Pakistan Journal of Engineering Technology and Science (PJETS)
Volume 12. Issue.01, PP. 125-138, June 2024
https://doi.org/10.22555/pjets.v12i1.1111
E-ISSN 2224-2333
P-ISSN 2222-9930

Improving Cloud Streamed Gaming Quality Using Latency Compensating Gameplay Enhancements

Nabeel Alam [1], Syed Muhammad Khaliq-ur-Rahman Raazi [1*], Bilal Mehboob [2], Syed Mubashir Ali [1]
[1] Faculty of Computing, Mohammad Ali Jinnah University, Karachi, Pakistan.
[2] Superior University, Pakistan

*Corresponding author: raazi.m.syed@ieee.org

Abstract:

The arrival of Cloud Gaming (CG) has enabled video games to not require gamers to have a powerful machine with high-end processing capacity and rather have the option to play them by streaming them while they get processed on a server, just like an internet video streaming site. Despite the ease of use that the service provides, its heavy reliance on an internet connection with high bandwidth and low latency hinders the service from being adopted more than it has so far, especially as higher latency results in poor game performance and decreases Quality of Experience (QoE). In this report, we discuss the current literature on the effects of delay in cloud gaming on players' QoE, simulate variable delays that range from low to high randomly, and try to mitigate negative impacts on QoE caused by these delays by applying some proposed latency compensating enhancements on a game called "Shooting Star" which we developed for this experiment. We evaluated results after experimenting by having the game played by crowd-workers, gathered their performance metrics, and surveyed them about their feedback on their perceived QoE and responsiveness in different conditions such as no input delay, constant delays, and varying delays along with gameplay enhancement scenarios, the latter designed to mimic real-life Cloud Gaming conditions. The results of the study enable us to establish more accurately whether implementation of delay countering enhancements has a positive impact and is feasible to deliver depending on the worth of their reward. The study we conducted involved 140 participants, and the results we analyzed indicated that the gameplay enhancements that we designed had a significant impact in mitigating the negative effects of delay on the players' performance, as well as their perceived quality of experience and responsiveness.

***Keywords**: Cloud gaming, Streaming, Video games, Delay, Latency, Quality of Experience, Game performance*

I. INTRODUCTION

Cloud Gaming (CG) is the preferred name that is commonly used to refer to modern video gaming and multimedia services that enable users to play video games through a video stream hosted on a cloud server over the internet, instead of rendering the video game application on their hardware. By using these services, users avoid needing to be concerned about upgrading their hardware (consoles, phones, a PC and its peripherals), or software (operating systems), due to these services only requiring minimal input hardware and a strong internet connection. The hardware rendering of the video game itself is essentially performed on a cloud-hosted server which performs the heavy processing on its powerful hardware and then serves the generated visual output through the cloud. On the client side, the user provides input, most popularly through a keyboard, mouse, or a video game controller, and the input subsequently gets processed by an application that sends it to the host cloud server to which the client is connected [1].

Cloud Gaming saw its rise after the popularity and increased demand of high-end and hardware-intensive video games, and is aimed at consumers who already have a high-speed internet connection or would prefer to spend on attaining one rather than being interested in investing in expensive gaming equipment such as powerful PCs or gaming consoles. The users who CG aims to gain as an audience also tend to demand the high-fidelity in-game visuals that recent video games can to achieve because of the advancement in 3D graphics and video game animation.

Moreover, as the number of mobile users and mobile gamers has increased, so has the demand for higher-quality video games to be released on cell phones and other portable media [2][3]. Due to mobile hardware limitations, heavy CPU- and GPU-related processing isn't feasible. Video games require more processing, graphics power, and memory than the average mobile application, so the performance of games suffers on these platforms. Also, [2] and [3] show that due to the large and scattered variety of mobile platform hardware levels, it is difficult for video game developers to be able to deliver high-fidelity, visually demanding games that run at playable speeds on all available mobile devices.

Another motivation for video game developers to support the rise of cloud gaming has been its aid in restricting video game piracy. In CG, since the video game application is being run on the host cloud server itself, and the source code remains on the host's hardware and is never shared with the client, piracy is avoided as the client cannot replicate the video game code or software for cracking, or reproduction [4]. This tactic helps video game developers avoid the unpopular practice of integrating a digital rights management (DRM) system in their game's code to ensure that players are using a legitimately purchased and licensed copy of the game.

As the usage of physical media such as CDs, DVDs, and BluRay discs has died down with the advent of downloadable media and increasing internet speeds, an active internet connection has become almost a requirement for players to be able to play games, even the ones that they have legally purchased and downloaded on their systems. This sort of protection is called DRM and has been popularly included in high-selling video games by big-name companies. The system ensures that each copy of a video game possesses a unique digital license, which has to be verified through the game developers' servers each time the user wants to play it [5]. Illegal copies are detected through invalid or replicated IDs and, subsequently, lock the player out of the application.

Input lag is a phenomenon in which there is a delay between the user's input (using the computer's hardware such as a keyboard or mouse) and the feedback of the input event. This is due to the constraints of the hardware itself, as it delivers signals to and from the computer depending on its specifications [6]. Another – far more common and impactful – issue faced by gamers using Cloud Gaming services is response delay (latency between the user's input and the on-screen visual feedback) [7][8], caused by higher-than-optimal response times from the user's input to the actual feedback sent back to the user through video output by the cloud server due to the constraints of the end-to-end internet connection.

To the best of our knowledge, up to this point in time, there have only been experiments conducted to investigate gameplay improvement techniques and enhancements that focus on simulating those modifications with fixed delays and preconfigured enhancements. The studies we researched are discussed later in this report, along with comments on their results and findings. Generally, they have also focused primarily on single-player experiences where enhancements were only tested in environments that did not evaluate the QoE improvement and fairness factor when those modifications were applied to multiplayer games.

II. LITERATURE REVIEW

To establish a stronger perspective on one of the primary objectives of this report, which is to understand and examine the relationship between delay and players' quality of experience (QoE) and responsiveness quality when playing cloud games, we shall chronologically recount and discuss findings of a relevant portion of the vast amount of literature that can be found on the subject of cloud gaming (or any cloud streamed computing service), and point out the significance that they present specifically to discussions about response delay (or input latency), and its effects on users' quality of experience.

