Spatial Mastermind VR

Serious game in VR to improve the visuospatial ability

Óscar Silvestre Payá

Final Degree Work
Bachelor's Degree in
Video Game Design and Development
Universitat Jaume I

July 18, 2023

Supervised by: Carlos González Ballester

Acknowledgments

I would like to express my sincere gratitude and appreciation to all those who have contributed to the successful completion of my Final Degree project. I extend my heartfelt thanks to the following individuals:

First and foremost, I am immensely grateful to my esteemed university colleagues, Guillermo Arnau Tamarit and David Castellà Corral, for their invaluable guidance and assistance in overcoming coding-related challenges that arose during the development of this project. Their expertise and dedicated efforts significantly contributed to its successful implementation. Additionally, I would like to acknowledge the valuable contributions of Iker Gimbert Sánchez, whose assistance in the modeling tasks of the project. I would like to extend recognition to Haroun Hamri, for his active participation and determination, particularly in recording the primary narrator's voice throughout the project.

Furthermore, I am deeply grateful to my friends and family for their support, profound understanding, and continuous encouragement throughout the demanding journey of this project. Their belief in my abilities and steadfast motivation served as a constant source of inspiration, empowering me to overcome obstacles encountered along the way.

Lastly, I would like to acknowledge Raúl Montoliu Colás, the current coordinator of our degree, for generously lending me the Oculus Rift S, which played a pivotal role in facilitating the development of this project.

In conclusion, I extend my gratitude to all other individuals who have contributed in any capacity to the successful completion of this project, even if their specific roles have not been explicitly mentioned.

Abstract

This document presents the project report of the Video Games Design and Development Degree Final project conducted by Óscar Silvestre Payá.

The project focuses on the development of a serious game leveraging virtual reality (VR) technology, with the explicit purpose of enhancing diverse visuospatial abilities inherent to individuals. The core gameplay revolves around a series of intricately designed mini-games, which engage players in the application of logical thinking and ingenuity, thereby facilitating substantial improvements in their visual-spatial capabilities. Moreover, the project endeavors to seamlessly integrate psychoeducational principles into the gaming experience, engendering an effective serious game that engrosses players within an immersive virtual reality environment. To ascertain progress in the aforementioned abilities, the game incorporates parameter tracking and assessment mechanisms.

The development process entailed the utilization of Unity 3D, today's most widely used game engine, in conjunction with a VR device, to deliver an elevated computer-based gameplay experience. By embracing innovative technologies and adopting a multifaceted approach that combines gaming elements with educational objectives, this project contributes to the expanding realm of serious games, while simultaneously offering a captivating avenue for players to augment their visual-spatial proficiencies.

Key Words

This section provides the key words associated with the project discussed in this thesis. These key words encapsulate the main ideas and concepts explored throughout the research and development process. Understanding and utilizing these key words will facilitate a comprehensive comprehension of the project's scope and objectives.

- **Video Games Design and Development**
- **Serious Games**
- **Virtual Reality technology**
- **Visuospatial abilities**
- **Psychoeducational principles**
- **Innovative technologies**
- **Educational objectives**

Contents

CHAPTER 1

Introduction

Contents

This chapter provides an elucidation of the motivations that led to the inception of the project idea, outlines the initially established objectives, and traces the developmental trajectory of the idea [7].

1.1 Work Motivation

By way of introduction, it is essential to specify that visuospatial skills refer to the cognitive abilities involved in mentally representing, analyzing, and manipulating objects. Many tasks classified as visuospatial also entail the utilization of other visual processing skills, which we encounter in our daily performance of various activities. While individuals possess unique abilities and skills, the development of visuospatial skills varies among people. Deficiencies in this domain can manifest as movement restrictions, accommodation difficulties, and impaired hand-eye coordination.

To address these challenges and enhance individuals' quality of life, the proposal is to develop a serious virtual reality (VR) game. This game aims to assist children and individuals lacking visuospatial skills in improving their abilities, leading to significant personal growth. Key aspects of visuospatial skills include spatial awareness, visual perception, mental rotation, visual memory, imagery, and spatial reasoning. These abilities facilitate effective navigation, interaction with the environment, and engagement in sports and practical tasks.

Improvement in visuospatial skills can be achieved through various training and rehabilitation methods. These methods encompass cognitive training, physical therapy, artistic activities, visual-motor training, visual-perceptual therapy, and computer-based training. However, it is important to consider that individual outcomes may vary, necessitating consultation with a cognitive rehabilitation therapist to determine the most suitable approach for each individual.

Visuospatial processing skills find practical application across various domains. In mathematics, the ability to mentally visualize objects rotating in space proves critical for disciplines such as trigonometry and calculus. Similarly, proficient reading comprehension necessitates an understanding that the orientation of specific shapes alters their meaning on the page. In sports, the interplay between visuospatial processing and visuo-motor skills enables individuals to synchronize movements based on visual cues, facilitating actions like catching a ball. Moreover, activity books featuring tasks like maze completion require adept visuospatial processing.

Challenging the common misconception of inherent mathematical deficiency, research demonstrates that deliberate practice and targeted stimulation of visual and spatial skills yield tangible improvements in these domains. Driven by a desire to provide a partial solution to the challenges faced by numerous individuals, addressing these issues serves as the impetus behind this initiative.

