March 2019

The Effects of Network Latency on Player Gaming Experience

Mei Xintong Zha
Worcester Polytechnic Institute

Ying Zhang
Worcester Polytechnic Institute

The Effects of Network Latency on Player Gaming Experience

An Interactive Qualifying Project Report

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Bachelor of Science

By

Meixintong Zha

Ying Zhang

Submitted to

Professor Mark Claypool

Worcester Polytechnic Institute

Abstract

Humans spend 3 billion hours a week playing video games [6]. While playing, network latencies can cause interaction delay between the client and the server and affect players' gaming experiences. While there is research about the effects of delay on whole network game systems, there is little research on the effects of delays on fundamental player actions. We built two games that isolate two fundamental game actions (shooting and movement) to evaluate how players are affected by network latency. Game statistics were used to evaluate performance, and emotion detection software and a heart rate monitor were used to evaluate players' stress level during gameplay. Our results from a 36-person user study show that players' performance decreases as latency increases and players' stress level increases as latency increases.

Content

Section 1 Introduction

Imagine playing a ranked game in League of Legends (Riot Games, 2009), with your most skilled champion and a commitment to winning this match. Suddenly your network acts up—every time you cast spells, especially those that are direction-targeted, you can feel a delay between when you click and the spell triggers and as a result, you miss everything. How much patience do you have before quitting the game? How angry are you before trying to smash something?

Like highway traffic congestion in real life, latency because of network congestion can be really annoying in the virtual world. A network delay of 500 additional milliseconds can cause online game players to leave the game, and the abandonment rate can reach 87 percent with a 2-second delay in load time [6]. In order to keep players playing and to provide a better gaming experience, game developers and researchers are seeking to improve delay compensation techniques for games.

For finding the best delay compensation techniques, we need to understand how latency affects the player's performance in the game. Player's actions vary according to different types of games. For example, for a MOBA (Multiplayer Online Battle Arena) game like League of Legends (Riot Games, 2009), players sometimes are required to aim in a certain direction or area to hit other champions instead of targeting other champions directly like some actions in First Person Shooters (FPS) games [10]. While there are many studies about how the player is impacted by the delay from local systems, networks and servers, little research had been done on delay impact on the fundamental player actions. Experiment on fundamental player actions like movement and shooting at moving targets under different amounts of delay can be extrapolated to different types of actual games and used in improving delay compensation techniques.

Different genres of real-time games possess different network requirements in order to provide a smooth gameplay experience for users. However, online gameplays are always accompanied by some amount of delay. Sometimes, the delays are not high enough to be noticeable to players. Studies show that while Warcraft III (Blizzard, 2002) is played in real-time, reaction time plays a small role compared to understanding the game [3]. The effects of typical network delay(less than a second) do not impact the overall outcome. RTS (Real Time Strategy) and RTT (Real Time Tactic) games have delay requirements mostly similar to that of Web browsing games (on the order of seconds) [3]. Actions that require precise (precision), rapid responses (short deadline) are greatly impacted by degradations in frame rates and delay [2]; FPS (First Person Shooter) and MOBA (Multiplayer Online Battle Arena) are two typical types of game which typically lower less tolerance to latencies.

To understand how lag affects a player's experience, we need to understand the factors of Quality of Experiences (QoE) such as stress and performance. Online gameplay involves multiple motivations, such as achievement, social interaction, and immersion. Researchers found that openness and agreeableness are positively related to more hours of online gameplay, while gamer conscientiousness and emotional stabilities are negatively correlated to more online gameplay [5]. Latency affects user emotion when their performances are degraded; this creates frustrating and stressful gameplay situations for users, maybe causing them to leave the game

[4]. High latencies can exasperate the cycle of stress and poor performance. It may be helpful for game developers and network providers to know how latency affects the player.

Our goal for this project is to conduct a user study with an atomic action based game for evaluating how the player's gaming experience is affected by network latency. We break down the measurement of experience into two subsets— performance and stress. In our experiments, we build two games that isolated two atomic actions commonly used in most games; clicking to shoot at a target in *Sushi Shooter*, and clicking to dodge the falling object in *Square Dodger*. We apply the "Method of Limits" to determine what range of latency is acceptable versus unacceptable by gradually increasing the intensity of latency in discrete steps and analyzing players' performance while measuring stress through emotion detection software and a heart rate monitor [1].

Results of a 36-person user study show that, in both games, players' performance decreases as latency increases. In other words, as the added latency increases, players are more likely to miss the target in *Sushi Shooter* and more likely to get hit by the falling squares in the *Square Dodger*. Results from the emotion detection data show that players' positive emotion decreases as latency increases, and players' negative emotions increase with the added latency. This suggests that players' emotion is negatively affected by network latency during gameplay, which indicates an increase in players' stress as latency increases.

The rest of this report is organized as follows: section 2 describes the background and related work; section 3 presents our methodology for game design and experiment design; section 4 analyzes the experiment results, including data-visualized graphs; and section 5 summarizes the report and discusses possible future works.

Section 2 Background and Related Works

For online video games, network latency is one type of lag that affects the ping of the online game. Players' gaming performance and stress usually degrade due to network latency; the stress level of players will be affected. This chapter proving background knowledge for our experiments on the effects of network latency on the players gaming experience, including sources of delay, Quality of Experience, stress measurement, and game types/actions.

2.1 Sources of Delay

Unlike single-player games which operate on the local machine, an online-game runs on a central server to maintain consistency between individual clients. The clients send change requests to the server and update the local game state by receiving updates from the server. This delay in communication between clients and server is the fundamental source of lag. Other sources of lag can be categorized to 1) hardware deficient at the client-server, and 2) a poor network connection between the client and server [13].

The hardware problems tie to the game architecture. Games consist of a loop of frames, and accept user input and perform calculations during each frame. The game updates the state and produces output. The frequency of generating frames is referred to as the *frame rate*. For online games, the updates are sent to the central server from the client and back to the client to complete the update. A low frame rate makes the game less responsive to updates and may force it to skip outdated data.

Network delays are caused by bandwidth, congestion, the physical distance between the end-systems and the wireless network [14]. Wireless networks tend to cause higher latency than wired networks because of electromagnetic interference come from other devices such as microwaves.

2.2 QoE

Quality of Experience (QoE) can be impacted by network latency during gameplay. *QoE* is a measure of the delight or annoyance of a customer's experiences with a service [11], which in our study refers to online games. A variety of methods can be used for QoE measurement with a common method being human rating, Claypool studied about how computer games player actions are impacted by the delay from the local system, networks and servers [8]; other subjective tests are done by Rahul Amin and his team. They developed a Mean Opinion Score (MOS) metric to determine each gamers' QoE, based on four post-survey questions: Gameplay Satisfaction, Gameplay Frustration, Impact of Lag on Gameplay, and Likeliness to change network service providers [6].

The method of measuring user subjective QoE in our paper is by conducting a survey, which asks users to rate the quality of responsiveness from 1 (low) to 5 (high) for each combination of delay and speed.

In addition to subjective tests for measuring user QoE, objective tests, which are more technology-centered, can provide quality results faster [12]. Boyan et al, measured players' QoE by detecting their emotional states. They first measured the level of presence of basic emotions like happiness, sadness, engagement, anger, and fear. Then detected physiological data like skin conductivity and/or heart rate variability. They found statistically significant correlations between attention and presence, eye closure and presence, eye closure and flow, and engagement and surprise [7]. Similarly, Drachen et al, studied heart rate and electrodermal activity correlations with subjective gameplay experience testing the feasibility of these measures in commercial game development contexts. Their results indicate a significant correlation ($p < 0.01$) between psychophysiological arousal and self-reported gameplay experience [9].

2.3 Stress Measurement

Stress is an important factor to measure in many fields related to human psychology and is an aspect of our study for finding out how player's stress levels are influenced by network latency. Many methods have been developed for stress measurement. For example, stress assessment tests can indicate people's level of stress by asking them questions; Biodots that can indicate people's level of stress by different colors when stuck on a person's skin. Unfortunately, the Biodots does not work for our project, because the color change is based on skin temperature, assuming that people's skin temperature decrease when they are stressful. While the situation could be really different for people who are playing an exciting game, their skin temperature could increase when their stress levels increase.

Automatic facial behavior analysis software can provide an effective measurement of stress by detecting and tracking human faces. According to Deshmukh et al [21], there are three stages that are most commonly used in facial emotion recognition: face detection and tracking, where the software finds the face without human intervention; feature extraction, where the software extracts the information from the facial expressions; and expression classification, where the software recognized the emotion based on the given information [21]. There are many effective emotion detection applications that have free use by the public. The software we used is called the EmotionDetectionAsset, which is developed by the RAGE project of the Open University of the Netherlands (OUNL) [https://github.com/rageappliedgame/EmotionDetectionAsset].

2.4 Game Types/Actions

An online game is a video game that is partially or primarily played through the Internet. Online games can be from many genres, including First-person shooter (FPS), Real-time Strategy (RTS), Multiplayer Online Battle Arena (MOBA), Massively Multiplayer Online (MMO), Role-Playing Game (RPG), Action, or Sport [15]. Each genre has different game actions. RTS games have command issuing (e.g attack, defend, end turn) game actions, while FPS games have character movement, shooting, and base capture game actions. RPG and MOBA games design a collection of characters with unique appearance and skills for different game action outputs. Generally, character movements and projectile skill shots are two common game actions for Action, FPS, and MOBA games. We study the effects on latency on both in the project.

Section 3 Methodology

This chapter contains the game design choice, player performance output, stress measurement, testing environment, and procedure.

3.1 Game Design

Two fundamental atomic actions are represented presented in our games: i) shooting ii) dodging.

