UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

BACHELOR THESIS No. 1131

Development of a Local Multiplayer Competitive Game With Different User Interfaces

Marin Petric

Zagreb, June 2023

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Zagreb, 10 March 2023

BACHELOR THESIS ASSIGNMENT No. 1131

Student:	**Marin Petric (0036535607)**
Study:	Electrical Engineering and Information Technology and Computing
Module:	Computing
Mentor:	prof. Lea Skorin-Kapov

Title: **Development of a Local Multiplayer Competitive Game With Different User Interfaces**

Description:

Modern commercial Virtual Reality (VR) systems are based on technologies that enable the tracking of user movements in six degrees of freedom, thus providing an incentive for game and service developers to design creative methods of interacting with virtual environments. VR-based games have come to represent a significant part of the game market in the last few years, primarily due to the development of VR systems that are financially accessible to a wider consumer market. Although they share many features with games designed for other platforms, VR games require a number of specific design and development considerations. Of particular interest are local multiplayer games that allow one or more players to play using a VR system, while other players play simultaneously via a personal computer (PC). Your task is to design and implement a competitive local multiplayer game where players interact via a shared virtual world. The game should enable two players to play against each other, with one player using a head-mounted display and accessing the game via VR, while the other player accesses the game via a PC and uses a keyboard and mouse as input devices. The game should allow the PC player to create enemy units and obstacles so as to prevent the VR player from completing given objectives.

Submission date: 09 June 2023

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 10. ožujka 2023.

ZAVRŠNI ZADATAK br. 1131

Pristupnik:	**Marin Petric (0036535607)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentorica:	prof. dr. sc. Lea Skorin-Kapov

Zadatak: **Razvoj lokalne višekorisničke kompetitivne igre s različitim korisničkim sučeljima**

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 te služe kao poticaj razvijateljima usluga i igara za razvoj kreativnih metoda interakcije s virtualnim svijetom. Digitalne igre razvijene za virtualnu stvarnost su u posljednjih nekoliko godina postale značajan dio tržišta igara, prvenstveno zbog razvoja VR sustava koji su financijski prihvatljivi širem tržištu. Iako dijele mnoge značajke s igrama na ostalim platformama, igre u virtualnoj stvarnosti zahtijevaju poseban pristup prilikom dizajna i razvoja. Posebno su zanimljive lokalne višekorisničke igre koje omogućuju jednom ili više igrača igranje putem VR sustava, dok drugi igrači simultano igraju putem računala. Vaš je zadatak oblikovati i implementirati kompetitivnu lokalnu višekorisničku igru u okviru koje su igrači spojeni u dijeljeni virtualni svijet igre. Dva igrača igraju jedan protiv drugoga, pri čemu jedan igrač koristi zaslon montiran na glavi i pristupa igri putem tehnologije virtualne stvarnosti dok drugi igrač pristupa putem računala i koristi tipkovnicu i miš. Potrebno je omogućiti igraču za računalom stvaranje neprijateljskih jedinica te prepreka s kojima pokušava spriječiti igrača u VR-u da izvrši zadane ciljeve.

Rok za predaju rada: 9. lipnja 2023.

CONTENTS

1. Introduction

Virtual Reality (VR) is an expanding market segment that shows promise of becoming a crucial part of our daily lives in the coming decades. While VR has stagnated in popularity in the last years, the development of the technology has not, and with the recent announcement of the Apple Vision Pro *Head Mounted Display* (HMD), the industry is set to make significant leaps in the coming years [33][32]. The combination of *Augmented Reality* (AR) and VR promises to expand the spectrum of entertainment and experiences possible for people around the world to enjoy, letting people visit faraway lands with a click or talk to people from around the world as if they were there.

Because of the cost of entry for VR and its limited adoption thus far, many possibilities for the technology have yet to be explored [28]. VR's largest market segment is for commercial uses. These uses range from real estate to vehicle showrooms and retail stores, but the fastest expanding sector is the healthcare sector [6]. While smaller, the consumer market is important nonetheless. The games industry is a significant part of the consumer market, but because of VR's small user base, it hasn't received sufficient funding and development. Furthermore, local multiplayer games are even less represented and, as such, are the focus of this thesis.

Multiplayer games can be divided into multiple categories. Networked multiplayer games are games in which players can play on different devices which do not need to be physically close to each other. Such games typically use the Internet to connect players, but using a local area network is also possible. Local multiplayer games, on the other hand, can only be played on the same device. Another distinction can be made between competitive and cooperative multiplayer games. In cooperative multiplayer games, players help each other reach a common goal, whilst in competitive multiplayer games they actively play against each other.

This thesis will focus on implementing a local multiplayer competitive game for VR and desktop *Personal Computer* (PC). As such, this thesis will focus on these objectives:

– give an introduction to the concept of VR, its history, and games,

– describe the design of the VR Tactics game and how it was implemented,

– explain the challenges found during the development of the game and the possibilities for the further expansion of the game,

– summarise everything learned from this thesis.

These objectives are accomplished in the following four chapters of this thesis. The second chapter gives an overview of VR's history, an example of its most influential games, and explains the concept of local multiplayer games and their representation in the VR gaming sphere. The third chapter talks about the design of the VR Tactics game, what the game is about, and how it was implemented. The fourth chapter explores the challenges encountered during the game's development and the paths left open for expanding the game further. Finally, the fifth chapter outlines everything learned from this thesis.

2. Virtual Reality

VR is an advanced technology that immerses users in a simulated environment, creating a sense of presence and interaction [22]. It is a subset of *Extended Reality* (XR) in which a user is completely immersed in a digital world [9]. XR is a large spectrum that spans technologies like AR and VR. Contrary to VR, AR displays a modified version of the real world on top of which digital elements are added [17]. One of the earliest examples of this were the *Heads-Up-Displays* (HUD)s used in military aeroplanes to display instrument information to a pilot's helmet display.

On the other hand, VR, leveraging computer-generated graphics, realistic sounds, and interactive feedback systems, enables users to engage with digital content as if they were physically present within it. The applications of VR extend across various fields, offering transformative experiences and unlocking new possibilities in industries ranging from gaming and entertainment to healthcare, education, and beyond.