1.2 Objectives

The primary objective of this research is to explore the integration of Virtual Reality (VR) technology in the field of psycho-education, with the goal of implementing innovative techniques that enhance the learning experience. Essentially, the aim is to provide players with a virtual reality experience through a serious game.

In addition to the primary objective, there are several secondary objectives:

1. Investigate the transformative potential of puzzle-solving activities, examining how the development of problem-solving skills can lead to improved mathematical and reading abilities in players.
2. Provide a platform for players to practice and enhance their visual and spatial skills. By leveraging the interactive nature of the game, the intention is to facilitate significant improvements in these cognitive abilities.
3. Foster a comfortable atmosphere where players can interact with different games and puzzles within the virtual environment.

This research seeks to revolutionize cognitive development and academic progress by combining VR technology, psychoeducational principles, and immersive gameplay. By integrating extended reality (XR) technologies into educational contexts, the project aims to open up new avenues for nurturing and applying visual and spatial skills across various domains.

1.3 Environment and Initial State

The inspiration for developing a serious game in virtual reality (VR) emerged through a rather peculiar way. While reading Elsa Punset's book [21] in a Berlin laundromat, the idea of incorporating cutting-edge technologies like VR into the realm of education caught my attention. Punset's book centered on addressing social anxiety and fear of public speaking, prompting me to investigate existing solutions in this domain. Indeed, I discovered a few games that aimed to address psychological deficiencies or challenges faced by players. As I delved deeper into my research, I stumbled upon a prevalent issue affecting a substantial portion of the global population: the lack of training in visuospatial skills, including the mental manipulation of spatial rotations for various geometric forms.

With the topic selected, the next step involved devising a development plan. To navigate this process, I sought the guidance of a tutor experienced in the field of virtual reality. Collaboratively, we deliberated on the project's approach and determined the necessary tools.

The project was executed utilizing the Unity game engine, and visual assets were obtained from the available free assets in the Unity asset store, with additional contributions from 3D modeler Iker Gimbert. To ensure continuous improvement in project quality, regular meetings were conducted with the supervisor to review and provide feedback on each completed level. At its inception, the project relied on a modular Unity asset and incorporated the Oculus Integration package to facilitate user interactions. With this initial framework in place, subsequent efforts focused on the design and programming of each minigame, building upon the foundation established earlier.

Chapter 2

Planning and resources evaluation

Contents

This chapter delves into the technical aspects of the work, providing insights into the planning process for the project's development.

2.1 Planning

Effective project planning is an essential undertaking that ensures the maintenance of a cohesive workflow. In this regard, the subsequent lines elucidate the tasks accomplished in the course of project development. It is noteworthy that not all tasks were strictly executed sequentially, with certain activities being performed in an alternating manner. Complementing this description, a visual representation in the form of a Gantt chart [20] is provided in Figure 2.1, offering a more intuitive depiction of the aforementioned tasks.

- **Research and Learn (20 hours):** Extensive research was conducted to find articles and theses supporting the project's core idea of enhancing visuospatial skills through VR. Additionally, thorough investigation was undertaken to identify relevant references for mini-games inspiration and adaptation. Furthermore, dedicated efforts were made to comprehend and familiarize myself with the Oculus SDK [19], a new environment for the project.
- **Game concept design (10 hours):** The initial phase encompassed the design of each mini-game within the video game, outlining the map layout, object place-

ment, mechanics, narrator dialogue, level interactions and other pertinent details. Concurrently, efforts were dedicated to sourcing appropriate assets and models to complement the game design.

- **Game design and Development (190 hours):**
  - **Menu Scene (25 hours):** The implementation phase encompassed the integration of the entire map layout, including the creation of an interactive object room to facilitate player familiarity with the virtual environment. Additionally, some scene components such as the teleportation system, general character functionality, and camera operations were developed and incorporated into the project.
  - **Fix O'Clock (30 hours):** The development process involved designing and programming Level 1, which encompassed the creation of the level's visual aesthetics and the implementation of its corresponding programming elements. This mini-game is an adaptation of *NeuronUP* [15] activity.
  - **Shadow Puppets (35 hours):** The development process involved designing and programming Level 2, which encompassed the creation of the level's visual aesthetics and the implementation of its corresponding programming elements. This mini-game is an adaptation of *Jigsaw and Hidden Shapes test* [22].
  - **Drag The Cubes (40 hours):** The development process involved designing and programming Level 3, which encompassed the creation of the level's visual aesthetics and the implementation of its corresponding programming elements. This mini-game is an adaptation of *NeuronUP* activity.
  - **Throw Your Choice (40 hours):** The development process involved designing and programming Level 4, which encompassed the creation of the level's visual aesthetics and the implementation of its corresponding programming elements. This mini-game is an adaptation of a *Visual-Spatial Aptitude Test* [11].
  - **Results Scene (20 hours):** This phase involved the integration and design of interactive canvases, which were designed to respond to the precise ray tracking of the Oculus controllers. Additionally, a small floating interface was created to enable users to swiftly navigate between different levels within the game. Furthermore, a robust scoring system was implemented to ensure that the player's performance in each mini-game was accurately recorded and communicated throughout the game.
- **Debug and Improves (20 hours):** During this phase, comprehensive testing was conducted to identify and rectify visual anomalies, such as lighting inconsistencies, issues pertaining to the main character's height and movement, and project optimization, among others. Additionally, significant focus was placed on creating and refining the final build of the project.

