Secrets of Gosu: Understanding Physical Combat Skills of Professional Players in First-Person Shooters

Eunji Park
KAIST
Daejeon, Korea
eunji.park@kaist.ac.kr

Sangyoon Lee
KAIST
Daejeon, Korea
sl2820@kaist.ac.kr

Auejin Ham
DGIST and KAIST,
Daegu and Daejeon, Korea
hamaj@kaist.ac.kr

Minyeop Choi
KAIST
Daejeon, Korea
okas832@kaist.ac.kr

Sunjun Kim
DGIST
Daegu, Korea
sunjun_kim@dgist.ac.kr

Byungjoo Lee*
Yonsei University
Seoul, Korea
byungjoo.lee@yonsei.ac.kr

ABSTRACT

In first-person shooters (FPS), professional players (a.k.a., *Gosu*) outperform amateur players. The secrets behind the performance of professional FPS players have been debated in online communities with many conjectures; however, attempts of scientific verification have been limited. We addressed this conundrum through a data-collection study of the gameplay of eight professional and eight amateur players in the commercial FPS Counter-Strike: Global Offensive. The collected data cover behavioral data from six sensors (motion capture, eye tracker, mouse, keyboard, electromyography armband, and pulse sensor) and in-game data (player data and event logs). We examined conjectures in four categories: aiming, character movement, physicality, and device and settings. Only 6 out of 13 conjectures were supported with statistically sufficient evidence.

CCS CONCEPTS

• **Human-centered computing → User studies**.

KEYWORDS

first person shooters, games, e-sports, performance metrics, datasets

ACM Reference Format:
Eunji Park, Sangyoon Lee, Auejin Ham, Minyeop Choi, Sunjun Kim, and Byungjoo Lee. 2021. Secrets of Gosu: Understanding Physical Combat Skills of Professional Players in First-Person Shooters. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3411764.3445217

*Corresponding author

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8096-6/21/05…$15.00
https://doi.org/10.1145/3411764.3445217

1 INTRODUCTION

In first-person shooters (FPS), one of the most representative genres of e-sports, players require a high level of gaming combat skills to compete. Such skills include the ability to accurately shoot small, fast-moving enemies, incapacitate an enemy attack through unpredictable movement, and create a more manageable shooting situation in the game than the enemy. These skills are not easily acquired through short-term exercises and thus have been regarded as part of successful FPS players' core competitiveness. Professional FPS players have overwhelming gaming combat skills compared to amateur players. In gaming communities worldwide, these professional players are often called Gosu, a Korean term meaning a highly skilled person, and they are revered for their dominating performance.

However, our understanding of FPS players' gaming combat skills remains at initial level. Understanding the mechanisms through which professional FPS players can gain overwhelming gaming combat skills is an essential issue in e-sports science. It enables more credible player training and player trading, and as a result, like traditional sports, e-sports can be taken seriously. Typically, the gaming physical combat skills of FPS players have primarily been measured using in-game statistics, such as the kill/death ratio or average damage per game. Such aggregated metrics only confirm the consequential and straightforward fact that a particular player is superior to other players and cannot tell us *how* the player performs better.

Various online communities have speculated on the secrets behind the performance of professional FPS players. Typically, the conjectures include how professional players aim, move, or what type of devices to use. More specifically, one conjecture states that firing a weapon with a single or short burst of controlled clicks (called tapping) or aiming a target with the arm instead of the wrist is effective for higher performance. Another conjecture is that a player should make unpredictable movements, combined with jumping and crouching. However, these conjectures have not been scientifically verified, and credible training guides for amateur players are virtually nonexistent. Consequently, amateur FPS players subjectively analyze professional players' play videos and follow input devices used by professional players. Moreover, many of them practice with community-suggested shooting training programs, such as Aim Hero,[1] Aim Lab,[2] and KovaaK 2.0,[3] which have not proven effective.

This study quantitatively compares professional and amateur FPS players' performance to solve this problem and confirms whether famous conjectures on the web are correct. First, we selected conjectures from sources such as the official blogs of companies developing FPS related software and YouTube videos with least 100,000 to over 2,000,000 views. Then, we filtered those that can be commonly applied to general FPS games. We excluded those that depend on the game (e.g., "Hide behind the second box in that map") or are difficult to quantify (e.g., "Do image training on shooting situation"). Table 1 summarizes the conjectures for the current study. Our goal is to verify these conjectures scientifically and to fulfill the desired knowledge of amateur FPS players.

To achieve the goal, we devised 15 quantitative performance metrics that compare professional and amateur players in the following four aspects of FPS play: *aiming*, *character's movement*, *physical skills*, and *device and settings*. Furthermore, we built a comprehensive gameplay logging system that can measure the proposed metrics in `Counter Strike: Global Offensive` (CS:GO), a popular commercial FPS game. The system can log data from six behavioral sensors (i.e., eye tracker, motion capture, dual-sensor mouse, `Wooting` analog keyboard, heartbeat sensor, and an electromyography (EMG) sensor) and data from in-game situations during gameplay. The timestamps between the sensors are synchronized with high accuracy and precision.

We recruited eight professional FPS players and eight amateur players and measured their performance with our logging system as they played CS:GO. Each group played against another player within the same group to create a general gameplay situation. By analyzing the resulting data set with the proposed metrics, we revealed that only 6 out of 13 of the conjectures are supported by statistically significant evidence. Finally, the contributions of this study can be summarized as follows:

- We surveyed conjectures on the web about FPS performance and summarized them into four specific categories.
- We developed an advanced gameplay logging system that can measure signals from six behavioral sensors and in-game situations with synchronized timestamps.
- We present 15 quantitative performance metrics that can scientifically evaluate the conjectures of FPS performance.
- We verified the conjectures on the web based on the proposed metrics.
- We presented the collected dataset to the public for future research [4].

2 RELATED WORK

The quantitative evaluation of athletic performance is considered essential in all sports genres. It aids both in-game and out-of-game decision-making [9] for coaches, clubs, and athletes. It also provides clues to improve athletic performance on individual and team levels [17]. Sports fans also are interested in such data interpretation. Stoltz argued that, as the movement of sabermetrics in baseball expands, audiences who search for new baseball knowledge have increased correspondingly, creating a new era of baseball experience [46]. Other studies have found that the presentation and visualization of such data lead to a better understanding of the game and increase engagement and excitement concerning the game experience for sports and e-sports fans. [12, 16, 19, 23, 54]. For similar reasons, performance measurements and analyses in e-sports are crucial, particularly considering that the data generated from e-sports are comparable to even the most data-rich traditional sports, such as Formula One [12].

Moreover, e-sports can be classified into several representative genres, such as Real-Time Strategy (RTS), Multiplayer Online Battle Arena (MOBA), and FPS games. Each of these genres usually has different quantitative metrics for player performance. For instance, MOBA games typically require active collaborations and efficient communication. As a result, statistics, such as kill death assist (KDA) per game or wards purchase (WA) per game, are often used for quantitative metrics. In addition, KDA is calculated as the addition of the number of kills and assists divided by the number of deaths, and WP measures the number of ward purchases. Moreover, RTS games demand a high level of multitasking abilities and use Actions Per Minute (APM) or Effective Actions Per Minute (EAPM) as the quantitative measurement of player performance, measured by the number of actions, such as keyboard inputs and mouse clicks, created by the player in a minute.

However, an FPS game requires players to kill fast-moving enemies. Therefore, they typically use quantitative measurements, such as the Kill/Death Ratio (KDR), Average Damage Per Game (ADG), or Headshot Percentage (HS%). KDR is measured by the number of kills over the number of deaths in a game and ADG is measured by the amount of damage to enemies in a game. HS% is measured by the percentage of headshots in a game. For the quantitative approach to an FPS game, several studies have suggested implicit measures that can represent a player's skills that are not officially given from the game. In an FPS game called `Red Eclipse`, Buckley et al. [15] found that various keyboard input events, such as the average and the total number of keys pressed or the number of keys pressed at once, demonstrated a higher prediction of skill level than mouse input events, such as the number of clicks. Other studies on keyboard and mouse input have also shown that measurements, such as the time spent holding a specific key or the total mouse distance moved, can also explain the skills of an FPS player [14, 45]. To characterize the skill level of players in CS:GO, Velichkovsky et al. employed eye gaze data, especially the analysis of the fixation duration on the computer screen [48]. They found that professional players generated higher frequencies of a longer fixation duration than amateur players.

Beyond the proposal of measurements on single behavioral data, some researchers have proposed a particular system that can monitor and measure player behavior and in-game data in an integrated way in an FPS game. In CS:GO, Korotin et al. [27] proposed a system that can comprehensively measure the player's gaze, keyboard input, mouse input, heart rate, and in-game data. Stepanov et al. [45] extended the previous system by adding a few more sensors, including galvanic skin response sensor, EMG, and inertial measurement unit sensor. They proposed a technique to maintain synchronization

[1] *Aim Hero*, ProGames Studio, 2016
[2] *Aim Lab*, Statespace, 2018
[3] *KovaaK 2.0*, The Meta, 2018
[4] http://leebyungjoo.com/secrets-of-gosu

Table 1: Summary of conjectures from gaming communities.