Huang, C.-Y., et al. claim to have created the world's first-ever cloud gaming system which is completely open-source, allowing anyone to obtain it, and they call the platform "GamingAnywhere", mentioning how services such as OnLive and StreamMyGame suffer from high server response delays; ranging from 134 ms to 375 ms on average; this range – and, of course, any more elevated amount – of time has been considered "high network latency" in their study, therefore to counteract this their open-source system is meant to be highly extensible, allow for cross-platform functionality, and be portable to multiple operating systems, aiming for efficiency, scalability, and greater responsiveness than other available services which are of a similar type [9].

In [9], Huang, C.-Y., et al. present statistical evidence of their cloud gaming platform outperforming such services as OnLive in the matter of total response delay (RD). However, their optimizations to achieve lower latency have more to do with the system architecture rather than making amendments to the gameplay. Despite this, it is evident from their research that a lower response delay is ideal due to players' QoE dramatically lowering whenever they encounter higher latencies.

In their 2015 article, Tian, H., et al. also narrate their purpose to take on the task of reducing the different kinds of costs endured by cloud gaming services, for which they focus on delivering a better standard of choosing the servers (data centers) to serve the player according to his or her geographical location [10]. Beyond the data centers, they also venture into video output optimization through adjustment of bitrate and improving the allocation systems of VMs required to process the services that are demanded by the user.

The relevance of [10] with the subject matter of our study is emphasized by the primary motivation behind the purpose of Tian, H., et al., which they claim to be about improving the QoE for gamers so that it reaches a sufficient, or "good-enough" level. Furthering the research done in [10], Gao, Y., et al. propose their own implementation in the same area, claiming to improve upon the previous work using an improved version of the grey wolf optimization algorithm [11], as opposed to the algorithm proposed in [10] which is based on the Lyapunov optimization theory. The main difference between the studies [10] and [11] is that, in the latter, Gao, Y., et al. set out to reduce electricity costs endured by providers managing multiplayer cloud sessions and to make improvements to the inter-player cloud gaming experience, but their purpose of improving player QoE matches that of the study they propose the enhancement upon.

One interesting study by Hossain, et al. discusses the effects of the psychological state of the player on the QoE that is perceived by them. These psychological states come into play when they manifest as significant factors that decide the kind of experience that is had by a player [12]. Using emotion detection through hardware assistance, they attempt to maintain a positive mental state by manipulating screen effects while playing online games [12]. While the specifics of the aforementioned research [12] may not be exceedingly relevant to gameplay and network quality, as is the subject of our proposed study, it is still noteworthy that a broader perspective, such as psychological and emotional state has also been stipulated for examination and improvement of user QoE, rather than just a bare-bones technical viewpoint.

Going back to the open-source cloud platforms, Beyer, J., et al. present their platform based solely on the Android platform, which utilizes a mobile Android operating system Android Open Source Project (AOSP) to run on its instances [13]. The server used by the platform is a Gaming Anywhere server, on which it streams "innumerable" Android games that exist on the OS.

Sabet, S. S., et al., in their 2018 study "Towards Applying Game Adaptation to Decrease the Impact of Delay on Quality of Experience", investigate the relationship between three factors, namely; delay, performance, and Quality of Experience [14]. They believe their investigation to be the first of its kind; they explore the primary subject of this research, which is the effects of QoE that are caused by delay.

The study [14] includes experiments conducted on two video games of different genres: Need for Speed Shift 2, a popular car racing video game, and the Table Tennis minigame that is a part of London 2012: The Official Video Game of the Olympic Games. These were streamed on a cloud gaming platform with network emulation to introduce varying levels of delay in the stream to simulate a real-life input lag and response delay situation. After applying these negative effects, they analyze the changes in player performance and discuss the total impact on QoE. The study finds a clear relationship between players' performance in the game, and their overall QoE, with delay also having its impact on it. In addition, the delay also affects the performance of the players, creating a compounded negative effect. To mitigate the effects of the delay, Sabet, S. S., et al. add delay compensating enhancements and find that with more relaxed criteria of scoring, as performance increases, so does the QoE[14].

Sabet, S. S., et al. also further investigate the subject of quality of experience in a later study titled "A Latency Compensation Technique Based on Game Characteristics to Mitigate the Influence of Delay on Cloud Gaming Quality of Experience" [15]. The primary topic is the same as the one in their article mentioned above. They extend their previous literature by introducing 4 different games which represent 4 distinct gameplay types. They choose 5 gameplay characters based on which they apply enhancements to each game and conduct a similar survey to obtain data on the players' experience in different response delay scenarios.

The results in [15] follow the same pattern as [14], as Sabet, S. S., et al. observe that there is a direct correlation between the delay experienced by players, their performance in the game, and the negative impact on their QoEwhenever delay induces a decreased level of performance, and with the application of delay compensating gameplay enhancements based on the aforementioned characteristics, the players' QoE reportedly increases. They conclude by proposing that future work on the subject may focus on the application of gameplay enhancements.

Another example of a study conducted about game adaption and enhancement is [16], in which Halbhuber, D., et al. invoke deep learning and apply artificial neural networks (ANN) in a custom-made 3D shooting game. They investigate the effects of their machine-learning-based adaption of game enhancements such as "aim-assist," which they test on multiple values of delay. There has been more recent work in this regard using artificial intelligence-based prediction [28] and [29]. However, these works have not been conclusive. Some datasets have also been discussed in the recent literature [30]. Carlos et. Al [31] has worked on other key quality indicators in cloud gaming services. Also, authors in [32] have discussed recent work in detail.

After an investigation involving conducting tests consisting of 96 participants, [16] yielded a result that favored game enhancements improving overall QoE as players reported that with each increase in the delay, its mitigation through the modifications applied to the shooting game yielded better performance by them and a higher QoE. This supplies further credence to the notion that mitigation of delays encountered in video games through adaption and modification of the gameplay can provide a higher QoE.

