in avoiding deaths but only up to 120 ms.

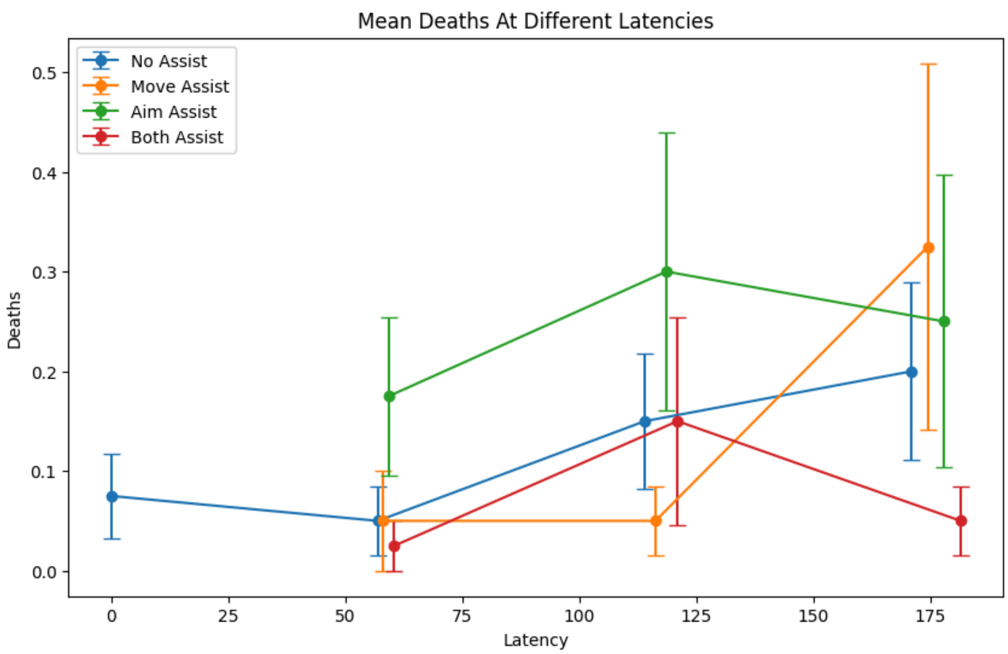

Figure 9: Mean Deaths vs Latency

4.4. Win Rate

Figure 10 illustrates the win rate at different latencies. The win rate is determined if the player has successfully defeated the boss encounter before the round timer reaches 0. As shown by the results, aim assist helps with shot accuracy, improving the players ability to clear the round. While having both assists also improved the win rate, aim assistance has a more pronounced impact on the results.

