A NEW TOUCH-BASED INPUT METHOD FOR MOBILE FIRST-PERSON-SHOOTER GAMES AND COMPARATIVELY EVALUATING IT USING THE ISO 9241-9 TEST

BEYZA HİLAL DURAK

Master's Thesis

Graduate School
İzmir University of Economics
İzmir
2022

A NEW TOUCH-BASED INPUT METHOD FOR MOBILE FIRST-PERSON-SHOOTER GAMES AND COMPARATIVELY EVALUATING IT USING THE ISO 9241-9 TEST

BEYZA HİLAL DURAK

A Thesis Submitted to
The Graduate School of Izmir University of Economics
Master's Program in Computer Engineering

İzmir
2022

ABSTRACT

A NEW TOUCH-BASED INPUT METHOD FOR MOBILE FIRST-PERSON-SHOOTER GAMES AND COMPARATIVELY EVALUATING IT USING THE ISO 9241-9 TEST

Durak, Beyza Hilal

Master's Program in Computer Engineering

Advisor: Asst. Prof. Dr. Gazihan Alankuş

July, 2022

The study aims to compare the available touch-based input methods which we called as Position-Button and Velocity-Screen, regarding accuracy and speed in point selection tasks and analyse the missing parts. Two novel input methods are proposed called Two-Finger Indirect and Two-Finger Direct, and then all of them were compared. The evaluation has been performed using multi-directional tapping task in ISO 9241-9 on a 9.7-inch iPad Pro. It has been observed that one of the proposed methods, Two-Finger Direct, is better than existing methods in terms of throughput although it also had the worst accuracy. These results indicate that existing methods can be improved.

Keywords: Fitts' law, ISO 9241-9, Target shooting, First-person shooter games

ÖZET

MOBİL BİRİNCİ ŞAHIS NİŞANCI OYUNLARINDA KULLANILAN DOKUNMA BAZLI GİRDİ YÖNTEMLERİNİN VE YENİ ÖNERECEĞİMİZ SİSTEMİN ISO 9241-9 TESTİ KULLANILARAK KARŞILAŞTIRILMALI OLARAK DEĞERLENDIRILMESI

Durak, Beyza Hilal

Bilgisayar Mühendisliği Yüksek Lisans Programı

Tez Danışmanı: Dr. Ögr. Üyesi Gazihan Alankuş

Temmuz, 2022

Çalışma, “Konum-Buton” ve “Hız-Ekran” olarak adlandırdığımız mevcut dokunmatik tabanlı giriş yöntemlerini nokta seçim görevlerinde doğruluk ve hız açısından karşılaştırmayı ve eksik kısımları analiz etmeyi amaçlamaktadır. “İki Parmak İndirekt” ve “İki Parmak Direkt” olarak adlandırılan iki yeni giriş yöntemi önerilmiş ve ardından hepsi karşılaştırılmıştır. ISO 9241-9 testindeki çok-yönlü tıklama görevi kullanılarak bir 9.7 inç iPad Pro üzerinde değerlendirme yapılmıştır. Önerilen yöntemlerden biri olan “İki Parmak Direkt”'in, en kötü doğruluğa sahip olmasına rağmen, verim açısından mevcut yöntemlerden daha iyi olduğu gözlemlenmiştir. Bu sonuçlar mevcut yöntemlerin geliştirilebileceğini göstermektedir.

Anahtar Kelimeler: Fitts’in yasası, ISO 9241-9, Hedef vurma, Birinci şahıs nişancı

oyunları

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my advisor Gazihan Alankuş for his guidance throughout the research. Additionally, I would like to thank my family and all my friends, especially Ümmügülsüm Ünlüer, who supported me during this research process.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

FPS: First-person shooter

HCI: Human-Computer Interaction

TP: Throughput

ID: Index of Difficulty

A: Amplitude

W: Width

CHAPTER 1: INTRODUCTION

Games have reached a new era with the rapid development in computer and network technologies. The widespread use of computers and smartphones has made it possible to play games on various new platforms. According to Newzoo Global Games Market Report, the number of players in 2021 is 3 billion globally. Most of these players play on mobile devices. The number of mobile players is 2.8 billion whereas the number of PC players is 1.4 billion. It is estimated that 3.1 billion mobile players will be in 2024 (Newzoo Global Games Market Report, 2021).

Gaming on touchscreen devices is commonplace. Increased computing power in smartphones enables these devices to run complex video games. As smartphones with touch input interfaces become more common, it is important to examine the strengths and weaknesses of touch input in a mobile context. First-person shooter (FPS) games are known for their input-heavy controls. The player must aim and shoot at fast opponents while simultaneously moving. Combining hard controls with rapid game action is a challenge for game designers on any platform. Player experience in FPS games is especially sensitive to the performance of the pointing device used such as keyboard, mouse, joystick, gamepad, or touchscreen. Touch-based controls for FPS games may not be a great match and there are few studies on the evaluation of touch input methods used in mobile games.

Within the scope of this project, we compare the current touch input methods used in mobile FPS games in terms of accuracy and speed in point selection tasks. In addition, we propose new input methods and compare them with existing methods. We prepared an experiment according to Fitts' Law and the ISO 9241-9 test to compare touch-based input methods. The 9.7-inch iPad Pro was used as the experiment device. We compared four input methods in an experiment, which took approximately one hour per participant. This allowed us to objectively compare the proposed methods to the existing methods. We found that one of the proposed methods, named Two-Finger Direct, was better than existing methods in terms of throughput. However, it also had the worst accuracy with the highest error rate. This suggests that the proposed method enabled users to input more data albeit with a reduced level of correctness, which may be preferable in the context of games.

CHAPTER 2: BACKGROUND

2.1 First-Person Shooter (FPS) Games

First-person shooter (FPS) is a genre of action game played from the hero's perspective. FPS games often map the user's inputs to the character's movements and simulate what a real person would see and do in the game. An FPS game usually shows the hero's arms at the bottom of the screen, often equipped with a weapon. A crosshair is shown at the center of the screen for aiming, which represents the forwards ray of the camera attached to the avatar's head. The user can rotate the avatar's head and move the avatar forward, backward, sideways, etc. using the game controller. As a result, the crosshair points to various locations in the game level. The character shoots through this crosshair when the user presses the fire button. This point and shoot mechanic makes shooting in FPS games a point selection task (see *Section 2.2 Point Selection Task*).

2.2 Point Selection Task

Point selection or target selection task is defined as the operation of selecting a graphic image that represents a target by controlling a cursor on the screen (MacKenzie, 2013). The point selection task was modelled after Fitts' Law, which is a predictive model that predicts that the time required to select a target is related to the distance to the target and the size of the target (Fitts, 1954; Fitts and Peterson, 1964).

For mouse or other indirect input devices, the task involves manipulating the device to move an on-screen cursor over a specified amplitude (A) to acquire a target of a specified width (W). Selection takes a final button press. For direct input, targets appear in the display and the input device operates directly on the targets.

Point selection task is often used to evaluate alternative devices or techniques and compare them to the mouse (MacKenzie, 2015). Shooting in FPS games can be thought of as a point selection task as it involves controlling a crosshair and hitting targets. This is similar to the multi-directional tapping task, which is a kind of point selection task that is a part of ISO 9241-9. Therefore, we use the multi-directional tapping task to compare input methods used in FPS games (see *2.5.1 Multi-Directional Tapping Task*).

2.3 Input Devices

An input device can be defined as an apparatus that provides information about the physical properties of humans, the environment, and things to a system (Hinckley and Wigdor, 2002). There is a variety of input devices with different properties. This variety requires special effort for designers to choose the suitable one that matches the demands of user and task requirements (Buxton, 2010). Several different input devices exist in terms of FPS games. The most popular of these are mouse, keyboard, gamepad, and touch screens.

2.3.1 Mouse and Keyboard

The mouse and keyboard duo are the most used input devices for FPS games. Mouse movement controls the viewpoint of the character while the directional keys or WASD keys of the keyboard control character movement. The left button of the mouse is usually used for shooting in FPS games.

2.3.2 Gamepad

Gamepad is usually used with game consoles like Xbox and PlayStation. Besides a gamepad can be connected to computers and mobile phones and can be used to control games similarly to consoles. Figure 1 shows a gamepad. In FPS games, the joystick on the left side of the gamepad is usually responsible for the character's movement similarly to the WASD keys on the keyboard. Another joystick on the right side of the gamepad controls the character's aim similarly to the mouse. Usually, the bumper or trigger button on the right side of the gamepad is used as the fire button.

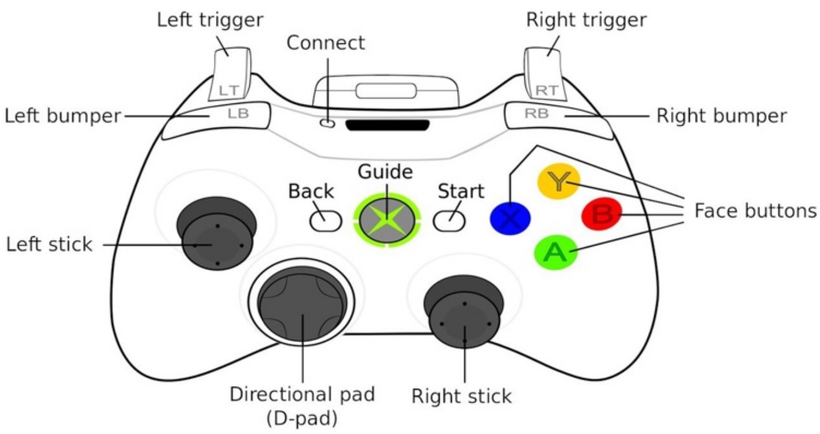


Figure 1. Gamepad

2.3.3 Touch Screen

A touch screen is both a display and an input device. It allows the user to interact with a computer or a mobile device by touching areas on the screen. Unlike using keyboard and mouse duo and a gamepad, the touch screen does not have a specific button location assigned for a function. It depends on the user interface of applications and games. When used for FPS games they usually simulate the gamepad with on-screen virtual joysticks. The visualization of buttons or indicators on the screen are a part of the game's user interface, which is often called a HUD (heads-up display). A sample game HUD from Call of Duty Mobile is shown in Figure 2. This game's HUD, which is inspired by the gamepad's logic, has a virtual joystick on the left side of the screen responsible for the character's movement while the right side of the screen controls the aim of the character. There is no visual button for this because the whole part of the right side of the screen except buttons is used for aiming. There is at least one fire button on the screen. The game HUD can usually be adjusted by the user in most such games.

