The Effects of Latency and Game Device on Moving Target Selection

By Aritra Kundu

A Thesis
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirements for the Degree
of Master of Science
in
Computer Science

May 2021

Primary Advisor:

Prof. Mark Claypool, Professor, Worcester Polytechnic Institute.

Co Advisor:

Prof. Ragnhild Eg, Associate Professor, Kristiania University College.

Reader:

Prof. Charles Roberts, Assistant Professor, Worcester Polytechnic Institute.

APPROVED

Professor Mark Claypool: ______________________________

Professor Charles Roberts: ______________________________

Professor Craig Wills: ______________________________

Acknowledgements

I would like to thank Professor Claypool and Professor Ragnhild for their continued support for this work.

Contents

Table Of Figures

Table Of Tables

Abstract

With the increasing variety of gaming devices, the way people can play computer games has expanded. While computer gaming often involves interacting with a computer with a keyboard and mouse or via a touchscreen with a human finger, more recent gaming can be done in immersive virtual reality (VR) environments with a controller. For online games, these platforms have network latency in addition to local latency added by the system hardware and software. Since most online computer games provide several player actions and have latency compensation techniques built into the system, it is difficult to assess the effects of latency. The goal of this thesis is to compare the effects of latency on three different gaming devices with different inputs: a PC with a mouse, a tablet with a finger and a VR device with a controller. We created a cross platform application for the isolated action of moving target selection with controlled amounts of input delay and conducted a user study. Based on our analysis from 30 participants, we found delays and task difficulties affect the performance as well as the quality of experience of the users. The PC with mouse has the highest number of clicks and the longest selection times and the tablet has the lowest. The higher delays and higher task difficulties according to target size or speed have longer selection times compared to lower delays and task difficulties.

1. Introduction

Playing computer games is one of the most popular forms of entertainment today. The industry today is worth somewhere around 130 billion dollars [1]. Recent advances in technology have also changed the devices used to play computer games. Nearly three-quarters of U.S adults own desktop or laptop computers, while roughly half own tablet computers [2]. New technology such as virtual reality (VR) provides a new platform for gaming [3]. Devices such as an Oculus by Facebook [4] offer complete virtual reality (VR) experiences in a portable device with options for playing online games. There is a shift towards gaming on cloud services, such as Google Stadia or NVIDIA Geforce, where user input is sent to a cloud server, the server handles the game engine logic-such as applying physics to game objects, resolving collisions or processing animations and renders the game frames which are streamed as video to the game client.

One of the major challenges facing playing computer games over a network is the latency. Real-time games require players to make many time-sensitive actions that degrade when delayed and even delays as small as a few milliseconds can negatively affect the interplay between players' actions and intended results. For example, latency when aiming a scope in a first-person shooting game can make it difficult for a player to hit a moving target, decreasing the player's performance and degrading the quality of experience. In a cloud-based game, all user input is subject to network latency which decreases player performance. While most games a latency compensation technique built into their system such as dead reckoning, sticky targets or world alteration [5], many of these techniques are not effective for local latencies nor can all of them be used for cloud-based games.

Computer games played today are of a variety of genres and the effects of latency depends upon the game actions. Most gameplay require multiple actions from the user. For example, in a first-person shooter, the common actions are movement, targeting and shooting. Given this, an effective way to study the effects of latency is to focus on individual, isolated actions. Player actions can be categorized according to their precision or deadline requirements [6] with actions also parameterized based on difficulty [7]. For example, the difficulty of target selection can depend on size of the target as well as distance of the target [8].

Game platforms are proliferating with devices of different input methods being used for playing games including: a PC with a mouse, a tablet with a finger or a

VR device with a controller. Although moving target selection has been studied previously [9], to the best of our knowledge, a cross device comparison of moving target selection with latency across PC, Tablet and VR has not been done before.

We developed a cross device application using HTML, CSS and Javascript to study the effects of latency on a game action on a PC, a tablet and virtual reality (VR). The PC was chosen as it provides mouse and keyboard for input, the tablet provides touch input and VR is an emerging form interaction with a hand controller as input. In the application, users had to select targets of different sizes that move at various speeds. Users selected the targets by moving a cursor. The settings were identical for the devices except for the interaction mode. All input was delayed by fixed amounts and difficulty varied each round.

Based on our analysis from 30 participants, we found that the PC has the longest selection times, followed by VR and the tablet has the shortest selection times. PC has the highest clicks, followed by VR and tablet has least number of clicks. Users found the tablet to have the best interaction experience, ease of use and control and VR to be the most immersive device.

The rest of the thesis is organized as follows: Chapter 2 provides contextual information on areas related to our study, Chapter 3 gives a review of existing work in this area, Chapter 4 describes in detail various aspects of the devices used, the application that was developed and the user study, Chapter 5 provides an analysis of the data obtained from the participants of the user study, Chapter 6 discusses our research questions, and Chapters 7 and 8 summarize our conclusions and possible future work.

2. Background

This section provides background information on the topics related to our research: latency in games, game architectures and target selection.

2.1 Latency in Games

The latency experienced by players in online games is mainly from two sources: network latency and local latency.

Network latency in a game can be attributed to the network connection between the game client and the game server. Games often refer to network latency as the ping between the client and the server. This is the time taken for packets to travel from the client to the server and back again. Data can theoretically travel with a lower bound of the speed of light; it takes approximately a minimum of 25ms for data to cross a continent. One of the sources of network latency is from each of the switches and routers between the client and the server. Another source of latency is the time taken to transmit an IP packet on each individual link. A third source of latency is when data has to be queued by a congested router before it can be transmitted further.

Local latency is the time taken by the local system from when a user starts an action on a device to the point at which the effect of that action is visible. Local latency can be attributed to polling of input devices, the refresh rate of the output device or monitor and processing time by the CPU and GPU.

2.2 Game Architectures

In traditional multiplayer networked games, the game client can apply game logic such as computing physics, handling collisions or doing client-side prediction. The client sends a request to the server where the game action is processed and a response is sent back to the client after the game state is updated. In order to reduce the effects of latency in these architectures, prediction models can be used where the client locally responds to user input before acknowledgements are sent back from the server. The player actions are still subject to local latency at the client.

In cloud-based games the game clients are lightweight, only sending game input and outputting the game image or sounds they receive. All heavyweight game logic such as collision handling, artificial intelligence and rendering are done by the server with the game frames being streamed to the client. All game actions are subject to round-trip network latency in this model as well as local latency on the device the user is interacting with.

2.3 Target Selection

The fundamental isolated action of selecting a moving target with a pointing device is common for computer games. Figure 1 shows a player interacting in a First-Person Shooter (FPS) where a number of gaming actions are possible such as changing the location of the character or aiming and shooting at a target. Fast moving target selection is a challenging task and one that is frequently required of FPS players. Traditional target selection studies and resulting models such as Fitts' law [8] focus on stationary targets and do not take effects such as latency into consideration.

Figure 1. A User Selecting a Moving Character in an FPS Game [10]

In Figure 1, in order to shoot the opponent that is moving, the player would first have to place the reticle on the moving character and select it.

We study the effects of latency across devices where all actions are delayed such as in a cloud-based architecture for the task of moving target selection.

3. Related Work

This section describes work on topics related to our research: latency studies, target selection and task difficulty.

3.1 Latency Studies

All real-time computer games are subject to a latency from the time the player starts an action on an input device such as a computer mouse or controller, to the time the effect of that action is displayed on the screen. In a typical client server architecture, a server processes a request from the client and sends a response back, after which the result is rendered on the client.

Quax et al. [11] studied the effects of latency on a First Person Shooter, Unreal Tournament, and have provided both a subjective and objective evaluation of player performance. Claypool et al. [6] found the effects of latency depends upon the game action.

Xu et al. [12] studied the effects of latency on Stadia, a cloud gaming service by Google. Liu et al. [13] model the effects of latency as in a cloud game where every action is delayed.

With recent advances in mobile processing capabilities, it is possible for users to interact with 360 degrees virtual reality experiences on mobile devices [14]. Existing studies have focused on latency studies using a thumbstick [15] and there has also been studies for understanding the effects of latency in VR [16].

Claypool et al. [17] found that actions that have a higher precision and tighter deadline requirement result in a lower score for the player experiencing delay than actions that have a lower precision or looser deadline requirements. They suggest that the genre of a game based on precision and deadline can affect player performance.

Bernier et al. [18] observed that most networked games have latency compensation techniques that help offset the effect latency can have on player performance in an online game.

Raaen et al. [16] introduced an exploratory setup for an empirical investigation of the effects of latency in a custom-built VR game. They provided a platform for future studies on the constraints and consequences of delayed VR interaction.

3.2 Target Selection

Claypool et al. [15] studied selection of a target with constant velocity and the effects of latency on such a user action. An analytical model was developed for calculating time to select a target based on latency, speed as well as combined interaction terms from data collected from a user study. The study compared data using a thumbstick as a controller with a keyboard and mouse setup from a previous study.

Claypool et al. [9] modelled target selection by developing a game with latency along with added motion parameters of turn angle, time between target turns as well as force-based motion. One of the difficulties noted was that it is hard to compare user studies with different interaction techniques as well as the different users view of the game world which makes cross comparison of data difficult.

Claypool et al. [19] identified several objective parameters such as latency, target size and type of motion for characterizing similar user studies and do a cross comparison of data from similar studies. They noted a linear increase in reaction times over a range of delays. They found both the distance between the mouse pointer and moving target impact performance, and the time it takes the user to select the moving target increases linearly over the range of delays tested. They also found a sharp increase in selection time for higher delays. Such analysis can provide useful information to game developers for mitigating the effects of both local and network latency on a device.

Gutwin et al. [20] built a custom game that emulated classic pointing and interception tasks. The game was tested on users using a touchscreen, mouse, gamepad and drawing tablet. They found that local latency affects games in particular, among other computer applications, and noted the difficulty of studying games because of diversity in genre, task and input devices. They also found that latency affects each device differently with latency affecting touchscreens the least.

For target selection with pointing devices, Teather et al. [21] studied the affects of network latency and jitter for 2 dimensional as well as 3 dimensional object movement tasks. They found latency to have a strong affect on performance.

MacKenzie et al. [22] found that when a latency of 225 ms was introduced in a target acquisition task using the Fitts' law, player performance is reduced drastically. They introduced a model where latency is incorporated into the prediction.

Friston et al. [23] constructed a system with local latency of 1 ms and display latency of 5 ms. They concluded that higher latencies may result in lower movement times on processing and steering tasks due to the degraded effects of latency on the processes of the motor system.

Ivkovic et al. [24] studied local latency on third-person shooter games as latency affects different genres of games differently. They found that latencies as low as 41 ms can have a negative affect on third-person shooter games.

Claypool, Eg and Raaen [25] found an upward trend in response times with increasing latencies and they also found that the data appears exponential in nature. Long and Gutwin [20] used path metrics to analyze data and find a correlation between movement error and player performance. Pavlovych et al. [26] used Lissajous curves to model target selection. Claypool, Cockburn and Gutwin [9] used an exponential model based on latency and the speed of the object.

3.3 Task Difficulty

Pioneering work done by Fitts' [8] models stationary target selection, where difficulty of the task or ID Score mathematically depends on the width of the target (W) as well as the distance to the target (D). The Index of Performance also known as throughput depends on ID score and movement time (MT).

$$ID\ Score = \log_2(2D/W)$$

$$IP\ Score = (ID\ Score/MT)$$

$$MT = a + b * ID\ Score$$

where a and b are constants

The model tells us that as difficulty of a task increases, the time to select a target increases. The throughput or average rate of information depends on ID Score and movement time. The movement depends on the ID Score. This model has been modified by researchers to incorporate factors such as latency or speed into the model [8] and newer models have been proposed.

An important component of our study is the difficulty of the task of target selection. Claypool et al. [15] take latency into consideration to provide better models of task difficulty in the appropriate setting. Jiang et al. [7] studied reaction times according to self-rated skill with respect to decision complexity and task dexterity for a PC.

To the best of our knowledge, an experiment to compare the effects of network latency for an isolated game action across different types of devices and evaluating the results according to difficulty of the task has not been done. Evaluating target selection across devices while taking the effects of latency and task difficulty is the objective of this thesis. We modify speed of the target as well as target size to understand the effects of task difficulty in addition to latency across the devices.

4. Methodology

In order to study game player performance for target selection with latency and different devices, we created a cross platform application and conducted a user study.

4.1 Research Questions

The research questions that motivated this study are:

- RQ1-How does player performance differ according to device?
- RQ2-How does player performance differ according to latency?
- RQ3-How does player performance differ according to task difficulty?
- RQ4-What are the differences in device/interaction mode on the effect of latency on performance?
- RQ5-What is the user experience according to latency, task difficulty and device?

