UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

MASTER THESIS No. 3009

QUALITY OF EXPERIENCE ASSESSMENT OF INTERACTION MECHANICS IN VIRTUAL REALITY

Monika Matokanović

Zagreb, June 2022

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

MASTER THESIS No. 3009

QUALITY OF EXPERIENCE ASSESSMENT OF INTERACTION MECHANICS IN VIRTUAL REALITY

Monika Matokanović

Zagreb, June 2022

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Zagreb, 11 March 2022

MASTER THESIS ASSIGNMENT No. 3009

Student: **Monika Matokanović (0054036413)**

Study: Computing

Profile: Computer Science

Mentor: prof. Lea Skorin-Kapov

Title: **Quality of Experience Assessment of Interaction Mechanics in Virtual Reality**

Description:

Modern commercial virtual reality (VR) systems support tracking user movements in six degrees of freedom. This type of interaction with the virtual environment offers an alternative to typical mouse and keyboard controls or standard control devices, allowing VR game developers to achieve physically demanding game mechanics, inspired by real-world movements and actions. Different parameters of specific game mechanics affect the satisfaction or annoyance of end users, with special challenges stemming from different ways of interacting with the virtual world. Your task is to modify and extend a previously developed application designed to test different types of interaction mechanics that are often represented in VR games (shoot, slice). It is necessary to implement the pick-and-place mechanic, enabling a user to handle objects of different shapes, sizes and orientations, and move them from one position to another. Objects can move or be placed in different positions, be occluded by other objects, etc. Within the scope of the application, your task is to implement a user interface that allows the test administrator to configure the parameters of each test scenario. Furthermore, it is necessary to enable the collection of objective performance metrics (e.g., time it took the user to place a particular object in the right place, how precisely the task was performed). Finally, your task is to conduct a user study to examine the impact of selected parameters of interaction mechanics on user experience. All the necessary literature, data, and working conditions will be provided to you by the Department of Telecommunications.

Submission date: 27 June 2022

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 11. ožujka 2022.

DIPLOMSKI ZADATAK br. 3009

Pristupnica:	**Monika Matokanović (0054036413)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentorica:	prof. dr. sc. Lea Skorin-Kapov

Zadatak: **Ispitivanje iskustvene kvalitete interakcijskih mehanika u virtualnoj stvarnosti**

Opis zadatka:

Suvremeni komercijalni sustavi za virtualnu stvarnost (engl. Virtual Reality, VR) zasnovani su na tehnologiji koja omogućava praćenje korisnikovih pokreta u šest stupnjeva slobode. Takva vrsta interakcije s virtualnim okolišem nameće se kao alternativa upravljanju pomoću miša i tipkovnice ili standardnih upravljačkih uređaja te omogućuje razvijateljima VR igara ostvarenje fizički zahtjevnih mehanika igre, inspiranih pokretima i radnjama iz stvarnoga svijeta. Različiti parametri pojedine mehanike igre utječu na zadovoljstvo ili iziritiranost krajnjih korisnika. Vaš zadatak je nadograditi postojeću aplikaciju koja služi za testiranje različitih vrsta interakcijskih mehanika koje su često zastupljene u VR igrama (pucanje, rukovanje sabljom). Potrebno je implementirati mehaniku igre "uzmi i pozicioniraj" (engl. pick-and-place) kako bi korisnik mogao isprobati rukovanje objektima različitog oblika, veličine i orijentacije te ih uzimati s različitih pozicija i stavljati na različite pozicije. Pritom se objekti mogu kretati ili pojavljivati na različitim pozicijama, biti prekriveni nekim drugim objektom i sl. U okviru aplikacije, potrebno je implementirati korisničko sučelje koje omogućuje osobi koja provodi testiranje da konfigurira parametre pojedinog scenarija. Nadalje, potrebno je prikupljati objektivne metrike performansi korisnika (npr. vrijeme potrebno da korisnik postavi određeni objekt na pravo mjesto, koliko je precizno izvršen zadatak). Na kraju, potrebno je provesti korisničku studiju s ciljem ispitivanja utjecaja odabranih parametara interakcijskih mehanika na korisničko iskustvo. Svu potrebnu literaturu i uvjete za rad osigurat će Vam Zavod za telekomunikacije.

Rok za predaju rada: 27. lipnja 2022.

Table of contents

Introduction

The concept of virtual reality (VR) has been around since the 19th century, but the virtual reality technology used today is relatively recent [1]. The VR market is on the rise, with the gaming and entertainment industries holding a large part of that market. Even though the number of VR users is increasing every year, growth is lessened by poor user experience [2].

One of the main design concerns that highly impacts the quality of user experience is developing how a user can interact with the virtual environment [3]. This is especially true for virtual reality applications because they try to replicate real-world interactions, and poorly designed interactions could negatively impact immersion. Finding the optimal interaction mechanic parameters, however, is still an open research challenge. Additionally, games present another challenge in interaction optimization because they must present some difficulty to the player to be entertaining, and overly optimized interaction mechanics would not present enough challenge for the player [4].

The goal of this thesis was to develop a testing framework for the evaluation of different interaction types and parameters. Moreover, a pilot study was conducted, and the results were analyzed to give an understanding of the impact of different parameters on the Quality of Experience (QoE) and to help plan future studies using the developed testing framework.

This thesis is divided into seven main chapters, with the introduction at the beginning and the conclusion at the end. The first chapter gives an introduction to virtual reality and interaction mechanics optimization and types. Additionally, some related studies are mentioned. The second chapter gives a general overview of the testing framework developed, and the third chapter states and describes the technologies used in the development. The fourth chapter gives a detailed description of the testing framework implementation, including descriptions of the gameplay objectives, design, and interaction parameters that can be tested using this framework, as well as the objective data collected for each game. The fifth chapter explains how the QoE testing was executed, and the sixth chapter details both the objective and the subjective data collected in the conducted study, and an analysis of the results is given. The last chapter gives suggestions for future work using the developed virtual reality interaction mechanics testing framework.

The conclusion summarizes the most important findings and results of this thesis. Finally, the additional chapters include a list of references, a summary in both English and Croatian languages, and an appendix which contains the original QoE questionnaire.

1. Interaction mechanics in virtual reality

Virtual reality (VR) is the computer-generated simulation of a three-dimensional environment that enables user interaction in a seemingly real and physical way [5]. The concept of VR dates back to the 19^{th} century and the invention of the stereoscope, an optical device that creates an illusion of a three-dimensional object by combining two two-dimensional images [6]. Since then, there have been several attempts in history at making viable VR technology. For example, Sensorama (Figure 1.1), developed by Morton Heilig in 1956, was a device that simulated a motorcycle ride using a stereoscopic display, moving chair, odor emitters, and a ventilator [7]. Moreover, VPL Research developed the first commercially available VR technology in the 1980s, and they were the first to use the term "virtual reality." This encouraged companies such as Nintendo and Sega to invest in VR technology for video games. In the early 1990s, W Industries developed a VR device called Virtuality that was available in video arcades in the USA. However, because of the high costs and technical limitations of that time, enthusiasm and, subsequently, investments in VR died off [8]. In 2012, Oculus VR developed a virtual reality device that renewed the public's interest in this technology and created a market that has been steadily growing since then [6].

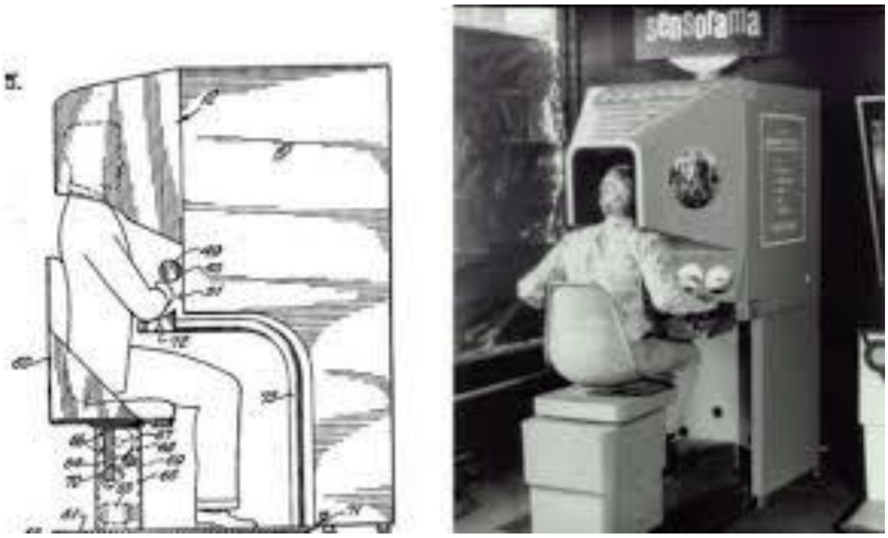

Figure 1.1 Sensorama technical drawing (left) and the pictures shown at the presentation of the experimental product (right) [9].

Today, there are over 171 million estimated active users worldwide, with gaming and entertainment media holding 40.5% of the VR market share [10]. Even though the number of VR users is expected to rise, according to the Perkins Coie survey conducted in 2020 [2], poor user experience is one of the biggest obstacles to the mass adoption of VR technologies.

One of the main design concerns in the development of VR applications is determining game mechanics. Game mechanics are a set of rules by which the player can interact with game objects and make changes in the virtual environment in order to achieve a goal [3]. Interaction mechanics are game mechanics that are controlled by player input [4] and directly control how a user interacts with the virtual environment. Interaction mechanics are a bridge between the user and the application, which causes the quality of these interactions to directly impact the quality of the user experience.

When developing applications for VR, many of the interaction mechanics used in desktop experiences can be reused in VR. However, this approach may not be the best solution for the best user experience. Interactions with objects in virtual reality environments can be more natural compared to desktop experiences because of the immersive nature of VR and the ability to mimic real-life interactions [11].

Modern VR technology is relatively new and interaction with 3D virtual environments that are made possible by this type of technology offers users immersive and interactive experiences. However, given that the design of optimal interaction mechanics is still considered an open research challenge, developers are often forced to rely on guessing and estimating to achieve the best user experience.

1.1. Interaction mechanics optimization

The goal of virtual reality is to emulate real-life interactions, therefore, interactions in VR should be as natural as possible. The user must be provided with a set of interaction mechanics that enable them to focus on the high-level tasks of the VR application instead of focusing on physical activities and interactions, such as object manipulations [12].

According to the *Universal Simulation Principle,* “any interaction mechanism from the real world can be simulated in VR” [13]. Such interaction mechanics would require no learning effort from the user because they would already be familiar, and any cognitive effort spent on interaction mechanics would be eliminated. Although interactions in the virtual environment are similar to real-world interactions, there are still significant differences that can negatively affect the user experience. For example, when grabbing objects in virtual reality, there is no tactile feedback for each finger, making this type of interaction unintuitive and cumbersome. Furthermore, VR may offer experiences that are unlike anything possible in the real world, making one-to-one mapping impossible [13].

In the development of VR interaction mechanics, some design considerations are ease of learning, reducing cognitive load, overall comfort over extended periods of use, and effectiveness in task execution [13]. The difficulty of learning interactions in VR should be as low as possible because users cannot be expected to spend a lot of time learning how to interact with an application in order to use it. Similarly, reducing the cognitive load of interaction mechanics enables users to focus only on application tasks. Interaction in VR usually should not cause physical pain or discomfort, thus special care should be given to ergonomic design. Finally, users should be able to achieve the required accuracy, speed, and overall effectiveness of task execution.

However, games require distinct design considerations because they are meant to be challenging [4]. Specifically, interaction mechanics should not be too optimized to diminish the entertainment value of a challenge. On the other hand, some optimization is still required to ensure that a goal is achievable, otherwise, players can become discouraged and frustrated.

1.2. Classification of interaction mechanics

The basic classification of interactions in virtual reality categorizes interactions into three groups: manipulation, navigation, and communication [14]. Some sources classify selection and scaling as separate interaction categories [15], but for the purpose of this thesis, they are considered as a part of the manipulation's classification. Manipulation allows user to alter objects in the VR environments, navigation enables users to move through the virtual world, and communication provides interactions with other users or agents in the virtual world. Each of these interactions consists of a primary task and several subtasks necessary to achieve the primary task. For example, the primary task of grabbing an object consists of subtasks such as selecting and activating the grab action. The focus of this thesis are the different manipulation interactions.

Manipulation of objects in VR consists of selection, rotation, scaling, changing position, and placement [15]. The two primary selection techniques are local and at-a-distance. Local selection is closest to the real-world selection because the object must be within the user's reach and is then selected using a predefined action such as pressing a button on the VR controller or performing a grab gesture such as closing the hand into a fist. At-a-distance selection is used for objects outside of the player's reach by pointing in the direction of the object and performing a predefined action. Additionally, occlusion may negatively impact the selection interaction because the probability of unintended selection increases [12]. Rotational manipulation can have the center of rotation placed around the hand that is holding the object, or it can have a remote center of rotation, for example, rotation around the hinge axis of virtual doors. Scaling manipulation is defined by the center of scaling, which can be the center of the object or the point at which the object is grabbed at, and the scaling factor, which can be hand controlled such that pulling or pushing the object between two hands scales the object, or it can be controlled by pressing physical or virtual buttons and controls. Changing the position of an object is the most intuitive form of object manipulation because objects are moved the same way as in the real world. Placement entails dropping the held object in the desired position and with the desired orientation. This interaction can be executed by a predefined action such as pressing a button or using a gesture like opening a hand. A snap zone is a space around the desired placement position that grabs the object that was dropped inside this zone and places it in the target position and orientation. The scale of these snap zones determines the required precision the user must achieve before dropping the object. Alternatively, users may be asked to place objects without any snap zones enabled, which would require very high precision [13].

Depending on the types of tasks performed, interactions may be non-mediated or tool-mediated [4]. Non-mediated interactions are direct interactions that enable a player to interact with the virtual environment only within the physical boundaries of the play area. One-to-one mapping is most commonly used to implement this type of interaction, resulting in the limited reach of a player's influence inside the virtual reality environment. On the other hand, tool-mediated interactions extend the boundaries in which the player can act in by providing the player with a tool. This tool can be attached to the players' hands, or they can replace the hand models. Non-mediated interactions are mostly used for picking up and placing virtual objects, while tool-mediated interactions are used for shooting, hitting, throwing, or slashing objects in VR [4].

1.3. Related studies

A study focusing on object manipulation methods in VR was conducted in 1997 [12]. The goal of this study was to develop a formal methodology and experimental tools that would help VR application developers make better design decisions. In this study, a testing framework was developed for evaluation and optimization of manipulation interactions in VR, and a pilot study was conducted. However, the application evaluated only the selection, positioning, and orienting interactions, and no tool-mediated interactions were tested. Moreover, this study was conducted before the modern VR technology used today was developed.

A study conducted in 2017 [16] tested VR navigation and interactions with objects in a single-player 1st person horror VR game with the goal of reducing simulation sickness by optimizing locomotion mechanics. Several VR locomotion and interaction mechanics were evaluated using in-game parameter tracking and player self-assessment, however, interaction is focused on selecting and transporting virtual objects, and no additional manipulation interactions or tool-mediated interactions were tested.

Research article written by Weise et al. [17] in 2020 attempted to classify different interaction mechanics and a study was conducted in which several selection and manipulation techniques were evaluated, concentrating primarily on selection, positioning, rotation, and scaling. The goal of this study was to explore the design space by investigating as many dimensions of various interaction mechanics as possible, but only non-mediated interactions were evaluated.

A more recent study from 2021 [18] compared controller-based interactions and hand-tracking interactions while performing reach-pick-place tasks. The focus of this study was to determine how hand-tracking impacts immersion, effectiveness, and the perceived mental workload, and no specific interaction mechanics were evaluated.

2. Solution description

In the scope of this thesis, an application was developed with the purpose of testing and analyzing the impact of different interaction mechanics on objective success and subjective enjoyment of gameplay in virtual reality. The solution is designed in accordance with the INTERACT framework [4] principles. These principles include

- **independent interface** – separate configuration interface for the study administrator to use.
- **neutralize nuisance/confounding variables** – keeping some parameters constant in the experiments and including some degree of randomization in the implementation of the test scenarios.
- **track multiple measures** – tracking and recording multiple objective measures.
- **exporting collected data** – data should be stored locally or in a database and grouped by test scenario and study participant.
- **repeatability** – saving configuration settings for easier reuse in testing.
- **automation** – virtual test environments are automatically created based on the selected configurations.
- **customization** – flexibility of the virtual environment.
- **Totality of the VR gaming experience** – study tasks within the test subjects' feasible and comfortable physical and mental limits.

For the purpose of QoE testing, three separate VR games are available, each with their own set of configurable parameters. Each of the three games features a different game genre and, accordingly, different interaction mechanics, tasks, and goals. In the *Shooter* game, the goal is to shoot as many targets as possible in a set time using projectile weapons. The goal of the *Box Smash* game is to slice targets using virtual swords, while in the *Pick-And-Place* game, the user must assemble a 3D puzzle by grabbing and positioning pieces. The application provides an independent desktop interface for the test administrator to customize various configurable settings of the implemented interaction mechanics, granting them control over testing scenarios while test subjects only experience the games in VR.

Neutralizing nuisance variables is achieved by randomizing some parameters within a set range. For example, in the *Shooter* game, the movement of the targets is set randomly with a certain predefined probability, and in the *Pick-And-Place* game, the selected number of puzzle pieces is generated randomly for each round using the self-avoiding walk algorithm which ensures different puzzle solutions in each round. Additionally, in the conducted QoE study, scenarios were grouped so that only one interaction parameter changed values while the other parameters were set to default recommended values.

For each game round, multiple objective measures are tracked and stored for an objective analysis of players' success in completing game tasks. Data is stored locally in two separate files, one for general information and average values, and the other for each target or puzzle piece generated in the game round.

The solution is designed for repeatability by saving the latest values of parameters and auto-populating them in the desktop menus. Those values are then used to automatically create a custom virtual environment for the test subjects to evaluate. In the development of the games, most of the implemented game mechanics were designed to be customizable and provide a high level of flexibility in each game. This also provides a lot of options for various testing scenarios because a multitude of interaction mechanics values can be modified. To ensure that the tested interaction mechanics are feasible and comfortable for the test subjects, a minimum, a maximum, and a default value are given for each configurable interaction mechanic implemented. The goal of this application is to create a test framework for surveying and analyzing interaction mechanics parameters' impacts on the quality of user experience and objective effectiveness and success in VR games.

3. Technologies and tools

3.1. Unity

Unity is a platform for creating and operating interactive, real-time 3D content [19]. It enables game development for both 2D and 3D games, as well as support for virtual and augmented reality applications. Aside from game development, this platform is used in architectural and automotive design and visualization, and film effects. Unity also offers the Unity Asset Store, an online store with both free and paid assets such as models, scripts, and tools that can be added to a Unity project. In the development of the application for testing interaction mechanics in VR version 2020.3.30f1 of Unity was used.

3.2. Microsoft Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) that is used in application development [20]. It contains a source code editor that supports *IntelliSense*, code refactoring, and a debugger. Visual Studio was used to create C# scripts that model application logic and behaviors in Unity.

3.3. Virtual Reality Interaction Framework

The Virtual Reality Interaction Framework (VRIF) is a collection of scripts and prefabs developed by Bearded Ninja Games and used in the development of interaction mechanics in VR [21]. It provides examples of common VR interactions, such as teleportation, levers, weapons, and more. Moreover, it enables developers to create interactive objects by using modular components. Some of the features provided by this framework are physical grabbing and throwing objects, smooth locomotion and teleportation, custom hand pose support, climbing, two-handed weapons, a simple damage system with destructible objects, and snap zones. This framework was used for implementing all of the basic interactions in the developed application, such as object grabbing, weapon wielding, and target destruction.

3.4. HTC Vive Pro

The HTC Vive is a virtual reality device by HTC Corporation [22]. It consists of a stereoscopic head-mounted display (headset), controllers, and two base stations for tracking users' position and movement. The headset is tethered to a computer and features a high-resolution display, an outward-facing camera, over-ear speakers, a microphone, and built-in eye tracking. Figure 3.1 shows the HTC Vive Pro headset, controllers, and base stations.

Figure 3.1 HTC Vive Pro headset, controllers, and base stations [22].

3.5. Valve Index Controllers

Valve Index Controllers (Figure 3.2) are virtual reality controllers created and manufactured by Valve [23]. They are intended to be used with Valve Index VR headsets but are also compatible with the HTC Vive Pro. Controllers have an array of sensors that enable tracking hand and finger position, motion, and pressure, for more natural user interaction. Valve Index controllers are designed specifically for open-handed interactions, allowing users to drop or throw objects.

Figure 3.2 Valve Index controllers [24].

3.6. Google Forms

Google Forms are Google's online form and survey tool [25]. It enables adding various types of questions, such as multiple-choice, linear scale, text answer, and more. Google Forms also offers analyzing collected responses by creating automatic summaries and charts, but users can also download results in a .csv format.

4. Solution implementation

The application was developed using the Unity game development platform, and it consists of a desktop main menu, three distinct VR games, and an interaction parameters modification menu for each game. Additionally, data is collected and stored locally for each player and each round played. The solution is an extension of the application created by student Filip Nemec [26] in the scope of the Master's program at the University of Zagreb, Faculty of Electrical Engineering and Computing. The original application consisted of two games – *Shooter* and *Box Smash*. In the scope of this thesis, a *Pick-And-Place* game was developed, some additional interaction parameters were added to the existing games, and the application was redesigned to create a more uniform aesthetic design across all three games.

The desktop user interface is intended for the use of the test administrator performing the study. The main menu (Figure 4.1) is the start screen in which the survey administrator selects one of the three available games from the drop-down menu and inputs study and participant identifiers. The provided information is used for naming and organizing collected data. Each data file is named after the type of game played and the date and time of creation. Data files are stored in a local folder named after the participant's identifier, which in turn is stored in a folder named after the study identifier. The structuring of data files can be seen in Figure 4.2.

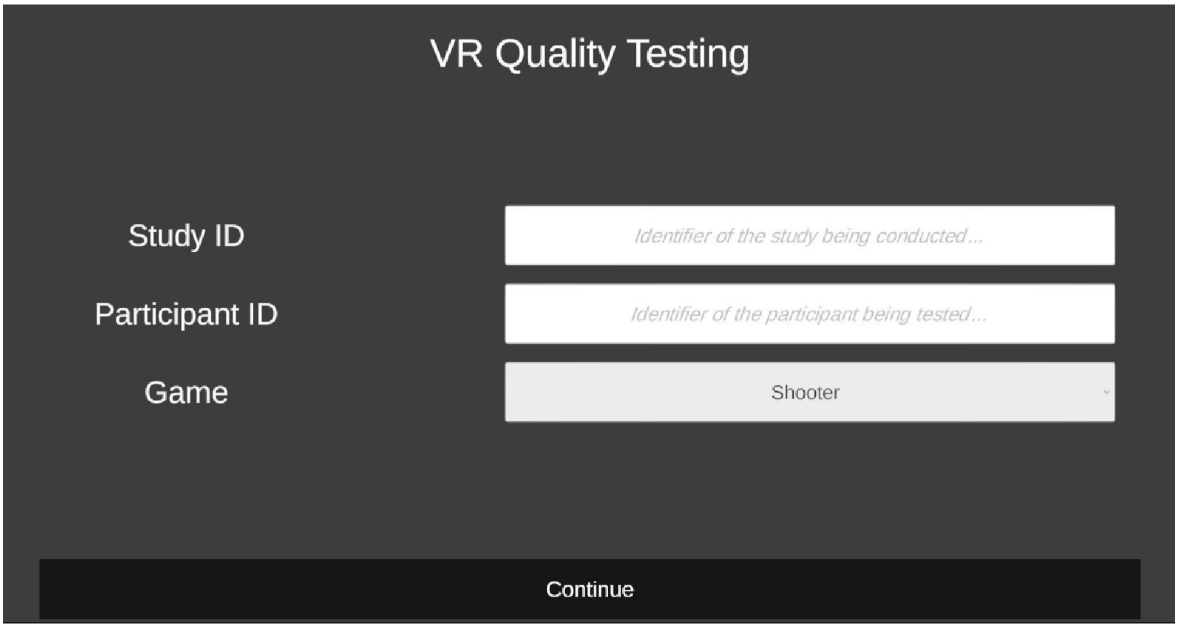

Figure 4.1 The application's main menu.

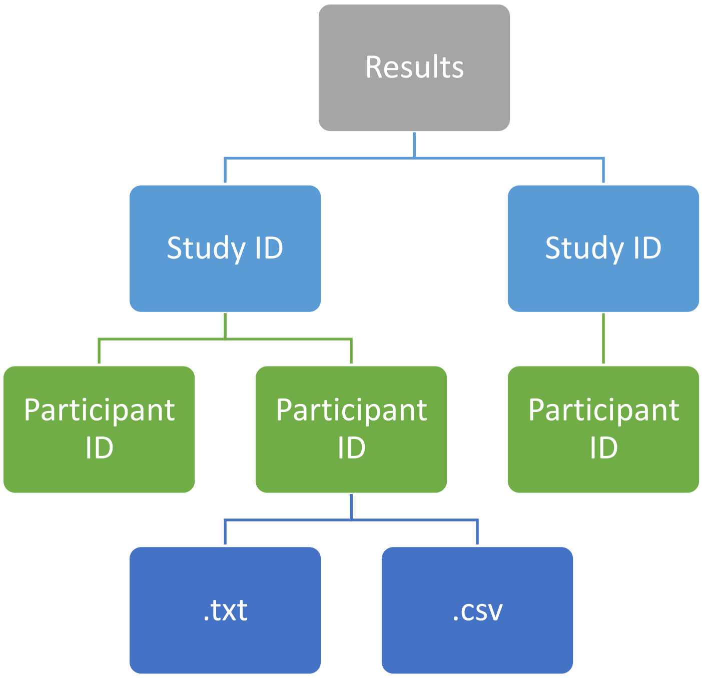

Figure 4.2 The structuring of data files.

Depending on the game type chosen, selecting the *Continue* button redirects to one of the three interaction parameter selection screens. Each of the menus contains parameters specific to the selected game. Figure 4.3 shows the menu for the shooter game. Parameters are grouped by game objects they influence and separated into menu tabs. Selecting the *Play* button starts the previously selected game, while selecting the *Back* button returns the user to the main menu screen. On selecting either of the two buttons, current values of interaction parameters are stored and used to repopulate menu fields once the user returns to this screen. This data is stored persistently and is repopulated even if the application is terminated, which simplifies setting interaction parameters' values. All the user interface menus are available only to the desktop user, while in virtual reality a message is shown, instructing users to wait for the game to start.

Figure 4.3 Parameter selection menu of the *Shooter* game.

4.1. Gameplay objectives

The application contains three separate games, each implementing a different type of game interaction mechanics. The *Shooter* game implements shooting mechanics, the *Box Smash* game implements slashing mechanics, and the *Pick-And-Place* game implements pick and place interaction mechanics.

The *Shooter* game equips the player with projectile weapons, one for each hand. For this application, all weapons have been set to automatically reload, removing ammunition concerns for the player. The goal of this game is to destroy as many targets as possible in a given time. Depending on the interaction mechanics selected, targets may be moving or stationary, and their speed and direction of movement can be modified as well. Targets do not attack the player, so there are no penalties or dangers to the player for missing targets or being slow in destroying them.

The *Box Smash* game equips the player with an edged weapon in both hands. Cannons are spawned radially around the initial position of the player, and they shoot targets upwards at random time intervals. The player must destroy as many targets as possible in the given time by swinging the weapon and slashing the targets.

The goal of the *Pick-And-Place* game is to assemble a cube using the given puzzle pieces. Once the initial cube is divided into smaller cubes (Figure 4.4), the puzzle pieces are created randomly using the self-avoiding walk algorithm so that each node is the previously created smaller cube. This makes the puzzle pieces' shapes and positions within the solution random for each testing scenario. This approach was taken to eliminate the influence of a player learning the same puzzle over several different testing scenarios. Once created, puzzle pieces are given a random color, randomly rotated, and placed inside of a grid layout in a randomized order.

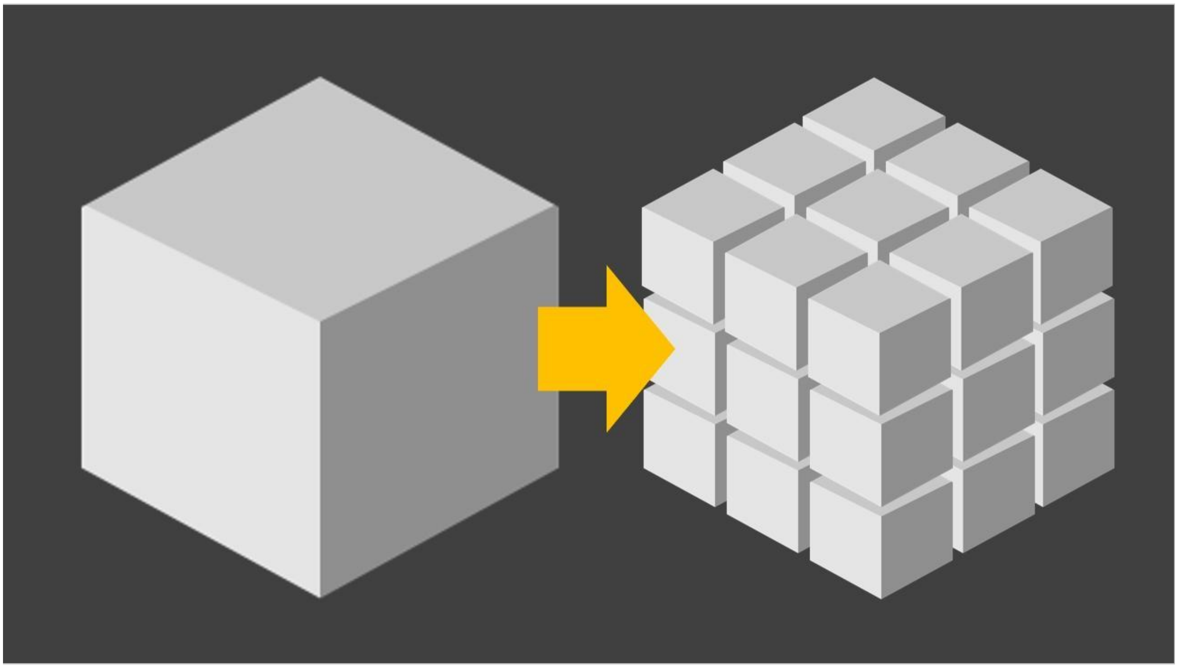

Figure 4.4 Division of the initial cube into 3 cubes per axis.

Puzzle pieces are assembled inside a solution cube, and each puzzle piece has only one correct placement inside the solution. When the player picks up a puzzle piece, the corresponding place inside the solution cube is highlighted with the same color as the puzzle piece being held. The player can then move the piece to the correct position, and if the positioning is precise enough, the corresponding spaces in the solution cube will become highlighted using a luminous material (Figure 4.5). The player may then drop the puzzle piece and it will snap to the correct position and rotation inside the solution cube.

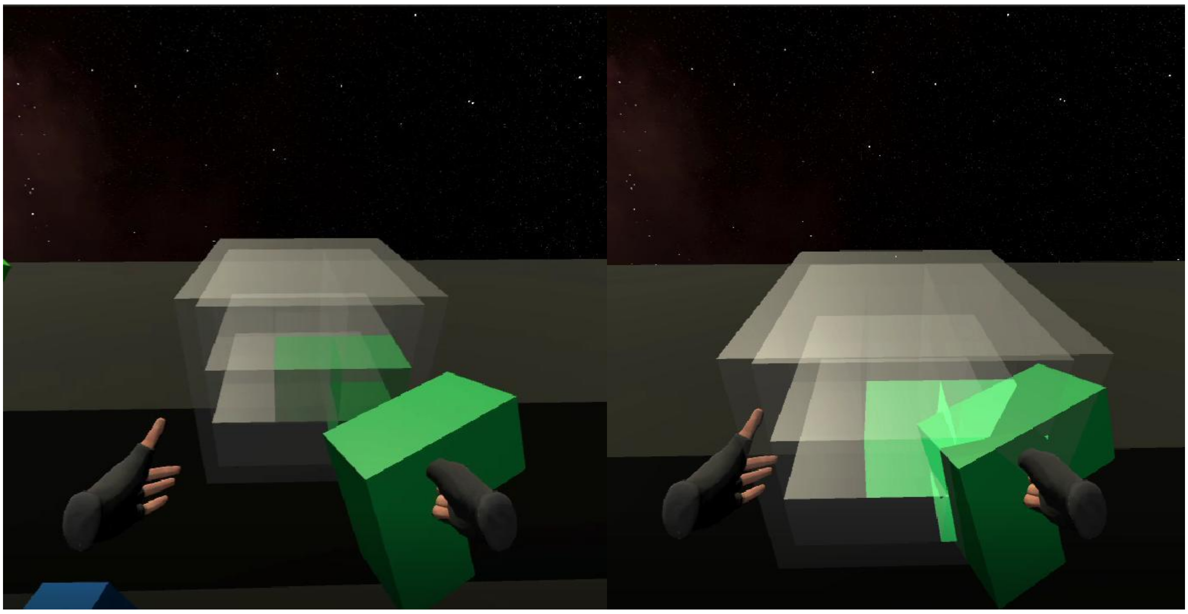

Figure 4.5 Highlighting solution space on grab (left) and when it is positioned precisely enough to be accepted and snapped into place when released (right).