	Conjectures	Performance Metrics
Aiming		
A-1	"When track aiming, keep the crosshairs on the target at all times." [4, 22, 24]	Enemy-crosshair stickiness
A-2	"When flicking (snapping) aim, quickly and accurately hit a moving target before they can react." [4, 24]	Angular velocity of the mouse
A-3	"Move your mouse in the opposite direction of each weapon's spray or recoil pattern. This compensates for the kicks during shooting." [2, 3]	Amplitude of the recoil compensating movement
		Duration of shooting (tapping vs. spraying)
A-4	"Most professional players have their own sensitivity." [24, 47]	Force inefficiency
Character Movement		
M-1	"Be unpredictable. Make random moves." [3]	Entropy of pressed keys
M-2	"Make your special movement combinations: Sidestep, Strafe shooting, Jumping, Crouching." [2, 3]	Frequency of movement combinations
M-3	"Don't reload habitually. Know the right time to reload." [13]	Reload efficiency
Physical Skills		
P-1	"Most professional players use arm aiming." [5]	Rotation ratio of elbow and wrist
P-2	"Let loose of your arm and wrist, especially during aiming." [6]	Muscle activity
P-3	"Scan all surroundings continuously." [6, 44]	Duration of fixation
		Number of saccades
P-4	"Stay calm and have a composure." [6]	Composure
Device and Settings		
D-1	"Most professional players use low sensitivity." [22, 47]	Used area of the mousepad
D-2	"To minimize the stress of a player's wrist, place a keyboard perpendicular." [53]	Keyboard perpendicularity

between sensors within 10 ms of temporal accuracy. However, the contributions of these studies focused on designing a measurement system.

Some studies aimed to analyze the skills of FPS players qualitatively. From a video analysis of gameplays, Reeves et al. found that expert FPS players tend to deceive their positions and take clever sidesteps to make it difficult for the opponent to shoot or frequently sit while shooting to increase shooting accuracy [40]. Fanfarelli [20] interviewed professional players from `Overwatch`, a popular FPS game, to identify the critical elements of FPS skills, as follows: survival, anticipation/prediction, communication, thoughtfulness, aim, ability usage, movement and positioning, and team-based mechanical synergy. In addition, Fanfarelli mentioned that most game genres require these elements, but especially in FPS, the mechanics covering the aiming ability are considered essential for expertise in FPS play. Witkowski [50] emphasized the kinesthetic ability in e-sports through a qualitative study that interviewed Counter-Strike players and organizers at the three e-sports tournaments: *2009 The eXperience, 2009 DreanHack Winter, and 2010 World Cyber Games.*

To summarize, many previous studies have analyzed different genres of e-sports from diverse perspectives. Among the various genres, the FPS has frequently been used as the research subject due to its versatile and dynamic environment. However, many research questions from previous studies are different from what actual gamers want to know. In this study, we aim to bridge the gap between the researchers and gamers by validating conjectures from online gaming communities. Our findings present a greater external validity than previous studies by analyzing a comprehensive dataset collected from the gameplay between professional and amateur FPS players.

3 GAME PLAY LOGGING SYSTEM

To verify the conjectures in Table 1, we experimented with an uncontrolled FPS game playing situation to collect realistic gameplay data from players. We chose CS:GO for the FPS gameplay because of its global popularity and abundant open sources. We aimed to record various types of physical performance of the players during an actual gameplay situation. To this end, we developed a gameplay logging system consisting of six external sensors and an in-game data logger. In this section, we describe the implementation of the logging system. In the system, the six external sensors collect user behavioral data, and the in-game logger exploited the internal game parameters through a Dynamic Link Library (DLL)-injection attack (Figure 1).

3.1 Behavioral Data

We devised the *loggerslate* to begin or halt the logging process by sending a User Datagram Protocol (UDP) packet to all six external sensors simultaneously. When we turn on the system, loggerslate sends a wake signal to the local-time stamper, motion-capture sensor, and analog input keyboard, and the three systems record the epoch time obtained from the system clock of the Operating System (OS). To reduce the operational burden on the OS and minimize the asynchrony between the timestamps of each sensor, the four sensors (dual-sensor mouse, eye tracker, EMG sensor, and pulse sensor) receive the local time from the local-time stamper and record the

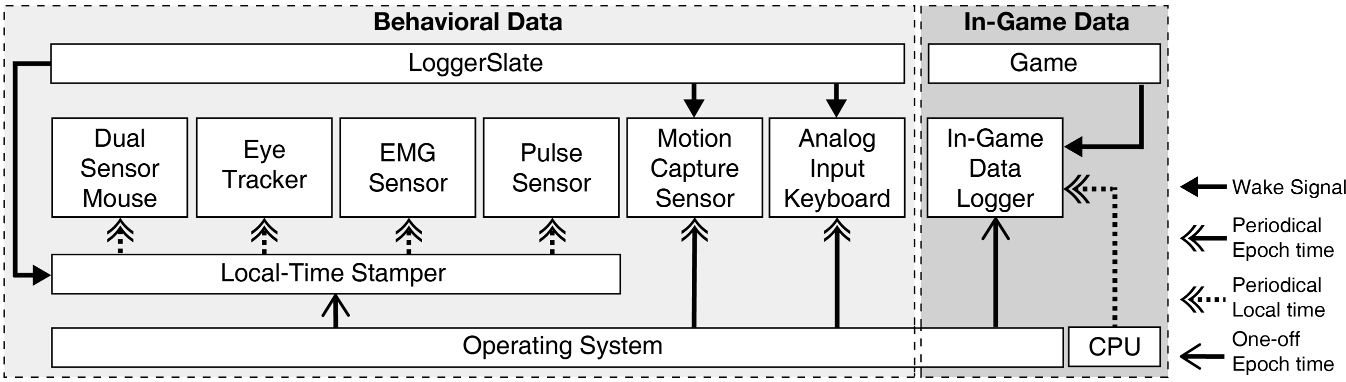

Figure 1: Gameplay logging system: The system collects a comprehensive dataset of six behavioral and in-game data. The motion-capture sensor and analog input keyboard receive epoch times from the operating system (OS) periodically. The dual-sensor mouse, eye tracker, electromyography (EMG) sensor, and pulse sensor receive global timestamps by adding the periodical local time to the one-off epoch time. The in-game logger receives a performance counter as the local time. The collected data were synchronized using a unified time-keeping mechanism.

local time using a different frequency for each sensor. The local time is set to zero when the local-time stamper receives the wake signal from the loggerslate, and the resolution of the local time is 1 ms. The global timestamps of the four sensors are acquired by adding the local time to the epoch time recorded when the local-time stamper first receives the wake signal.

3.1.1 Dual-sensor mouse. We fabricated a mouse device for our study by adopting a dual-sensor mouse [26]. We collected the length, width, and height of the mice used by 255 professional players from prosettings.net [7] and designed a mouse according to the average size. The resulting dimensions resemble the Logitech G Pro Wireless mouse. The dual sensors of the mouse allow measuring the rotation angle and clutching duration of the mouse. We can also control the virtual sensor position using the two static optical sensors, but the virtual sensor position was fixed at the center of the mouse in this study. The device polling rate was set at 500 Hz, and the sensor resolution was set to 12k dots per inch (dpi) internally and was downsampled to a user-defined dpi. The dual-sensor mouse data include a timestamp in milliseconds and microseconds, dx and dy of the front and rear sensors in counts, dx and dy of the cursor in pixels (px), and button states, such as left or right clicks. The rotation angle and clutching duration were calculated from these data afterward.

3.1.2 Analog input keyboard. We used a `Wooting One` analog input keyboard, which can detect can detect the gradual movement of each key like a joystick, which enables logging how far the user presses down. It is connected via a universal serial bus (USB), and the data were logged through the Wooting Keyboard Library [43]. The analog input keyboard data include a timestamp in milliseconds, the pressed key, and pressed depth of the key as an integer with the range from 0 to 231.

3.1.3 Motion-capture sensor. We used the OptiTrack three dimensional (3D) position tracking system for real-time position tracking of the objects attached with specialized reflective markers. Our system consists of eight high-speed tracking cameras and the motion-capture software MOTIVE. OptiTrack cameras were connected using Cat 6 Ethernet connections to the OptiTrack switch for data consolidation and power distribution. The switch sends UDP packets with the position and orientation data for each marker. The resolution of the marker displacement is accurate to 1-mm under a sampling rate of 240 Hz. The data of the motion-capture sensors includes the timestamp in milliseconds; frame number; marker ID in integers; x, y, z coordinates; and marker diameter. When the marker is added, ID; x, y, z coordinates; and diameter for each marker are added to the data column.

