WORCESTER POLYTECHNIC INSTITUTE

MASTER'S THESIS

A Comparison of Automatic versus Manual World Adjustment for Network Game Latency Compensation

Author:
Robert Salay

Supervisor:
Mark Claypool

Readers:
Lane Harrison
Charles Roberts

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

in

Interactive Media and Game Development

August 13, 2020

Contents

List of Figures

List of Tables

Abstract

Computer networks add latency to online games, degrading the quality of experience of players. With the advent of game streaming services, even single-player games are affected by network latency, creating challenges, but also presenting new opportunities for latency compensation techniques for single-player games. This paper studies a genre of latency compensation called world alteration, that adjusts the game world to keep the precision needed to play competently consistent for a player regardless of latency. To evaluate world alteration, we modified an existing single-player game to have controlled amounts of latency and world adjustments to compensate for high latency conditions. We conducted a user study where participants played the game under different amounts of latency, with both automated and manually controlled latency compensation, recording their scores as well as their self-reported quality of experiences. Analysis of the results from 18 users shows both manual and automated world alteration can improve player performance and quality of experience, though extreme amounts of latency still cause degradation of both.

1 Introduction

Latency is an inherent part of playing any video game. Input needs to travel from the player's controller into the computer operating the game, and the computer needs to send its output to screens and speakers so the player can understand what effect their input has had on the game. Due to hardware and software limitations, those transmissions take time to complete, and that delay is latency.

Quality of Experience, or QoE, is a less concrete phenomenon. As the name suggests, QoE is the measure of a player's experience with technology, including how enjoyable a game is to play [7]. QoE is a qualitative and subjective measurement. Although difficult to measure, a high QoE is generally the goal of game designers who wish for their product to be commercially successful.

While latency does exist even in single-player video games, it generally is not noticeable except for online multiplayer games, where signals must travel much greater distances to servers and other clients, connecting with computers possibly on the other side of the world. When latency becomes noticeable, it negatively impacts QoE [13]. High levels of latency make games feel less responsive and make it more difficult for players to give time-sensitive input [13]. One hundred milliseconds of latency or less is generally acceptable for online games, and 20-40 ms is considered ideal [4].

Developers use latency compensation to hide latency wherever possible [1]. This is to minimize the effect latency has on QoE. There are several different methods to do so, and each one can have a different effect on how the game is played. As such, choosing which method to use is as much about how the game is designed as it is how much latency is tolerable in the finished product.

One method, which is explored in this thesis, is called geometric compensation [8]. In this paper, it is called world alteration, as more attributes can be affected by this method than just the geometry of game objects. In world alteration, instead of reducing latency, the game is made easier so that players experience a level of difficulty similar to what they would have encountered if there was no latency at all. This method is difficult to implement in multiplayer games since players with different network environments would see the game world differently, but the rise of single-player games on streaming services provides an opportunity for world alteration to be used more broadly.

As a part of our experimentation with world alteration, we used two methods of applying it to our study. The first is an automatic application of the method, where the extent of the world's alteration is determined by the amount of latency present in the game. The other is a manual application, in which the extent of alteration can be changed by the player, and the amount of world alteration functions similarly to more traditional difficulty levels in games. World alteration focuses on ways to make a game easier with increased latency. For example, in a first-person shooter it would make the game easier if the enemies had less health, or by making the hitboxes of enemies extend past what is visible. Both of these changes focus on reducing the need for precision in a game.

This thesis examines world alteration as a latency compensation method. We studied how manual and automatic application of world alteration affected players scores and QoE. For this experiment, we modified an open-source version of the game Flappy Bird [3], called FlapPy Bird [15] for experimentation. The game was altered to include variable amounts of latency and both manual and automatic world alteration. It also automatically administered surveys and recorded game data.

One round of testing was done on the campus of WPI. With the use of WPI's computer labs, the physical specifications of computers used by test participants could be controlled. Due to the COVID 19 pandemic, the second round of testing was performed over the Internet to prevent the spread of disease. These tests were performed using the participants' personal computers. In total, 18 test participants participated in this study. Twelve tests were performed on campus, and the other six were performed remotely. The purpose of these tests was to determine if world alteration could mitigate the diminished QoE caused by latency.