4.2 Devices

The devices chosen for this study were a PC, a tablet and a VR device. The tablet and PC were chosen for this study as they provide different yet commonly used methods of interacting with a computer, which is human touch for a tablet and the mouse for a traditional PC. The VR device was chosen as it is growing in use and provides a interaction via a controller.

4.2.1 PC

For measuring player performance on a PC we used a Lenovo Ideapad 510 shown in Figure 2. The laptop runs Windows 10 with an Intel Core i7 processor with 2 cores with clock speed of 2.5 GHz, 20 Gigabytes of RAM and a NVIDIA GeForce 940 mx graphics card. The application was run on Google Chrome version 89.0.4389. The input device was a built in keyboard and external mouse with 1000 DPI and a polling rate of 125 Hz. The screen resolution was 1920 *1080 pixels with a 15.6 inch screen with a refresh rate of 60Hz. The browser window was resized to match the screen size of the tablet.

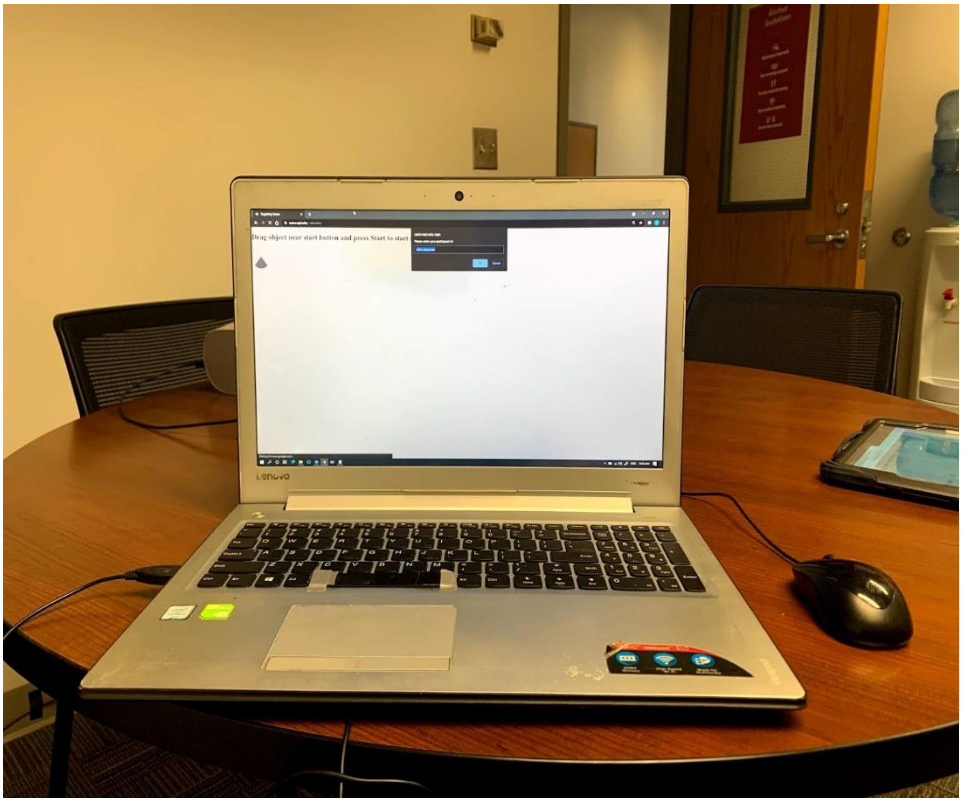

Figure 2. Laptop Used in the Study

4.2.2 Tablet

For measuring player performance on a tablet we used a Samsung Galaxy Tab S3 shown in Figure 3. The tablet runs Android with a Qualcomm Snapdragon 810 processor with a quad core 2.15 GHz processor and 4 Gigabytes of RAM. The application was run on Google Chrome version 90.0.4430.82 and the pointing device was the human finger. For providing the QoE input, the proprietary virtual keyboard was used. The screen resolution was 1024 * 768 pixels with a 9.7 inch screen with a refresh rate of up to 120Hz that was used in landscape mode.

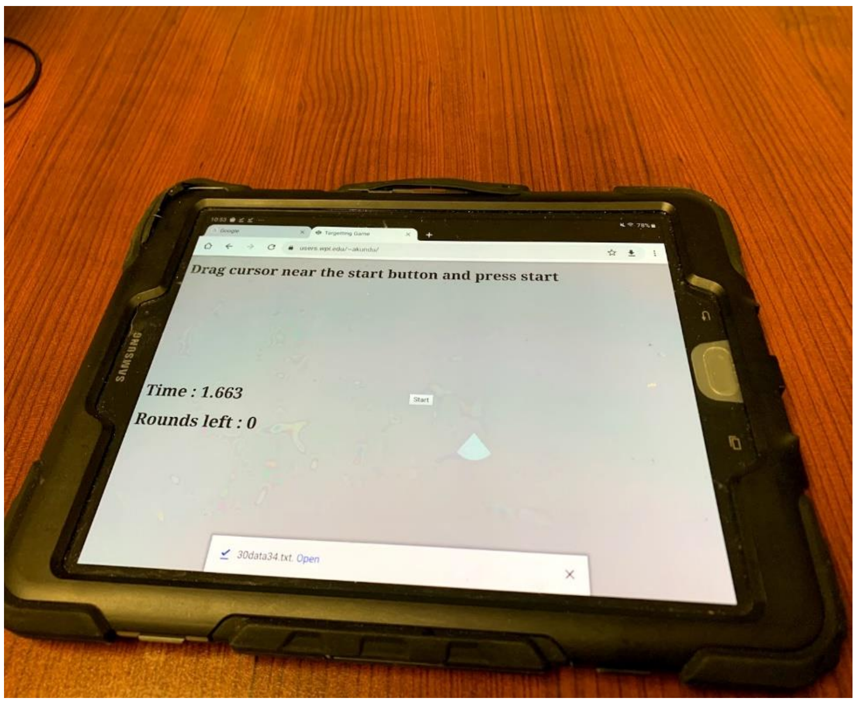

Figure 3. Tablet Used in the Study

4.2.3 Virtual Reality

For measuring player performance in a VR environment we used an Oculus Go shown in Figure 4. The Oculus runs Android with a Qualcomm Snapdragon 821 processor with a quad core 2.15GHz processor and 3 Gigabytes of RAM. It is a standalone virtual reality headset developed by Facebook in partnership with Qaulcomm and Xiaomi [27]. The device has a controller connected via Bluetooth. The controller has a trigger button which is similar to the click event on a traditional computer and for providing the QoE input the proprietary virtual keyboard was used. The screen has a resolution of 2560 * 1460 pixels and a refresh rate of 60Hz. The VR browser was resized to that of the tablet.

Figure 4. Virtual Reality Device Used in the Study

4.3 Measuring Local Latency

In online games there are two primary sources of latency that players experience: network latency and local latency.

For measuring the local latency on the three devices, recordings were taken using a high speed Samsung Galaxy phone camera at 245 frames per second and was analyzed with 5 trials per device. Each trial was of a participant performing a click action and recorded for a period of 2 seconds (refer to Figure 4). The time was measured from the moment the participant started the click action to when the pointer started to change color in the browser. The frames were counted to determine local latency.

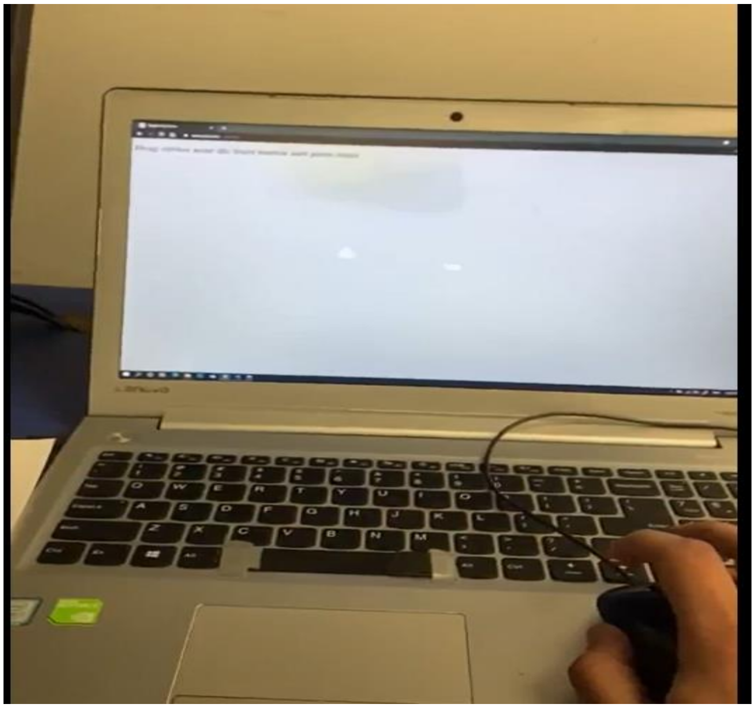

Figure 5. Measurement of Local Latency

The results of the trials are shown in Table 1.

Table 1. Local Latencies

Round	1	2	3	4	5	Average
PC	65ms	69ms	86ms	52ms	74ms	69ms
VR	28ms	28ms	37ms	41ms	32ms	33ms
Tablet	37ms	33ms	33ms	29ms	32ms	33ms

4.4 Game Design

This section describes the game that was built. Overall there is a cursor with which the user is required to select a moving target as fast as possible after pressing the start button at the center of the screen. Difficulty is controlled by the latency, the size of the target and the speed of the target.

4.4.1 Pointer

In order to maintain cross platform compatibility we created an application using HTML, CSS and Javascript libraries. We used the InteractJS library [28] for creating an object to represent a pointer in the game that can be dragged using both touch gestures as well as mouse movement for dragging and selecting the target object under different conditions. The CSS style of the object was modified to represent a triangle inside a semi-circle so the center point of the object which is used for calculating the hit condition is clearly visible. The user has to drag this pointer to the center of the screen at the start of every round before selecting the target.

Figure 6. Pointer Without Clicking and Figure 7. Pointer Object When Clicked.

The pointer shown in Figure 6 and Figure 7 changes colors whenever it is clicked so the participant gets feedback as to when a click is actually registered under different conditions of latency. Both click and drag movements of the object were delayed equally.

4.4.2 Target

The target object is a red circle of varying sizes. The JQuery animation library was used to make the target move at a constant velocity.

In order to ensure that the target is moving at a constant velocity, the speed of the first animation is stored in a variable and that same speed is used to calculate the time the successive animations on the screen takes using the formula:

$$Speed = \frac{Distance}{Time} \tag{2}$$

In order to ensure that the target does not speed up or slow down before a collision on the edges of the screen the linear easing option of the animate function was used as shown in Figure 8.

```
$(myclass).animate({ top: newq[0], left: newq[1]},{
          duration: Number(time),
          easing: "linear",
          step: function(currentLeft){
            document.getElementById("myHeader").innerHTML=parseInt($(this).css('left'));
            document.getElementById("myHeader1").innerHTML=parseInt($(this).css('top'));

          },
```

Figure 8. Sample Code for Animation

Three different velocity speeds were chosen to vary the difficulty of the task of moving target selection based on preliminary trials.

The target spawns at a random location on the screen and then moves with constant velocity in a straight line. When the target collides with any of the edges of the screen it bounces in a direction at an approximately equal angle with the assumption of physics with a frictionless surface. This concept is illustrated in Figure 9.

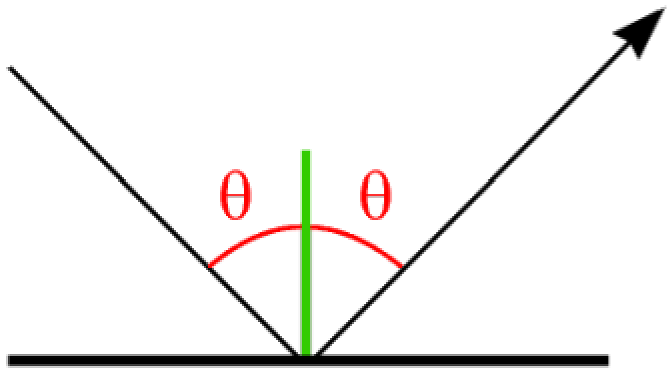

Figure 9. The Physics of the Bounce [29]

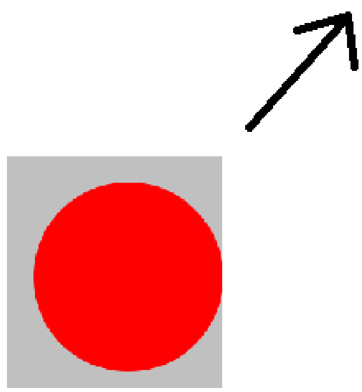

Figure 10. The Target to be Selected by the User

The target is one of 3 sizes on the screen. The diameters are of 50, 60 or 70 pixels. These values were chosen based on trial studies. When the pointer is inside the corresponding diameter in red and there is an onmouseup or ontouchend event then it is considered as a successful hit and the user has selected the target successfully.