- **Documentation (60 hours):** Preparation of essential documents, including the Final Degree Work report, presentation materials, and any other necessary documentation required for the successful completion of the project.

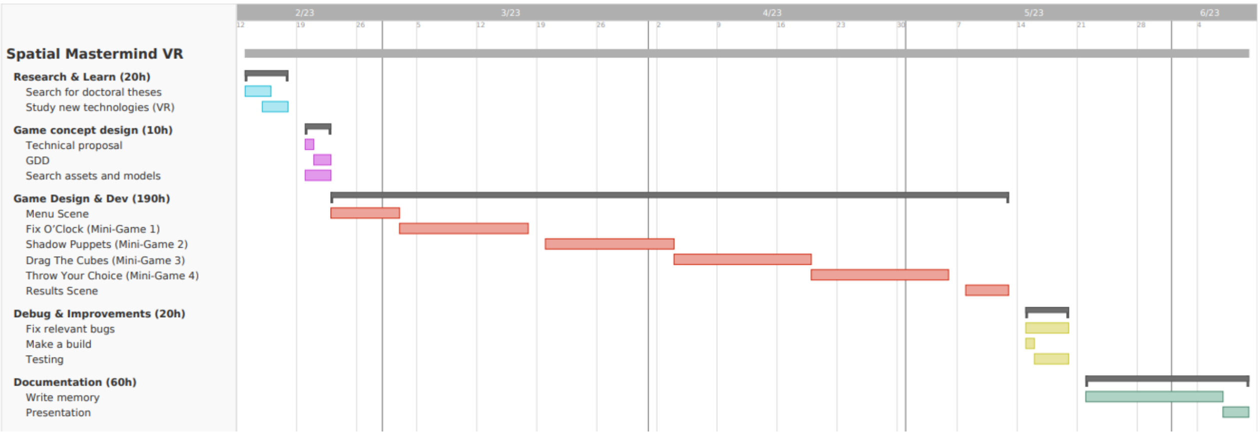

Figure 2.1: Gantt chart

2.2 Resource Evaluation

After discussing the project's timeline, this section proceeds to provide a comprehensive breakdown of the minimum hardware and software components required for the successful execution of the project. It is essential to note that the mentioned prices are approximate and subject to fluctuations due to the ever-evolving nature of the market.

- **Hardware**
  - **GPU:** NVIDIA GeFroce GTX 950. Cost: 130€
  - **CPU:** Intel Core i5-6400 2.70GHz. Cost: 150€
  - **RAM:** 8 GB. Cost: 24€
  - **VR Device:** Oculus Rift S. Cost: 400€ (when it was available on the market)
- **Software (Free)**
  - **Unity v.2021.3.19f1:** Unity is a versatile cross-platform game engine, enabling the creation of immersive 2D and 3D games, including virtual reality (VR) experiences [24].
  - **Visual Studio 2019:** An invaluable tool that accompanies Unity 3D is the integration with Visual Studio, providing a seamless coding experience and facilitating effortless transitioning between the Unity and Visual Studio environments. Moreover, Visual Studio offers a powerful suite of debugging tools that aid in the identification and resolution of software bugs [14].

- **GitHub (GitHub Desktop):** Is a web-based platform that serves as a central hub for software development and version control. GitHub allows users to store, manage, and share their source code repositories [13].
- **Krita:** Is a free and open-source digital painting software. It provides a comprehensive set of tools and features to create artwork and digital illustrations [12].
- **Unity Asset Store:** Is an online marketplace specifically designed for the Unity game development engine. It serves as a centralized platform where developers can find and acquire a wide range of digital assets, resources, and tools to enhance their Unity projects [25].
- **Overleaf:** Is an online collaborative platform for creating, editing, and sharing scientific documents in LaTeX, a typesetting system commonly used in academia and research. It simplifies the process of writing and collaborating on scientific papers, reports, theses, and other scholarly documents [17].
- **Blender:** Is a cross-platform computer program, especially dedicated to modelling, lighting, rendering, animation and the creation of three-dimensional graphics. It also deals with digital compositing using the procedural node technique, video editing, sculpture and digital pai [2].
- **Sketchfab:** Is a 3D modeling platform website to publish, share, discover, buy and sell 3D, VR and AR content. It provides a viewer based on the WebGL and WebXR technologies that allows users to display 3D models on the web, to be viewed on any mobile browser, desktop browser or Virtual Reality headset [8].
- **Turbosquid:** Is an American animation studio and digital media company that sells stock 3D models used in 3D graphics to a variety of industries, including computer games, architecture, and interactive training [23].

Chapter 3

System Analysis and Design

Contents

This chapter encompasses the analysis of requirements, design considerations, and architecture of the proposed project. Furthermore, it includes a detailed exploration of the graphics and styling aspects, as well as the design of the user interface.

3.1 Requirement Analysis

Let's commence by providing a clear overview of the game's mechanics and unique aspects, followed by an examination of the specific requirements it entails. Upon launching the game, the player will find themselves immersed in the *Main Menu* room, free from any intrusive heads-up display (HUD) elements that may disrupt the gaming experience. Within this room, a variety of objects are present, offering opportunities for player interaction and enabling them to familiarize themselves with the VR system. This noteworthy characteristic is further explained in Section 3.5.