Game design principles that we incorporated are:

1. Control the amount of latency between the player input and the resulting game action.
2. Game actions are atomic and controlled by mouse clicks.
3. Game difficulty—balanced between gamers as well as users who are not familiar with computer games
4. Reward system that motivates players to win and influences their heart rates or emotions
5. Record user and game event for performance measurement

According to these principles, we designed two games with each game corresponding to one atomic action. The two games were designed with different color schemes and character appearances to reduce visual fatigue for the players. The shooting game, named *Sushi Shooter,* was designed for testing the shooting action and its hit rate under different input latency. The dodging game, named *Square Dodger*, was designed for testing the horizontal movement action and the success rate of dodging under different input latency.

3.1.1 Design Choice

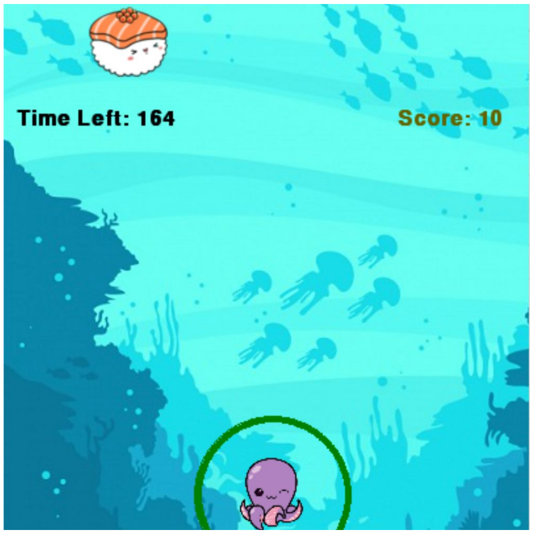


Figure 3.1.1a

Our goal was to design a simple shooting game where the player clicks the mouse to shoot a bullet from the bottom of the window toward the enemy located at the top of the window. We built *Sushi Shooter* with Python using Pygame [https://www.pygame.org/docs/]. The Pygame library has the display, sound, event, draw, and time modules to develop a game; it is highly portable and can run on nearly every platform and operating system. The screen size of *Sushi Shooter* is 400 x 400 pixels with the game played in fullscreen mode to reduce background distraction. The player-controlled character is an octopus and the target is sushi. We use an ocean background image. The purpose of the light blue color and simple character appearance is to reduce any visual distraction and make the player focus on the controls and scoring.

Python has the “pygame.time” module for monitoring time: pygame.time.set_timer (*USEREVENT*, *millisecond*). The “set_timer” function takes the user-event ID and the time in milliseconds to activate an event in the corresponding time. We attached the shooting event with latency after the mouse click action by using this function.

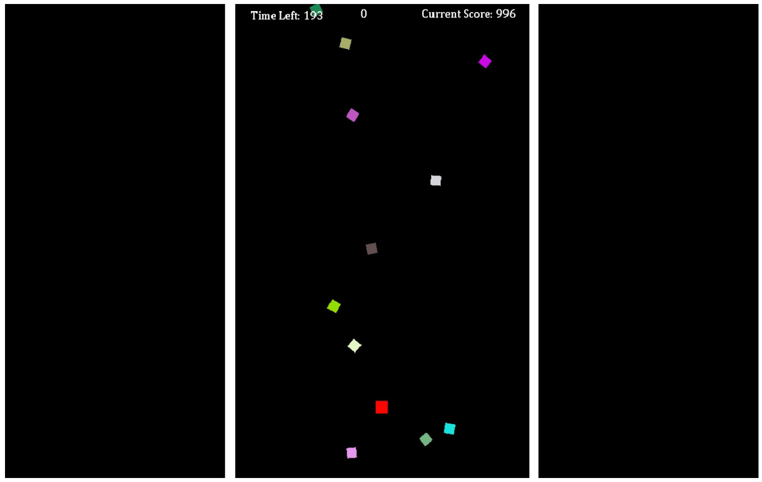


Figure 3.1.1b

The second fundamental action we wanted to test is movement - specifically horizontal movement by mouse clicks. During our prototype designing stage, skillshot dodging and maze-based dodging designs were considered because these game types reduce distractions. Our final design is to dodge falling squares with different speeds. The dodging game was built in GameMaker Studio, which is a cross-platform game engine developed by YoYo Games [https://www.yoyogames.com/]. As a popular tool among amateur and freeware developers, GameMaker is flexible and supports fast game development, especially for our project. The in-built alarm function allows simulated lags by adjusting the response time of the mouse click.

The original size of the game is 510 x 768 pixels and played in fullscreen during testing. The background was black, the speed of falling squares was randomly assigned from 5-20 pixels per second; the color of falling squares are randomly assigned from the color palette, the size of the falling squares is 8x8 pixels while the size of the player controlled red box is 10 x 10 pixels.

3.1.2 Player Controls and Features

In *Sushi Shooter*, the player clicks the left mouse button to shoot a bullet vertically up. We also ask the player to press spacebar whenever the player feels the latency. This function adds a subjective measurement to the player quality of experience. Each time as the player press space bar, the octopus will blink once. There is a countdown clock on the left side and a score tracker on the right-hand side.

For *Square Dodger*, the player can only move horizontally with left and right mouse clicks. Participants were asked to press spacebar whenever they feeling the latency, and there is an indication at the upper left corner when the spacebar was pressed (Figure 3.1.2). Every time the player is hit, the health at upper right corner decreases by 5. The maximum health was set to 1000 so that it is impossible for a player to fall to negative health before the game ends.

Figure 3.1.2

3.2 Sushi Shooter: Game Balance

The steady and repetitive movement of the target sushi and the cooldown of the bullet make *Sushi Shooter* a performance capped game. We adjust the speed of the target and bullet so that the game is challenging for most players while it is still not too difficult for the player to hit the target.

3.2.1 Movement Speed

In *Sushi Shooter*, the three variables that decide the difficulty of the game are the bullet speed, target sushi movement speed, and added latency. We initially determined the bullet speed and sushi movement speed based on the average projectile skill shot speed and character movement speed in League of Legends to ensure playability. The average character movement speed in L*eague of Legends* is 450 units per second[17]. The skill shot speed in *Legends of Legends* ranges from 600 to 2000 units per second[18]. The ratio of skill shot speed to character movement speed is ranged from 1.3 to 4.4. We tested the shooting with different speeds and finalize the two variable: 3 pixels per millisecond for the sushi movement speed and 7 pixels per millisecond for the bullet speed, results in a 2.3 ratio.

3.2.2 Maximum Score and Target Score

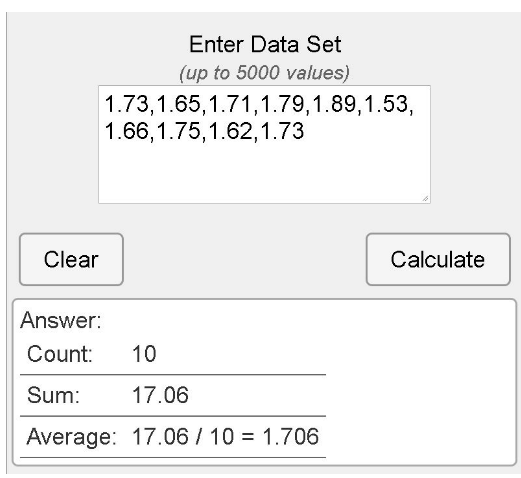


Figure 3.2.2

We used a stopwatch and found that the time for the sushi to traverse from one side to another is 1.7 seconds. Accounting for the bullet cooldown time and the increasing lag, it is possible to get one score each time the sushi traverses the screen. The maximum possible score for a 200-second game is $\frac{200sec}{1.7sec} \approx 118$. Featuring scores and feedback is one component of the flow of games that serve as an intrinsic motivator in brand use and selection[16]. *Sushi Shooter* is a straightforward game with one atomic action. We want the user to try throughout the entire 3 minutes 20 second of the game. We decided to tell the user a target score to maintain the game homeostasis - a process of achieving optimal levels of affective well-being by engaging in activities that raise arousal if it is currently too low, or activities that lower arousal if it is too high [16].

3.2.3 Beta Testing and Modification

We conducted beta tests and surveys with 8 people. We told the testers that the maximum possible score is 118, and we asked users to press the spacebar whenever they feel the latency. Their scores ranged from is 20 to 62, with an average of 43.5. In the survey, testers respond that the game mechanic is simplistic enough for them to get familiar with 20 seconds. Testers thought the possible motivational target score for the experiment should be ranged from 60 to 100. Some of the testers did not feel lag until the 400 to 500 ms ping. Testers experienced different levels of anxiety varying from person to person in the last minute of the game, but most testers forgot to press the spacebar when they felt the latency. To obtain the subjective measure from the spacebar, we have to remind the participants to press the spacebar when they feel the lag before each game.

The target score for the beta test is 118 - the maximum possible score; the target score for the experiment is adjusted according to the top score from beta testing. The top score from beta testing is 80.

3.3 Player Performance Output

One objective measure of Quality of Experience is player performance. We predict that the player will play less optimally with higher latency. We are particularly interested in click counts, hit rate (miss rate), and spacebar counts (for subjective Quality of Experience) versus latency, so the user events and the corresponding time need to be recorded. *Sushi Shooter* and *Square Dodger* record each player's action, such as mouse clicks and spacebar presses, along with the result of the action, such as missing the target in *Sushi Shooter* or being hit by a square in *Square Dodger*.

3.4 Stress Measurement Design

Stress is defined as a state of mental or emotional strain caused by adverse circumstances. We measure the heart rate and emotions of the player to evaluate the change of stress level.