If the positioning was not precise enough, the piece will fall through the solution cube. Each puzzle piece is formed out of smaller cubes, and each of the cubes has a modifiable collider attached, shown in Figure 4.6. Likewise, the solution cube is formed out of identical smaller cubes that also have modifiable colliders attached. Only once each of the puzzle piece colliders are in collision with the correct solution cube colliders will the solution be accepted when dropped. The C# script modeling the *On Release* behavior is shown in Figure 4.7. Once the puzzle piece is inside the correct position in the solution cube, it can be removed simply by grabbing it again. If any of the puzzle pieces or the solution cube fall to the ground, they will be returned to the position they were in at the start of the game after the number of seconds specified in the interaction parameters for this game.

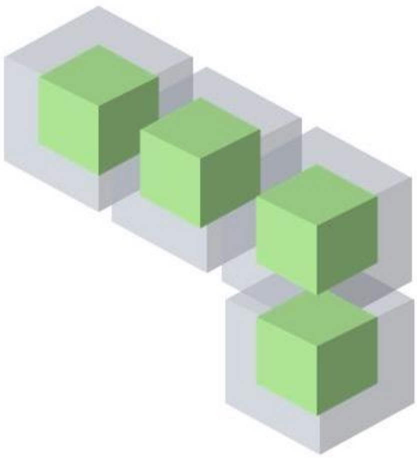

Figure 4.6 A puzzle piece consisting of smaller cubes and their colliders (green).

```
public override void OnRelease()
{
	grabbed = false;
	highlight.RemoveHighlight();

	// log time metrics
	pm.timeFromFirstGrab = Time.time - pm.timeFromFirstGrab;
	pm.timeFromLastGrab = Time.time - pm.timeFromLastGrab;

	// check if all parts of puzzle piece collide with a solution cube
	foreach (PositionDetection childCube in childCubes)
	{
		// if child cube is not colliding with solution cube
		if(childCube.solutionCubes.Count == 0)
		{
			return;
		}
		childCube.solutionCubes.Clear();
	}

	gameObject.GetComponent<Rigidbody>().isKinematic = true;
	transform.parent = solutionCube.transform;

	Vector3 onReleaseRotation = transform.eulerAngles;
	transform.localEulerAngles = correctRotation;

	// log end rotatio
	pm.endRotation = transform.eulerAngles;

	// log rotation difference
	pm.angleDifference = Vector3.Angle(onReleaseRotation, pm.endRotation);

	// log distance to target
	pm.targetDistance = Vector3.Distance(transform.position, center);

	// move puzzle piece to solution
	transform.position = correctPosition.transform.position;

	// log end position
	pm.endPosition = transform.localPosition;
	pm.endDistanceToPlayer = Vector3.Distance(transform.localPosition,
	player.transform.localPosition);

	pm.inPlace = true;

	// add to solution count
	if (!inSolution)
	{
		puzzleSolved.inSolution.Add(pm);
		inSolution = true;
		puzzleSolved.CheckPuzzleSolved();
	}

	// hide solution cubes to avoid mesh overlaping
	foreach (GameObject cube in highlight.solutionCubes)
	{
		cube.SetActive(false);
	}

}
```

Figure 4.7 The C# script modeling the behavior of puzzle pieces when they are released.

4.2. Scene design

Scene design was kept uniform across all three games, using the space theme as the main concept. A nebula skybox is used, and players are placed on platforms of limited size, floating in the empty space. The goal of this design was to remove any unnecessary distractions for the player. Furthermore, by contrasting the muted and dark colors of the environment with the colorful and bright colors of targets and puzzle pieces, more emphasis is given to the game's tasks.

Since the duration of the game round is limited, players are given time to orient and prepare before the countdown is started. This is accomplished by adding a button to start the round inside the virtual reality environment (Figure 4.8). In each game, the player is instructed to position themselves in front of the red button and press it when they are ready to start the round. Once the round is started, the button disappears, and the targets or puzzle pieces become visible. For easier orientation of the player inside VR, a symbol that marks the center of the play area and the forward position is placed on the ground.

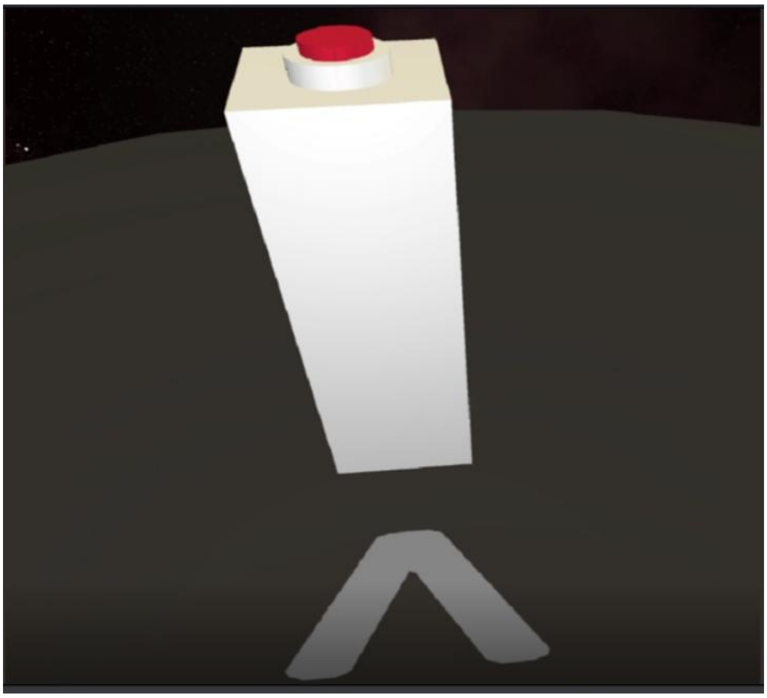

Figure 4.8 A virtual button for starting the round and the symbol for the forward position.

In both the *Shooter* and the *Box Smash* games, targets are cubes of randomly assigned colors, which relates to the colorful cube puzzle in the *Pick-And-Place* game. Once a target is destroyed, an audio effect plays, and the target is dissolved into smaller cubes to create a shattered effect (Figure 4.9). Because targets can be spawned all around the player, in the

Shooter game, targets emit a constant sound effect that helps the player locate them. In the *Box Smash* game, cannons emit a sound each time a target is spawned from them.

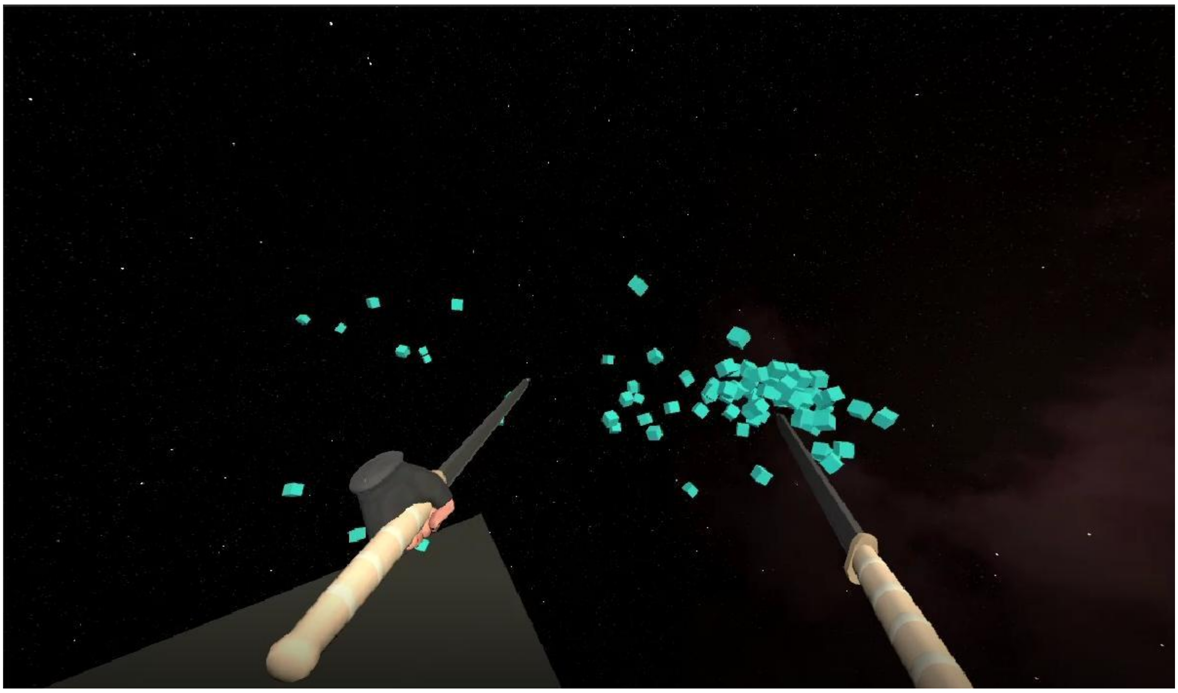

Figure 4.9 A destroyed target dissolving into smaller cubes in the *Box Smash* game.

The puzzle pieces and the solution cube for the *Pick-And-Place* game are located on a virtual table. The solution cube is translucent and divided into a grid corresponding to the smaller cubes that form the puzzle pieces. Once a puzzle piece is grabbed, the correct cubes inside the solution cube are tinted the same color as the puzzle piece, and once the puzzle piece is correctly positioned inside the solution cube, the same solution cubes get a luminescent highlight, indicating to the player that the puzzle piece can be dropped. Players' virtual hands and the puzzle pieces can be moved through the solution cube. However, the solution cube interacts with the table and can be dropped on the ground. Additionally, the solution cube can be grabbed and rotated in the same way that the puzzle pieces can be manipulated. A configurable solution cube offset is provided for easier manipulation of the solution cube when puzzle pieces are already placed to minimize accidental removals of placed puzzle pieces.

4.3. Interaction mechanics parameters

Each interaction mechanic parameter available in the selection menu has recommended minimum, default, and maximum values. These values and a short description of the interaction parameter can be seen in the application by hovering the mouse pointer over the name of the interaction parameter (Figure 4.10).

The menu for the *Shooter* game contains four groups of parameters: target spawner, target, weapon, and round. The target spawner group contains the following parameters:

- **Minimum distance and maximum distance** – range of distances from the player in which a target will be randomly spawned.
- **Minimum distance and maximum height** – range of heights from the ground on which a target will be randomly spawned.
- **Spawn angle** – circle segment angle in degrees inside which targets will be spawned with respect to the player's initial position and orientation.
- **Spawn count** – number of targets that will be spawned in one set.
- **Duration between spawns** – the time between sets of targets spawned in seconds. If the value is zero or less, the next set of targets is created only after all the targets from the previous set have been destroyed. Otherwise, a new target will be spawned every *n* seconds.

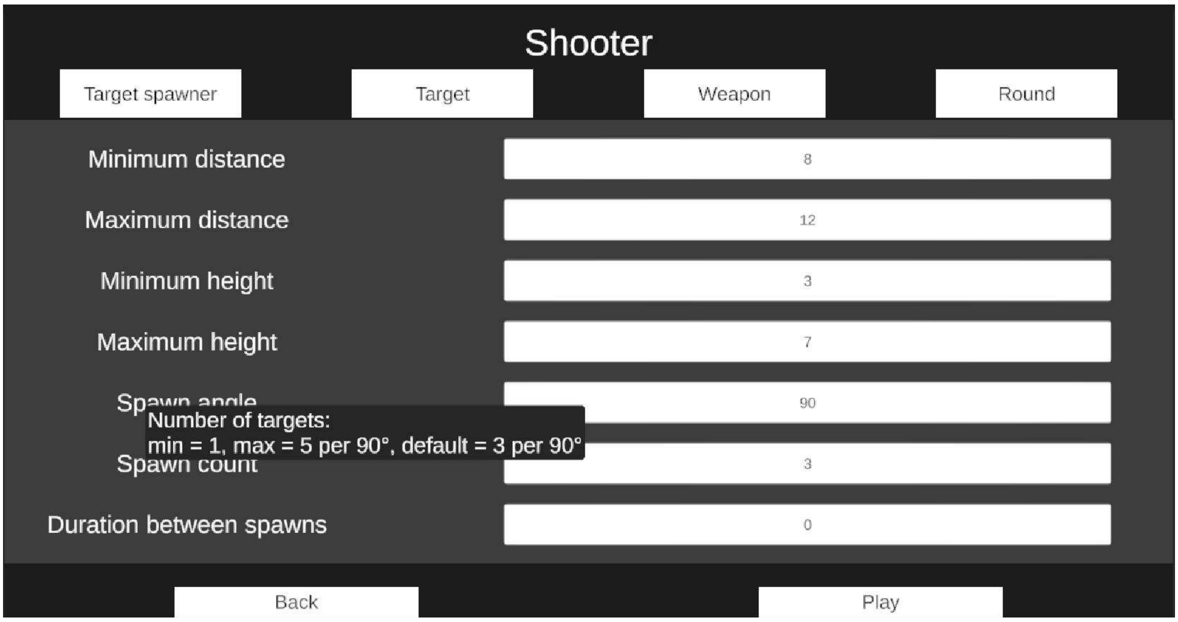

Figure 4.10 Menu tooltip for the recommended interaction parameter values.

Target tab parameters determine target behavior after it was created. These parameters are the following:

- **Minimum size and maximum size** – a range of target scale in meters.
- **Moving probability** – the probability of the target moving once it was created.

- **Minimum velocity and maximum velocity** – a range of velocity with which a moving target will travel.
- **Minimum offset and maximum offset** – the range of distance a moving target will traverse.
- **Move direction** – direction of the moving target can be up-down, left-right, and forward-backward in reference to the initial position and orientation of the player. Any combination of the moving directions can be selected, however, if no directions are selected, targets will not move.

In the *Shooter* game, the player uses a projectile weapon to destroy targets. Weapon settings can be modified by the following interaction mechanics parameters:

- **Weapon type** – can be a pistol or a rifle.
- **Use laser** – selecting this option attaches a laser sight to the weapon.
- **Show bullet trajectory** – selecting this option enables projectiles trajectory to be visible.
- **Show muzzle flash** – selecting this option enables a visible effect in the form of a light flashing each time a projectile is shot.
- **Bullet trajectory time** – determines how long the bullet trajectory will be visible after the projectile has been shot.
- **Shoot force** – determines the force at which a projectile is shot from the weapon muzzle. This force affects the speed of the projectile and the shape of the projectile's path.

The round parameter group contains the *round duration* property, which limits the time in seconds the player will spend playing a test scenario. Table 4.1 shows all the available parameters for the *Shooter* game alongside recommended minimum, maximum, and default values.

Interaction parameter	Minimum value	Maximum value	Default value	Unit of measurement
Target spawner				
Minimum distance	2	16	8	meters

Maximum distance	Minimum distance	20	12	meters
Minimum height	0	16	3	meters
Maximum height	Minimum height	16	7	meters
Spawn angle	0°	360°	90°	degrees
Spawn count	1	5/ 90°	3	-
Duration between spawns	0	5	0	seconds
Target				
Minimum size	0.2	3	0.5	meters
Maximum size	Minimum size	3	2	meters
Moving probability	0	1	0.2	-
Minimum velocity	0	10	0.5	meters/second
Maximum velocity	Minimum velocity	10	5	meters/second
Minimum offset	0	15	2	meters
Maximum offset	Minimum offset	15	8	meters
Weapon				
Weapon type	Pistol, Rifle		Pistol	-
Use laser	True, False		False	-
Show bullet trajectory	True, False		True	-
Show muzzle flash	True, False		False	-
Bullet trajectory time	0.05	1	0.5	seconds
Shoot force	5	80	40	newtons

Round				
Round duration	60	300	90	seconds

Table 4.1 The available parameters for the *Shooter* game and their minimum, maximum, and default values and units of measurement.

The menu for the *Box Smash* game contains four groups of parameters: cannon spawner, cannon, weapon, and round. The Cannon spawner group contains the following parameters:

- **Spawn distance** – the distance of a target spawning cannon from the initial position of the player in meters.
- **Spawn height** – the height of the target spawning cannon from the ground in meters.
- **Spawn angle** – circle segment angle in degrees inside which cannons are spawned with respect to the player's initial position and orientation.
- **Spawn count** – number of cannons created.
- **Tilt angle** – angle of the cannon. This parameter currently has no impact on targets created by the cannon and only changes the angle of the cannon visually.
- **Minimum force** – minimum force weapon needs to be applied to the target for it to be destroyed.
- **Use force relative to the cube** – selecting this option will calculate the force applied as relative to the target. This means that the speed of the target impacts the needed force in such a way that less speed of the weapon is needed to achieve the same force compared to when relative force is not used.

Cannon interaction parameters affect targets that are shot from the cannon:

- **Minimum shoot force and maximum shoot force** – a range of force at which a target will be shot at from the cannon. The force impacts the target's speed and maximum achieved height.
- **Minimum duration between shots and maximum duration between shots** – a range of time in seconds until the next target is shot from the cannon. This time is specific to each cannon, so the targets are not synchronized between all cannons created.
- **Minimum box size and maximum box size** – a range of target scale in meters.

In the *Box Smash* game, targets are destroyed using a bladed weapon. Interaction parameters for that weapon are the following:

- **Weapon type left hand and weapon type right hand** – the type of bladed weapon that can be set for each hand independently.
- **Weapon length left hand and weapon length right hand** – length scaling factor that can be set for each hand independently. A scaling factor of less than one shortens the default weapon length, while a scaling factor greater than one lengthens it. The default weapon length is one meter.
- **Use haptic feedback** – selecting this option enables controller haptic feedback in the form of vibrating the controller each time the weapon collides with another game object.
- **Vibration amplitude** – amplitude of the haptic feedback vibration.
- **Vibration frequency** – frequency of the haptic feedback vibration.
- **Vibration duration** – duration of the haptic feedback vibration in seconds.

The gameplay round parameter group contains the *round duration* property which determines the time limits in seconds the player will play a test scenario. Table 4.2 shows all the available parameters for the *Box Smash* game alongside recommended minimum, maximum, and default values.

Interaction parameter	Minimum value	Maximum value	Default value	Unit of measurement
Cannon spawner				
Spawn distance	1	1.5	1	meters
Spawn height	-0.5	0.5	0	meters
Spawn angle	0°	360°	90°	degrees
Spawn count	1	4/90°	4	-
Tilt angle	0°	180°	90°	degrees
Minimum force	0	6	2	newtons
Minimum relative force	0	20	8	newtons
Cannon				
Min shoot force	2	10	6	newtons

Max shoot force	Min shoot force	10	7	newtons
Min Duration Between Shots	1	5	3	seconds
Max Duration Between Shots	Min Duration Between Shots	5	4	seconds
Min Box Size	0.1	0.7	0.3	meters
Max Box Size	Min Box Size	0.7	0.4	meters
Weapon				
Weapon type	Katana		Katana	
Weapon length	0.5	1.5	1	meters
Use haptic feedback	True, False		True	-
Vibration amplitude	0	1	0.5	-
Vibration frequency	0.3	0.6	0.3	Hertz
Vibration duration	0.1	0.5	0.2	seconds
Round				
Round duration	60	300	90	seconds

Table 4.2 The available parameters for the *Box Smash* game and their minimum, maximum, and default values and units of measurement.

The menu for the *Pick-And-Place* game contains four groups of parameters: puzzle spawner, target, puzzle pieces, and round. The puzzle spawner group models puzzle creation and contains the following parameters:

- **Puzzle scale** – the size of one edge of the completed cube puzzle in meters.
- **Cubes per axis** – number of cubes the solution cube is divided into when creating puzzle pieces.

- **Number of pieces** – number of pieces in the puzzle.
- **Number of columns** – number of columns in the grid in which the pieces will be placed on the table.
- **Space between pieces** – the distance between puzzle pieces in the grid they are initially placed on at the start of the game.
- **Maximum depth** – maximum depth distance of a puzzle piece with respect to the initial position and orientation of the player.
- **Maximum side offset** – maximum side distance of a puzzle piece with respect to the initial position and orientation of the player.

The target interaction parameters affect the solution space of the puzzle and contain the following:

- **Collider scale** – the scale of the colliders in the solution space. This parameter affects the precision necessary for a puzzle piece to be accepted as a solution.
- **Solution depth** – maximum depth distance of the solution space with respect to the initial position and orientation of the player.
- **Solution side offset** – maximum side distance of the solution space with respect to the initial position and orientation of the player.
- **Solution grab offset** – offset of the solution cube necessary for grabbing and manipulating, expressed in meters.

Puzzle pieces can be modified with the following interaction parameters:

- **Remote grab** – selecting this option will enable selecting and grabbing puzzle pieces from a distance.
- **Remote grab distance** – the maximum distance at which the puzzle piece can be remotely grabbed, expressed in meters.
- **Haptic feedback** – selecting this option enables controller haptic feedback in the form of a vibrating controller each time the weapon collides with another game object.
- **Vibration amplitude** – amplitude of the haptic feedback vibration.
- **Vibration frequency** – frequency of the haptic feedback vibration.
- **Vibration duration** – duration of the haptic feedback vibration in seconds.
- **Scale offset** – the factor by which the puzzle piece is scaled. Puzzle pieces are scaled down to create a space between pieces placed in the solution space to enable easier

placement inside the solution cube. The scale of the puzzle piece is determined by the scale of the cubes it is formed out of, and the final scale of these cubes expressed in meters can be calculated by multiplying the puzzle scale and the scale offset and dividing them by the number of cubes per axis.

- **Respawn time** – time in seconds after which a puzzle piece or the solution cube dropped on the ground will be returned to the position they were in at the start of the game.

The gameplay round in the *Pick-And-Place* game can be limited to a fixed number of seconds or it can end only once the puzzle has been completed. Table 4.3 shows all the available parameters for the *Pick-And-Place* game alongside recommended minimum, maximum, and default values.

Interaction parameter	Minimum value	Maximum value	Default value	Unit of measurement
Puzzle spawner				
Puzzle scale	0.1	1	0.4	meters
Cubes per axis	3	4	3	-
Number of pieces	4	7	7	-
Number of columns	1	Number of pieces	4	-
Space between pieces	0	0.5	0.3	meters
Maximum depth	0	1.2	1	meters
Maximum side offset	0.5	1.5	1.5	meters
Target				
Collider scale	0.1	1.5	0.5	meters
Solution depth	0	1.2	0.3	meters
Solution side offset	0	1.5	0.5	meters

Solution grab offset	0.05	0.2	0.1	meters
Puzzle pieces				
Remote grab	True, False		False	-
Remote grab distance	0.5	5	2	meters
Haptic feedback	True, False		True	-
Vibration amplitude	0	1	0.5	-
Vibration frequency	0.3	0.6	0.3	Hertz
Vibration duration	0.1	0.5	0.2	seconds
Respawn time	0	5	1	seconds
Scale offset	0.8	1	0.9	-
Round				
Limit round duration	True, False		True	-
Round duration	60	300	90	seconds

Table 4.3 The available parameters for the *Pick-And-Place* game and their minimum, maximum, and default values and units of measurement.

4.4. Data collected

After each round, information about the game is stored locally in two separate files. One is a text file containing general information about the gameplay and the interaction parameter values that were set in the game, while the other is a comma-separated file containing information about each target or each puzzle piece.

The general information recorded for the *Shooter* game and stored in a .txt file (Figure 4.11) are:

- total shots fired,

- total hits,
- total misses,
- total accuracy,

and each of these parameters separately for each of the hands. The information recorded for each destroyed target and stored in a .csv file (Figure 4.12) are:

- the distance to player in meters at the time of death,
- distance from the hit point to the center of target in meters,
- angle from predefined forward axis in degrees,
- lifetime expressed in milliseconds, size of the target in meters,
- velocity of the target expressed in meters/seconds,
- offset of the moving target from the initial position in meters,
- hand side by which the target was destroyed,
- birth and death timestamps.

```
# Round results
Total shots fired: 109
Total hits: 82
Total misses: 27
Total accuracy: 0.7522936

Right hand shots fired: 64
Right hand hits: 48
Right hand misses: 16
Right hand accuracy: 0.75

Left hand shots fired: 45
Left hand hits: 34
Left hand misses: 11
Left hand accuracy: 0.7555556

Duration: 90

# Target spawner settings
...
```

Figure 4.11 An excerpt of the general information exported to a .txt file for the *Shooter* game.

Distance to target	Distance to center	Angle from forward	Lifetime	Size	Velocity
11.89506	0.2578073	0.9373261	1077	1.439677	0
11.29783	0.3543931	43.85156	2086	1.820617	0
11.39762	0.9216068	68.80981	2640	1.926158	2.690081
12.05573	0.4114989	25.8529	1842	1.576922	0
12.62228	0.2643943	21.89173	2731	1.309208	0
10.70068	0.06392658	13.7731	3170	1.634268	0
9.547591	0.190668	1.344503	1136	0.637884	0
11.74847	0.2358036	14.99612	2045	0.591295	0
11.10907	0.1122941	26.02205	3301	0.913121	0
11.37171	0.1568528	0.1219466	998	1.43087	0
10.8073	0.7840086	38.69042	1670	1.547357	0
11.06596	0.3983874	39.30469	2365	1.102793	0
13.37445	1.266838	5.544892	1561	0.742411	3.58818

Figure 4.12 An excerpt of the information for each target in the *Shooter* game, stored in a .csv file format.

The general information recorded for the *Box Smash* game and stored in a .txt file (Figure 4.13) are:

- the total number of targets spawned,
- total hits,
- hits by the right hand,
- hits by the left hand,
- number of targets that were not destroyed,
- number of targets that were not hit,
- accuracy,
- average force applied to hit targets expressed in newtons,
- and average force applied to destroyed targets expressed in newtons.

The information recorded for each spawned target and stored in a .csv file (Figure 4.14) are:

- if the target was destroyed,
- force by which it was hit in newtons,
- hand side by which it was hit,
- size of the target in meters,
- lifetime expressed in milliseconds,
- angle from predefined forward axis in degrees,
- and birth and death timestamps.

Round results
Total boxes: 104
Hits: 49
Hits by right hand: 41
Hits by left hand: 8
Misses: 55
Missed completely: 32
Accuracy: 0.4711539
Average force: 3.330349
Average force of destroyed: 4.496575
Duration: 90

Cannon spawner settings
Spawn distance: 1
Spawn height: 0
Spawn angle: 90
Spawn count: 4
Tilt angle: 90
Use relative force: False
Minimal force to destroy: 2

Cannon settings
Min shoot force: 6
Max shoot force: 7
Min duration between shots: 3
Max duration between shots: 4
...

Figure 4.13 An excerpt of the general information exported to a .txt file for the *Box Smash* game.

Was smashed	Smash force	Hand side	Box size	Lifetime	Birth timestamp
TRUE	3.919777	Right	0.3993629	774	15:28:39:321
TRUE	3.491606	Left	0.3070126	793	15:28:39:323
TRUE	6.240345	Right	0.3023401	1473	15:28:39:320
FALSE	0	-	0.3438452	2652	15:28:39:317
TRUE	4.05801	Right	0.3580054	487	15:28:42:426
TRUE	4.886932	Left	0.3820389	733	15:28:42:370
TRUE	2.095541	Right	0.3357872	387	15:28:43:260
TRUE	3.478419	Right	0.3330235	1267	15:28:42:414
FALSE	1.602775	-	0.3302122	2565	15:28:45:757
FALSE	0	-	0.3074339	2798	15:28:45:945
FALSE	0.5993694	-	0.3625569	3220	15:28:46:345
FALSE	0	-	0.3097728	2687	15:28:46:935
TRUE	5.387053	Right	0.3950298	933	15:28:48:811
TRUE	6.375124	Right	0.37435	1089	15:28:49:644

Figure 4.14 An excerpt of the information for each target in the *Box Smash* game, stored in a .csv file format.

The general information recorded for the *Pick-And-Place* game and stored in a .txt file (Figure 4.15) are:

- the total number of pieces in the puzzle,
- the number of puzzle pieces that were placed in the solution at the end of the round,
- average position offset in meters,
- average rotation offset in degrees,
- and the duration of the round in seconds.

The information recorded for each puzzle piece and stored in a .csv file (Figure 4.16) are:

- start position vector,
- end position vector,
- distance between the start and end positions in meters,
- start rotation vector,
- end rotation vector,
- angle between the start and end rotations in degrees,
- start distance from the player in meters,
- end distance of the player in meters,
- position offset from the correct position in meters,
- rotation offset from the correct rotation in degrees,
- which hand side grabbed the piece,
- time from the first grab until the placement in the solution expressed in seconds,
- time from the last grab until the placement in the solution expressed in seconds,
- number of times the puzzle piece was grabbed,
- and if the piece was in the solution at the end of the round.

Round results
Total number of pieces: 7
Number of pieces placed: 7
Average position accuracy: 0.02529245
Average rotation accuracy: 36.80756
Limited round duration: True
Duration: 1203

Puzzle spawn settings
Puzzle scale: 0.1
Cubes per axis: 3
Number of columns: 4
Space between pieces: 0
Depth distance: 1
Side distance: 1.5

Solution settings
Collider scale: 0.5
Solution depth distance: 1
Solution side distance: 1
Solution grab offset: 0.1
…

Figure 4.15 An excerpt of the general information exported to a .txt file for the *Pick-And-Place* game.

Hand side	Start position	End position	Distance traveled	Start rotation
Right	(0.00 1.20 0.05)	(-0.11 0.17 0.06)	1.039304	(314.10 251.94 180.42)
Right	(1.50 1.20 0.55)	(0.08 0.08 0.17)	1.844134	(357.30 241.74 270.88)
Left	(0.50 1.20 0.55)	(-0.13 -0.08 0.12)	1.48936	(58.07 183.49 351.38)
Right	(0.50 1.20 0.05)	(0.13 0.08 -0.04)	1.181513	(329.70 252.19 268.04)
Right	(1.00 1.20 0.05)	(-0.13 0.04 -0.17)	1.629204	(358.21 12.77 256.32)
Right	(1.50 1.20 0.05)	(-0.04 -0.12 -0.08)	2.037188	(35.26 102.40 71.45)
Left	(1.00 1.20 0.55)	(0.17 -0.12 -0.04)	1.673361	(325.49 283.70 43.41)

Figure 4.16 An excerpt of the information for each target in the *Pick-And-Place* game, stored in a .csv file format.

5. Methodology of QoE testing

For this thesis, a user experience study was conducted with the goal of determining the impact of a selected group of interaction mechanics on objective and subjective Quality of Experience metrics. The purpose of this pilot study was to establish which of the interaction mechanics implemented in the application are feasible for future studies and to demonstrate the applicability of the implemented testing framework. For each of the games created, a separate study was administered because each game implemented its own separate and distinct interaction mechanics, and there were no requirements for test subjects to compare their experiences across all three games. On the contrary, any form of comparison, intentional or not, was not the purpose of this study. Nevertheless, study participants were grouped into three equally sized groups with similar age, gender, and prior VR experience distribution, and were assigned to one of the games evaluated.

The study was conducted using an HTC Vive Pro [22] device that was connected to the same computer throughout the whole study, and it was paired with the Valve Index controllers [23]. Figure 5.1 shows the setup used in the study. Before beginning the experiment, test subjects were asked to fill out a general information survey that contained questions about age, gender, dominant hand, experience with VR, and sentiment towards VR. Following that, test subjects played one round of a game with all of the interaction mechanic values set to the default recommended values stated in chapter 4.3 with the purpose of familiarizing with the gameplay and VR. For this round, no data was recorded, and test subjects did not answer the questionnaire.