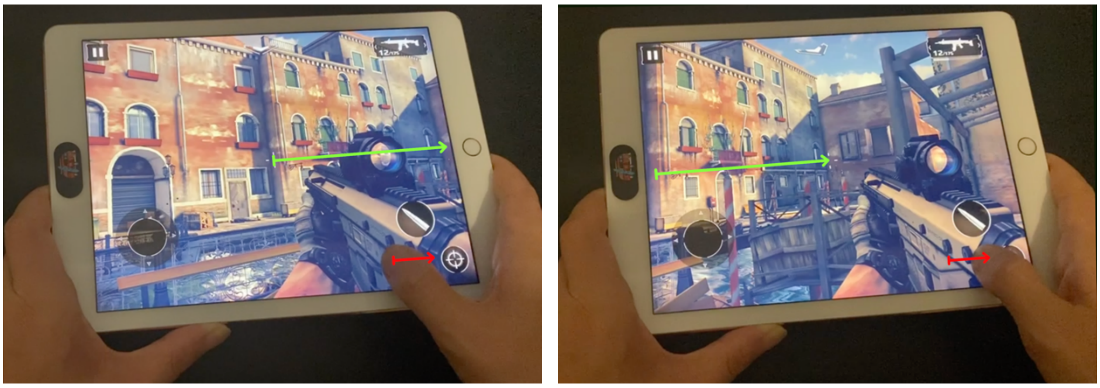
Figure 2. Game Hud of Call of Duty Mobile

A game HUD can be designed in any way by game producers. This can be thought of as each game producer can produce its own game controllers on a touch-based platform. Although the field provides so much freedom of development, the producers prefer to use the classic gamepad logic, which we mentioned in the previous paragraph, in most of the games in the industry. However, we see that using two different types of control for aiming when we analyse the touch-based FPS games. So far, there is no study on this in the literature. Thus, we have called them **position-based control** and **velocity-based control** and we define them here. Moving the character (walking) is out of this thesis scope because of that we study the aiming control and how to improve it.

2.3.3.1 Position-Based Control

First, recall aiming in FPS games, a crosshair is placed at the center of the screen. As the character rotates, the angle of view of the camera and therefore the location where the crosshair points out changes. As a matter of fact, it can be said that aiming is the rotation of character.

Position-based control refers to the control in which the amount of movement of the user's finger on the screen is proportional to the amount of rotation of the character. For instance, the character rotates 30 degrees when the finger moves 5 units horizontally on the screen. In Figure 3, two sample view from Modern Combat 5 is shown. The player uses position-based control and the aiming point at the moment the player starts dragging their finger is shown Figure 3.a. When the player drags their finger to the distance indicated by the red arrow, the change in aiming or the change in the character's rotation is indicated by the green arrow. Aiming will not occur unless the player drags their finger even if their finger touches the screen. These amounts and the proportion change according to games. In some games, the player can adjust these parameters named sensitivity in settings.



a. The aiming point when the drag starts

b. The aiming point when finished the drag

Figure 3. Sample Views from Modern Combat 5 (Position-Based Control)

2.3.3.2 Velocity-Based Control

The user manipulates a virtual joystick when velocity-based control is used. Pulling the virtual joystick applies some velocity to the character rotation. The velocity applied is change according to draw amount. The draw amount is calculated continuously according to the distance from the center. This simulates analog joysticks. No power is applied when the joystick is in the middle. The amount of power increases while it is pulled towards the edge, and maximum power is applied at the corners. A field is drawn on the game HUD to indicate these corners. Figure 4, shows the HUD from Modern Combat 5.

Figure 4. Sample View from Modern Combat 5 (Velocity-Based Control)

In this thesis scope, the touch screen is selected as an input device for FPS games, and we will examine different touch-based input methods, details in *CHAPTER 4: INPUT METHODS.*

2.4 Evaluating Input Devices

The 1980s gave birth to the field of Human-Computer Interaction (HCI) with the widespread use of computers in the workplace and homes. Previously, computers have been in locked rooms in university research labs or corporate or government facilities. Humans who had access to these computers were either engineers or computer scientists. Interaction with these computers was not an issue as they were experts. However, the interaction became an issue as personal use of computers increased with the launch of Xerox Star (1981) and the Apple Macintosh (1984) in the 1980s (MacKenzie, 2013). To use a computer, it was not necessary to be an expert or a scientist anymore. Thus, the field of HCI has emerged. This field examines the design of computer technology and the interaction between humans and computers in terms of usability. Computer technologies began to be evaluated from the point of view of their users, regardless of their designers. To this day, successful computer products are developed with this approach.

2.5 ISO 9241-9

Improvements in the field of Human-Computer Interaction have led to the writing of ISO standards for evaluating input devices. ISO 9241 is the full standard called Ergonomic design for office work with visual display terminals (VDTs), included seventeen parts. The Part-9 called Requirements for non-keyboard input devices has published in 2000. An updated version was released in 2012 as ISO/TC 9241-411.

ISO 9241 specifies methods of determining conformity by measuring the performance and physical properties of various devices by observation. ISO 9241-9 provides requirements and recommendations for the design of non-keyboard input devices. Includes only devices with sufficient published ergonomics information. ISO 9241-9 applies to a variety of non-keyboard input devices designed for fixed use. Mice, discs, joysticks, trackballs, tablets and skins, touchscreens, styli, and light pens provide ergonomic considerations-based guidance for input devices. It guides the design of these devices used for typical office tasks so that the limitations and capabilities of the users are taken into account (MacKenzie, 2018).

The ISO 9241-9 standard is also for the evaluation of non-keyboard input devices. It proposes a standardized methodology for evaluating performance and comfort. Performance is evaluated against one of six tasks and measured in terms of efficiency based on Fitts' Performance Index (MacKenzie, 1992). These six tasks are one-direction (horizontal) tapping, multi-directional tapping (two-directional), dragging, free-hand tracing (drawing), free-hand input (hand-written characters or pictures) and grasp and park (homing/device switching). In addition, the ISO standard presents a questionnaire that helps in measuring participants' device comfort. This questionnaire will be present in section 4.5.

2.5.1 Multi-Directional Tapping Task

Multi-directional tapping or 2D task is one of the six tasks described in the ISO 9241-9 standard (MacKenzie, 2018). This task is a point selection task (see *section 2.2 Point*

Selection Task). There is a sequence of targets placed in circular order and the highlighted target is selected via a pointing device, which is a touch screen in this study. After selecting the highlighted target, the opposite target is highlighted. This process continues in clockwise order until all targets are highlighted. The multi-directional tapping task is over when each target is highlighted once.

For the multi-directional tapping task, the evaluation measure is ***Throughput (TP)*** which is based on Fitts' Performance Index. Throughput is the amount of information transmitted per second through the user of the pointing device (MacKenzie, 2018). We explain how to calculate Throughput in *section 2.5.2 Throughput.*

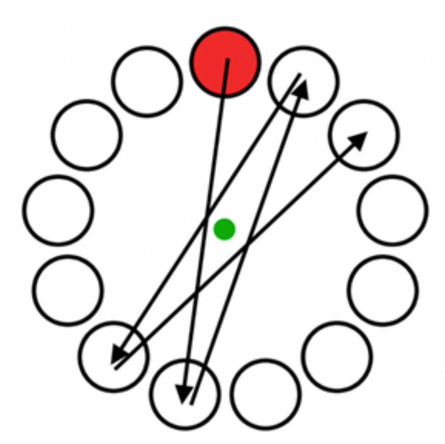


Figure 5. Multi-Directional Task

The selection of one target is called a ***trial*** (see Table 1). The selection of all targets placed in a circular order is called a ***sequence of trials*** (see Table 1). A sequence of trials contains a specific number of targets that is determined as a part of the experiment design. We chose to use 13 targets in each sequence of trials in this experiment (see Figure 5).

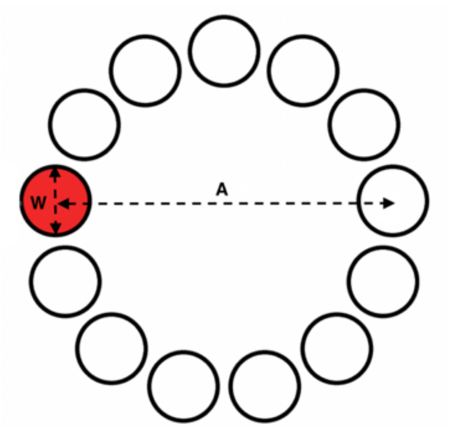
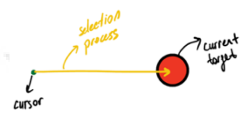
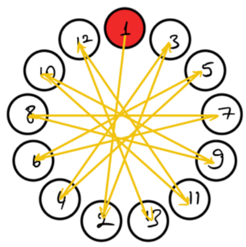
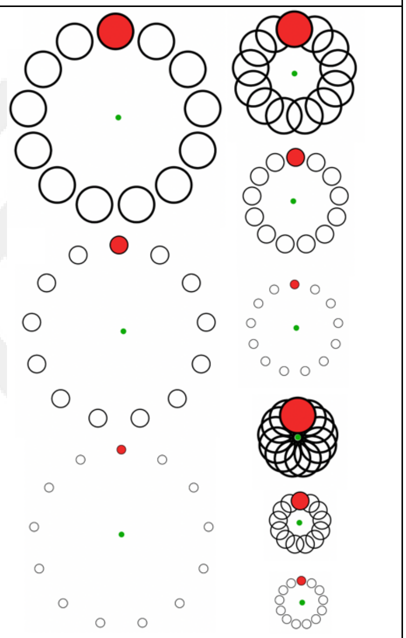


Figure 6. Width and Amplitude

In the multi-directional tapping task, the width of a target and distance between two opposite targets is significant as they are used to calculate throughput. The distance between two opposite targets is called amplitude. In Figure 6, width and amplitude are shown. A sequence of trials contains targets with specific amplitude and width. An experiment includes sequences with numerous combinations of width and amplitude. The more different combinations there are, the more inclusive the experiment will be (MacKenzie, 2018). The number of widths and amplitudes and their values is determined at the beginning of the experiment. There is a sequence of trials for each width and amplitude condition. The set of all width and amplitude combination is called a ***block*** (see Table 1). For instance, in this experiment, a sequence of trials consists of hitting 13 targets, and since there are 9 combinations of width and amplitude (see *section 5.4 Design),* a block consists of 9 sequences of trials. It is recommended to have repetitions of the same block in the experiment as more data will be collected and thus more consistent throughput will be obtained (MacKenzie, 2013). We explain the design of our experiment in *section 5.4 Design,* but we visualized the processes performed by a participant for one device in the experiment in Figure 7.

The concepts of a trial, sequence of trials, and a block are significant to understanding ISO 9241-9 test and calculating throughput. Table 1 contains the definition and visualization of these concepts.