The games that playtesters had to participate in experiments would not simulate the conditions on cloud platforms accurately as delay is usually not fixed in bad network conditions, and latency can sway along with network jitters. We discussed that when gaming in the cloud environment, multiple conditions throughout the network play a role in determining the overall latency faced by the player, and due to this and the constantly changing nature of the structure, players face varying levels of delay in one session. The enhancements they applied would be locked in along with the fixed delay, so in different delay conditions, the enhancements would not reconfigure themselves to balance the removed difficulty when delay conditions may drop.

Additionally, the current literature primely focuses on a single-player, as adapting enhancements in games with multiple players on the same network would undoubtedly give rise to concerns of bias; and have a negative impact on the fairness of the game, as one participant may face a simpler difficulty based on their network conditions than someone with ideal conditions. This renders the players with more stable connections to the server at a disadvantage.

III. METHODOLOGY

In the previous sections, we described the related work that has been done on the subject and discussed the gaps and potential areas that can be further probed to obtain some new data that can be analyzed to find the relationship between delay and quality of experience. This study attempts to fill some of those identified gaps in knowledge. The objective was desired to be achieved by the execution of the following three-part strategy.

Figure 1: Flow diagram of the Methodology

The diagram in Figure 1 represents the overall flow of execution of the solution that was proposed and developed. Figure 1 shows the 3 major parts of our strategy, which represent each section of the methodology that we followed to conduct this study.

A. Game Development

For the first part of the strategy, a simple video game was developed that represents a generic gameplay characteristic of shooter games, in which temporal and spatial accuracy plays a vital role in the matter of determining the score that the player achieves. This is the video game used to apply simulated delay and gameplay enhancements, and then test the resulting impact on the QoE perceived by playtesters when these enhancements were enabled, compared to the results of perceived QoE when the enhancements were disabled.

The Phaser 3 library, developed by Richard Davey and the Phaser community, provides a robust framework for creating 2D games that run on web browsers [17] [18]. It offers a comprehensive set of features and a flexible architecture that facilitates game development. The library is released under the MIT License, making it freely available for commercial and non-commercial use, which aligns perfectly with our project requirements.

"Target-Practice-Game," an open-source Python game, was also incorporated into our work. The project – available on GitHub – provided a solid foundation for the development endeavor undertaken for this study [19]. To develop a game accessible to a wider audience and leverage the capabilities of the JavaScript Phaser 3 library, we undertook the task of porting Target-Practice-Game core game mechanics from Python to JavaScript. This involved not only adapting the game logic but also utilizing the existing game assets, including music, sound effects, images, and more. In this way, an established foundation was enabled to be built upon, while expanding its reach through JavaScript implementation.

In higher latency situations, especially in online multiplayer competitive shooters, the user's delayed input usually yields unfavorable gameplay rewards since by the time the input reaches the server, the "true" situation of the game world may be much different than what the player sees. For example, a player with 300ms network latency may accurately shoot at a target that they see on their screen, but by the time their input reaches the server and server has to decide whether they score the point (known as "kills" in shooting games), the target may have already moved away from where the player had attempted the shot. Therefore, despite what the player sees on their screen, their performance will become moot due to the network, decreasing QoE[16][20].

We decided to give this game the title "Shooting Star." The control scheme was kept simple; using the mouse for movement, and the left mouse click to shoot. This was done to make the game accessible for maximum playtesters who would most probably be using a mouse.

In the screenshot displayed in Figure 2, we can see the game screen and the different features of Shooting Star. The interface includes minimum "changing" imagery, so only the target and the reticle would be able to move and the target is the only object capable of changing size; with an artistic background that shows hills and primarily a sky, the target is a dartboard that is made of different bright colors for maximum visibility on the screen, a generic font text for the scorecard, and an overall minimal look with a simple black reticle used as the representation of the player's aim. The screen displays the player's score and accuracy. The target dartboard appears randomly at any

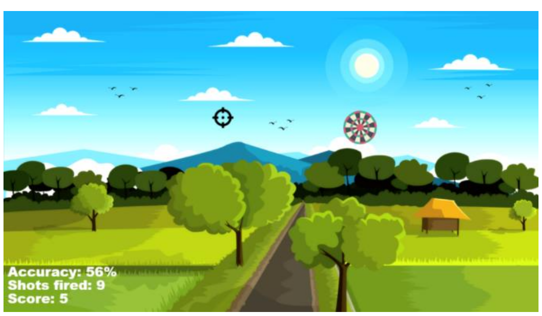

Figure 2: The main gameplay sequence of Shooting Star

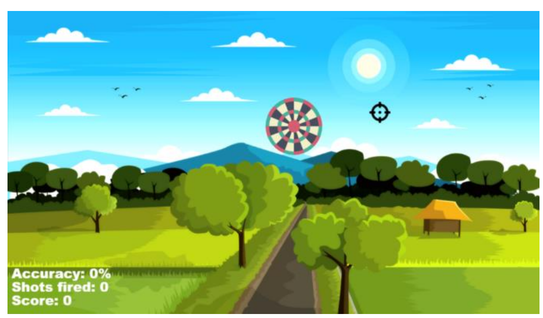

Figure 3: The main gameplay sequence of Shooting Star with Enlarged Dartboard

point of the game window for a limited amount of time (1s without enhancements), the duration in which the player has to aim at the dartboard and press the shoot button before it disappears and reappears at a different location. When the TA enhancement is enabled, the target will be present for a longer period, allowing the player to have more time to detect the location of the target and aim the reticle at the correct position.

Pictured above, Figure 3 is a screenshot of the game screen when the gameplay enhancements are enabled, and a higher input delay is being simulated. The increased size of the dartboard compared to its size observed in Figure 1 provides a visual of the SA (Spatial Accuracy) enhancement. The target is enlarged as compared to its size when no enhancement is applied or when the input delay drops to ideal conditions. The larger size not only eases the visual difficulty by being easier to detect, but it also enlarges the hitbox (the shootable area) of the dartboard. This means that the player will have to make a shorter trip from one area of the screen to the hitbox of the target when they are moving the reticle to shoot the target at the new location. While this may seem like it would have a minor effect, the actual influence is a lot more impactful as the added pixels provide the player valuable milliseconds to reach the dartboard when facing input lag.