2.1. History of VR

The Sword of Damocles (Fig. 2.1) is widely considered to be the first VR headset, created by Ivan Sutherland in 1968 at Harvard University and so called because it was so heavy it had to be suspended above the head of the user, similar to the sword hanging above Damocles' head [24][34]. The headset consisted of two *Cathode Ray Tubes* (CRT) mounted to the user's head, drawing a wireframe representation of simple *Three-Dimensional* (3D) objects onto the real world. Needless to say, the headset was too cumbersome to become mainstream, but it paved the way for future development of the technology. VR, has experienced multiple booms and busts over its history. Until recently, the technology was too expensive to enter the consumer market in any meaningful way, only having practical applications in military aviation and the training of astronauts [3][7]. A notable attempt at a consumer VR headset was Nintendo's Virtual Boy (Fig. 2.2) [12]. With a monochrome display (Fig. 2.3) and being too heavy to carry on a person's head, the Virtual Boy was too uncomfortable for the

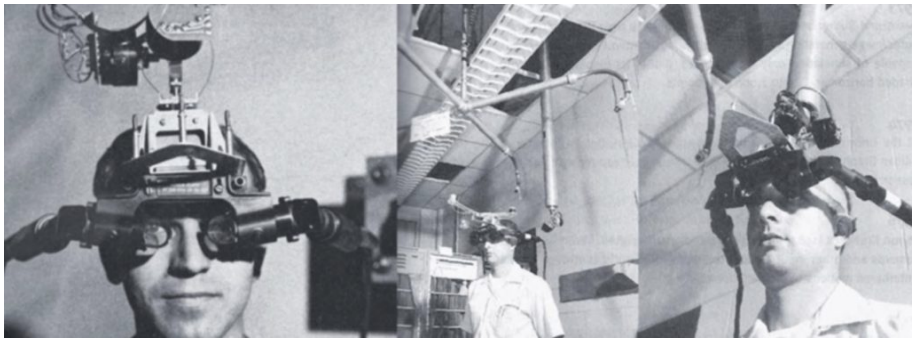

Figure 2.1: *From left to right: a) The head-mounted display optics with miniature CRTs, b) The mechanical head position sensor in use, c) The ultrasonic head position sensor in use (figure taken from [24][34])*

average consumer. This combined with a small games library led to an underwhelming number of sales, and Nintendo discontinued the headset in 1996.

Figure 2.2: *Virtual Boy (figure taken from [14])*

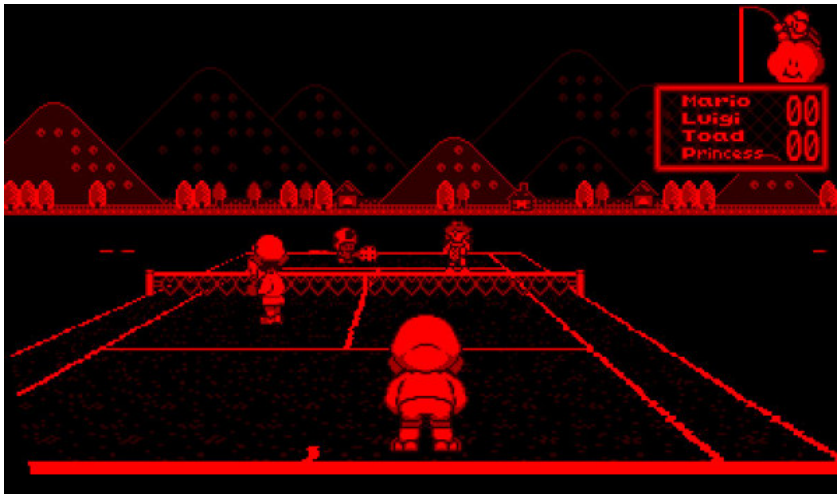

Figure 2.3: *Mario Tennis on the Virtual Boy (figure taken from [12])*

The first breakthrough for VR came in 2012 when Plamer Lucky started a Kickstarter campaign for the Oculus Rift (Fig. 2.4) [1]. After raising US$2.5 million, Oculus was acquired by Facebook in 2014 for US$2 billion [25]. This gave VR the push it needed to enter the consumer market, and soon after, hundreds of companies started announcing VR products. Another notable headset was the HTC Vive (Fig. 2.5), the first VR headset to implement a room-scale VR experience [18]. Room-scale VR refers to a setup where users can freely move and interact within a specified area, creating a more immersive and interactive experience that allows users to move around a virtual world without using controllers.

Figure 2.4: *Oculus Rift (figure taken from [15])*

The introduction of room-scale VR and Valve's push to get consumers to experience VR for the first time led to a spike in interest in VR. Room-scale allowed for new

Figure 2.5: *HTC Vive (figure taken from[18])*

experiences, letting consumers feel immersed in a virtual world where they were able to fully experience all six degrees of freedom. Consumers were no longer shackled to one spot, unable to move, now they could move around a game world using their own two feet. This hype, however, didn't last long, and even though Facebook, now known as Meta, released the Oculus Quest 2 in 2020, the best-selling VR headset, the interest in VR started to fade as the industry stagnated [19]. This spike in interest, followed by a sharp decline, is a phenomena often dubbed the *Gartner Hype Cycle*, according to the name of the research firm Gartner which publishes the graphical representations [26]. Meta's substantial investments into VR technology have yet to pay off, while the Oculus Quest 2 was a loss of money for Meta as they subsidised the headsets, selling them for less than they cost to produce [16]. Today the VR market is still growing, firms are still investing millions into the industry, but the excitement has waned compared to the mid-2010s.

2.2. VR Games

While the hype over VR may be temporarily dwindling, many highly appraised VR games still exist. One cannot talk about VR gaming without mentioning the two most important VR games to come out, *Half-Life: Alyx* and *Beat Saber* [8].

Half-Life: Alyx (Fig. 2.6), a sequel to Valve's highly appraised *Half-Life* series, is a *First-Person Shooter* (FPS) based in the Half-Life universe, and takes place after the events of *Half-Life* but before *Half-Life 2*. The series is well known for groundbreaking achievements that ushered new eras of gaming, from the first *Half-Life* game which had an immersive game world, introducing physics puzzles and a wide variety of weapons, to *Half-Life 2* which made physics a core gameplay element and further

Figure 2.6: *Half-Life: Alyx (figure taken from [27])*

developed the rich plot [20]. *Half-Life: Alyx* would not be a letdown, being the first game of the series in VR and becoming the industry standard benchmark for how to make a proper VR game. *Half-Life: Alyx* managed to deliver on all of the crucial parts needed to make a VR game satisfying. With spectacular graphics, satisfying and realistic physics, an engaging story, and a stunning amount of attention to detail, *Half-Life: Alyx* is one of the highest-quality VR games to come out thus far.

Compared to *Half-Life: Alyx*[1], *Beat Saber*[2] is not as visually impressive (Fig. 2.7), but it is not the best-selling VR game for no reason. *Beat Saber* manages to perfectly take advantage of VR's strengths, the freedom of movement and the ability to fully utilise your body, while skillfully avoiding its drawbacks, the immersion breaking caused by play-space boundaries. *Beat Saber* is a rhythm-based game in which players have to slash blocks which are flying towards them. Each successful slash grants the player points, and a miss penalises them. The game utilises VR's freedom of movement, allowing players to dodge obstacles with their bodies and slash the blocks with their hands. Since the player stands in one place and doesn't move (the blocks fly towards them), this reduces the chance player's immersion breaks because they are at the edge of their play space. The game also has a thriving mod community which improves upon the game, adding new songs and custom leaderboards. All of this, combined with the fact that the game has one of the lowest hardware requirements for VR games, makes the game extremely popular.