To navigate within the game, players can utilize the *WASD* keys (see Figure 3.1) for movement, coupled with the VR viewer. However, to interact with objects, the use of controllers is indispensable. The *Left Joystick* controls player movement, while the *Right Joystick* governs camera movement, supplemented by the natural head mobility afforded by the VR headset. Additionally, by closing the left virtual fist (by pressing all interaction buttons), the player's avatar gains increased speed, allowing sprint movement. Object interaction involves uniformly grasping objects by closing either fist and bringing the hand into contact with the desired item. Furthermore, players can interact with in-game buttons and pressers by aligning their hand with the corresponding position of the controller and applying pressure. If the player press *Esc* key, the game will be stopped instantly.

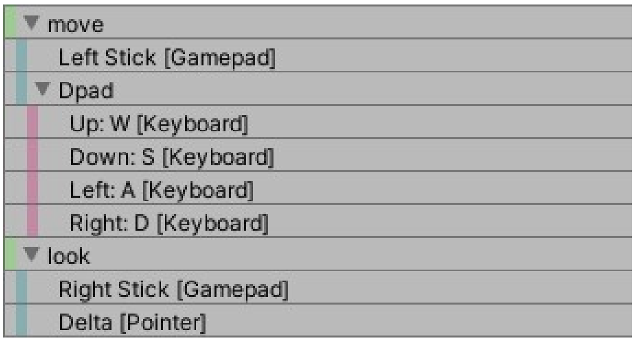

Figure 3.1: Game controls

In the *Results Menu* room, another type of interaction is available. Here, players can direct either controller towards any toggle on the main canvas and make selections by pressing the trigger button of the controller. This enables them to navigate through the menu and perform desired actions.

3.1.1 Functional Requirements

With a clear understanding of the aforementioned explanation, we will now proceed to provide a comprehensive breakdown to understand the functional requirements that constitute the videogame. These are the following found:

- **R1:** The player can start the game
- **R2:** The player can quit the game.
- **R3:** The player can move through the mini-games and Main Menu.
- **R4:** The player can sprint.
- **R5:** The player can interact with grabbable objects.
- **R6:** The player can press VR buttons with the virtual hands.
- **R7:** The player can access any mini-game regardless of the order of access.
- **R8:** The player can return to the Main Menu from any other room.
- **R9:** The player can access to the Results Menu only from the Main Menu.

- **R10:** The player can interact with the Results Menu room canvas by ray-tracing of controller.
- **R11:** The system will be able to generate items in the scene.
- **R12:** The system will be able to generate random order of objects in the scene.
- **R13:** The system will be able to save punctuations throw diferents plays.
- **R14:** The system will be able to move the player between the different scenes.

3.1.2 Non-functional Requirements

The design or implementation of a project is governed by non-functional requirements, which impose specific conditions. In the context of this project, the following non-functional requirements have been identified:

- **R15:** The game will be maintain a consistent frame rate (60 frames per second) to ensure smooth gameplay and prevent motion sickness in VR.
- **R16:** The game will be playable on PC with Oculus VR device.
- **R17:** The game will be efficiently utilize system resources like CPU, GPU and memory to avoid excessive hardware requirements.
- **R18:** The game will use realistic models.
- **R19:** The game's interface and controls will be intuitive and user-friendly to facilitate easy learning and engagement.
- **R20:** The virtual environments and objects will be visually appealing and realistic to enhance immersion and enjoyment.
- **R21:** The game will be provide clear instructions and guidance to players, ensuring they understand the objectives and mechanics of each mini-game.
- **R22:** The game will be provide write instructions and audio descriptions, to make the game accessible to users with disabilities.

3.2 System Design

This section introduces the conceptual and operational planning of the system under development. The logical design is illustrated through a use case diagram (see Figure 3.2), showcasing the various functionalities and interactions of the system. Furthermore, comprehensive descriptions of each use case are outlined in the following sections, outlining their specific actions and outcomes.

Requirement:	R1
Actor:	Player
Description:	The player starts the game launching the app
Preconditions:	None
Normal sequence:	1. The system loads the *Main Menu*
Alternative sequence:	None

Table 3.1: Case of use «Start game»

Requirement:	R2
Actor:	Player
Description:	The player exits from the game
Preconditions:	1. The player must be in the video game
Normal sequence:	1. The player presses the *Esc* key 2. The system quits the game
Alternative sequence:	None

Table 3.2: Case of use «Quit game»

Requirement:	R3
Actor:	Player
Description:	The player moves through the mini-games and *Main Menu*
Preconditions:	1. The player must be in any room except the *Result Menu* room
Normal sequence:	1. The player moves with *Left Joystick* and moves the camera with *Right Joystick* 2. The character and his/her camera moves in the direction assigned to these joysticks
Alternative sequence:	1. The player presses *WASD* keys 2. The character moves in the direction assigned to that button

Table 3.3: Case of use «Move»

Requirement:	R4
Actor:	Player
Description:	The player moves sprinting
Preconditions:	1. The player must be in any room except the *Result Menu* room
Normal sequence:	1. The player closes the left virtual fist by pressing all interaction buttons 2. The character sprints in the direction pointed to
Alternative sequence:	1. The player presses *SHIFT* key

Table 3.4: Case of use «Sprint movement»