4.4.3 Latency

In order to simulate the effects of latency on the cursor, the setTimeout() function in Javascript was used to delay the move event of InteractJS for delaying dragging of the cursor as well as for the mouseup(), mousedown() functions for delaying clicking. For the touch screen, the ontouchstart() or ontouchend() functions were used for delaying clicking on the tablet. The movement of the local machine mouse pointer or VR pointer was not delayed. The function is used to delay the execution of a function by a certain amount of time. This is show in Figure 11.

```
interact(".item").
draggable({
  //InteractJS CODE FOR DRAGGING
  manualStart: true,
  listeners: {
    move(event){
      setTimeout(function(){
      position.x += event.dx;
      position.y += event.dy;
     event.target.style.transform = `translate(${position.x}px, ${position.y}px)`;
```

Figure 11 Sample Code Illustrating the setTimeout Function to Delay Drag

One of the limitations of using the setTimeout() function is that it is subject to variation in the intended delay. The intended delay is the minimum time to execute, but the actual time can be late. This delay was recorded using the difference in time from when an event is added to the queue and when the event is expected to be executed by using the getTime() function. Figure 12 analyzes the time taken for the function to execute a command after delay. The x axis is the time taken in milliseconds and the y axis is the distribution. The statistics of the delay is shown in Table 2. At the time we took these measurements the Tablet device was not available for this study.

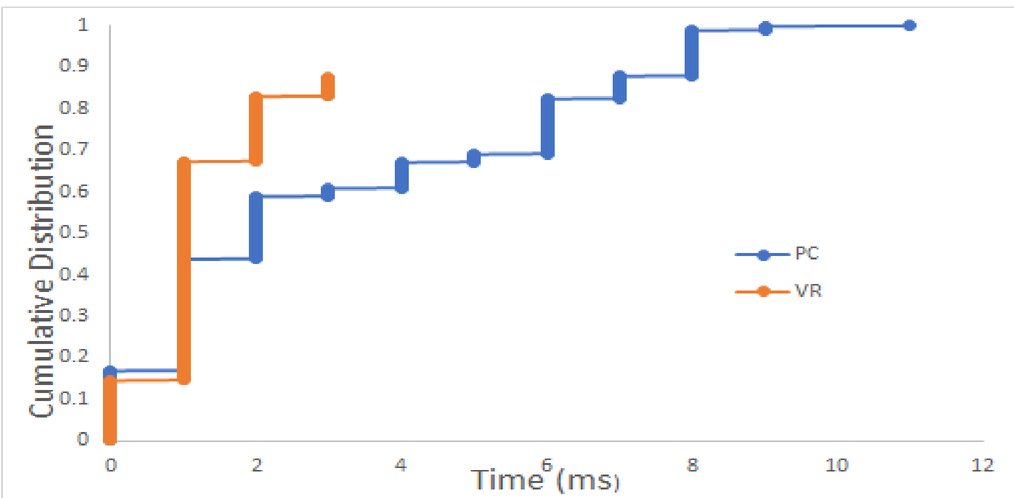

Figure 12. CDF of Extra Delay of setTimeout Function Grouped by Device.

Table 2. Statistics of Extra Delay in setTimeout Function

Device	Mean	Standard Deviation	Minimum	Maximum
PC	3.15 ms	2.85 ms	0 ms	11 ms
VR	1.16 ms	0.73 ms	0 ms	3 ms

4.4.4 Game Rounds

In order to test the user under varying conditions of speed and size, the user was made to perform the task twice under all combinations of low, medium and high settings as shown in Table 2. The rounds were in a random order. The first two rounds were practice rounds and were discarded during analysis. The user was asked a quality of experience question once for each setting. Each round finished with either the user successfully selecting the target or 30 seconds of gameplay, whichever was first.

The round information was stored in a SessionStorage variable so that if the user were to accidentally hit the refresh button or switch to a different application during gameplay the state information would still be stored in a persistent manner.

The settings were shuffled using a random sort so every game session has a different setting which is unknown to the user.

Table 3. Table of Settings

	Low	Medium	High
Added Latency	0ms	75ms	150ms
Total Latency Experienced	69ms	144ms	219ms
Size	50px	60px	70px
Speed	0 px/second	400 px/second	500 px/second

4.4.5 Data

Data was collected for the pointer object in JSON format. JSON was chosen as it is a universal format for storing front end data and its ease of use in terms of human readability of the data being stored.

The pointer object is shown in Figure 13. The data stored contains information such as x and y co-ordinates in pixels (xcord and ycord respectively), the timestamp of the record in seconds from when the round was started by pressing the start button, click information such as mousedown or mouseup events, as well as whether the object has an overlap condition with the target object.

```
var movedata ={
    obj: "g",
    xcord: x1,
    ycord: x2,
    timestamp: temptime,
    mousedown:0,
    mouseup:1,
    overlap:0
 }
```

Figure 13. Sample Code Illustrating Data for Pointer Object

For the target object, the data recorded is the x and y coordinates, the timestamp of the record as well as the number of times the object bounces from one of the edges of the screen.

Data for both the target object and the pointer object was combined with the Quality of Experience rating of the corresponding round after every round, each of which corresponds to one of the 27 settings. The datafile also contained round information in order to identify the corresponding difficulty settings. The data was automatically downloaded as a text file on the local machine after the round was completed.. As the entire application was created using HTML, CSS and Javascript, the data had to be downloaded locally on the machine.

After each participant, the data was manually collected from the tablet, PC and VR device and stored on the cloud.

4.5 The Application

The UI of the final application is as show in Figure 14. On the left is the time taken in the previous round as well as number of rounds left for that device.

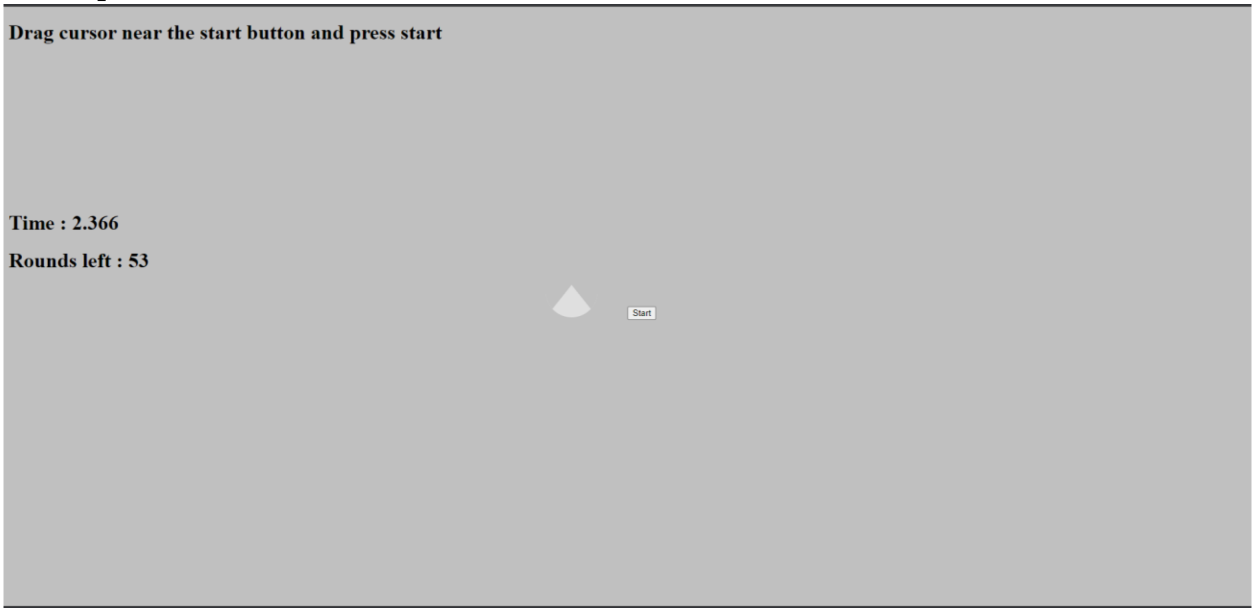

Figure 14. Application User Interface

The first two rounds on each device are practice rounds for the user to get familiar with the device. They were kept at the easiest setting for latency, speed and size. These were discarded during analysis. The user interface contains a start button at the center of the screen which starts the round and the timer for measuring the time taken by the user to select the target. The user then has to drag the cursor towards the target as quickly as possible and release it. When the cursor is inside the radius of the target and there is a mouseup event or touchend event for the touch and virtual reality interface the round is finished. The user is asked to rate the previous game session once for each setting (27 times per device) after which the recorded data is downloaded onto the local machine. The user is then expected to drag the cursor near the center of the screen again and the process is repeated to start a new round.

Figure 15. User Dragging Cursor to Target

The user is not provided any information about the changing settings in each round and the user is able to see the rounds left for the current device and time taken to select the target in the previous round.

When the user is finished with all the 56 rounds on a device the user is alerted to proceed to the next device after answering questions related to the device (full survey questions shown in the Appendix). The device orders were randomized as shown in Table 3 and there was an equal number of participants for each order.

Table 4. Table of Device Orders

First Device	Second Device	Third Device
PC	Tablet	VR
PC	VR	Tablet
Tablet	PC	VR
Tablet	VR	PC
VR	Tablet	PC
VR	PC	Tablet

4.6 Procedure for Participants

The procedure for each participant was as follows:

1. The user sanitized their hands.

2. The user signed an IRB approved consent form.

3. The user was explained the objective of the study which is to select the moving target as fast as possible and were told the order of devices they would use and assigned a participant ID.

4. The user answered a demographic questionnaire (full survey questions shown in the Appendix).

5. The user played the application rounds on a device with varying latency and difficulty and answered QoE questions.

6. After each device the user answered the device related questionnaire.

7. User repeats steps 5-6 for each of the devices.

5. Analysis

This section analyzes the data obtained from the user study.

5.1 Demographic Information

Our user study had 30 participants. Table 4 summarizes the main demographics with standard deviations in parentheses. "Gamer" is a self rating from 1-low to 5-high (full survey questions shown in the Appendix).

Table 5. Demographics Summary

N	Age	Gender	Gamer
30	20.7 (2.89)	19 (M) 10 (F) 1?	3.6 (1.21)

Table 5 summarizes the mean of how many hours per week the participant plays games and how many hours per week they use the study devices with standard deviations in parentheses.

Table 6. Hours Per Week Usage of Devices

Playing games Hours/Week	PC Hours/Week	Tablet Hours/Week	Virtual Reality Hours/Week
10.69 (10.29)	46.9 (17.4)	3.96 (7.64)	0.24 (0.68)

5.2 Performance and Quality Of Experience

5.2.1 Points at Maximum Time

Two data points for the PC and two data points for VR took more than 30 seconds, the maximum time allowed for the study. All 4 of these were at the high latency condition with small target size without QoE ratings.

5.2.2 Selection Time Analysis

This section analyzes the time taken to select the target according to latency, target speed along with target size and device type. In the cumulative distribution function the x axis is the selection time in seconds, and the y axis is the distribution. In the mean value graph, mean values with 95% confidence intervals are shown, the y axis is the selection time in seconds and the x axis is the parameter being grouped by.

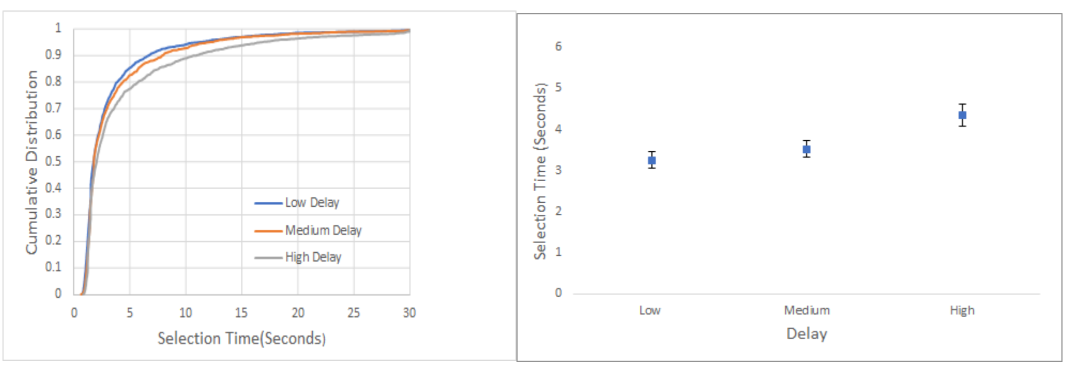

Figure 16. a) Distribution of Selection Time Grouped by Latency and b) Mean Selection Time Versus Delay with 95% Confidence Intervals

In Figure 16a, the distribution of selection time of low latency are relatively close to each other and similar whereas the high latency distribution has longer selection times. The median is around 0.2 seconds for all. All the distributions are similar in shape.