3.1.4 Eye tracker. We used the SMI REDn Scientific System, with iViewRED software as a monitor-mounted eye-tracking system in this work. The eye tracker requires a calibration process for each user to establish the relationship between the eye position in the camera view and the gaze point in space. We performed a five-point calibration each time we started logging. The eye tracker data include the timestamp in milliseconds, x, y coordinates of both the left and right eye staring at the gaze point, and the diameter of each pupil.

3.1.5 EMG sensor. We also used the MYO armband consisting of eight EMG sensors and nine axis IMUs. The MYO armband was worn on the forearm below the elbow. The intuitive sensor logging system receives data from the MYO armband via Bluetooth 4.0. The sampling frequencies are 50 Hz for the IMU and 200 Hz for the EMG. The data from the MYO armband include the timestamp in milliseconds and data from the eight EMG sensors. From the IMU, the orientation (row, pitch, and yaw) data in radians, three-axis accelerometer data in G, and gyroscope data in G are logged. The operating and calibration status and whether the band is worn on the left or right arm were also recorded.

3.1.6 Pulse sensor. We used a Grove ear-clip heart-rate sensor attached to an Arduino UNO. The ear-clip contains an optical sensor to monitor the heart rate of users at 500 Hz. The pulse sensor data include the timestamp in milliseconds, beats per minute (BPM), inter beat interval (IBI), and signal (SIG).

Figure 2: Experimental setup of the gameplay logging system: Box 1 shows the desktop setup of eight motion-capture cameras. Boxes 2 and 3 are the positions of the reflective markers and MYO armband. Boxes 4 and 5 are the dual-sensor mouse and two sensor positions. The ear-clip type pulse sensor was attached to the player's right earlobe (not pictured).

3.2 In-Game Data

We devised an in-game data logger to obtain detailed in-game data, such as character location, fired weapons, health, and more. We modified Osiris [18], an open-source pure C++ CS:GO internal cheat by DLL injection to extract the data from memory. We used the epoch time as the timestamp for the in-game data to synchronize the time with other sensors. First, when the system starts to work, the system records the start time (the local time of the machine) in epoch time using the *GetSystemTimeAsFileTime*. Then, we used a hardware performance counter to measure the elapsed time more precisely. The counter has a high resolution and can be used as an accurate timer. The final timestamp was calculated by adding *QueryPerformanceFrequency* divided by *QueryPerformanceCounter* to the start time.

3.2.1 Player data. To collect the player data coordinated with the player's viewpoint, we employed the *paintTraverse* function from the CS:GO game engine, which iterates over entities in the game. We added a logger hooked into *paintTraverse* function because every entity will be already drawn to the screen before the *paintTraverse* is called. As *paintTraverse* is called by the game engine every frame, the logger collects the player's data listed below by iterating all of the entities existing in the interface. We logged four player data items: player information, the current 3D position of the player, and the enemy player's skeletal position and visibility. Player information includes the players and their opponents' names and health points between 0 and 100. The current player's 3D position includes the x, y, and z 3D coordinates. The opponents' skeletal data include the x and y 2D coordinates based on the screen resolution (1920 x 1080). Because we only need the general shape of a character's body, the logger records 19 positions out of 85: seven bones from the spine (from the head to the pelvis), six bones for the left and right arms (three bones each), and six bones for the left and right legs (three bones each). The enemy's skeletal data include the visibility of the enemy player's skeletal position based on the current player's viewpoint, where 0 is invisible, and 1 is visible.

3.2.2 Game events. The game engine generates events for broadcasting various character actions. By placing the event listener for specific target events, the logger can manage these events and logs as the events arise. We logged six items for game events: player death, weapon fire, weapon reload, weapon zoom, player spawn, and player jump. When each event occurs, the name of the player who generates the event is logged. When a player death event occurs, the dead player's name and whether it was a headshot are also logged.

3.2.3 Output format. The logger saves these data into CSV format. The rows contain a timestamp and either player data or game events. For player data, the timestamp, current player's name, health points, weapon type, and 3D position are shown first. Then, the enemy's name, health point, and each of the 19 2D skeletal positions with visibility status are shown.

4 DATA ACQUISITION

With the developed logging system, 16 FPS players (eight amateurs and eight professionals) participated in the data collection. Each player performed a 30-minute gameplay session against another player within the same expertise group while collecting data.

4.1 Participants

According to their performance level, we recruited 16 participants from the following two groups: a professional player group and an amateur player group. For the professional player group, eight professional FPS players of Rainbow Six Siege[5] participated, because it was difficult to recruit CS:GO professional players in the area where the study was conducted. Instead, we confirmed that

[5] *Tom Clancy's Rainbow Six Siege*, Ubisoft, 2015

Figure 3: Map overview used for one-on-one gameplay.

they have universal professional skills that can be commonly applied to different types of FPS games. In a preliminary survey, the participants in the professional group replied that they were also performing best in Overwatch and PUBG (e.g., grandmasters (top 1%) in Overwatch or listed in the top 100 places in PUBG as a solo).

For the amateur player group, eight players from a local university who recreationally play FPS games participated. During recruitment, we filtered only those who had sufficient FPS experience using their usual FPS rank; For instance, among those who play Overwatch, we only recruited those who have a higher rank than the gold rank (top 71%).

The average amount of FPS playing time per week was 47 hours (SD = 12.97) for the professional player group and 9.13 hours (SD = 5.77) for the amateur player group. The average age of the participants was 22.75 years old (SD = 3.37) for the professional player group and 23.5 years old (SD = 3.38) for the amateur player group. All participants were male and right-handed, and none had previous experience playing CS:GO.

4.2 Task

Each participant played a one-on-one match against an opponent player for 30 minutes. The opponent player was selected within the same group to replicate the real-world play environment and maintain the gameplay comptetitiveness. In a typical FPS game, players are automatically assigned to other players with a similar performance level. The goal of the task was to kill the opponent as many times as possible, using only the designated weapons. The participants were in terrorist group and used a assault rifle (the AK-47) for the first 20 minutes and used a sniper rifle (the AWP) for the remaining 10 minutes, while the opponent used a assault rifle (M4A4) for 30 minutes. The map used for gameplay was a conventional workshop map called *aim_map*[6] which is used for one-on-one play. The map did not include specific missions, such as bomb planting or rescuing hostages, to keep the goal as simple as possible. The overview of the map is depicted in Figure 3.

4.3 Apparatus

The gameplay logging system we described in Section 3.1 was implemented on a Windows desktop computer (64-bit Windows 10 Pro, Intel Core i7-9800X CPU @ 3.80 GHz, 16 GB RAM with NVIDIA GeForce GTX 2080 Ti). We used an Alienware 27 gaming monitor with a native resolution of 1920 x 1080 at 240 Hz. The mousepad was a SteelSeries QcK heavy model measuring 450 mm x 400 mm x 6 mm.

[6] *aim_map*, CS:GO Workshop, FREIHH & leplubodeslapin, 2013

4.4 Setup and Procedure

The apparatus was installed on a regular desk. Eight motion-capture cameras were positioned around the desk to capture the 12 markers placed on the keyboard, mousepad, and participant's body. We explain the specific locations of the markers in Section 4.1.2 on motion-capture data labeling. The participants wore an MYO armband on their right forearm without interfering with the motion-capture markers. The eye tracker was placed on the bottom side of the monitor. For each participant, we properly adjusted the monitor height and angle to place the eyes within the tracking field. We set the Windows mouse setting to 1000 dpi for all participants, and they adjusted the mouse sensitivity in the game if they desired. The entire experimental setup is illustrated in Figure 2.

For the experiment, each participant was instructed to sit on a gaming chair in front of the experimental setup. Because the participants did not have any experience with CS:GO, we allowed 15 minutes of practice to familiarize players with the game basics. During this period, each participant adjusted the in-game settings, such as the mouse sensitivity, if they desired. After practice, the participants wore the MYO armband and calibrated the sensor. The participants then put on seven markers on the designated positions and placed the pulse sensor on the right earlobe. Next, we adjusted the monitor height and angle to place the participant's eyes within the eye tracker tracking field. The calibration of the eye tracker was performed as the gameplay logging system was initiated. Before the actual gameplay, we asked the participants to stay calm and still for a minute to measure the their base states of pulse and MYO sensors. Once the base measurements were complete, the actual gameplay began, and the participants played the game for the next 30 minutes. We collected 7.3 GB of data in total and 75.9 GB of high-resolution video recordings.

5 DATA ANALYSIS

We analyze the obtained dataset to verify the conjectures (Table 1). More specifically, we propose quantitative metrics that can verify the conjectures, which are divided into four categories: aiming, character's movement, physical skills, and device and settings (see Table 3). This section first describes the preprocessing of the data, then explains in detail what the proposed metrics are, and explains the data analysis results based on those metrics. Note that the Mann-Whitney U test was used to assess the statistical significance ($p < 0.05$) as the data are not normally distributed.