Table 1. Major Concepts About Multi-Directional Tapping Task

<table>
<tr><th></th><th>Trial</th><th>Sequence of Trials</th><th>Block</th></tr>
<tr><td>Definition</td><td>A trial is an operation of selecting a single target.</td><td>A sequence of trials refers to the selection process of all targets in a specific number placed in a circular pattern. The number is 13 in this experiment.</td><td>Block is a set of "sequences of trials" containing all the width and amplitude combinations determined at the beginning of the experiment. A block includes 9 sequences of trials in this study. *</td></tr>
<tr><td>Visual</td><td>
</td><td>
</td><td></td></tr>
<tr><td colspan="4">* The width and amplitude numbers were determined as 3 in the experiment. These are 264, 132, 66 pixels for width and 1232, 616, 308 pixels for amplitudes (see section 5.4 Design).</td></tr>
</table>

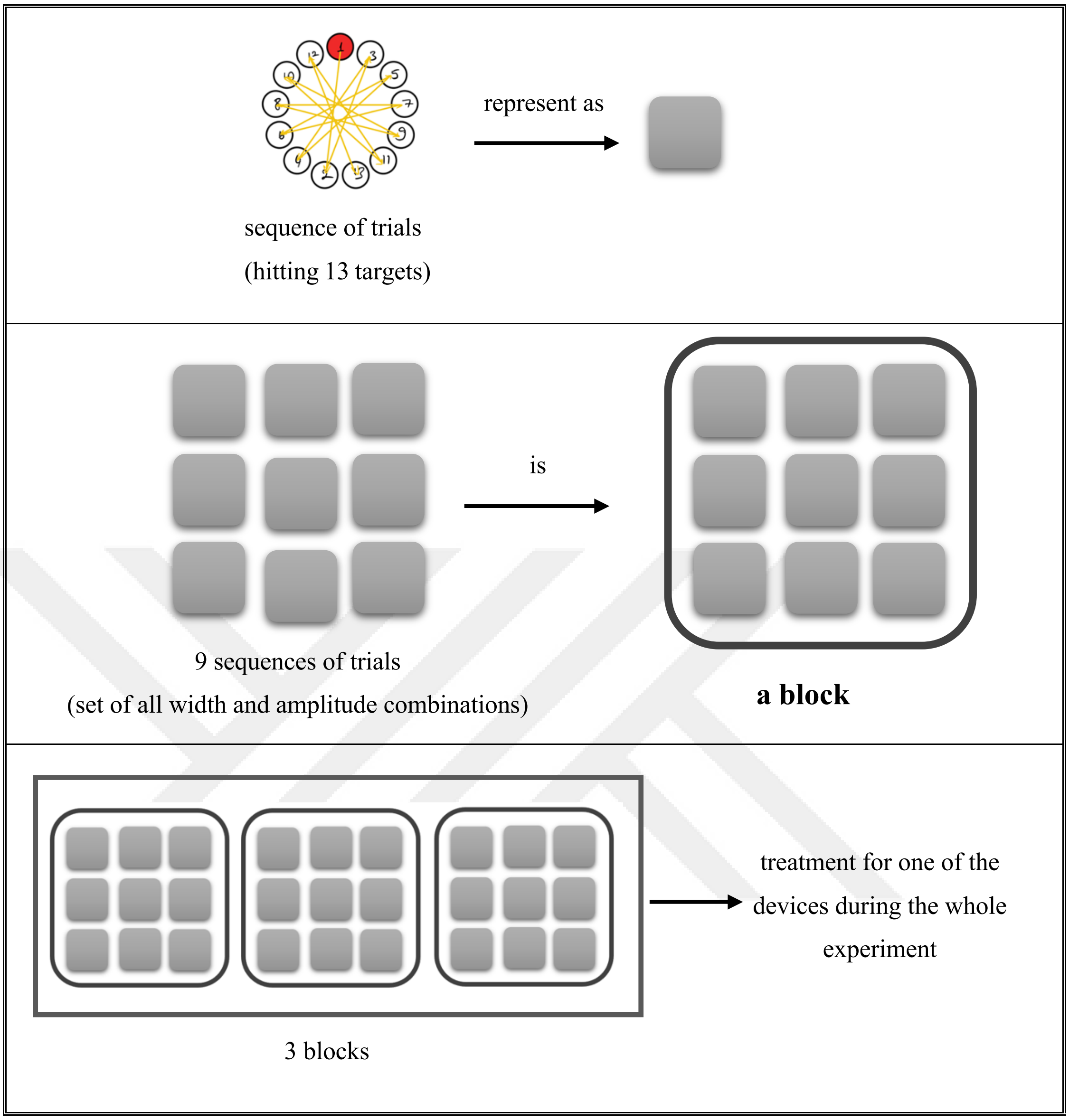


Figure 7. The Visualization of All Processes for One Device

2.5.2 Throughput

Throughput is the amount of information transmitted per second through the user of the pointing device. It is based on Fitts' Index of Performance. Fitts has described a formula that measures human performance in target acquisition tasks (Fitts, 1954). Over the years, Fitts' offers were developed and evolved into the throughput formula currently used in the ISO standard. There are still studies about how to make enhancements on the calculation of throughput in the literature. However, we calculated it as described in Mackenzie's paper published in 2015 which is used in the ISO standard (MacKenzie, 2015).

According to this, throughput (TP) is

$$TP = ID_e/MT \tag{1}$$

where ID_e is a task's effective *index of difficulty* (in bits) computed from the movement amplitude *(A)* and target width *(W)* and *MT* is the mean movement time (in seconds) recorded over a sequence of trials. A and W are shown in Figure 6. The ID_e-term in Eq. 1 expands as follows:

$$ID_e = log_2(A_e / W_e + 1) \tag{2}$$

The effective values (subscript "e") is used to include actual participant behaviors. With this, W_e is computed as $4.133 \times SD_x$, where SD_x is the standard deviation in the selection coordinates and A_e is the mean of the actual movement amplitudes in the sequence of trials.

$$W_e = 4.133 \times SD_x \tag{3}$$

Considering this adjustment, it is said that throughput is a single measure of human performance that includes both speed and accuracy in human responses.

CHAPTER 3: LITERATURE REVIEW

Touch-based controls for FPS games are often used in mobile games. However, we were not able to find any empirical comparison of touch-based controls for FPS games in the literature. This study is a first attempt to provide such a comparison. In the next subsections, we highlight research that explain how to carry out an ISO 9241-9 test for evaluation. Then we summarize some of the research relevant to the evaluation of pointing devices, regardless of their adherence to ISO standards or lack thereof.

3.1 Applying ISO 9241-9 Standards for Evaluation

The ISO 9241-9 standard is for the evaluation of non-keyboard input devices (Smith, 1996). It proposes a standardized methodology for evaluating performance and comfort. Performance is evaluated against one of six tasks and measured in terms of efficiency based on Fitts' Performance Index (MacKenzie, 1992). These tasks are one-direction (horizontal) tapping, multi-directional tapping, dragging, free-hand tracing (drawing), free-hand input (hand-written characters or pictures), and grasp and park (homing/device switching). The primary ISO dependent measure is throughput.

The scientific validity and applicability of ISO 9241-9 are evaluated by Douglas et al. (1999). An experiment was conducted to apply the performance and comfort elements of the ISO testing by comparing a touchpad and a finger-controlled isometric joystick. Participants were asked to perform a point selection task and then rate device comfort in a questionnaire. Throughput was calculated for each device.

Research by MacKenzie (2015) detailed the method of calculating throughput which is the dependent measure of ISO 9241-9 regarding task characteristics and data collection. An experiment was conducted to elaborate on the distinction between indirect and direct pointing devices. He used the examples of a mouse as an indirect pointing device and a finger as a direct pointing device. Throughput and error rate was calculated for each device.

Another study by MacKenzie (2018) described how Fitts' Law has been improved for use in the evaluation of input devices. He used a subset of his experiment conducted in 2015 as the user study. This experiment investigated touch-based target selection on two task types as 1D and 2D. Throughput was calculated for each task type.

3.2 Evaluation of Pointing Devices

3.2.1 Research Using the ISO 92419 standard

Natapov et al. (2009) evaluated user performances for the pointing task using ISO 9141-9 with Nintendo Wii Remote and Classic Controller. A standard mouse was used as the baseline condition. Both console controllers were reported to have lower performance, speed and error rates compared to mouse.

Natapov and MacKenzie (2010) used ISO 9241-9 to evaluate a prototype game controller made by them for the point selection task. The prototype game controller had been created by replacing the right analog stick of a standard game controller (used for pointing and camera control) with a trackball. They compared the prototype game controller to a standard game controller by using two groups of participants (novice and advanced). They informed the result that the trackball controller's throughput was 2.69 bps while the standard controller's throughput is 1.68 bps for the novice group. In the advanced group, the trackball controller's throughput was 3.19 bps while the standard controller's throughput is 2.01 bps.

Pino et al. (2013) evaluated the performance of computer-based 2D and 3D pointing tasks. They applied ISO 9241-9 standard methodology for 2D and for the 3D experiments they presented a novel experiment layout, supplementing the ISO standard. They compared a Microsoft Kinect device and a mouse for the 2D and 3D tasks. Though throughput was calculated for each case, they calculated also new metrics which as Target Re-Entry and Missed Click.

Zaranek et al. (2014) applied a design that measures the targeting performance of several modern game input devices in their research. These are designated as a mouse, a game controller, PS Move and Kinect. A 3D first-person shooting game mission based on the ISO 9241-9 experimental paradigm was used to evaluate pointing devices in the research. Comparison of performance measures showed that the mouse was the best. It has been determined that the performance of 3D input devices (Move and Kinect) is very bad compared to the mouse.

Ramcharitar and Teather (2017) evaluated three different game input methods offered by the Steam Controller such as a thumb-based touchpad, a thumbstick, and a gyrosensor. They compared these three input methods and the mouse (as a baseline condition) using ISO 9241-9. The best throughput at 4.73 bps is belong to the mouse.

This is followed by the touchpad at 2.98 bps, then the gyrosensor at 2.85 bps, and finally the thumbstick at 2.39 bps. They say this shows that despite the prevalence of the thumbstick in modern game controllers, the touchpad and gyro sensor are good alternatives to the traditional thumbstick.

Magee et al. (2015) used ISO 9241-9 to evaluate the Camera Mouse configured with two selection methods. They compared dwell-time click generation and detecting a single intentional muscle contraction with an attached sensor (ClickerAID). ClickerAID generates button events by detecting an intentional muscle contraction from a piezoelectric sensor that contacts the user's skin. The sensor was placed under a headband in contact with the wearer's eyebrow muscle. A traditional laptop touchpad was used for baseline condition. In a ten-subject user study, outputs are specified as 2.10 bps (touchpad), 1.28 bps (Camera Mouse with dwell time selection), and 1.43 bps (Camera Mouse with ClickerAID).

Hassan et al. (2019) used ISO 9241-9 for a similar study. They evaluate hands-free and hands-on point selection tasks on a laptop computer. They used face tracking software called Camera Mouse. This was compared with three hands-on methods, a touchpad with dwell time selection, a touchpad with touch selection, and face tracking with touch selection. The throughput for hands-free entry was 0.65 bps. The highest efficiency was found for the touch selectable touchpad (2.30 bps). The hands-free state has been reported to show erratic cursor control with frequent target re-entries before selection, especially for cooldown selection.

3.2.2 Research Not Using the ISO 92419 standard

Klochek and MacKenzie (2006) compared an Xbox gamepad to a standard PC mouse in constrained 3D environments, similar to first-person genre games. They presented five new performance metrics which are Mean Speed Variance, Mean Acceleration Variance, Percent View Moving, Target Leading Analysis, and Mean Time-to-Reacquire.

In the study of Farmani and Teather (2017), player performance was evaluated with different input devices identified as a Razer Hydra, an Xbox gamepad, and a mouse. They used a custom VR FPS game. They used Fitts' law to measure movement time. They also calculated the number of missing targets as the error rate during testing. According to the results of the analysis, it was stated that Razer Hydra was the slowest and the mouse was the fastest. It was observed that the movement time increased since it was more difficult to hit small targets according to Fitts's law. It was stated that the research results were not efficient because Razer Hydra had the highest error rate.