In Figure 16b, the mean values of selection time of low and medium delays are similar around 3.3 seconds whereas for high delays it is 4.3 seconds. Generally higher delays have higher selection times.

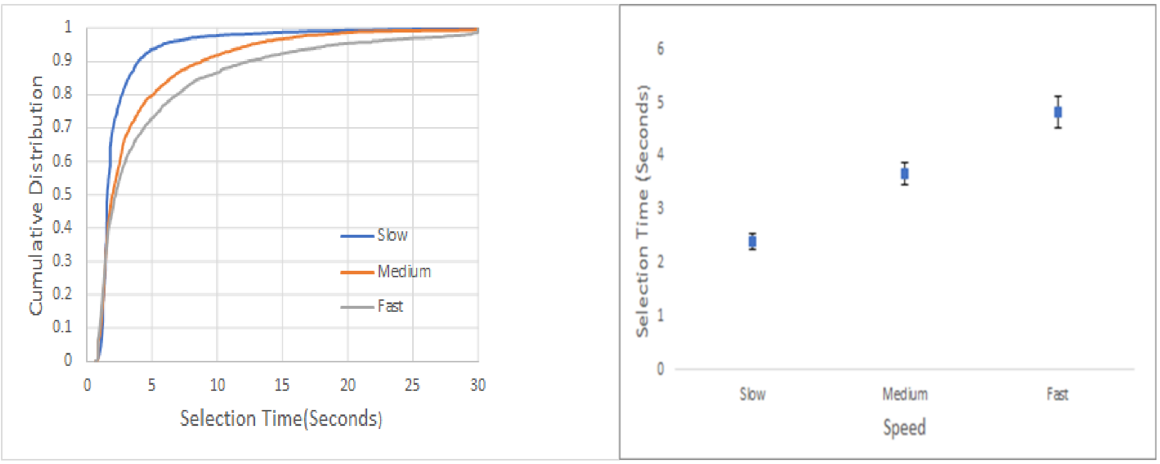

Figure 17. a) Distribution of Selection Time Grouped by Speed and b) Mean Selection Time Versus Speed with 95% Confidence Intervals

In Figure 17a, the selection time dsitributions distributions of medium and fast speeds are close to each other and similar whereas the slow speed distribution take a shorter selection time. The medians are at 0.2, 0.23 and 0.25 seconds, respectively.

In Figure 17b, the mean values of slow, medium and fast speeds are different at 2.4, 3.6 and 4.8 seconds, respectively.

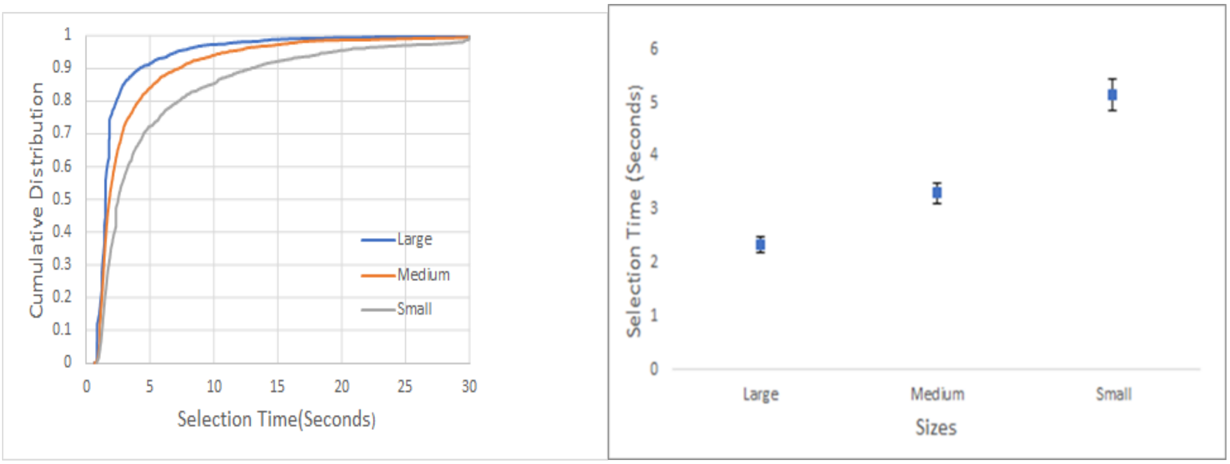

Figure 18. a) Distribution of Selection Time Grouped by Sizes and b) Mean Selection Time Versus Sizes with 95% Confidence Intervals

In Figure 18a, distributions of the large, medium and small size are equally spaced with small targets taking the longest selection times. The medians are at 0.20, 0.25 and 0.30 seconds for large, medium and small targets.

In Figure18b, the means of large, medium and small targets are separated at 2.3, 3.3 and 5.1 seconds, respectively.

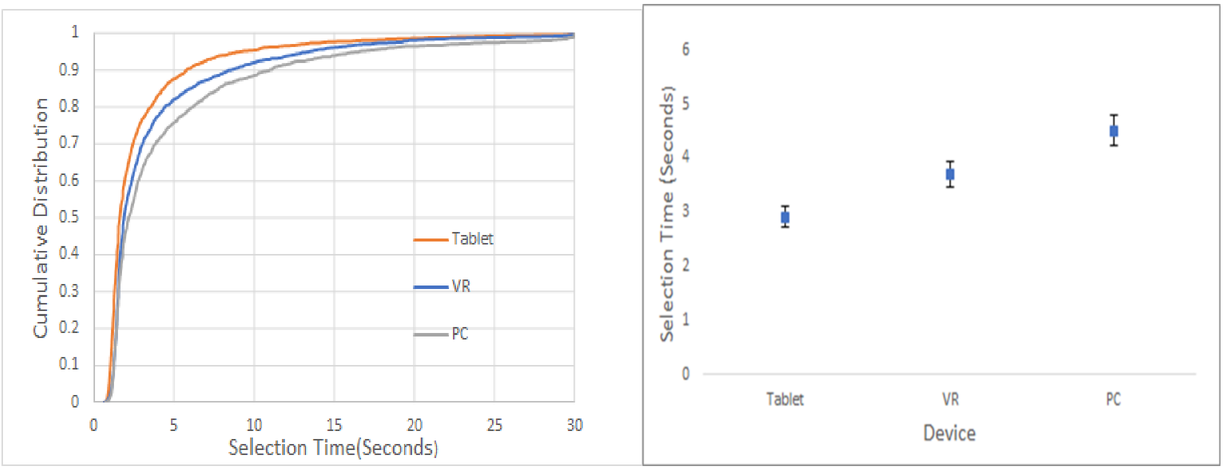

Figure 19. a) Distribution of Selection Time Grouped by Sizes and b) Mean Selection Time Versus Device with 95% Confidence Intervals

In Figure 19a, the distributions of the selection times across devices are equally separated with VR taking the shortest selection time and PC taking longest selection times. The medians are near 0.25 seconds for all.

In Figure 19b, the means of selection times across devices are well separated at 2.9, 3.7 and 4.5 seconds respectively.

5.2.3 Statistical Analysis of Selection Times

Kruskal Wallis tests with Bonferroni Correction and an alpha value of 0.05 showed significant differences in selection time across latency, speed, size and device. The statistics are as shown in Table 6.

Table 7. Table of Statistical Values for Selection Time

		Degree of Freedom (df)	H-Statistic	p value
	Low-Medium	1	10.0	<0.01
	Medium-High	1	11.9	<0.01
Delay	High-Low	1	41.6	<0.01
	Slow-Medium	1	56.6	<0.01
	Slow-Fast	1	89.26	0.01
Speed	Medium-Fast	1	6.55	0.01
	Small-Medium	1	104.8	0.02
Size	Medium-Large	1	152.72	0.01
	Large-Small	1	496.7	0.00
	PC-VR	1	15.58	<0.01
Device	VR-Tablet	1	91.4	0.02
	Tablet-PC	1	167	0.02

5.2.4 Click Analysis

This section analyzes the number of clicks of the different difficulty settings according to latency, target speed, target size and device type. The cumulative distribution functions are shown where the x axis is number of clicks and y axis is the distribution. The mean of the number of clicks are shown with 95% confidence intervals, where they axis is the number of clicks and the x axis is the parameter being grouped by.

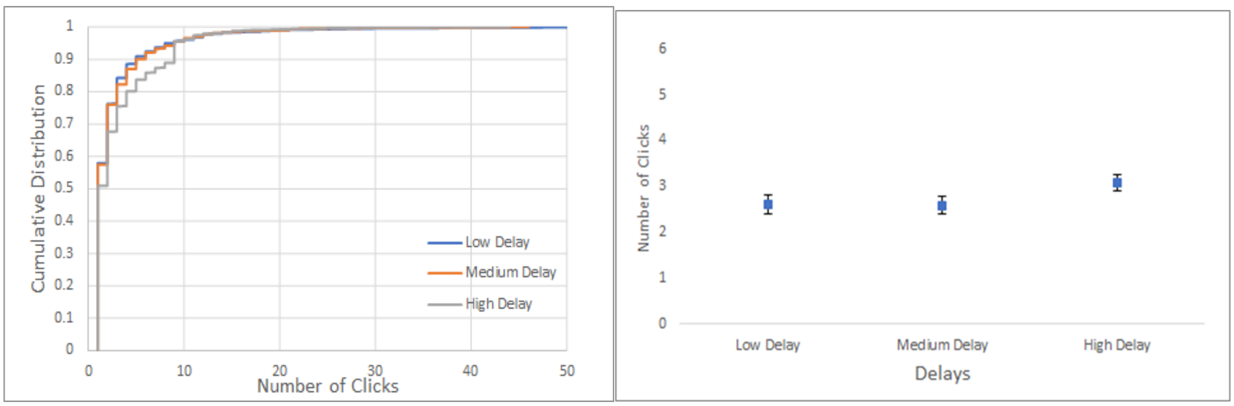

Figure 20. a) Distribution of Clicks Grouped by Latency and b) Mean Number of Clicks Versus Delays with 95% Confidence Intervals

In Figure 20a, the distributions of the low and medium latencies are similar. The high latency clicks are different in the distribution of values with more number of clicks. The median for all is 2 clicks.

In Figure 20b, the means for low and medium are similar at 2.6 seconds and the mean is 3.0 for high delays.

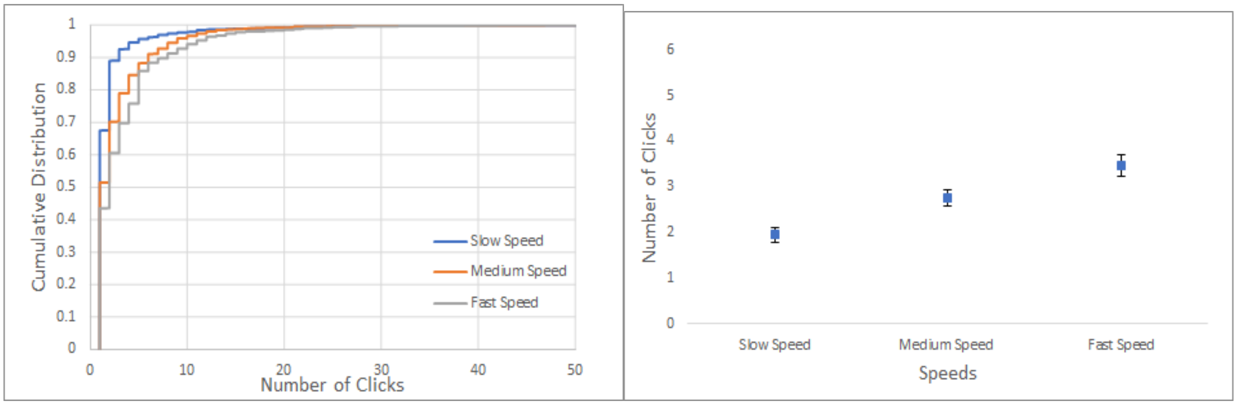

Figure 21. a) Distribution of Clicks Grouped by Speed and b) Mean Number of Clicks Grouped Versus Speeds with 95% Confidence Intervals

In Figure 21a, the distribution of number of clicks is similar for medium and high speeds with low speeds taking significantly fewer clicks. The median is 2 clicks for medium and low speeds and 3 for high speeds.