5.1 Data Preprocessing

5.1.1 Time synchronization. Timestamp synchronization was performed by compensating for the physical delays of the six sensors and the software delay of the in-game data logger. For the dual mouse, analog input keyboard, and pulse sensor, we measured the absolute duration from the physical input until the brightness change of the light-emitting diode (LED) or display. The duration was recorded using a high-speed camera, the Sony RX100 MK5, with a frame rate of 960 frames per second (FPS). We calculated the delay by measuring the time difference in the dual-sensor mouse data for the motion capture and EMG sensors. Moreover, we compared the timestamp for each data point that a specific event (e.g., the press of a mouse button) was simultaneously logged. Detailed

Table 2: Measured delay for time synchronization

	Dual Sensor Mouse	Analog Input Keyboard	Motion Capture Sensor	Eye Tracker	EMG Sensor	In-Game Pulse Sensor	Data Logger
Dual Sensor Mouse			11.64		16.13		23.7
Monitor	38.133	24.06		60.0		45.83	

methods for measuring the delay using each of the sensors with the dual-sensor mouse are explained below.

(1) Dual-sensor mouse: When the user clicks the left mouse button, the display color switches from black to white or vice versa. We recorded the delay from the deepest click stroke until the brightness change using a high-speed camera. The average delay measured 38.13 ms (*SD* = 5.832, *N* = 20). We evaluated the same number of tasks for the condition of black to white and white to black.

(2) Analog input keyboard: When the user presses a key, the display color switches from black to white or vice versa. We recorded the delay from the deepest keystroke until the brightness change using a high-speed camera. The average delay measured 24.06 ms (*SD* = 4.07, *N* = 20). We evaluated the same number of tasks for the condition of black to white and white to black.

(3) Motion-capture sensor: A marker was attached to the user's right index finger tip. Then, we recorded the marker's z-axis value (orthogonal to the mouse pad) and the mouse click logs. We calculated the delay by comparing the timestamps of the moment when z was minimum in the motion-capture data and the moment when the button-press event was logged into the mouse data. The average delay measured 11.64 ms (*SD* = 14.77, *N* = 25).

(4) Eye tracker: We recorded eye images on the iViewRED software interface and the experimenter's pupil movement in a scene with a high-speed camera. Then, we measured the delay from the pupil movement until the change in eye images. The average delay measured 60 ms (*SD* = 15.47, *N* = 20).

(5) Electromyography sensor: The user wearing an MYO armband dragged the mouse quickly in the vertical direction from the direction in which the user was sitting. Since the MYO armband contains an IMU sensor, we could compare the timestamps when each sensor's acceleration is 0. The average delay of the IMU sensor from the mouse sensor measured 16.13 ms (*SD* = 15.57, *N* = 23).

(6) Pulse sensor: The pulse sensor contains an optical sensor for a heart-rate monitor. When the sensor detects light, it changes the signal value provided by the pulse sensor. We placed a pulse sensor attached to an LED light source into a dark box. Then, we measured the delay from turning the LED on until the brightness change using a high-speed camera. The average delay measured 45.83 ms (*SD* = 3.423, *N* = 5).

(7) In-game data: We compared the timestamps when the *weapon fire* event was logged in the in-game data and the moment when the left button-press event was logged in the mouse data. We performed a delay test for 20 shots to compare the delay between the mouse and in-game data with all connected sensors. The average delay was 23.7 ms (*SD* = 9.12, *N* = 20).

5.1.2 Motion-capture data labeling. We used 12 markers in this experiment, as depicted in Figure 4. (b): two markers on the top side of the keyboard (4 and 8), two markers on the left hand (5) and wrist (6), three markers on the index finger of the right hand (9, 10, and 11), two markers on the right wrist and the elbow (12 and 7), and three markers attached to each corner of the mouse pad (1, 2, and 3), except for the lower left part to avoid interfering with the player's arm movement. We labeled the markers based on their size and location for each frame during the game. We put the largest markers (1, 2, 3, 4, and 8) on the keyboard and mouse pad, and the medium-sized markers (5, 6, 7, and 12) on the left hand, right wrist, and right elbow. The smallest three markers (9, 10, and 11) were attached to the right index finger. We first classified the two left-most two of the largest markers as keyboard markers (4 and 8), and the three right-most markers as mouse pad markers (1, 2, and 3). The two medium-sized markers on the left half were labeled with the left wrist and hand (6 and 5) in order by proximity to the body. Finally, the medium and smallest markers on the right half were labeled with the right elbow, wrist, and index finger in order by proximity to the body. We only labeled frames with all 12 markers present and the frames during the gameplay.

5.1.3 Electromyography data normalization and filtering. The signal from the MYO band contains noise and may appear differently depending on various factors, such as the participant's hair, fatty tissue, or sweat [1]. Hence, we performed two types of preprocessing, which are common operations conducted before analyzing the EMG data: normalizing and filtering [11]. First, we normalized the data from the eight sensors to a range from -1 to 1. Then, the Gaussian kernel filter (*sigma*=3) was used to remove the noise.

5.1.4 Classification of combat process. Regarding submovement segmentation, according to the adaptive model theory, all voluntary continuous movements performed by a human, such as a tracking task, comprise a concatenated sequence of submovements [37]. In our study, we used a mouse speed profile to segment the player movement into submovements. The speed profile was obtained in units of pixels per milliseconds by dividing the cursor dx and dy by dt. Because noise generated from sensing hardware reduces the accuracy or precision of a signal, we used a Gaussian kernel filer (σ=3) to reduce the noise. As the local minimum point of the speed profile is considered the boundary separating submovements, we used the `findpeaks` function of `MATLAB` to determine the local maximum points of the inverted speed function. Then, we found the last minimum point right before the shooting as presented in Figure 4. (a).

Regarding shooting, during gameplay, a player typically repeats a sequence of actions, including shooting a weapon. Using the weapon fire events, we grouped a sequence of firing events into one shooting chunk if the interval between the weapon fire events is shorter than 1000ms. When we plotted the histogram of the

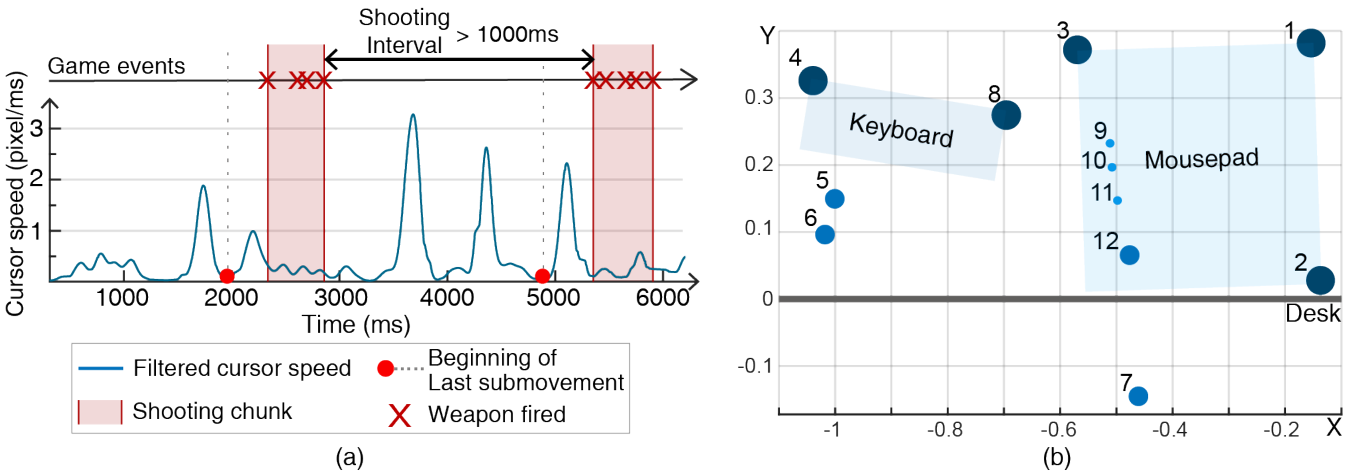

Figure 4: (a) Classification of the last submovement and shooting chunk from the combat process and (b) marker locations from the captured motion data

intervals between 'weapon fire' events, we found out that most were distributed within 1000 ms. Thus, we empirically chose the time window to 1000 ms. If the interval is longer than 1000ms, we defined it as an event in the subsequent shooting chunk.