CHAPTER 4: INPUT METHODS

In this study, four touch-based input methods for the point selection task were compared. Two of them are existing methods which uses in many popular fps games, we called them *Position-Button* and *Velocity-Screen*, we will explain these in this chapter. The other two methods are new methods proposed by us, we called them *Two-Finger Indirect* and *Two-Finger Direct*.

4.1 Existing Methods

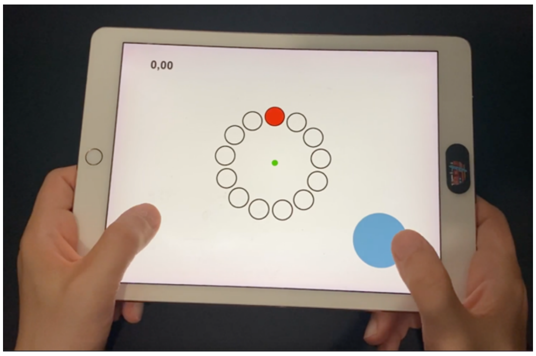
There are two existing methods, both use the thumbs of both hands. Also, the sides of the screen are used for aiming or shooting. In this experiment, we adjust the side according to the participants' dominant hand. Thus, the dominant hand side is used for shooting and the non-dominant hand side is used to move the cursor. We refer to the dominant hand below as the right hand for simplicity.

4.1.1 Position-Button Input Method

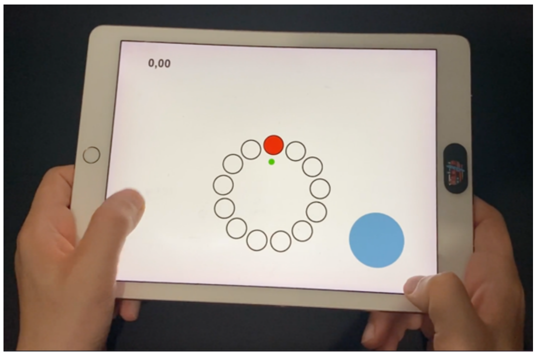
In the position-button input method both thumbs are used. Cursor movement is indirect position-based. The change in the left thumb position is applied to the cursor position whereas lifting up and bringing down the finger has no effect on the cursor (see *section 2.3.3* Touch Screen). Shooting is performed by touching a button on the bottom right of the screen.

Figure 8.a. shows the way the device is held. The left thumb is used to control the movement of the cursor by dragging on the whole screen. The right thumb is used to touch the blue button on the bottom-right of the screen to shoot the target that the cursor is on.

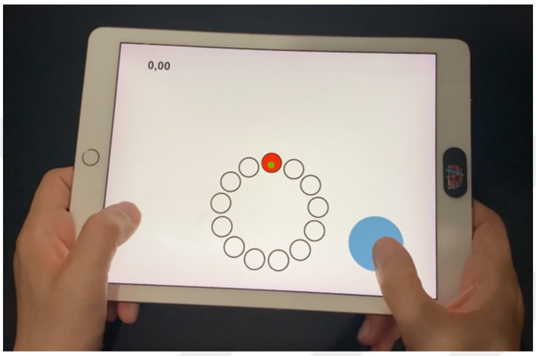
In Figure 8.b., the user moved the cursor up with the left thumb. In Figure 8.c. the user touched the blue button with the right thumb to shoot the red target that the cursor is on. Figure 8.d. shows the next target becoming active.



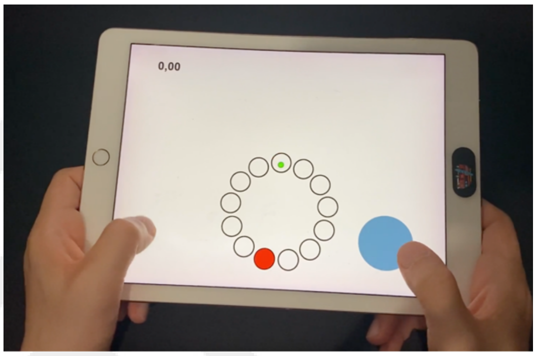
a. Holding position



b. Moving of the cursor with left hand’s thumb



c. Shooting by touch the button with right hand’s thumb



d. After shooting the target

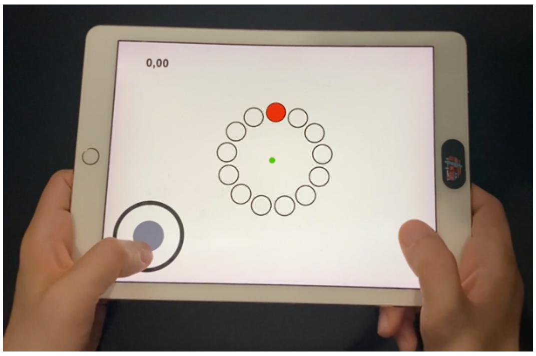
Figure 8. The Case of the Position-Button

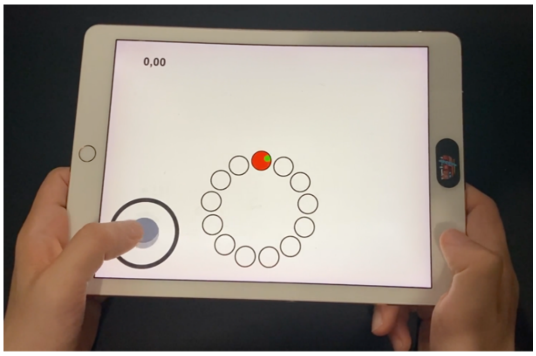
4.1.2 Velocity-Screen Input Method

In the velocity-screen input method, cursor movement is velocity-based. Left thumb hold and drag provides a velocity vector for the cursor to continuously move with. (see *section 2.3.3* Touch Screen). Shooting is performed by touching the right hand side of the screen.

Figure 9.a. shows the way the device is held. The left thumb provides velocity for the cursor by controlling a virtual joystick on the left bottom of the screen. The right thumb is used to shoot by tapping any place on the right side of the screen.

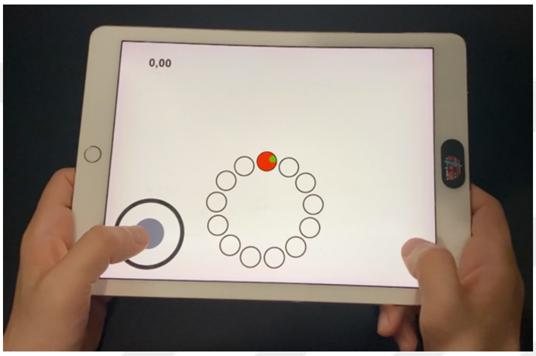
In Figure 9.b. the participant drags up the grey button of the virtual joystick, which applies an upward velocity to the cursor. The speed depends on the distance that the grey button is dragged to. The maximum speed is reached when the grey button is on the black border. In Figure 9.c. the participant shoots the red target that the cursor is on by tapping the screen with the right thumb. Figure 9.d. shows the next target becoming active.

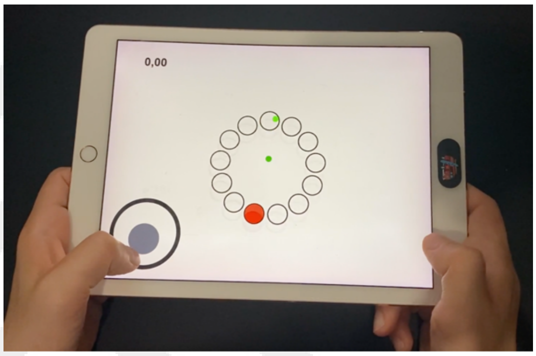




a. Holding position

b. Moving the cursor by controlling the virtual joystick with the left thumb





c. Shooting by tapping the screen with the right thumb

d. After shooting the target

Figure 9. The Case of the Velocity-Screen

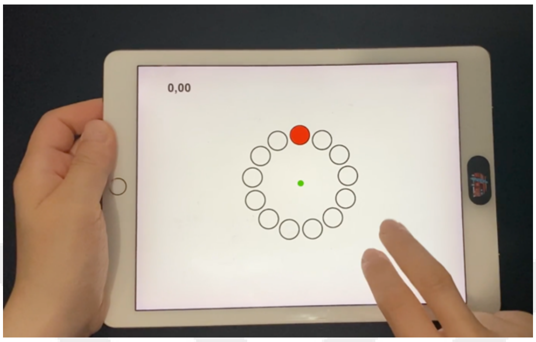
4.2 New Proposed Methods

Studying the existing methods, we observed that position input is better suited to this task rather than velocity input. Furthermore, we observed that aiming and shooting being done by different hands may be introducing some difficulty. We believed that if position-based input and shooting is done by the same hand, aiming would become more automatic and faster. We propose two new methods based on this idea. With these methods, our goal is to give the user more control over the mechanics and get more accurate results in aiming and hitting the right target.

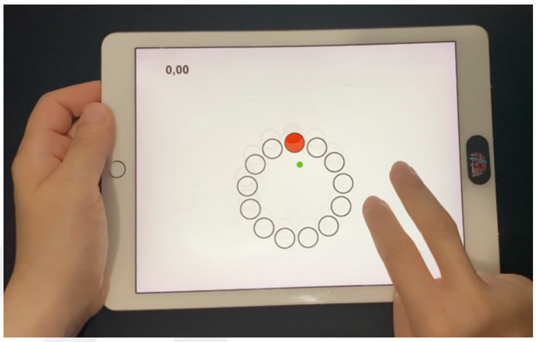
4.2.1 Two-Finger Indirect

In the two-finger indirect input method, the user uses the right index and middle fingers. Cursor movement is indirect position-based. The change in the right index finger position is applied to the cursor position whereas lifting up and bringing down the finger has no effect on the cursor (see *section 2.3.3* Touch Screen). Shooting is performed by tapping the screen with the right middle finger without releasing the right index finger.

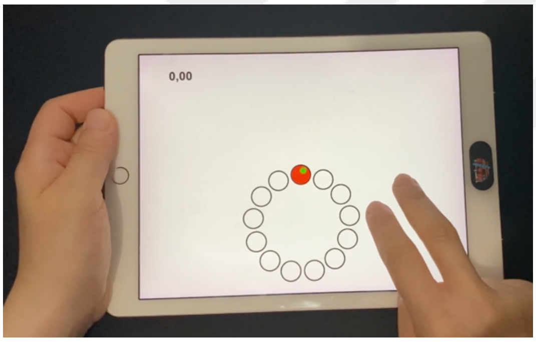
Figure 10.a. shows the way the device is held. The index finger is used to control the movement of the cursor by dragging on the whole screen. In Figure 10.b. the user has dragged the index finger up to move the cursor up. The middle finger is used to shoot by tapping any place on the screen while holding the index finger down. Figure 10.c. shows this shooting movement. Figure 10.d. shows the next target becoming active.