Requirement:	R5
Actor:	Player
Description:	The player interacts with all the grabbable objects
Preconditions:	1. The object must has grabbable property
Normal sequence:	1. The player has to move his right or left hand until it collides with the object 2. he player has to close the right or left virtual fist by pressing all interaction buttons
Alternative sequence:	None

Table 3.5: Case of use «Grabbable interactions»

Requirement:	R6
Actor:	Player
Description:	The player presses VR buttons to play *Mini-game 1*, *Mini-game 2* and *Mini-game 3*
Preconditions:	1. The player must be in the correspondent mini-game
Normal sequence:	1. The player presses with virtual hands the VR button 2. The button collides with the hand and sinks to its base 3. The system checks whether the conditions of each mini-game
Alternative sequence:	1. The player presses with a grabbable object the VR button

Table 3.6: Case of use «Press VR buttons»

Requirement:	R7
Actor:	Player
Description:	The player accesses to all the mini-games regardless of the order of access
Preconditions:	1. The player must be at *Main Menu*
Normal sequence:	1. The player understands the location of each mini-game in the map 2. The player moves to the location of each portal 3. The player collides with the portal and the room of the corresponding mini-game is loaded by the system
Alternative sequence:	1. The player moves to the *Results Menu* 2. The player collides with the portal and the room is loaded 3. The player selects by collision of the controller with the floating buttons panel 4. The system loads the corresponding mini-game

Table 3.7: Case of use «Access to mini-games»

Requirement:	R8
Actor:	Player
Description:	The player returns to the *Main Menu* from any other room
Preconditions:	1. The player is in the *Mini-game 1* or *Mini-game 2* or *Mini-game 3* or *Mini-game 4*
Normal sequence:	1. The player moves to the left door, taking the red light as a reference 2. The player collides with the invisible collider of the door and an animation and opening sound is triggered 3. The player has to throw himself into the light orb to activate the collider 4. The system loads the *Main Menu* room
Alternative sequence:	1. The player is in the *Results Menu* 2. The player selects by collision of the controller with the *Main Menu* button of the floating panel 3. The system loads the *Main Menu* room

Table 3.8: Case of use «Return to the Main Menu»

Requirement:	R9
Actor:	Player
Description:	The player accesses to the *Results Menu* only from the *Main Menu*
Preconditions:	1. The player must be in the *Main Menu*
Normal sequence:	1. The player understands the location of the *Results Menu* in the map 2. The player moves to the *Results Menu* 3. The player collides with the portal and the room is loaded 4. The system loads the corresponding room
Alternative sequence:	None

Table 3.9: Case of use «Access to the Results Menu»

Requirement:	R10
Actor:	Player
Description:	The player interacts with the punctuation's canvas by ray-tracing of controller
Preconditions:	1. The player must be in the *Results Menu*
Normal sequence:	1. The player points with his controls through a visible ray 2. The player interacts with the four toggle buttons on the main canvas 3. The player selects one of the buttons by pointing the ray on top of it and pulling the controller trigger 4. The system changes the colour of the ray when the trigger is pulled 5. The system loads a new canvas on the right side showing the correct scores 6. The player pulls the trigger outside the buttons to make the right canvas disappear
Alternative sequence:	None

Table 3.10: Case of use «XR interactions»

Figure 3.2: Case use diagram (made with `https://online.visual-paradigm.com`)

3.3 System Architecture

The prerequisites for running the project build on a PC are as follows (see Table 3.11):

Component	**Recommended Specifications**	**Minimum Specifications**
Processor	Intel i5-4590/AMD Ryzen 5 1500X or higher	Intel i3-6100/AMD Ryzen 3 1200, FX4350 or higher
Graphics Card	NVIDIA GTX 1060/AMD Radeon RX 480 or higher	NVIDIA GTX 1050 Ti/AMD Radeon RX 470 or higher
Alternative Graphics Card	NVIDIA GTX 970/AMD Radeon R9 290 or higher	NVIDIA GTX 960 4 GB/AMD Radeon R9 290 or higher
Memory	8 GB of RAM or more	8 GB of RAM or more
Operating System	Windows 10 or later versions	Windows 10 or later versions
USB Ports	One USB 3.0 port	One USB 3.0 port
Video Output	DisplayPort compatible video output	MiniDisplayPort video output compatible (miniDisplayPort to DisplayPort adapter included with Rift S)

Table 3.11: Oculus Rift S Specifications

Furthermore, the utilization of an Oculus Rift S headset, accompanied by its corresponding controllers, is essential. These prerequisites are based on the specifications provided by Meta in their documentation [18].

3.4 Graphics and Styling

The video game's style and graphics are characterized by a combination of simplicity, clarity, and a futuristic aesthetic, which adds an energetic ambiance to the overall experience. It is important to note that all the game's artwork is sourced from free portals and other available resources. This deliberate choice reflects the project's focus on programming and development rather than the design and artistic aspects of the video game. Moreover, the project ensures the proper recognition and respect of the copyrights belonging to the owners of the assets used.

Upon initiating the game, players will find themselves in a futuristic ship or home environment consisting of modular rooms. Each room represents a distinct game screen with unique challenges (see Figure 3.3).