3.4.1 Emotional Detection

The hardware that was used for detecting participants' face is a 1080p webcam from Logitech as the figure 3.4.1a.

- Full HD 1080p video calling (up to 1920 x 1080 pixels) with the latest version of Skype for Windows
- 720p HD video calling (up to 1280 x 720 pixels) with supported clients
- Full HD video recording (up to 1920 x 1080 pixels)
- H.264 video compression
- Full HD glass lens with precise autofocus
- Built-in dual mics with automatic noise reduction
- Automatic low-light correction
- Tripod-ready universal clip fits laptops and LCD monitors

Figure 3.4.1a

The software that was used for analyzing participants' emotion, named EmotionDetectionAsset, is developed by the RAGE project of the Open University of the Netherlands(OUNL) [https://github.com/rageappliedgame/EmotionDetectionAsset]. This asset is a client-side software component that can detect emotions from players' faces in real-time. It returns a string representing six basic emotions: happiness, sadness, surprise, fear, disgust, anger, and can also detect the neutral face. It is recommended by the authors to use this software in games to collect emotion data during playtesting, which is what we did for our project.

The EmotionDetectionAsset was run in the background to monitor the player's emotion while playing the game with different delay ranges. The data for the six basic emotion was extracted to analyze the relationship between network delay and the player's emotion.

3.4.2 Heart Rate Measurement

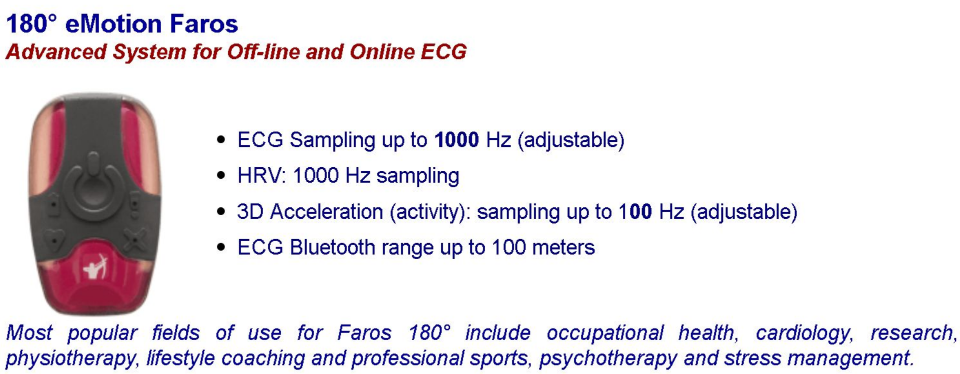


Figure 3.4.2a

Heart rate, or pulse, is the number of times the heart beats per minute. We chose to use the eMotion Faros sensor [http://ecg.biomation.com/faros.htm] to obtain users' ECG, accelerometer_x, accelerometer_y, accelerometer_z, marker, and HRV(heart rate variability). These data are in EDF format. We used EDFbrowser to convert the EDF file to a plain text file. Each line of the text file contains the time and the above six fields. We parse the file with the start time from the game output files to generate a CSV file with attributes of time and corresponding HRV.

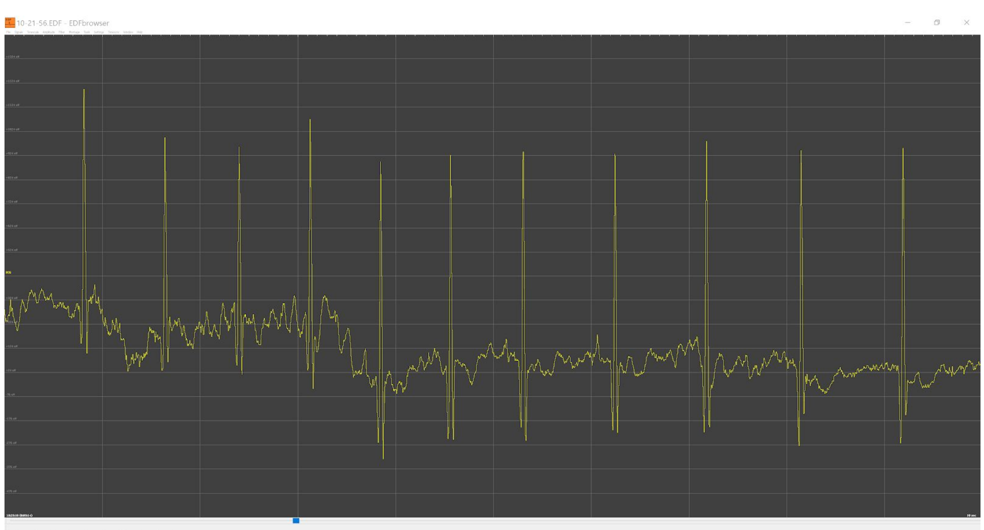


Figure 3.4.2b

495.068000,-126.000000,,,,
495.072000,-167.000000,,,,
495.076000,-231.000000,,,,
495.080000,-328.000000,976.500000,259.500000,246.000000,,
495.084000,-191.000000,,,,
495.088000,-159.000000,,,,
495.092000,-436.000000,,,,
495.096000,-395.000000,,,,
495.100000,-264.000000,,,,
495.104000,-268.000000,,,,
495.108000,-122.000000,,,,
495.112000,-3.000000,,,,
495.116000,243.000000,,,,
495.120000,515.000000,976.500000,255.500000,267.500000,,
495.124000,1016.000000,,,,
495.128000,1668.000000,,,,
495.132000,2042.000000,,,,
495.136000,1838.000000,,,,
495.140000,1288.000000,,,,
495.144000,761.000000,,,,
495.148000,231.000000,,,,
495.152000,-399.000000,,,,
495.156000,-700.000000,,,,
495.160000,-681.000000,992.000000,220.500000,267.500000,,
495.164000,-447.000000,,,,
495.168000,-254.000000,,,,
495.172000,-262.000000,,,,
495.176000,-386.000000,,,,
495.180000,-297.000000,,,,
495.184000,-98.000000,,,,
495.188000,-169.000000,,,,
495.192000,-203.000000,,,,
495.196000,-45.000000,,,,
495.200000,-16.000000,994.000000,189.000000,250.000000,,552.00000
0
495.204000,-236.000000,,,,
495.208000,-381.000000,,,,
495.212000,-411.000000,,,,
495.216000,-240.000000,,,,
495.220000,62.000000,,,,
495.224000,-108.000000,,,,
495.228000,-218.000000,,,,
495.232000,-108.000000,,,,
495.236000,-16.000000,,,,
495.240000,31.000000,992.000000,195.000000,228.500000,,
495.244000,66.000000,,,,

Figure 3.4.2c

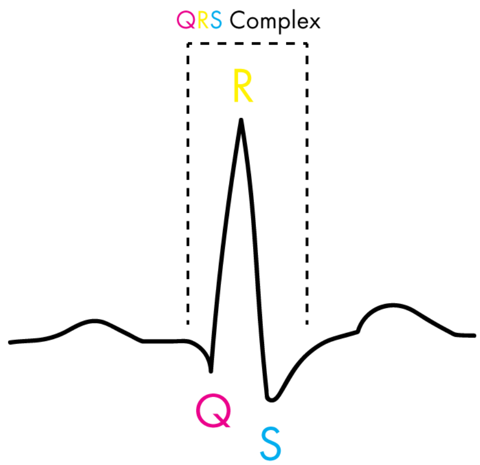


Figure 3.4.2d

Heart rate variability (HRV) measures the specific changes in time (or variability) between successive heartbeats. The time between beats is measured in milliseconds and is called an “R-R interval” or “inter-beat interval”(IBI) [19]. Researchers have found that an increase in HRV is related to increased self-control abilities, greater social skills, and better abilities to cope with stress, among other findings [19]. HR can be found by $\frac{60 \times 1000}{average\ HRV}$.

3.5 Testing Environment

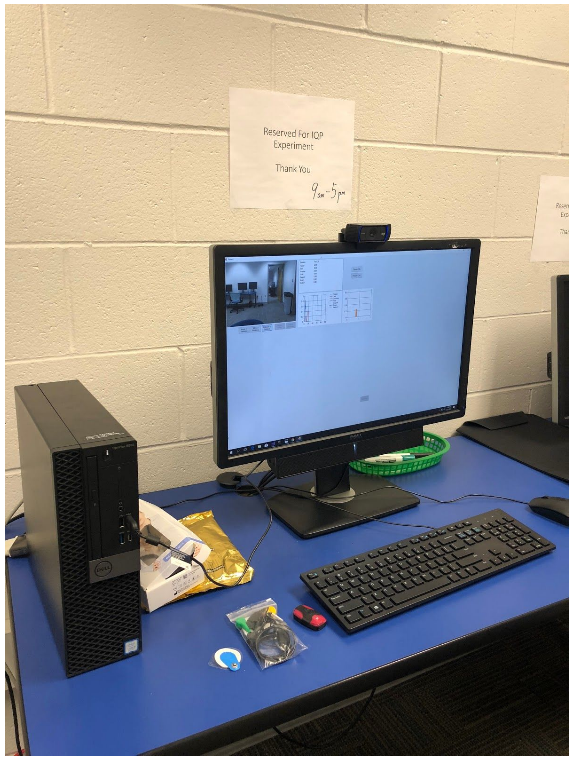


Figure 3.5

The experiment was conducted at Zoo lab in Fuller laboratories of WPI. The room was isolated from outside to eliminate potential distraction factors.

3.6 Procedure

The procedure followed for our user study was:

1. Confirm the email address with the participants and ask them to sign the Consent Form.
2. Ask participants to fill out the online Demographic Question survey.
3. Instruct participants on how to wear the heart sensor.
4. Open the emotion detection software in the background, adjust the angle of the camera to make sure it can detect the participants' face.