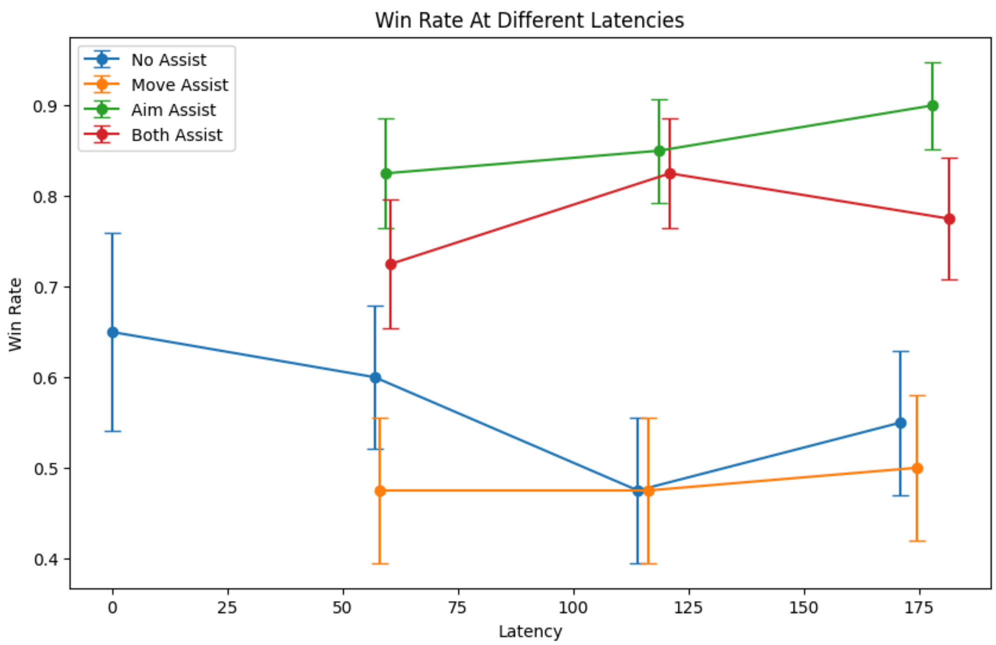

Figure 10: Win Rate vs Latency

4.5. Shooting Accuracy

Figure 11 illustrates the shooting accuracy of the player. As seen in the results, aim assist again had an impact on player performance. From the results we can infer that bullet magnetism is effective in helping players offset the latency in pointing tasks that require precesion.

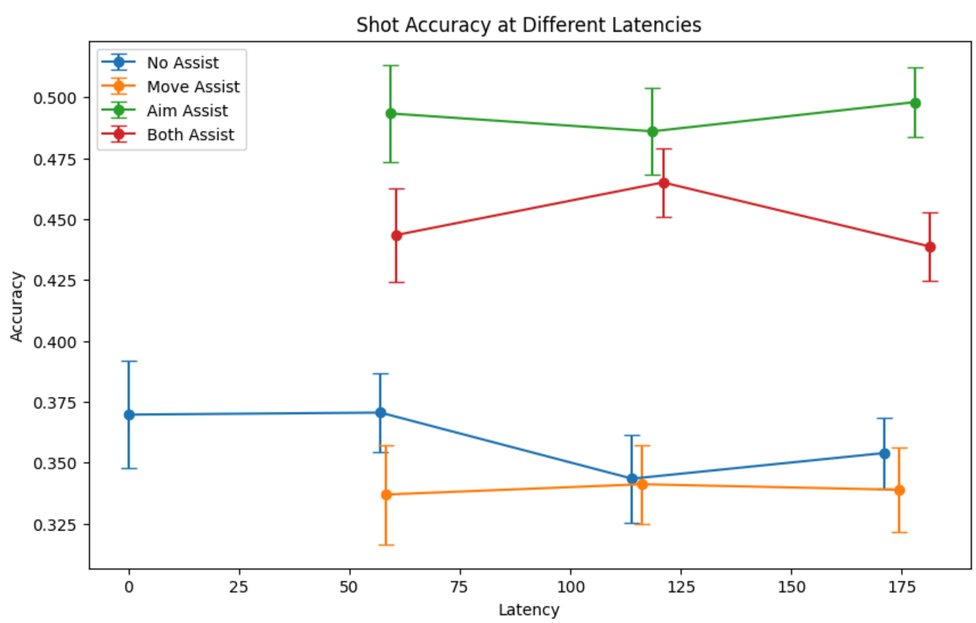

Figure 11: Shot Accuracy vs Latency

4.6. Quality of Experience (QoE)

Figure 12 is a distribution of the user feedback on the QoE of each round over different latencies. From the results, both assistance techniques improved the experience for the players. However, aim assistance had a relatively bigger impact on the player experience under the effects of latency.