A delay-simulating system was also developed and implemented in the game, allowing us to simulate a real cloud-gaming scenario in which process and network delays cause increased input lag and server response times. This simulation will include dynamic variations in the total latency time that the player will experience, which will mimic the way cloud gaming is experienced in real life – whose factors depend upon the scenery, image size, and changing network conditions on both the client and server sides.

The applied delays may be grouped into 3 ranges; a lower-latency range (e.g., 30-120ms), a medium-latency range (e.g.,120-280ms), and a high-latency range (e.g.,280-500ms). However, when the enhancements are applied, they will be applied dynamically.

Two gameplay enhancements were designed and implemented in the game as well. These enhancements are based on the objective of amending the target temporal accuracy (TA) and spatial accuracy (SA) that the player must achieve to increase their score. There would be a base TA and SA required to score a point. These values will be represented by two values, which we will call the hitbox (HB) size and the hit-time (HT).

At reference level (0 ms simulated input delay), the target size (hitbox) and the duration of the target on the screen (hit-time) will also be at reference levels (80x80 pixels, 1 second), and then with an increase in the input delay, the hitbox and hit-time will also be increased by two factors respectively, the hitbox increase factor (HIF) and the hit-time increase factor (HTIF).

After rigorous playtesting, we decided to set the HIF to 0.09, and the HTIF to 2.21. These are the factors by which the HB and HT increase and decrease depending on the simulated ping. For example, to determine the HB when enhancements are enabled, we use the value of the base size and add the product of the HIF and the current input delay (ping in ms). The following formulas are used to calculate the new HB and HT:

$$HB = \text{Base size in px} + (HIF * \text{Ping in ms})$$

$$\text{E.g. } HB = 80px + (0.09 * 100ms) = 89px$$

$$HT = \text{Base time in ms} + (HTIF * \text{Ping in ms})$$

$$\text{E.g. } HT = 1s + (2.21 * 100ms) = 1.221s$$

As the response time (delay) increases both of these enhancements are dynamically applied to the game as the player progresses through it. Similarly, with a decrease in response time, the effect of the enhancement is diminished with the same HIF and HTIF values, returning the gameplay back to its original state when reference ping is reached, with the required TA and SA back to base level.

In order to effectively introduce delays within the gaming experience, an approach was employed, known as input queueing. This method was strategically implemented to ensure that the Phaser game seamlessly integrates with the user's actions in a manner that fulfills our requirements while the game loop perpetually executes in the background. Traditionally, when games are streamed from cloud servers, the input delay is a prevalent concern due to the inherent lag in transmitting user actions over a network. However, in the case of locally hosted games, where the game software operates directly on the user's hardware, the delay between input and response is typically minimal and often considered inconsequential.

To replicate the cloud gaming experience with its characteristic input delay, an additional layer of queueing was introduced into the game's architecture. This mechanism receives a continuous stream of the player's input commands, in much the same way as the actual standard input stream of the hardware; it captures and stores them for subsequent processing. The size of this input queue plays a pivotal role in determining the extent of the desired delayed input time because the queue is initially padded with a series of "blank" inputs.

The purpose behind this padding is to effectively create a temporal buffer, whereby any subsequent user input is appended to the end of the queue. Consequently, the actual input from the user experiences a delayed journey, gradually progressing toward the top of our designed input queue. This manipulation of the input queue is what culminates in the delay effect, reminiscent of the perceived "video" output delay commonly observed when gaming through cloud-based platforms.

B. Experiment Platform

In recent times, there has been a shift toward the utilization of crowdsourcing methodologies as a more practical and efficient alternative. This modern approach involves harnessing the collective intelligence of a large group of individuals, typically referred to as “crowd workers”. They are systematically engaged through digital platforms such as Amazon's Mechanical Turk, Upwork, Fiverr, or Crowdee. The tasks assigned to these workers are generally small, yet require a certain level of human cognition and discernment, a distinctive aspect that sets them apart from pure computation-based tasks. As an incentive for their contribution, these crowd workers are remunerated with financial rewards. It's noteworthy that this crowdsourcing approach has been previously evaluated within the context of interactive gaming QoE. As referenced in the study [21], Hearth, M. et al. investigated that revealed results which were obtained from the crowdsourcing methodology were consistent and aligned with those procured in traditional lab-based environments. This discovery supports the potential of the crowdsourcing approach in contributing valuable insights for QoE assessment in a more diverse, cost-effective, and efficient manner.

For our study, we utilized Amazon's Mechanical Turk (MTurk), a pioneering service from Amazon Web Services (AWS). It is an online crowdsourcing marketplace that has made a noticeable impact on the way research and surveys are conducted, by enabling access to a large and globally distributed pool of participants since its launch in 2005, particularly helping researchers and other requesters to achieve results for tasks that involve perception and judgment [22] [23]. We orchestrated a HIT on the MTurk platform to experiment for this study. The task included a series of questionnaires and gameplay of the Shooting Star game. We estimated that the survey would take anywhere between 8-12 minutes for the worker to complete. With each of the 5 rounds lasting for 60 seconds, a pre-game questionnaire, and a post-round questionnaire.

To ensure the quality and relevance of our data, we implemented a set of prerequisite conditions for potential participants and utilized the worker qualification options by using built-in quality control measures offered by MTurk, such as worker qualifications that participants must meet to accept specific HITs. This feature allows for precise targeting of specific participant populations [25].

Survey Parameters: The experiment was based upon recommendations provided in the ITU-T Rec. P.808 and ITU-T Rec. P.809, which are important standards developed by the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) that provide guidelines for evaluating the quality of multimedia streaming applications and services. These standards play a crucial role in assessing the quality of experience (QoE) for end-users and ensuring optimal performance of multimedia applications [26][27]. Recommendations from these standards include but are not limited to, gathering information about the crowd-worker, such as the web browser they are using to interact with our survey, details about the device that they are using, such as the resolution of their screen, window size, and any other sort interaction that they have with the survey. This information can then be stored for further evaluation and filtration. For the experiment conducted for this study, we ensured that we checked these details of the crowd-worker to ensure that they performed the survey in the desired conditions [27].