In Figure 221b, the means of low, medium and high speeds increase with the speed of the target.

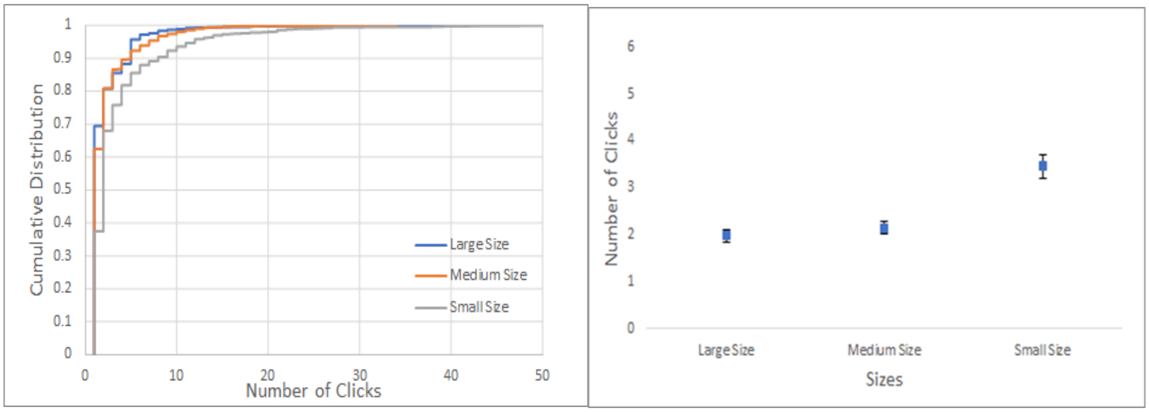

Figure 22. a) Distribution of Clicks Grouped by Size and b) Mean Number of Clicks Versus Sizes with 95% Confidence Interval

.

In Figure 22a, the distribution of large and medium size targets are similar whereas the distribution of small size targets is different and is highest. The median is 2 clicks for large and medium sizes and 3 for small size targets.

In Figure 22b, the selection time means of large and medium size targets are similar whereas the selection time mean is higher for small size targets.

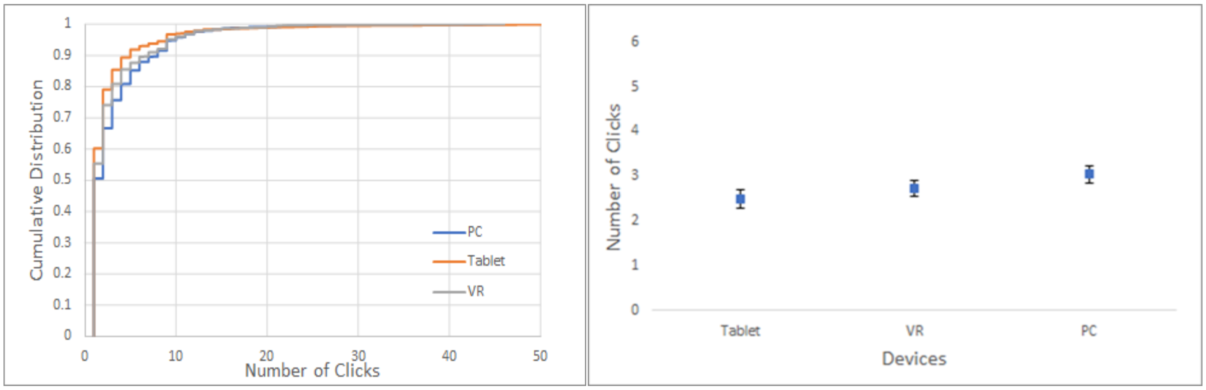

Figure 23. a) Distribution of Clicks Grouped by Device and b) Mean Number of Clicks Versus Device with 95% Confidence Intervals.

In Figure 23a, the distribution is different for the devices with PC having the higher number of clicks, followed by VR and then tablet. The medians for all is 2 clicks.

In Figure 23b, the mean of number of clicks is highest for PC at 3 clicks, 2.4 for tablet and 2.7 for VR.

5.2.5 User Experience

This section analyzes the Mean Opinion Score from 1 (Bad) to 5 (Excellent) of the different difficulty settings (full survey questions shown in the Appendix) according to latency, target speed, target size and device type. The cumulative distribution functions where the x axis is the mean opinion score and the y axis is the distribution are shown. The means of the mean opinion scores with 95% confidence intervals are shown where the y axis is the mean opinion score and the x axis is the parameter being grouped by.

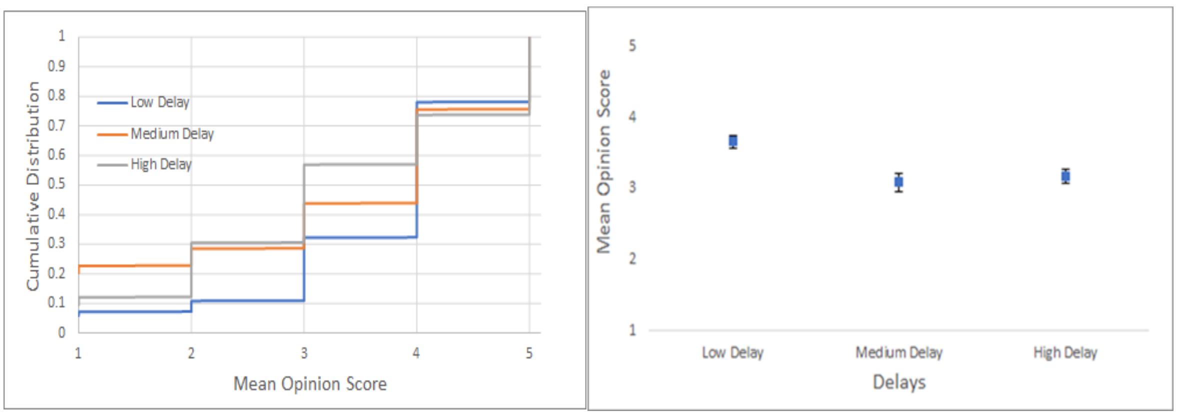

Figure 24. a) Distribution of Mean Opinion Score Grouped by Latency and b) Means of Mean Opinion Score versus Latency with 95% Confidence Intervals

.

In Figure 24a, the distributions of the low delay and medium delay values are shifted down and there are more 4 values and fewer 1-3 ratings. The median for high latency is 3 and for medium and low latency it is 4.

In Figure 24b, the means are distributed at 3.6, 3.1 and 3.2 for low medium and high delays respectively with low latency having the highest score.

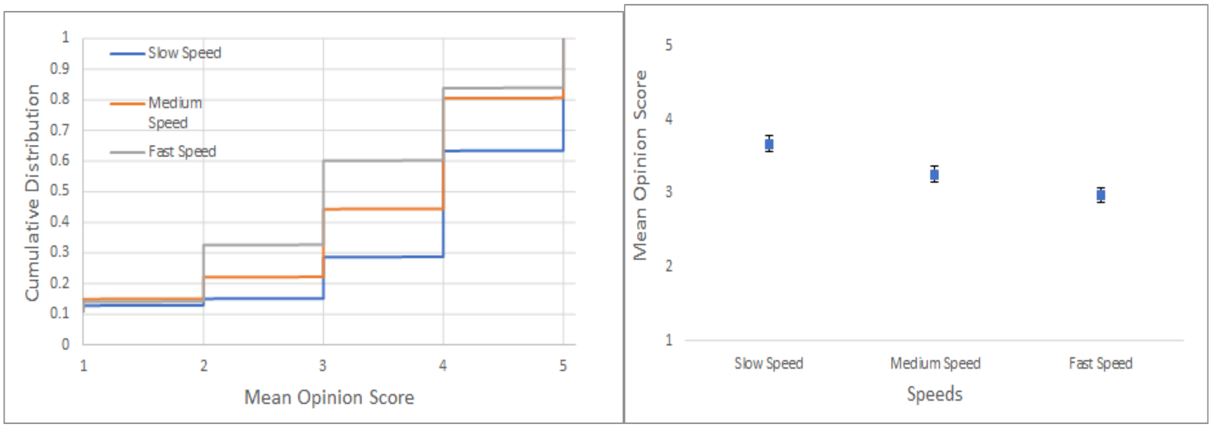

Figure 25. a) Distribution of Mean Opinion Score Grouped by Speed and b) Means of Mean Opinion Score versus Speed with 95% Confidence Intervals

In Figure 25a, for majority of the distribution the high speed has the highest score followed by medium speeds and low speeds. The median is around 3 for all.

In Figure 25b, the means of the low, medium and high speed scores are 3.6, 3.2 and 3.0 respectively.

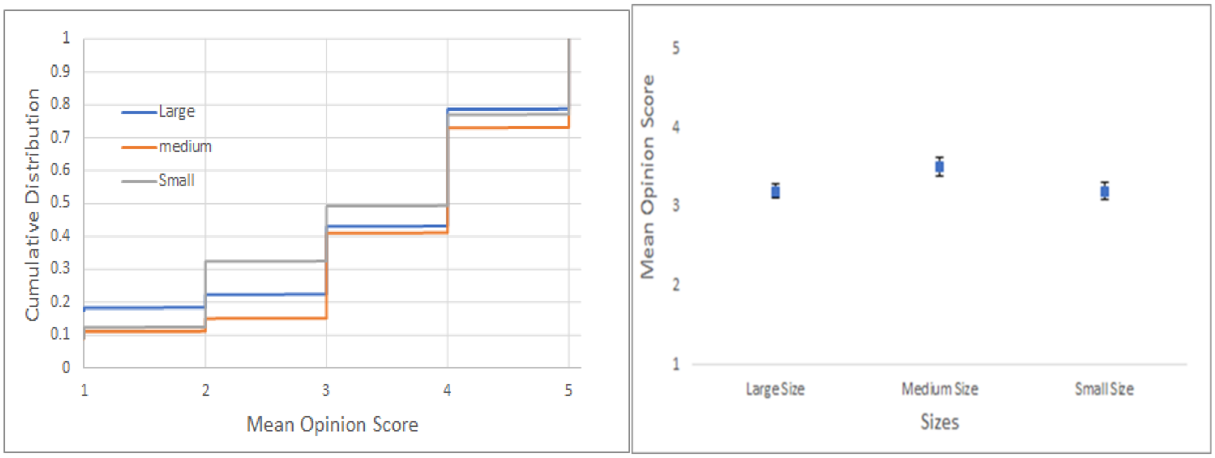

Figure 26. a) Distribution of Mean Opinion Score Grouped by Size and b) Means of Mean Opinion Score versus Sizes with 95% Confidence Intervals.

In Figure 26a, the gap between large and medium values gets smaller till 4 and there are more 4 values than 1-3 ratings. The median is at 3 for all.

In Figure 26b, the mean scores for large, medium and small sizes are 3.2, 3.5 and 3.2 respectively.

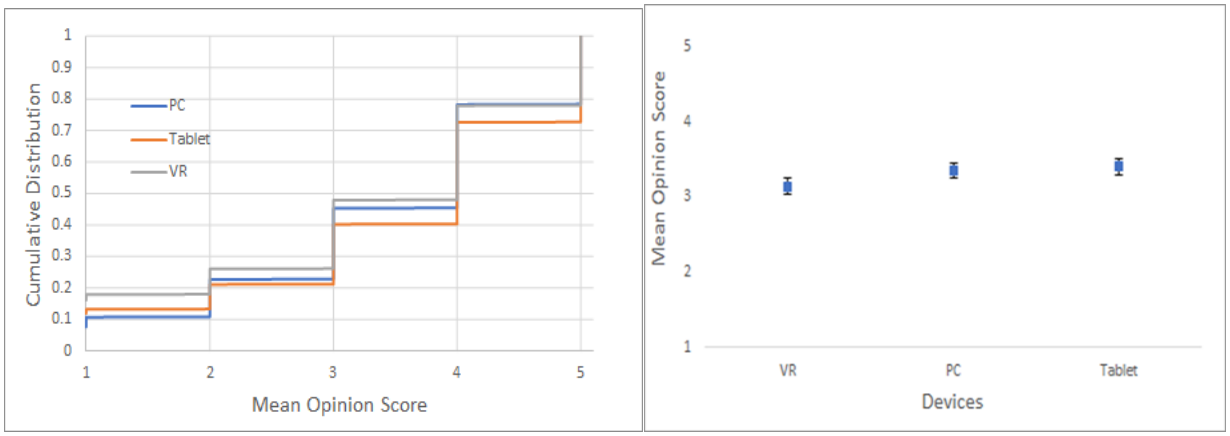

Figure 27. a) Distribution of Mean Opinion Score Grouped by Devices and b) Means of Mean Opinion Score versus Devices with 95% Confidence Intervals.