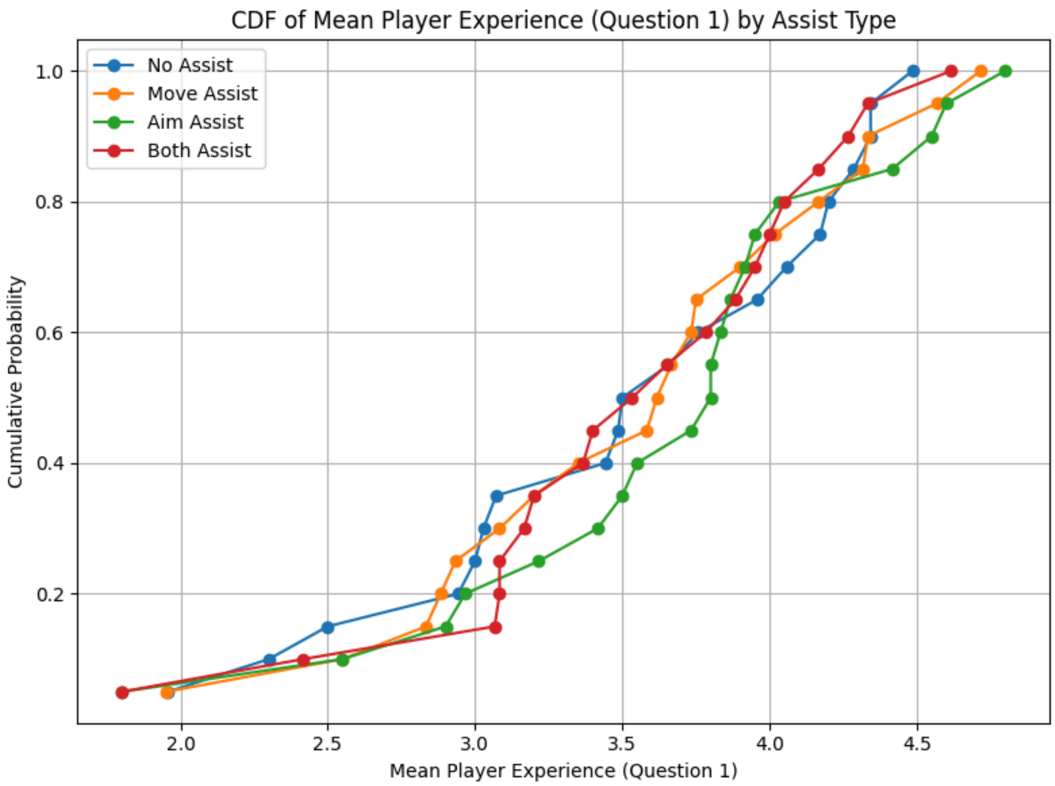

Figure 12: Distribution of Player Experience (QoE)

5. Conclusion

In cloud-based gaming, network latency poses a significant obstacle to delivering seamless and engaging gameplay experiences. This thesis tackled this challenge by investigating the efficacy of control assistance techniques in enhancing the gameplay experience of "Spectres", a bullet - hell game across various latency conditions.

Our study aimed to address the impact of latency on player performance and enjoyment, as well as explore the potential of movement and aiming control assistance techniques in mitigating latency-induced delays. To achieve this, a user study was conducted involving 20 university students with diverse gaming backgrounds using our custom game “Spectres”.

Our analysis of the study results revealed several key findings. Aim assistance is effective in enhancing scoring performance, survivability, win rate, and shooting accuracy, showcasing its effectiveness in compensating for latency-induced delays. Additionally, movement assistance demonstrated positive effects on survivability, particularly at moderate latency levels. The combined effect of aim and movement assistance also yielded promising results, albeit to a lesser extent compared to aim assistance alone.

In conclusion, the findings from this study show the potential of integrating control assistance techniques, particularly aim Assistance, into cloud gaming platforms to enhance player experience and engagement. By leveraging these techniques, game developers can help overcome the challenges posed by latency and provide players with enjoyable gaming experiences. Looking ahead, future research could explore novel control assistance mechanisms and their integration into cloud gaming architecture.

6. Future Work

Building upon the foundations established in this project, there are several avenues for further exploration and development, focusing on short-term and medium-term objectives.