[1] `https://store.steampowered.com/app/546560/HalfLife_Alyx/`
[2] `https://www.beatsaber.com/`

Figure 2.7: *Beat Saber (figure taken from [27])*

2.3. Local Multiplayer Games

Local multiplayer games are a genre as old as computer gaming itself. Some of the first video games were local multiplayer, like 1958's Tennis for Two and 1972's Pong (Fig. 2.8), the first commercially successful arcade game [13]. Local multiplayer games can be defined as games in which two or more players play on the same device at the same time, either versus each other or working cooperatively. When it comes to VR, however, local multiplayer games are rare. The most significant reason is the fact that most systems do not support multiple VR headsets at the same time. If multiple VR headsets were supported there would still be problems. VR headsets are exceedingly expensive peripherals. Controllers are an order of magnitude cheaper than a VR headset. Secondly, VR is hard to run on modern hardware. Even high-end systems can struggle in some VR games. Adding multiple players to the same system would make rendering video in a sufficient number of frames per second impossible.

Local multiplayer games with one VR player and one or multiple desktop players do not suffer from these problems. Such games have one player wearing a VR headset and the rest of the players looking at the computer's main display, meaning that the performance required to run such games and the related costs are reduced. An excellent example of such a game would be *Takelings House Party* (Fig. 2.9)[3].

Takelings house party is a local multiplayer VR game developed by *DimnHouse*. In the game, the player wearing the VR headset is tasked with eliminating an infestation

[3] `https://store.steampowered.com/app/868150/Takelings_House_Party/`

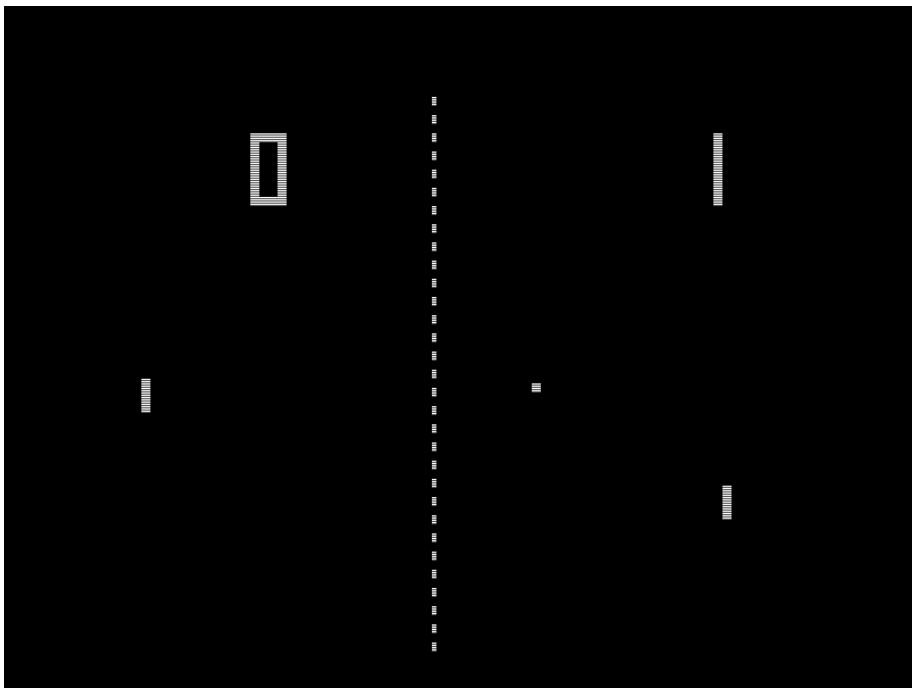

Figure 2.8: *Pong (figure taken from [10])*

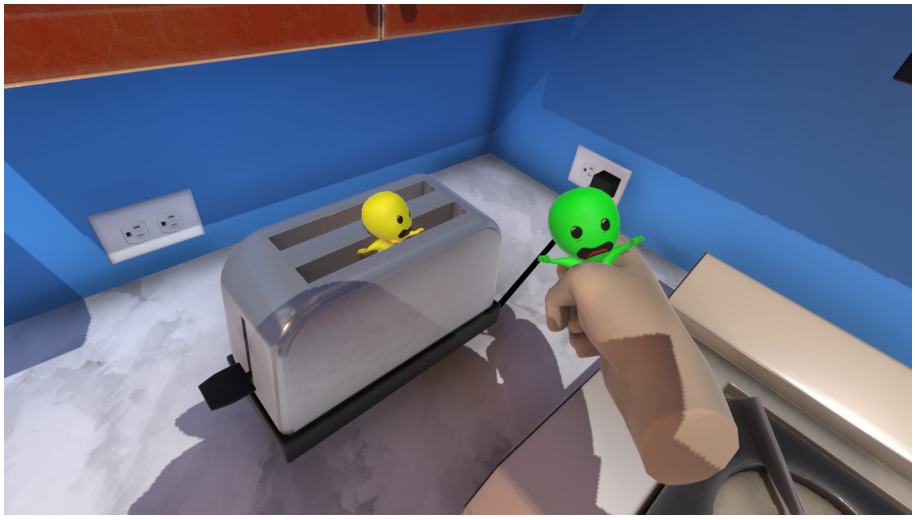

Figure 2.9: *Takelings House Party (figure taken from [2])*

of so-called *takelings*. These takelings are controlled by the other players using either a controller or their mobile phones. The takelings job is to survive as long as possible and collect coins to increase their score. The VR player has multiple ways of dealing with the takelings, usually using their environment or a set of tools, depending on what game mode is being played. The game makes excellent use of VR's advantages. With a large assortment of tools and methods for eliminating the takelings, the game gives the VR player a large amount of freedom to play however they wish using their tools however they wish.

3. Design and Development of the VR Tactics Game

This chapter describes the design and development of the local multiplayer competitive game *VR Tactics*. The development goal for this game was to create a proof of concept game that could showcase the potential of mixed experience games, combining a VR and classical desktop experience into one. Because of the performance limitations of VR and the scope of this game, a simple low poly art style and multiple third-party assets were used.

VR Tactics is a mixed experience local multiplayer competitive game between a player using a VR headset (the VR player) and a desktop player (the *Real-Time Strategy* (RTS) player). The game is set in a low poly city where the two players face off. The VR player's goal is to destroy the RTS player's Communication truck, the RTS player's connection to the area. The RTS player must defend this truck and use the resources at their command to stop the VR player, using any means necessary. The RTS player can create roadblocks and spawn units (Fig. 3.1) which they can use to engage the VR player. These units are SWAT police armed with assault rifles.

Figure 3.1: *An example of the RTS player placing down units*

While the RTS player can control each unit individually and give them explicit commands on where to go, the units can also move on their own. Once a unit has engaged the VR player, they can reposition themselves so that they are harder to hit for the VR player. If the VR player escapes from the unit's line of sight, the unit will run towards the VR player's last known position. This allows the RTS player to both be able to micro-manage the units and also for them to let the units fight on their own, not needing to babysit them the entire time.