5.2 Descriptive Statistics

We logged the data from 1097 rounds of the game, and the total playtime for 16 participants at 30 minutes each was 480 minutes. During the 30 minutes, the number of effective shots generated by the professional players was 158.63 (SD = 14.52) on average, and was 142.44 (SD = 13.64) for the amateur players. Even with the same amount of time, professional players generated more effective shots than amateur players (p = 0.03). In addition, the two groups exhibited differences in reaction time for shooting, which is the time interval from the enemy's appearance to the first shooting when using a sniper rifle (p = 0.02). The average reaction times for the professional and amateur groups were 459.54 ms and 504.36 ms, respectively. When using the assault rifle, no significant difference was found in the reaction time between the two groups.

The two groups did not exhibit any significant difference in KDR because the opponent player was selected within the group to replicate the real-world play environment. For example, in Overwatch, the average KDR of the top 50 players in the Grandmaster tier is 2.72 (SD = 0.44), and the average KDR of the top 50 in the platinum tier is 2.65 (SD = 0.55), which is not significantly different [8]. In our experiment, the average KDR was 1.47 (SD = 0.92) for the professional player and 1.14 (SD = 0.79) for the amateur player group.

5.3 Metrics on Aiming

Regarding the aiming skills of FPS players, this study proposes five general metrics: (1) enemy-crosshair stickiness, (2) angular velocity of the mouse, (3) amplitude of the recoil compensating movement, (4) duration of shooting, and (5) force inefficiency. This section describes each metric in detail and presents the analysis results of each metric.

5.3.1 *Enemy-Crosshair Stickiness.*

Tracking is an aiming style that keeps the enemy's position in the middle of the crosshair. It is one of the skills used with guns that are fired continuously, such as the AK-47 of M4-A4. It becomes easier to shoot the enemy quickly and consistently with a closer distance between the enemy's location and the player's crosshair. To compare players' tracking performance, we calculated the average distance between the center of the player's crosshair and the enemy's head and body location.

5.3.2 *Angular Velocity of Mouse.*

The flicking aim is another aiming style that snaps the wrist to move the crosshair to the enemy's position. If the enemy is moving erratically, it is difficult to accurately follow their position with tracking aim. Hence, the player must quickly move the crosshair and hit the moving target before the opponent can react. In general, as the plan for how to click on the target occurs in the last submovement [38], we assumed that the plan for how to shoot the opponent and how to move the mouse or crosshair for the shooting would also take place in the last submovement. We calculated the average angular velocity of the mouse (ω_{mouse}) during the last submovement to measure the degree of wrist snapping, which can be calculated as follows:

$$\omega_{mouse} = \frac{d\theta}{dt} = d\left(\frac{dX_{front} - dX_{rear}}{r_{mouse}}\right)/dt$$

where dX_{front} and dX_{rear} are the dx logged from the front and rear sensors of the mouse. In addition, r_{mouse} is the distance between the two sensors, 72 mm in our experiment [26].

5.3.3 *Amplitude of the recoil compensating movement.*

In general, most assault rifles used in FPS games have recoil, and some guns have their own patterns. However, for most guns, the muzzle moves upwards if a player keeps firing, similar to the movement in a realistic situation. Therefore, some have conjectured that players must move the mouse in the opposite direction of the weapon's recoil to compensate for the kicks during shooting. To compare the amplitude of the recoil compensating movement, we measured the average dy of the mouse during shooting.

Table 3: Mean, standard deviation, and statistical significance of the two groups. The Mann-Whitney U test was used to test the statistical significance at the 5% level.

Metrics		Professional		Amateur		M.W. stat		
		M	SD	M	SD	U	p	
Descriptive Statistics								
Number of effective shots per minute		5.29	0.48	4.75	0.45	14	0.03	●
Reaction time of shooting – Assault rifle (ms)		426.77	30.96	451.26	49.30	21	0.14	
Reaction time of shooting – Sniper rifle (ms)		459.54	29.10	504.36	45.15	12	0.02	●
Aiming								
A-1	Enemy-crosshair stickiness – Assault rifle & shooting (px)	22.56	5.98	16.19	4.49	11	0.02	●
	Enemy-crosshair stickiness – Assault rifle & no shooting (px)	64.89	25.01	72.06	28.09	28	0.36	
A-2	Angular velocity of mouse – Assault rifle (radian/ms)	0.03	0.02	0.01	0.01	8	0.01	●
	Angular velocity of mouse – Sniper rifle (radian/ms)	0.05	0.02	0.03	0.02	13	0.03	●
	Angular velocity of mouse ∗ sensitivity – Assault rifle	0.03	0.01	0.03	0.02	28	0.36	
	Angular velocity of mouse ∗ sensitivity – Sniper rifle	0.05	0.02	0.06	0.04	32	0.48	
A-3	Amplitude of the recoil compensating movement – Assault rifle (px)	190.81	162.31	62.48	62.50	11	0.02	●
	Duration of shooting (tapping and spraying) – Assault rifle (ms)	232.12	81.65	145.02	53.73	12	0.02	●
A-4	Force inefficiency – Assault rifle	4.13	0.47	4.57	0.58	16	0.05	
	Force inefficiency – Sniper rifle	2.73	0.09	2.97	0.28	8	<0.01	●
Character Movement								
M-1	Entropy of pressed keys – Movement combination	1.51	0.12	1.63	0.29	15	0.04	●
M-2	The number of stuttering step pattern (ADA or DAD)	5547	4518	4873	7908	20	0.11	
	The number of crouching pattern (A/S/D/W + Ctrl)	677	302	349	226	12	0.03	●
M-3	Reload efficiency – Assault rifle	7.61	1.33	7.33	2.68	31	0.48	
	Reload efficiency – Sniper rifle	0.84	0.25	0.82	0.24	30	0.44	
Physical Skills								
P-1	Rotation ratio of elbow and wrist (elbow/wrist)	2.95	2.21	1.22	0.78	15	0.04	●
P-2	Muscle activity	1.07	0.12	1.11	0.14	24	0.22	
P-3	Duration of fixation – visible (ms)	713.71	264.99	482.40	220.89	16	0.05	
	Duration of fixation – invisible (ms)	559.73	217.91	321.09	89.13	8	<0.01	●
	Number of saccades – visible (count/s)	2.51	0.25	2.52	0.68	23	0.19	
	Number of saccades – invisible (count/s)	2.75	0.28	2.58	0.84	30	0.44	
P-4	Composure – beats per minute (under30/over30)	0.99	0.02	1.01	0.03	23	0.19	
	Composure – interbeat interval (under30/over30)	1.01	0.03	0.98	0.03	25	0.25	
	Composure – pupil diameter (under30/over30)	1.01	0.04	1.02	0.02	29	0.40	
Device and Settings								
D-1	Used area of mousepad (cm^2)	1479	845	473	604	7	0.01	●
D-2	Keyboard perpendicularity (degree)	18.32	16.99	2.85	3.64	9	0.01	●

5.3.4 Duration of shooting. There are two main methods exist to shoot a weapon: tapping and spraying. Tapping is a single or short burst of controlled weapon fire using discrete mouse clicks. In contrast, spraying is a stacked or a more extended period of weapon fire typically done by continuously pressing the mouse. To measure the two types of shooting, we calculated the average time intervals of individual mouse clicks. A single mouse click interval was measured to be the time between pressing and releasing the mouse. A shorter interval is closer to tapping, ans a longer interval is closer to spraying.

5.3.5 Force inefficiency. In FPS games, the mouse sensitivity is a crucial factors that affects aiming. According to some conjectures, professional players have their own optimal mouse sensitivity, and amateur players have also tried to discover this. After the setting a specific sensitivity, players practice learning movements that fit the sensitivity. Players who are well trained and accustomed to their sensitivity can move the mouse exactly where they desire at once. To measure how familiar the players are with their mouse sensitivity settings, we calculated the force inefficiency by counting the number of the zero crossings in a mouse's acceleration function [29] for 460.48 ms before shooting. We set the interval to 460.48 ms because the players fired the weapon 460.48 ms on average after the enemy appeared.

5.3.6 Results. Unlike our expectations, when shooting and tracking the target (A-1), the average distance between the target's head and player's crosshair for professional players was farther than that for the amateur players. The average distances for the professional and amateur players were 22.56 px and 16.19 px, respectively. Before shooting, both groups aimed at a similar distance from the target at 64.89 px and 72.06 px. In this regard, by comparing the amplitude of the recoil compensation (A-3), we found that professional players tend to drag the mouse down more than amateur players during shooting. The professional and amateur players dragged the mouse down an average of 190.81 px and 62.48 px, respectively. We also found that professional players exhibit faster angular velocity (A-2) for both cases with assault rifles (0.03 rad/ms) and sniper rifles (0.05 rad/ms) compared to amateur players at 0.01 rad/ms for assault rifles and 0.03 rad/ms for sniper rifles. However, the angular velocity multiplied by the mouse sensitivity for each player demonstrated no significant difference. The professional players' mouse movement was more efficient in terms of force efficiency (A-5). The force inefficiency of professional players is 2.73, which means that professional players changed the moving direction of the mouse less than amateur players at 2.97. In addition, the shooting duration was 190.81 ms and 62.48 ms for the professional and the amateur groups, respectively. The professional players displayed a relatively close duration to the spraying shooting.