5. Remind participants to press spacebar whenever they feel lag.

6. Provide basic instruction for *Sushi Shooter* controls.

7. Remind participants of the target score (80).

8. Ask participants to play *Sushi Shooter.*

9. Save the emotion output file and reopen the emotion webcam.

10. Provide instructions on for *Square Dodger* controls

11. Remind participants of the game goal.

12. Ask participants to play *Square Dodger.*

13. Save the emotion output file.

14. Ask participants to fill out the online opinion summary questionnaire.

15. Debrief the participants and collect the heart rate sensor.

Section 4 Analysis

This chapter contains an overview of the test demographics, the tools used for data parsing, and the analysis of user performance, heart rate variability, and emotion data.

4.1 Demographics

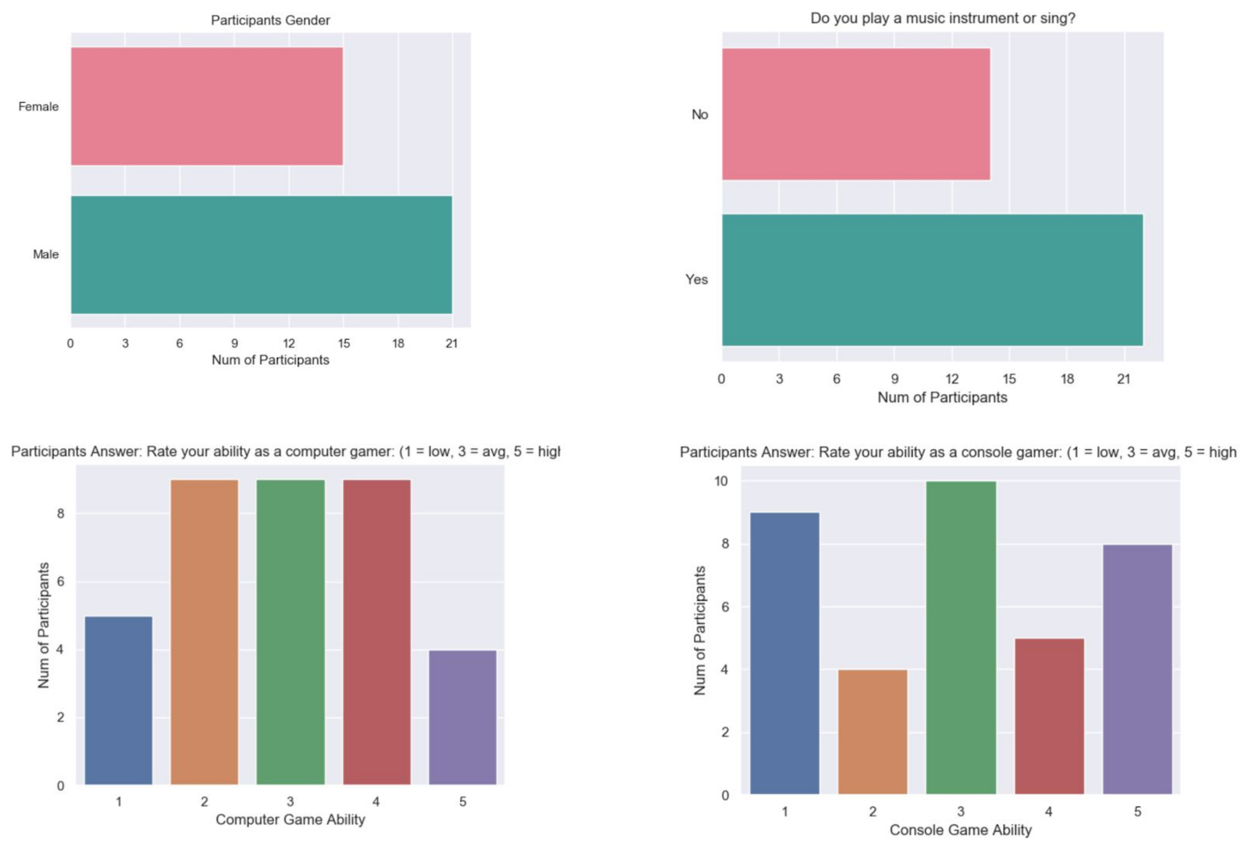


Figure 4.a: Demographics

We recruited participants from Worcester Polytechnic Institute students. There were 36 participants with 21 males and 15 females (Figure 4.a: top left). Participants have different levels of computer/console gaming ability (Figure 4.a: bottom left and bottom right). The average computer gaming ability of the participants is 2.83. The average console gaming ability of the participants is 3.11.

4.2 Data Parsing

The performance and emotion raw data are in text format. The heart rate variability data are in EDF format. We used EDFbrowser (https://www.teuniz.net/edfbrowser/) to export the EDF files to text files. We used the Python CSV module (https://docs.python.org/2/library/csv.html) to create CSV files from the text files. We used Python Pandas module (http://pandas.pydata.org/pandas-docs/stable/) and Google Spreadsheet to calculate the average, counts, and aggregation of data attributes.

4.3 User Performance Analysis

4.3.1 Sushi Shooter

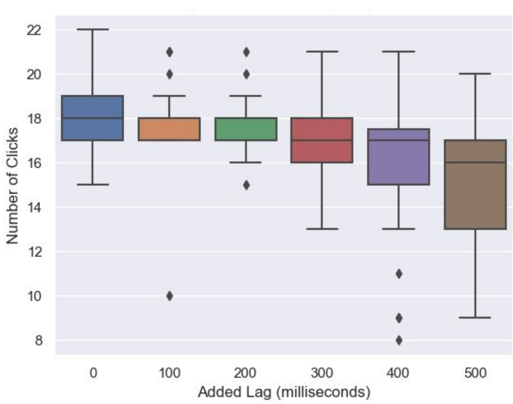


Figure 4.b.1 Sushi Shooter: Click vs Added Lag
Boxplot: distribution of all participants

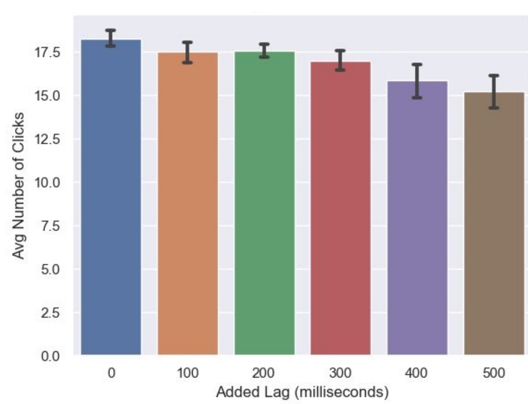


Figure 4.b.2 Sushi Shooter: Click vs Added Lag
Bar chart: average click counts with standard error

Figure 4.b.1 shows a boxplot of the distribution of users clicks counts at six latency levels. The x-axis represents added lag in milliseconds; the y-axis represents the number of mouse clicks.

Figure 4.b.2 shows a bar chart of the average number of clicks for all participants at six latency levels. The x-axis represents the added lag in milliseconds; the y-axis represents the average number of clicks for all participants. The black bar on top of each colored bar represents the standard error. These graphs show a decreasing trend which means users tend to click less as added lag increases.

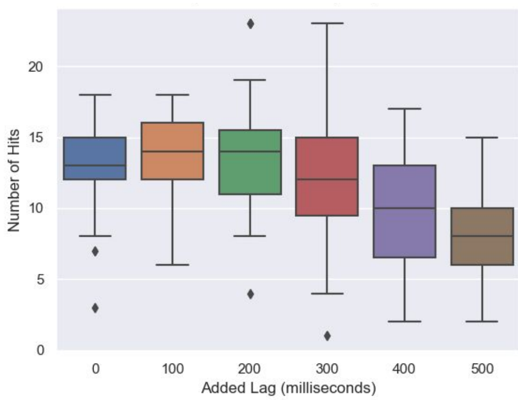


Figure 4.b.3 Sushi Shooter:Hit vs Added Lag
Boxplot: distribution of all participants

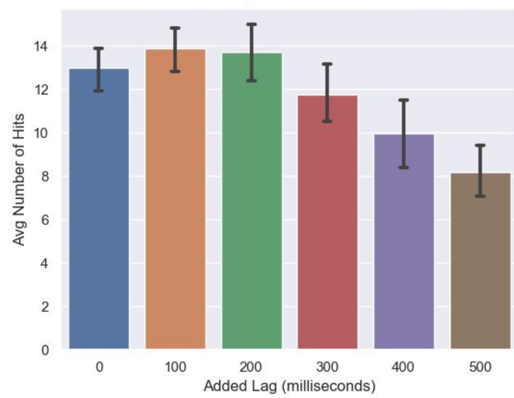


Figure 4.b.4 Sushi Shooter: Hit vs Added Lag
Bar chart: average hit counts with standard error

Figure 4.b.3 shows a boxplot of the distribution of users hit counts at six latency levels. The x-axis represents added lag in milliseconds; the y-axis represents the number of bullets that hit the sushi. Show with standard error bars, Figure 4.b.4 is a bar chart of the average number of hits for all participants in six latency levels. The x-axis represents the added lag in milliseconds; the y-axis represents the average number of misses for all participants. These graphs show an overall decreasing trend which means users tend to hit the sushi fewer time as the added lag increases.