Another recommendation that was made part of this survey was the inclusion of the “trapping question.” This is a question that is added to the survey such that the answer to the question can determine whether the crowd-worker has randomly chosen an answer, or has provided a genuine response based on their experience with the survey. If the crowd-worker fails to satisfy the requirement of this engagement test, then their response submission can be discarded from the final data analysis. The questionnaire utilized a 5-point absolute category rating (ACR) scale in order to receive feedback from the participants. The ACR scale has been used in studies in the past to conduct experiments gauging the Quality of Experience and input (responsiveness) quality [15].

The language used for the scale included two kinds, the primary one being “Strongly disagree” (1), “Disagree” (2), “Neutral” (3), “Agree” (4), and “Strongly Agree” (5). This was used in most assessments, and when assessing the input quality by asking the user to answer a question such as “The responsiveness (input quality) in this round was as expected” which was asked in the context of the answer being relative to the ideal conditions of the reference round. A second set of options was used when gauging QoE, which asked the participant to rate their enjoyment and QoE in the round by selecting one of the options from “Very poor” (1), “Poor” (2), “Fair” (3), “Good” (4), and “Ideal” (5). The rating values attached to these answers were then used for calculating the mean of all ratings.

Execution of Experiment: With a maximum of 140 allowed submissions to the HIT that was created on MTurk, the submissions received exhausted this limit. Out of these, 98 submissions were obtained after filtering. The first filter applied was the removal of any responses in which the trapping question was answered unsatisfactorily. The second filter involved a closer inspection of the scores and the player's interaction with the game. Those responders who were found to have not actively participated in the game were also disqualified from the final list of crowd-worker responses. All crowd-workers had to participate in our designated task to complete their HIT. The primary task was structured into two main sections: a pre-game questionnaire and the interactive 5-round gaming experience of Shooting Star interspersed with post-round surveys. Table 1 below details this information about the participants.

Table 1: Statistical Data about the qualified participants’ ages

Number of participants	Minimum age	Maximum age	Average age	Std. Deviation
98	21 years	53 years	32.23 years	7.67

Subsequently, the participants were directed to play five rounds of the designated game with each round corresponding to one of the scenarios discussed below. Each gaming round was meticulously followed by a survey, allowing the participants to express their opinions on their experience based on the gameplay they had just completed. These sequential surveys ensured we received real-time feedback, capturing participants' immediate impressions after each round. After playing 60 seconds of the reference gameplay of Shooting Star, the participants played 3 rounds of simulated input delay, constant in each round, and increasing from low, medium, to high. The tested scenarios are explained in Table 2, detailing which scenario (and its subpart) contained what kind of delay simulation, the latency, and whether or not enhancements were enabled.

C. Limitations of our Work

Although our work is novel and unique, there are a few limitations of our work. Firstly, the outcome depends on the feedback given by the user. This makes the study prone to user bias based on area, liking of the game, etc. Secondly, our work has been validated on a simple game. This was due to the complexity of the research. The more complexities we add the more difficult it becomes for users to express the quality of Experience.

Table 2: Scenarios Related to Delays and Enhancements

Scenario	Delay Type	Ping (ms)	Enhancements
1	None (Reference test)	0 ms	None
2A	Constant, low-latency	120 ms	None
2B	Constant, medium-latency	280 ms	None
2C	Constant, high-latency	550 ms	None
3	Variable, none to high-latency	0-550 ms	Enabled

IV. RESULTS AND DISCUSSIONS

A. Gaming Habits:

The participant's playing habits were self-reported, and to ensure a structured and coherent interpretation of the data, these habits were categorized into five key categories: "Never", "Rarely", "Sometimes", "Often", and "Very often". These categories represent the spectrum of engagement with gaming, from complete non-involvement to high frequency of play. The description of what these categories represent is explained in Table 3.

For a visual representation of the percentage in each category, a pie chart is provided in Figure 4. The figure depicts the percentage of participants in each category from a total of 98. This visualization allows for a clearer comparison between the different categories. About one-third of the population in question can be grouped as participants who represent non-gamers, or gamers who are not experienced in playing shooter games.

B. Game Performance:

The description of performance matrices is given described below in Table 4. The scores were analyzed and grouped using the mean score in each round by all participants. Each round had its conditions. The first round was the reference round with no delay and no enhancements. The second round, with its 3 parts, had increasing levels of delay in each part, and the last round had dynamically changing delays with enhancements enabled. Figures 5, 6, 7, and 8 provide the statistics of this metric, which were recorded. This yielded one expected conclusion: with the increasing delay in the user inputs, the difficulty of the game would increase, and scoring would become more difficult, as apparent in the participants' scores in this experiment.

Table 3: Description of Playing Habits

Playing habit	Description
Never	Indicates participants who reported no engagement with shooter games. It should be mentioned that this report should not be decisively indicative of the players' habits in playing games belonging to other genres.
Rarely	Refers to those participants who play sporadically or have very infrequent engagement with shooter games.
Sometimes	Encapsulates the group of players who engage with games on an intermittent but consistent basis.
Often	Pertains to participants who reported frequent and regular engagement with activities involving shooter games.
Very often	Refers to the segment of participants for whom playing shooter games is a highly frequent and integral part of their routines.

Table 4: Descriptions of the Performance Metrics

Metric	Description
Score	The number of times the participant successfully shot a target.
Accuracy (%)	The percentage of successful hits to the total number of shots made.
Success rate (%)	The percentage of successful hits compared to the total number of target appearances.