5.3.7 Discussion. Several interesting points were found in the results for aiming. First, contrary to the conjectures, the enemy-crosshair stickiness during the shooting was shorter for amateur players, which is because professional players dragged the crosshair 128.33 px farther down to compensate for gun recoil, whereas amateur players did not. For nonshooting situations, where the recoil compensation is unnecessary, two groups did not exhibit a significant difference in the enemy-crosshair stickiness. In CS:GO, the center of the crosshair and the coordinates of the fired bullet are slightly different due to the recoil of the gun. Although we could not locate the exact trajectory of the fired bullets from the in-game data, the recoil makes the bullets spatter to higher positions than the crosshair center. Because this is a CS:GO game-dependent feature, further verification of the metric is needed using another game.

Second, a significant difference in the angular mouse velocity was found between the two groups, which can be a clue concerning flicking aims. However, when we compensated for the mouse sensitivity, the angular mouse velocity was not significant, as shown in the A-2 in Table 3. Therefore, the crosshair movement speed in the game was similar in both groups, but, in reality, professional players move the mouse with a higher angular speed than amateur players. Professional players use a lower sensitivity because it allows the player to control movements more precisely. In other words, they can move the same distance with a higher resolution. In our experiment, this was verified through professional players using a wider area of the mouse pad by 1006 cm^2 than amateur players (D-1 in Table 3).

Third, professional players demonstrated less force inefficiency when using sniper rifles, which means that professional players switched the mouse direction less often than amateur players to aim at an enemy because they did not need corrective movements. Typically, players who are more trained and familiar with the mouse sensitivity can aim at the desired position at once without the corrective movements, because they tend not to undershoot or overshoot. From the results, professional players are more capable than amateur players in finding suitable sensitivities. However, no significant difference was found when using the assault rifle. In general, mouse sensitivity is considered more important when using a flicking aims, but assault rifle usage requires more tracking aim than flicking aim.

5.4 Metrics on Character's Movement

We proposes three metrics for the character's movement: (1) entropy of combinations of keys, (2) frequency of movement combinations for accurate aiming, and (3) reload efficiency. Details for each metric and results are provided below.

5.4.1 Entropy of combinations of keys. In FPS games, neutralizing an enemy's attack is just as important as killing an enemy. To do so, players must make dynamic and unpredictable movements employing various combinations of keys. To measure the random combinations of keys, we calculated the entropy based on compression algorithms [36], a widely used approach in calculating the entropy of a symbolic sequence. The algorithm states that symbolic sequence entropy can be estimated by finding the shortest mismatch in the sequence, which can be compressed without information loss.

5.4.2 Frequency of movement combinations for accurate aiming. Moving while shooting significantly reduces aiming accuracy. Therefore, professional players often make stuttering idesteps, crouch, or walk right before shooting to stabilize the aiming. We measured two types of movement combinations for stabilizing the aiming suggested by the community: stuttering side steps and a crouching pattern while moving. For stuttering side steps, we measured the frequency of the key combinations A-D-A (left-right-left) or D-A-D (right-left-right). For crouching, we measured the frequency of combinations of each movement keys (A, S, D, and W) and the crouching key (Ctrl). We counted the combinations of each movement from the shortest sequence patterns extracted from the Lempel-Ziv compression algorithm to obtain the two frequencies.

5.4.3 Reload Efficiency. The gun magazine bullets are limited in FPS games so that the players must reload when the bullets runs out. Because players cannot shoot while loading, players are in a dangerous situation if they encounter an opponent while loading. The conjecture suggests that professional players do not reload habitually and know the right time to reload. To measure the reload efficiency, we counted the number of bullets left when the player reloads.

5.4.4 Results. Regarding the character movement, the two groups exhibited a significant difference in the entropy of the pressed keys (M-1) and the number of combinations that presses the crouching key right before shooting (M-2). The average entropy for professional players was 1.51, which means that the specific movement combination pattern was repeated many times compared to the amateur players (M = 1.63). The professional players used crouching combinations more often than amateur players (p = 0.03). Nevertheless, the number of stuttering combinations and the reload efficiency (M-3) exhibited no significant difference.

5.4.5 Discussion. In an FPS game, dodging the opponent's bullets is as vital as aiming accurately. Therefore, many communities recommend moving randomly so that the opponent cannot anticipate the movements. According to the entropy calculation for the M-1 conjecture, a lower entropy value indicates a higher compression rate for the key sequences, resulting from a higher repetition of specific movement patterns. Thus, the lower entropy of the professional players means that their movements are more predictable than those of amateur players. In other words, more similar patterns were observed in the movement sequence of the professional players.

The analysis results of the frequency of movement combinations for accurate aiming indicate that professional players tend to mix crouching in the middle of their movements more often (p = 0.03), yet making a similar level of stuttering steps compared to the amateur players. Crouching right before shooting is a habit recommended by many skillful players because it lowers the chance of a headshot from the opponent and, increases the player's shootnig accuracy resulting in more effective shots. Moreover, more frequent generation of the same movement patterns, such as crouching before shooting, may explain the lower entropy of the professional players' in the previous observation.

5.5 Metrics on Physical Skills

For the physical skills, we proposes four general metrics: (1) rotation ratio of elbow and wrist, (2) arm electromyography (EMG) activity, (3) duration of fixation and the number of saccades, and (4) composure. Details for each metric and results are provided below.

5.5.1 Rotation ratio of elbow and wrist. There are two primary methods exist to aim: arm and wrist aiming. Wrist aiming using delicate muscles allows the player to control the mouse more finely. However, according to conjecture, most professional players primarily use their arms to aim because, assuming the players use the same sensitivity, arm aiming is advantageous for quickly dragging the mouse to a distant location at once and puts less strain on the muscles. To measure the arm and wrist usage proportion, we calculated the elbow rotation angle (the horizontal vector of the desk to the lower arm) and wrist (lower arm to hand). Then, we compared the elbow and wrist rotation ratio by dividing the rotation angle of the wrist by elbow rotation angle.

5.5.2 Arm Electromyography (EMG) Activity. The conjecture suggests building muscle memory in the arm and wrist to aim precisely. Traditionally, the term 'muscle memory' has been used by FPS players as a synonym for motor learning, which is the acquisition of motor skills to enhance performance [35]. We analyzed the EMG to measure the magnitude of muscle activation of the arm and wrist and compared the magnitude of muscle activation when shooting and not shooting.

5.5.3 Duration of fixation and the number of saccades. Eye movement tracking data are commonly used to measure the system usability [21] or user proficiency [28]. Fixation and saccades are essential metrics for eye movement data analysis. Among several methods, we used a robust and highly accurate method, dispersion-threshold identification (I-DT) [42], to separate and label the fixation and saccades. The I-DT method requires two thresholds: the duration and dispersion thresholds. Typically, the fixation duration is at least 100 ms; thus in the dispersion-based identification technique, the minimum duration threshold is set as 100 ms to 200 ms [49]. Regarding the dispersion threshold, if we know the distance from the eye to the screen, the dispersion threshold can be set to cover $1/2^{\circ}$ to 1° of the viewing angle [42]. In our experiment, the distance between the monitor and user was about 400 mm. Therefore, we set the duration threshold at 200 ms and the dispersion threshold as 13.20 px (3.49 mm). We divided the situation into when the enemy was visible and invisible and calculated the fixation duration and the number of saccades.

5.5.4 Composure. During the FPS play, it is essential to maintain composure. Especially in urgent situations, such as encountering an enemy, shooting at an enemy, or having low health points, a more composed player has a higher chance of survival. We calculated the player's composure using three measurements, *average heartbeat*, *average inter-beat interval*, and *average pupil diameter*. Each measurement was calculated for a specific game situation, where a player's health points were lower than 30. Because the sensor's absolute magnitude for each participant varies, we calculated the sensor measurement ratio, with fewer than 30 health points and more than 30 over health points.

5.5.5 Results. In the physical skills category, the elbow and wrist rotation ration exhibited a significant difference (p = 0.04). The professional players' rotation ratio was 2.95, which demonstrates that the elbow rotation angle is about three times larger than that of the wrist. In contrast, the rotation ratio for amateur players wa 1.22, which means the amateur players used their wrists and arms in a similar proportions. When the enemy was invisible, the two groups demonstrated a significant difference in the fixation duration ($p <$ 0.01) at 559.73 ms for professional players and 321.09 ms for amateur players. The professional players had a 238.64 ms longer duration than the amateur players, but the number of saccades was similar. No significant difference was found for both the fixation duration and number of the saccades in the visible situation.