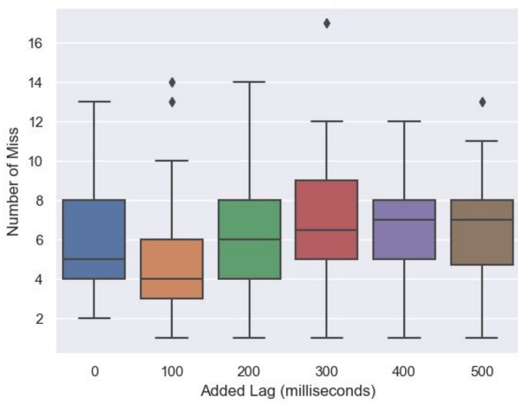


Figure 4.b.5 Sushi Shooter: Miss vs Added Lag
Boxplot: distribution of all participants

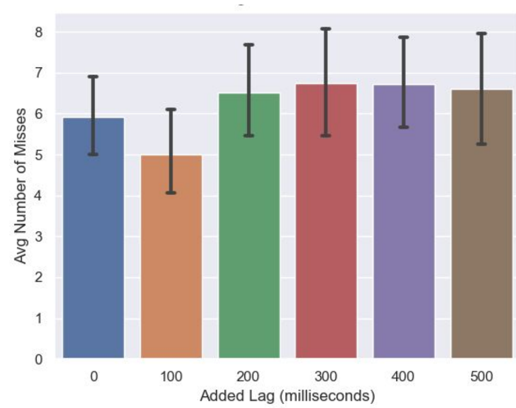
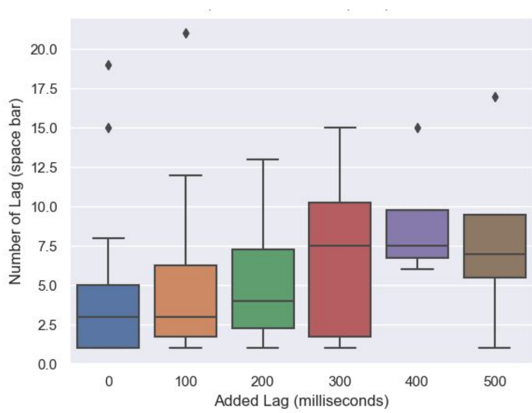


Figure 4.b.6 Sushi Shooter: Miss vs Added Lag
Bar chart: average miss counts with standard error

Figure 4.b.5 shows a boxplot of the distribution of users misses counts at six latency levels. The x-axis represents added lag in milliseconds; the y-axis represents the number of bullets misses the sushi. This graph shows that users tend to miss the same number of shooting from 200 to 500 milliseconds of added lag. There is no visual trend in this graph which suggests there is no space correlation of added lag and miss rate. Figure 4.b.6 is a bar chart of the average number of misses for all participants at six latency levels shown with standard error bars. The x-axis represents added lag in milliseconds; the y-axis represents the average number of hits for all participants. This graph shows that, on average, users tend to miss the same number of times when shooting from 200 to 500 millisecond of added lag.



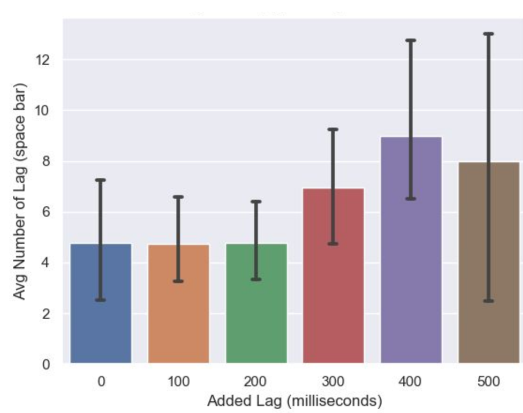


Figure 4.b.7 Sushi Shooter: Lag (spacebar) vs Added Lag
Boxplot: distribution of all participants

Figure 4.b.8 Sushi Shooter: Lag (spacebar) vs Added Lag
Bar chart: average spacebar counts with standard error

Figure 4.b.7 shows a boxplot of the distribution of users spacebar presses at six latency levels. The x-axis represents added lag in milliseconds; the y-axis represents the number of space bar presses. This graph shows an increasing trend from 0 to 300 ms added lag and a decreasing trend from 300 to 500 ms added lag. There is no apparent visual correlation between added lag and user self-report of lag.

Figure 4.b.8 is a bar chart of the average number of users spacebar presses for all participants at six latency levels shown with standard error bars. The x-axis represents added lag in milliseconds; the y-axis represents the average number of spacebar presses. This graph shows that, on average, users tend to press the spacebar same number of times from 0 to 200 millisecond of added lag. The standard error bars of 0, 400, 500 suggest no statistical significance.

4.3.2 Square Dodger

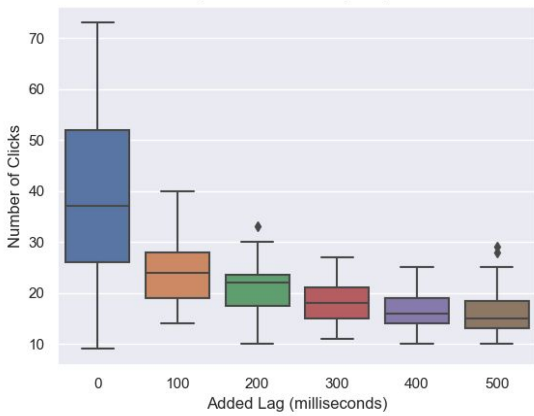


Figure 4.c.1 Square Dodger: Click vs Added Lag
Boxplot: distribution of all participants

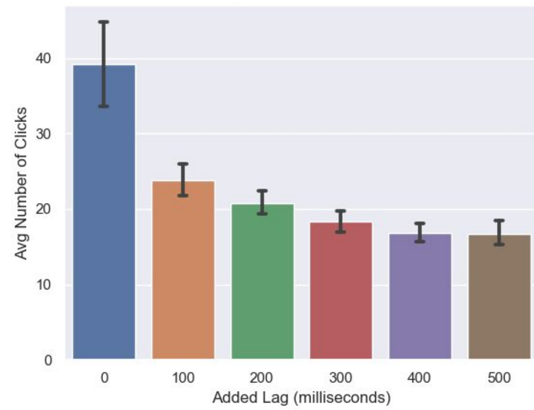


Figure 4.c.2 Square Dodger: Click vs Added Lag
Bar chart: average click counts with standard error

Figure 4.c.1 shows a boxplot of the distribution of users clicks counts at six latency levels. The x-axis represents added lag in milliseconds; the y-axis represents the number of mouse clicks. This graph shows an exponentially decreasing trend which suggests users tend to click less as the added lag increases and the clicks do not reduce as drastically once added lag reaches a threshold.

Figure 4.c.2 is a bar chart of the average number of clicks for all participants at six latency levels shown with standard error bars. The x-axis represents the added lag in milliseconds; the y-axis represents the average number of clicks for all participants. The decreasing trend in this graph matches Figure 4.c.1. The high standard error at 0 ms latency added means the users have a wide range in the number of clicks when there is no added lag.

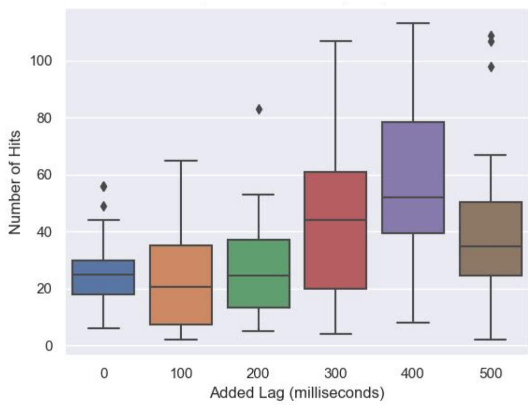


Figure 4.c.3 Square Dodger: Hit vs Added Lag
Boxplot: distribution of all participants

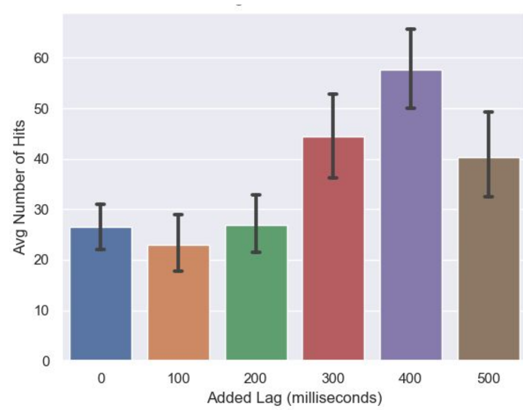


Figure 4.c.4 Square Dodger: Hit vs Added Lag
Bar chart: average hit counts with standard error

Figure 4.c.3 shows a boxplot of the distribution of users hit counts at six latency levels. The x-axis represents added lag in milliseconds; the y-axis represents the number of falling squares that hit the player. This graph shows an overall increasing trend which means users tend to be hit by more boxes with the increasing added lag.

Figure 4.c.4 shows a bar chart of the average number of hits for all participants at six latency levels with standard error bars. The x-axis represents added lag in milliseconds; the y-axis represents the average number of hits by the falling squares. The increasing trend of this graph matches Figure 4.c.3.