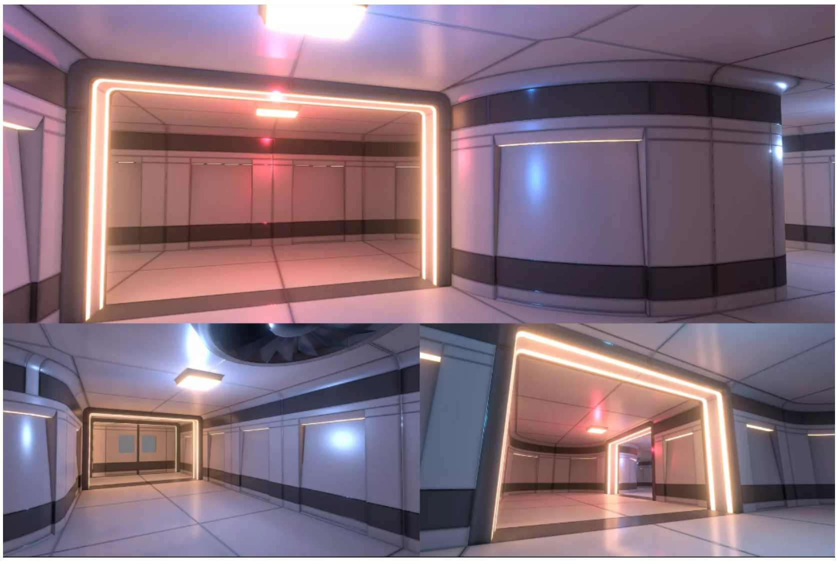

Figure 3.3: Asset modular map

The mini-game rooms exhibit a consistent visual theme, with slight variations in their design. Upon entering each room, players are presented with an instructional screen that provides guidance on the task they are about to undertake. The results room serves as the sole interface within the game, allowing players to analyze their performance in the tests and track their progression throughout the challenges.

While the primary focus of this project lies in the programming and development aspects, it is important to note that I also took on some design responsibilities. Although my involvement in the artistic aspects was limited, I contributed to the project by creating several simple designs. These include button designs for mini-games 2 and 4, the map layout, information panels, and canvas elements throughout the game [3]. Additionally, I took on the tasks of logo design [5], crafting the narrator's script [10], and conducting post-production work on all audio clips [1].

3.5 Interface Design

This design approach eliminates the need for a heads-up display (HUD) or any form of on-screen interface, as the game aims for complete immersion. Players are encouraged to explore the virtual world they inhabit and interact with its functionalities.

Due to the aforementioned reasons, the traditional tutorial commonly found in video games is represented in this project by the initial room encountered by the player upon starting the game (see Figure 3.4). Within this room, the player will come across a variety of interactive objects, each exhibiting distinct physics properties. This arrangement enables users to familiarize themselves with the virtual reality controls while engaging with these objects.

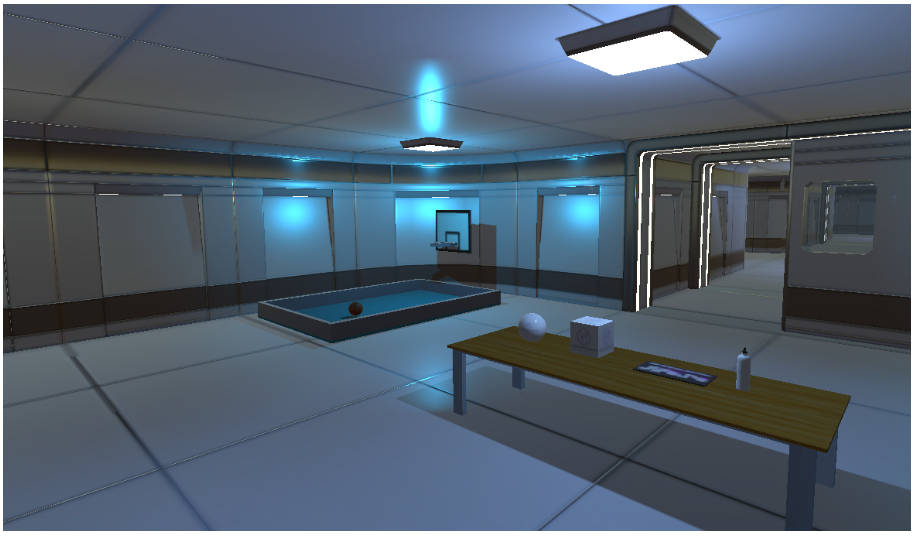

Figure 3.4: Main Menu

On the other hand, we have the game map (see Figure 3.5), which serves as a tangible object accessible to the player within the main menu. As previously described, the map functions as an interactive element, compelling the player to manipulate it in order to navigate the virtual space and locate each room mentally within the video game. Through this interaction, the player is prompted to engage in tasks involving rotation and spatial orientation.

Another significant aspect of the interface is the carefully selected lighting scheme. Within each mini-game and at the entrance of every portal, a distinct red light is employed to indicate the designated location for the player to proceed and carry out the correct task. While this element may often go unnoticed, it plays a crucial role in guiding the player's actions and ensuring a seamless progression through the game.