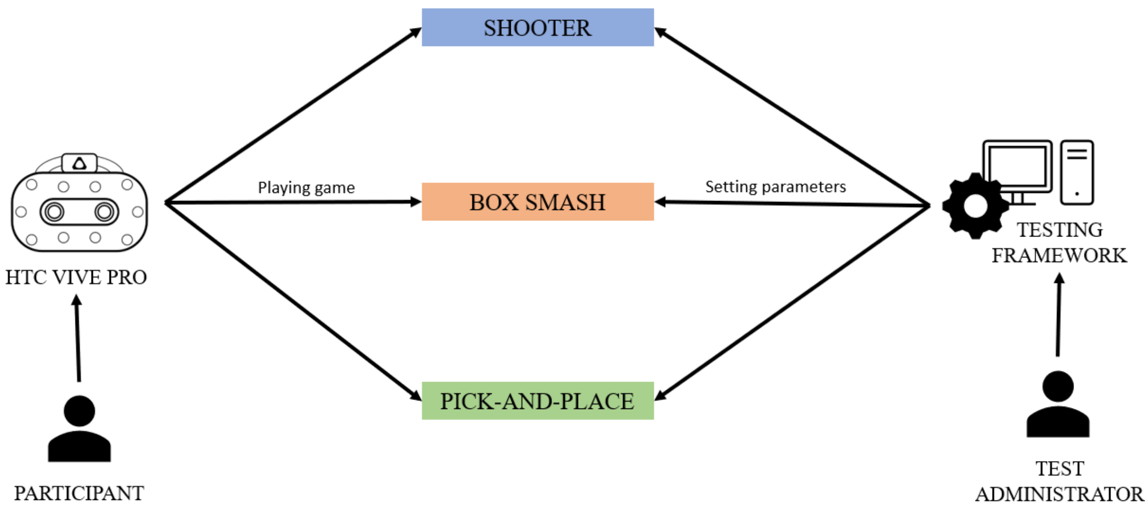

Figure 5.1 Diagram of the study setup.

For each game, several interaction mechanics were tested, and for each interaction mechanic, several values were assessed while the values of the rest of the interaction mechanics parameters were kept as the recommended default values. Testing scenarios were grouped by interaction mechanic tested, and test subjects were given a questionnaire after each testing scenario and an additional questionnaire at the end of each testing scenario group in which test subjects were asked to compare the testing scenarios in which the values of one interaction mechanic were changing. To reduce the need for removing and placing the VR headset to fill out the questionnaire, test subjects were instructed to open a desktop view inside VR. This enabled them to see and answer the questionnaire opened on the desktop computer without taking off the headset. For the short answer questions, test subjects were asked to give their answers verbally, while the test administrator wrote down their answers. The Quality of Experience questionnaire was created and conducted using Google Forms. During the study, test subjects were reminded of the game controls, testing parameters, and question clarification when needed. The duration of the study for each participant was approximately 40 minutes. Figure 5.2 shows the diagram of the methodology of the conducted QoE study.

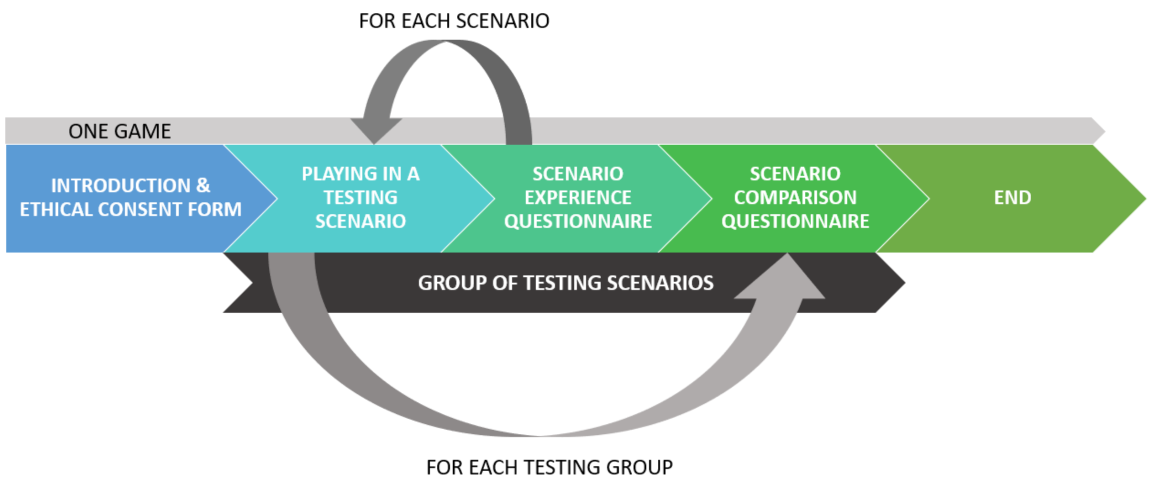

Figure 5.2 Diagram of the conducted QoE methodology.

For the *Shooter* game, a total of ten scenarios were tested, and they were grouped into three groups. The first group is comprised of four testing scenarios, and it evaluates the impact of aiming aids by enabling weapon laser sight, showing projectile trajectory, both, and neither. The second group tests three different target spawning angles (Figure 5.3) with the goal of evaluating users' willingness to turn around, and the overall affect it has on the user experience and effectiveness. The last group assesses three different projectile shooting forces that affect bullet speed and the curvature of the projectile hyperbola. The testing scenarios and their values are shown in Table 5.1.

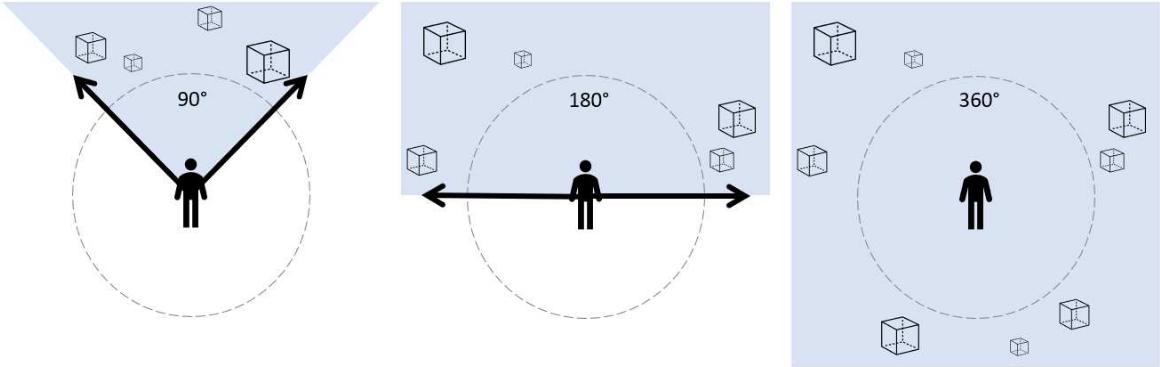

Figure 5.3 Different target spawning angles in the *Shooter* game.

Scenario identifier	Interaction parameter	Parameter value
	Group 1	
1	Use laser, Show bullet trajectory	True, True
2	Use laser, Show bullet trajectory	True, False
3	Use laser, Show bullet trajectory	False, True
4	Use laser, Show bullet trajectory	False, False
	Group 2	
5	Spawn angle	90°
6	Spawn angle	180°
7	Spawn angle	360°
	Group 3	
8	Shoot force	20
9	Shoot force	40
10	Shoot force	80

Table 5.1 Testing scenarios groups and values for the *Shooter* game.

For the *Box Smash* game, a total of nine scenarios grouped into three groups were tested. The first group is comprised of three testing scenarios, and it evaluates different target spawning angles. The second group tests three different minimum forces to destroy a target needed by the player. The last group assesses the effects of three different weapon lengths. The testing scenarios and their values are shown in Table 5.2.

Scenario identifier	**Interaction parameter**	**Parameter value**
	Group 1	
1	Spawn angle	90°
2	Spawn angle	180°
3	Spawn angle	360°
	Group 2	
4	Force to destroy	0
5	Force to destroy	2
6	Force to destroy	6
	Group 3	
7	Weapon length	1
8	Weapon length	0.7
9	Weapon length	1.3

Table 5.2 Testing scenarios groups and values for the *Box Smash* game.

For the *Pick-And-Place* game, a total of eleven scenarios grouped into four groups were tested. The first group is comprised of three testing scenarios, and it evaluates different puzzle scales. The second group tests three different collider scales, which impact the precision needed for a puzzle piece placement to be accepted and snapped into place. The third group compares gameplay when remote grab of puzzle pieces is enabled versus when remote grabbing is not available. The final testing group evaluates different puzzle piece scale offsets that affect the difficulty of placing puzzle pieces into the solution space when there are already some pieces in place. The testing scenarios and their values are shown in Table 5.3.

Scenario identifier	Interaction parameter	Parameter value
Group 1		
1	Puzzle scale	0.1
2	Puzzle scale	0.4
3	Puzzle scale	0.7
Group 2		
4	Collider scale	0.2
5	Collider scale	0.5
6	Collider scale	1
Group 3		
7	Remote grab	False
8	Remote grab	True
Group 4		
9	Scale offset	0.8
10	Scale offset	0.9
11	Scale offset	1

Table 5.3 Testing scenarios groups and values for the *Pick-And-Place* game.

The order of testing for each scenario group was the same for all of the test subjects, however, testing scenarios inside each scenario group were randomly ordered for each test subject. For each game, the round duration was limited to 90 seconds.

After each scenario, participants were asked to evaluate the total QoE on a linear scale of 1 to 5, with 1 representing "bad" and 5 representing "excellent". The next set of questions are inspired by The Game Experience Questionnaire [27] and evaluate the feeling of competence, how challenging the game was, and how entertaining the game was on a linear scale of 1 to 5, with 1 representing "very low" and 5 representing "very high". A set of questions evaluating perceived workload were adapted from the NASA-TLX/SIM-TLX questionnaires [28] [29], and participants were asked to assess the subjective mental demand, physical demand, and task control difficulty on a linear scale of 1 to 5, with 1 representing “very low” and 5 representing “very high”. The last two questions assessed the experienced muscle pain and the experienced physical discomfort on a linear scale of 1 to 5, with 1 representing “very low” and 5 representing “very high”. Finally, participants were asked if they were willing to continue playing the game under the evaluated scenario's conditions. The language of the questionnaire was Croatian.

After all the scenarios in the same testing group were evaluated, participants were asked to select the best and the worst scenario in that group, and the reason for each. The entire questionnaire is provided in the appendix.

6. Results and analysis

The study was conducted on a total of fifteen participants, divided into three groups, one group per game. The first group of five participants tested the *Shooter* game. The ages of this group's participants ranged from 21 to 34 years old, with an average age of 24.4. A total of three male participants and two female participants were a part of this group. All five participants indicated their right hand as their dominant hand. Two participants had no prior experience with VR, two participants had tried virtual reality previously but less than 3 times in their life, while one participant uses VR occasionally but less than once a month on average. All group participants had a positive attitude towards VR, with the score averaging at 4.4 out of 5, with 5 representing "excellent".

The second group of five participants tested the *Box Smash* game. The ages of these group participants ranged from 22 to 29 years old, with an average age of 24.2. A total of three male participants and two female participants were a part of this group. All five participants indicated their right hand as their dominant hand. One participant had no prior experience with VR, three participants had previously tried VR but less than 3 times in their life, while one participant uses VR once a month or more often. All group participants had a positive attitude towards VR, with the score averaging at 4.2 out of 5.

The third group of five participants tested the *Pick-And-Place* game. The ages of these group participants ranged from 21 to 31 years old, with an average age of 24.4. A total of three male participants and two female participants were a part of this group. All five participants indicated their right hand as their dominant hand. Two of the participants had no prior experience with VR, two participants had previously tried VR but less than 3 times in their life, while one participant uses VR once a month or more often. All group participants had a positive attitude towards VR, with the score averaging at 4.4 out of 5.

6.1. Shooter

6.1.1. Objective measures

The first tested group of scenarios in the *Shooter* game compared different target aiming aids. In the first scenario, both the laser sight and the bullet trajectory were shown. An average of 208.2 bullets were fired, with the lowest number of shots fired being 109 and the highest number of shots fired being 434. A total average of 129.2 shots were fired by the right hand, while an average of 79 shots were fired by the left hand, meaning an average of 62.1% of shots were fired by the dominant right hand. The highest average number of shots fired was when the test subjects had laser sights attached to the weapons. In this scenario, an average of 221.4 shots were fired, ranging from 170 to 365 shots fired. An average of 64.4% of the total shots fired were fired using the right hand. The lowest average number of shots fired was in the testing scenario where only bullet trajectory was shown, 144.8 shots. The number of shots fired ranged from 70 to 193 shots, with 71.1% of the total average shots being shot using the right hand. In the last testing scenario, where no aiming aid was provided, the average number of shots fired was 180.2 with the lowest number of shots fired being 56 and the highest number of shots fired being 314. An average of 66.7% of the total number of shots was fired by the right hand. The average number of shots fired by each hand can be seen in Figure 6.1.

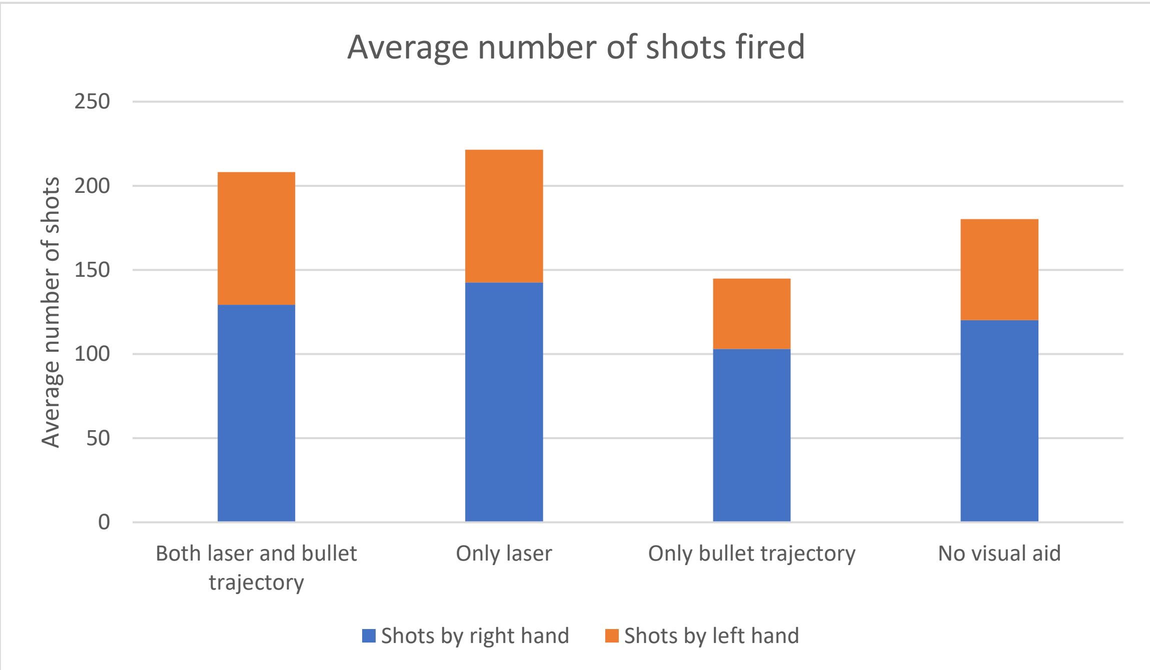

Figure 6.1 Average number of shots fired in the *Shooter* game using different aiming aids.

Accuracy, calculated as the ratio of shots that hit the target and the total number of shots fired, is shown in Figure 6.2. When both the laser sight and the bullet trajectory were visible, the total average accuracy was 66.6%, with an average accuracy of 67.9% for the right hand and 48.5% for the left hand. The average number of targets destroyed was 112.8. Participants achieved the best average accuracy of 72.9% when the laser sights aiming aid was enabled.

In that scenario, the average accuracy was 74.5% for the right hand and 53.3% for the left hand. The average number of targets destroyed was 148.4. The testing scenario in which only the bullet trajectory was shown had the lowest difference in average accuracy between the right hand and the left hand. The average number of targets destroyed was 91.2 and the average accuracy totaled 62.7%, with an average accuracy of 62.3% for the right hand, and 61.0% for the left hand. When no aiming aid was provided, the average accuracy amounted to 53.1%, 54.7% for the right hand and 35.3% for the left hand. An average of 86.2 targets were destroyed.

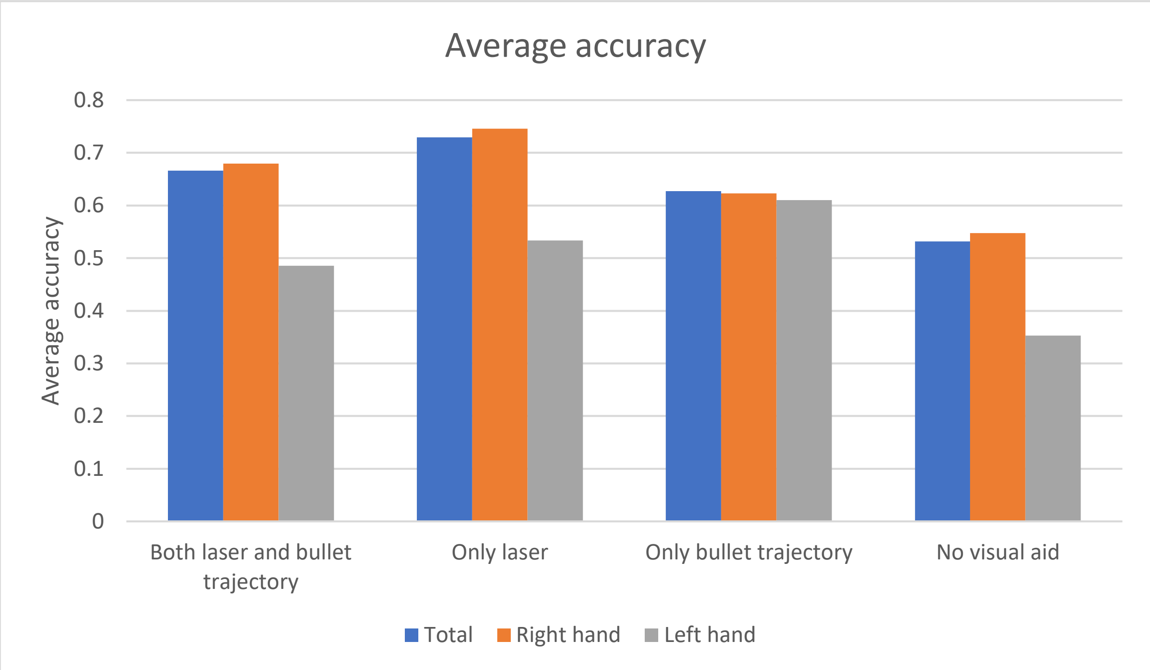

Figure 6.2 Average accuracies in the *Shooter* game using different aiming aids.

The second group of scenarios tested different circle segment angles the targets will spawn at with respect to the players position and orientation. When the spawn angle was 90°, the average number of shots fired was 210.8, with the number ranging from 101 to 437. Of the total number of shots fired, 60.1% were fired by the right hand. When the spawn angle was 180°, the total number of shots averaged at 176.8 out of which 62.9% were by the right hand. In this scenario, the number of shots fired ranged from 93 to 392 shots. The spawn angle of 360° had the lowest number of shots fired. Out of the average total of 114.4 shots fired, 64.3% were shot by the right hand. The average number of shots fired for each testing scenario can be seen in Figure 6.3.

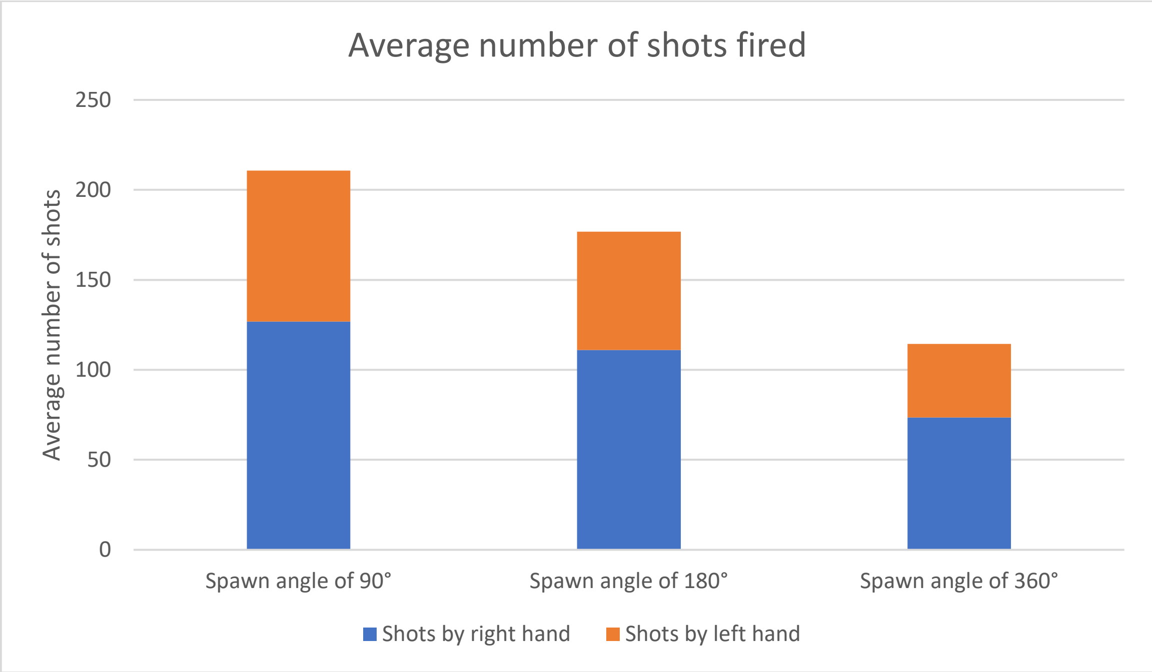

Figure 6.3 Average number of shots fired in the *Shooter* game using different spawning angles.

The average accuracy for all three scenarios is relatively similar. For the scenario in which the spawn angle was set to 90°, it averaged at 57.4% total, 58.8% for the right hand, and 41.0% for the left hand. An average of 98.8 targets were destroyed. When the spawn angle was set to 180°, the average accuracy was slightly lower, 54.6% in total, 54.1% for the right hand, and 41.4% for the left hand. The average number of targets destroyed amounted to 76.6. The average accuracy when the spawn angle was set to 360° totaled 56.9%, with an average accuracy of 59.1% for the right hand and 38.4% for the left hand. The number of destroyed targets averaged 57.6. The average accuracies are shown in Figure 6.4.

The last group of scenarios tested different shooting forces. The average number of shots fired is consistent across all three scenarios tested and is shown in Figure 6.5. When the shoot force was set to 20, the number of shots averaged at 204.6, ranging from 92 to 485 shots fired. 57.2% of the total shots were fired by the right hand. The average number of shots with the shoot force set to 40 amounted to 212, with the lowest number of shots being 114 and the highest number of shots counting at 378. Of the total number of shots fired, 61.4% were fired by the right hand. When the shoot force was set at 80, the average number of shots ranged from 155 to 274 with a total of 209.8 average shots fired and 63.2% of shots were fired by the right hand.

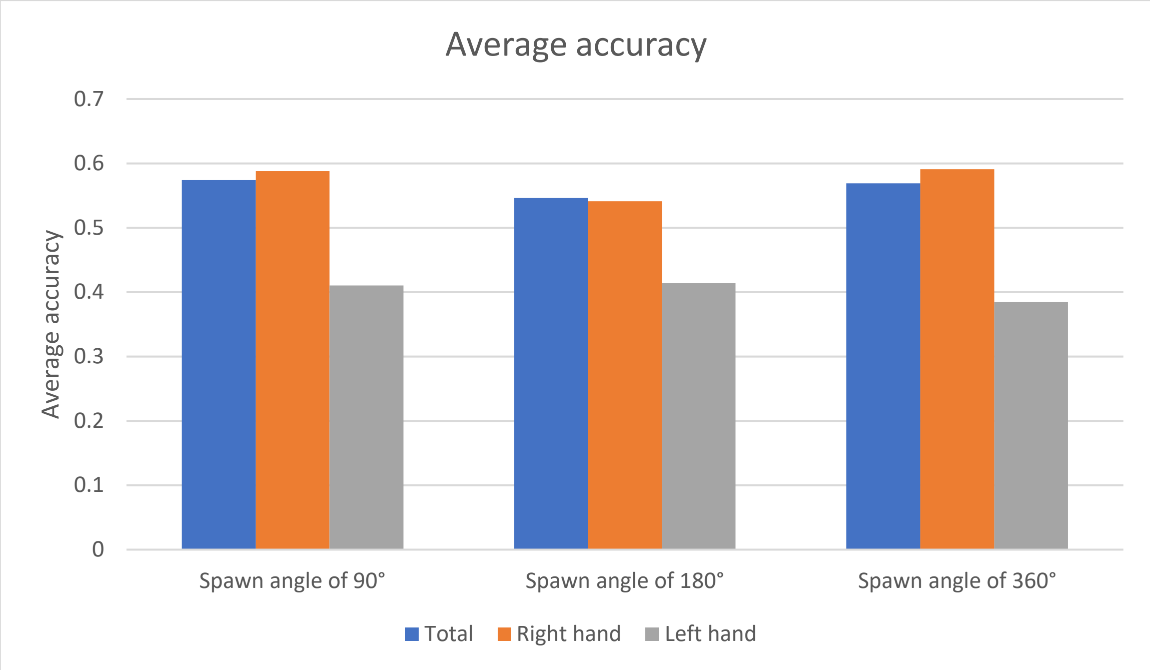

Figure 6.4 Average accuracies in the *Shooter* game using different spawning angles.

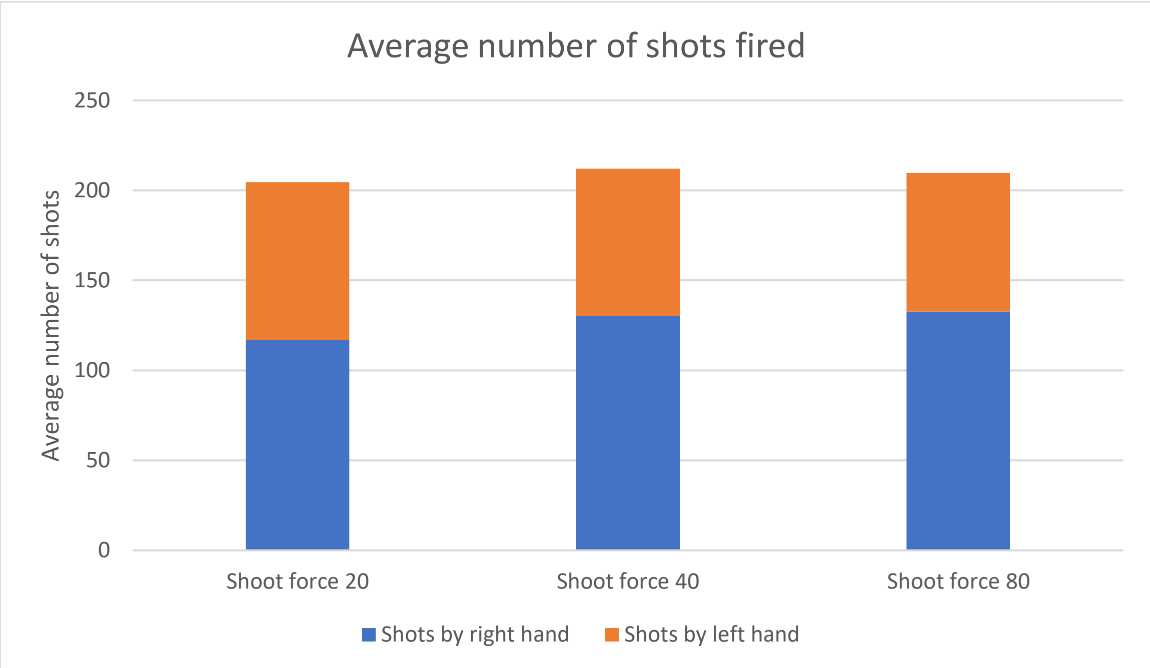

Figure 6.5 Average number of shots fired in the *Shooter* game using different shooting forces.

The average accuracy was lowest when the shoot force was set to 20, 49.9% in total, 53.2% for the right hand, and 32.7% for the left hand. An average of 77.8 targets were destroyed. With the shoot force set to 40, the average number of targets destroyed was 101.8, while the average accuracy amounted to 54.3% in total, 55.8% for the right hand, and 38.8% for the left hand. The highest average accuracy was achieved when the shoot force was set to 80. The average number of targets destroyed in this scenario was 123.2, the average accuracy was 60.7%, 64.1% for the right hand, and 39.9% for the left hand. The average accuracies for this group of scenarios are shown in Figure 6.6.

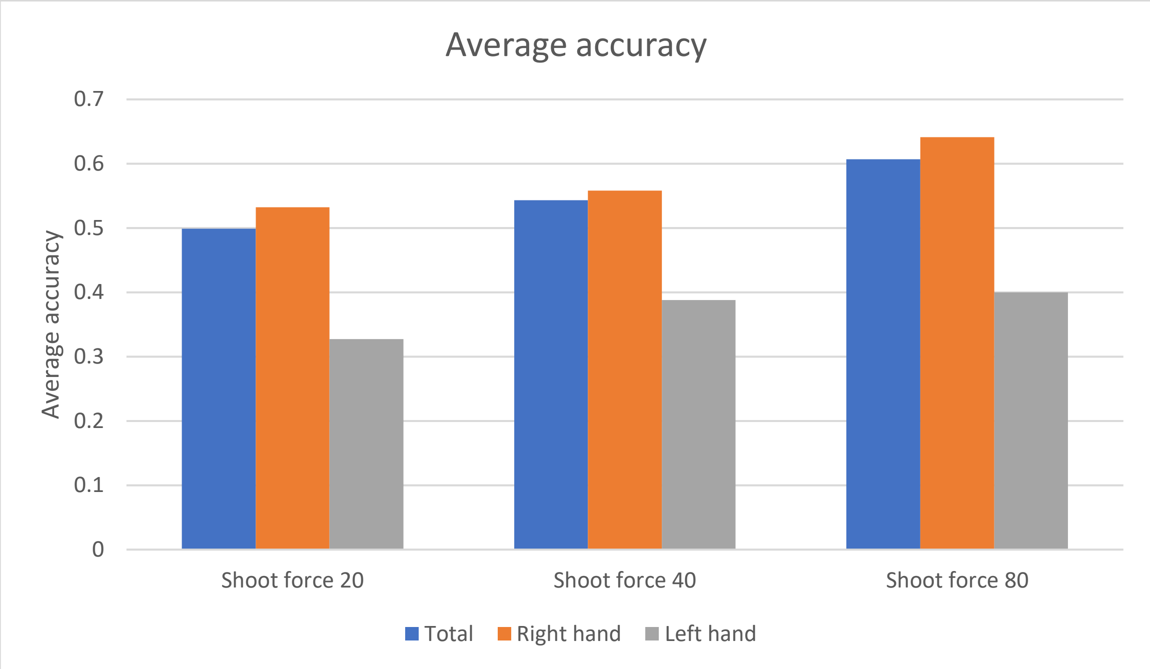

Figure 6.6 Average accuracies in the *Shooter* game using different shooting forces.

6.1.2. Subjective measures

The results of the post-experience questionnaire for the first group of scenarios that tested different aiming aids are shown in Figure 6.7. Participants were asked to rate the overall QoE for each scenario on a scale of 1 to 5, with 1 representing “bad” and 5 representing “excellent”. The average QoE scores are similar across all four scenarios, with the lowest score of 4.4 out of 5 belonging to the scenario in which both the laser sight and the bullet trajectory were visible, and the highest score of 4.8 out of 5 belonging to the scenario in which only the bullet trajectory was shown. The total QoE score for the scenario in which only the laser sight was attached and the scenario where no aiming aid was provided is the same and averages at 4.6 out of 5.

When both the laser sight and the bullet trajectory were provided, participants evaluated their competence in executing the game tasks with an average score of 4 out of 5, with 5 being very high. However, when asked to evaluate how challenging the game was, an average score of 2.2 was given, with a score of 1 representing very low. When evaluating the entertainment factor of the game, this scenario averaged 4 out of 5, the lowest of the four scenarios in this group.

In the testing scenario where the laser sight aiming aid was enabled, the average perceived competence amounted to 4.4 out of 5, the participants evaluated the game with an average score of 2.4 out of 5 for being challenging, and an average score of 4.8 out of 5 for the entertainment factor.

When only the bullet trajectory was shown, the average score for perceived competence averaged at 3.4 out of 5, which is the lowest average score in this group of testing scenarios. Correspondingly, the average score of how challenging the game was the highest for this scenario and averages at 3.2 out of 5. The entertainment factor of the game was evaluated the same as in the scenario where only the laser sight was attached to the weapon, 4.8 out of 5, and these scores are the highest in this group of scenarios.

The average evaluated feeling of competence in the scenario where no aiming aid was provided amounted to 4 out of 5, the same as when both laser sight and bullet trajectory were shown. When evaluating how challenging the game was, an average score of 3 out of 5 was given, and an average score of 4.6 out of 5 was given in the assessment of the game's entertainment factor.