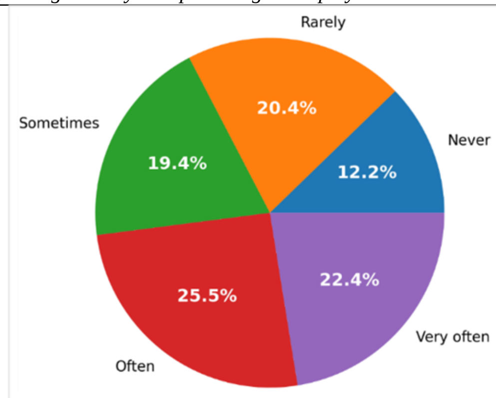

Figure 4: Percentage of participants in each category of playing habits

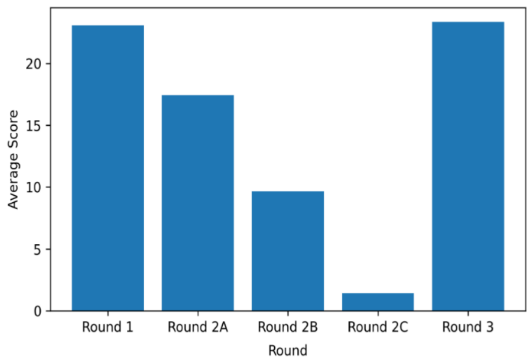

Figure 5: Average Scores in each round

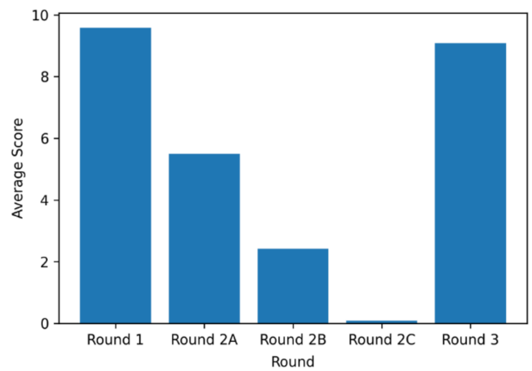

Figure 6: Average Scores of participants "never" engaged in shooting games

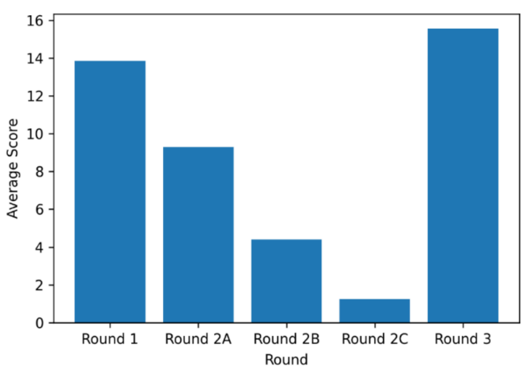

Figure 7: Average Scores of participants who "rarely" played shooting games

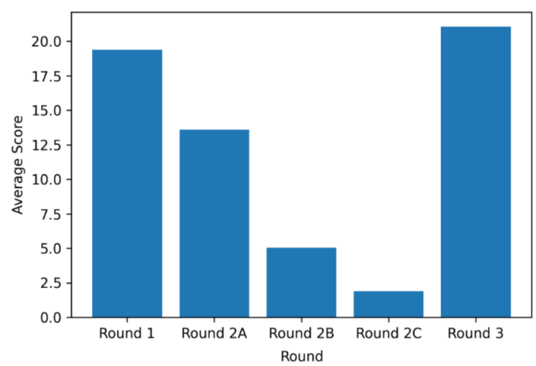

Figure 8: Average Scores of participants who "sometimes" played shooting games

C. Qualitative Analysis:

Two primary objectives of this study were to report the perceived quality of experience (QoE) and responsiveness quality as the test participants reported after completing the experiment. Instead of tracking scores or accuracies from the test game, this data relied upon the players providing their reports, leading to us performing a qualitative analysis of the data from the recorded responses. This inspection provided a clearer perspective on the reports, enabling the comparison of the perceived ratings of each round (each round representing a test scenario) or possibly aided in the drawing of any conclusion from the analysis. The participants' responses to our questionnaires were the primary sources of obtaining the measurements such as the quality of their experience, their perceived performance, and the quality of responsiveness that they perceived throughout them playing the game.

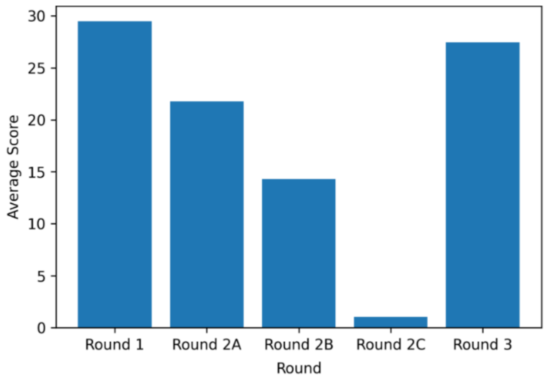

Figure 9: Average Scores of participants who "often" played shooting games

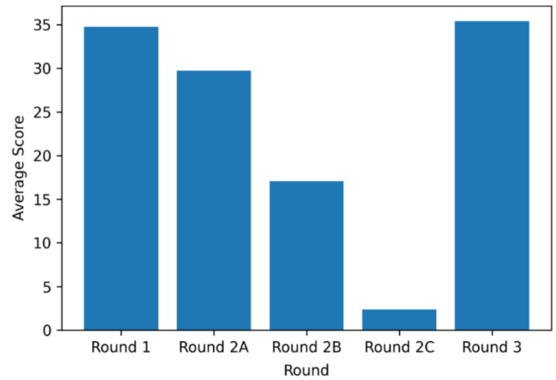

Figure 10: Average Scores of participants who played shooting games "very often"