At the commencement of each level, a contextual instruction panel is presented (see Figure 3.6), providing the player with background information and outlining the objectives. This instructional panel is featured at the beginning of every mini-game, serving

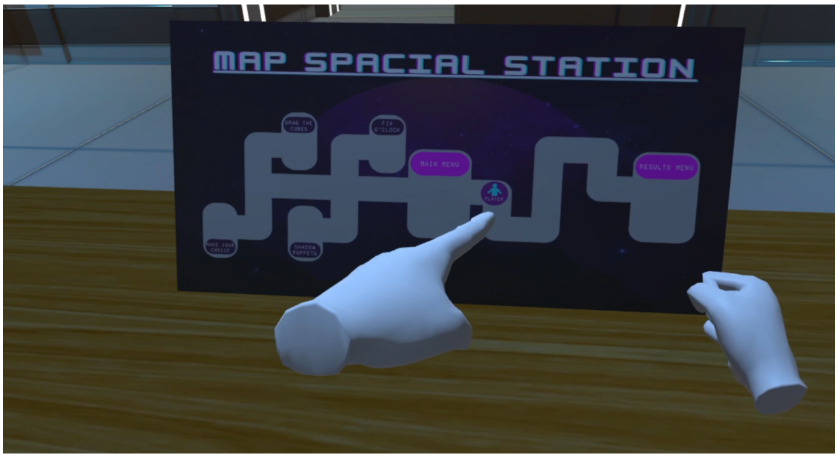

Figure 3.5: Map of the spacial station in-game

to familiarize the player with its mechanics and gameplay. Furthermore, to accommodate players who may encounter difficulties in the reading process, audio explanations of each challenge are provided by the narrator. This ensures that players can engage with the game effectively regardless of their reading abilities. Additionally, the game provides feedback to the player during each test through auditory cues, such as distinct sounds indicating success or failure, enhancing the overall game experience.

Figure 3.6: Instructions panel in mini-game 1

Upon completion of each mini-game, a floating canvas will emerge before the player, displaying numerical values representing the total count of successes and failures achieved throughout the challenge (see Figure 3.7). In addition, certain games, such as mini-games 3 and 4, incorporate a timer to establish a time constraint within which the player must complete each level. This timer serves as a guiding element, urging players to efficiently manage their time and enhance their performance (see Figure 3.8).

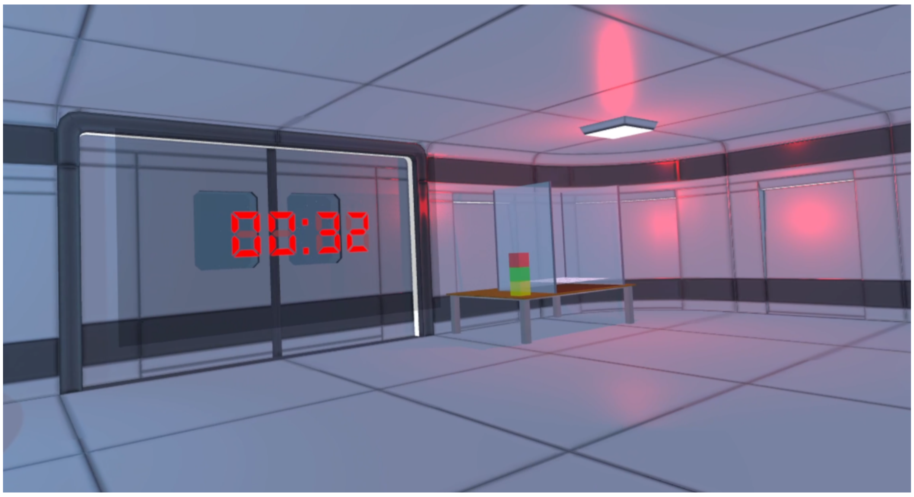

Figure 3.7: Timer in mini-game 3

Figure 3.8: Final canvas in mini-game 2

In the results room, the player is presented with the only interfaces that can be interacted with using extended reality (XR) technology. The interactions within this menu are categorized into two types. Firstly, utilizing ray-tracking of the controllers, the player can select the desired mini-game statistic to access and review their progress. By using the trigger on the controller, the player can choose one of the four screens displayed on the front canvas, prompting a floating canvas to appear on the left side, presenting the corresponding results. The second type of interaction involves a small floating panel located to the left of the player, serving as a scene switcher. Instead of having to navigate through the entire map, the player can simply bring the controller closer to the panel. Without XR technology, a physical collision between the controller and the button is detected, triggering the scene change without the need for extensive scrolling or navigation (see Figure 3.9).

Figure 3.9: Results Menu in-game

Chapter 4

Work Development and Results

Contents

This chapter serves as a detailed account of the project's evolution from its inception to its completion. It encompasses an evaluation of the results obtained and examines any alterations made to the original ideas, which may have been influenced by the author's evolving perspective or other factors that emerged throughout the process.

4.1 Work Development

This section aims to provide an overview of the project's progression, focusing on the significant aspects and decision-making processes involved. While delving into the project's development, only the most relevant elements will be discussed in detail. The acquisition of knowledge and insights crucial to the project's realization was obtained from multiple sources, including valuable contributions from Guillermo, a colleague, and educational tutorials available on YouTube. These valuable resources have been compiled in a dedicated YouTube playlist [26].

It is essential to note that the narrative of the work development section adheres to a chronological order, reflecting the sequential completion of project components. However, it is important to highlight that this order does not necessarily correspond to the accessibility of the game levels, as they function independently of one another.