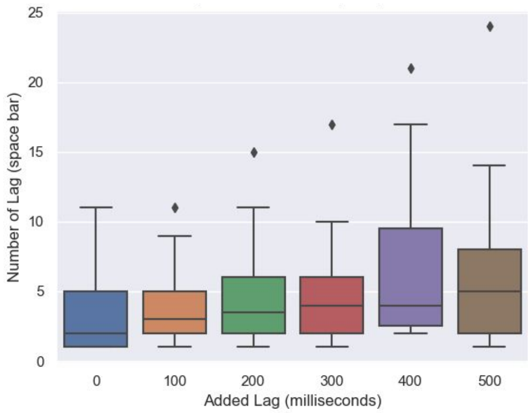


Figure 4.c.5 Square Dodger: Lag (spacebar) vs Added Lag
Boxplot: distribution of all participants

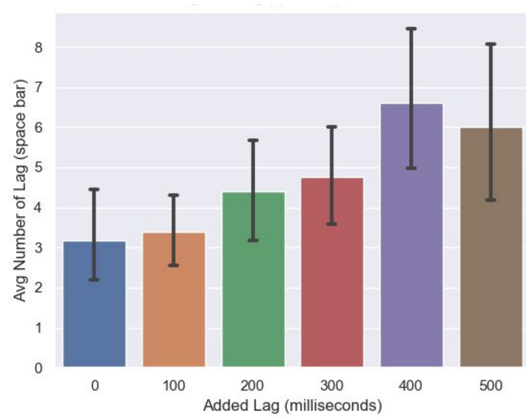


Figure 4.c.6 Square Dodger: Lag (spacebar) vs Added Lag
Bar chart: average spacebar counts with standard error

Figure 4.c.5 shows a boxplot of the distribution of users spacebar presses at six latency levels. The x-axis represents added lag in milliseconds; the y-axis represents the number of space bar presses. There are no apparent correlations between added lag and user self-report of lag.

Figure 4.c.6 shows a bar chart of the average number of users spacebar presses for all participants at six latency levels shown with standard error bars. The x-axis represents added lag in milliseconds; the y-axis represents the average number of spacebar presses. This graph shows a positive correlation between the added lag and the average number of self-reported lag. The standard error suggests that this data is not precise enough to extrapolate results.

4.3.3 Summary

The average number of clicks for both games are negatively correlated to added lag. *Sushi Shooter* has a linearly decaying correlation while *Square Dodger* has an exponentially decaying correlation. In *Sushi Shooter*, users' average number of hits on the target decreases as the added lag increases. In *Square Dodger*, users' average number of being hit by the falling squares increases as the added lag increases. Users have an increasing perception of lag in *Square Dodger* with an increase in added lag while there is no apparent trend between lag perception and added lag in *Sushi Shooter*. We can summarize that added lag negatively influence users performance in both games. The *Square Dodger* is less tolerant of the lag compares to *Sushi Shooter*.

4.4 Heart Rate Variability Data Analysis

A normal resting heart rate for adults ranges from 60 to 100 beats which correspond to 1000 to 600 ms HRV. A well-trained athlete can 40 heart beats per minute which is 1500 milliseconds HRV. We discarded all the HRV (Heart Rate Variability) higher than 1500 milliseconds which suggests the data to be inaccurate reduce inaccurately. Out of 36 data files, we discarded 3 files with a longer/shorter time duration which caused by human error during the experiment.

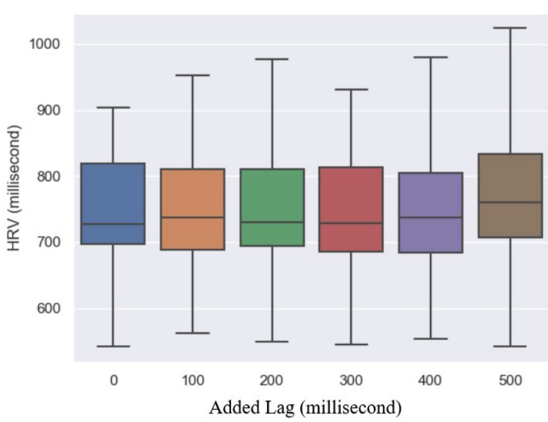


Figure 4.d.1 Sushi Shooter: HRV vs Added Lag
Boxplot: distribution of all participants

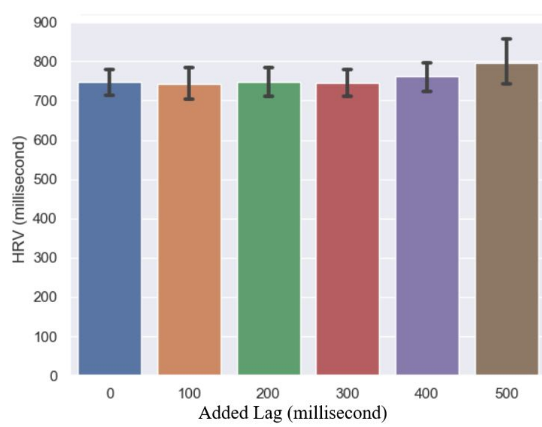


Figure 4.d.2 Sushi Shooter: HRV vs Added Lag
Bar chart: average HRV with standard error

Figure 4.d.1 shows a boxplot of the distribution of users HRV at six latency levels. The x-axis represents added lag in milliseconds; the y-axis represents HRV. This graph shows that the average HRV for each level of added lag is around 720 milliseconds. The average HRV is $\frac{60000\ milliseconds}{720\ milliseconds} \approx 83$ heart beats per minute which are in the range of adults resting heart rate.

Figure 4.d.2 is a bar chart of the average value of HRV for all participants at six latency levels shown with standard error bars. The x-axis represents added lag in milliseconds; the y-axis represents the average HRV for all participants. The HRV raised 50 milliseconds from 300 ms added lag to 500 ms added lag.

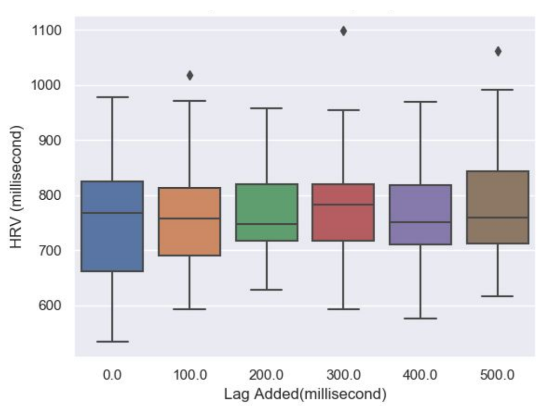


Figure 4.d.3 Square Shooter: HRV vs Added Lag
Boxplot: distribution of all participants

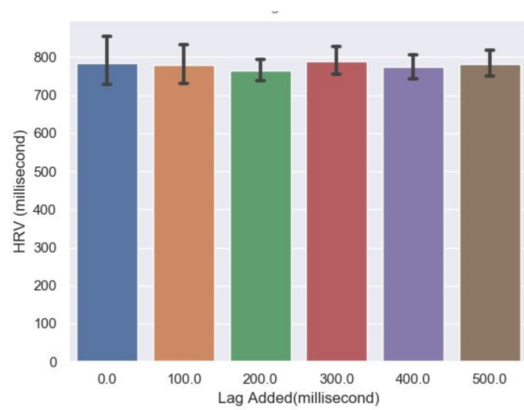


Figure 4.d.4 Square Shooter: HRV vs Added Lag
Bar chart: average HRV with standard error

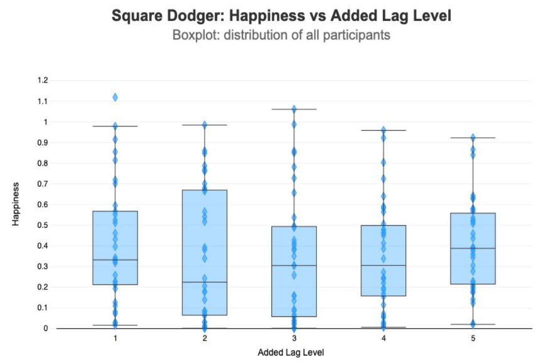
Figure 4.d.3 shows a boxplot of the distribution of users HRV at six latency levels. The x-axis represents added lag in milliseconds; the y-axis represents HRV. This graph shows that the average HRV for each level of added lag is around 760 milliseconds. The average HRV is $\frac{60000\ milliseconds}{760\ milliseconds} \approx 78$ heart beats per minute which are in the range of adults resting heart rate.

Figure 4.d.2 is a bar chart of the average value of HRV for all participants at six latency levels shown with standard error bars. The x-axis represents added lag in milliseconds; the y-axis represents the average HRV for all participants. On average, participants' HRV remain within the range of 760 to 790 millisecond during the whole game with no apparent trend with added lag.

4.5 Emotion Data Analysis

The Emotion Detection Asset provided data for six basic facial expressions: happiness, sadness, surprise, fear, disgust, and anger, which are all universal emotions defined by modern psychology. It also provided data for Neutral emotion. For our experiments, the data from Fear and Sadness emotions were near zero, so we did not analyze them further. We also did not analyze Natural emotion. The emotion data are limited within the range of 0-2, but most do not exceed 1. A number that is close to one indicates the intense emotion of that type. The emotion data sets were divided into five lag levels from 1-5. Because there are different intervals between the time we opened and closed the software and the time participants started and ended playing, we can not determine the exact start and end times. So, we analyzed the data set by five lag levels, where each level has a higher lag in general.