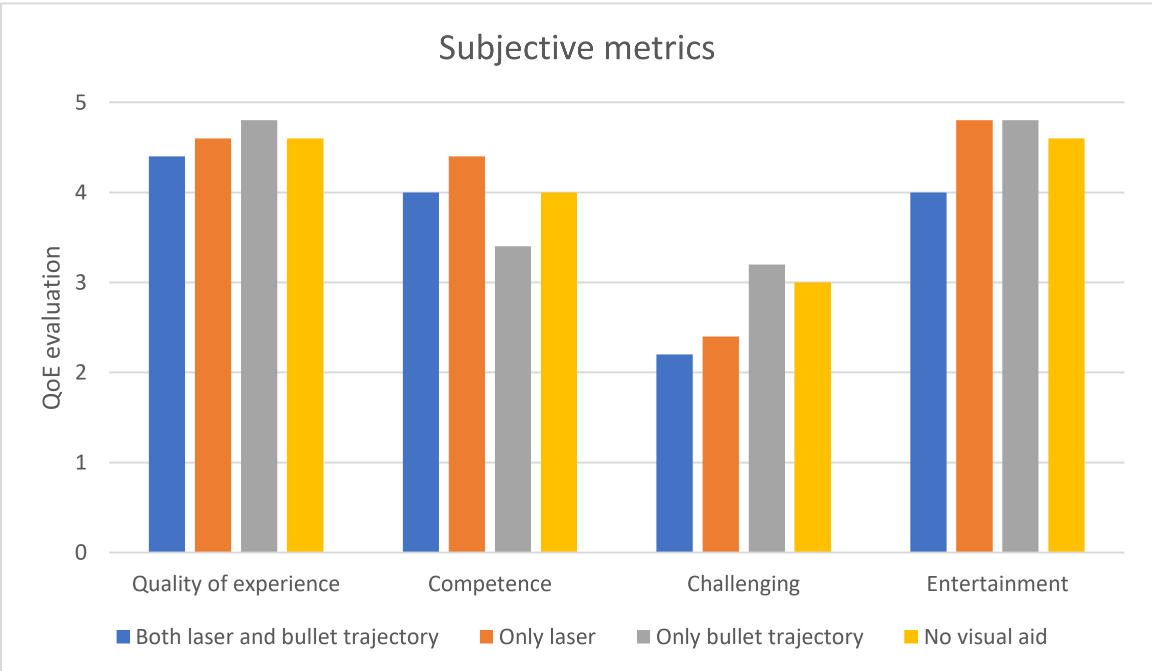

Figure 6.7 Evaluation of QoE (on a scale of 1: bad to 5: excellent), competence, challenge, and entertainment (on a scale of 1: very low to 5: very high) in the Shooter game using different aiming aids.

Figure 6.8 shows the simulation workload measures evaluation results for the group of scenarios in which different aiming aids were tested. Participants were asked to evaluate these measures on a scale of 1 to 5, with 1 representing very low, and 5 representing very high. When both the laser and the bullet trajectory were used, the mental demand was evaluated with an average score of 1.8 out of 5, the physical demand was evaluated with an average score of 1.2 out of 5, and the difficulty of navigating and controlling was evaluated with an average score of 2 out of 5. The mental demand and the physical demand average scores are identical for the scenario in which only the laser sight was attached, and in the scenario where only the bullet trajectory was shown, and it amounts to 1.4 out of 5 for both measures. However, the difficulty of task control in the scenario where only the laser was shown is higher compared to the scenario in which only the bullet trajectory was shown. It averages at 1.6 for the former, and 1.4 for the latter.

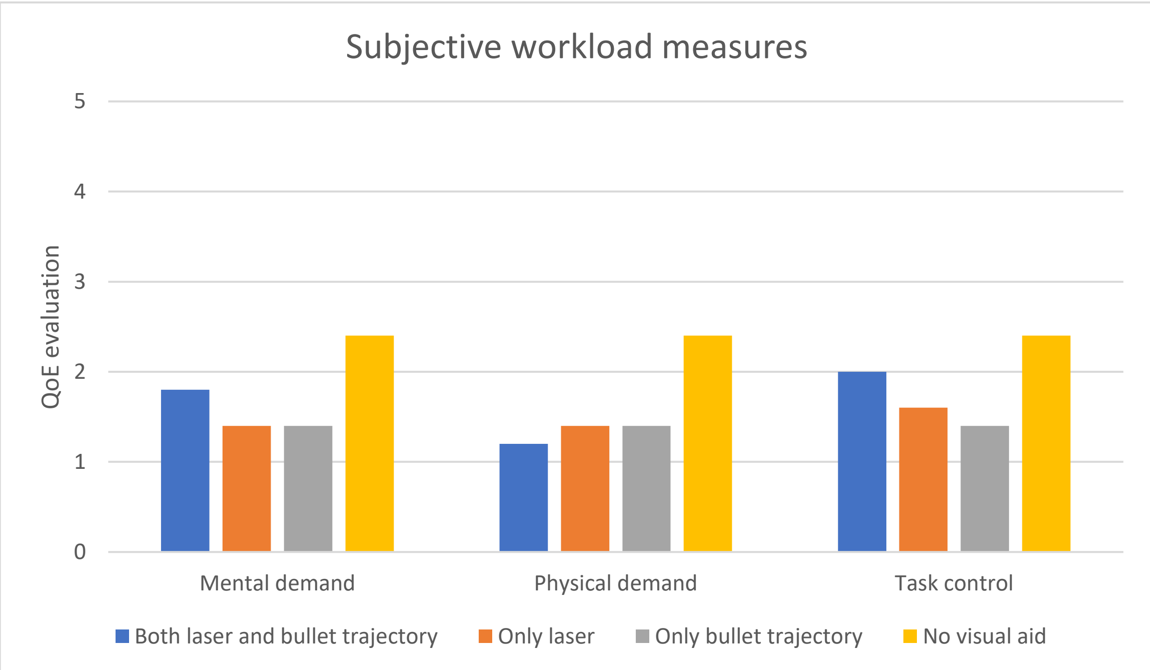

Figure 6.8 Evaluation of simulation workload in the *Shooter* game using different aiming aids (on a scale of 1: very low to 5: very high).

The scenario in which no aiming aid was provided has the highest average scores for all three workload measures. The evaluation of mental demand, the evaluation of physical demand, and the evaluation of task control difficulty all average at 2.4 out of 5.

Additionally, participants were asked to express their willingness to resume gameplay under the test scenario's conditions. For all the scenarios in this group, participants were willing to resume the gameplay, except for one participant in the testing scenario in which both the laser and the bullet trajectory were shown, who was unwilling to resume the game in those conditions.

Out of the four tested scenarios, 3 out of 5 participants chose the scenario in which no aiming aid was provided as the best scenario, expressing closeness to reality and a greater feeling of challenge as the deciding factors. One participant chose the scenario with laser sight enabled as the best, citing a greater sense of competence, and one participant chose the scenario where bullet trajectory was shown as the best because it enabled them to adjust their aim more efficiently. On the other hand, 3 out of 5 participants chose the scenario in which the bullet trajectory was shown as the scenario they liked the least, expressing a feeling of less competence and frustration due to the parabolic trajectory of the bullet. The remaining two participants chose the scenario where both the laser and the bullet trajectory were shown because the differences between the laser and the bullet trajectory were too distracting.

The results of the post-experience questionnaire for the second group of scenarios in which different target spawning angles were evaluated are shown in Figure 6.9. The participants were asked to evaluate the total QoE for each scenario on a scale of 1 to 5, with 1 representing very low and 5 representing very high QoE. The evaluated QoE is the same for the scenario in which the spawn angle was set to 90 degrees and the scenario in which the spawn angle was set to 360 degrees, and it averages at 4.6 out of 5. In the scenario where the spawn angle was set to 180 degrees, the average QoE was 5 out of 5.

When the target spawn angle was set to 90 degrees, participants evaluated their competence in executing the game tasks with an average score of 4.2 out of 5, which is the highest score compared to the rest of the scenarios in this group. However, when asked to evaluate how challenging the game was, an average score of 2.2 was given, with a score of 1 representing very low. When evaluating the entertainment factor of the game, this scenario averaged 4.6 out of 5.

In the scenario where the spawn angle was set to 180 degrees, the average perceived competence amounted to 3.8 out of 5, the average score of how challenging the game was equaled to 3 out of 5, while the entertainment factor averaged at 4.8 out of 5. When the spawn angle was set to 360 degrees, the participant felt the least competent, with an average score of 3.2 out of 5, and they evaluated the difficulty of the game as 3.4 out of 5. The average evaluation of the game's entertainment factor is the same as when the spawn angle was set to 180 degrees, and it amounts to 4.8 out of 5.

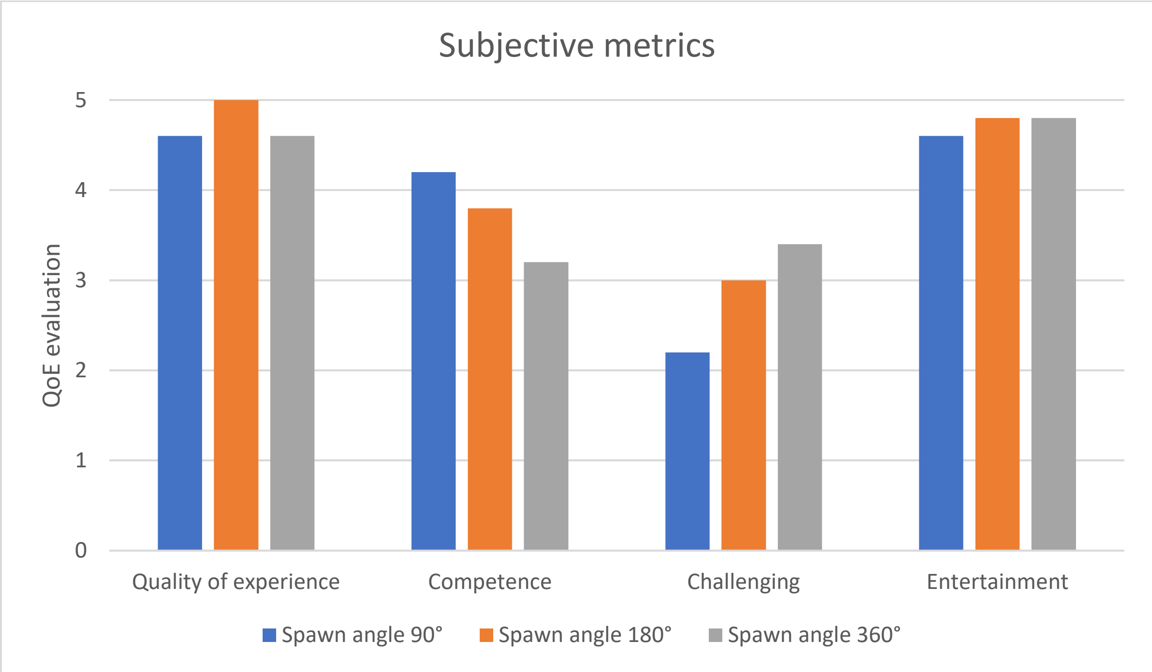

Figure 6.9 Evaluation of QoE (on a scale of 1: bad to 5: excellent), competence, challenge, and entertainment (on a scale of 1: very low to 5: very high) in the *Shooter* game using different target spawning angles.

Figure 6.10 shows the simulation workload measures evaluation results for the group of scenarios in which different target spawn angles were evaluated. Participants were asked to evaluate these measures on a scale of 1 to 5, with 1 representing very low, and 5 representing very high. The average mental demand evaluated was lowest when the spawn angle was set to 90 degrees, and it amounted to 1.2 out of 5. In the scenario where the spawn angle was 180 degrees, the average mental load was 1.4 out of 5, and when the spawn angle was set to 360 degrees, the game was estimated to be the most mentally demanding, with an average score of 2 out of 5. The estimated average physical demand is identical across all three scenarios, and it amounts to 1.6 out of 5. The evaluated task control difficulty is the lowest for the scenario in which the spawn angle was 90 degrees, and it averages at 1.4 out of 5. When the spawn angle was set to 180 degrees, the task control difficulty averaged at 1.8, and when the spawn angle was set to 360 degrees, it was the highest, averaging at 2.2 out of 5.

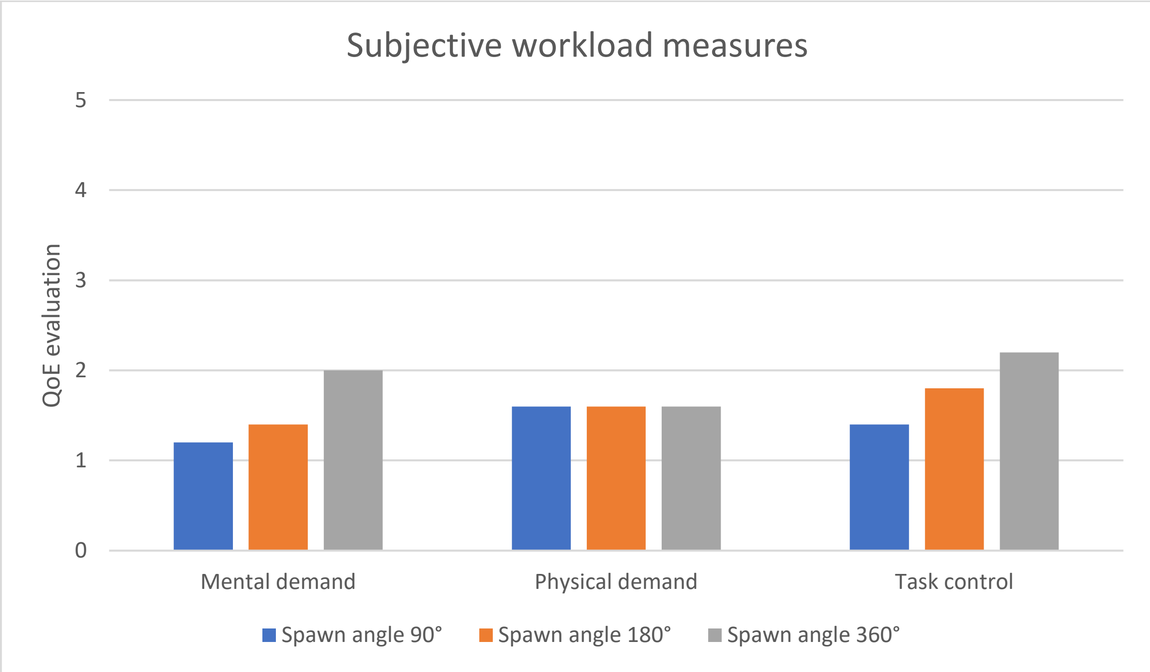

Figure 6.10 Evaluation of simulation workload in the *Shooter* game using different target spawning angles (on a scale of 1: very low to 5: very high).

For all the scenarios in this group, all participants were willing to resume the gameplay. When asked to select the best scenario in this group, 3 out of 5 participants selected the scenario in which the spawn angle was set to 360 degrees, stating it was the most challenging and the most entertaining. One participant selected the scenario in which the spawning angle was set to 180 degrees because they had the feeling the targets were coming from all sides, while the remaining participant chose the scenario in which the spawn angle was set to 90 degrees because they had the impression of the bullet moving faster.

When asked to select the scenario which they liked the least, 3 out of 5 participants selected the scenario in which the spawn angle was set to 180 degrees, stating the least entertainment and difficulty navigating. The remaining two participants selected the scenario in which the spawn angle was set to 90 degrees because it was the least entertaining.

The results of the post-experience questionnaire for the first group of scenarios which evaluated different shoot forces are shown in Figure 6.11. In this testing group, the average QoE was lowest in the scenario in which the shoot force was set to 20, and it amounted to 4.4 out of 5. The feeling of competence was also the lowest for this scenario, averaging at 3.4 out of 5. On the other hand, participants rated this scenario as the most challenging in this group, with an average score of 2.8 out of 5. Moreover, this scenario was the least entertaining compared to the other two scenarios in this group, with an average score of 4.4

out of 5. When the shoot force was set to 40, the QoE averaged at 4.6 out of 5, the average feeling of competence amounted to 4.2 out of 5, the average sense of challenge was 1.6 out of 5, and the average entertainment score was 4.6 out of 5. The highest average QoE score was when the shoot force was set to 80, and it amounted to 4.8 out of 5. The feeling of competence was also the highest in this scenario, averaging 4.8 out of 5. An average score of 1.8 out of 5 was given for the difficulty, while the entertainment factor scored an average of 4.8 out of 5.

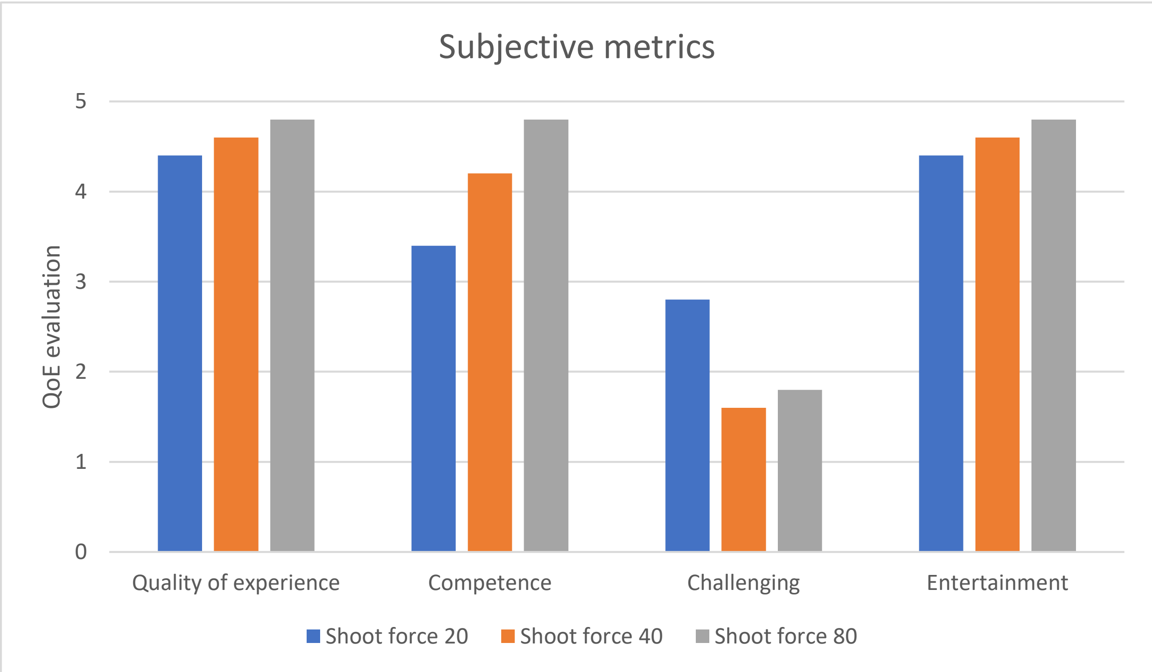

Figure 6.11 Evaluation of QoE (on a scale of 1: bad to 5: excellent), competence, challenge, and entertainment (on a scale of 1: very low to 5: very high) in the *Shooter* game using different shooting forces.

Figure 6.12 shows the simulation workload measures evaluation results for the group of scenarios in which different shoot forces were evaluated. The mental demand was evaluated the highest for the scenario in which the shoot force was set to 20, and it averaged at 2.4 out of 5. The remaining two scenarios both have an average mental demand score of 1.2 out of 5. All three scenarios have similar average scores for the physical demand measure, with the scenario in which the shoot force was 20 having an average score of 1.6, the scenario in which the shoot force was 40 having an average score of 1.2, and the scenario in which the shoot force was set to 80 having an average score of 1.4 out of 5. When the shoot force was set to 20, the average task control difficulty was 2.4, when the shoot force was set to 40, it averaged at 1.8, and when the shoot force was set to 80, it had an average value of 1.6 out of 5.

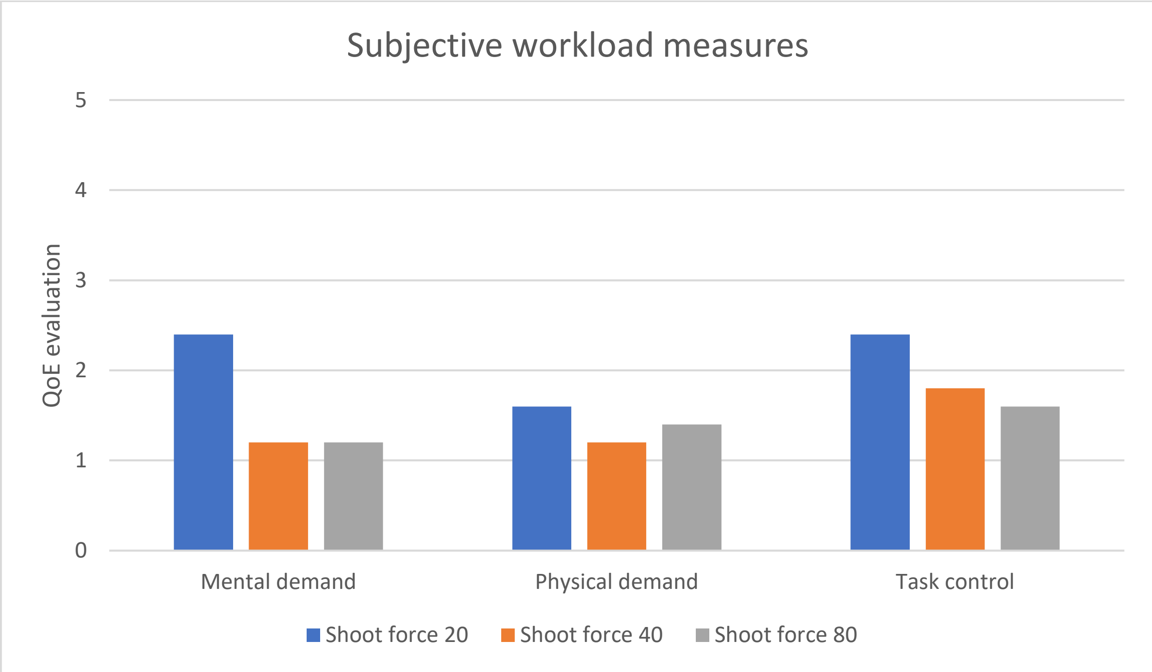

Figure 6.12 Evaluation of simulation workload in the *Shooter* game using different shooting forces (on a scale of 1: very low to 5: very high).

When the shoot force was set to 20, 2 out of 5 participants were unwilling to continue the gameplay in those conditions. In the remaining two scenarios in this group, all participants were willing to continue playing the game under the scenario's conditions. When asked to select the best scenario in this group, 4 out of 5 participants chose the scenario with the shoot force of 80 because it made the game more realistic and the shooting more efficient and entertaining, while the remaining participant did not perceive any differences between the scenarios in this group. Participants were also asked to choose their least favorite scenario in this group, and 4 out of 5 participants chose the scenario in which the shoot force was set to 20 because the bullet was too slow. The remaining participant did not perceive any differences between the scenarios in this group.

Subjective discomfort measures of experienced muscle pain and physical discomfort collected in this study were not analyzed in this thesis. However, this data is available and can be analyzed in future work.

6.1.3. Analysis

When both laser sight and bullet trajectory were shown, there was the least difference between the number of shots fired by the dominant right hand and the number of shots fired by the left hand, and 62.1% of shots were fired by the right hand, which means that the participants were more likely to use both hands equally in this scenario. On the other hand, in the scenario where only the bullet trajectory was shown, the difference between the number of shots fired by each hand is the highest, averaging at 71.7% of shots being shot by the right hand. However, the average accuracies for the right hand and for the left hand are close to identical in this scenario, while for the other scenarios in this group, the left-hand accuracies are significantly lower than the accuracy of the right hand. Moreover, in this scenario, the least number of shots were fired compared to the rest of the scenarios in this group, which could indicate that the participants took more time to aim in this scenario compared to the rest of the scenarios in this group.

Figure 6.13 shows the comparison of the objective recorded accuracy, the subjective evaluated feelings of competence, and the subjective evaluated feeling of entertainment. The subjective measures in this chart are expressed as a percentage of the maximum possible score for that metric.

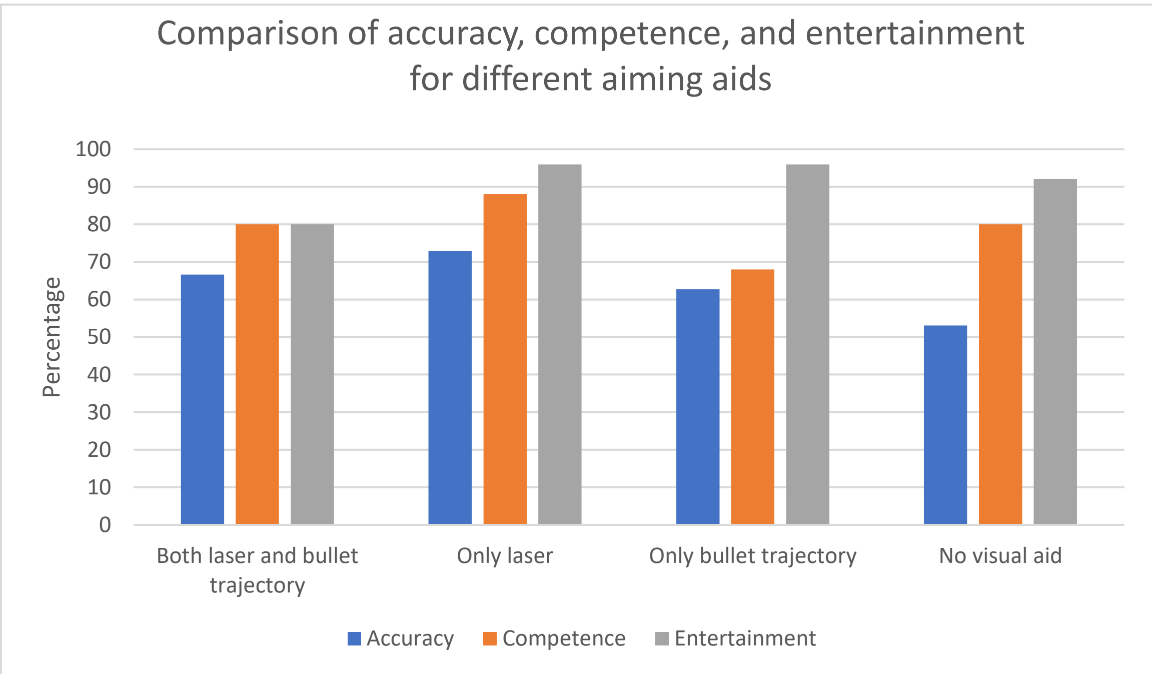

Figure 6.13 Comparison of accuracy, competence, and entertainment in the *Shooter* game using different aiming aids.

Even though the average objective shooting accuracy was the lowest when no aiming aid was provided, the average subjective feeling of competence was the lowest when only the bullet trajectory was shown. However, this low feeling of competence did not have a negative impact on the game's entertainment factor, as this scenario scored the highest, alongside the scenario where only the laser was shown, with an average score of 4.8 out of

5. The highest objective accuracy, which was recorded for the scenario where only the laser was shown, corresponds with the highest subjective feeling of competence.

In the subjective workload measures, both the scenario where only a laser sight was provided and the scenario where only the bullet trajectory was shown, scored similarly for measures of mental demand, physical demand, and task control difficulties. When both laser and bullet trajectory were shown, participants expressed a higher average mental demand and task control difficulty, and 40% of participants selected this scenario as the least favorite in this group because using both aiming aids simultaneously was distracting and confusing. This scenario also had the lowest average entertainment score. The remaining 60% of the participants selected the scenario in which only the bullet trajectory was shown as the least favorite in this group, because of the pronounced downward parabolic curve of the bullet's path, which participants found unrealistic and disadvantageous for aiming. However, using different shooting forces that change the speed and path of the bullet could change the QoE in this scenario. The scenario where no visual aid was provided has the highest average scores for mental demand, physical demand, and task control difficulty. Despite that, 60% of participants selected this scenario as the best in this testing group, finding it the most entertaining. As the testing round lasted only 90 seconds, the negative impact of higher average mental demand, physical demand, and control difficulty recorded in this testing scenario did not influence the overall QoE.

The average objective accuracies measured for the group of scenarios where the spawn angle was evaluated is similar for all three scenarios. The perceived competence, however, is the highest for the scenario where the spawn angle was set to 90 degrees (4.2), lower in the scenario where the spawn angle was set to 180 degrees (3.8), and the lowest when the spawn angle was set to 360 degrees (3.2). This can be attributed to the different average number of shots fired in each scenario, with the highest average being when the spawn angle was set to 90 degrees (210.8), and the lowest when the spawn angle was set to 360 degrees (114.4). With a higher target spawning angle, participants had to spend more time finding the targets, so fewer targets were destroyed compared to scenarios with a lower spawn angle, contributing to a lower sense of competence. However, different feelings of competence had no impact on perceived average entertainment and QoE because they are similar for all three scenarios in this testing group. The comparison of the objective recorded accuracy, the subjective evaluated feelings of competence, and the subjective evaluated feeling of entertainment of the scenarios in this group is shown in Figure 6.14.

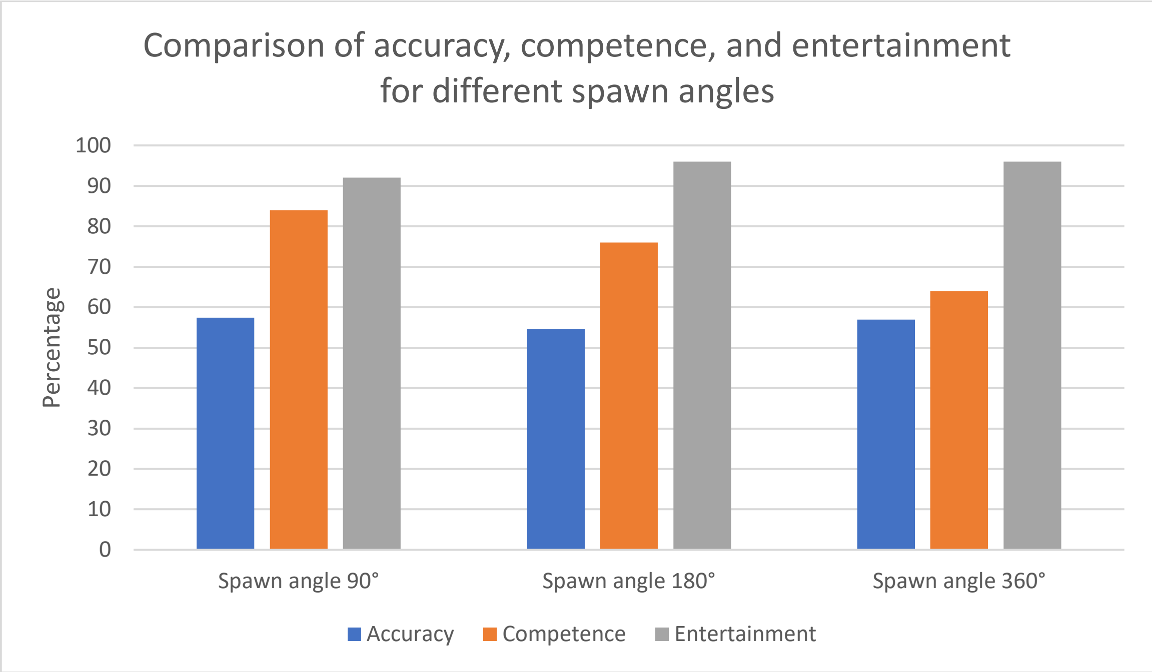

Figure 6.14 Comparison of accuracy, competence, and entertainment in the *Shooter* game using different target spawning angles.

The average physical demand is the same for all three scenarios, however, the average mental demand and task control difficulty increase as the spawn angle is increased. Most of the participants (60%) selected the scenario where the spawn angle was 360 degrees as the favorite scenario in this group because it was the most challenging and the most entertaining. But, considering the higher average workload measures and the relatively short round duration, longer gameplay under this scenario condition could be evaluated more negatively.

In the group of scenarios where different shooting forces were evaluated, the average number of shots fired was similar for all three scenarios. Both hands were used more equally when the shoot force was set to 20 (57.2% of the total shots were fired by the right hand), compared to when the shoot force was set to 40 (61.4% of the total shots were fired by the right hand) and when the shoot force was set to 80 (63.2% of the total shots were fired by the right hand). Figure 6.15 shows the comparison of the objective recorded accuracy, the subjective evaluated feelings of competence, and the subjective evaluated feeling of entertainment. The average objective accuracies are lower when the shoot force was lower, averaging at 49.9% for the shoot force of 20, 54.3% when the shoot force was 40, and 60.7% when the shoot force was 80. Similarly, the averaged perceived competence is lowest when the shoot force was 20 (3.4), higher when the shoot force was set to 40 (4.2), and the highest when the shoot force was set to 80 (4.8). Participants found the scenario in which the shoot force was set to

20 to be the most challenging, with an average score of 2.8, compared to the scenario with the shoot force set to 40 (1.6) and the scenario with the shoot force set to 80 (1.8). The QoE scores and the entertainment scores averaged 4.4 when the shoot force was 20, 4.6 when the shoot force was 40, and 4.8 when the shoot force was 80.