Analysis of our data shows no explored application of world alteration performed as well as the games without added latency, but both applications of world alteration performed better than no compensation at all. When latency was added artificially to games, their QoE suffered. When rated for Quality of Experience on a scale of 1-5, games without added latency scored an average of 3.35, games with no latency compensation scored an average of 2.69, and games with manual latency compensation scored an average of 2.84, and automated compensation scored an average of 2.97. While subjectively the effects of world alteration were slight, there was a significant difference in objective measures. On average, players scored eight times higher in games with latency compensation than games without.

This document details our exploration of this subject. Section 2 explores related research and the background of this problem. Section 3 details our methodology for the experiment. Section 4 contains the results obtained from the experiment, and finally Section 5 has conclusions and any possible future work.

2 Related Work

Even small delays have been known to degrade QoE [6]. Chang *et al.* [9] found latency had a profound negative effect on the Quality of Experience (QoE) of a game, but that effect could be heavily mitigated through compensation.

Online games need to hide latency during gameplay, which is more complicated. Methods to combat network latency have often incorporated manipulation of the game world, either predicting object positions or rolling back virtual time to previous, lagged states [1]. All techniques have drawbacks, such as adding additional computation overhead or sacrificing consistency for responsiveness. For example, *Time Warp* [5] simulates players' actions as though they were received at the same time they were performed, minimizing the perceived latency of these actions. However, Time Warp can cause players to be affected by events retroactively. This is most notable in first-person shooters, where players with high latency can find themselves shot after taking cover. Techniques have been proposed to mitigate this issue [9]. Ultimately, what kind of latency compensation is used in a particular video game is as much a design choice as it is a necessity [1] to maintain playability across networks.

This thesis evaluates world alteration as a complementary form of latency compensation given the emergence of cloud-based game streaming

Last year, Google announced Stadia, a cloud gaming service [12]. Unlike console game systems, where the purchased hardware contains a self-sufficient computer, Stadia uses cloud computing to run video games. All computations are performed on servers owned by Google, and the game is streamed into the homes of players. This allows Stadia subscriptions to be sold for less than other competing systems and avoids the need for lengthy game updates. One major criticism against Stadia is that it introduces network latency to single-player games that otherwise would have only local system latency, as even single-player games will have to connect to the cloud.

In multiplayer games, latency compensation techniques have to account for multiple players, and function in a way that aids all players proportionately. For example, a recent study examined the effects of a kind of latency compensation where latency was intentionally added to make all players delay equal [10]. While the experimental compensation did make the game feel fairer, QoE still suffered. Traditional single-player games only had to compensate for latency when loading portions of the game. Cloud-based games are in a unique position, where they have to compensate for latency during gameplay, but they only have to be concerned with the experience of one player. Multiplayer games exist on Stadia, but their need for latency compensation is already served by existing latency compensation techniques. Single-player games can use new latency compensation techniques, such as world alteration.

In world alteration, the game world is changed to provide a gameplay experience that requires precision comparable to the unaltered game in a lower latency environment. For example, in a target gallery game, the targets would be made larger and/or move more slowly. This kind of latency compensation works to either increase the deadline of a games actions or decrease the amount of precision needed to perform actions properly. As discussed in Claypool and Claypools work [2], increasing the deadline and decreasing the precision of actions in a game decreases the effect of latency on those actions, making a game easier to play in high-latency environments. In a multiplayer game, such a solution would be untenable since the same world would appear differently to players with different latencies, but a cloud-based single-player game can support it because there is only one world view. Furthermore, experiments have been done [14] showing that low apparent latency is not inherently necessary for a good user experience. In other words, a latency compensation technique that does not hide latency such as world alteration can still have an improved quality of experience.

Previous studies by Lee *et al.* [8] created a version of Flappy Bird with world alteration and used it to explore the effects of world alteration to compensate for latency. They found world alteration to be a satisfactory method of latency compensation. Our research expands upon this work by studying the subjective measure of QoE to compliment the objective measure that was already determined. Our research also compares automatic world alteration to a manual version controlled by the player.

3 Methodology

With this project, the effectiveness of world alteration as a latency compensation method was examined. For this to be tested, We needed a game to serve as a base. The game needed to be short, to maximize how many times tests could be run; simple, to maximize the number of people that could play it; and heavily reaction-based, so that the effects of latency would be as obvious as possible. What attributes would be affected by world alteration, and the extent thereof, also had to be determined. Surveys and other data collection tools were then constructed as well.