In Figure 27a, for the majority of the distribution VR has the highest scores, followed by PC, followed by tablet. The median is at 3 for all.

In Figure 27b, the mean scores for VR, PC and tablet are 3.1, 3.3 and 3.4 seconds respectively.

5.3 Interaction Analysis

In this section we analyze the interactions between device type, latency speed and size of the target. The y axis is the performance metric or QoE rating and the x axis is the parameter being grouped by.

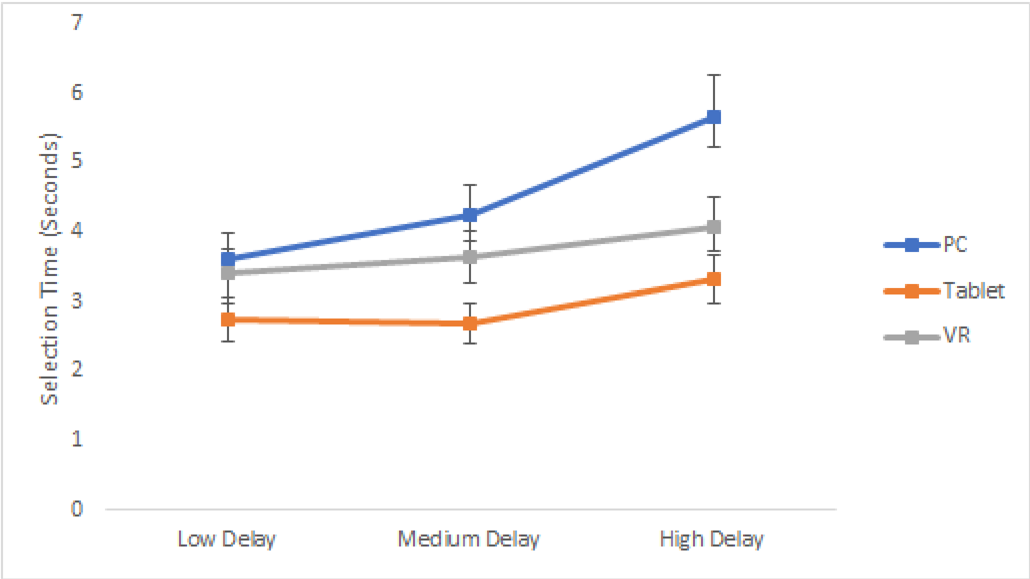

Figure 28. Interaction of Selection Time with Device and Latency

In Figure 28, for delay, PCs have higher selection times than VR, which in turn are higher than tablets. The largest change in selection time is for medium to high latency of the PC.

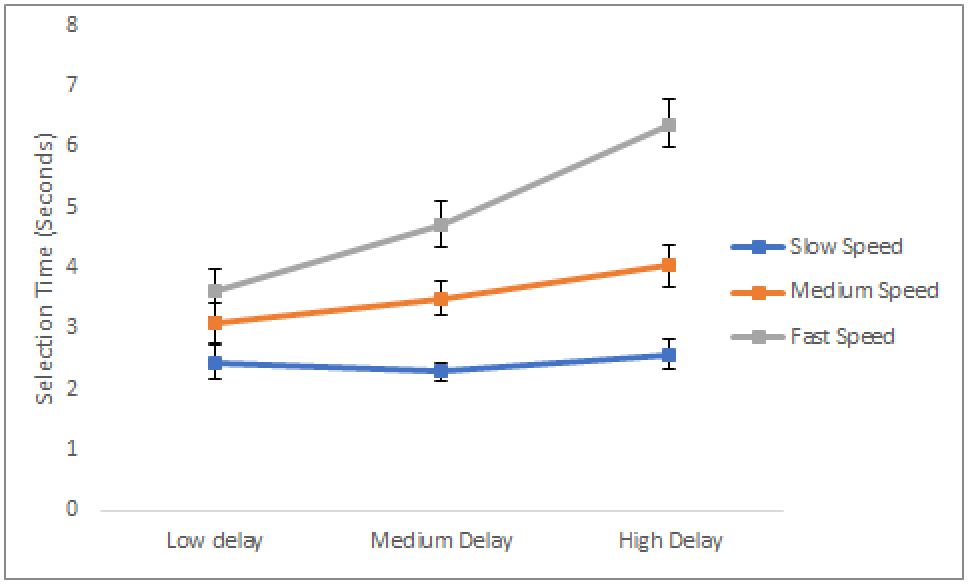

Figure 29. Interaction of Selection Time Between Speed and Latency

In Figure 29, selection times increase across delays for medium and fast speeds but it remains similar for slow speed targets.

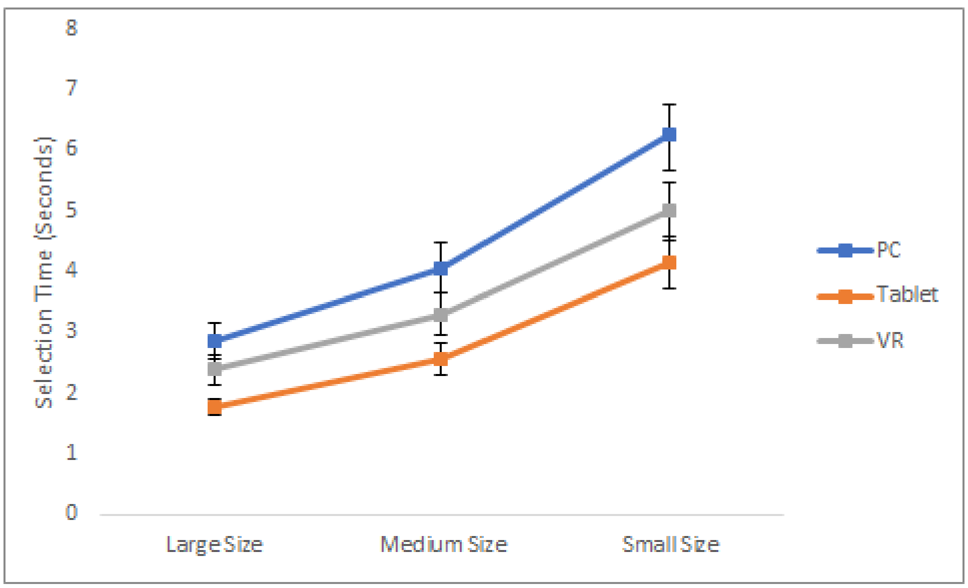

Figure 30. Interaction of Selection Time Between Size and Device

In Figure 30, a similar trend of selection time across devices with PC having higher selection time than VR which has higher selection times than tablet.

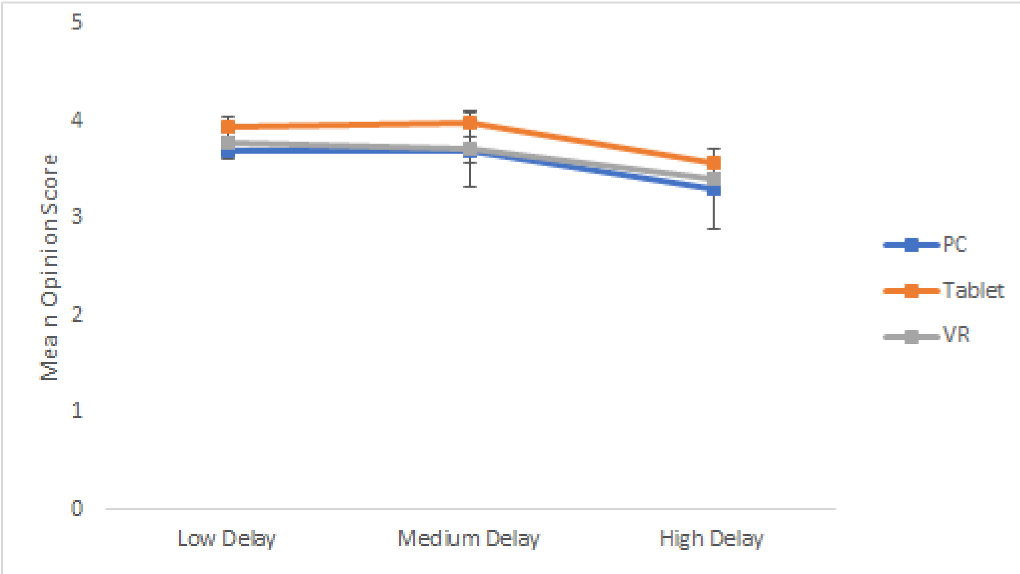

Figure 31. Interaction of Mean Opinion Score Between latency and device

In Figure 31, the trend of Mean Opinion Score across delays is similar for VR and PCs but is higher for tablets.

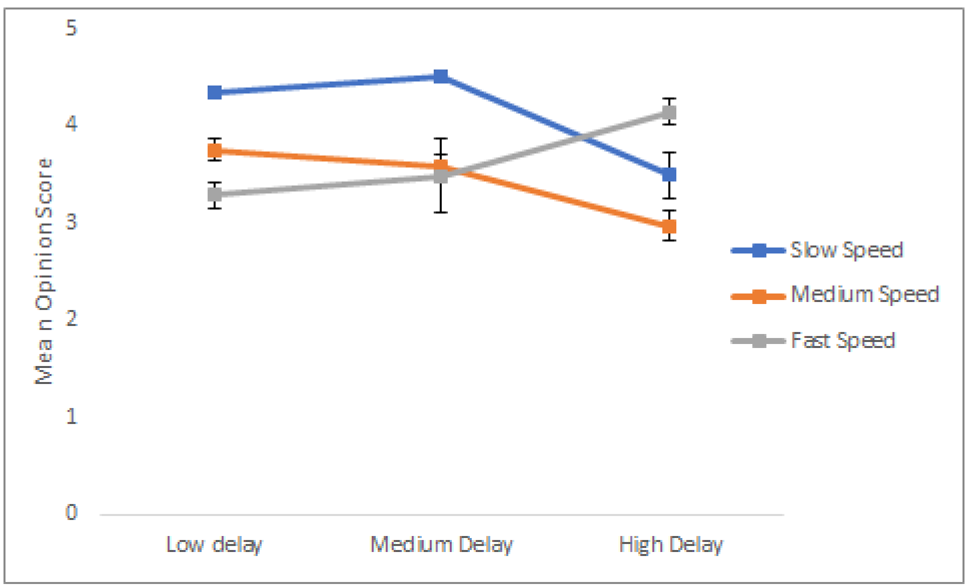

Figure 32. Interaction of MOS Score Between Speed and Latency

In Figure 32, the Mean Opinion Score has a decreasing trend with latency for slow and medium speeds but is increasing for fast speeds.

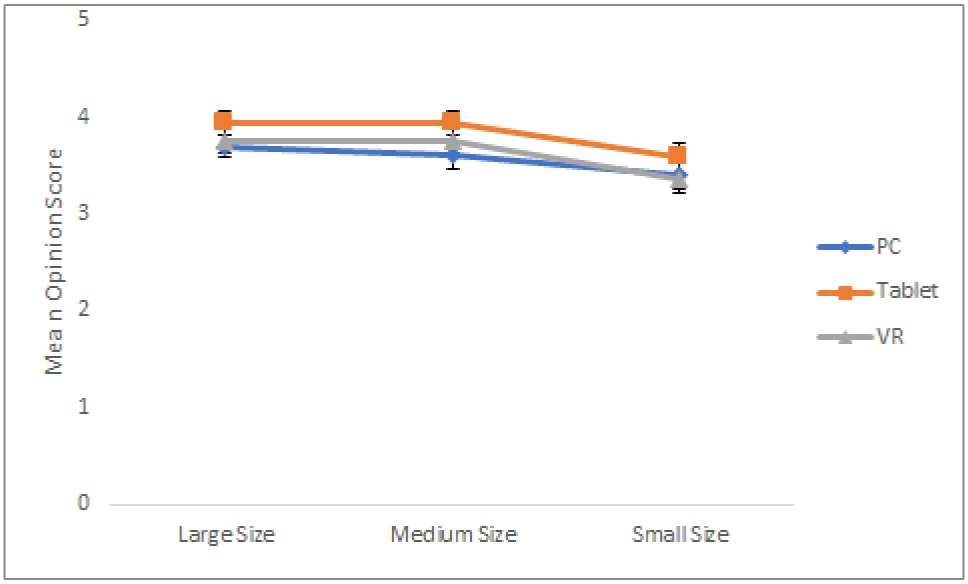

Figure 33. Interaction of Mean Opinion Score Between Size and Device

In Figure 33, Mean Opinion Score is decreasing for increasing difficulty of sizes from large to small target size.