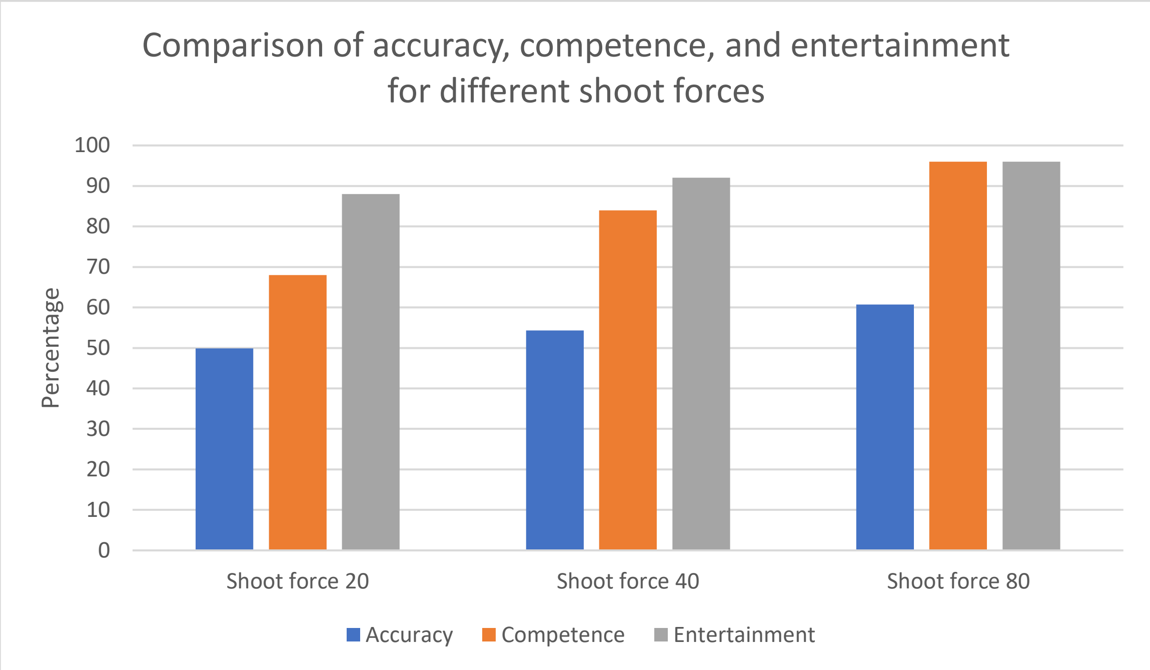

Figure 6.15 Comparison of accuracy, competence, and entertainment in the *Shooter* game using different shooting forces.

The average mental demand and the average task control difficulty were the highest when the shoot force was set to 20, resulting in 2 out of 5 participants expressing unwillingness to continue the gameplay under those conditions. Moreover, all but one participant selected this scenario as the least favorite because the more pronounced bullet path curve made it more difficult to aim, and the slower bullets caused frustration. On the other hand, all but one participant selected the scenario with the shoot force set to 80 as the best in this group because it was more entertaining, and the participants felt more competent. One participant did not notice any difference between the scenarios in this group.

6.2. Box Smash

6.2.1. Objective measures

The first tested group of scenarios in the *Box Smash* game compared different target spawning angles. In the testing scenario where the spawn angle was set to 90 degrees, the average number of targets spawned was 102.4, and a total average of 64.8 targets were destroyed, making the average accuracy 63.3%. Out of the total average of 37.6 missed targets, 26.4 were missed completely, while the remaining 11.2 were not hit with the minimum required force. When the spawn angle was set to 180 degrees, a total average of 102.8 targets were spawned, out of which 55.6 targets were destroyed, 39.2 were missed completely, and 8 were not hit with enough force. The average accuracy in this scenario amounts to 54.1%. A total average of 101.8 targets were created when the spawn angle was set to 360 degrees. From those targets, an average of 34 were destroyed, 62.2 were missed completely, and 5.6 were hit with a force less than the minimum required force. The average accuracy in this scenario is 33.3%. The average number of hits and misses for this group of scenarios is shown in Figure 6.16.

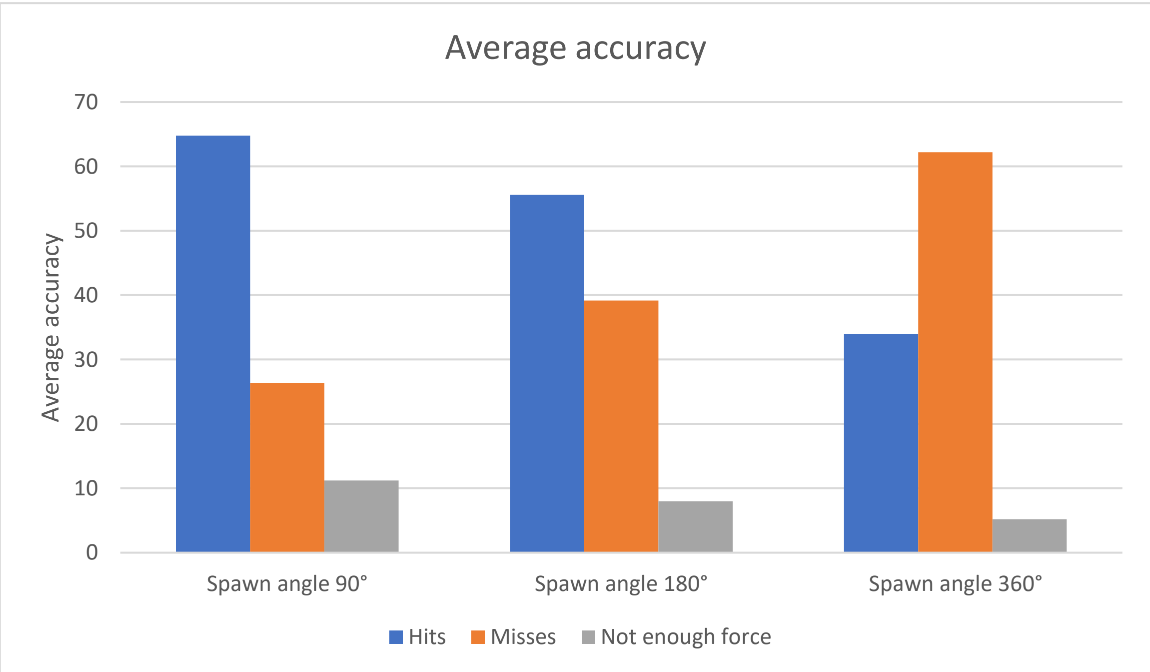

Figure 6.16 The average accuracies in the *Box Smash* game using different target spawning angles.

Figure 6.17 shows the average number of targets destroyed by the right hand and the average number of targets destroyed by the left hand in this group of scenarios. When the spawn angle was set to 90 degrees, 77.2% of the targets were destroyed by the dominant right hand.

When the spawn angle was 180 degrees, 66.5% of targets were destroyed by the right hand, and when the spawn angle was set to 360 degrees, 68.2% of destroyed targets were destroyed using the right hand.

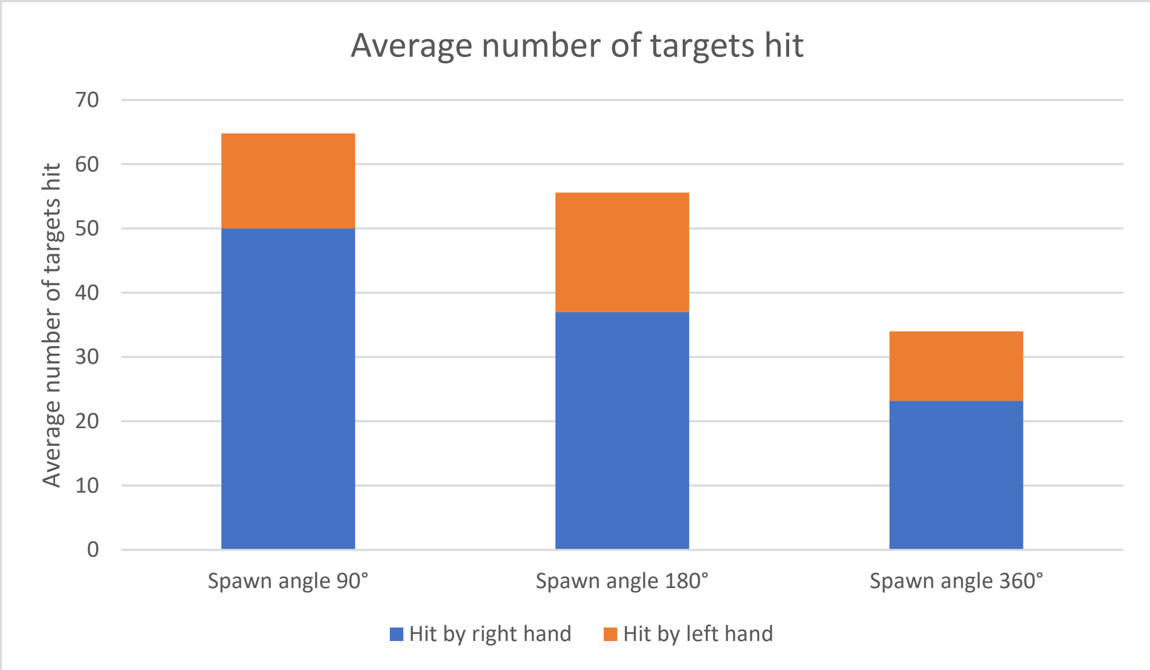

Figure 6.17 The average number of targets hit in the *Box Smash* game using different target spawning angles.

The average hitting force is similar across all three scenarios in this testing group. In the scenario where the spawn angle is 90 degrees, the total average force is 3.8, while the average force on destroyed targets is 4.4. When the spawn angle was 180 degrees, the average force was 4.0 and 4.5 on the destroyed targets. The average force on targets when the spawn angle was set to 360 degrees was 4.1 for all targets and 4.6 for the destroyed targets. The average hitting forces for this group of scenarios are shown in Figure 6.18.

The second group of scenarios evaluated different minimum forces required to destroy the targets. In the first scenario, where the minimum force was set to 0, the average number of targets spawned was 104.2. An average of 80 targets were destroyed, and 24.2 were missed completely, making the average accuracy amount to 76.8%. When the minimum force of 2 was required, a total average of 103.2 targets were spawned, out of which 71.2 were destroyed, 23 were missed completely, and 9 were not hit with enough force. The accuracy in this scenario averaged at 69.0%. In the scenario with the minimum required force of 6, a total average of 99.2 targets was spawned, 32.6 targets were destroyed, 26.8 targets were missed, and 39.8 targets were hit with a force less than the minimum required force. The accuracy in this scenario averaged at 32.8%. The average number of hit and missed targets in this testing scenario is shown in Figure 6.19.

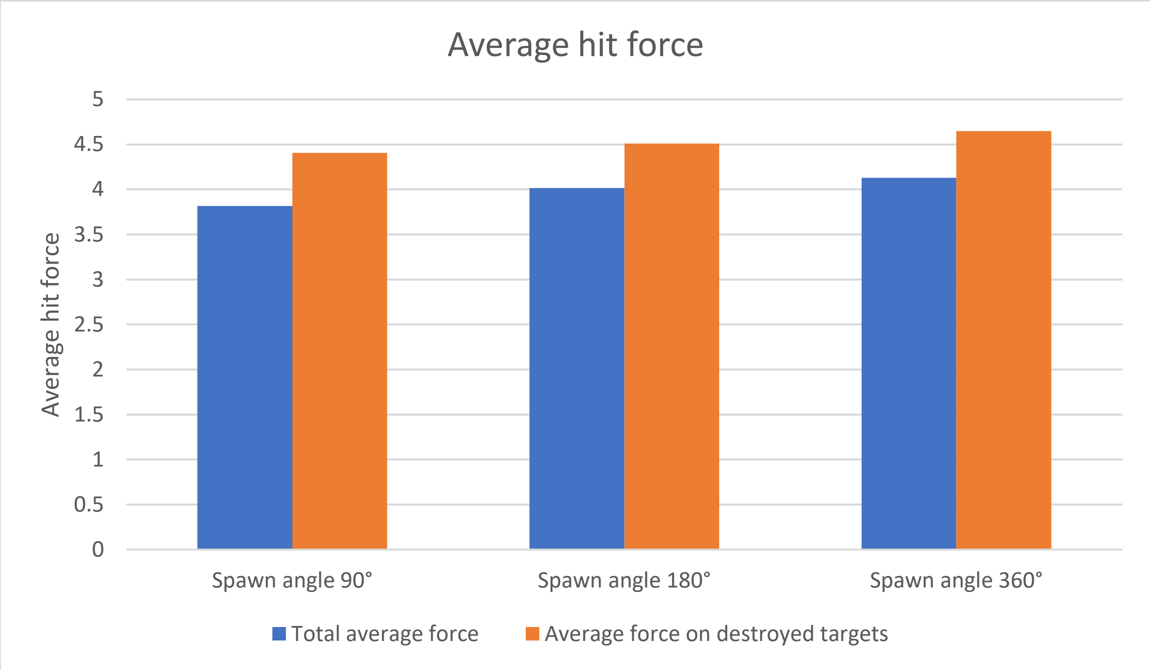

Figure 6.18 Average hit force in the *Box Smash* game using different target spawning angles.

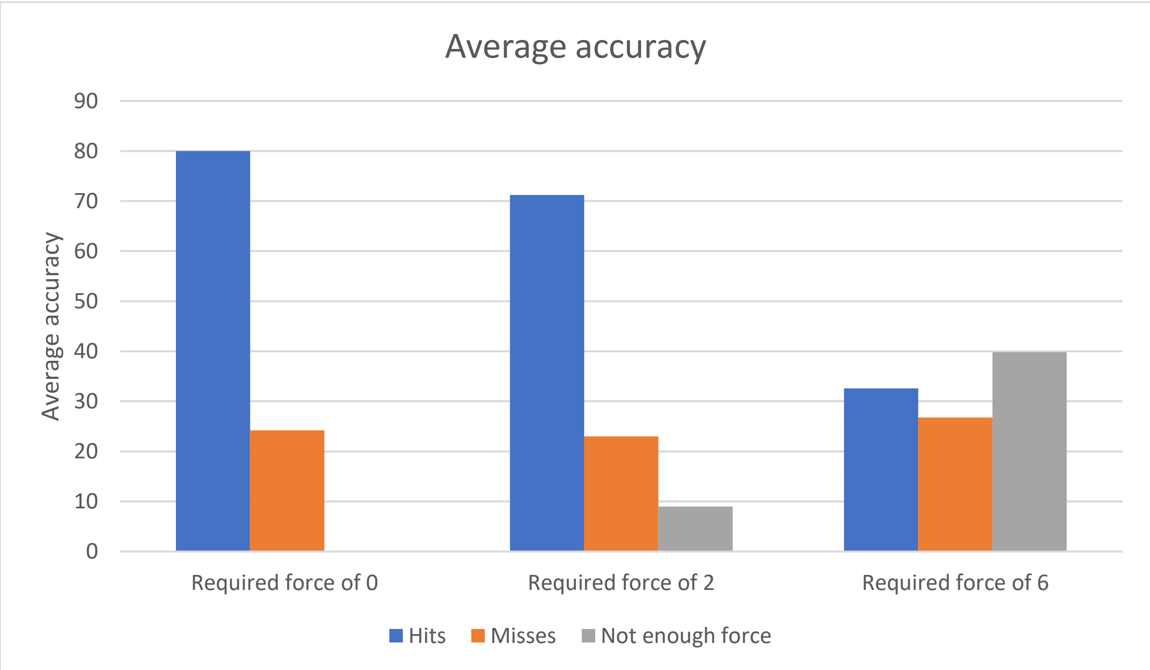

Figure 6.19 Average accuracies in the *Box Smash* game using different minimum required forces.

Figure 6.20 shows the average number of targets destroyed by the right hand and the targets destroyed by the left hand in the group of scenarios that evaluate different minimum forces required to destroy the targets. When the minimum force was set to 0, an average of 65.5%

of the targets were destroyed by the right hand, when the minimum required force was set to 2, an average of 64% of targets were destroyed by the right hand, and when the minimum required force was set to 6, an average of 75.5% of destroyed targets were destroyed by the right-hand weapon.

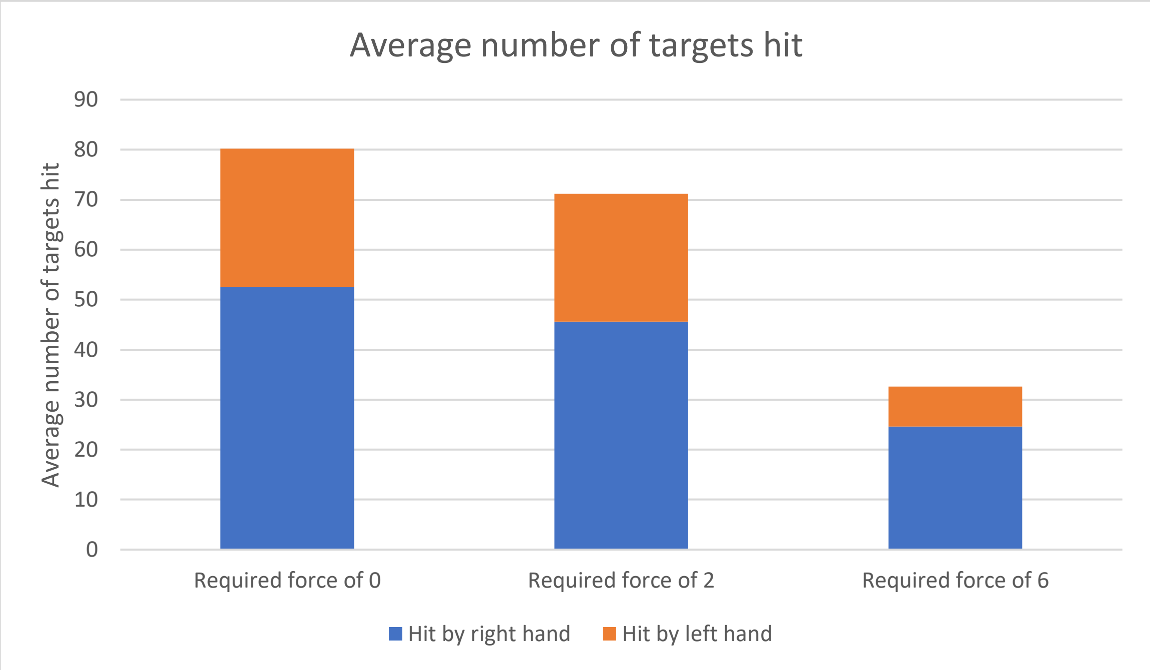

Figure 6.20 Average number of targets hit in the *Box Smash* game using different minimum required forces.

In the scenario where the minimum force required to destroy a target was set to 0, targets were hit with an average force of 2.8. When the minimum required force was set to 2, the average force on targets was 3.8 for all targets and 4.2 for the destroyed targets. The average force on targets when the minimum required force was set to 6 amounted to 5.1 for all targets and 6.9 for the destroyed targets. The average forces on targets in this group of scenarios are shown in Figure 6.21.

In the last group of scenarios, the scale of weapon length was evaluated, and the average number of hit and missed targets is shown in Figure 6.22. When the weapon scale was set to 1, the average number of targets spawned was 101.8, an average of 71.6 targets were destroyed, 20 targets were missed completely, and 10.2 were hit with a force less than the minimum required force. The average accuracy in this scenario amounted to 70.3%. With the weapon length scale of 0.7, a total average of 103.2 targets were spawned, 69.4 targets were destroyed, 25.6 were missed completely, and 8,2 were not hit with enough force, making the average accuracy in this scenario 67.2%. In the scenario where the scale of the weapon's length was set to 1.3, a total average of 102.6 targets were spawned, 71.2 targets were destroyed, 18.4 targets were missed, and 13 targets were hit with a force less than the minimum required force. The accuracy in this scenario averaged at 69.3%.

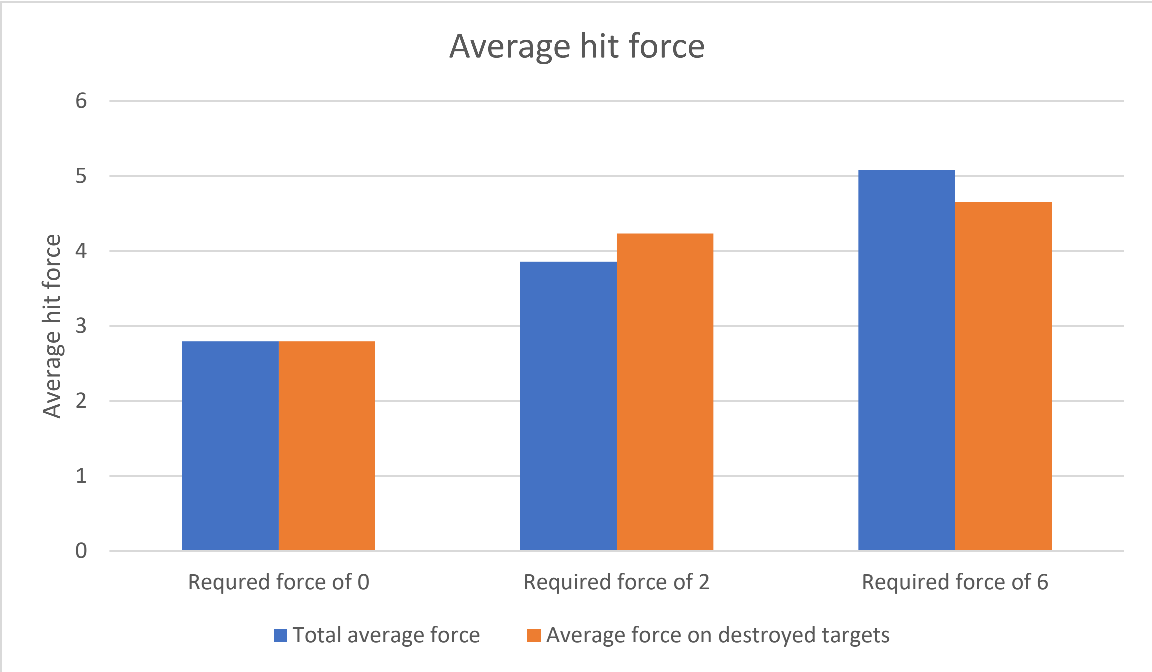

Figure 6.21 Average hit force in the *Box Smash* game using different minimum required forces.

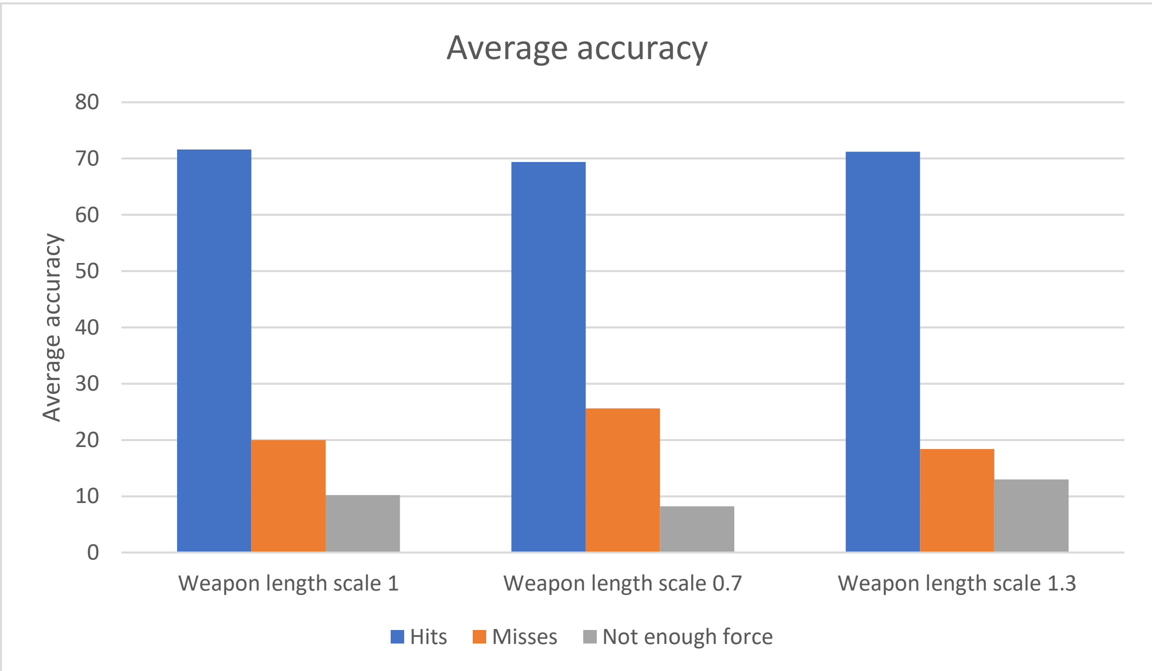

Figure 6.22 Average accuracies in the *Box Smash* game using different weapon scales.

In the scenario where the weapon scale was 1, 67% of targets destroyed were destroyed by the right-hand weapon. When the scale was set to 0.7, 60.8% of targets were destroyed by the right hand, and when the scale of the weapon was set to 1.3, 66.8% of the destroyed targets were hit by the right hand. The average number of targets destroyed by each hand is shown in Figure 6.23.

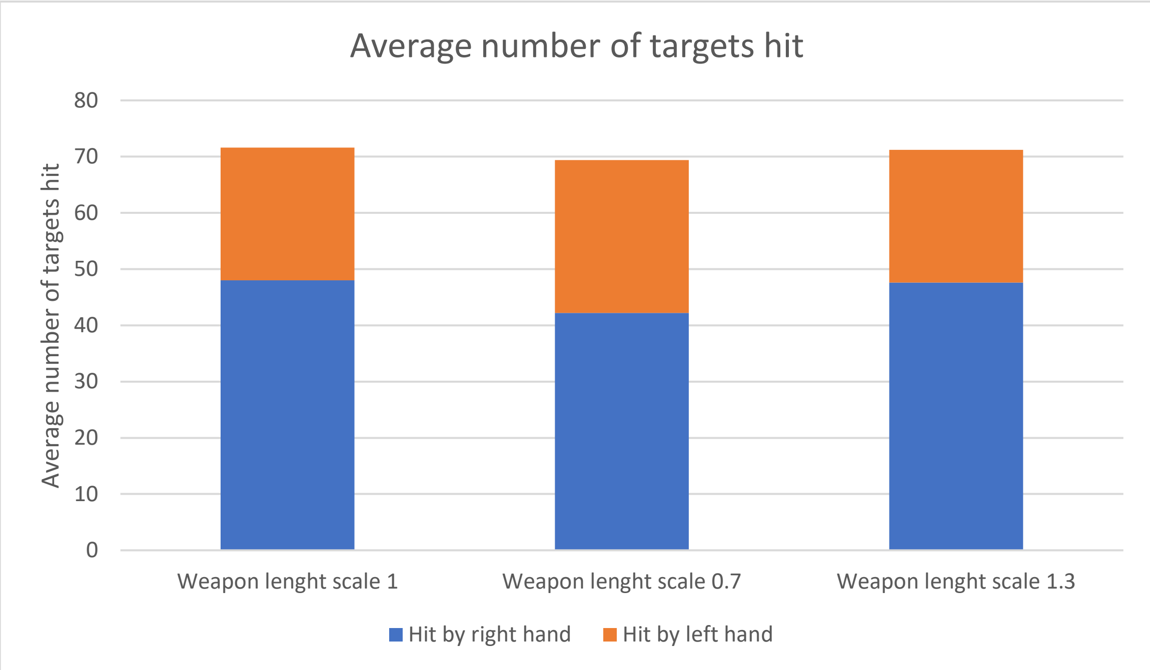

Figure 6.23 Average number of targets hit in the *Box Smash* game using different weapon scales.

Figure 6.24 shows the average hitting forces on targets in the group of scenarios where the weapon's length scale was evaluated. The average force for the weapon scale of 1 and for the weapon scale of 0.7 is the same, and it amounts to 3.9 for all targets and 4.2 for the destroyed targets. With a weapon scale of 1.3, the average force was 3.3 for all targets and 3.7 for the destroyed targets.

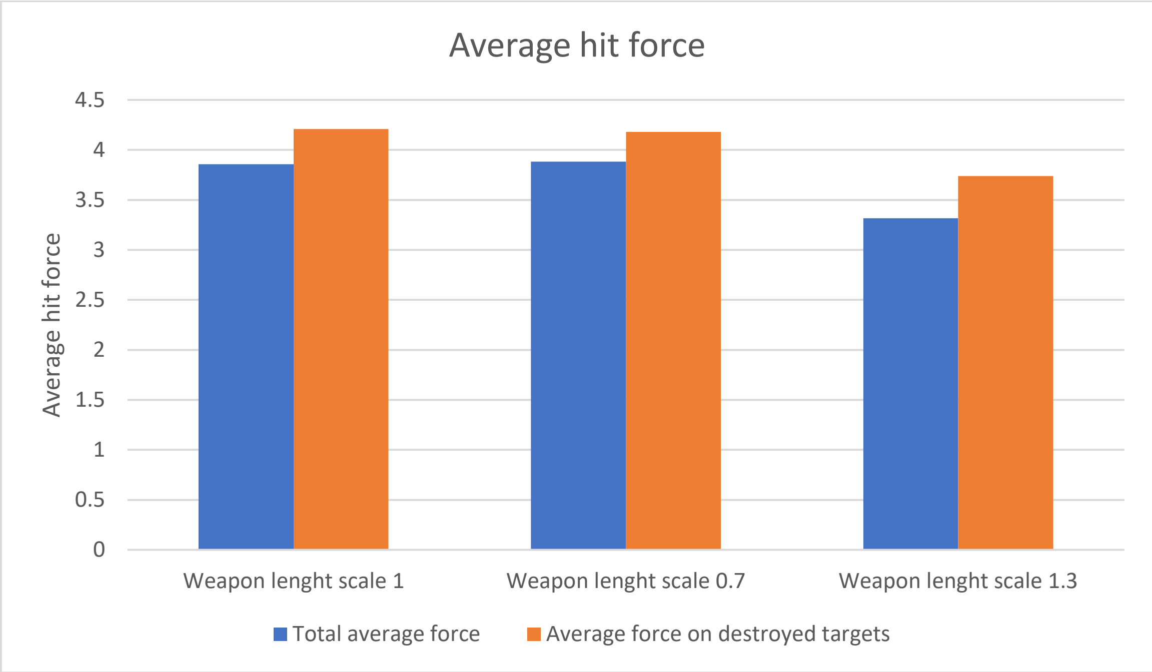

Figure 6.24 Average hit force in the *Box Smash* game using different weapon scales.

6.2.2. Subjective measures

The results of the post-experience questionnaire for the first group of scenarios which evaluated different target spawning angles are shown in Figure 6.25. The participants were asked to evaluate the total QoE for each scenario on a scale of 1 to 5, with 1 representing very low and 5 representing very high QoE. The highest average QoE was when the spawn angle was set to 90 degrees, and it amounted to 4.2 out of 5. The feeling of competence is also highest in this scenario compared to the other two scenarios in this group, and it averages at 4 out of 5. Moreover, this scenario was the least challenging with an average score of 2.8 out of 5, but the most entertaining with an average score of 4.2 out of 5. When the spawn angle was set to 180 degrees, the average QoE was evaluated as 3.8 out of 5, the average feeling of competence was 3.2, the average difficulty of the game was 3.8, and the average entertainment score was 4 out of 5. The average QoE was the lowest when the spawn angle was set to 360 degrees, only 2.6 out of 5. The feeling of competence was also significantly lower and averaged at 1.8, while the game was evaluated as challenging with an average score of 4.6 out of 5. The participants' enjoyment of the game was also the lowest in this group of scenarios, with an average score of 2.6 out of 5.