3.1 Base Game

A great deal of care and attention went into choosing which game would be used and how it would be modified. In the end, FlapPy Bird, a python-based version of the mobile game Flappy Bird [3] made by Sourabh Verma [15] was chosen. The core gameplay of Flappy Bird is simple and heavily affected by reaction time, and thus also by latency. In Flappy Bird, players must navigate their bird through gaps in walls moving towards them. The only gameplay is to either press a button to flap and move upwards or not press the button to let gravity pull them down. The simplicity of the games controls would allow for a wider audience for testing, while the difficulty of the game would make the effects of latency more readily apparent.

3.2 Modifications

Three major modifications were made to the game for the sake of world alteration. Artificial latency was added to the game by simply delaying the effects of button presses by an adjustable amount. For this experiment, four levels of latency were chosen: 10ms, 100ms, 200ms, and 400ms. Due to the implementation of latency into the game, a zero value was not possible, so 10ms was chosen as the minimum amount of latency instead, as it was still small enough to prevent it from superficially affecting gameplay.

When world alteration was added to the game, the next decision was how exactly the world should be altered. The strength of gravity, the strength of flapping, and the size of the gaps in pipes were chosen for this purpose. The gap size was chosen to reduce the precision needed to proceed through the game, and the strength of flapping and gravity were chosen because as they are reduced in tandem, fewer actions per second become necessary to play the game. The world alteration could then be applied to the game world in one of three different ways; automatic, manual, and none. Automatic application would change the game world a set amount based on how much latency was applied. Manual application starts at one level of world alteration, but the player was free to change it at any time. If world alteration was not applied, there would be no adjustment to the world regardless of latency. The exact amount of alteration at each level of latency is discussed in more detail later.

Automated data collection was also added to the game to maximize the effectiveness of each test. This data collection kept track of how long each game took, what versions of the game were played and in what order, the final score of each game, and what difficulties a game was set to in manual application mode. This data is used to provide an objective measure of world alterations effects alongside the subjective measure gained from post-game surveys.

3.3 Pilot Studies

Before finalizing the world alteration system, pilot studies were performed to fine-tune the adjustments it would make to the game. The goal was to adjust the game world, so its difficulty was comparable to that of the original game despite the introduction of latency. This was achieved by finding the average play score of an unaltered game of FlapPy Bird, and then adjusting the amount of world alteration present for each amount of latency until the average play score was within one point of the unaltered game.

Table 1: World alteration by latency amount

Latency	Gravity Power	Flap Power	Pipe Gap Size	Pipe Gap (in.)
10 ms	100%	100%	100%	1.3
100 ms	80%	89%	110%	1.4
200 ms	75%	80%	130%	1.7
400 ms	60%	72%	200%	2.6
Manual control	40%	67%	240%	3.1

Table 1 shows the final world alteration levels
chosen after the pilot tests were completed. All measurements in
inches were measured on a 15.5” screen at 1920 by 1080p resolution. Manual

world alteration starts at the manual control starting level and can be adjusted by players at any time while playing. 10 ms was used as the minimum amount of latency added because the code could not account for zero miliseconds of latency without error.

Figure 1 shows the difference between minimal world alteration and maximal world alteration. With maximal world alteration, the vertical mobility of the bird is also reduced, but cannot be seen in a picture.

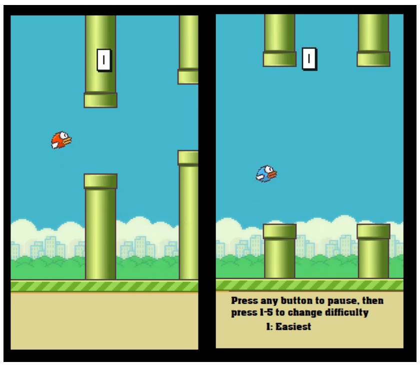

Figure 1: No world alteration (left) compared to maximum alteration (right)

3.4 Surveys

To minimize annoyance and survey fatigue on the part of the test participants, they were given two kinds of surveys. Smaller surveys were administered after playing specific versions of the game, and a larger survey was administered after all versions were finished. The in-game surveys only focused on the quality of experience offered by the last version the participant played, while the large survey focused on the demographic information of the participant.