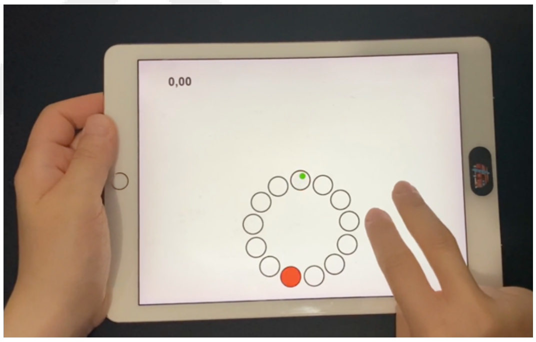
a. Holding position



b. Moving the cursor by dragging the right hand's index finger



c. Shooting by tapping the screen with the right middle finger while the index finger remains on the screen



d. After shooting the target

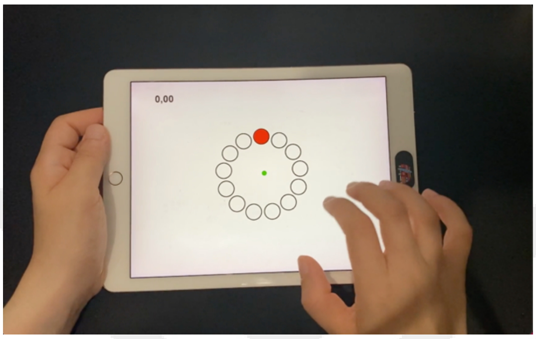
Figure 10. The Case of the Two-Finger Indirect

4.2.2 Two-Finger Direct

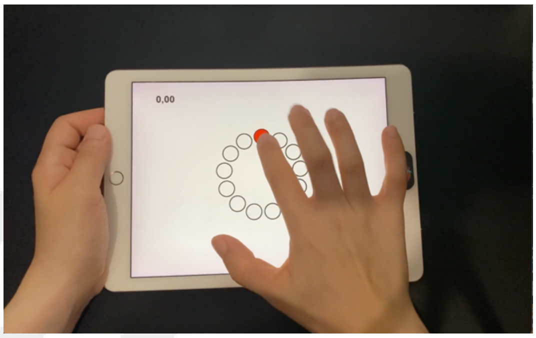
The two-finger direct input method is very similar to the two-finger indirect. Two adjacent fingers of the right hand are used, cursor movement is position-based, and shooting is performed by touching the screen.

In this method, the major difference is that the location of the cursor is mapped directly to the location of the right index finger, slightly higher to avoid the finger from blocking its view. Lifting and bringing down the finger brings the cursor to the place where the finger is. Since the screen and the input are in the same device, we can directly touch the target and the cursor comes right there. Naturally, dragging the index finger on the screen drags the cursor with it. The middle finger is used to shoot by tapping anywhere on the screen, as with the two-finger indirect input method.

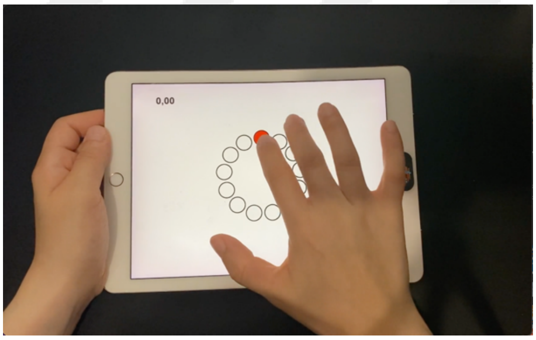
Figure 11.a. shows the way the device is held. In Figure 11.b. the user touches the index finger on the target to bring the cursor there. In Figure 11.c. the user shoots by tapping the middle finger while the index finger remains on the screen.



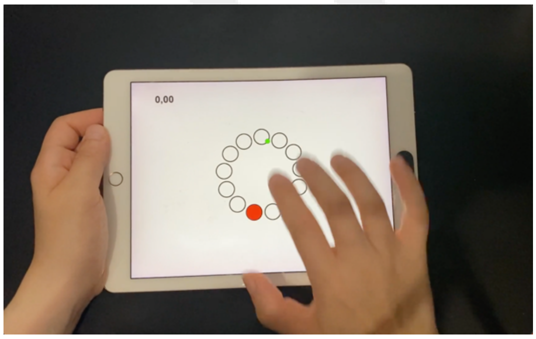
a. Holding position



b. Repositioning the cursor by tapping the desired target on the screen with the right index finger



c. Shooting by tapping the screen with the right middle finger while the index finger remains on the screen



d. After shooting the target

Figure 11. The Case of the Two-Finger Direct

CHAPTER 5: METHOD

In this study, four touch-based input methods for the point selection task were compered. The point selection task is defined as a task in which a user is asked to hit a highlighted target by controlling a cursor (see *section 2.2 Point Selection Task*).

There are two existing touch-based input methods (***position-button*** and ***velocity-screen***) for point selection task that are already in use in existing mobile first-person shooter games. In the first, position-button method, aiming is done with indirect position-based input and firing is done by touching a button. In the second method, velocity-screen, aiming is done with velocity-based input and firing is done by touching the screen. We have explained these in more detail in *section 4.1 Existing Methods*.

In addition to these, we propose two more input methods. First is **"two-finger indirect"**. In this method, the user uses two fingers; the index finger is used for aiming and the middle finger is used for firing. Second is **"two-finger direct"**. It is similar to a two-finger indirect method, just one difference is that aiming is done by direct touching the place where you want to aim, no need to move the cursor like in other methods. The middle-finger is still used to fire. We have explained these in more detail in *section 4.2 New Proposed Methods*.

5.1 Participants

Twelve paid participants (eight female, four male) were recruited from local community and the local university campus. One participant was left-handed. The experiment was adjusted according to the dominant hand of the participant. Participant ages ranged from 21 to 29 (mean = 24,9).

Before the experiment, a questionnaire was applied to all the participants in order to understand their gaming habits. All of the participants stated that they played games with touch devices such as phones and tablets at some point in their lives. It was determined that 6 of the participants played PUBG in the past. When asked how often they played FPS type games, 9 of the participants stated that they played in the past. When asked about platform preferences for FPS, 5 people answered computer, 1 person all (PC/Mobile Devices/Consoles), 1 person phone. The participants consist of people who applied for the call made within the university. Therefore, the distribution of participants was random. We believe that this group of participants creates a good variety for comparing case types.

5.2 Apparatuses

5.2.1 Hardware

The experiment is conducted using a 9.7-inch iPad Pro with a resolution of 1536 x 2048 running iOS version 13.4. The touch sample rate is 120 hz.

Participants sat in an armchair and were free to hold the iPad in their hands or to place it on a desk. This sitting arrangement simulates a typical home entertainment environment. The state of a participant performing the experiment was shown in Figure 5.

Figure 12. The State of a Participant Performing the Experiment.

5.2.2 Software

The experiment software was written in Unity - C# and run on the iPad as an app. The whole experiment is touch-based. The app contains the serial 2D tasks commonly used in ISO 9241-9 experiments. The same set of width and amplitude combinations were used for all input methods. In all, nine combinations were used: $A = \{1232, 616, 308\}$ px $\times$ $W = \{264, 132, 66\}$ px (see *sections 2.5.1 Multi-Directional Tapping Task* and *2.5.2 Throughput*)

The 2D conditions include 13 targets, which was the number of trials in a sequence. The target to select is highlighted in red. After the selection attempt, the next target is highlighted, which is opposite to the current one in the wheel. Selections proceed in a rotating pattern around the layout circle until all targets are selected. Selection operations should be done as fast and accurately as possible. This procedure for the first five targets is shown in Figure 6.

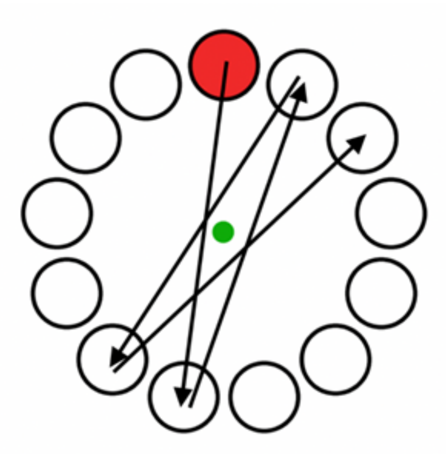


Figure 13. The first five trials of a Sequence

Data is collected for each sequence, which begins on the first tap and ends after 13 target selections. The duration between clicking two the targets is used as the duration of one trial. We also logged the total duration of one sequence of trials, including the 13 targets. Additionally, we logged the coordinates of the tapped coordinate and its distance along the task axis from the center of the target. Finally, we logged whether the tap was a hit (inside the target) or a miss (outside the target).

5.3 Procedure

The experiment began with a brief explanation of the aim of the experiment. The participants were asked to perform a serial target selection task on the iPad for the four cases (Position-Button, Velocity-Screen, Two-Finger Indirect, Two-Finger Direct) and to be as fast and accurate as possible. Each case was explained with text on the screen and by watching a video of a person performing the case. Then, one sequence of trials (hitting 13 targets, see *section 2.5.1 Multi-Directional Tapping Task*) was done as a practice session for the current case. The result of this practice session was not included the analysis. After this practice session, the experiment started for the current case.

Each participants perform three blocks of trials for each input methods. Each block consisted of nine Index of Difficulty (ID) levels, for more detail about calculation ID see *section 2.5.2 Throughput*. The nine *IDs* were derived from different target distance and width combinations. There were three target widths (66, 132, and 264 pixels), and three target distances (amplitude) (308, 616 and 1232 pixels). These values were chosen based on the size and resolution of the hardware used. The largest width was determined as the number of pixels corresponding to an inch on the iPad used

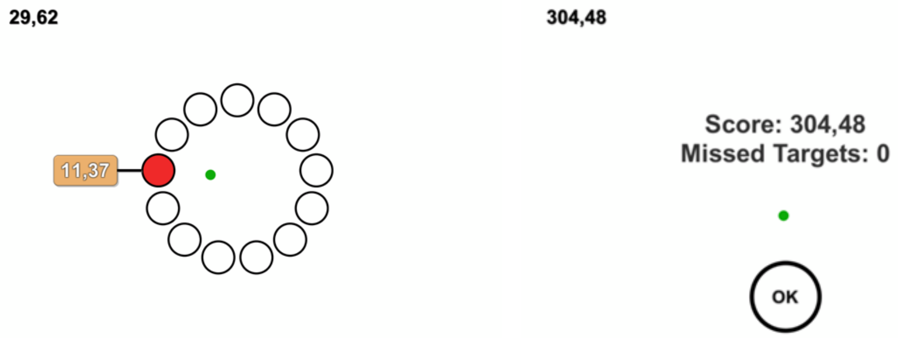
(W=264). The distance between the top and bottom points of the screen was chosen as the widest amplitude when the gap between the edge of the screen and target is 20 px, W is 264 px and the screen width of the iPad is 1536 px. The resulting IDs used in this experiment are given in Table *2*.

Table 2. Index of Difficulty (IDs) values used in the experiment

Amplitude (pixels)	Width (pixels)	ID (bits)
1232	264	2.50
1232	132	3.37
1232	66	4.30
616	264	1.74
616	132	2.50
616	66	3.37
308	264	1.12
308	132	1.74
308	66	2.50

In the experiment, an auditory beep sound indicates that a target was missed. At the end of each sequence, a score appears showing the user performance of the sequence. The score refers to how fast the targets are hit. After the first hit, a countdown timer runs for the current target. When the participants hit the target, the remaining time is added to the score. If the target is missed, there is nothing to add to the score. Thus, the faster the participant shoots, the higher the score. This score is not valuable for the experiments. The score screen is shown in Figure 14. It aims to attract the participant's attention and to entice them to act quickly, like playing a game.