4.5.1 Happiness vs Added Lag



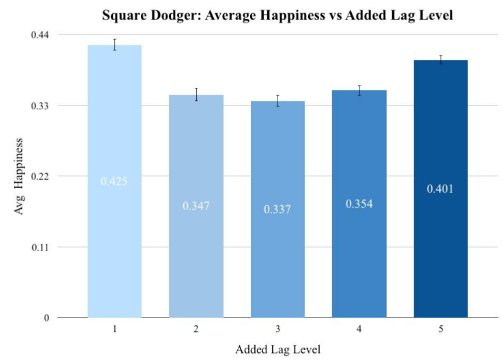


Figure 4.5.1.1a: Square Dodger: Happiness vs Added Lag Level
Boxplot: distribution of all participants

Figure 4.5.1.1b: Square Dodger: Avg Happy vs Added Lag Level
Bar chart: average happiness over five increasing lag periods

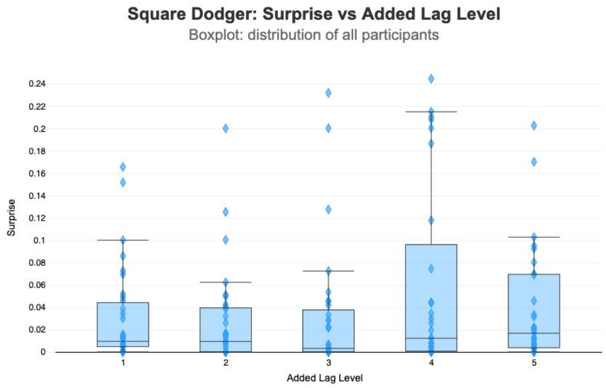
Figure 4.5.1.1a shows a boxplot of the distribution of all participants' happiness versus lag level for the *Square Dodger* game. The x-axis represents the added lag level; the y-axis represents the happiness of all participants. Figure 4.5.1.1b shows a bar chart of the average happiness with standard error bars from all participants versus lag level for the *Square Dodger* game. The x-axis represents the added lag level; the y-axis represents the average happiness. There is a slight negative correlation between added lag and participants' happiness. Note that the participants' happiness increased for the last lag period, which might be explained by their feedback, indicating they had never played a game with that much lag before, and it was amusing when they missed the shoot or got hit.

Figure 4.5.1.2a: Sushi Shooter: Happiness vs Added Lag Level
Boxplot: distribution of all participants

Figure 4.5.1.2b: Sushi Shooter: Avg Happiness vs Added Lag Level
Bar chart: average happiness over five increasing lag periods

Figure 4.5.1.2a shows a boxplot of distribution of all participants' happiness versus lag level for the *Sushi Shooter* game. The x-axis represents the added lag level; the y-axis represents the happiness of all participants. Figure 4.5.1.2b shows a bar chart of the average happiness with standard error bars from all participants versus lag level for the *Sushi Shooter* game. The x-axis represents the added lag level; the y-axis represents the average happiness. Similar to *Square dodger*, the participants' happiness decreased when the added lag increased, and there is also increased happiness in the last lag period. Overall participants have slightly higher happiness in *Square Dodger* than in *Sushi Shooter*.

4.5.2 Surprise vs Added Lag



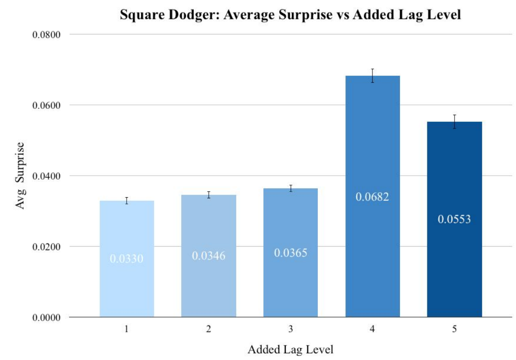


Figure 4.5.2.1a: Square Dodger: Surprise vs Added Lag Level
Boxplot: distribution of all participants

Figure 4.5.2.1b: Square Dodger: Avg Surprise vs Added Lag Level
Bar chart: average surprises over five increasing lag periods

Figure 4.5.2.1a shows a boxplot of distribution of all participants' surprise versus lag level for the *Square Dodger* game. The x-axis represents the added lag level; the y-axis represents the surprise of all participants. Figure 4.5.2.1b shows a bar chart of the average surprise with standard error bars from all participants versus lag level for the *Square Dodger* game. The x-axis represents the added lag level; the y-axis represents the average surprise. The surprise emotion is indicated when the eyebrows are raised, the eyes are wide open, and the mouth is opened. On average, participants' surprise sharply increased during the fourth lag period, which is also the period when they pressed the space bar more and when they were hit the most.

Figure 4.5.2.2a: Sushi Shooter: Surprise vs Added Lag Level
Boxplot: distribution of all participants

Figure 4.5.2.2b: Sushi Shooter: Avg Surprise vs Added Lag Level
Bar chart: average surprises over five increasing lag periods

Figure 4.5.2.2a shows a boxplot of distribution of all participants' surprise versus lag level for the Sushi Shooter game. The x-axis represents the added lag level; the y-axis represents the surprise of all participants. Figure 4.5.2.2b shows a bar chart of the average surprise with standard error bars from all participants versus lag level for the Sushi Shooter game. The x-axis represents the added lag level; the y-axis represents the average surprise. On average, participants' surprise is higher in the first, third and fifth lag period, and sharply decreased in the second and fourth lag period.

4.5.3 Disgust vs Added Lag

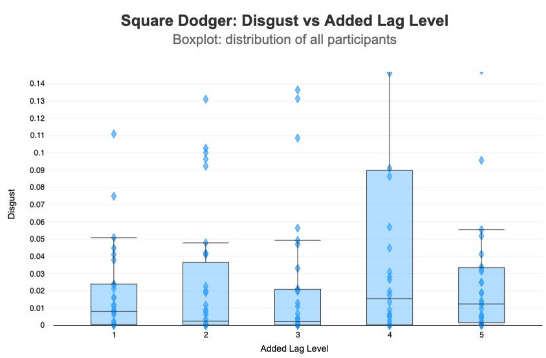


Figure 4.5.3.1a: Square Dodger: Disgust vs Added Lag Level
Boxplot: distribution of all participants

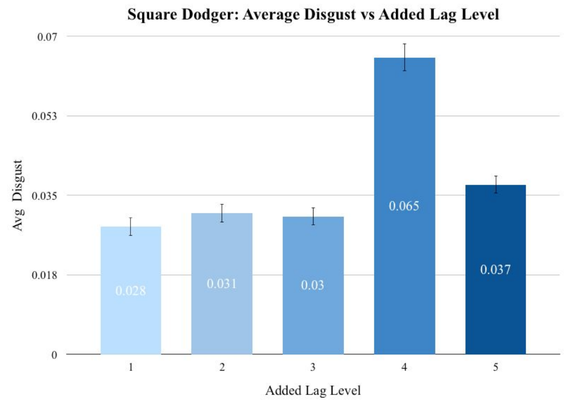


Figure 4.5.3.1b: Square Dodger: Avg Disgust vs Added Lag Level
Bar chart: average surprises over five increasing lag periods

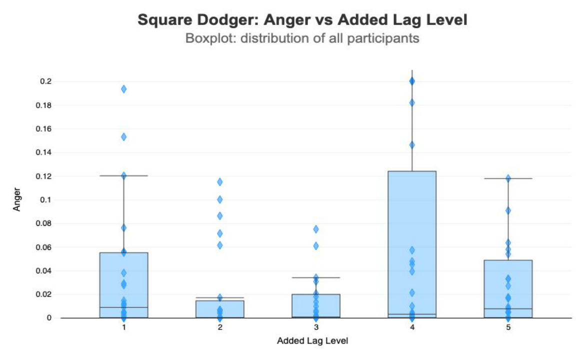
Figure 4.5.3.1a shows a boxplot of distribution of all participants' disgust versus lag level for the *Square Dodger* game. The x-axis represents the added lag level; the y-axis represents the disgust of all participants. Figure 4.5.3.1b shows a bar chart of the average disgust with standard error bars from all participants versus lag level for the *Square Dodger* game. The x-axis represents the added lag level; the y-axis represents the average disgust. Similar to surprise emotion, participants' average disgust sharply increased during the fourth lag period, which is the period when they pressed the space bar more and when they were hit the most.

Figure 4.5.3.2a: Sushi Shooter: Disgust vs Added Lag Level
Boxplot: distribution of all participants

Figure 4.5.3.2b: Sushi Shooter: Avg Disgust vs Added Lag Level
Bar chart: average disgust over five increasing lag periods

Figure 4.5.3.2a shows a boxplot of distribution of all participants' disgust versus lag level for the *Sushi Shooter* game. The x-axis represents the added lag level; the y-axis represents the disgust of all participants. Figure 4.5.3.2b shows a bar chart of the average disgust with standard error bars from all participants versus lag level for the *Sushi Shooter* game. The x-axis represents the added lag level; the y-axis represents the average disgust. On average, participants' disgust sharply increased on the fifth lag period, which is when the lag is most added. This plot shows a positive correlation between participants' disgust and the added lag.

4.5.4 Anger vs Added Lag



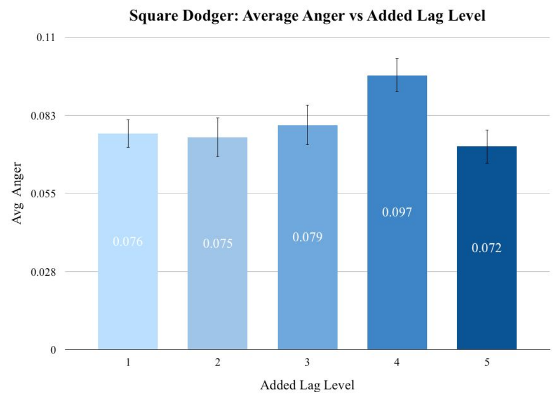


Figure 4.5.4.1a: Square Dodger: Anger vs Added Lag Level
Boxplot: distribution of all participants

Figure 4.5.4.1b: Square Dodger: Avg Anger vs Added Lag Level
Bar chart for average anger over five increasing lag periods

Figure 4.5.4.1a shows a boxplot of distribution of all participants' anger versus lag level for the *Square Dodger* game. The x-axis represents the added lag level; the y-axis represents the anger of all participants. Figure 4.5.4.1b shows a bar chart of the average anger with standard error bars from all participants versus lag level for the *Square Dodger* game. The x-axis represents the added lag level; the y-axis represents the average anger. On average, there is a positive correlation between added lag and participants' anger; anger increased when the added lag increased.