The in-game surveys were administered between games so as to have minimal interruption. Each participant played each test case three times in a random order, and this survey was administered the second time they played a test case. The user was asked to rate four aspects of their previous game on a scale from one (the lowest) to five (the highest). Those aspects were:

1. Difficulty
2. Impact of latency
3. How well the game flowed
4. Quality of experience

10 ms delay no comp.	10 ms delay auto. comp.	10 ms delay manual comp.
100 ms delay no comp.	100 ms delay auto. comp.	100 ms delay manual comp.
200 ms delay no comp.	200 ms delay auto. comp.	200 ms delay manual comp.
400 ms delay no comp.	400 ms delay auto. comp.	400 ms delay manual comp.

Table 3: Game versions

After all the games were completed, a demographic survey was administered. Table 2 shows simplified versions of the questions asked. Full-text copies can be found in Appendix A.

Question	**Category**
Game time per week	Bin
Genres preferred	Multiple choice
Device preferred	Multiple choice
Has played Flappy Bird	Yes or no
Age	Text input
Gender	Text input

Table 2: Post Game Survey

3.5 Game Data

Aside from the surveys, the game also automatically collected the gameplay data from participants. This includes how long each version of the game was played, what version of the game was played each time, and the participants score.

3.6 Test Procedure

The participants were first given a brief verbal explanation of the test and signed an informed consent form that further explained the purpose of testing. The first three games of FlapPy Bird were unaltered for test participants to acclimate to the game. After that, the participants played through all twelve versions of the game in a random order, with each version being played three times. In-game surveys were automatically administered after the second playthrough of each version. Post-game surveys were automatically administered at the end of the final game.

Table 3 shows all versions of FlapPy Bird used in this study. Four levels of latency and three methods of latency compensation combine to create 12 test cases.

3.7 Testing Locations

The first 12 tests were all performed on computers in various computer labs on WPI. Due to the Coronavirus outbreak and subsequent stay at home orders, the remaining tests were completed over the Internet, with participants using their own computers to reduce the risk of disease. While the change in testing platform and location could alter results, all other aspects of the tests are constant. Moreover, as a within-subject study, the relative difference between candidates are still relevant.

4 Results

This chapter evaluates the effect world alteration has on the quality of experience (QoE) participants experience. In addition, this chapter compares automatic world alteration to manual methods of difficulty adjustment.

4.1 Demographics

We first analyze the demographics of our participants. Our test group consisted of 18 users. Of the 18 users, 15 (83%) of them played Flappy Bird previously. This means that a majority of the people tested did not have to learn how to play the game during the test.

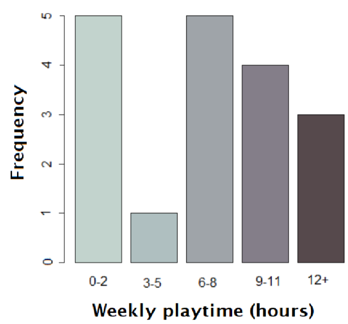

Figure 2: Self-reported gameplay habits of participants

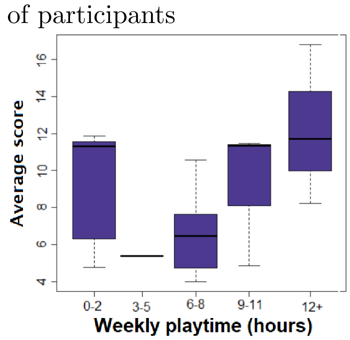

Figure 3: Average score of participants compared to weekly time playing video games

Figure 2 shows how much time participants dedicated to playing games outside of the experiment, as their familiarity with games may affect their performance. The x axis is the number of hours played in bins of three hours, and the y axis is the frequency of that response. While many users played games rarely, 27.8% playing less than two hours a week, two thirds of participants reported playing games more than 6 hours a week. According to research done by Limelight Networks in their State of Online Gaming 2019 [11], the average participant in that study spent almost 6 and a half hours playing games. Similarly, 6-8 hours per week is the most common response by our users.

Figure 3 is a box plot with the weekly playtime of participants as its x axis, and the average score each one achieved as its y axis. It shows the average score participants obtained across all playthroughs compared to how much time they spend playing games. As expected, Figure 3 shows a positive categorical correlation between time spent playing and score. However, it should be noted that participants that play little, between zero and two hours a week, have a greater variety in their scores (M=9.2, SD=3.4) than other low playtime categories, and some players are able to match the scores of those who play for more than 12 hours a week (M=12.2, SD=4.3).