The RTS player's *User Interface* (UI) is simple and consists of only two buttons which they can use to create new units. Once created, a unit can be selected by simply holding down the left mouse button and dragging the mouse cursor until a selection box is created above the unit (Fig. 3.2).

Figure 3.2: *An example of the RTS player selecting units*

3.1. Motivation

VR games represent a small fraction of today's games industry. A significant cause of this is that the number of users is small. A sector that is being ignored are local multiplayer VR games with mixed experiences. This market segment is especially under-served when considering the fact that only one person out of a group of friends would need a VR headset for everyone to enjoy such an experience.

VR Tactics represents a brand new look at the possibilities of VR games, combining an RTS with an immersive VR experience, giving players the ability to craft their own experiences each time. For the VR player, this allows them to feel like they are fighting real human beings who react according to the situation and are not placed manually

beforehand with fixed behaviours. The RTS player, on the other hand, is challenged by the skill of an actual human player controlling a character not bound by the constraints of classical desktop games. The VR player is able to think outside the box and use the environment to his or her advantage, forcing the RTS player to think about all the possibilities.

3.2. Technologies and tools

For the purpose of game development, a number of different tools were used, as described in this section.

Unity is a *Two-Dimensional* (2D) and 3D cross-platform game engine [31]. It is a flexible but easy-to-learn tool popular with many indie and mobile game developers [36]. Once developed, a game can be published to any of the large swath of platforms supported by Unity, including PC, Android, iOS, PlayStation, Xbox, and others. Unity uses C# as its scripting language, but it is also possible to write shaders using the *High-Level Shader Language* (HLSL) or Unity's Shader-Lab[30].

C# is an open-source, cross-platform object-oriented language developed by Mads Torgersen from Microsoft [11]. It has its roots in the C family of languages like Java and C++ and is used in game development and business applications. C# runs on an interpreter using Just-In-Time compilation to convert an intermediate language to machine code, giving the language a high degree of portability, only requiring that the host machines have a .NET runtime installed.

VR Interaction Framework **(VRIF)** is a collection of scripts and Unity prefabs developed by *Bearded Ninja Games* (BNG) [5]. The framework includes many ready-to-use systems, including a system that can be used to interact with objects, a player movement system, an input system and a weapons system. It can be used to get a project off the ground quickly and increase the pace of development of crucial game components.

Visual Studio Code (popularly referred to as VS Code) is a lightweight, open-source source-code editor developed by Microsoft [21]. It is a highly flexible, modular editor with thousands of user-made extensions available to download directly from the app. The unmodified version supports only a handful of languages, mainly JavaScript, HTML and CSS, but many extensions add support for other languages like Java, C/C++, C#, Python and others.

Git is a free, open-source distributed version control system authored by Linus Torvalds [35][29]. It is used for tracking project changes, usually for collaborative

software development, but it can be used for managing personal projects as well. It's main benefits are speed, reliability, and the ability to revert erroneous changes or combine multiple parallel workflows. While developing the game it was mainly used as a way to easily backup the game's source files and to be able to easily revert mistakes.

3.3. Implementation and features

3.3.1. Controllable units

Controllable units are all of the units that the RTS player can control. For the RTS player to control the units, they must be selectable and be able to move when given an order. As such, each unit has a script called *Controllable unit* which takes care of all the logic a single unit has to process. . The **Controllable unit** (fig. 3.3) script has multiple methods, the most important of which are *checkIfPlayerInLOS* and *setDestination*.

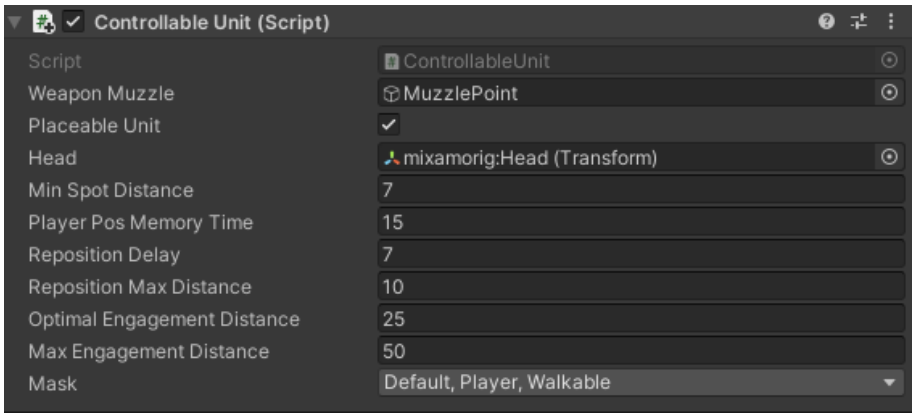

Figure 3.3: *Controllable unit script*

The *checkIfPlayerInLOS* (snipp. 3.3.1) method checks whether the VR player is currently in line of sight. If the player is in line of sight, their position is stored, and if the unit is not currently executing a move command, the unit turns towards the player and starts to shoot. The unit knows the position of everything within a certain radius around it, denoted by *min spot distance*. The consequence is that if the player ever gets closer to the unit than the specified distance, the unit will notice them, even if they are behind the unit. This emulates the unit's hearing and their general sense of awareness. Once the player is spotted, the unit will remember the player's position for *Player pos memory time* seconds. When the unit looses line of sight to the player it will go towards their last known position.

Snippet 3.1: checkIfPlayerInLOS method

```
1  private void checkIfPlayerInLOS()
2  {
3      Vector3 forward = head.forward;
4      forward.y = 0;
5      forward = forward.normalized * 0.5f;
6      Vector3 offsetHead = head.position + forward;
7
8      Vector3 playerRelativePos = VRPlayerController.position -
           offsetHead;
9      Ray ray = new Ray(offsetHead, playerRelativePos);
10     RaycastHit hit;
11     if(Physics.Raycast(ray, out hit, 500.0f, mask) &&
12         hit.transform.gameObject.layer == playerLayer)
13     {
14         Vector3 fwd = head.forward;
15         playerRelativePos.y = 0;
16         fwd.y = 0;
17
18         //If dot product is positive, the player is in front
19         if(Vector3.Dot(fwd, playerRelativePos) > 0)
20         {
21             //Don't run away if the player showed up in front
22             if(playerInLOS == false && lastMoved >
                   repositionDelay) lastMoved = Time.time;
23             playerInLOS = true;
24             playerPositionFound();
25         }
26         else playerInLOS = false;
27     }
28     else playerInLOS = false;
29     Debug.DrawLine(offsetHead, hit.point, Color.red);
30
31     //If the player is close to the unit
32     if(playerRelativePos.magnitude < minSpotDistance)
33     {
34         playerPositionFound();
35         return;
```

```
36      }
37
38      if(Time.time - lastPlayerSighting > playerPosMemoryTime)
            lastKnownPlayerPos = Vector3.zero;
39  }
```

The *setDestination* method is crucial for the movement of the units. The method is used whenever the RTS player gives a unit a movement command or whenever the unit decides to move somewhere. It updates the state of the unit so that it is consistent.