The muscle activity of the lower arm (P-2) measured by the MYO band was not significantly different for shooting and non shooting situations. Moreover, they maintained a constant number of BPM, regardless of the number of health points remaining. Likewise, no significant differences in the inter-beat interval and pupil diameter were found (P-4).

5.5.6 Discussion. By comparing the elbow and wrist rotation ratio, we found that the professional players tend to use arm rotation more than wrist rotation when aiming, verifying the conjecture P-1. However, both groups had similar levels of muscle activation when shooting and not shooting. According to the fixation duration, professional players stared at a specific position longer when the enemy was invisible. However, the conjecture recommends looking around as quickly and as much as possible, contrary to the findings. For composure, we could not find any significant difference in the metrics.

We summarize the following two reasons for such results: the small map size and the different experimental environment than the actual competition. First, the map in this experiment was small and straightforward, so it was relatively easy to locate the opponent, which means that the player did not need to scan their surroundings diligently to find the enemy.

In the post-experiment survey, some participants answered that it was easier to predict where the enemy would appear than in the actual game. In addition, because the experiment did not generate psychological pressure comparable to the real competition, we presume no fluctuations occurred in the players' composure.

5.6 Metrics on Device and Settings

Regarding the metrics on device and settings, we proposes two metrics: (1) used area of mousepad and (2) keyboard perpendicularity. Details for each metric and results are provided below.

5.6.1 Used Area of Mousepad. The DPI and in-game sensitivity are measurements of mouse sensitivity. Using low sensitivity means the player can move the same distance with a higher resolution when using the mouse in the game. Hence, low sensitivity offers an advantage to accurate and precise aiming, and most professional players used low sensitivity. Because it is recommended to use the mouse pad widely with low sensitivity, we compare the square measure of the used mousepad area. The square measure was calculated using the maximum and minimum x and y values of the right index finger tip.

5.6.2 Keyboard Perpendicularity. It is inevitable to use the keyboard in the same posture for a long time when playing FPS games. However, the wrist flexion or extension angle and radial and ulnar deviation angle affect carpal tunnel pressure [41]. Several communities have advised placing the keyboard perpendicular to the player's sitting direction due to stress on the wrist. We calculated an angle between the keyboard and the horizontal vector to the desk, which is assumed to be the player's sitting direction.

5.6.3 Results. The total used area of the mousepad (D-1) and the perpendicularity of the keyboard (D-2) displayed significant differences between the two groups. The average sensitivity of the professional players was 0.91 (SD = 0.25) for a assault rifle and 0.99 (SD = 0.27) for a sniper rifle. Amateur players set the sensitivity to 2.03 (SD = 0.85) for a assault rifle and 2.10 (SD = 0.27) for a sniper rifle. These values were higher than those for the professional players. Professional players used an area of 1479 cm^2 on average, and the amateur players used an area of 473 cm^2. The keyboard of professional players was placed at 18.32° from the horizontal direction of the desk, whereas the amateur players placed it at 2.85°.

5.6.4 Discussion. Most professional players used lower sensitivities than the amateur players, which also resulted in using a larger mousepad area, as indicated in the result for D-1. Therefore, we confirm the suggestions from many communities to lower the sensitivity and use a larger mouse pad to increase aiming accuracy. The results also confirmed that professional players tend to use the keyboard at a larger angle.

5.7 Post-Experiment Questionnaire

In the post-experiment survey, participants answered questions asked about the commonalities and differences between their usual FPS game and the experiment they conducted, and what they had considered most crucial to win the game. Both amateur and professional players responded that the aiming method was similar because the rifles used in the experiment are familiar. They also responded that the map structure and the presence of recoil on the rifles were similar to the actual game. In particular, 5 out of 8 professional players mentioned that they had to react quickly to the opponent's movements like in the actual FPS game. For the difference, the two groups replied that the rifles in CS:GO had a greater recoil amplitude than the weapons in other games. Because of this feature, the players had to stop moving before shooting. In addition, professional players responded that it was awkward that they could not use their own equipment, such as a mouse or keyboard. Lastly, professional players pointed out the four most important strategies to win the game: adapting to mouse sensitivity, flicking accurately, predicting the opponent's position, and compensating for the recoil. Among them, interestingly, adapting to mouse sensitivity and compensating the recoil were matched to the A-4 and A-3, which exhibited statistically significant difference.

6 CONCLUSION AND LIMITATIONS

This study quantitatively verified and statically examine conjectures on FPS performance from online communities. We developed a comprehensive gameplay logging system and collected a dataset from professional and amateur players playing the commercial FPS CS:GO. We confirmed the conjectures A-3, A-4, M-2, P-1, D-1, and D-2 to be meaningful by analyzing the collected data. Although A-1, A-2, M-1, and P-3 were statistically significant, we withheld confirmation because they were either partially verified or went against the conjectures.

In more detail, the combat skills of professional players we found in the experiment can be summarized as follows: while shooting using a assault rifle, professional players tended to lower the crosshairs downwards with a greater amplitude than amateur players did to compensate for recoil. Also, professional players showed lower force inefficiency than amateur players when using sniper rifles. This means that professional players were more familiar with their sensitivity and needed less corrective movement to aim accurately. By comparing the rotation ratio of the elbow against the wrist, we found out that professional players used more elbows than the wrist during aiming, and the professional players' rotation ratio was higher than the amateurs'. For character movement, professional players showed a pattern of crouching just before shooting more frequently than amateur players. During the experiment, professional players used a larger area of the mousepad than amateur players and placed the keyboard more vertically.

This study also reveals two interesting points about the origin of the high performance of professional players. First, it seems that the combat strategy of professional players is largely determined by the settings of the game interface such as the input device [25, 26, 30, 32]. For example, looking at the results of the A-2 related metrics, the reason professional players move their mouse faster than regular players may simply be because they have a lower mouse sensitivity setting. Further research is needed on whether the behavior of professional players will significantly differ from those of amateur players even in the same interface setting.

Second, from the fact that professional players showed lower reaction times than amateur players during the gameplay, we can infer that the high performance of professional players may be due to their innate cognitive ability. This is related to the issue of whether there is a difference in innate talent among individuals that cannot be overcome by training in e-sports. This hypothesis could be tested in the future through controlled lab experiments using cognitive tasks related to FPS gameplay, such as pointing [34], choice-reaction [39], temporal pointing [33], and moving-target acquisition [31, 38]

However, this study has some obvious limitations in terms of external validity and generalizability, which should be supplemented by further studies in the future. First, the players recruited in this study did not have the experience of playing CS:GO, the FPS game used in the experiment. Although we recruited players who are believed to have universal skills that can be applied to various FPS games and also gave them enough time to practice during the experiment, this could have led to lower statistical power in this study, considering that professional players generally spend much longer training their main games.

Second, due to the peculiarity of CS:GO, the FPS game used in this study, the findings in this study may not be able to generalize to other FPS games. For example, as discussed earlier, the fact that the crosshair and bullet trajectory not being completely matched in the game can influence the conjecture A-1. However, the performance metrics we have proposed do not depend on the type of FPS game, so if research on other FPS games based on the same metrics is carried out later, the limitation could be supplemented.

The controlled experimental setup also presented several limitations. First, professional players are typically keen about using their own equipment, such as the mouse and keyboard; however, we had to provide a special mouse and keyboard to log behavioral data. Second, although we used small motion-capture markers and ear-clip type pulse sensor instead of the finger-clip type, the sensors may have affected to the participants' movement. Third, gameplay's restricted circumstances, such as a small map and one on one fight, only required simplistic movements. Fourth, the experiment in the lab could not generate sufficient immersion similar to real gameplay for the players to exhibit psychological changes, such as BPM or pupil diameter. Finally, this study did not fully consider the fact that many FPS games assume multiplayer settings. In a multiplayer setting, a team's success is affected not only by the individual player's combat skills, but also by how well players communicate with each other [10, 51, 52]. In the future, a more generalized experiment addressing these limitations may increase the external validity.

ACKNOWLEDGMENTS

This research was funded by National Research Foundation of Korea (2020R1A2C4002146) and Korea Creative Content Agency (R2020040211).