Figure 4 shows the age of the participants, with an x axis of their age and y axis of frequency of response. In this study, 12 of the participants (66.7%) were male, five of them (27.8%) were female, and one chose not to disclose their gender. The demographic data of participants suggests that this study had a noticeable bias towards young, male participants, most likely due to the participant body being selected from the WPI Campus. Two thirds of the participants are male, and 55.6 percent of the group is under the age of 21.

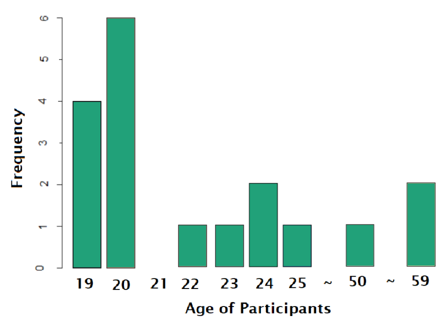

Figure 4: Age of participants

Once again referring to the State of Online Gaming 2019 [11], over half of their 4,500 participants are over the age of 35, and males make up only 51.8% of their polling group. The deviation from the general gaming population may mean that the data collected in this study should not be extrapolated to the general population without further study.

4.2 Objective Results

Table 4 shows the various game modes that were tested as a part of this experiment. The leftmost column is the numerical indicator given to each game mode, the middle is the amount of latency that game mode added in milliseconds, and the rightmost column indicates the kind of latency compensation used in the game mode. "None" compensation had no alterations made to the world,

Game Mode	Addtl. Latency (ms)	Compensation
1	10	None
2	100	None
3	200	None
4	400	None
5	10	Auto
6	100	Auto
7	200	Auto
8	400	Auto
9	10	Manual
10	100	Manual
11	200	Manual
12	400	Manual

Table 4: Game modes and their effects

"Automatic" compensation used a premade level of compensation depending on the amount of added latency, and "Manual" gave the participants control over how much world alteration was used during play. Each game mode participants played has been assigned a number for visual coherency in the following graphs. These numbers are grouped by their latency compensation style and ordered the amount of latency that was added to that mode. "None" compensation means there was no world alteration at all, regardless of latency. "Automatic" compensation means a level of world alteration was used based on the amount of added latency (as shown in Table 1), and the participant could not change it. "Manual" compensation set world alteration to a default state, then gave the participant control over how much world alteration was used in the game.

Figures 5 and 6 illustrate the effects latency has on the performance of participants. According to ANOVA tests [F(11,594)=25.62, p¡.0001], game modes 9-11 had significantly higher scores compared to the other game modes at those added latency amounts. This is most likely due to the default difficulty level being quite easy, as well as a lack of incentive for participants to make the game harder for themselves. About one third (35.2%) of games played with manual latency compensation ended without the participant changing the difficulty of their game.

The manual control cases also had much higher variance than others, due to

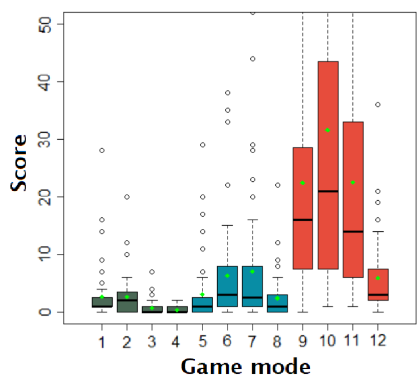

Figure 5: Average scores achieved in each game mode

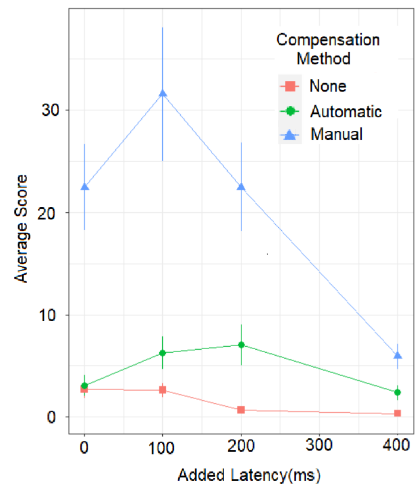

Figure 6: Average score achieved in each game mode, organized by latency added. 80% confidence interval

the nature of player choice. Players who remained on the lowest difficulty would have an easier time than those who chose to challenge themselves, causing the greater spread in scores that can be seen in figures 5 and 6.