If a unit has been engaging the VR player for some time it will reposition itself automatically. This is where the *reposition delay*, *reposition max distance* and *optimal engagement distance* parameters come in. The *reposition delay* parameter tells the unit after how many seconds of engaging the player it should reposition itself. This means that if the player remains in line of sight for an extended period, the units will start to reposition themselves to make it harder for the player to hit them. The *reposition max distance* parameter defines the maximum distance the unit can decide to reposition by. Units randomly decide on a point in a circle the radius of *reposition max distance* around themselves and move to that point. Finally, the *optimal engagement distance* tells the units how far away they should keep themselves from the VR player. If they are closer to the player than the specified amount, their next repositioning will be biased away from the player. If they are too far from the player, the repositioning will be biased towards the player.

Finally, the *max engagement distance* parameter defines the max distance at which a unit can start engaging the player. If a unit spots the VR player but is outside of this range, it will move towards the player until it is closer than this maximum distance.

3.3.2. Selecting units

Units need to be selected for the RTS player to be able to control them. This is handled by the selection scripts *Unit selector* (fig. 3.4), *Click unit selector* and *Area unit selector*.

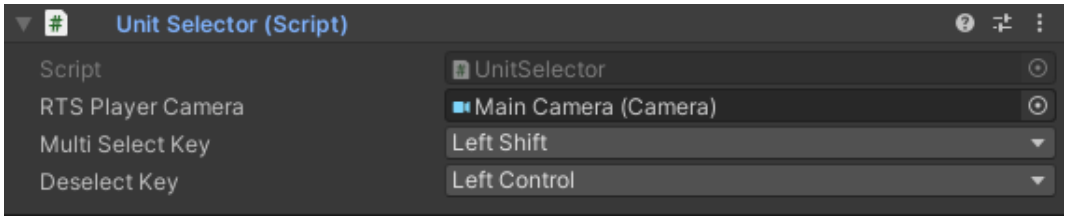

Figure 3.4: *Unit selector script*

The *Unit selector script* is the main script of the three. The *Click unit selector* and *Area unit selector* scripts tell the *Unit selector script* which units to select, but the *Unit selector script* is the one which actually selects units and stores references to them. The *Unit controller* script uses the *Unit selector script* directly to get a reference to all of the currently selected units when issuing commands. This allows us to easily abstract multiple selection methods beneath this one script, letting them handle finding the units that need to be selected.

Selected units are highlighted so that they are better visible to the RTS player (fig. 3.5). These highlights are however invisible to the VR player.

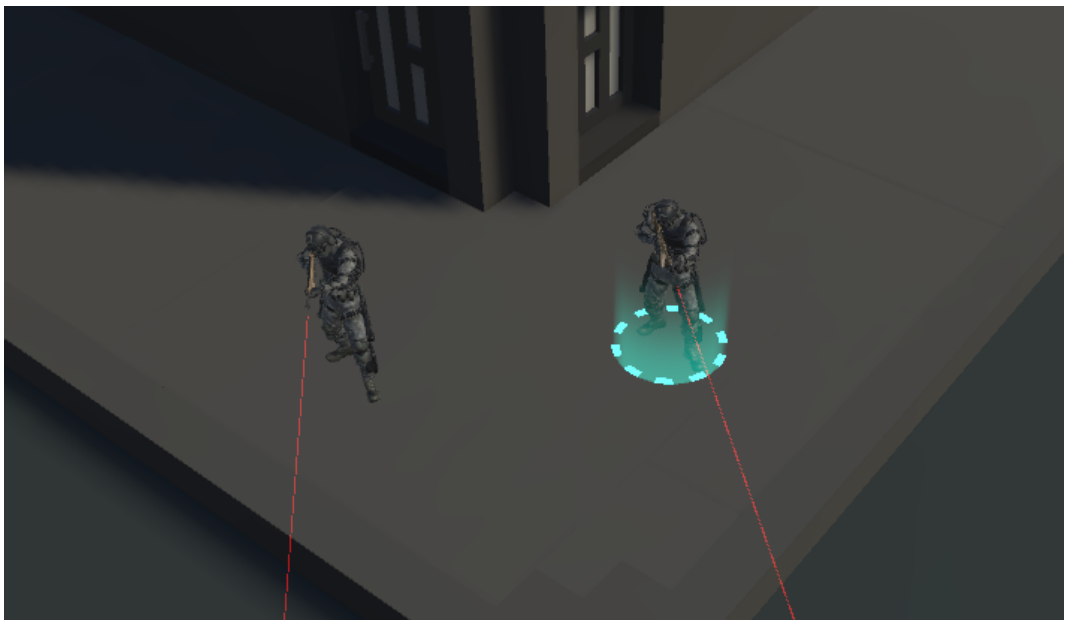

Figure 3.5: *An example of a selected unit next to a unit that is not selected*

3.3.3. VR player controls

Since managing the VR player's movement and controls is challenging and not the goal of this project, VRIF was selected as the framework which would handle all of the VR player's controls. This framework contains multiple useful features, including scripts that can be added to objects to make them be able to be picked up by the player, prefabs for the VR player and weapons. Some of the scripts and prefabs were modified to fit this project. The changes were small and have been put into separate scripts and prefabs.

The main components of VRIF used are the *XR Rig* prefab 3.6 and the weapons

prefabs. The *XR Rig* prefab is a complete VR player prefab which contains the player's model, hands, and teleportation indicators. The prefab was modified so that the VR player's model, which is a robot, is not visible to theVR player but is to the RTS player. This was achieved by putting the VR player's model into a separate layer visible only to the RTS player. The hands were modified so that only the VR player is able to see them using the same technique used for the player model, only changing the layer so that it is only visible to the VR player. The same technique was used for the teleportation indicator as well.

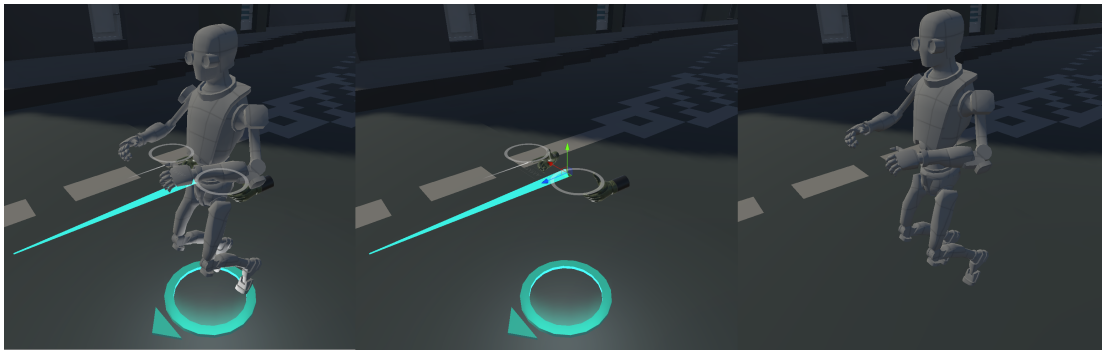

Figure 3.6: *From left to right: a) The VR player's model, b) The VR player's model as seen by the VR player, c) The VR player's model as seen by the RTS player*

The weapon prefabs used are the *Rifle*, *Rifle With Grip Variant* and *Shotgun*. The weapons used by the VR player only had their damage modified, but the weapons used by the RTS player's had other changes. Since the RTS player's units do not use the *Grabber* script from VRIF to hold the weapons, the *Rigid body* and *Grabbable* components were removed from them. In addition, the projectiles used by the guns were modified so that the bullet holes they produce are not visible to the VR player, since this can lead to the player's vision getting obscured by them if they hit them.