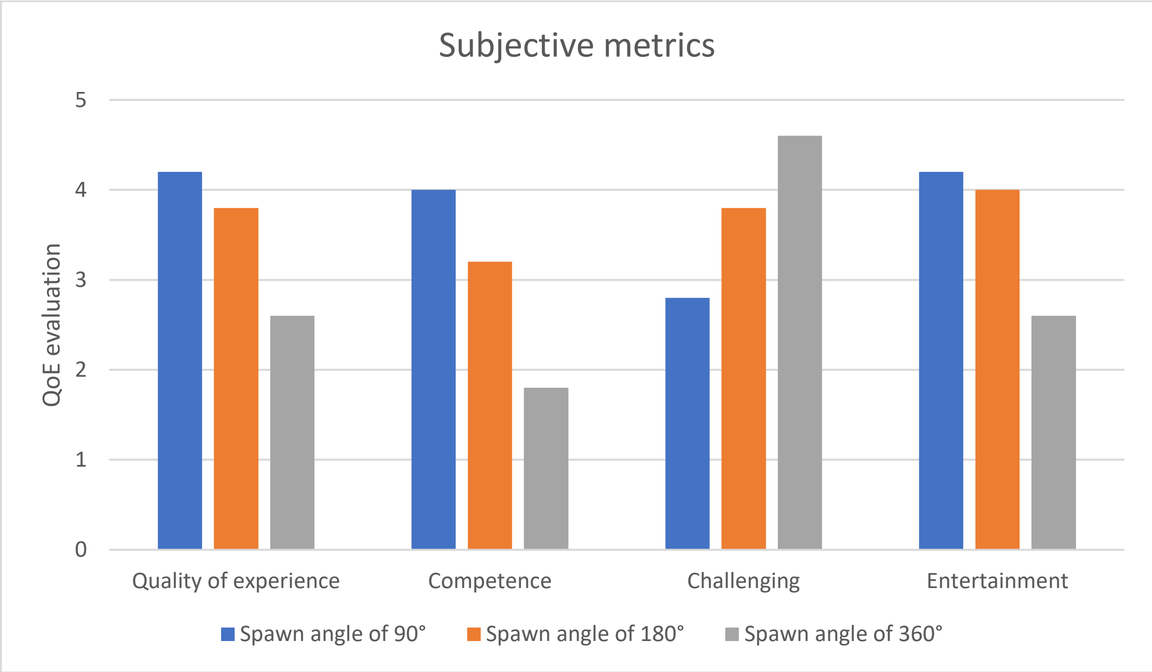

Figure 6.25 Evaluation of QoE (on a scale of 1: bad to 5: excellent), competence, challenge, and entertainment (on a scale of 1: very low to 5: very high) in the *Box Smash* game using different target spawning angles.

Figure 6.26 shows the simulation workload measures evaluation results for the group of scenarios in which different target spawn angles were evaluated. The mental demand measure was evaluated the lowest for the scenario in which the spawn angle was set to 90 degrees, and it averaged at 1.6 out of 5. When the spawn angle was set to 180 degrees, the average mental demand was 2.6 out of 5, and when the spawn angle was set to 360 degrees, the mental demand was the highest and it averaged at 3.4 out of 5. The average physical demand of 2.2 was the same in the scenario with the spawn angle of 90 degrees and in the scenario with the spawn angle of 180 degrees, however, it was slightly higher in the scenario with the spawn angle of 360 degrees, with a score averaging 2.6 out of 5. The task control difficulty score for the spawn angle of 90 degrees averaged at 2.2, for the spawn angle of 180 degrees it averaged at 2.8, and for the spawn angle of 360 degrees it averaged at 3.8.

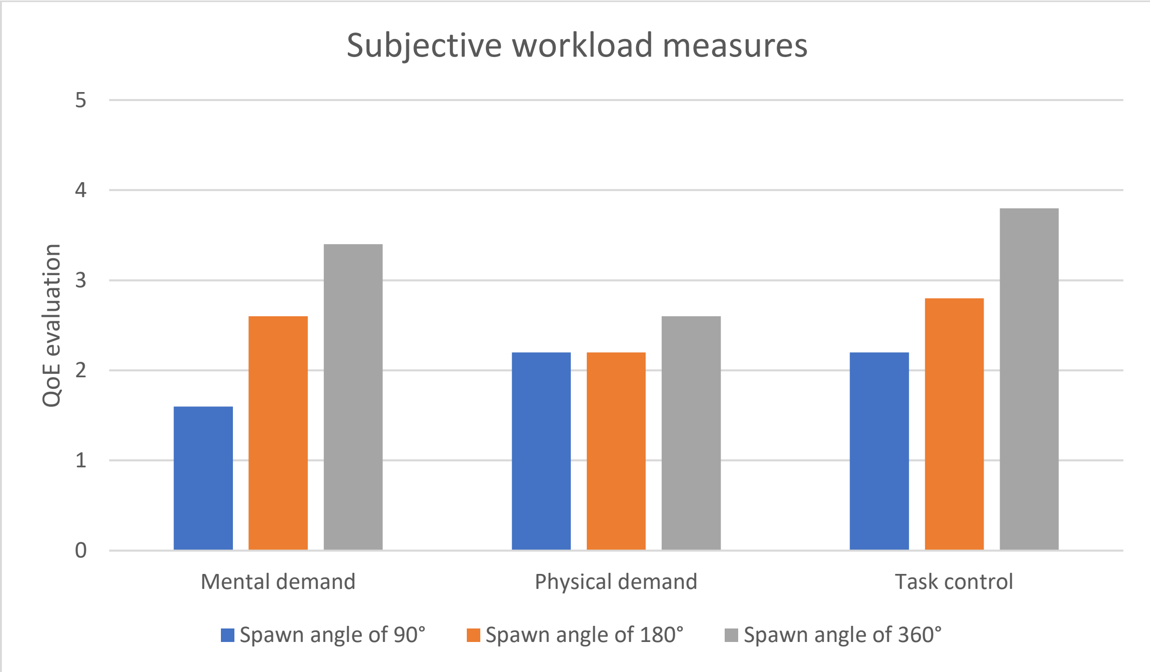

Figure 6.26 Evaluation of simulation workload in the *Box Smash* game using different target spawning angles (on a scale of 1: very low to 5: very high).

For the scenario with the spawn angle of 90 degrees and the scenario with the spawn angle of 180 degrees, all participants were willing to continue playing the game under the scenario's conditions, while for the scenario with the spawn angle of 360 degrees, only one of the five participants was willing to continue gameplay. Participants were asked to choose the best scenario in this group, and 3 out of 5 participants chose the scenario where the spawn angle was set to 90 degrees because the targets were in the field of view and the participants felt more competent. The remaining two participants chose the scenario in which the spawn angle was set to 180 degrees as the best because it was both challenging and achievable. When asked to select the least favorite scenario in this group, participants unanimously selected the scenario in which the spawn angle was 360 degrees because they could not see all the targets and the gameplay was too confusing and challenging.

The second group of scenarios assessed different minimum forces required to destroy a target. When the minimum force necessary was set to 0, the average QoE was 4 out of 5, the average feeling of competence was 4.2 out of 5, the challenging aspect of the game was scored at 2.2 out of 5, and the entertainment factor was 4.2 out of 5. When the minimum force required was set to 2, the average QoE was the highest in this group with a score of 4.4 out of 5. The average feeling of competence was 3.8, the average perceived game difficulty was 3.2, and the entertainment value of the game was scored at 4.4 out of 5. The scenario with the minimum force set to 6 has the lowest QoE score, averaging at 2.8 out of 5. The average feeling of competence of 1.8 in this scenario is also significantly lower compared to the other two scenarios in this group. Similarly, the feeling of challenge in this scenario is significantly higher and it averages at 4.6 out of 5. On the other hand, the entertainment factor in this scenario is lower compared to the other two scenarios, and it averages at 3.2 out of 5. The results are shown in figure 6.27.

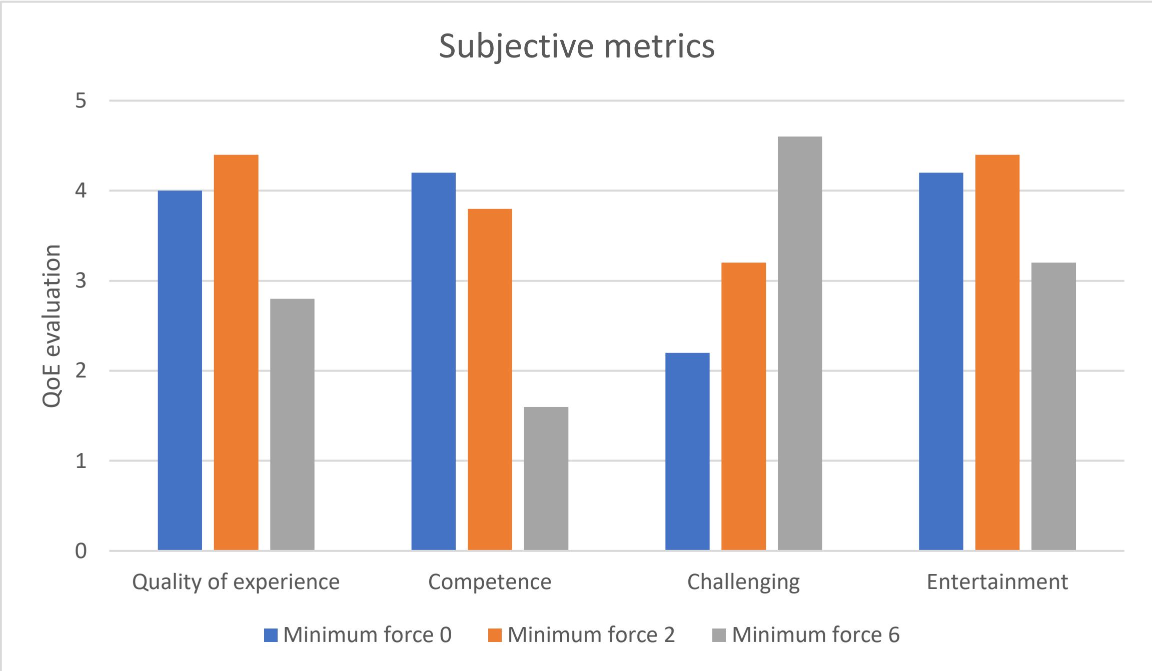

Figure 6.27 Evaluation of QoE (on a scale of 1: bad to 5: excellent), competence, challenge, and entertainment (on a scale of 1: very low to 5: very high) in the *Box Smasher* game using different minimum required forces.

Figure 6.28 shows the simulation workload measures evaluation results for the group of scenarios in which different minimum required forces were evaluated. The mental demand measure was evaluated the lowest for the scenario in which the minimum required force was zero, and it averaged at 1.2. When the force required was set to 2, the average mental demand was 1.4, and when the force required was set to 6, the average mental demand was 1.8. In the scenario where the minimum force was 0, participants expressed an average physical demand of 1.6, and when the force required was set to 2, the average physical demand was 2 out of 5. The average physical demand of 3.2, scored in the scenario with the minimum required force of 6, is the highest in this group of scenarios. The task control difficulty averaged at 2 out of 5 when the minimum force was set to 0, 2.4 when the minimum force was set to 2, and 4 out of 5 when the force was set to 6, which is a significant increase compared to the first two scenarios.

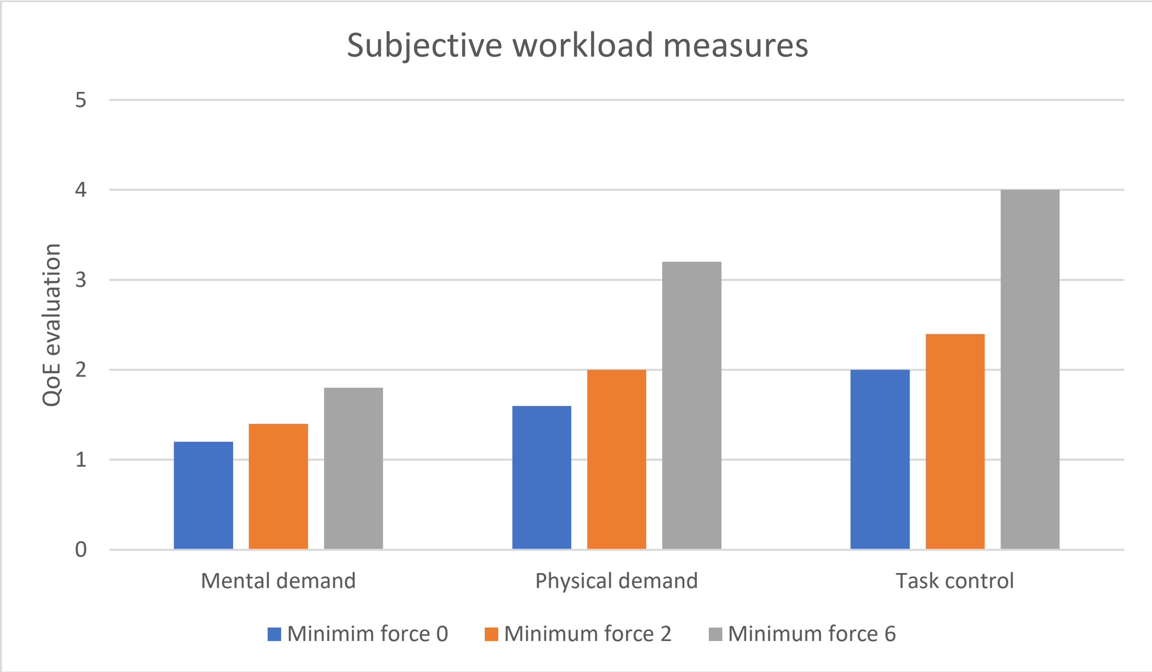

Figure 6.28 Evaluation of simulation workload in the *Box Smash* game using different minimum required forces (on a scale of 1: very low to 5: very high).

For all the scenarios evaluated in this group, participants were willing to continue playing the game under the same conditions. When asked to select the best scenario in this group, 3 out of 5 participants selected the scenario in which the minimum force required was set to 0 because it was more fun and not too challenging. The remaining two participants selected the scenario in which the minimum force required was set to 2 as the best because it was not too easy and not too hard. When asked to select the least favorite scenario in this group, participants unanimously selected the scenario in which the minimum force was set to 6 because it was frustrating and too challenging.

The last group of scenarios for the *Box Smash* game evaluated different weapon length scales. When the scale was set to 1, the average QoE was 4 out of 5, while in the scenario where the length scale was 0.7 and the scenario where the length scale was 1.3, the perceived QoE was the same and it averaged at 4.2 out of 5. The average feeling of competence is the same for the weapons scale of 1 and the weapons scale of 0.7, and it amounts to 4 out of 5. When the weapon length scale was set to 1.3, the average feeling of competence was 4.2 out of 5. When the weapon scale was set to 1, an average score of 2.6 was given for how challenging the game was, when the scale was set to 0.7 an average score of 3.4 was given, and when the weapon scale was set to 1.3 the score averaged at 2.8. The subjective entertainment of the game is the highest when the scale was set to 1, averaging at 4.4, and it averaged at 4.2 for both the scenario with the weapon scale of 0.7 and the scenario with the weapon scale of 1.3. The average scores for QoE, feeling of competence, game difficulty, and entertainment value are shown in Figure 6.29.

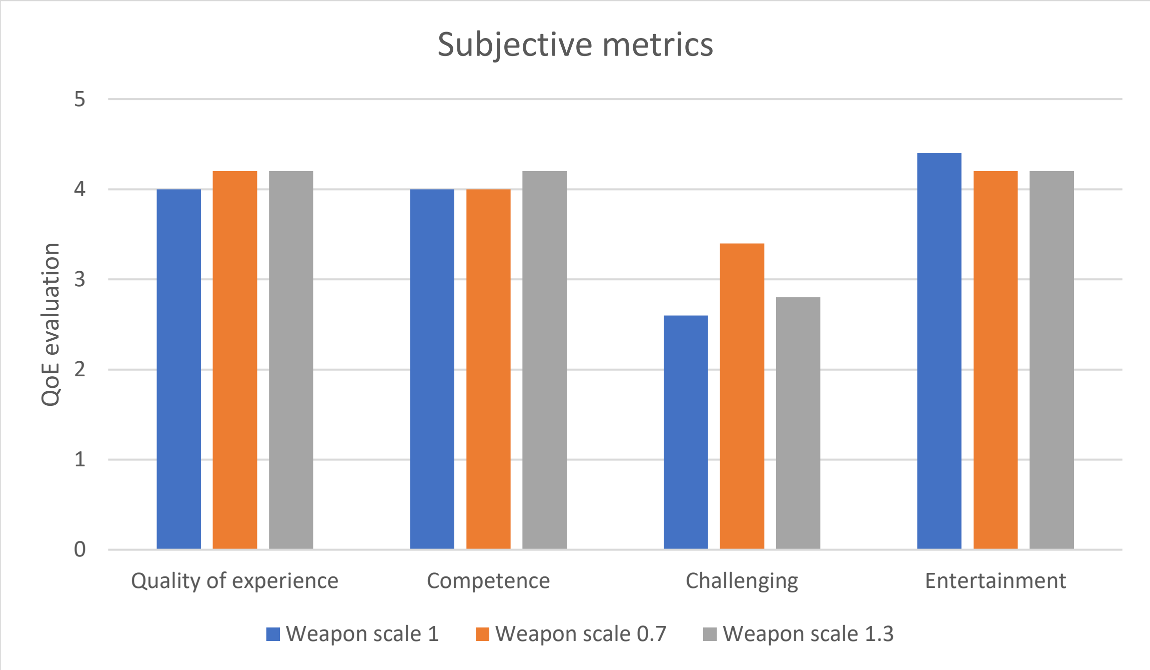

Figure 6.29 Evaluation of QoE (on a scale of 1: bad to 5: excellent), competence, challenge, and entertainment (on a scale of 1: very low to 5: very high) in the *Box Smash* game using different weapon scales.

Figure 6.30 shows the simulation workload measures evaluation results for the group of scenarios in which different weapon length scales were evaluated. In the scenario with the scale set to 1, the average mental demand was 1.4, the average physical demand was 1.8, and the average task control difficulty was 2.2. When the weapon length scale was set to 0.7, the average mental demand was 1.8, the average physical demand was 1.6, and the average task control difficulty was 2.4. In the scenario with the scale set to 1.3, the average mental demand was 1.6, the average physical demand was 1.8, and the average task control difficulty was 2.6.

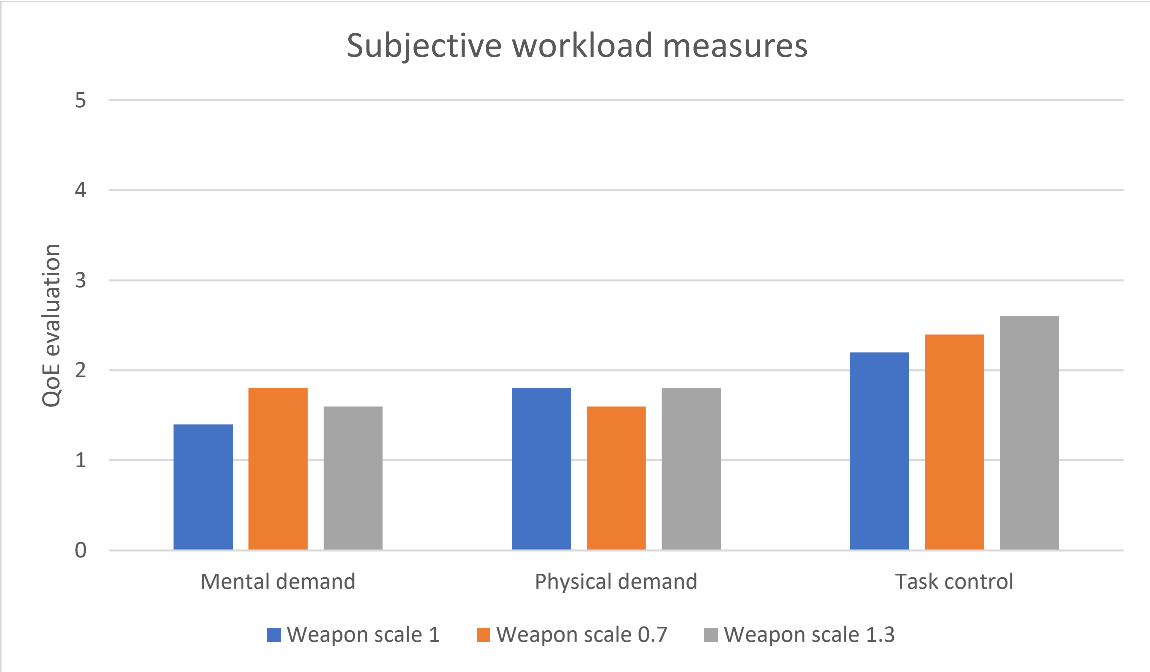

Figure 6.30 Evaluation of simulation workload in the *Box Smash* game using different weapon scales (on a scale of 1: very low to 5: very high).

Participants were asked to select their favorite scenario in this group, and 3 out of 5 participants selected the scenario where the weapon length scale was set to 1 because they found that length to be the optimal. One participant selected the scenario where the scale was set to 0.7 as their favorite because it was the most fun, while one participant selected the scenario where the scale was set to 1.3 as their favorite.

When asked to select the least favorite scenario in this group, 3 out of 5 participants selected the scenario where the scale was set to 0.7 because they felt less efficient in destroying the targets. The remaining 2 participants selected the scenario where the scale was set to 1.3 because the weapons were hitting the ground. On average, most participants did not find a significant difference between the scenarios in this group.

Subjective discomfort measures of experienced muscle pain and physical discomfort collected in this study were not analyzed in this thesis. However, this data is available and can be analyzed in future work.

6.2.3. Analysis

Figure 6.31 shows the comparison of objective accuracy measured as a ratio of the number of destroyed targets and the total number of targets, the subjective feeling of competence, and the subjective evaluation of the game's entertainment for the group of scenarios where different spawning angles were evaluated. The average objective accuracy corresponds to the average perceived competency. When the spawn angle was set to 90 degrees, the accuracy of hitting targets averaged at 63.3% and the average feeling of competence had an average score of 4 out of 5. In the scenario where the spawn angle was 180 degrees, the average accuracy was 54.1% and the average experienced feeling of competence was 3.2. With the spawn angle set to 360 degrees, the average accuracy is the lowest at 33.3% and the feeling of competence averaged at 1.8 out of 5. The experienced entertainment factor is similar between the scenarios where the spawn angle is 90 degrees (4.2) and the scenario where the spawn angle is 180 degrees (4). However, it is significantly lower (2.6) when the spawn angle was set to 360 degrees. Additionally, with the spawn angle set to 360 degrees, the average QoE was the lowest (2.6) compared to the scenario with the spawn angle of 90 degrees (4.2) and the scenario with the spawn angle of 180 degrees (3.8). On average, the ratio of hits done by the right hand is higher when the spawn angle is lower, with the percentage of hits done by the right hand amounting to 77.2% when the spawn angle was 90 degrees compared to 66.5% when the spawn angle was 180 degrees and 68.2% when the spawn angle was 360 degrees. Because the same number of target spawning cannons were created in each scenario, there was more lateral distance between targets, which could have resulted in participants using their non-dominant hand more frequently.

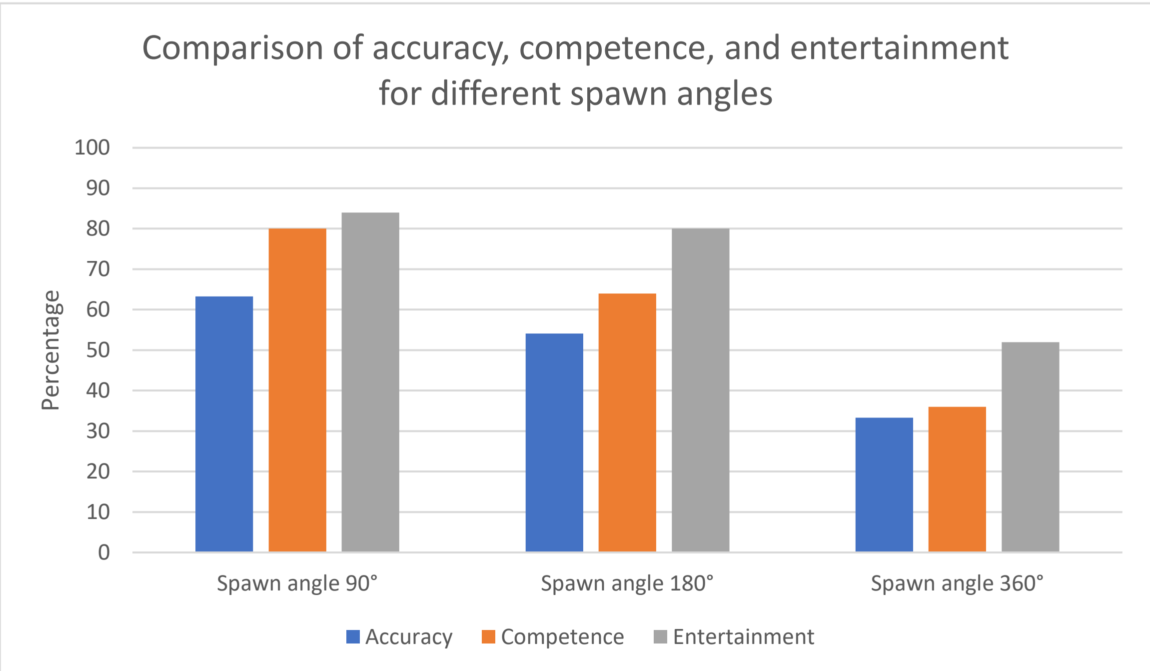

Figure 6.31 Comparison of accuracy, competence, and entertainment in the *Box Smash* game using different target spawning angles.

The average physical demand is similar for all three scenarios, amounting to 2.2 for the spawn angle of 90 degrees and the spawn angle of 180 degrees, while for the spawn angle of 360 degrees it averaged at 2.6 out of 5. On the other hand, the average subjective mental demand and task control difficulty increased as the spawn angle increased and were at their highest when the spawn angle was 360 degrees, amounting to 3.6 for the mental demand and 3.8 for the task control difficulty. Consequently, all the participants found the scenario with the spawn angle of 360 degrees to be the worst in this group. Being unable to see all the targets and the lower chance of destroying multiple targets with one sweep of the weapon caused by greater lateral distances between targets contributed to the lowest QoE scores for this scenario. Meanwhile, 40% of participants chose the spawn angle of 180 degrees as their favorite in this group, even though it was more challenging and required higher average mental demand compared to the scenario where the spawn angle was 90 degrees, preferring more challenge but still feeling that hitting the targets was achievable. Still, most participants preferred it when the spawn angle was 90 degrees. Since the timeframe in which the target needs to be destroyed is limited by the time it takes for the target to fall to the ground once it has been ejected from the target spawning cannon, successfully destroying the targets is a challenge of its own. Therefore, introducing the additional challenge of finding all the targets in time greatly reduces hitting accuracy and causes frustration for most participants.

In the group of scenarios where different minimum forces required to destroy a target were evaluated, there is a slight difference between the scenario in which the minimum required force is 0 and the scenario in which the minimum required force is 2. This could be attributed to the fact that most participants did not notice that no force was required to destroy the targets in the scenario with the minimum force set to 0, resulting in an average hitting force of 2.8. In comparison, the average hitting force when the minimum force was set to 2 amounted to 3.8. Moreover, there is a difference in accuracies in these two scenarios. The average accuracy in the scenario where the minimum force required was 0 amounts to 76.8%, while in the scenario where the minimum force required was to 2, the accuracy was 69% with 28.1% of all missed targets were missed because of insufficient force. The average QoE score was higher when the force was set to 2 (4.4) compared to the average QoE when the force was set to 0 (4). Even though the average feeling of competence was lower in the scenario with the minimum force of 2 (3.8) compared to the scenario where the minimum force was 0 (4.2), the average entertainment factor was higher when the minimum force required was set to 2 (4.4) than in the scenario where the minimum force required was set to

0 (4.2). This could be attributed to the game being less challenging when no force was required (2.2) compared to when the minimum force required was 2 (3.2). Additionally, some participants enjoyed catching targets between swords instead of just destroying them, which was impossible to do when the minimum force was 0. There is no significant difference in average scores for subjective mental demand, physical demand, and task control difficulty between these two scenarios.

On the other hand, all participants selected the scenario where the minimum force required was 6 as the least favorite in this group of scenarios. The average force with which the targets were hit is 5.1, which is lower than the minimum required force, resulting in an average accuracy of only 32.8%, because 59.8% of targets were hit with insufficient force. This caused significantly lower feeling of competence which averaged at 1.8 out of 5. The subjective physical demand was also the highest in this scenario (3.2) compared to the scenario where the minimum required force was 0 (1.6) and the scenario where the minimum required force was 2 (2). Moreover, the task control difficulty was significantly higher when the minimum force was set to 6 (4). When the minimum force required was 6, participants used their dominant hand more often (75.5%), while the ratio of using the dominant right hand is similar when the minimum required force was 0 (65.5%) and when the minimum required force was 2 (64%). This means that, on average, participants use their dominant hand to destroy the targets slightly more often than their non-dominant hand, but when the task becomes more difficult and requires more force, participants use their dominant hand with a higher frequency. The comparison of average accuracies, competence, and entertainment in this group is shown in Figure 6.32.

The last group of scenarios evaluated different weapon length scales, and Figure 6.33 shows the comparison between objective accuracy, feeling of competence, and feeling of entertainment for each scenario in this group. The average objective accuracies, the subjective QoE, feeling of competence, and the game's entertainment are similar for all three scenarios. The subjective feeling of challenge, however, is slightly higher in the scenario where the scale was set to 0.7 (3.4) compared to the scenario with the weapon scale of 1 (2.6) and the scenario with the weapon scale of 1.3 (2.8). Additionally, 60% of participants selected this scenario as their least favorite because it was more difficult to hit the targets and participants felt they had to move their hands too much. In most cases, participants did not notice any difference between these three scenarios. The workload measures are relatively similar for all three scenarios in this testing group.

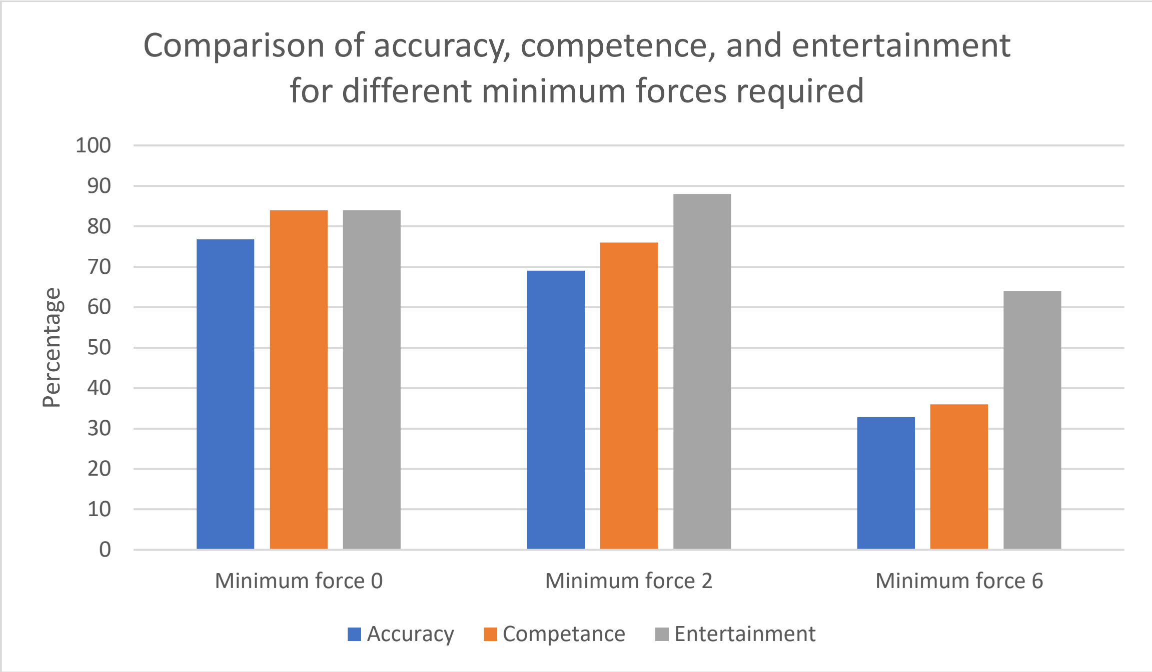

Figure 6.32 Comparison of accuracy, competence, and entertainment in the *Box Smash* game using different minimum required forces.