4.3 Quality of Experience (QoE)

Finally, we come to our main goal of this experiment: the effect world alteration has on players QoE. With these compensation techniques, we strived to create a game that would produce a QoE comparable to a version of the game with no additional latency regardless of how much latency was added.

Figure 7 is a box plot, grouping together the three methods of compensation on the x axis and measuring their Quality of Experience on the y axis. The horizontal line running through the graph is the average QoE of all games with less than 100 ms of added latency. Both methods of compensation used in this project performed similarly to the version with no compensation at all. According to ANOVA tests [$F(2,140)=0.615$, $p=0.542$], there was no statistically significant difference between the three methods at this granularity.

Figure 8 is a box plot grouping each game mode together and measuring

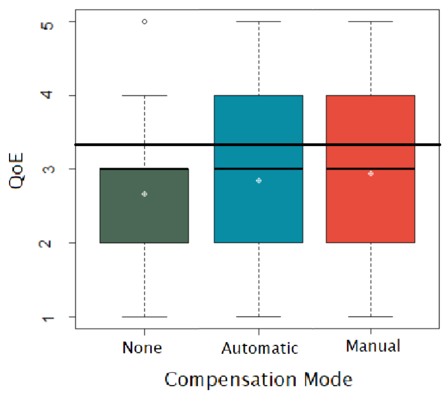

Figure 7: Average QoE of test cases, organized by compensation method. Only game modes with 100 ms of additional latency or more were used.

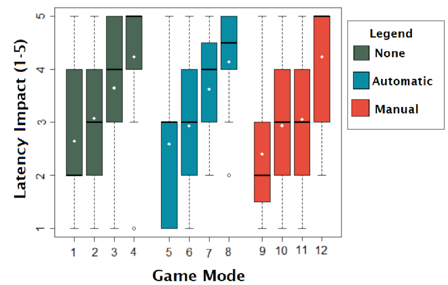

Figure 8: Comparing perceived impact of latency in each game mode. The color of boxes correspond to the compensation method.

Test Pair	P
10 ms, None/Auto.	1
10 ms, None/Manu.	0
10 ms, Manu./Auto.	0
100 ms, None/Auto.	.98
100 ms, None/Manu.	0
100 ms, Manu./Auto.	0
200 ms, None/Auto.	.52
200 ms, None/Manu.	0
200 ms, Manu./Auto.	0
400 ms, None/Auto.	1
400 ms, None/Manu.	.72
400 ms, Manu./Auto.	.99

Table 5: Pairwise ANOVA tests of comparable game modes and their scores, df=11

Test Pair	P
10 ms, None/Auto.	1
10 ms, None/Manu.	.97
10 ms, Manu./Auto.	.75
100 ms, None/Auto.	1
100 ms, None/Manu.	1
100 ms, Manu./Auto.	1
200 ms, None/Auto.	1
200 ms, None/Manu.	1
200 ms, Manu./Auto.	1
400 ms, None/Auto.	1
400 ms, None/Manu.	1
400 ms, Manu./Auto.	1

Table 6: Pairwise ANOVA tests of comparable game modes and their QoE, df=11

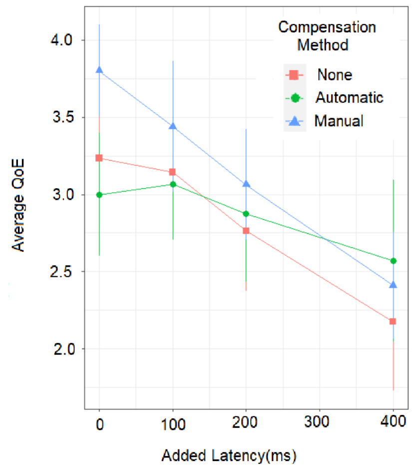

Figure 9: The average QoE compared by compensation method, organized by amount of added latency with an 80% confidence interval.