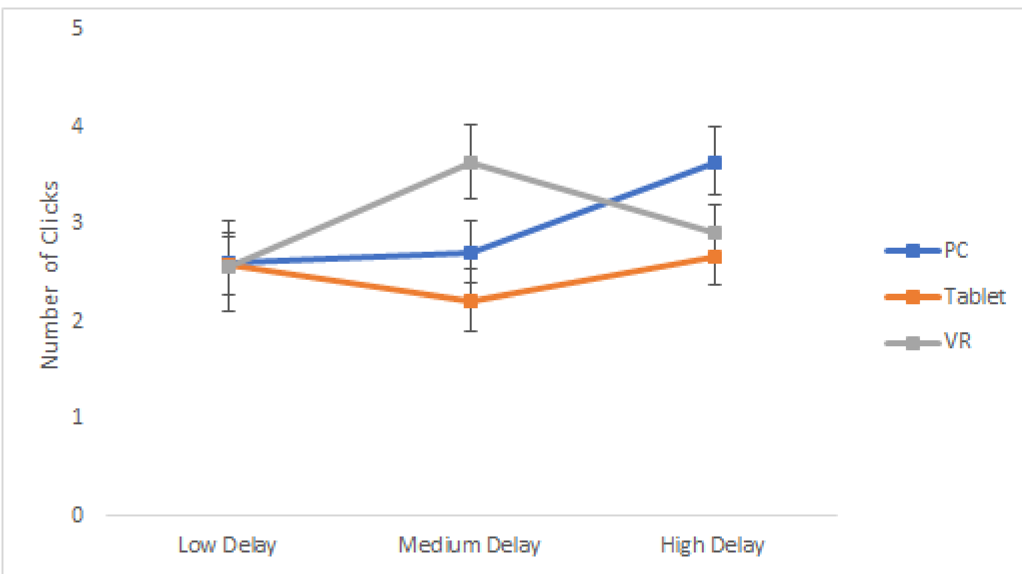

Figure 34. Interaction of Clicks Between Latency and Device

In Figure 34, number of clicks and latency do not have any clear relationship.

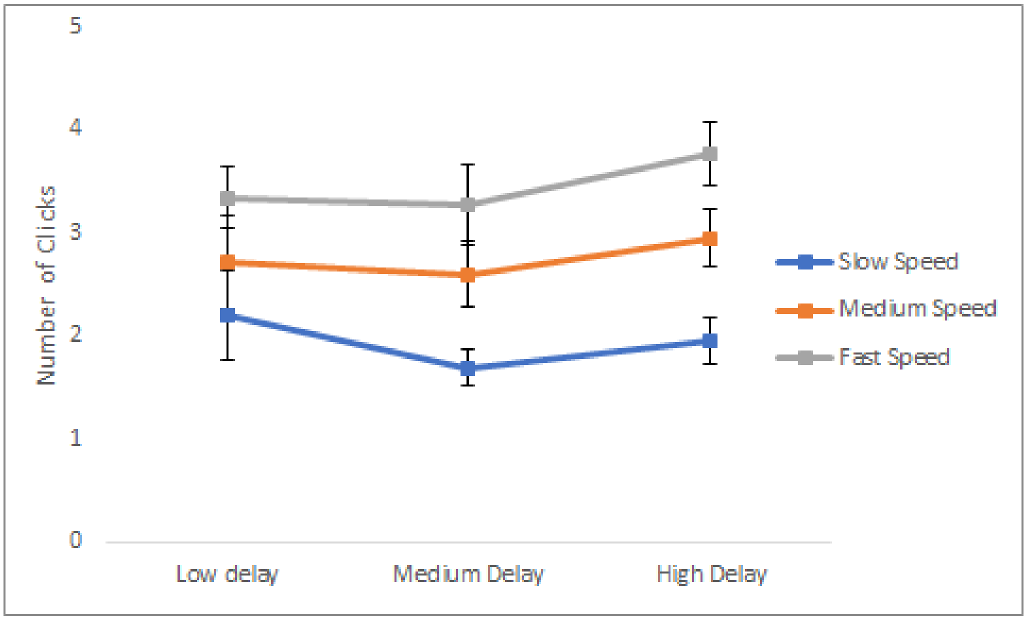

Figure 35. Interaction of Clicks Between Latency and Speed

In Figure 35, fasts speeds have more clicks than medium speeds which in turn have more clicks than slow speeds.

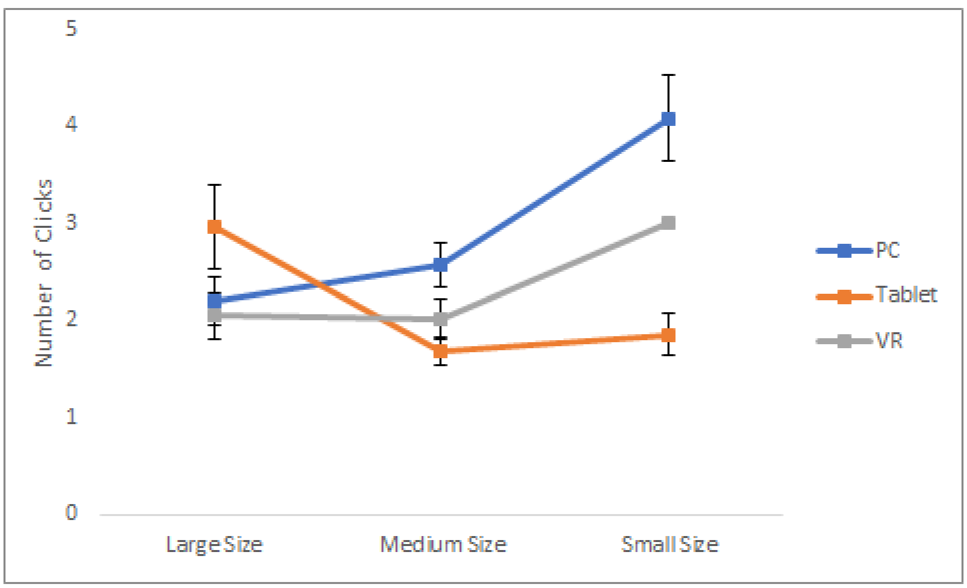

Figure 36. Interaction of Clicks Between Size and Device

In Figure 36, clicks decreases sharply for large and medium size targets but is increasing for PC and VR.

5.4 Fitts' Law Analysis

This section analyzes the selection time and ID Score of task difficulty for the stationary targets according to device type and latency. According to Fitts' Law [8] the difficulty of selecting a target is dependent on two factors: the distance to reach the target as well as the width of the target. The Index of Difficulty (ID) Score for selecting a stationary target is computed as:

$$ID\ Score = \log_2(2D/W) \qquad \text{.... (1)}$$

Where D is the distance to the target and W is the width of the target.

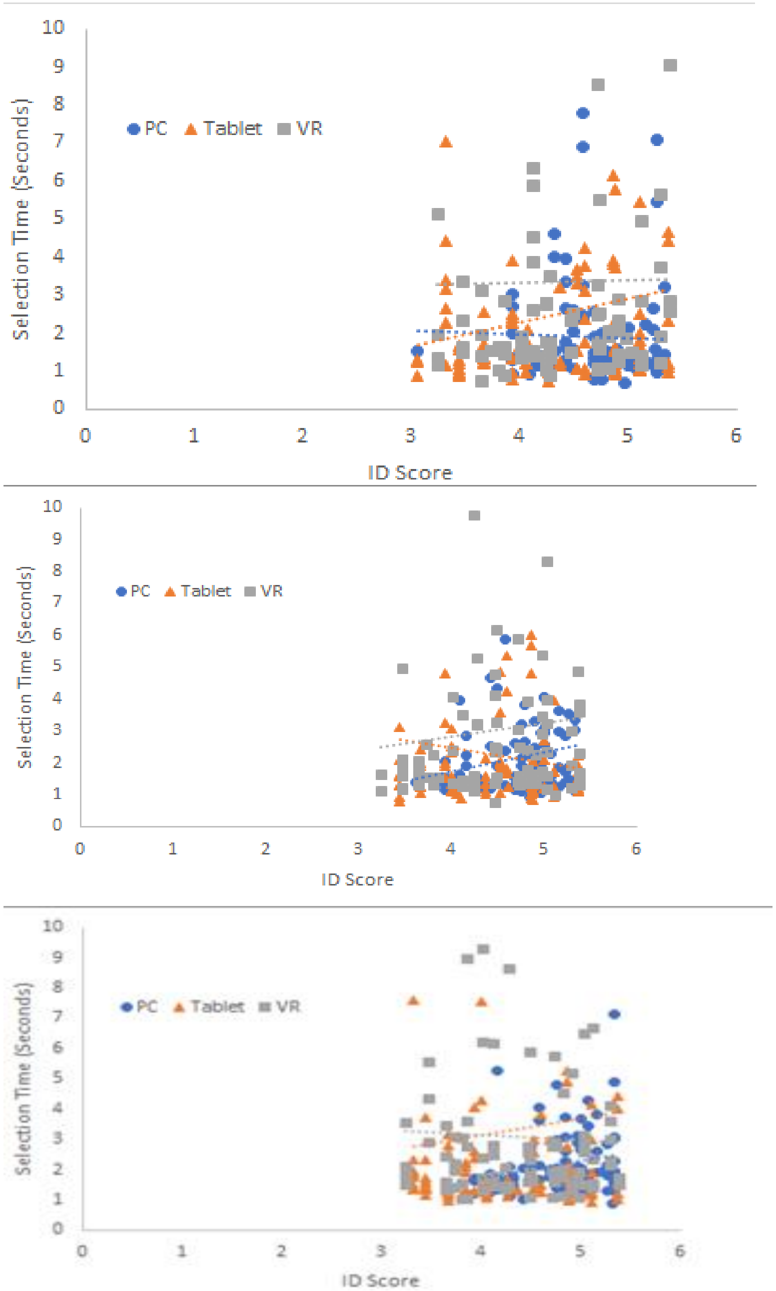

Figure 37. Selection Time versus ID Score for low, medium and high latency

The Slope, Y intercept and R^2 values of each group are as shown in Table 7.

Table 8. Table of Linear Regression Metrics and R^2 Values

Latency	Device	R^2	Slope	Y Intercept	P Value
Low	PC	0.00	0.0	2.3	0.78
Low	Tablet	0.00	0.1	1.8	0.28
Low	VR	0.00	0.0	3.1	0.34
Medium	PC	0.02	0.5	0.6	0.11
Medium	Tablet	0.03	-0.3	3.7	0.28
Medium	VR	0.00	0.4	1.0	0.75
High	PC	0.05	0.5	-0.5	0.02
High	Tablet	0.67	0.5	1.0	0.48
High	VR	0.00	-0.1	3.8	0.03

In Figure 37, as the Index of Difficulty is increasing the time to select the target also increases and the ID scores can be separated for the devices. There is an upward trend in the majority of the data which is in accordance with Fitts' Law.

5.5 Analysis of Devices

Table 4 summarizes the data from device usage. All the participants were asked to rate their interaction experience, ease of use of the device, feel of control and immersiveness of the device on a scale of 1(low) to 5(high) (full survey questions shown in the Appendix). Table 8 shows the mean of the responses with the standard deviation in paranthesis.

Table 9. Table of Device Interaction Summary

Device	Interaction Experience	Ease of Use of Device	Feeling of Control	Immersiveness of Device
PC	3.26 (1.28)	3.59 (1.19)	3.1 (1.21)	2.4 (1.22)
Tablet	3.9 (1.02)	3.9 (0.99)	3.6 (1.1)	2.83 (1.01)
VR	3.76 (0.85)	3.6 (0.99)	3.53 (1.04)	3.9 (1.12)

6. Discussion

This section answers our research questions based on our analysis.

- RQ1-How does player performance differ according to device?

  The PC with the mouse has the highest selection time. One of the possible reasons for this is the effect of friction on dragging the mouse whereas the VR controller is moved in free air and the tablet requires human touch. The tablet has the lowest selection time with no intermediate pointing device since the interaction is with the human finger. The VR controller has selection times in-between the tablet and the PC. The tablet has fewest number of clicks followed by VR. The PC with mouse has the highest number of clicks.