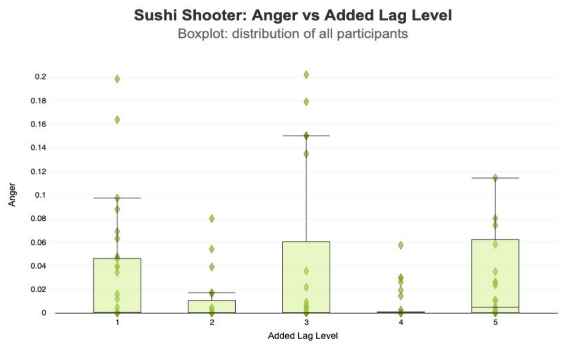


Figure 4.5.4.2a: Sushi Shooter: Anger vs Added Lag Level
Boxplot: distribution of all participants

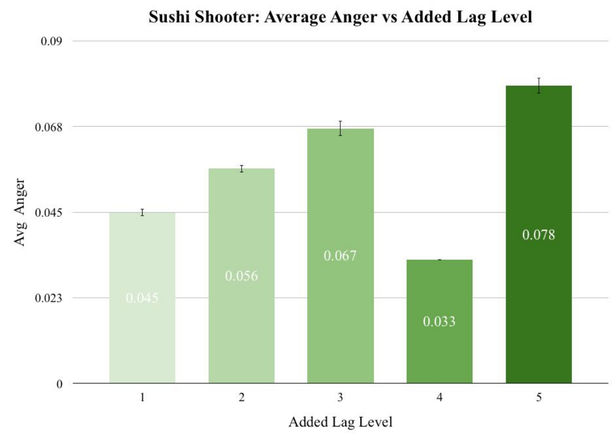


Figure 4.5.4.2b: Sushi Shooter: Avg Anger vs Added Lag Level
Bar chart for average anger over five increasing lag periods

Figure 4.5.4.2a shows a boxplot of distribution of all participants' anger versus lag level for the *Sushi Shooter* game. The x-axis represents the added lag level; the y-axis represents the anger of all participants. Figure 4.5.4.2b shows a bar chart of the average anger with standard error bars from all participants versus lag level for the *Sushi Shooter* game. The x-axis represents the added lag level; the y-axis represents the average anger. Similar to the *Square Dodger*, on average participants' anger increased over the five increasing lag periods, which shows a positive correlation between participants' anger and the added lag.

Section 5 Conclusions and Future Study

Video games have become increasingly popular over the past several decades, from the earliest computer game *Bertie the Brain* (a computer game of tic-tac-toe, Josef Kates 2019) to nowadays' massively multiplayer online battle arena video game like *League of Legends* (Riot Games, 2009), over sixty thousand video games have been released since 1950 to meet player demand [20]. With Internet growth, games are increasingly networked. Unfortunately, network latency can cause a delay of information between the client and the server and greatly affects players' gameplay and may cause them to quit playing. It is important for game developers to understand the effects of network latency on player gaming experience so that they can find better delay compensation techniques during game development. Instead of studying the effects of delay for a whole game system where there can be many confounding factors, our project focuses on studying the effects of latency on the most fundamental game actions, movement and shooting.

We designed two video games for movement and shooting actions separately: *Sushi Shooter* is for the shooting action and *Square Dodger* is for the movement action. Both games have 100 milliseconds of lag added every 30 seconds, for 200 seconds of play time We measured the players' gaming experience along three independent variables: performance, emotion, and heart rate. During 36-person user study, participants were asked to play *Sushi Shooter* and then *Square Dodger* while the emotion detection software and heart rate sensor gathered data. During play, participants were asked to press the spacebar when they felt lag.

Our result showed that for both games, players tend to click less as added latency increases and they tend to miss the target in *Sushi Shooter* more and get hit by the falling squares more in the *Square Dodger*. In other words, the players' performance is negatively affected as latency increases. Our results also show a positive correlation between the added latency and the average lag perception in *Square Dodger*, which indicates that players are aware of the increased latency. The emotion detection data shows that players' average happiness level decreases as latency increases, and players' negative emotion, which includes anger and disgust, increase with the added latency. This suggests that players' emotion is negatively affected by network latency during gameplay, which also indicates the increase in players' stress level. Our study does not show a direct correlation between players' heart rate and added latency as we expected.

For future study, researchers could investigate further into the movement and shooting actions with the two games we currently have. We did the experiment with 36 participants in our study. A more comprehensive study with a larger number of data samples could provide more statistical relevance. Future work could also design different games with different fundamental actions other than shooting and dodging to test how latency affects players' performance on other actions. Emotion detection during gameplay is also a possible area to explore, and researchers could develop software for more accurate emotion collection, as well as study more specific emotions beyond happiness, surprise, anger, and disgust. Although we did not find a direct correlation between participants' heart rate and added latency, it also might be worth exploring further. Perhaps by using more exciting games that have more potential to raise heart rate.

References

[1] Mccarthy J, Sasse M, Miras D. Sharp or smooth?: Comparing the effects of quantization vs. frame rate for streamed video. 2004:535-542. DOI: 10.1145/985692.985760.

[2] Claypool K, Claypool M. On frame rate and player performance in first person shooter games. *Multimedia Systems*. 2007;13(1):3-17. DOI: 10.1007/s00530-007-0081-1.

[3] Sheldon N, Girard E, Borg S, Claypool M, Agu E. The effect of latency on user performance in Warcraft III. 2003:3-14. DOI: 10.1145/963900.963901.

[4] Maymin P. An open-sourced optical tracking and advanced eSports analytics platform for League of Legends. In *2018 MIT Sloan Sports Analytics Conference* (Vol. 18). Boston, MA, US, 2018. Online at: http://www.sloansportsconference.com/wp-content/uploads/2018/02/1002.pdf

[5] Teng C.I, Jeng S.P, Chang H.K.C, Wu S. (2012). Who plays games online?: The relationship between gamer personality and online game use. *International Journal of E-Business Research (IJEBR)*, *8*(4), 1-14. DOI: 10.4018/jebr.2012100101.

[6] Amin R, Jackson F, Gilbert J.E, Martin J, Shaw T. Assessing the impact of latency and jitter on the perceived quality of Call of Duty Modern Warfare 2. HCI 2013. DOI:10.1007/978-3-642-39265-8_11

[7] Bontchev B, Vassileva D. Assessing engagement in an emotionally-adaptive applied game. 2016;02-04-:747-754. DOI: 10.1145/3012430.3012602.

[8] Claypool M, Eg R, Raaen K. The effects of delay on game actions: Moving target selection with a mouse. 2016:117-123. DOI: 10.1145/2968120.2987743.

[9] Drachen A, Nacke L, Yannakakis G, Pedersen A. Correlation between heart rate, electrodermal activity and player experience in first-person shooter games. 2010:49-54. DOI: 10.1145/1836135.1836143.

[10] Kuipers F, Kooij R, De Vleeschauwer D, Brunnström K. Techniques for measuring quality of experience. In *International Conference on Wired/Wireless Internet Communications* (pp. 216-227). Springer, Berlin, Heidelberg, 2010. Online at: https://link.springer.com/chapter/10.1007/978-3-642-13315-2_18

[11] Brunnström K, Beker S.A, De Moor K, Dooms A, Egger S, Garcia M.N, Lawlor B. 2013. Qualinet white paper on definitions of quality of experience. Online at: https://hal.archives-ouvertes.fr/hal-00977812/document

[12] Koumaras H, Kourtis A, Martakos D, Lauterjung J. Quantified PQoS assessment based on fast estimation of the spatial and temporal activity level. *Multimed Tools Appl*. 2007;34(3):355-374. DOI: 10.1007/s11042-007-0111-1.

[13] Cronin E, Filstrup B, Kurc A. 2001. A distributed multiplayer game server system. In the *University of Michigan*. Online at: https://www.semanticscholar.org/paper/A-Distributed-Multiplayer-Game-Server-System-Kurc/72e12a6b228f385619479c5058e93b2e4ad47dd0

[14] Wall Street Journal. D8 Video: OnLive demoed on iPad, PC, Mac, Console, iPhone [Video file]. 2010-08-09. Retrieved from http://video.allthingsd.com/video/d8-video-onlive-demo/9D57A2C6-24ED-4351-8266-F3F7BA0C4D18/.

[15] Quandt T, Kröger S. London: Routledge. 2014. Multiplayer: The social aspects of digital gaming. ISBN 978-0415828864.

[16] Buncher M.W. The effects of video game difficulty selection on flow experience. 2013. ETD Archive. 374. Online at: https://engagedscholarship.csuohio.edu/etdarchive/374

[17] League of Legends • skill shot speeds • /u/thatdankworkaccounts. Reddit. Web. 25 July. 2017. Online at: https://www.reddit.com/r/summonerschool/comments/6pjtpv/champion_skillshot_speeds_and_ranges_list/

[18] Champions Stats - League of Legends Movement Speed. LoL Champions & Summoners Stats & Rankings - LeagueGraphs. Web. 05 Feb. 2017. Online at: https://leagueoflegends.fandom.com/wiki/Movement_speed

[19] Farnsworth B. Heart rate variability – how to analyze ECG data. IMOTIONS. Accessed Online 1/19/2019. Online at: https://imotions.com/blog/heart-rate-variability/

[20] How many video games exist?. David Mullich. Quora. Web. 15 May. 2017. Online at: https://www.quora.com/How-many-video-games-exist.

[21] Deshmukh R.S, Jagtap V, Paygude S. Facial emotion recognition system through machine learning approach. *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 272-277, Madurai, Indian, 2017. DOI: 10.1109/ICCONS.2017.8250725.