3.3.4. Objective

The objective consists of a handmade communications truck (fig. 3.7) model made in Blender[1]. The truck has a script which displays the amount of damage the truck has withstood as fire on top of the truck. If the truck is destroyed, the script activates an explosion and invokes the *onGameOver* event handled by the *Event manager* script.

[1] https://www.blender.org/

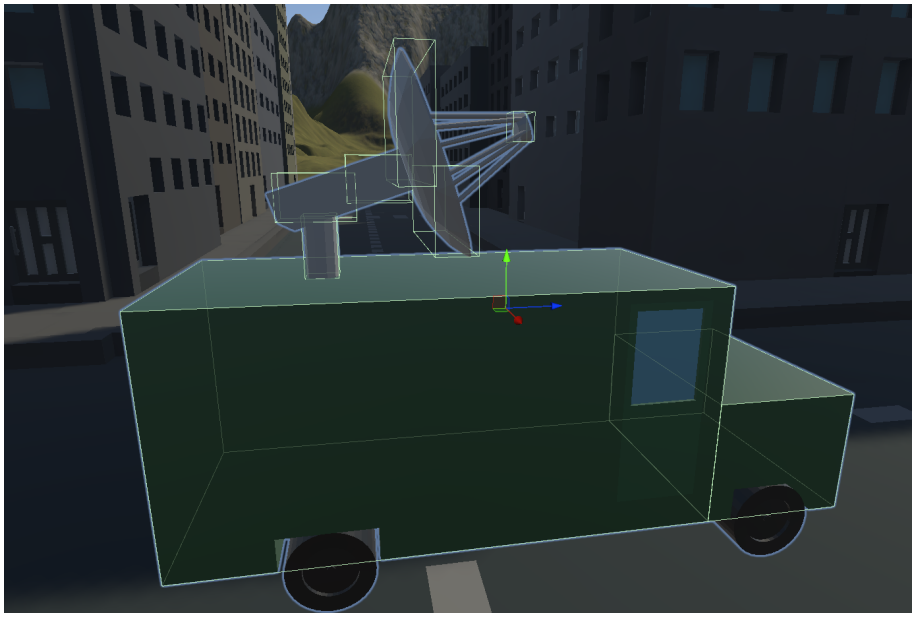

Figure 3.7: *The communications truck with its hitboxes visible*

3.3.5. Animation and models

The models and animations used for the RTS player's units were acquired from Adobe's Mixamo web service[2] which offers free high-quality 3D models and animations. Once downloaded, the animations and models can easily be imported into Unity. To connect the animations to the units, an animator (fig. 3.8) is used. This animator has multiple states between which it switches according to preset rules. These rules include the speed of the unit, whether the unit has been hit and whether the unit is dead. The *Walk* and *Run* states are Blend Trees (fig. 3.9) composed of multiple animations which are blended according to the speed and direction of the unit.

[2]https://www.mixamo.com

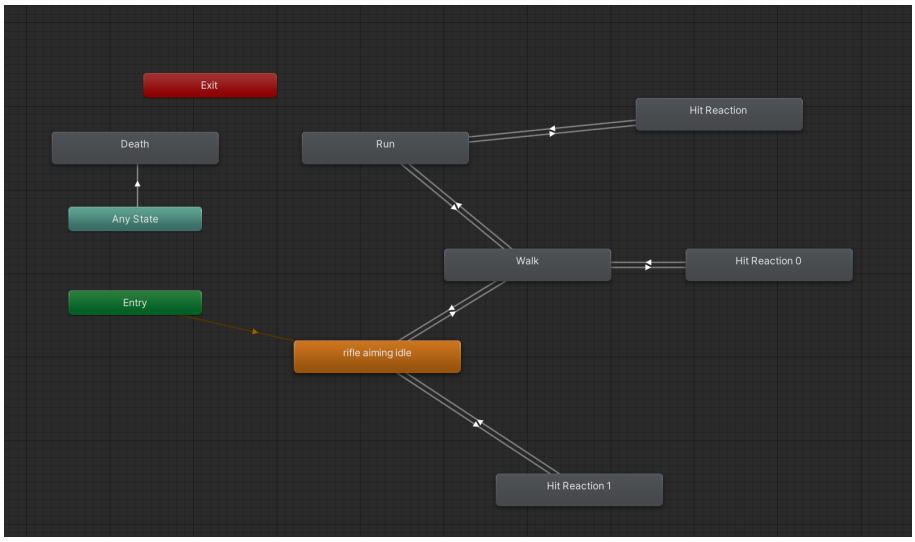

Figure 3.8: *The RTS player unit animator*

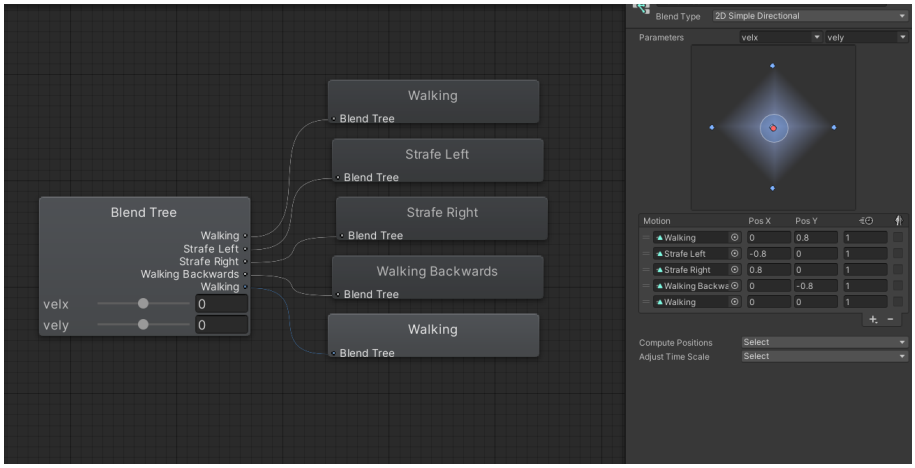

Figure 3.9: *The Walk state Blend Tree*

4. Challenges and Future Development

The development of this game has uncovered some of the limitations of today's technology. Nevertheless, further development of this game shows promise.

4.1. Challenges faced

During development there were multiple challenges faced, most stemming from the fact that mixed experiences for VR and desktop are not well documented and there are few developers working on them.