After three blocks, task of one case has finished and participants has been asked to fill out a questionnaire to capture their experience on the task of the case. The order of these four cases is defined according to Latin Squares. The whole experiment session took approximately one hour per participant.



a. A snippet when the countdown timer running

b. The score screen at end of the sequence of trials

Figure 14. The Score

5.4 Design

The experiment is employed within-subjects with the following independent variables and levels:

Input methods	: Position-Button, Velocity-Screen, Two-Finger Indirect, Two-Finger Direct
Task type	: 2D
Blocks	: 1, 2, 3
Amplitude	: 1232, 616, 308
Width	: 264, 132, 66

The primary independent variable is input method. Blocks, amplitude, and width is included to gather a sufficient quantity of data over a reasonable range of task difficulty. There are two dependent variables as *throughput* and *error rate.*

For each method, participants perform a sequence of 13 trials in 3 blocks. The 4 input methods conditions are assigned using balanced Latin Squares with 4 participants per order. The amplitude and width conditions are randomized within blocks. Thus, there are 4 input methods x 3 blocks x 3 amplitude x 3 width x 13 trials = *1404 trials* per participant.

5.5 Data

Before starting the experiment, each participant was asked questions to understand their relationships with FPS games. There were eight open-ended questions as follows:

1. *How old are you? What's your job?*
2. *Do you play games?*
3. *How often do you play computer games?*

4. *How often do you play games on your phone/tablet (touchscreen devices)?*
5. *How often do you play FPS type games?*
6. *Which is your preferred platform for FPS/TPS (aiming) games?*
7. *Do you play PUBG, if yes how often do you play it?*
8. *How often do you play FPS/TPS (aiming) games on phone/tablet (touchscreen devices)?*

After each case, participants rated the device using questions from ISO 9241-9 on device comfort in a way similar to other relevant studies (Douglas, Kirkpatrick and MacKenzie, 1999; Natapov, Castellucci and MacKenzie, 1999). There were nine questions, each with a rating from 1 to 5, as follows:

1. *The force required for actuation was*
   (1: too low – 5: too high)
2. *Smoothness during operation was*
   (1: very rough – 5: very smooth)
3. *The mental effort required for operation was*
   (1: too low – 5: too high)
4. *Accurate pointing was*
   (1: easy – 5: difficult)
5. *Operation speed was*
   (1: too fast – 5: too slow)
6. *Finger fatigue*
   (1: none – 5: very high)
7. *Wrist fatigue*
   (1: none – 5: very high)
8. *General comfort*
   (1: very uncomfortable – 5: very comfortable)
9. *Overall the input method was*
   (1: very difficult to use – 5: very easy to use)

After performing all four cases and completing the experiment, participants were given one final instruction:

> *"Please rate the control types in the order you would prefer to use them for pointing tasks. Please explain your decision to rate them this way, and*

comment on what you liked and disliked about each device. Feel free to add any additional comments."

CHAPTER 6: EXPERIMENTAL RESULTS

6.1 Throughput

Throughput was used to compare the four touch-based input methods, as described in ISO 9241-9 (see *section 2.5.2 Throughput*). Throughput is measured in bps (bit per second) and used as a quantitative measure for comparing input methods' performance. Throughput was calculated for each input method on a per participants basis.

The results of the comparison of the throughput values according to the case types are given in Table 3. Accordingly, it was determined that the throughput values differed statistically according to the case types ($p<0.05$).

Table 3. Throughput Values According to Case Types

Case Types	N	Mean	Std.	Min.	Max.	p*
Position Button	324	1.87	.99	.02	4.07	0.000
Velocity Screen	324	1.01	.43	.01	2.40	
Two Finger Indirect	324	1.73	.95	.03	4.01	
Two Finger Direct	324	2.33	1.41	.02	5.98	

**p<0.05 statistical difference. Anova Test*

Post-hoc analysis results of the comparison of throughput values according to case types are given in Table 4. It was found that there is a statistical difference between the throughput values of Position-Button, Velocity-Screen and Two-Finger Direct ($p<0.05$). However, there was no statistically significant difference between Position-Button and Two-Finger Indirect throughput values ($p>0.05$). There is a statistical difference between the Velocity-Screen and Position-Button, Two-Finger Indirect and Two-Finger Direct throughput values ($p<0.05$). There is a statistical difference between Two-Finger Indirect and Velocity-Screen and Two-Finger Direct throughput values ($p<0.05$).

Table 4. Posthoc Analysis Results of Throughput Values According to Case Types

(I) case type	(J) case type	Mean Dif. (I-J)	Std. Error	Sig.
Position Button	Velocity Screen	.86483*	.06036	.000
	Two Finger Indirect	.14010	.07657	.344
	Two Finger Direct	-.45677*	.09606	.000
Velocity Screen	Position Button	-.86483*	.06036	.000
	Two Finger Indirect	-.72473*	.05831	.000
	Two Finger Direct	-1.32161*	.08224	.000
Two Finger Indirect	Position Button	-.14010	.07657	.344
	Velocity Screen	.72473*	.05831	.000
	Two Finger Direct	-.59687*	.09478	.000
Two Finger Direct	Position Button	.45677*	.09606	.000
	Velocity Screen	1.32161*	.08224	.000
	Two Finger Indirect	.59687*	.09478	.000

**p<0.05 statistical difference. Anova-Posthoc Test*

As seen from the results, the Two-Finger Direct method has the highest throughput value and is statistically different from the other 3 methods. The Velocity-Screen method has a lower throughput value. When we look at the Position-Button and Two-Finger Indirect, we see that the results are close to each other in terms of throughput and there is no statistical difference. In line with these results, we can say that the method that gives the best results is the Two-Finger Direct, and the Velocity-Screen method is the worst. Figure 15 shows throughputs by case type as a graph.

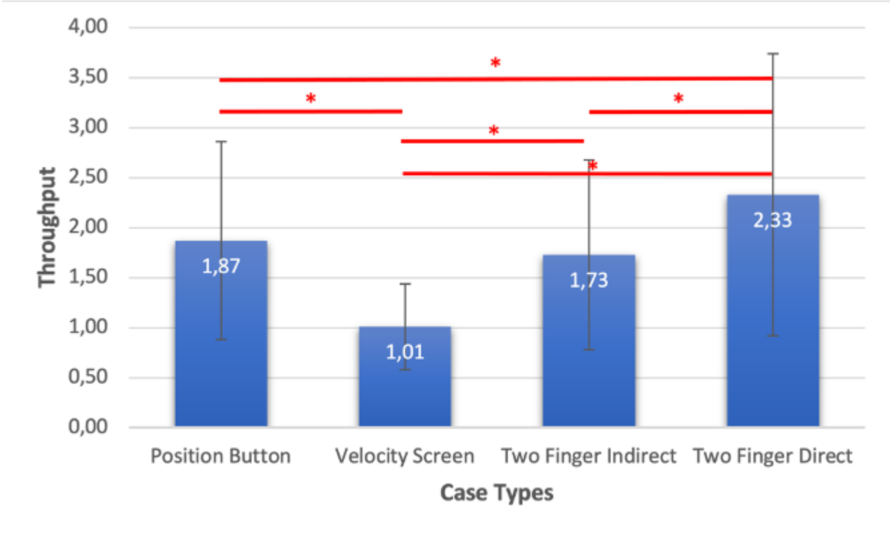


Figure 15. Throughput (bps) by case types

6.2 Error Rate

Throughput provides a good measurement includes how fast and accurately performs by participants for each input method overall. However, it does not represent how successful the task of hitting targets was. This information is significant regarding to evaluate an input method. Therefore, error rate or missed target has also analysed, even though error rate analyses are not required according to ISO 9241-9.

The results of the comparison of the error rate values according to the case types are given in Table 5. Accordingly, it was determined that the error rate values differed statistically according to the case types ($p<0.05$).

Table 5. Error Rate Values According to Case Types

	Error (-)	Error (+)	Total	p*
	n(%)	n(%)	n(%)	
Position Button	4128 (98.10)	80 (1.90)	4208 (100)	0,000
Velocity Screen	3573 (91.90)	316 (8.10)	3889 (100)	
Two Finger Indirect	3736(96.10)	152 (3.90)	3888 (100)	
Two Finger Direct	3568 (91.80)	320 (8.20)	3888 (100)	

**p<0.05 statistical difference. Chi-Square Test*

The method with the best accuracy, that is, the least error rate is the Position-Button method. There are two input methods with the highest error rate: the Velocity-Screen and the Two-Finger Direct methods. Figure 16 shows error rates by case type as a graph.

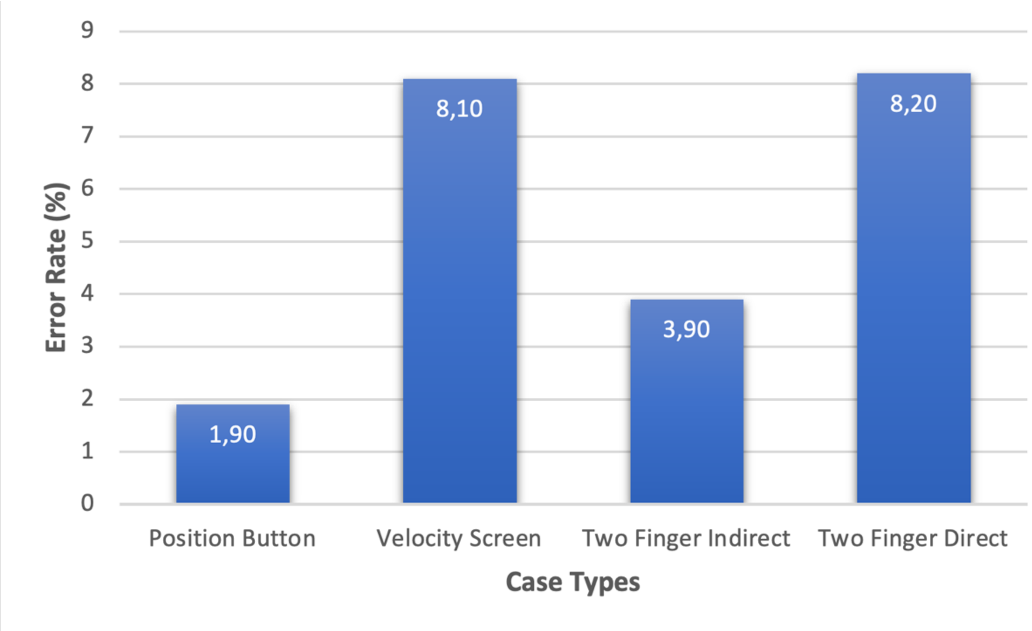


Figure 16. Error Rate by Case Types

6.3 Questionnaire

Participants' device comfort conditions according to case types of results are given in Table 6. The highest force required to actuation was found in the Velocity-Screen method, and the force levels were statistically different according to the case types ($p<0.05$). Smoothness during operation is higher in the Two-Finger Direct method than in others and smoothness during the operation was statistically different according to the case types ($p<0.05$). The mental effort required for operation is higher in the Velocity-Screen method than in others and mental effort levels were statistically different according to the case types ($p<0.05$). For accurate pointing, the Velocity-Screen method was found to be the most difficult, and accurate pointing was statistically different according to the case types ($p<0.05$). Operation speed, finger fatigue, and wrist fatigue did not differ according to case type ($p>0.05$). General comfort and overall input method levels were found to be the best in the Two-Finger Direct method ($p<0.05$).