- RQ2-How does player performance differ according to latency?

  The high latency targets have higher selection times compared to the medium latency targets, and the medium latency targets in turn have higher selection times compared to the low latency targets. As latency increases, selection time increases. The number of clicks is similar for low and medium delays but is higher for high delays.

- RQ3-How does player performance differ according to task difficulty?

  Task difficulty varied with target speed (slow, medium and fast) and target size (large, medium and small) as show in Table 2. Both selection times have a linear relationship with target size and speed. As task difficulty increases, selection time increases. The number of clicks is similar for large and medium size targets but is higher for large size targets.

- RQ4-What are the differences in device/interaction mode on the effect of latency on performance?

  For medium to high delay, the PC has the highest increase in selection times followed by the tablet followed by VR. These changes are not as abrupt for low to medium delay. For increasing delay, fast moving targets have higher selection times than targets moving at a medium speed. Targets moving at a medium speed have higher selection times than targets moving at a slow speed. As the size of the targets decrease, the PC has the highest increase in selection times followed by VR, followed by the tablet.

- RQ5-What is the user experience according to latency, task difficulty and device?

  The Mean Opinion Score decreases for high latency compared to medium latency targets, suggesting a degradation in player experience. With slow speed targets, users have rated a higher MOS score compared to medium speed targets. With medium speed targets, users have rated a higher MOS score compared to fast speed targets. Medium size targets have a higher mean opinion score than small or large targets.

7. Conclusion

Both local latency and network latency impact player performance and experience for computer games on a variety of devices. Since most games have a combination of different actions, it is hard to understand the effects of latency on gaming. We focus on the isolated action in gaming rather than the entire gaming application. Although moving target selection has been studied before, to the best of our knowledge, a study across PC, virtual reality (VR) and tablets with varying delays and task difficulty has not been done. These devices were chosen as they provide different interaction techniques: the mouse for the PC, human touch for the tablet, as well as virtual reality with a controller.

We built a custom application using HTML, CSS and Javascript to study the isolated action of moving target selection across devices. We used the same setup for each device, varying delay and difficulty. We tested our application and obtained data from 30 participants, mostly students in their early 20's.

Both varying delays and task difficulty affects the selection times of the users, clicks and quality of experience. The higher delays and higher task difficulties according to target size or speed have longer selection times compared to targets with lower delays and lower speeds. The tablet also has shortest selection time and lowest number of clicks. The PC with mouse has highest number of clicks and longest selection times.

Low delay has a higher mean opinion score compared to medium and high delays. The tablet has the highest mean opinion score. The tablet also has the highest ratings for ease of use, interaction experience and feeling of control and the VR device has the highest rating for immersiveness.

8. Future Work

This section presents possible future work in this area**.** Our work has focused on PC, tablet and VR devices. Similar studies are possible for different gaming devices such as different game controller input devices and other handheld devices including interaction with the help of a stylus. For the VR device, we have focused on a portable VR but the same study can be extended to non-portable VR devices which have wires attached and different controller for both hands.

Future studies can recruit participants with a broader demographics and gamer experience. Most participants in this study were more experienced with PC and Tablet devices compared to VR as it is not commonly used. Future studies could be done with participants that have equal experience of usage across the devices.

We have focused on the isolated task of moving target selection, similar studies can be done for different gaming actions such as exploring or shooting.

Our study simulated latency as would be expected in a cloud gaming platform, but similar studies across devices can be done to simulate network latency as would be experienced by traditional multiplayer networked game architectures.

9. Appendix

9.1 Demographic Questions:

Target Clicker

Survey Questions For User Study

*Required

1. What is your participant ID? *

2. What is your age?

3. What is your gender?

   Mark only one oval.

   - Male
   - Female
   - Non Binary
   - Prefer Not To Say

4. How many hours per week do you play computer games?

5. How many hours per week do you use a PC?

6. How many hours per week do you use a tablet?

7. How many hours per week do you use a VR device?

8. Rate your experience as a gamer

   Mark only one oval.

	1	2	3	4	5	
Low						High

Figure 38. Demographic Questions

9.2 Device Related Questions

1. What is your participation ID? *

2. Rate your interaction experience

   Mark only one oval.

   - Excellent
   - Good
   - Fair
   - Poor
   - Bad

3. Rate ease of use of this device

   Mark only one oval.

	1	2	3	4	5	
Low						High

4. Rate your feeling of control on this device

   Mark only one oval.

	1	2	3	4	5	
Low						High

5. Rate how immersive was this device

   Mark only one oval.

	1	2	3	4	5	
Low						High

Figure 39. Device Related Questions

9.3 Round Related Questions

- How was the previous game session? (Enter a number [1-5])", "1.Bad 2.Poor 3.Fair 4.Good 5.Excellent:

References

1. *Statista.* [Online] [Cited: March 15, 2021] https://www.statista.com/statistics/246888/value-of-the-global-video-game-market/

2. *Pew Research Center*. [Online] [Cited: March 15, 2021] https://www.pewresearch.org/internet/fact-sheet/mobile.

3. *Comparing the Gear VR, Oculus Go, and Oculus Quest. Unreal for Mobile and Standalone VR.* C. Hillmann. 2019. pp. 141-167.

4. *Oculus.* [Online] [Cited: March 15, 2021]. https://www.oculus.com/.

5. *A Comparison of Automatic versus Manual World Alteration for Network Game Latency Compensation.* R. Salay, M. Claypool. 2020. ACM Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '20). pp. 355-359.

6. *Latency and Player Actions in Online Game.* M. Claypool, K. Claypool. New York : Association for Computing Machiner, 2006. Communications of the ACM.

7. *Game Player Response Times versus Task Dexterity and Decision Complexity.* C. Jiang, A. Kundu, M. Claypool. 2020. Proceedings of the ACM Annual Symposium on Computer-Human Interaction in Play (CHI PLAY).

8. *The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement.* P. Fitts. 1954. pp. 381-391.

9. *Game Input with Delay - Moving Target Selection Parameters.* M. Claypool, A. Cockburn, C. Gutwin. 2019. In Proceedings of the 10th ACM Multimedia Systems Conference (MMSys), Amherst, MA, USA.

10. *Android Authority.* [Cited: March 15, 2021.] https://www.androidauthority.com/pubg-vs-pubg-mobile-858416/.

11. *N. Objective and Subjective Evaluation of the Influence of Small Amounts of Delay and Jitter on a Recent First Person Shooter Game.* P. Quax., P. Monsieurs., W. Lamotte,D.D. Vleeschauwer,N. Degrande. 2004. Proceedings of ACMNetGames.

12. *A First Look at the Network Turbulence for Google Stadia Cloud-based Game Streaming,.* X. Xu, M. Claypool. 2021. Proceedings of the IEEE Global Internet Symposium (GI), Virtual Conference.

13. *Game Input with Delay - A Model of the Time Distribution for Selecting a Moving Target with a Mouse.* S. Liu, M. Claypool. 2021. Proceedings of the 27th International Conference on MultiMedia Modeling (MMM).

14. *Oculus Go.* [Online] March 15, 2021. https://www.oculus.com/go/.

15. *Game Input with Delay - Moving Target Selection with a Game Controller Thumbstick.* M. Claypool. 2018. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM).Special Issue on Delay-Sensitive Video Computing in the Cloud.

16. *Playing with delay: An Interactive VR Demonstration.* K. Raaen, R. Eg, I. Kjellmo. 2019. Association for Computing Machinery. Proceedings of the 11th ACM Workshop on Immersive Mixed and Virtual Environment Systems MMVE '19. pp. 19–21.

17. *Latency Can Kill: Precision and Deadline in Online Games.* M. Claypool, K. Claypool. 2010. Proceedings of the first annual ACM SIGMM conference on Multimedia systems (MMSys '10). pp. 215–222.

18. *Latency Compensating Methods in Client/Server Ingame Protocol Design and Optimization.* Y.W.Bernier. 2001. Proceedings of the Game Developers. p. 13.

19. *The Impact of Motion and Delay on Selecting Game Targets with a Mouse.* M. Claypool, A. Cockburn, C. Gutwin. 2020. ACM Transactions on Multimedia Computing, Communications. Appl. 16, 2s, Article 73.

20. *Effects of Local Latency on Game Pointing Devices and Game Pointing Tasks.* C. Gutwin, M. Long. 2019. In Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI).

21. *Effects of Tracking Technology, Latency, and Spatial Jitter on Object Movement.* R. J. Teather, A. Pavlovych, W. Stuerzlinger, and I. S. MacKenzie. 2009. Proceedings of the IEEE Symposium on 3D User Interfaces. pp. 43-50.

22. *Lag as a Determinant of Human Performance in Interactive Systems.* Ware, I. S. MacKenzie and C. Ware. 1993. Proceedings of the CHI Conference on Human Factors in Computing Systems. p. 6.

23. *The Effects Of Low Latency On Pointing and Steering Tasks.* S. Friston, P. Karlström, and A. Steed. 2016. IEEE Transactions on Visualization and Computer Graphics 22. pp. 1605–1615.

24. *Quantifying and Mitigating The Negative Effects of Local Latencies on Aiming in 3D Shooter Games.* Z. Ivkovic, I. Stavness, C. Gutwin, and S. Sutcliffe. 2015. Proceedings of the 2015 CHI International Conference on Human Factors in Computing Systems. pp. 135–144.

25. *Modeling User Performance For Moving Target Selection With a Delayed Mouse.* M. Claypool, R. Eg and K. Raaen. 2017. Proceedings of the 23rd International Conference on MultiMedia Modeling (MMM).

26. *Assessing Target Acquisition and Tracking Performance For Complex Moving Targets in The Presence of Latency and Jitter.* C.Gutwin, A. Pavlovych. 2012. Proceedings of the Graphics Interface Conference. pp. 109-116.

27. *Wikipedia*. [Cited: March 15, 2021] https://en.wikipedia.org/wiki/Oculus_Go.

28. *Interact.JS*. [Cited: March 15, 2021]. https://interactjs.io/.

29. *Game Development Stack Exchange*. [Cited: March 15, 2021] https://gamedev.stackexchange.com/questions/23672/determine-resulting-angle-of-wall-collision.

30. F*actors Influencing the Latency of Simple Reaction Time.* D.L. Woods, J.M. Wyma, E.W. Yund, T.J. Herron, B. Reed. 2015. Frontiers In Human Neuroscience. p. 9:131.

31. *Is There Any Evidence of Historical Slowing of Reaction Time? No, Unless We Compare Apples and Oranges.* Y. A. Dordonova, Y. S. Dordonova. 2013. Intelligence., pp. 674–687.

32. *Human BenchMark*. [Online] [Cited: March 15, 2021.] https://humanbenchmark.com/dashboard..

33. *Understanding Tablet Use: A Multi-Method Exploration.* H. Müller, J. Gove, and J. Webb. 2012. Proceedings of the 14th International conference on Human-computer interaction with Mobile Devices and Services (MobileHCI '12). pp. 1-10.

34. *Capabilities of ARCore and ARKit Platforms for AR/VR Applications.* P. Nowacki, M. Woda . 2019. Proceedings of the 14th International Conference on Dependability of Computer Systems DepCoS-RELCOMEX .

35. *Virtual Reality Experience: Immersion, Sense of Presence, and Cybersickness.* J. C. Servotte, M. Goosse, S. Campbell, N. Dardenne, B. Pilote, I. Simoneau, M.

Guillaume,I. Bragard, A. Ghuysen. 2020. Clinical Simulation in Nursing. pp. 35-43.

36. *Presence and Cybersickness In Virtual Reality Are Negatively Related: A Review.* S. Weech, S. Kenny, M. B. Cowan. 2019. Frontiers in Psychology. pp. 10-158.

37. *Virtual Reality Games: A Study About the Level Of Interaction vs. Narrative And The Gender in Presence And Cybersickness.* G. Gonçalves, M. Melo, M. Bessa. 2018. International Conference on Graphics and Interaction (ICGI) pp. 1-8.

38. *Accuracy of Different Modalities of Reaction Time Testing: Implications for Online Cognitive Assessment Tools.* J. Holden, E. Francisco, R. Lensch, A. Tommerdahl, B. Kirsch, Bryan, L. Zai, R. Dennis, M. Tommerdahl. 2019.

39. *Pew Research Center*. [Online] [Cited: March 15, 2021].https://www.pewresearch.org/fact-tank/2018/09/17/5-facts-about-americans-and-video-games/.

40. *Android Authority* [Online] [Cited: March 15, 2021]. https://www.androidauthority.com/pubg-vs-pubg-mobile-858416/.