4.1.1. UI interference

A source of problems was the interference of the VR player's UI system with the RTS player's UI system. This is because VRIF does not expect a desktop user, and as such, the RTS player's UI is sometimes acted on by the VR player unintentionally. Currently the only possible solution to this would be the re-implementation of VRIF's UI input system, but this is outside the scope of this project. The problem has been mitigated by multiple means, but the full implementation of this game would require a more permanent fix.

4.1.2. Sound

Sound is an important source of immersion, as such, it would be optimal for each user to be able to listen to a separate audio output. This, however, is impossible in Unity and Windows[4], which represents a problem. The only viable alternative would be to implement online or *Local Area Network* (LAN) functionality, but this would require an additional device for the game to be playable.

4.1.3. Scale

Scale is hard to judge when looking at a game through an editor. Some elements can turn out to be much larger or smaller than expected when faced with VR. An example of this is the scale of the town selected for the main map. The town was selected for its low poly art style and compatibility with the game's design. The reason for this was that a natural environment would offer too little cover and be too open as well as the fact that a city is more easily traversable by the RTS player's units. As such, the town was thought to be adequate for the needs of the project. During later stages of development, it was realised that the sidewalks and road separators were too high for the VR player to traverse. Because this was found out too late it was impossible to change the map in time. For a future version of the game, a more suitable map should be chosen.

4.2. Further development

Since this game is only a preview of the possibilities of this genre there are many features that could be implemented to improve upon this game. There are also many elements which could be improved upon.

Visually, the most important improvements could be adding post-processing and improving the lighting. Currently, the baked lighting contains visual bugs and looks bland. Adjusting the settings for the in-game objects and lighting could significantly improve the look of the game.

4.2.1. Online multiplayer

A very important feature missing from the current implementation would be online multiplayer. This would make the game more accessible and make it easier for users to play the game, since it wouldn't require them to be in the same physical place. Another benefit is the aforementioned lack of dual sound output. With the implementation of online multiplayer this problem would be fixed.

4.2.2. Gameplay features

Currently, the game features only two units, out of which one is static, and a single map with a single objective. Future renditions of the game should add more units to

the RTS player's arsenal. Some possible game modes could include a capture-the-flag-like game mode where the VR player has to capture a specific objective and retrieve it back to their base. Another game mode could be survival based. The VR player would have to survive for a set amount of time, while the RTS player's objective is to eliminate them before the timer runs out. New maps should be added to increase the variety in gameplay. Finally, new weapons could be added. A forcefield grenade could be added which could serve as protection for the VR player as they reload or heal themselves.

5. Conclusion

While the history of virtual reality dates back further than most would guess at first, the technology has still not entirely permeated the consumer market. Today we are closer than ever to VR becoming widely accepted and available, and when it manages to break the comfort and ease of use barriers, the new possibilities it will offer will be exceptionally impactful.

Many local multiplayer games have been created, but very few for VR. While there are reasons for this, in this thesis we have explored the possibility of developing one such game. Local multiplayer games offer an experience that is more easily accessible to players, lowering the barrier to entry for VR by requiring only one player to own a headset, allowing more people to experience VR. This, in turn, will help bring VR into the mainstream. VR Tactics, the product of this thesis, strives to showcase the many possibilities of mixed experience games, especially in the RTS and local multiplayer genres, which have been neglected. In the game, the VR player controls a character that can freely move around the world. Their goal is to destroy the RTS player's communication truck. The RTS player's goal is to defend the truck and eliminate the VR player using the units that they can spawn. This is achieved by implementing a system for creating units, selecting units and giving them commands, as well as a system that allows the units to move autonomously. For RTS players, VR Tactics offers a new challenge by allowing their opponents to use the environment in ways impossible for regular desktop users. For VR players, it gives them the ability to face off against enemies who, thanks to the RTS player, can think much more strategically than regular *Artificial intelligence* (AI) enemies.

As VR gaming is still a young industry, there are few resources available for developing VR applications. Because of this, it can be hard to find documentation or help when facing problems. While this is a considerable issue, the growing popularity of VR and the resulting financial incentive for developers to make games for the platform will alleviate this over time.

BIBLIOGRAPHY

[1] Oculus Rift: Step Into the Game, 2012. URL https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game. last visited on 03/06/2023.

[2] Takelings House Party, 2019. URL https://store.steampowered.com/app/868150/Takelings_House_Party/. last visited on 03/06/2023.

[3] History of VR, 2019. URL https://www.vrs.org.uk/virtual-reality/history.html. last visited on 02/06/2023.

[4] Output audio on multiple devices, 2021. URL https://answers.microsoft.com/en-us/windows/forum/all/output-audio-on-multiple-devices/bf67c2e8-81e3-49e1-8c96-6282540c1e0b. last visited on 03/06/2023.

[5] VRIF Wiki, 2023. URL https://wiki.beardedninjagames.com/. last visited on 27/05/2023.

[6] Virtual Reality Market Size, Share & Trends Analysis Report By Technology (Semi & Fully Immersive, Non-Immersive), By Device (HMD, GTD), By Component (Hardware, Software), By Application, By Region, And Segment Forecasts, 2023 - 2030, 2023. URL https://www.grandviewresearch.com/industry-analysis/virtual-reality-vr-market.

[7] AMPLYFI. Trends in VR: Is Virtual Reality losing its momentum?, 2023. URL https://amplyfi.com/2023/02/10/trends-in-vr-is-virtual-reality-losing-its-momentum/. last visited on 02/06/2023.

[8] Neil Barbour. Top 10 VR Games By Revenue, 2020. URL https://www.spglobal.com/marketintelligence/en/news-

insights/blog/top-10-vr-games-by-revenue. last visited on 03/06/2023.

[9] Rachel Breia. What Is Extended Reality - XR Explained, 2022. URL https://sensoriumxr.com/articles/what-is-extended-reality. last visited on 03/06/2023.

[10] Bumm13. Pong, 2006. URL https://en.wikipedia.org/wiki/File:Pong.png. last visited on 03/06/2023.

[11] C#. A tour of the C# language. URL https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/. last visited on 27/05/2023.

[12] Benj Edwards. Unraveling The Enigma Of Nintendo's Virtual Boy, 20 Years Later, 2015. URL https://www.fastcompany.com/3050016/unraveling-the-enigma-of-nintendos-virtual-boy-20-years-later. last visited on 28/05/2023.

[13] David Ellis. *The official price guide to classic video games : console, arcade, and handheld games*. New York : House of Collectibles, 1965. ISBN 0375720383. URL https://archive.org/details/officialpricegui00davi/page/3.

[14] Evan-Amos. Virtual Boy image, 2011. URL https://commons.wikimedia.org/wiki/File:Virtual-Boy-Set.jpg. last visited on 03/06/2023.

[15] Evan-Amos. Oculus Rift image, 2017. URL https://commons.wikimedia.org/wiki/File:Oculus-Rift-CV1-Headset-Front_with_transparent_background.png. last visited on 03/06/2023.