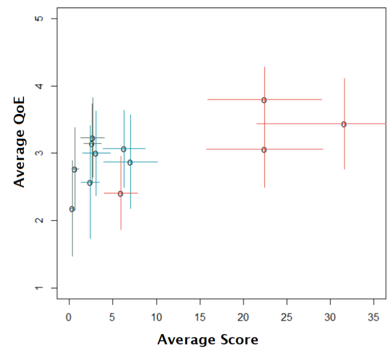

Figure 10: The average score and QoE of each game mode on two dimensions. The 95% confidence intervals are colored to match their compensation methods.

the reported impact of latency on the y axis. Each box is also color coded to match it to its compensation method. It shows a positive correlation between increasing amounts of added latency in a game and the perceived impact of that latency. There is also no statistical significance between corresponding game modes, suggesting that world alteration does not hide perceived latency from the player. This was also expected, as the point of world alteration is to make a game enjoyable despite the existence of latency.

Figure 9 is a line graph, with an x axis of added latency and a y axis of the average QoE. Each line is a different compensation method, and each point on the line is a single game mode. The vertical lines indicate the 80% confidence interval of each point. Similarly to Figure 6, all QoE also suffered with additional latency, regardless of compensation method. The difference between manual compensation and other methods was significant in low latency, likely due to that method making the game easier compared to the others, as all methods fared comparably in the 400 ms tests. The goal of the automatic latency compensation is that players experience the same QoE regardless of latency in the environment. While the automatic method has reduced the effect of latency, latency still has a noticeable effect.

4.4 Further Results

Figure 10 is a scatter plot with game modes as its points, measured on their score on the x axis and their QoE on the y axis. It shows the relationship between how high players scored in a game mode and the quality of experience players reported on those modes. We found the correlation between the two to be positive ($r = 0.64$). The three points that are separate from the others are game modes 9-11, with the likely reasons for this significant distance-how the application of manual compensation was done as explained in the Section 4.3.

5 Conclusion

The growth in cloud-based streaming game systems brings the challenge of latency but also opportunities for new techniques. World alteration is one such technique. In world alteration, the game world parameters are adjusted to make the world easier in response to latency. It is difficult to use this technique in multiplayer network games due to the issues that arise from multiple players altering the game world, but the rising popularity of cloud-based single-player games will give world alteration a niche use in the future. This paper assessed world alterations ability to reduce the effects of latency on player performance and their Quality of Experience in single-player games.

We examined a form of latency compensation where the obstacle sizes, gravity, and other effects is adjusted based on the severity of latency added. Three game modes were tested to assess how manual and automatic world alteration impacts QoE and performance in high-latency environments. A study was performed with 18 participants to evaluate the effectiveness of world alteration as a method of latency compensation. The study measured the objective effects latency and world alteration had on gameplay as well as the subjective effects they had on players' QoE.

Participants playing games with manual world alteration performed much better than either automatic compensation or none at all. The average score for games played with manually controlled latency compensation was 20.6 points, while automatic compensation had an average of 4.7 points, and no compensation only had an average of 1.58 points. However, all compensation methods were still affected by latency; at 400 ms of additional latency, the average score for all games regardless of compensation method was below 6 points per game.

The automatic application reduces this drop by over 50 percent on average, while manual application increases this drop by 30 percent on average due to its higher averages for low latency.

5.1 Future Work

The experiments were performed with a limited number of users, due to the COVID-19 epidemic. One of the most immediate improvements that could be made to this work is to expand the number of test participants used in future runs of this experiment. This will both increase the likelihood of statistical significance and also likely give us a pool of participants more indicative of the general population.

Given the limited number of participants in this study, it is also worth examining any trends in individuals to determine any bias in overall data that may have come from the small sample size.

Another improvement can be further refinement made to the world alteration system. The extreme discrepancies in difficulty between manual world alteration and all other compensation methods indicate that this method may have skewed results. Alterations to the difficulty settings could be done to prevent this from happening in the future. Another area that can be improved is examining the benefits and drawbacks to more attributes for world alteration.

Further examination of existing world alterations and their individual effects on QoE will also be useful.

Further work can improve the applicability of this method, starting with examining its effects on other games. Flappy Bird has been a popular target for latency experiments [8] due to its simple control scheme and reliance on timing, but it is important to explore world alterations effectiveness in other games, as well.

Finally, a longer-term goal is to implement world alteration in a way that is genre agnostic, such as inside a game engine, so that it may be implemented into games regardless of genre being used. This may significantly improve world alterations accessibility for game designers.