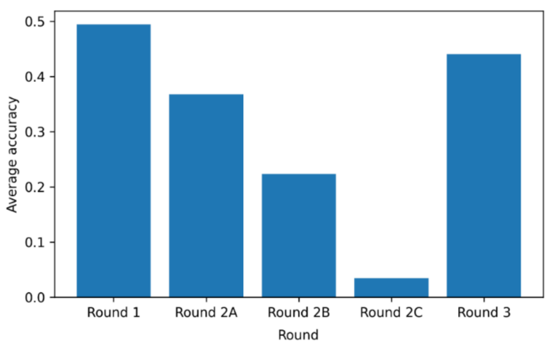

Figure 11: Average accuracy measured in each round for all participants

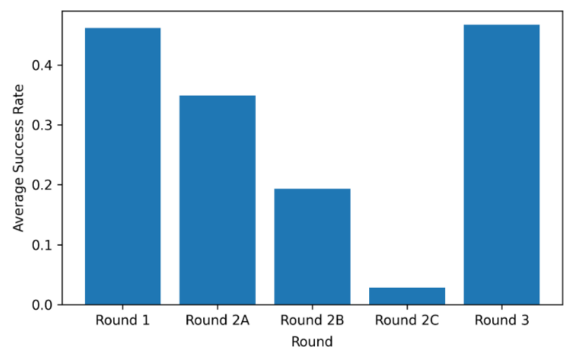

Figure 12: Average success rate measured in each round for all participants

D. Responsiveness (Input Quality):

The participants were asked after each round to rate the quality of the responsiveness that they perceived after playing each round. Figure 14 shows the results when represented on a scale from 1 to 5, with 1 being the poorest, and 5 being the most ideal rating. Since the reference round had 0 ms of delay, this was chosen as the reference for the ideal case. The responders were requested to rate the round that they played based on a comparison with the reference round. It was observed that the players immediately reported lower levels of responsiveness quality as soon as they were faced with a delay in the 2nd round (2A). The rating drops to ~2. This trend continued till the last delay round. For the last round, in which the participants faced delay but also had enhancements enabled for them, the perceived responsiveness quality showed a significant increase as compared to the rounds where enhancements were not enabled.

These findings followed a similar trend as found by Sabet, S. S., et al. such that the responsiveness rated by participants with enhanced gameplay yielded a higher rating of perceived quality [16]. One of the conclusions that may be drawn from the data, specifically the increased rating of responsiveness in the enhancement-enabled rounds, is that players may perceive responsiveness of their inputs subjectively according to their performance in the game, compared to how the participants feel they would perform in more ideal conditions. The increase in the time duration of the targets may also have been a major factor in players' perception of responsiveness. It may be due to the increased time of the target that participants were able to feel as though they were in more control of the reticle even in higher delay situations.

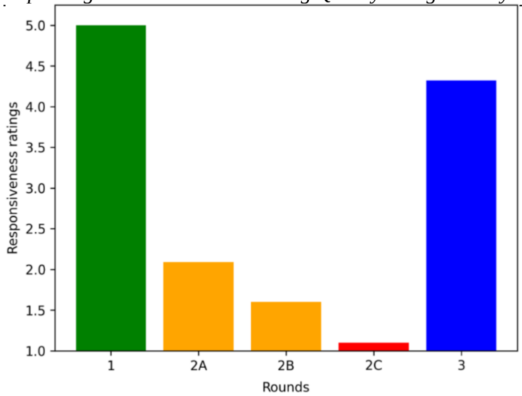

Figure 13: Reported responsiveness (input quality) ratings

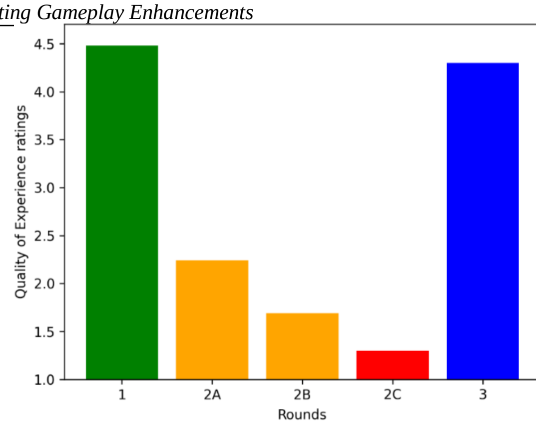

Figure 14: Perceived quality of experience (QoE) as rated by the participants

E. Quality of Experience:

The same method for calculating the mean rating was used for the perceived quality of experience (QoE) as was used for the responsiveness quality. The participants rated the quality of their gaming experience after each round. The average ratings by the participants are shown in the bar chart in Figure 14. We observed that players were reporting that the quality of their experience was almost equal to the perceived QoE reported for the reference round. This perception was possibly created due to the adjustment of the target time and size during higher levels of delay. The enhancements applied during the last round were designed to not just make scoring relatively easier in the round but were also balanced to attempt at not creating a target which was easier than the player would normally find to be challenging. This is possibly an explanation of the increased quality perceived by the participants, as their score level also reached similar levels as their scores in the reference round. Similar results were discussed in [16], as well as [14] and [15]. The trend in increased quality of experience in gameplay-enhanced rounds was continued in our study as well.

These findings also compare closely to the results of the reported input (responsiveness) quality, albeit the average quality of experience rating in the enhancements round exceeds the average responsiveness quality rating by a noticeable margin, which indicates that players found the quality of their gameplay experience to have improved more than the input quality.

While the last round of the test game did not rid the issue of the delay in the network, meaning that participants still faced delays, even with the enhancements enabled just with a reduced impact, the increased satisfaction of the participants of the responsiveness quality, while significant, was noticeably less than the rated QoE. Due to this factor, it is plausible that participants may have felt that ideal responsiveness was not achievable simply using enhancements. The average rating for the perceived quality of experience (QoE) provides a relatively optimistic viewpoint and indicates that the gameplay enhancements may have had a strong influence on players' QoE in cloud delay conditions.

V. CONCLUSIONS

To address the research gap observed in the existing literature, this study set the goal of evaluating dynamic gameplay enhancements across different latency ranges. By conducting thorough experimentation and analyzing the obtained data, valuable insights and results were obtained. These findings have the potential to contribute to understanding the effects of latency and gameplay enhancements for improved quality and performance in cloud gaming and provide crucial information for further improving overall cloud gaming experiences.

For future work on this subject, studies may shift focus towards performing these dynamic enhancement tests by applying them on more specific hardware. This may include conducting tests using only game controllers, so that more suitable enhancements may be discovered and studied in the same way that mouse-based gaming has been discussed in this study.

One more direction of the future work is to aid this work through artificial intelligence. At this point in time, we are concerned with user bias. Also, we hesitate to add complexity to our games as it would make the Quality of Experience more difficult and vaguer. If we use fuzzy systems and bots for playing the games and noting the differences, they can do it with more precision and with a lot more parameters that can enable us to get a Quality of Experience of even complex games.

References

[1] Cai, W., Shea, R., Huang, C. Y., Chen, K. T., Liu, J., Leung, V. C., & Hsu, C. H. (2016). A survey on cloud gaming: Future of computer games. IEEE Access, 4, 7605-7620.