[16] Michael L. Hicks. Despite Quest 2 sales success, Meta lost $10.2 billion on VR/AR last year, 2022. URL https://www.androidcentral.com/despite-quest-2-sales-success-meta-lost-102-billion-vrar-last-year. last visited on 29/05/2023.

[17] William L. Hosch. augmented reality, 2023. URL https://www.britannica.com/technology/augmented-reality. last visited on 03/06/2023.

[18] Paul James. HTC Vive Review: A Mesmerising VR Experience, if You Have the Space, 2016. URL https://www.roadtovr.com/htc-vive-review-room-scale-vr-mesmerising-vr-especially-if-you-have-the-space-steamvr/. last visited on 28/05/2023.

[19] Wagner James. Meta's Quest 2 Price Increase Due to Competition by Rec Room & VRChat, Contributes to Slow Install Base Growth, 2023. URL https://nwn.blogs.com/nwn/2023/02/meta-quest-2-install-base-q4-2022-under-18-million.html. last visited on 29/05/2023.

[20] Andy Kelly. The many ways Half-Life 2 changed gaming forever, 2014. URL https://www.gamesradar.com/7-ways-half-life-2-changed-gaming-forever/. last visited on 03/06/2023.

[21] Frederic Lardinois. Microsoft Launches Visual Studio Code, A Free Cross-Platform Code Editor For OS X, Linux And Windows, 2015. URL https://techcrunch.com/2015/04/29/microsoft-shocks-the-world-with-visual-studio-code-a-free-code-editor-for-os-x-linux-and-windows/. last visited on 27/05/2023.

[22] Henry E. Lowood. virtual reality, 2023. URL https://www.britannica.com/technology/virtual-reality. last visited on 03/06/2023.

[23] Merriam-Webster. Peripheral. URL https://www.merriam-webster.com/dictionary/peripherals. last visited on 09/06/2023.

[24] David Morelo. Meet the Sword of Damocles, the First VR Headset in the World, 2022. URL https://vrsource.com/meet-the-sword-of-damocles-the-first-vr-headset-in-the-world-17233/. last visited on 28/05/2023.

[25] Kyle Orland. Facebook purchases VR headset maker Oculus for $2 billion, 2014. URL https://arstechnica.com/gaming/2014/03/facebook-purchases-vr-headset-maker-oculus-for-2-billion/. last visited on 28/05/2023.

[26] Kasey Panetta. Top Trends in the Gartner Hype Cycle for Emerging Technologies, 2017, 2017. URL https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-

cycle-for-emerging-technologies-2017. last visited on 07/06/2023.

[27] Marin Petric. Image taken by thesis author, 2023.

[28] Felix Richter. AR & VR Adoption Is Still in Its Infancy, 2023. URL https://www.statista.com/chart/28467/virtual-and-augmented-reality-adoption-forecast/. last visited 09/06/2023.

[29] Ben Straub Scott Chacon. *Git pro 2nd Edition*. 2014. URL https://git-scm.com/book/en/v2.

[30] Shaders. Writing shaders overview, 2023. URL https://docs.unity3d.com/Manual/SL-ShadingLanguage.html. last visited on 27/05/2023.

[31] Adam Sinicki. What is Unity? Everything you need to know, 2021. URL https://www.androidauthority.com/what-is-unity-1131558/. last visited on 27/05/2023.

[32] Chris Smith. What does the Apple Vision Pro headset mean for the future of VR and AR?, 2023. URL https://bgr.com/tech/what-does-the-apple-vision-pro-headset-mean-for-the-future-of-vr-and-ar/. last visited 09/06/2023.

[33] Natalie Stechyson. Is virtual reality the future? Apple's betting on it, but VR experts say we may not be there yet, 2023. URL https://www.cbc.ca/news/business/apple-vision-pro-popularity-1.6867577. last visited 09/06/2023.

[34] Ivan E. Sutherland. A head-mounted three dimensional display. 1964.

[35] Linus Torvalds. Original git push, 2005. URL https://github.com/git/git/commit/e83c5163316f89bfbde7d9ab23ca2e25604af290/. last visited on 27/05/2023.

[36] Unity. Our company. URL https://unity.com/our-company. last visited on 27/05/2023.

LIST OF FIGURES

ABBREVIATIONS

AR *Augmented Reality*

HMD *Head Mounted Display*

XR *Extended Reality*

2D *Two-Dimensional*

3D *Three-Dimensional*

HLSL *High-Level Shader Language*

VR *Virtual Reality*

VRIF *VR Interaction Framework*

BNG *Bearded Ninja Games*

RTS *Real-Time Strategy*

UI *User Interface*

LAN *Local Area Network*

CRT *Cathode Ray Tubes*

AI *Artificial intelligence*

HUD *Heads-Up-Displays*

FPS *First-Person Shooter*

PC *Personal Computer*

GLOSSARY

game mode A specific gameplay variant or style within a video game, offering distinct rules and objectives for players to experience. 10

leaderboard In gaming, leaderboards are scoreboards that track and display players' rankings and achievements, fostering competition and motivating players to improve. 7

mixed experience In this thesis, a game that combines VR and PC desktop gameplay into one. 11, 12, 21, 24

mod Short for (Modification). In gaming, a mod is a user-created alteration to a game, adding new content or modifying existing elements. Mods enhance gameplay, customize experiences, and are shared within the gaming community. 7

multiplayer In gaming, multiplayer denotes the capability for multiple players to participate and interact simultaneously. 1, 2, 8, 11, 12, 22

peripheral A device connected to a computer to provide communication (such as input and output) or auxiliary functions (such as additional storage) [23]. 8

prefab In Unity, a reusable game object template that simplifies the creation and management of consistent elements within a game scene. 13, 17, 18

SWAT Short for special weapons and tactics, highly trained law enforcement units specializing in handling high-risk situations . 11

Development of a Local Multiplayer Competitive Game With Different User Interfaces

Abstract

This thesis explores virtual reality, its applications and history. A short history of VR is given, walking through the most important steps that have lead to today's technology. An analysis of the most successful modern VR games is given as well as an overview on the potential and problems of local multiplayer games in VR. The development process of the game, the motivation behind the game and the technologies used during development are explained. The problems found during the development are addressed, as well as the potential further expansion of the game.

Keywords: VR, RTS, Unity, local multiplayer games

Razvoj lokalne višekorisničke kompetitivne igre s različitim korisničkim sučeljima

Sažetak

Ovaj završni rad istražuje virtualnu stvarnost, njene primjene i povijest. Daje se kratka povijest virtualne stvarnosti (VR), prolazeći kroz najvažnije korake koji su doveli do današnje tehnologije. Također se daje analiza najuspješnijih modernih igara u VR-u, kao i pregled potencijala i problema lokalnih višekorisničkih igara u VR-u. Objašnjeno je i razvoj igre, motivacija iza igre i tehnologije korištene tijekom razvoja. Adresirani su problemi koji su se javili tijekom razvoja, kao i potencijalno daljnje proširenje igre.

Ključne riječi: VR, RTS, Unity, lokalne višekorisničke igre,