References

[1] Yahn W. Bernier. “Latency Compensating Methods in Client/Server In-game Protocol Design and Optimization”. In: *Proceedings of the Game Developers Conference*. San Francisco, CA, USA, Feb. 2001.

[2] Mark Claypool and Kajal Claypool. “Latency and Player Actions in Online Games”. In: *Communications of the ACM* 49.11 (2006).

[3] dotGears. *Flappy Bird*. Game [IOS]. dotGears, Hanoi, Vietnam. Hanoi, Vietnam, 2013.

[4] *How to Reduce Latency or Lag in Gaming*. Actiontec. Online:`https://www.actiontec.com/wifihelp/wifibooster/how-to-reduce-latency-or-lag-in-gaming-2/`. Accessed August 2020.

[5] K.A. Iskra, G.D. van Albada, and P.M.A. Sloot. “Time Warp cancellation optimizations on high latency networks”. In: *Proceedings Seventh IEEE International Symposium on Distributed Simulation and Real-Time Applications* (2003). [Online] https://ieeexplore.ieee.org/document/1243006.

[6] Zenja Ivkovic et al. “Quantifying and Mitigating the Negative Effects of Local Latencies on Aiming in 3D Shooter Games”. In: *CHI 2015, April 18 - 23, 2015, Seoul, Republic of Korea*. 2015.

[7] Khalil Ur Rehman Laghari and Kay Connelly. “Toward Total Quality of Experience: A QoE Model in a Communication Ecosystem”. In: *IEEE Communications Magazine* 50 (2012).

[8] Injung Lee, Sunjun Kim, and Byungjoo Lee. “Geometrically Compensating Effect of End-to-End Latency in Moving-Target Selection Games”. In: *CHI ’19: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (May 2019). [Online] https://dl.acm.org/doi/10.1145/3290605.3300790.

[9] Steven Lee and Rocky Chang. “Enhancing the Experience of Multiplayer Shooter Games via Advanced Lag Compensation”. In: *MMSys ’18: Proceedings of the 9th ACM Multimedia Systems Conference*. June 2018.

[10] Zhi Li et al. “Lag Compensation for First-Person Shooter Games in Cloud Gaming”. In: *Lecture Notes in Computer Science*. May 2018.

[11] *Market Research: The State of Online Gaming – 2019*. "Limelight Networks". Online: https://www.limelight.com/resources/white-paper/state-of-online-gaming-2019/. Accessed July 2020. Mar. 2019.

[12] Michael McWhertor. *Google Stadia is the New Streaming Gaming Platform from Google*. Polygon. Online:https://www.polygon.com/2019/3/19/18272856/google-stadia-gdc-2019-announcement. Accessed July 2020. Mar. 2019.

[13] Jose Saldana and Mirko Suznjevic. "QoE and Latency Issues in Networked Games". In: *Handbook of Digital Games and Entertainment Technologies*. 2015.

[14] Cheryl Savery et al. "The Effects of Consistency Maintenance Methods on Player Experience and Performance in Networked Games". In: *CSCW 2014*. Feb. 2014.

[15] Sourabh Verma. *FlapPy Bird*. "GitHub". Online: https://github.com/sourabhv/FlapPyBird. Accessed July 2020. Feb. 2020.

6 Appendices

6.1 Appendix A: Full text of surveys

How challenging was it to play this version of Flappy Bird?	1: Too easy, 2, 3, 4, 5: Too hard
How much of an impact did latency have on your gameplay?	1: A little, 2, 3, 4, 5: A lot
How difficult was it to get into a rhythm while playing?	1: Easy, 2, 3, 4, 5: Difficult
How enjoyable was it to play this version of the game?	1: Not enjoyable, 2, 3, 4, 5: Very Enjoyable

Table 7: Mid-game Survey

How much time per week do you spend playing video games?	0-2 hours, 3-5 hours, 6-8 hours, 9-11 hours, 12+ hours
What kind of games do you play in your free time?	Action, Hidden Object, Puzzle, Reaction-based/Rhythm, Role Playing, Simulation, Sports, None, Other
On what devices do you usually play video games?	Handheld system, Home Console, Smartphone, Computer, Arcade, Other
Have you ever played Flappy Bird before this test?	Yes, No
Age	
Gender	

Table 8: Postgame Survey