Short Term:

Refinement of control assistance techniques: Immediate efforts can be directed towards refining control assistance techniques, particularly aim assistance and movement assistance. A key focus could be on scaling assistance based on latency levels to optimize performance. This entails calibrating assistance mechanisms to strike a balance between effectiveness and responsiveness, ensuring that movement assistance remains beneficial for moderate latency levels while fine-tuning aim assistance to avoid over-tuning.

Medium Term:

Integration with cloud gaming platforms: A medium-term goal could be to explore the integration of control assistance techniques into existing cloud gaming platforms. This entails collaborating with cloud gaming service providers to implement and evaluate the effectiveness of these techniques in real-world gaming environments. By seamlessly integrating assistance mechanisms into cloud gaming infrastructures, we could enhance accessibility and adoption of latency compensation solutions among a wider audience of players.

Long Term:

Adaptive assistance mechanisms: Finally, a long-term objective is the development of adaptive assistance mechanisms that dynamically adjust to individual player preferences and skill levels. This involves leveraging machine learning algorithms to analyze player behavior and performance data in real-time, allowing for personalized assistance tailored to each player's unique gaming style and proficiency. By adapting assistance techniques on-the-fly, we could optimize player engagement and satisfaction in a dynamic gaming environment.

7. Reference

[1] Sabet, S. S., Schmidt, S., Zadtootaghaj, S., Naderi, B., Griwodz, C., & Möller, S. (2020, May). A latency compensation technique based on game characteristics to mitigate the influence of delay on cloud gaming quality of experience. In Proceedings of the 11th ACM Multimedia Systems Conference (pp. 15-25). Presented in Istanbul, Turkey.

[2] Liu, S., Xu, X., & Claypool, M. (2022). A survey and taxonomy of latency compensation techniques for network computer games. ACM Computing Surveys (CSUR), 54(11s), 1-34.

[3] Xu, X., Bosik, M., Desveaux, A., Garza, A., Hunt, A., Person, C., ... & Claypool, M. (2022, September). Compensating for latency in cloud-based game streaming using attribute scaling. In 2022 14th International Conference on Quality of Multimedia Experience (QoMEX) (pp. 1-4). IEEE. Presented in Lippstadt, Germany.

[4] Vicencio-Moreira, R., Mandryk, R. L., Gutwin, C., & Bateman, S. (2014, April). The effectiveness (or lack thereof) of aim-assist techniques in first-person shooter games. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 937-946). Presented in Toronto, Ontario, Canada.

[5] Claypool, M., & Claypool, K. (2010, February). Latency can kill: precision and deadline in online games. In Proceedings of the first annual ACM SIGMM conference on Multimedia Systems (pp. 215-222). Presented in Phoenix, Arizona, USA.

[6] Ahlström, D., Hitz, M., & Leitner, G. (2006, October). An evaluation of sticky and force enhanced targets in multi target situations. In Proceedings of the 4th Nordic Conference on Human-computer Interaction: changing roles (pp. 58-67). Presented in Oslo, Norway.

[7] Li, Z., Melvin, H., Bruzgiene, R., Pocta, P., Skorin-Kapov, L., & Zgank, A. (2018). Lag compensation for first-person shooter games in cloud gaming. In Autonomous Control for a Reliable Internet of Services: Methods, Models, Approaches, Techniques, Algorithms, and Tools (pp. 104-127). Cham: Springer International Publishing.

[8] Bateman, S., Mandryk, R. L., Stach, T., & Gutwin, C. (2011, May). Target assistance for subtly balancing competitive play. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 2355-2364). Presented in Vancouver, British Columbia, Canada.

[9] Reynolds, C. W. (1987, August). Flocks, herds and schools: A distributed behavioral model. In Proceedings of the 14th annual Conference on Computer Graphics and Interactive Techniques (pp. 25-34). Presented in Anaheim, California, USA.