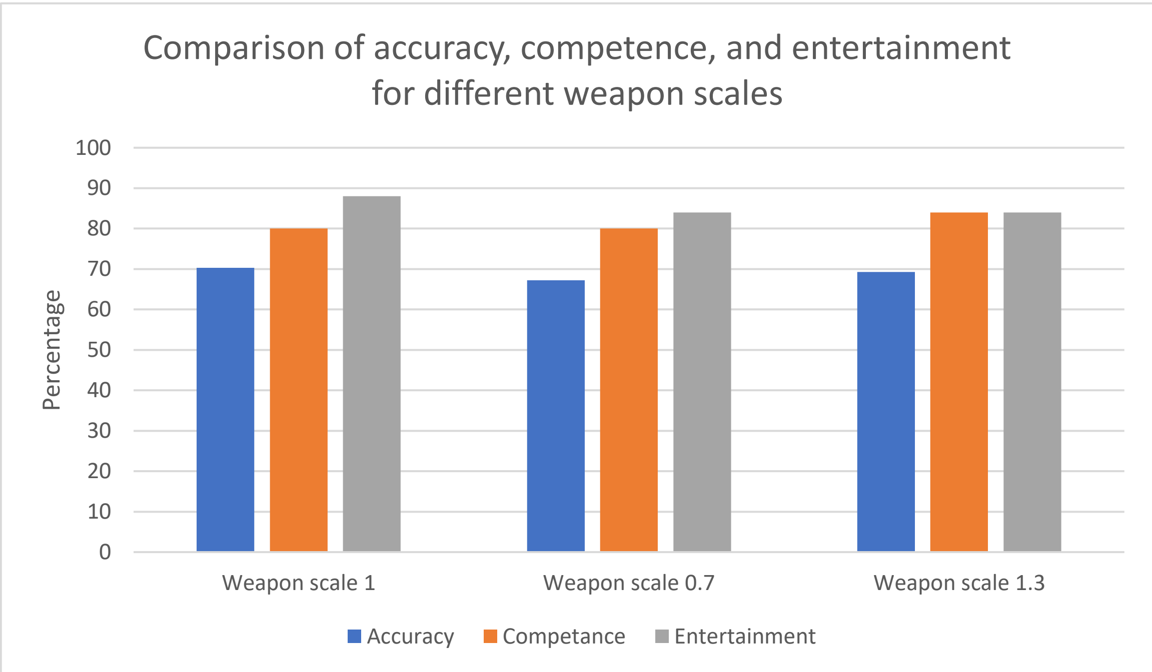

Figure 6.33 Comparison of accuracy, competence, and entertainment in the *Box Smash* game using different weapon scales.

6.3. Pick and Place

6.3.1. Objective measures

The first group of scenarios in the *Pick-And-Place* game compared puzzle scales. In the testing scenario where the scale was set to 0.1, the average number of pieces placed was 5.6 and the average duration of the round was 81.7 seconds, with only one participant completing the puzzle before the end of the round (48.7 seconds). When the puzzle scale was set to 0.4 an average of 6.2 out of 7 pieces were placed in the solution, and the average duration was 72.8 seconds, with 2 out of 5 participants not completing the puzzle before the end of the game round. In the scenario where the scale of the puzzle was set to 0.7, an average of 4.2 puzzle pieces were placed. The average duration of the game round in this scenario was 88.3 seconds, with only one participant finishing the puzzle before the end of the round (81.7 seconds). The average number of pieces placed in this group of scenarios are shown in Figure 6.34.

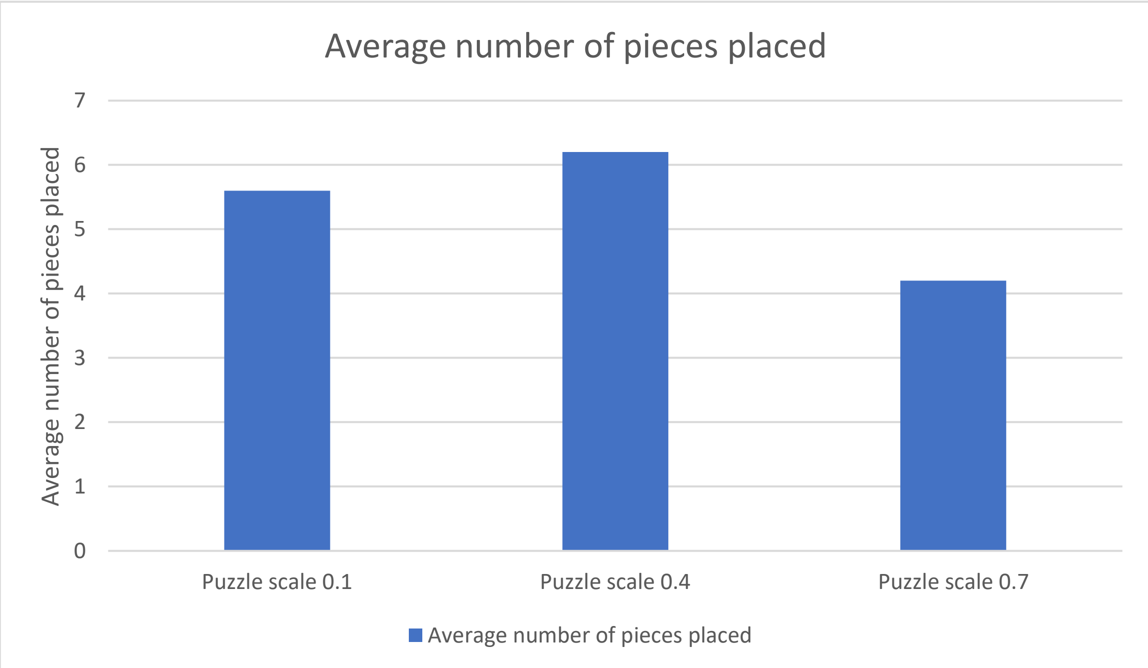

Figure 6.34 Average number of puzzle pieces placed in the *Pick-And-Place* game using different puzzle scales.

The second group of scenarios compared different collider scales. In the testing scenario where the scale was set to 0.2, the average number of pieces placed was 4.2 and the average duration of the round was 87.9 seconds, with only one participant completing the puzzle before the end of the round (79.5 seconds). When the collider scale was set to 0.5, all participants managed to place all the pieces in the solution before the end of the game round, and the average duration was 60.7 seconds. In the scenario where the scale of the colliders was set to 1, all participants finished the puzzle before the end of the round. The average duration of the game round in this scenario was 53 seconds. The average number of pieces placed in this group of scenarios are shown in Figure 6.35.

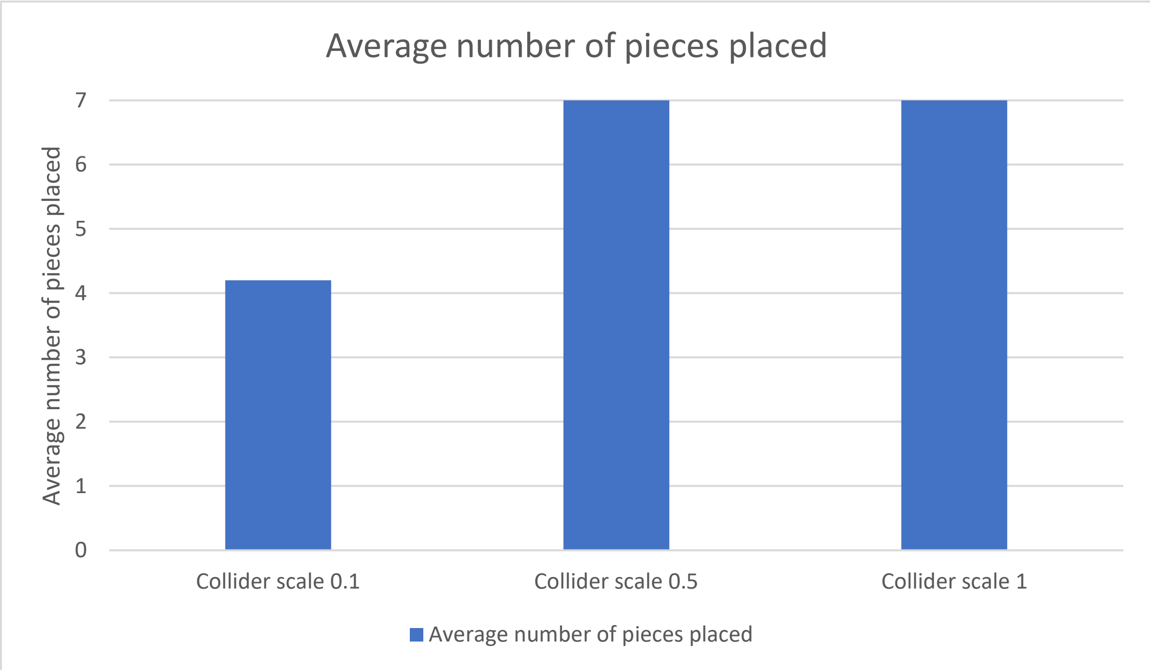

Figure 6.35 Average number of puzzle pieces placed in the *Pick-And-Place* game using different collider scales.

In the testing scenario where remote grab functionality was not enabled, all participants finished the puzzle before the end of the round, and the average duration of the round was 56.6 seconds). When remote grab was enabled, an average of 5.6 out of 7 pieces were placed in the solution, and the average duration was 73.2 seconds, with 2 out of 5 participants not completing the puzzle before the end of the game round. The average number of pieces placed in this group of scenarios is shown in Figure 6.36.

Figure 6.37 shows the average number of pieces placed in the group of scenarios where different puzzle piece scale offsets were evaluated. In the testing scenario where the scale offset was set to 0.8, all participants completed the puzzle before the end of the round, and the average duration of the round was 61.4 seconds. When the scale offset was set to 0.9, an average of 6.6 out of 7 pieces were placed in the solution, and the average duration was 69.3 seconds, with 2 out of 5 participants not completing the puzzle before the end of the game round. In the scenario where the scale offset was set to 1, an average of 6 puzzle pieces were placed. The average duration of the game round in this scenario was 78.9 seconds, with 2 out of 5 participants not completing the puzzle before the end of the game round.

Figure 6.36 Average number of puzzle pieces placed in the *Pick-And-Place* game using different grab types.

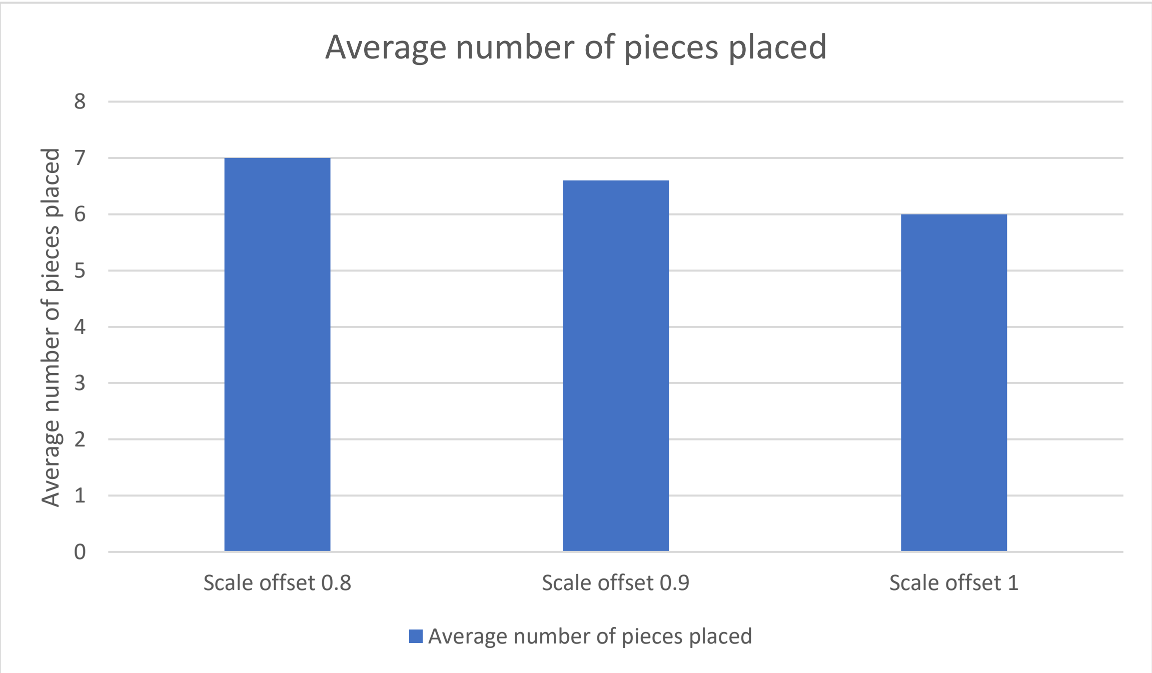

Figure 6.37 Average number of puzzle pieces placed in the *Pick-And-Place* game using different scale offsets.

6.3.2. Subjective measures

The results of the post-experience questionnaire for the first group of scenarios, which evaluated different puzzle scales are shown in Figure 6.38. The participants were asked to evaluate the total QoE for each scenario on a scale of 1 to 5, with 1 representing very low and 5 representing very high QoE. The average QoE amounted to 4.8 both when the puzzle scale was 0.1 and when the puzzle scale was 0.4. In the scenario where the puzzle scale was 0.7, the average QoE was 4 out of 5. The feeling of competence when the puzzle scale was 0.1 averaged at 3.8, when the puzzle scale was 0.4 it averaged at 4.2, and when the puzzle scale was 0.7 it averaged at 2.6. When evaluating how challenging the game was, an average score of 2.8 was given when the puzzle scale was 0.1, an average score of 3 was given when the puzzle scale was 0.4, and an average score of 3.8 was given when the puzzle scale was 0.7. The game's entertainment score averaged at 4.6 for the puzzle scale of 0.1, 4.4 for the puzzle scale of 0.4, and 4.2 for the puzzle scale of 0.7.

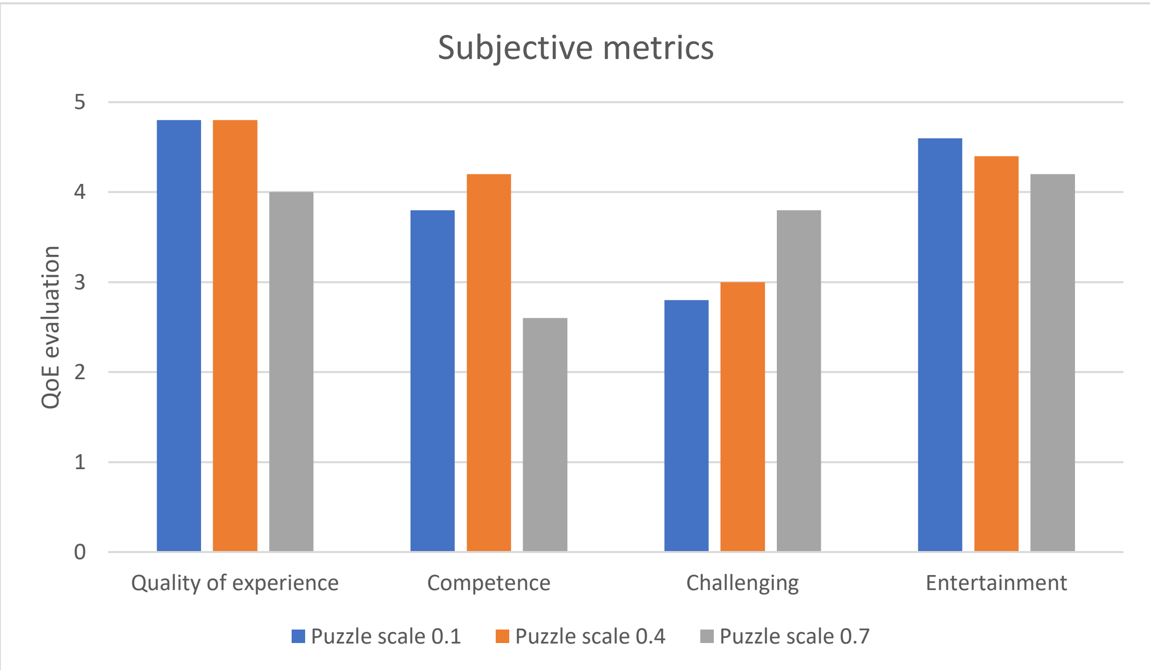

Figure 6.38 Evaluation of QoE (on a scale of 1: bad to 5: excellent), competence, challenge, and entertainment (on a scale of 1: very low to 5: very high) in the *Pick-And-Place* game using different puzzle scales.

Figure 6.39 shows the simulation workload measures evaluation results for the group of scenarios in which different puzzle scales were evaluated. The mental demand was evaluated at the same (2.4) for the scenario where the puzzle scale was 0.1 and the scenario where the puzzle scale was 0.7. When the puzzle scale was 0.4, the average mental demand was 2.2

out of 5. The average physical demand was the lowest when the puzzle scale was 0.1 (1.2) compared to when the puzzle scale was 0.4 (1.4) and when the puzzle scale was 0.7 (2). The task control difficulty averages at 1.6 for the puzzle scale of 0.1, 2.4 for the puzzle scale of 0.4, and 3.6 for the puzzle scale of 0.7.

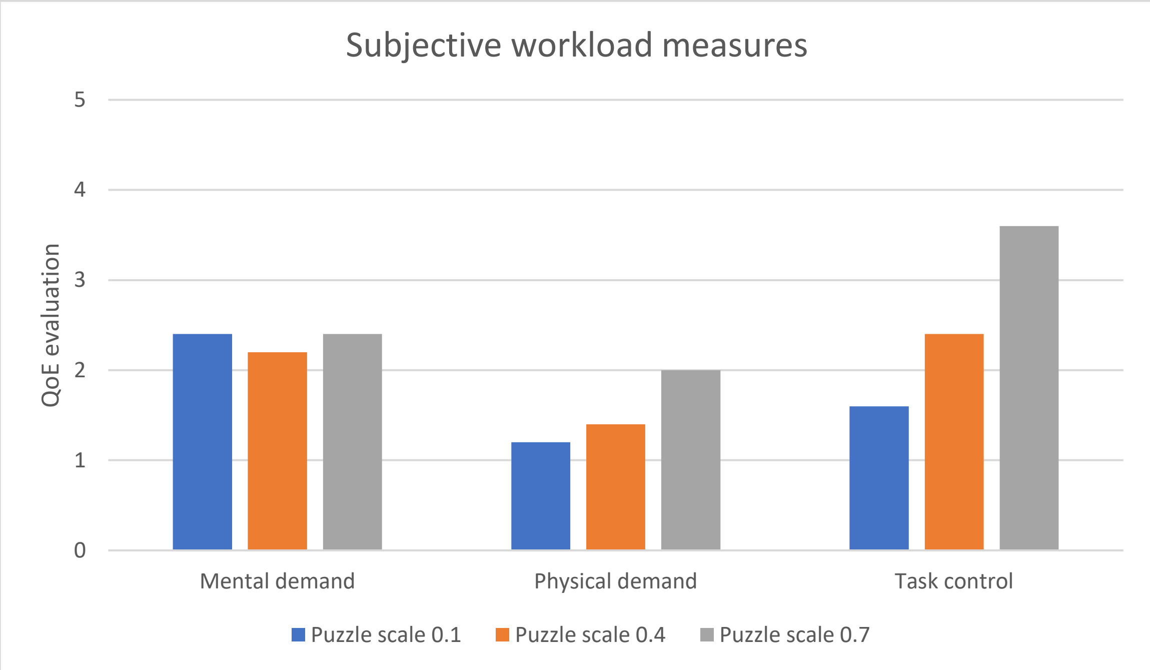

Figure 6.39 Evaluation of simulation workload in the *Pick-And-Place* game using different puzzle scales (on a scale of 1: very low to 5: very high).

When asked to select the best scenario in this group, 4 out of 5 participants selected the scenario where the puzzle scale was set to 0.1 because they felt the puzzle pieces were easier to handle. The remaining participant selected the scenario where the puzzle scale was set to 0.4 as the best because the puzzle pieces were the easiest to manage. All participants selected the scenario where the puzzle scale was set to 0.7 as the least favorite in this group of scenarios because it was harder to control the puzzle pieces.

The results of the post-experience questionnaire for the second group of scenarios, which evaluated different collider scales, are shown in Figure 6.40. When the collider scale was set to 0.2, the average QoE was 3.4, the average feeling of competence was 2.4, the average perceived difficulty of the game was 3.6, and the average perceived entertainment was 3.8. In the scenario where the collider scale was 0.5, the evaluated QoE averaged at 4.8, the self-assessed competence averaged at 4.6, the evaluation of how challenging the game was averaged at 2.6, and the game’s entertainment factor averaged at 4.6. With the collider scale set to 1, the participants evaluated the QoE with an average score of 4.6 out of 5. The average feeling of competence in this scenario was 4.8, the average evaluated difficulty was 2, and the average entertainment factor was 4.4.

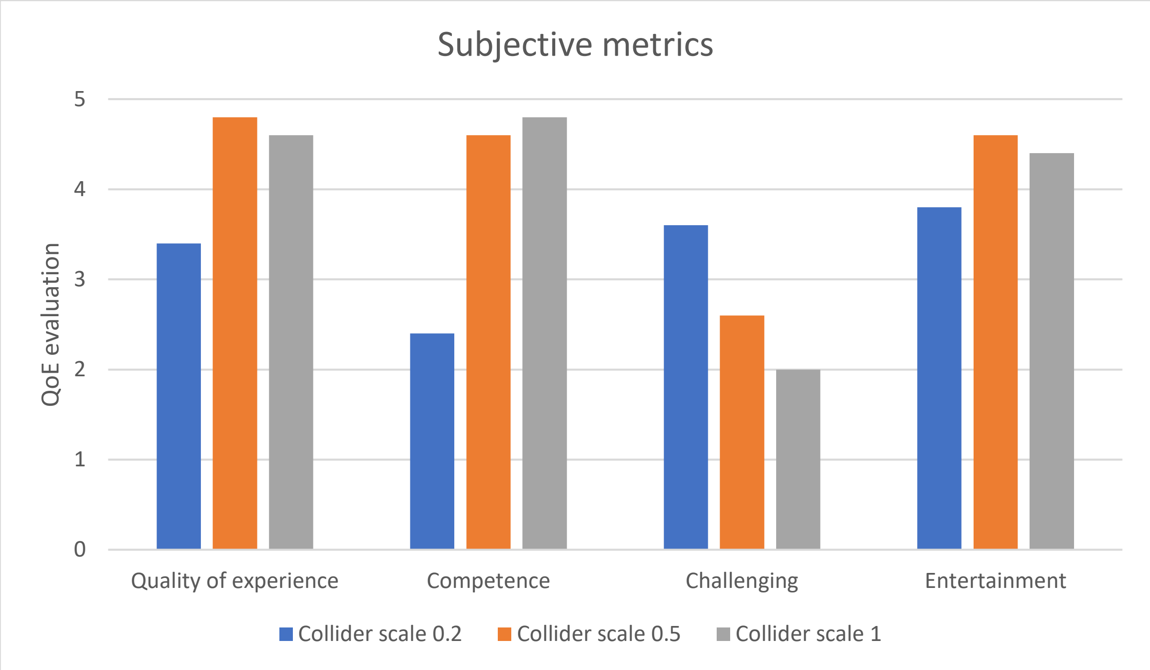

Figure 6.40 Evaluation of QoE (on a scale of 1: bad to 5: excellent), competence, challenge, and entertainment (on a scale of 1: very low to 5: very high) in the *Pick-And-Place* game using different collider scales.

Figure 6.41 shows the simulation workload measures evaluation results for the group of scenarios in which different shoot forces were evaluated. In the scenario where the collider scale was set to 0.2, the average mental demand was 2.6 and the average physical demand was 1.2. When the collider scale was 0.5, the average evaluated mental demand was 1.6, and the average evaluated physical demand was 1. With the collider scale set to 1, the average mental demand was 1.4 and the average physical demand was 1. The evaluated task control difficulty averaged at 3.4 when the collider scale was 0.2, 1.4 when the collider scale was 0.5, and 1.4 when the collider scale was 1.

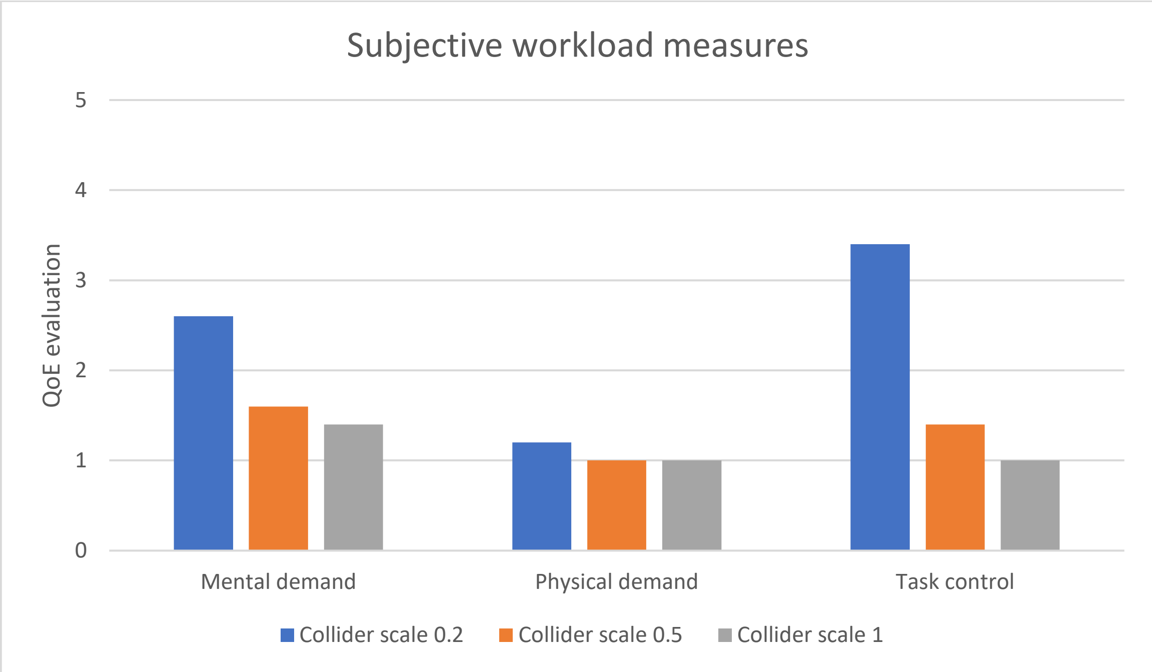

Figure 6.41 Evaluation of simulation workload in the *Pick-And-Place* game using different collider scales (on a scale of 1: very low to 5: very high).

When asked to select the best scenario in this group, 3 out of 5 participants selected the scenario where the collider scale was 1 because it was the easiest to complete. One participant selected the scenario where the collider scale was set to 0.5 because it presented a challenge without being too difficult, and one participant selected the scenario where the collider scale was set to 0.2 because it was the most challenging and therefore the most entertaining. When selecting the least favorite scenario in this group, 4 out of 5 participants selected the scenario where the collider scale was set to 0.2 because it was too challenging, and the remaining participant chose the scenario where the collider scale was set to 1 as the least favorite because it was too easy and not challenging enough.

Figure 6.42 shows the results of the post-experience questionnaire for the group of scenarios in which enabling remote grab was evaluated. When remote grab was disabled, the average QoE was 4.6, the average feeling of competence was 4, the average feeling of challenge was 2.8, and the average feeling of entertainment was 4.4. On the other hand, when remote grab was enabled, the average QoE was 4.8, the average feeling of competence was 3.6, the average feeling of challenge was 3.4, and the average feeling of entertainment was 4.6.

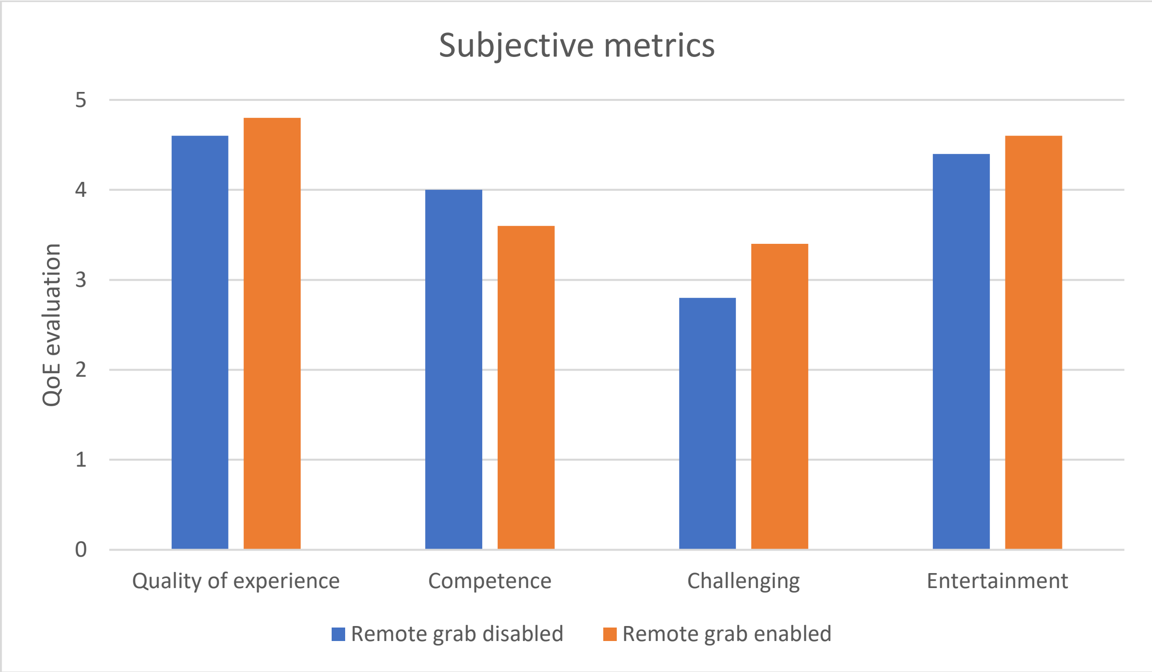

Figure 6.42 Evaluation of QoE (on a scale of 1: bad to 5: excellent), competence, challenge, and entertainment (on a scale of 1: very low to 5: very high) in the *Pick-And-Place* game using different grab types.

Figure 6.43 shows the simulation workload measures evaluation results for the group of scenarios in which enabling remote grab was evaluated. In the scenario where the remote grab was disabled, the average evaluated mental demand was 1.6, the average physical demand was 1, and the evaluated task control difficulty was 1.6. When remote grab was enabled, the average mental demand was 2, the average physical demand was 1.4, and the average task control difficulty was 2.2. When asked to select their favorite scenario in this group, participants unanimously selected the scenario where remote grab was enabled because it was more entertaining and unlike anything they had experienced in the real world.

The results of the post-experience questionnaire for the last group of scenarios that evaluated different puzzle scale offsets are shown in Figure 6.44. When the scale offset was set to 0.8, the average QoE was 4.6, the average feeling of competence was 4.2, the average feeling of challenge was 2.6, and the average entertainment score was 4.4. In the scenario where the puzzle piece scale offset was set to 0.9, the QoE averaged at 4.4, the average score for self-assessed competence was 4, the average score for how challenging the game was 3.2, and the average entertainment score was 4. With the scale offset set to 1, the average QoE was 4, the feeling of competence averaged at 3.2, the game's difficulty was assessed with an average score of 4.2, and the average assessment of the entertainment factor of the game amounted to 3.8.

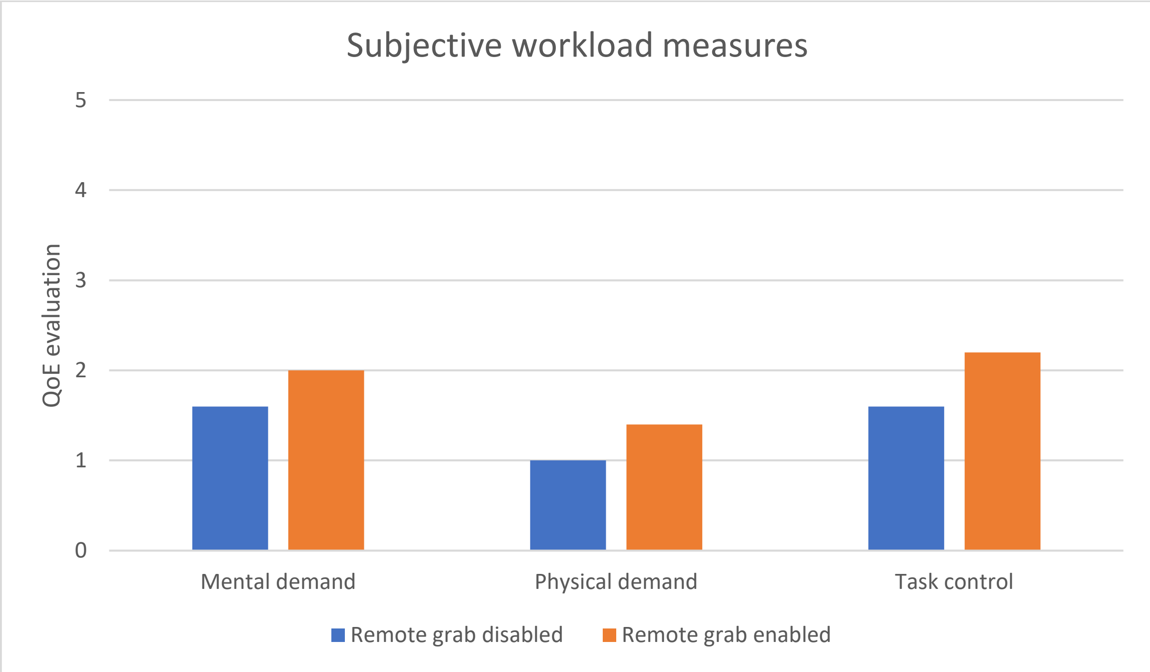

Figure 6.43 Evaluation of simulation workload in the *Pick-And-Place* game using different grab typed (on a scale of 1: very low to 5: very high).