[2] Cai, W., Shea, R., Huang, C. Y., Chen, K. T., Liu, J., Leung, V. C., & Hsu, C. H. (2016). The future of cloud gaming [point of view]. Proceedings of the IEEE, 104(4), 687-691.

[3] Lin, L., Liao, X., Tan, G., Jin, H., Yang, X., Zhang, W., & Li, B. (2014, November). LiveRender: A cloud gaming system based on compressed graphics streaming. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 347-356).

[4] Yates, R. D., Tavan, M., Hu, Y., & Raychaudhuri, D. (2017, May). Timely cloud gaming. In IEEE INFOCOM 2017-IEEE Conference on Computer Communications (pp. 1-9). IEEE.

[5] Yoon, O. (2021). Always Online DRM and Video Games. In AELJ Blog. 293, 2021.

[6] Ivkovic, Z., Stavness, I., Gutwin, C., & Sutcliffe, S. (2015, April). Quantifying and mitigating the negative effects of local latencies on aiming in 3d shooter games. In Proceedings of the 33rd annual acm conference on human factors in computing systems (pp. 135-144).

[7] Sabet, S. S. (2022). The influence of delay on cloud gaming quality of experience. Springer.

[8] Claypool, M., Eg, R., & Raaen, K. (2016, October). The effects of delay on game actions: Moving target selection with a mouse. In Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts (pp. 117-123).

[9] Huang, C. Y., Chen, K. T., Chen, D. Y., Hsu, H. J., & Hsu, C. H. (2014). GamingAnywhere: The first open source cloud gaming system. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 10(1s), 1-25.

[10] Tian, H., Wu, D., He, J., Xu, Y., & Chen, M. (2015). On achieving cost-effective adaptive cloud gaming in geo-distributed data centers. IEEE Transactions on Circuits and Systems for Video Technology, 25(12), 2064-2077.

[11] Gao, Y., Wang, L., & Zhou, J. (2019). Cost-efficient and quality of experience-aware provisioning of virtual machines for multiplayer cloud gaming in geographically distributed data centers. IEEE Access, 7, 142574-142585.

[12] Hossain, M. S., Muhammad, G., Song, B., Hassan, M. M., Alelaiwi, A., & Alamri, A. (2015). Audio–visual emotion-aware cloud gaming framework. IEEE Transactions on Circuits and Systems for Video Technology, 25(12), 2105-2118.

[13] Beyer, J., & Varbelow, R. (2015, December). Stream-A-Game: an open-source mobile cloud gaming platform. In 2015 International Workshop on Network and Systems Support for Games (NetGames) (pp. 1-3). IEEE.

[14] Sabet, S. S., Schmidt, S., Zadtootaghaj, S., Griwodz, C., & Moller, S. (2018, December). Towards applying game adaptation to decrease the impact of delay on quality of experience. In 2018 IEEE international symposium on multimedia (ISM) (pp. 114-121). IEEE.

[15] 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).

[16] Halbhuber, D., Henze, N., & Schwind, V. (2021). Increasing player performance and game experience in high latency systems. Proceedings of the ACM on Human-Computer Interaction, 5(CHI PLAY), 1-20.

[17] Davey, R. (2019). Phaser 3. GitHub. Retrieved from https://github.com/photonstorm/phaser/

[18] Phaser 3 API Documentation. Retrieved from https://photonstorm.github.io/phaser3-docs/

[19] Sandra1me. (2022). Target-Practice-Game. GitHub. Retrieved from https://github.com/Sandra1me/Target-Practice-Game/

[20] Liu, S. (2022). The Impact of Latency on Players in First-person Shooter Games (Doctoral dissertation, Worcester Polytechnic Institute).

[21] Hirth, M., Borchert, K., Allendorf, F., Metzger, F., & Hoßfeld, T. (2019). Crowd-based Study of Gameplay Impairments and Player Performance in DOTA 2. In Proceedings of the 4th Internet-QoE Workshop on QoE-based Analysis and Management of Data Communication Networks, pp. 19-24.

[22] Amazon Mechanical Turk (n.d.). In Amazon Web Services Documentation. Retrieved July 12, 2023, from https://aws.amazon.com/mturk/

[23] Ipeirotis, P. (2010). Analyzing the Amazon Mechanical Turk marketplace. XRDS: Crossroads, The ACM Magazine for Students, 17(2), 16-21.

[24] Amazon Mechanical Turk. (n.d.). Requester User Interface Overview. Retrieved July 12, 2023, from https://requester.mturk.com/

[25] Chandler, D., & Shapiro, D. (2016). Conducting Clinical Research Using Crowdsourced Convenience Samples. Annual Review of Clinical Psychology, 12, 53-81.

[26] ITU-T Recommendation P.808 (2018), Subjective evaluation of speech quality with a crowdsourcing approach.

[27] ITU-T Recommendation P.809, (2018). Subjective evaluation methods for gaming quality.

[28] Kougioumtzidis, G., Vlahov, A., Poulkov, V. K., Lazaridis, P. I., & Zaharis, Z. D. (2024). QoE Prediction for Gaming Video Streaming in ORAN Using Convolutional Neural Networks. IEEE Open Journal of the Communications Society.

[29] Wu, J., Guan, Y., Mao, Q., Cui, Y., Guo, Z., & Zhang, X. (2023, September). ZGaming: Zero-latency 3D cloud gaming by image prediction. In Proceedings of the ACM SIGCOMM 2023 Conference (pp. 710-723).

[30] Baena, C., Peñaherrera-Pulla, O. S., Camacho, L., Barco, R., & Fortes, S. (2023). Video Streaming and Cloud Gaming services over 4G and 5G: a complete network and service metrics dataset. IEEE Communications Magazine, 61(9), 154-160.

[31] Baena, C., Peñaherrera-Pulla, O. S., Barco, R., & Fortes, S. (2023). Measuring and estimating key quality indicators in cloud gaming services. Computer Networks, 231, 109808.

[32] Sabet, S. S. (2023). The influence of delay on cloud gaming quality of experience (pp. 1-127). Berlin/Heidelberg, Germany: Springer.