Table 6. Participants' Device Comfort Conditions According to Case Types

	Position Button	Velocity Screen	Two Finger Indirect	Two Finger Direct	p*
	mean±std	mean±std	mean±std	mean±std	
Force required for actuation	2.33±0.98	3.17±0.00	2.33±1.30	1.33±0.65	0.005
Smoothness during operation	3.83±1.34	2.17±1.11	3.67±1.23	4.67±0.49	0.000
The mental effort required for operation	2.00±0.74	3.08±0.90	2.33±1.07	1.83±1.19	0.017
Accurate pointing	2.25±1.48	4.08±0.79	2.58±1.38	2.42±1.31	0.003
Operation speed	2.25±1.48	2.33±1.61	2.25±1.36	1.75±0.97	0.718
Finger fatigue	2.58±1.00	2.75±0.97	2.83±1.27	2.08±1.31	0.388
Wrist fatigue	2.33±1.44	3.08±1.56	2.50±1.62	1.83±1.40	0.256
General comfort	4.00±0.74	2.42±0.90	3.58±1.31	4.25±0.87	0.000
Overall the input method	4.33±0.89	2.25±0.87	3.58±1.24	4.42±0.67	0.000

******p<0.05 statistical difference. Anova Test*

According to these results, we see that the method with the best results is the Two-Finger Direct method, we can say that users are generally satisfied with the Two-Finger Direct method. Looking at accurate pointing, it is seen that Velocity-Screen is the most difficult method. This also matches the error rate results. However, the Two-Finger Direct method with the highest error rate is ranked as almost easy by participants. Figure 17 shows the result of questionnaire as a graph.

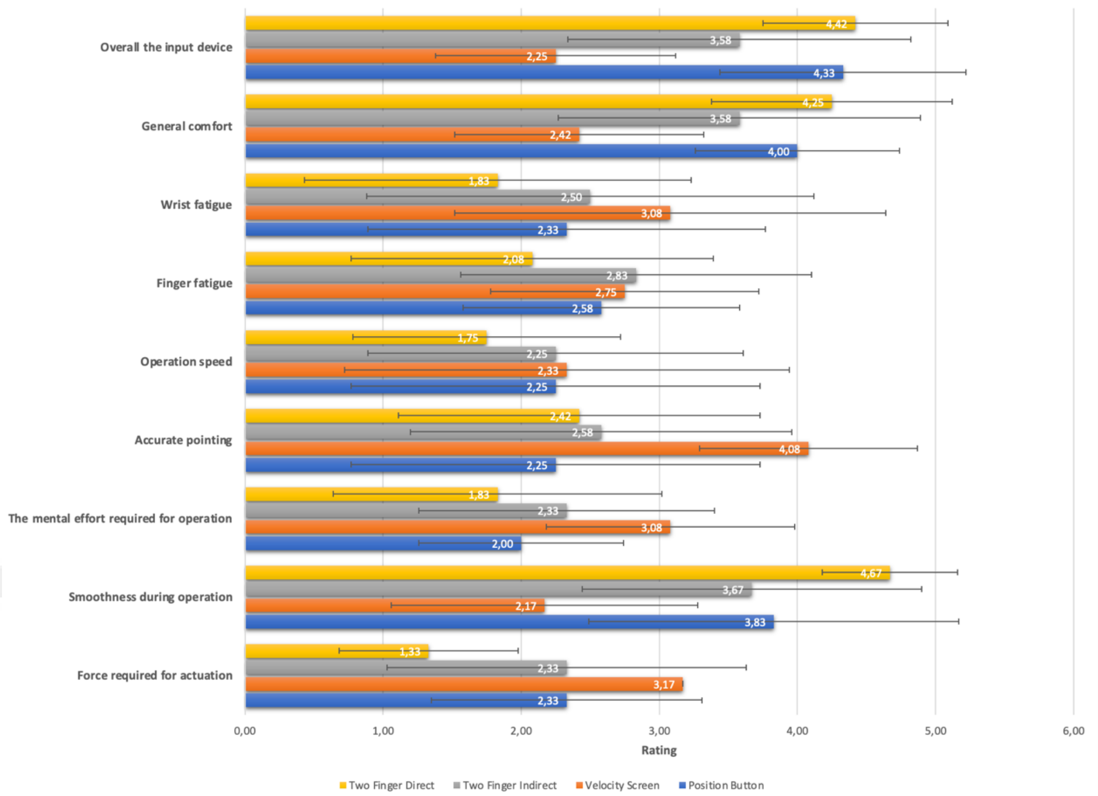


Figure 17. Questionnaire Results

6.4 Qualitative Result Observation

After performing all four cases and completing the experiment, participants were asked rating the control types and explain reasons for their choices and providing additional comments about control types if any. The notes we gathered can be summarized as follows.

- When asked which control type has been their first choice, 5 of the participants chosen Two-Finer Direct, 5 people Position-Button and 2 person Two-Finger Indirect.
- All participants have pointed out that the controlling of Velocity-Screen is the hardest. 2 of the participants have expressed that they could choose it if the sensitivity of the virtual joystick were adjustable.
- In general, all participants have provided positive comments about both of Two-Finger cases.
- The Position-Button case was the one that the participants found the most straightforward. They have rated other cases by using this case as a reference, although this was not asked. The reason of this behaviour could be that the

Position-Button is widely used in games, thus the participants feel familiar to that case.

- For the Position-Button, the characteristic reported as favorite is the aiming area being wide as it provides the user freedom of movement.
- When asked what their comments about Two-Finger Direct is, all participants expressed that they found this case as smooth, easy to aim and fast. Besides, they have pointed out that they missed the target because sometimes obstructed the cursor's view, or sometimes that they tapped fire too quickly before completing the whole cursor motion.
- In general, the one negative comment about both two finger cases is that it is sometime difficult to see targets due to the hand covering the screen.
- Another comment about the Two-Finger Indirect and Position-Button is that it is difficult to aim at the targets far from the current cursor position.

CHAPTER 7: CONCLUSION

In this study, four touch-based input methods for the point selection task were compared regarding accuracy and speed. There are two existing touch-based input methods for this task that are already used in mobile first-person shooter games: Position-Button and Velocity-Screen. In addition to these, there are two novel input methods proposed in this study: Two-Finger Direct and Two-Finger Indirect.

In this study, we compared these input methods using multi-directional tapping task in ISO 9241-9 on a 9.7-inch iPad Pro. It was found that the throughput and error rate values statistically significant according to the input methods ($p<0.05$).

The impact of input methods on throughput was statistically significant ($p<0.05$). Throughput for Two-Finger Direct, which is one of proposed methods, was measured at 2.33 bps, see Figure 15. It was found that there is a statistical difference between this value and the other three input methods ($p<0.05$). Throughput for Two-Finger Indirect, which is the other proposed method, was measured at 1.73 bps. It was found that there is a statistical difference between Two Finger Indirect and the two other input methods, Velocity-Screen and Two-Finger Direct ($p<0.05$).

The impact of input methods on error rate was statistically significant ($p<0.05$). The best accuracy has been observed at the Position-Button with an error rate of 1.9%, see Figure 16. The second accuracy belonged to the Two-Finger Indirect as the error rate is 3.9%. Surprisingly, the worst accuracy was found in Two-Finger Direct and Velocity-Screen, with respective error rates of 8.2% and 8.1%.

Looking at Two-Finger Indirect, the order of the input method is in the middle in terms of throughput and error rate (see Figure 15 and Figure 16). It has been the third position after Two-Finger Direct and Position-Button with its throughput value is 1.73 bps. Considering the error rate, it has been found much more accurate with an error rate of 3.9%. Participants' feedback matches these findings. They have expressed that Two-Finger Indirect is on a similar level of difficulty with Position-Button in terms of the force required for actuation, but that it is easier to hit the target by tapping the screen with the middle finger.

Considering on Two-Finger Direct, it is found that although it has the best throughput value, it has the worst error rate too. These findings were compatible with our qualitative observations. Participants have informed that Two-Finger Direct method is smooth, fast, and easy to aim with, but they sometimes missed targets due to being performed quickly.

To better express what high throughput and high error rate mean, we can give the types of typing on the keyboard as an example. Considering the two fingers typing and ten fingers typing, with two fingers, it is written slowly and with few errors while with ten fingers, it is speed but with many errors. If there are fifty pages to write, it takes five hours with two fingers and one hour with ten fingers to complete the task. In this case, it does not matter how many mistakes are made. Similar to this example, we can say that the Two-Finger Direct input method would be best in a game that can tolerate error and require speed.

Regarding all of these findings and reviews, it can be said that Two-Finger Direct would be the best input method for touch-based mobile first-person shooter games if enhancements are made about decreasing the error rate occurring due to performing quickly.

REFERENCES

Buxton, B. (2010) *A Touching Story: A Personal Perspective on the History of Touch Interfaces Past and Future,* Society for Information Display (SID) Symposium Digest of Technical Papers, Vol. 41(1), session 31, pp. 444-448.

Douglas, S. A., Kirkpatrick, A. E., and MacKenzie, I. S. (1999) *Testing pointing device performance and user assessment with the ISO 9241, Part 9 standard*, Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI '99. New York: ACM, pp. 215-222.

Farmani, Y. and, Teather, R. *Player performance with different input devices in virtual reality first-person shooter games*, in: SUI 2017 - *Proceedings of the 2017 Symposium on Spatial User Interaction*. Presented at the 5th ACM Symposium on Spatial User Interaction, SUI 2017, October 16–17, 2017,

Fitts, P.M. (1954) *The information capacity of the human motor system in controlling the amplitude of movement*, J. Exp. Psychol. Vol 47, pp. 381-391

Fitts, P.M., Peterson, J.R. (1964) *Information capacity of discrete motor responses,* J. Exp. Psychol. Vol. 67, pp. 103-112

Hassan, M., Magee, J. and MacKenzie, I. S. (2019) *A Fitts' law evaluation of hands-free and hands-on input on a laptop computer.* In International Conference on Human-Computer Interaction, Springer, Cham, pp. 234-249.

Hinckley, K., and Wigdor, D. (2012) *Input technologies and techniques* [Online]. Available at: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/11/Input-Technologies-and-Techniques-HCI-Handbook-3rd-Edition.pdf (Accessed: 01 August 2022)

ISO, 9421-9 Ergonomic requirements for office work with visual display terminals (VDTs) - Part 9: Requirements for non-keyboard input devices (2000), International Organization for Standardization

Klochek, C. and MacKenzie, I.S. (2006), *Performance measures of game controllers in a three-dimensional environment*, In Proceedings of Graphics Interface, Toronto, Canada

MacKenzie, I. S. (1992), *Fitts' law as a research and design tool in human-computer interaction*, Human-computer interaction, Vol. 7(1), pp. 91-139.