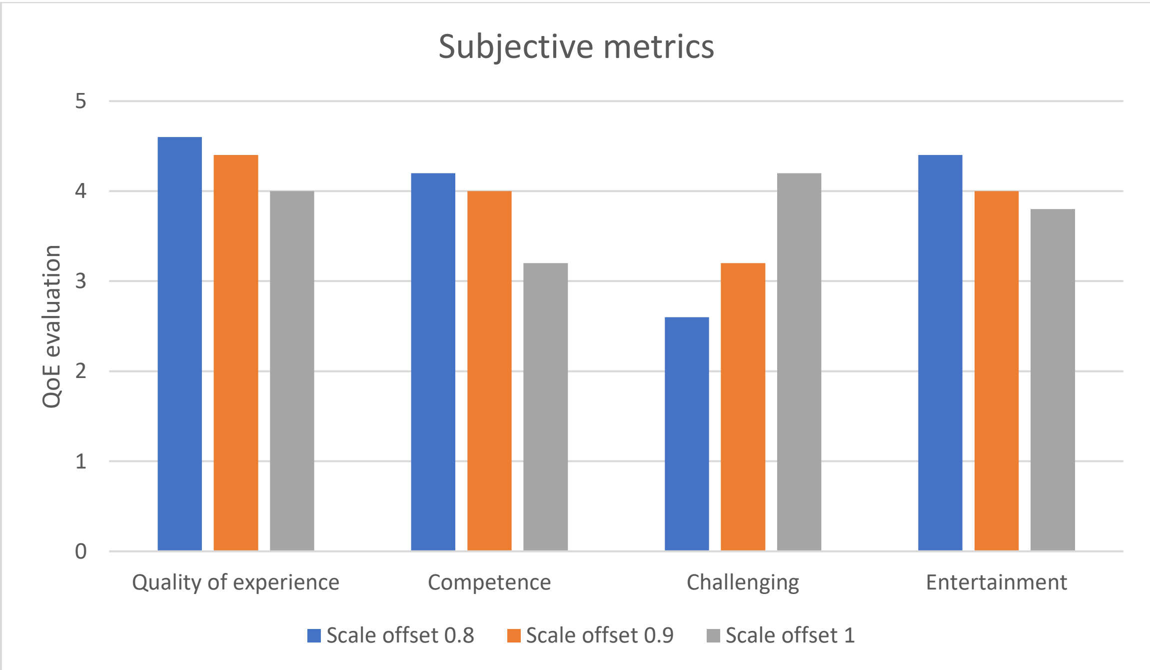

Figure 6.44 Evaluation of QoE (on a scale of 1: bad to 5: excellent), competence, challenge, and entertainment (on a scale of 1: very low to 5: very high) in the *Pick-And-Place* game using different scale offsets.

Figure 6.45 shows the simulation workload measures evaluation results for the group of scenarios in which different shoot forces were evaluated. In the scenario where the scale offset was set to 0.8, the average mental demand was 1.4 and the average physical demand was 1.2. When the scale offset was 0.9, the average evaluated mental demand was 1.8, and the average evaluated physical demand was 1.2. With the scale offset set to 1, the average mental demand was 2.2 and the average physical demand was 1.4. The evaluated task control difficulty averaged at 1.6 when the scale offset was 0.8, 2 when the scale offset was 0.9, and 3.4 when the scale offset was 1.

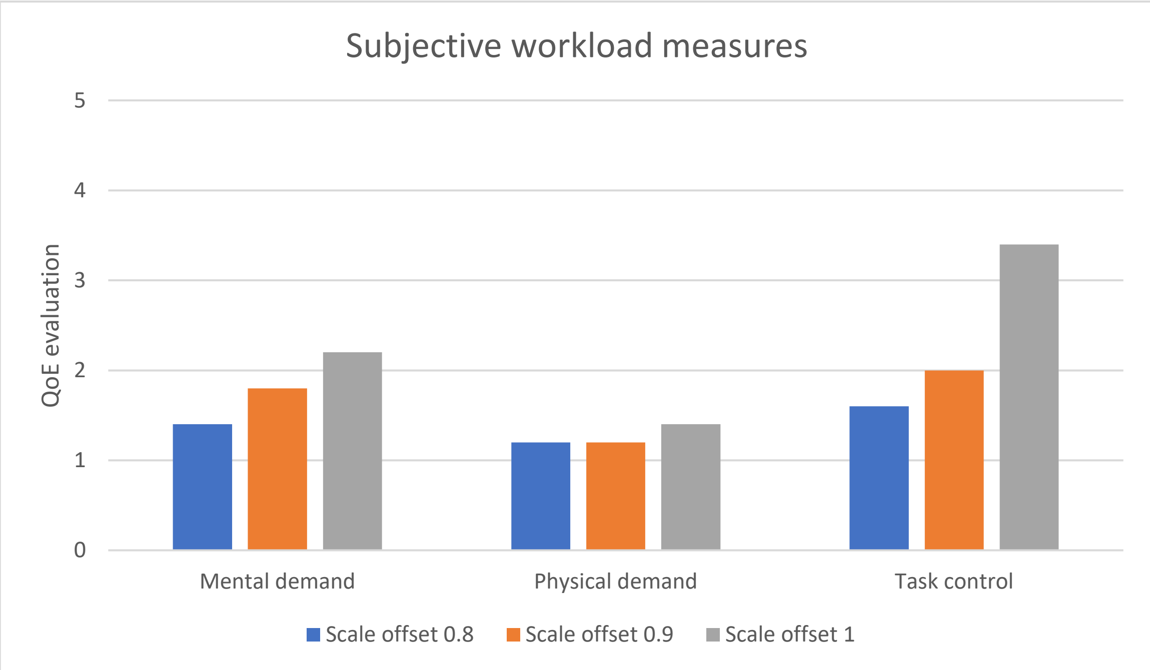

Figure 6.45 Evaluation of simulation workload in the *Pick-And-Place* game using different scale offsets (on a scale of 1: very low to 5: very high).

When asked to select the favorite scenario in this group, 4 out of 5 participants selected the scenario where the scale offset was set to 0.8 because they felt the most competent. The remaining participant selected the scenario where the scale offset was set to 1 because it presented the most challenge. When selecting the least favorite scenario in this group, 4 out of 5 participants chose the scenario where the scale offset was set to 1 because it was too difficult to complete, but one participant selected the scenario where the scale offset was set to 0.8 as the least favorite because they preffered it when more precision was required in completing the puzzle.

Subjective discomfort measures of experienced muscle pain and physical discomfort collected in this study were not analyzed in this thesis. However, this data is available and can be analyzed in future work.

6.3.3. Analysis

Figure 6.46 shows the comparison between objective accuracy, feeling of competence, and feeling of entertainment expressed as a percentage for the first group of scenarios which evaluated different puzzle scales. The participants were the most successful when the puzzle scale was set to 0.4, and the average feeling of competence was also the highest in this scenario (4.2) compared to the scenario where the puzzle scale was 0.1 (3.8) and the scenario where the puzzle scale was 0.7 (2.6). The participants were the least successful in completing the puzzle of scale 0.7, and the average number of pieces placed was 4.2 out of 7. Only one participant managed to complete the puzzle before the end of the round in the scenario with the scale set to 0.1 and the scenario with the scale set to 0.7, and this participant was an experienced user of VR. However, the highest average score was when the puzzle scale was 0.1 (4.6). Moreover, 4 out of 5 participants chose this scenario as their favorite in this group because the pieces were easier to manipulate. On the other hand, all participants selected the scenario where the puzzle scale was 0.7 as their least favorite in this group because it was harder to manipulate the pieces. The average task control difficulty for this scenario is also the highest (3.6) compared to the scenario where the puzzle scale was 0.1 (1.6) and the scenario where the puzzle scale was 0.4 (2.4).

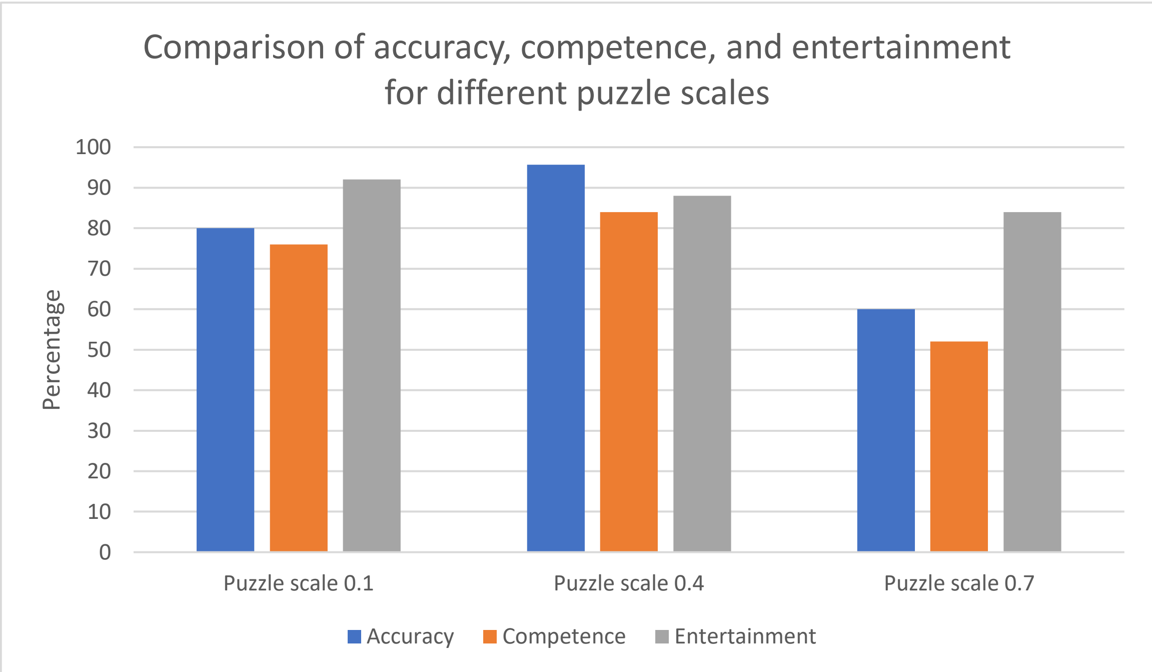

Figure 6.46 Comparison of accuracy, competence, and entertainment in the *Pick-And-Place* game using different puzzle scales.

The comparison between objective accuracy, feeling of competence, and feeling of entertainment for the group of scenarios which evaluated different collider scales is shown in Figure 6.47. The subjective measures are relatively similar for the scenario where the collider scale was set to 0.5 and the scenario where the collider scale was set to 1. When the collider scale was set to 0.2, however, the average QoE (3.4), the average feeling of competence (2.4), the average perceived entertainment (3.8) was the lowest in this group of scenarios. On the other hand, the average feeling of how challenging the game was the highest (3.6), compared to the scenarios where the collider scale was 0.5 (2.6) and the scenario where the collider scale was 1 (2). Moreover, when the collider scale was 0.2, the average mental demand (2.6) and the average task control difficulty (3.4) were significantly higher than for the other scenarios in this group. There was no significant difference in the evaluation of simulation workload components between the scenario where the collider scale was 0.5 and the scenario where the collider scale was 1.

All participants managed to complete the puzzle before the round ended in the scenario where the collider scale was set to 0.5 and the scenario where the collider scale was set to 1. However, when the collider scale was 0.2, only one participant completed the puzzle before the end of the round, and that participant was an experienced user of VR. Correspondingly, 80% of participants selected this scenario as their least favorite. On the other hand, most participants (60%) selected the scenario where the collider scale was 1 as the best in this testing group because it was the easiest to assemble the puzzle.

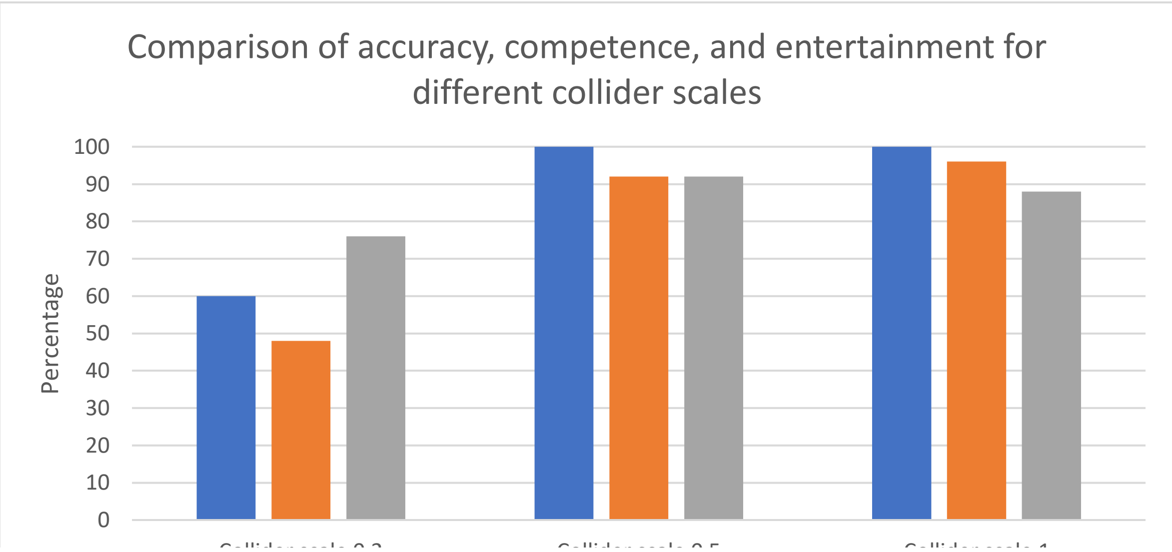

Figure 6.47 Comparison of accuracy, competence, and entertainment in the *Pick-And-Place* game using different collider scales.

Figure 6.48 shows the comparison between objective accuracy, feeling of competence, and feeling of entertainment expressed as a percentage for the group of scenarios where remote grab feature was enabled. The subjective evaluation of these two scenarios is relatively similar, but the participants were more successful in completing the puzzle when remote grab was disabled. In that scenario, all participants finished the puzzle before the end of the round, while in the scenario where the remote grab feature was enabled, 3 out of 5 participants finished the puzzle before the end of the round. This could be attributed to participants accidentally remotely grabbing the puzzle pieces already in the solution space, unintentionally decreasing the number of puzzle pieces placed into the solution cube at the end of the round. Additionally, task control difficulty had a higher average score (2.2) compared to the scenario where the remote grab feature was disabled (1.6).

On the other hand, the average scores for total QoE and the game's entertainment factor were slightly higher when remote grab was enabled. Moreover, all participants preferred the scenario where remote grab was enabled over the scenario where the remote grab feature was not available because they found it more entertaining and different from the real-world interactions.

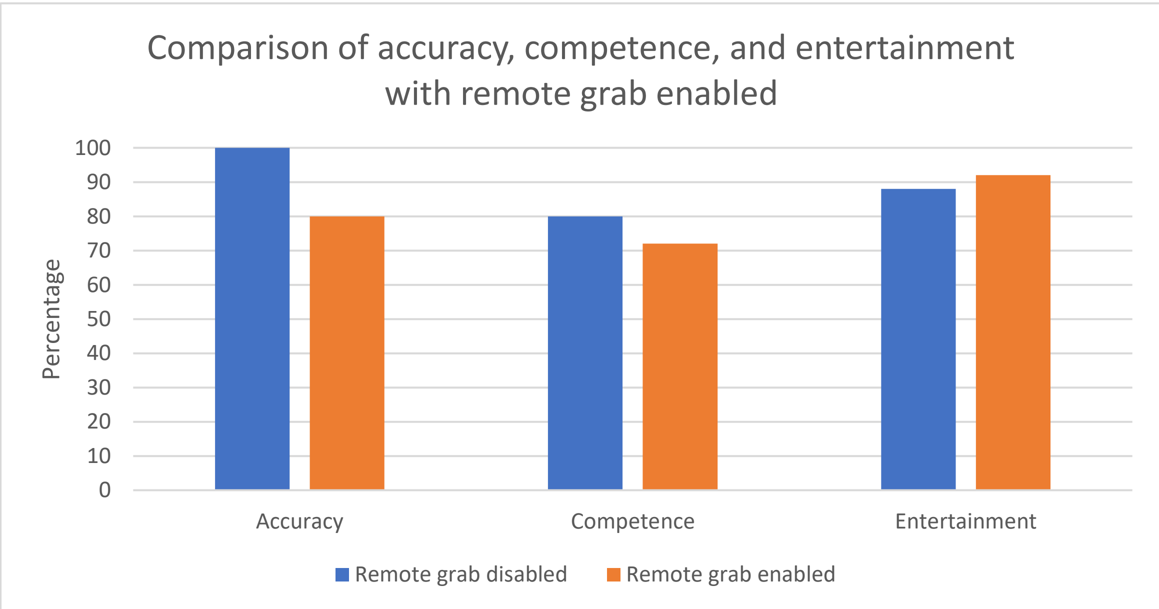

Figure 6.48 Comparison of accuracy, competence, and entertainment in the *Pick-And-Place* game using different grab types.

Figure 6.49 shows the comparison between objective accuracy, feeling of competence, and feeling of entertainment expressed as a percentage for the group of scenarios where different scale offsets were evaluated. The average perceived competence corresponds to the average success rate of completing the puzzle. When the scale offset was set to 0.8, all participants completed the puzzle before the end of the round. However, when the scale offset was 0.9, an average of 6.6 puzzle pieces were placed, and when the scale offset was 1, an average of 6 puzzle pieces were placed. Moreover, the average feeling of competence was the highest when the scale offset was 0.8 (4.2) compared to the scenario where the scale offset was 0.9 (4) and the scenario where the scale offset was 1 (3.2). The average user experience scores are relatively similar for the scenario where the scale offset was 0.8 and the scenario where the scale offset was 0.9, with the former having slightly better average scores. Correspondingly, most participants (4 out of 5) chose the scenario where the scale offset was 0.8 as their favorite scenario in this group. On the other hand, the majority of participants (4 out of 5) chose the scenario where the scale offset was 1 as their least favorite scenario in this group. This scenario had the highest average score on how challenging the game was (4.2) and how difficult the task control was (3.4) compared to the scenario where the scale offset was 0.8 (2.6 and 1.6) and the scenario where the scale offset was 0.9 (3.2 and 2).

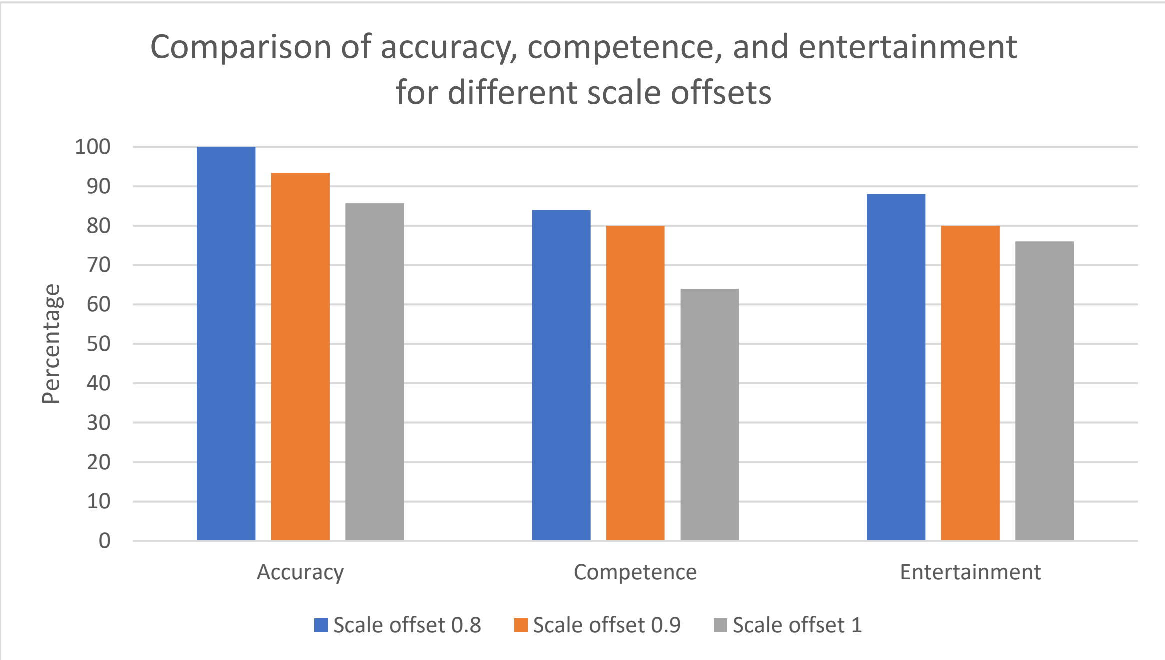

Figure 6.49 Comparison of accuracy, competence, and entertainment in the *Pick-And-Place* game using different scale offsets.

7. Future work

There are several ways the existing application can be improved and extended. In most games available for VR where the objective is to shoot targets with various projectile weapons, there is some form of urgency for the players to destroy the targets. An example would be the targets shooting back and causing virtual damage to the player. Since there is currently no penalty for the player in the *Shooter* game, the QoE results may be different and not represent the experiences of players in the commercially available shooter games. Therefore, an adjustment to the *Shooter* game is recommended in the form of a time limit for the targets to be destroyed and a limited number of targets that were not destroyed in time as a penalty for the player. Additionally, targets could be modified to shoot back at the player, directly imitating conditions in these types of games that are commercially available for VR.

Despite modifiable densities and occlusions being implemented in the *Pick-And-Place* game, they were not evaluated in the scope of this study because the players could assemble the puzzle in any order they preferred, so it would be impossible to select a wrong puzzle piece. As a result, it is suggested that the current implementation of the *Pick-And-Place* game be expanded by requiring a specific order in which the puzzle must be solved.

When conducting this study, participants were asked to open the desktop view inside the VR so they could fill out the questionnaire. However, since the feature of desktop view is not available for all virtual devices and because the purpose of the developed application is evaluating the QoE, a questionnaire that is available to study participants inside the virtual environment could be added to the existing testing framework.

Conclusion

There have been many attempts in history to make a widely used commercial virtual reality technology, however most were unsuccessful because of the technical limitations of their time, which led to poor user experiences. Today's VR technology has been the most successful in attracting a large number of users, however this number is limited by the QoE which is diminished by the design of interaction mechanics.

The goal of this thesis was to provide a way in which different interaction types and interaction mechanics parameters could be evaluated. A testing framework, which consists of a user interface for the test administrator to set interaction parameter values and a VR interface in which test subjects play games was developed. Three games that feature distinct types of interactions are available, and for each game, several modifiable interaction mechanics parameters are available.

Finally, a pilot study was conducted, and some of the data collected was analyzed. It was shown that some interaction parameter values have no significant impact on the overall QoE, while others significantly reduce the participant's willingness to continue playing the game. This framework could be a valuable tool for finding the best interaction mechanics parameters for games in VR for a better user experience and subsequently increase the number of users of VR.

Bibliography

[1] A. B. Craig, W. R. Sherman and J. D. Will, *Developing virtual reality applications: Foundations of effective design*, Morgan Kaufmann, 2009.

[2] Perkin Coie. (2022, June 6). *2020 Augmented and Virtual Reality Survey Results* [Online]. Available: https://www.perkinscoie.com/en/ar-vr-survey-results/2020-augmented-and-virtual-reality-survey-results.html

[3] A. Jarvinen, *Games without Frontiers - Theories and Methods for Game Studies and Design*, Tampere: University of Tampere, Finland, 2007.

[4] S. Vlahović, *A Framework for the Classification and Evaluation of Primary Game Mechanics for Virtual Reality Games*, Faculty of Electrical Engeenering and Computing, University of Zagreb, Zagreb, 2022.

[5] Britannica. (2022, May 28). *Definition of virtual reality* [Online]. Available: https://www.britannica.com/technology/virtual-reality

[6] The Franklin Institute. (2022 June 5). *History of virtual reality* [Online]. Available: https://www.fi.edu/virtual-reality/history-of-virtual-reality

[7] I. S. Pandžić, T. Pejša, K. Matković, H. Benko, A. Čereković i M. Matijašević, *Virtualna okruženja - Interaktivna 3D grafika i njene primjene*, Zagreb, Element, 2011.

[8] ALA TechSource. (2022, June 5). *Chapter 1. Introduction to Virtual and Augmented Reality* [Online]. Available: https://www.journals.ala.org/index.php/ltr/article/view/6778/9115

[9] A. Basso, "Advantages, Critics and Paradoxes of Virtual Reality Applied to Digital Systems of Architectural Prefiguration, the Phenomenon of Virtual Migration," in *Proceedings of the International and Interdisciplinary Conference IMMAGINI?*, Brixen, Italy, 2017.

[10] Kommando Tech. (2022, June 6). 30 *Virtual Reality Statistics for 2022* [Online]. Available: https://kommandotech.com/statistics/virtual-reality-statistics/

[11] The Pennsylvania State University. (2022, June 6). Interactions in VR [Online]. Available: https://www.e-education.psu.edu/geogvr/node/937

[12] I. Poupyrev, S. Weghorst, M. Billinghurst and T. Ichikawa, "A framework and testbed for studying manipulation techniques for immersive VR," in *Proceedings of the ACM symposium on Virtual reality software and technology*, 1997.

[13] S. M. LaValle, *Virtual Reality*, Cambridge University Press, 2016.

[14] W. R. Sherman i A. B. Craig, *Understanding Virtual Reality: Interface, Application, and Design*, Morgan Kaufmann, 2018.

[15] M. R. Mine, *Virtual Environment Interaction Techniques*, University of North Carolina, Department of Computer Science, 1995.

[16] D. P. Wiedemann, P. J. Passmore i M. Moar, "An experiment design: investigating VR locomotion & virtual object interaction mechanics," in *18th International Conference on Intelligent Games and Simulation*, Carlow, Ireland, 2017.

[17] M. Weise, R. Zender i U. Lucke, "How Can I Grab That?: Solving Issues of Interaction in VR by Choosing Suitable Selection and Manipulation Techniques," *i-com,* vol. 19, no. 2, pp. 67-85, 2020.

[18] A. Hameed, A. Perkis i S. Moller, "Evaluating Hand-tracking Interaction for Performing Motor-tasks in VR Learning Environments," in *Thirteenth International Conference on Quality of Multimedia Experience (QoMEX)*, 2021.

[19] Unity. (2022, June 7). *Welcome to Unity* [Online]. Available: https://unity.com/our-company

[20] Microsoft Visual Studio. (2022, June 7). *Meet the Visual Studio family* [Online]. Available: https://visualstudio.microsoft.com/

[21] Bearded Ninja Games. (2022, June 7). *Virtual Reality Interaction Framework* [Online]. Available: https://wiki.beardedninjagames.com/

[22] Vive. (2022, June 7). *HTC Vive Pro* [Online]. Available: https://www.vive.com/

[23] Valve. (2022, June 7). *Valve Index controllers* [Online]. Available: https://www.valvesoftware.com/en/index/controllers

[24] Amazon. (2022, June 7). *Image of Valve Index controllers* [Online]. Available: https://www.amazon.com/Valve-Index-Controllers-PC-Mac/dp/B086DJXXKV

[25] Google. (2022, June 7). *Google Forms* [Online]. Available: https://www.google.com/forms/about/

[26] F. Nemec, "Implementacija aplikacije za testiranje iskustvene kvalitete prolikom korištenja aplikacija u virtualnoj stvarnosti," Faculty of Electrical Engeneering and Computing, University of Zagreb, Zagreb, 2021.

[27] W. IJsselsteijn, Y. de Kort i K. Poels, "The Game Experience Questionnaire," Technische Universiteit Eindhoven, Eindhoven, 2013.

[28] NASA. (2022, June 21). *NASA Task Load Index* [Online]. Available: https://humansystems.arc.nasa.gov/groups/tlx/

[29] D. Harris, M. Wilson i S. Vine, "Development and validation of a simulation workload measure: The Simulation Task Load Index (SIM-TLX)," *Virtual Reality,* vol. 24, p. 557–566, 2019.

Summary

In this thesis, an overview of interaction mechanics and their significance in virtual reality is given. Three VR games - a target shooting game, a target slicing game, and a pick-and-place game - were created for the developed interaction testing framework that enables configuration of interaction parameters in those games. Using the developed solution, a QoE pilot study was conducted, and the methodology of the survey is described. Additionally, the results of the acquired subjective and objective data are presented and analyzed. Finally, recommendations for improvement and future user studies are listed.

Keywords: virtual reality, interactions mechanics, Quality of Experience

Sažetak

U sklopu ovog diplomskog rada dan je pregled interakcijskih mehanika i njihovog značaja za virtualnu stvarnost. Izrađena je aplikacija za testiranje interakcijskih mehanika i napravljene su tri igre u kojima se mogu podešavati različite vrijednosti nekoliko parametara interakcijskih mehanika. Provedeno je testiranje na manjem broju ispitanika, te su izneseni i analizirani prikupljeni podatci. Zaključno su navedene preporuke za unaprjeđenje postojećeg programskog rješenje i buduća korisnička ispitivanja.

Ključne riječi: virtualna stvarnost, interakcijske mehanike, ispitivanje iskustvene kvalitete

Appendix

General information

VR interaction mechanics testing

*Required

1. Unesite idenitifikacijski broj koji Vam je dodjeljen na početku ispitivanja *

____

2. Dob: *

____

3. Spol: *

Mark only one oval.

- muško
- žensko
- ne želim se izjasniti
- Other: _______________________

4. Koja vam je dominantna ruka? *

Mark only one oval.

- lijeva
- desna
- obje jednako

5. Odaberite tvrdnju koja najbolje opisuje Vaše iskustvo s uređajima za virtualnu stvarnost (engl. Virtual Reality, VR) *

Mark only one oval.

- Nikad nisam koristio/la uređaje za virtualnu stvarnost
- Isprobao/la sam uređaje za virtualnu stvarnost 1 do 3 puta u životu
- Povremeno koristim uređaje za virtualnu stvarnost, ali rjeđe od jednom mjesečno (u prosjeku)
- Koristim uređaje za virtualnu stvarnost jednom mjesečno ili češće

6. Kako biste opisali Vaš stav prema tehnologiji virtualne stvarnost (ako je još niste probali)?

Mark only one oval.

	1	2	3	4	5	
izrazito negativan						izrazito pozitivan

7. Kako biste opisali Vaš stav prema tehnologiji virtualne stvarnost (ako ste ju prethodno koristili)?

Mark only one oval.

	1	2	3	4	5	
izrazito negativan						izrazito pozitivan

Compare 3

VR interaction mechanics testing

*Required

1. Unesite idenitifikacijski broj koji Vam je dodjeljen na početku ispitivanja *

SCENARIJ 1/3

2. ID scenarija *

3. Kako biste ocijenili ukupnu kvalitetu ovog iskustva? *

Mark only one oval.

	1	2	3	4	5	
loše	⬭	⬭	⬭	⬭	⬭	odlično

4. U kojoj ste se mjeri osjećali kompetentno? *

Mark only one oval.

	1	2	3	4	5	
nimalo	⬭	⬭	⬭	⬭	⬭	jako

5. Koliko vam je igra bila izazovna? *

Mark only one oval.

	1	2	3	4	5	
nimalo	◯	◯	◯	◯	◯	jako

6. Koliko vam je igra bila zabavna? *

Mark only one oval.

	1	2	3	4	5	
nimalo	◯	◯	◯	◯	◯	jako

7. Koliko je zadatak bio mentalno umarajuć? *

Mark only one oval.

	1	2	3	4	5	
nimalo	◯	◯	◯	◯	◯	jako

8. Koliko je zadatak bio fizički umarajuć? *

Mark only one oval.

	1	2	3	4	5	
nimalo	◯	◯	◯	◯	◯	jako

9. Koliko je zadatak bio težak u smislu upravljanja/kontrola/navigiranja? *

Mark only one oval.

	1	2	3	4	5	
nimalo	⬭	⬭	⬭	⬭	⬭	jako

10. U kojoj ste mjeri osjećali fizičku bol u mišićima? *

Mark only one oval.

	1	2	3	4	5	
nimalo	⬭	⬭	⬭	⬭	⬭	jako

11. U kojoj ste mjeri osjećali fizičku nelagodu (npr. vrtoglavica, mučnina)? *

Mark only one oval.

	1	2	3	4	5	
nimalo	⬭	⬭	⬭	⬭	⬭	jako

12. Biste li nastavili igru u ovakvim uvjetima? *

Mark only one oval.

⬭ da
⬭ ne

USPOREDBA

35. Koji vam je scenarij bio najbolji u ovoj grupi? *

Mark only one oval.

- prvi
- drugi
- treći

36. Što vam se svidjelo u tom scenariju?

37. Koji vam je scenarij bio najgori u ovoj grupi? *

Mark only one oval.

- prvi
- drugi
- treći

38. Što vam je smetalo u tom scenariju?