REFERENCES

[1] 2014. Big Data: Raw EMG Free for Developers in December. https://developerblog.myo.com/big-data/
[2] 2014. How to be good at CS:GO for all skill levels. https://steamcommunity.com/sharedfiles/filedetails/?id=121447468
[3] 2017. CS:GO Movement Guide. https://www.youtube.com/watch?v=AGcgQEzCCrI
[4] 2018. Aiming Concepts and Glossary. https://aiming.pro/aiming-concepts-glossary
[5] 2018. Wrist Aiming vs. Arm Aiming in FPS Games. https://aiming.pro/wrist-aiming-vs-arm-aiming-in-fps-games
[6] 2019. 10 Scientifically Proven CS:GO Tips For Better Aim. https://www.youtube.com/watch?v=Vo2yeqQp1Ts
[7] 2020. The Best Settings and Setups in Esports. https://prosettings.net/
[8] 2020. The Best Settings and Setups in Esports. https://overlog.gg/leaderboards
[9] Jim Albert. 2010. Sabermetrics: The past, the present, and the future. *Mathematics and sports* (2010), 3–14.
[10] Sultan A Alharthi, Ruth C Torres, Ahmed S Khalaf, Zachary O Toups, Igor Dolgov, and Lennart E Nacke. 2018. Investigating the impact of annotation interfaces on player performance in distributed multiplayer games. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.
[11] Marco E Benalcázar, Cristhian Motoche, Jonathan A Zea, Andrés G Jaramillo, Carlos E Anchundia, Patricio Zambrano, Marco Segura, Freddy Benalcázar Palacios, and María Pérez. 2017. Real-time hand gesture recognition using the Myo armband and muscle activity detection. In *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*. IEEE, 1–6.
[12] Florian Block, Victoria Hodge, Stephen Hobson, Nick Sephton, Sam Devlin, Marian F Ursu, Anders Drachen, and Peter I Cowling. 2018. Narrative bytes: Data-driven content production in esports. In *Proceedings of the 2018 ACM International Conference on Interactive Experiences for TV and Online Video*. 29–41.
[13] Eugene Bozhenko. 2020. How to Get Better at CS:GO: Pro CS:GO Tips: DMarket: Blog. https://dmarket.com/blog/best-csgo-beginners-tips/
[14] David Buckley, Ke Chen, and Joshua Knowles. 2013. Predicting skill from gameplay input to a first-person shooter. In *2013 IEEE Conference on Computational Inteligence in Games (CIG)*. IEEE, 1–8.
[15] David Buckley, Ke Chen, and Joshua Knowles. 2015. Rapid skill capture in a first-person shooter. *IEEE Transactions on Computational Intelligence and AI in Games* 9, 1 (2015), 63–75.
[16] Gifford Cheung and Jeff Huang. 2011. Starcraft from the stands: understanding the game spectator. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 763–772.
[17] Tamara Clegg, Daniel M Greene, Nate Beard, and Jasmine Brunson. 2020. Data Everyday: Data Literacy Practices in a Division I College Sports Context. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.
[18] danielkrupinski. 2020. Osiris. https://github.com/danielkrupinski/Osiris.
[19] Carlos Dietrich, David Koop, Huy T Vo, and Cláudio T Silva. 2014. Baseball4d: A tool for baseball game reconstruction & visualization. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 23–32.
[20] Joey R Fanfarelli. 2018. Expertise in Professional Overwatch Play. *International Journal of Gaming and Computer-Mediated Simulations (IJGCMS)* 10, 1 (2018), 1–22.
[21] Joseph H Goldberg and Xerxes P Kotval. 1999. Computer interface evaluation using eye movements: methods and constructs. *International journal of industrial ergonomics* 24, 6 (1999), 631–645.
[22] Hi-Im-TwiX. 2020. The importance of crosshair placement, why you're doing it wrong, and how to fix it. https://www.reddit.com/r/VALORANT/comments/hs07zn/the_importance_of_crosshair_placement_why_youre/
[23] Thomas Horky and Philipp Pelka. 2017. Data Visualisation in Sports Journalism: Opportunities and challenges of data-driven journalism in German football. *Digital Journalism* 5, 5 (2017), 587–606.
[24] Kermit. 2019. FUNDAMENTALS OF AIM – VOLUME 3. https://aimlab.gg/fundemantals-of-aim-volume-3/
[25] Sunjun Kim, Byungjoo Lee, and Antti Oulasvirta. 2018. Impact activation improves rapid button pressing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–8.
[26] Sunjun Kim, Byungjoo Lee, Thomas Van Gemert, and Antti Oulasvirta. 2020. Optimal Sensor Position for a Computer Mouse. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.
[27] Alexander Korotin, Nikita Khromov, Anton Stepanov, Andrey Lange, Evgeny Burnaev, and Andrey Somov. 2019. Towards Understanding of eSports Athletes' Potentialities: The Sensing System for Data Collection and Analysis. *arXiv preprint arXiv:1908.06403* (2019).
[28] Benjamin Law, M Stella Atkins, Arthur E Kirkpatrick, and Alan J Lomax. 2004. Eye gaze patterns differentiate novice and experts in a virtual laparoscopic surgery training environment. In *Proceedings of the 2004 symposium on Eye tracking research & applications*. 41–48.
[29] Byungjoo Lee and Hyunwoo Bang. 2013. A kinematic analysis of directional effects on mouse control. *Ergonomics* 56, 11 (2013), 1754–1765.
[30] Byungjoo Lee and Hyunwoo Bang. 2015. A mouse with two optical sensors that eliminates coordinate disturbance during skilled strokes. *Human–Computer Interaction* 30, 2 (2015), 122–155.
[31] Byungjoo Lee, Sunjun Kim, Antti Oulasvirta, Jong-In Lee, and Eunji Park. 2018. Moving target selection: A cue integration model. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 230.

[32] Byungjoo Lee, Mathieu Nancel, Sunjun Kim, and Antti Oulasvirta. 2020. AutoGain: Gain Function Adaptation with Submovement Efficiency Optimization. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[33] Byungjoo Lee and Antti Oulasvirta. 2016. Modelling error rates in temporal pointing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1857–1868.

[34] I Scott MacKenzie. 1992. Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction* 7, 1 (1992), 91–139.

[35] Richard Magill and David Anderson. 2010. *Motor learning and control*. McGraw-Hill Publishing.

[36] Marcelo A Montemurro and Damián H Zanette. 2011. Universal entropy of word ordering across linguistic families. *PLoS One* 6, 5 (2011), e19875.

[37] PD Neilson, MD Neilson, and NJ O'dwyer. 1988. Internal models and intermittency: A theoretical account of human tracking behavior. *Biological cybernetics* 58, 2 (1988), 101–112.

[38] Eunji Park and Byungjoo Lee. 2020. An Intermittent Click Planning Model. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[39] Roger Ratcliff and Gail McKoon. 2008. The diffusion decision model: theory and data for two-choice decision tasks. *Neural computation* 20, 4 (2008), 873–922.

[40] Stuart Reeves, Barry Brown, and Eric Laurier. 2007. Experts at play: Understanding and designing expert skill.

[41] David M Rempel, Peter J Keir, and Joel M Bach. 2008. Effect of wrist posture on carpal tunnel pressure while typing. *Journal of orthopaedic research* 26, 9 (2008), 1269–1273.

[42] Dario D Salvucci and Joseph H Goldberg. 2000. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications*. 71–78.

[43] Byungjoo Lee Shinhoo Park, Sunjun Kim. 2018. A Processing Library for WootingKeyboard.

[44] Shroud. 2018. USING AN EYE TRACKER IN PUBG. https://www.youtube.com/watch?v=PYlLNyX2nCE&t=541s

[45] Anton Stepanov, Andrey Lange, Nikita Khromov, Alexander Korotin, Evgeny Burnaev, and Andrey Somov. 2019. Sensors and Game Synchronization for Data Analysis in eSports. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, Vol. 1. IEEE, 933–938.

[46] Nathaniel Stoltz. 2014. *Sabermetrics over time: Persuasion and symbolic convergence across a diffusion of innovations*. Ph.D. Dissertation. Wake Forest University.

[47] Struth. 2020. Aiming Essentials 2 - Wrist & Arm Aiming. https://themeta.com/blog/2020/2/8/aiming-essentials-2-wrist-amp-arm-aiming

[48] Boris B Velichkovsky, Nikita Khromov, Alexander Korotin, Evgeny Burnaev, and Andrey Somov. 2019. Visual Fixations Duration as an Indicator of Skill Level in eSports. *arXiv preprint arXiv:1906.01699* (2019).

[49] Heino Widdel. 1984. Operational problems in analysing eye movements. In *Advances in psychology*. Vol. 22. Elsevier, 21–29.

[50] Emma Witkowski. 2012. On the digital playing field: How we "do sport" with networked computer games. *Games and Culture* 7, 5 (2012), 349–374.

[51] Jason Wuertz, Sultan A Alharthi, William A Hamilton, Scott Bateman, Carl Gutwin, Anthony Tang, Zachary Toups, and Jessica Hammer. 2018. A design framework for awareness cues in distributed multiplayer games. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–14.

[52] Jason Wuertz, Scott Bateman, and Anthony Tang. 2017. Why players use pings and annotations in Dota 2. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 1978–2018.

[53] zandurs. 2019. Why do pro players play with their keyboards sideways? https://www.youtube.com/watch?v=xU37pZJzSBg

[54] Qiyu Zhi, Suwen Lin, Poorna Talkad Sukumar, and Ronald Metoyer. 2019. GameViews: Understanding and Supporting Data-driven Sports Storytelling. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.