MacKenzie, I. S. (2013), *Human-Computer Interaction: An Empirical Research Perspective*. 1st Edition, USA: Morgan Kaufmann Publishers

MacKenzie, I. S. (2015), *Fitts' Throughput and the Remarkable Case of Touch-Based Target Selection*, in: Kurosu, M. (Ed.), Human-Computer Interaction: Interaction Technologies, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 238–249

MacKenzie, I. S. (2018). Fitts' Law. In K.L, Norman (Ed.). *The Wiley handbook of human computer interaction*, pp. 349-370, Hoboken, NJ: John Wiley And Sons.

Magee, J., Felzer, T., MacKenzie, I.S. (2015), *Camera Mouse + ClickerAID: Dwell vs. Single-Muscle Click Actuation in Mouse-Replacement Interfaces*, in: Antona, M., Stephanidis, C. (Eds.), Universal Access in Human-Computer Interaction. Access to Today's Technologies, Lecture Notes in Computer Science. Springer International Publishing, Cham. pp. 74–84.

Napatov, D., Castellucci, S.J. and MacKenzie, I.S. (2009), *ISO 9241-9 evaluation of video game controllers*, In Proceedings of Graphics Interfaces, Toronto, Canada

Natapov, D., MacKenzie, I.S. (2010), *The trackball controller: improving the analog stick*, In: Proceedings of the International Academic Conference on the Future of Game Design and Technology - Futureplay '10. Presented at the International Academic Conference, ACM Press, Vancouver, British Columbia, Canada, p. 175.

Newzoo Global Games Market Report (2021), *Newzoo Global Games Market Report 2021* [Online]. Available at: https://newzoo.com/insights/trend-reports/newzoo-global-games-market-report-2021-free-version/ (Accessed: 13 April 2022).

Pino, A., Tzemis E., Ioannou, N., Kouroupetroglou, G. (2013), *Using Kinect for 2D and 3D Pointing Tasks: Performance Evaluation*, M. Kurosu (Ed.), Human-Computer Interaction, Part IV, HCII 2013, pp. 358–367

Ramcharitar, A. and Teather, R. J. (2017), *A Fitts' law evaluation of video game controllers: thumbstick, touchpad and gyrosensor,* In Proceedings of the 2017 chi conference extended abstracts on human factors in computing systems, pp. 2860-2866.

Shannon, C. E. and Weaver, W. (1949), *The mathematical theory of communication,* Urbana: Univ. of Illinois Press.

Smith, W. (1996), *ISO and ANSI ergonomic standards for computer products: A guide to implementation and products,* Prentice Hall, New York.

Zaranek, A., Ramoul, B., Yu, H. F., Yao, Y. and Teather, R. J. (2014), *Performance of modern gaming input devices in first-person shooter target acquisition*, In CHI'14 Extended Abstracts on Human Factors in Computing Systems, pp. 1495-1500.

Appendix A1: Questionnaires Answers of Participants

No	Participant No		Q1: Force required for actuation				Q2: Smoothness during operation				Q3: The mental effort required for operation				Q4: Accurate pointing				Q5: Operation speed				Q6: Finger fatigue				Q7: Wrist fatigue				Q8: General comfort				Q9: The input device overall			
		Control Type	PB	VS	TFI	TFD	PB	VS	TFI	TFD	PB	VS	TFI	TFD	PB	VS	TFI	TFD	PB	VS	TFI	TFD	PB	VS	TFI	TFD	PB	VS	TFI	TFD	PB	VS	TFI	TFD	PB	VS	TFI	TFD
1	#1		3	5	3	3	5	2	5	4	3	3	3	2	2	5	5	2	3	5	3	3	2	4	5	5	1	1	5	4	4	2	2	4	5	2	5	4
2	#2		2	5	1	1	5	2	5	5	1	4	1	1	5	5	1	1	1	4	1	1	3	4	1	1	1	4	1	1	4	2	5	5	5	2	5	5
3	#3		1	1	5	1	2	2	2	5	2	3	2	2	1	4	4	1	1	3	2	1	1	2	5	1	1	2	5	1	4	3	1	5	5	3	1	5
4	#4		1	2	2	2	5	3	5	5	3	3	4	3	1	4	3	2	1	1	1	1	2	4	3	3	2	2	1	1	5	4	5	5	5	2	4	4
5	#5		2	4	3	1	3	2	3	5	2	3	4	1	1	4	3	5	1	1	1	3	4	3	4	2	3	2	3	1	3	3	4	5	4	2	3	5
6	#6		2	2	1	1	2	1	4	5	3	2	3	2	1	3	1	2	4	2	3	1	2	2	2	2	3	4	4	5	5	2	4	4	5	4	5	5
7	#8		2	3	2	1	5	3	3	4	2	3	3	2	5	3	3	2	5	3	5	3	2	3	3	1	1	2	2	1	5	3	4	5	5	3	3	5
8	#9		2	2	1	1	5	2	5	4	2	2	1	1	2	4	1	3	1	1	1	3	2	3	2	1	1	1	1	1	4	3	5	3	4	2	4	3
9	#10		3	5	2	1	3	5	4	5	2	2	1	1	3	5	2	1	4	5	2	1	4	3	2	2	5	5	4	3	4	1	4	5	4	3	4	5
10	#11		4	4	4	2	5	2	2	4	1	3	2	5	3	4	3	4	1	1	1	2	2	2	2	2	4	4	1	1	3	2	2	3	4	2	2	4
11	#12		2	4	3	1	4	1	2	5	2	4	2	1	1	3	4	2	2	1	4	1	3	2	3	1	4	5	2	1	4	3	3	4	4	1	3	4
12	#13		4	1	1	1	2	1	4	5	1	5	2	1	2	5	1	4	3	1	3	1	4	1	2	4	2	5	1	2	3	1	4	3	2	1	4	4

Appendix A2: Participants' Profiles

No	Participant No	Age	Job	Oyun oynuyor musunuz?	Ne sıklıkla bilgisayarda oyun oynuyorsunuz?	Ne sıklıkla telefon/tablette (dokunmatik ekranlı cihazlarda) oyun oynuyorsunuz?	FPS türü oyunları ne sıklıkla oynuyorsunuz?	FPS/TPS (nişan almalı) oyunlar için tercih ettiğiniz platform hangisi?	PUBG oynuyor musunuz, evetse ne sıklıkla oynuyorsunuz?	Telefon/tablette (dokunmatik ekranlı cihazlarda) ne sıklıkla FPS/TPS (nişan almalı) oyunlar oynuyorsunuz?	Siralama	Notes	Rank of personel choice of PB	Rank of personel choice of TFD	Rank of personel choice of TFI	Rank of personel choice of VS
1	#1	23	Öğrenci	-	-	-	-	-	-	-	PB>TFD>TFI>VS	*VS hareket ettirmek cok zor. *TF'lerde bilek yorgunlugu oldu diyor, bos kalan elle tableti tutmak zorunda kaldigi icin. Onun disinda TF'ler iyi o yuzden 2. ve 3. sirada onlar.	4	3	2	1
2	#2	23	Öğrenci	-	-	Kelime oyunu	Eskiden oynardı	PC	Eskıden oynardı	Eskiden hergün	TFD>TFI>PB>VS	*TF'ler icin parmaklarimi daha rahat kullanabiliyorum, cok kolay nisan alabiliyorum. *VS, nisan almak cok zor.	2	4	3	1
3	#3	23	Öğrenci	Candy crush, kart	-	Her gun	-	-	-	-	PB>TFD>TFI>VS		4	3	2	1
4	#4	21	Yaratıcı topluluk bağla	-	Eskıden	Temple rın	kucuklugunde	PC	-	-	PB>TFI>TFD>VS	*PB nisan alirken ozgur, ama ates ederken tek bir nokta olmasi daha kolaydi. *TFI, kaydirarak hedef almak hizli, tab ile ates kolay oluyor. Bir parmagim surekli ekrandaydi, digerini ates etmek icin tap yapiyordum. *TFD, dokunup nisan alirken refleks olarak diger parmakta gidiyor, cok basiyorum. *VS, joystick cok hizliydi hedef alamadim.	4	2	3	1
5	#5	26	Tasarımcı	Basit, kelimelik	-	Ayda 2-3 kez	deneme amaclı (x)	PC	-	-	TFD>PB>TFI>VS	*TFD, tek el ve direk oldugu icin cok iyiydi ama bazen parmagimi koydum sanip oburunu bastigimda hata yaptim. *PB, alan belli olduğu için *TFI, parmak kaçabiliyor diye 3. *VS, çok kötü/hassas. Durarak gitmem gerekti.	3	4	2	1
6	#6	22	Öğrenci	Araba yarışı, NBI	Arada sırada, Asssassin Creed	Yakın zamanda, çok az	Nadir	-	Arada sırada	-	TFD>PB>TFI>VS	*TFD akıcı. *TFI'yı PB ye benzeterek oynadı. PUBG'deki kontrol oldugu gibi kalsin isterim, silah secme vs. oldugundan simdiki iyi.	3	4	2	1
7	#8	25	Öğrenci	cok degil	COD4,res 2 eskiden	-	eskiden	hepsi	gecen sene cok, gunde 40 kontol zor	-	TFD>PB>TFI>VS	*TFD, kontrol kolay * VS, zor.	3	4	2	1
8	#9	28	makine muhendisi	evet	-	tablet gunde 2 saat	eskiden	PC	eskiden, controller ile		TFI>VS>PB>TFD	*TFI, akicilik cok iyi, seri * TFD, iki tik hata veriyor akici ama hedef kaciyor, cok fazla hata yapiliyor, akicilik cok iyi ama.	2	1	4	3
9	#10	29	front-end developer	puzzle telefon ile	hic	cok nadir	hic	-	-	-	TFD>TFI>PB>VS	*TFD, hareket iyi sadece hedefi gormek zorlasiyor bzen *TFI,PB surukleme sevmiyorum zordu. *VS. cok zorlandim hedefe goturme kisminda.	2	4	3	1
10	#11	23	e-ticaret	deli gibi	nadiren	telefon, gunde yarim saat	eskiden	telefon	eskiden nadiren	nadiren	PB>TFD>TFI>VS	* PB, aliskanlik *TFD, yoruyor alisinca guzel *TFI, uzak mesafede yoruyor *vs, guzel ama hassas * TFD, TFI gibi kullandi.	4	3	2	1
11	#12	28	Yazılımcı	Çok nadir, strateji	online oyun çok nadir	ayda yılda bir kelime oyunu bulmaca oyunu	yıllar önce, among us	-	-	-	PB>TFD>TFI>VS	*PB ve TFD kolay zorluk ayni, dogruluk degisiyor sadece, TFD parmak onumu kapattigi icin dogruluk dusuyor.	4	3	2	1
12	#13	28	Yazılımcı	Nadiren	çok nadir	arada sırada	çok nadiran	PC	ilk çıktığında populerken	Brawl Stars	TFI>PB>TFD>VS	*TFI surekleme olayi guzel, esneklik saglamasindan dolayi, yeni opsiton sunuyor guzel bu. PB de uzak mesafeye goturmesu zor ama diger mesefelerda guzel. *TFD nereye bastigimi goremiyorum, cursor parmaginin altinda kaliyor, ikisini hareket ettirip basmak zor, yordu beni. *VS, sensivity ayarlanabilir olmali herkes kendisi.	3	2	4	1