The development of the application relies on the utilization of the Unity 3D video game engine (see Figure 4.1), specifically configured for compatibility with the Oculus

Rift S virtual reality headset (see Figure 4.2). The selection of the Unity 3D engine is based on our extensive familiarity and proficiency with this tool throughout our academic journey. Moreover, the decision to incorporate virtual reality technology stems from a desire to explore and expand our expertise into new technological domains. By combining our existing knowledge with innovative technologies, the ultimate objective is to create a Serious Game [9] experience.

Figure 4.1: Unity 3D game engine

Figure 4.2: VR headset Oculus Rift S

To provide readers with appropriate context, it is crucial to emphasize that this project does not fall within the genres of adventure games, role-playing games (RPGs), or any other similar categories. Rather, it is primarily an educational and therapeutic game, designed with the sole purpose of facilitating learning and well-being. Consequently, the absence of any narrative or character-driven elements should be noted.

Once the game engine and the target hardware for development, namely Virtual Reality (VR) technology, were determined, the construction of the video game commenced by systematically analyzing each step. Initially, the conceptual design of the game was meticulously undertaken, involving the creation of paper diagrams to outline the inter level communication, the identification of specific visuospatial skills addressed in each mini-game, and the formulation of the map's structure (see Figure 4.3), among numerous other aspects that will not be elaborated upon in this section.

Upon completion of the conceptual design phase, the construction of the main scene commenced, utilizing the *3D Modular Kit* [6] pack sourced from the Unity Asset Store. This assembly process yielded the initial outcome of the game, providing a visual representation of the interconnections of the mini-games within their respective rooms, as well as establishing the links between the Main Menu room and the Results Menu (see Figure 4.4).

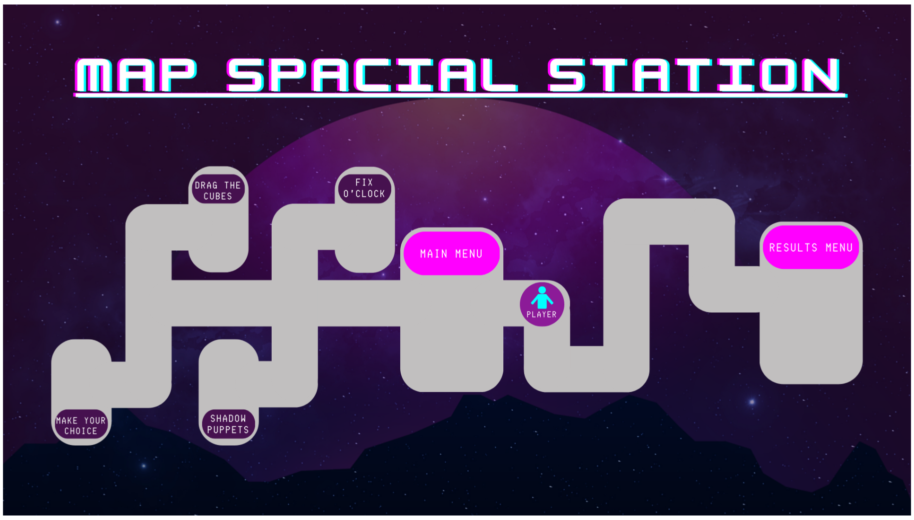

Figure 4.3: Design of the map

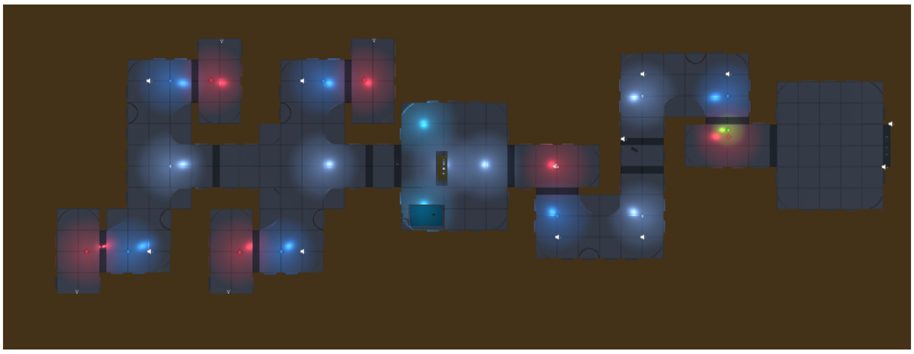

Figure 4.4: Appearance of the entire map in Unity editor

Subsequently, attention turned to the implementation of the main character controller within the game. To ensure an optimal selection, a comprehensive evaluation was conducted to identify a package that would align with the project's requirements while also being relatively easy to implement within the specified time constraints. Ultimately, the Meta Quest *Oculus Integration* [16] package was chosen for its suitability in facilitating the seamless interaction between the player and the virtual environment. The initial phase involved configuring the player's settings to ensure proper functionality. This entailed incorporating the OVRPlayerController prefab supplied by Oculus package and adjusting various parameters to enhance the player's experience. Key modifications encompassed the height adjustment, camera type and rotation speed, collider radius, and several other specifications aimed at achieving an immersive and realistic gameplay environment. To accomplish this task, guidance from a relevant YouTube tutorial, provided in the aforementioned list, was followed, resulting in the successful implementation of the main character within the game (see Figure 4.5).

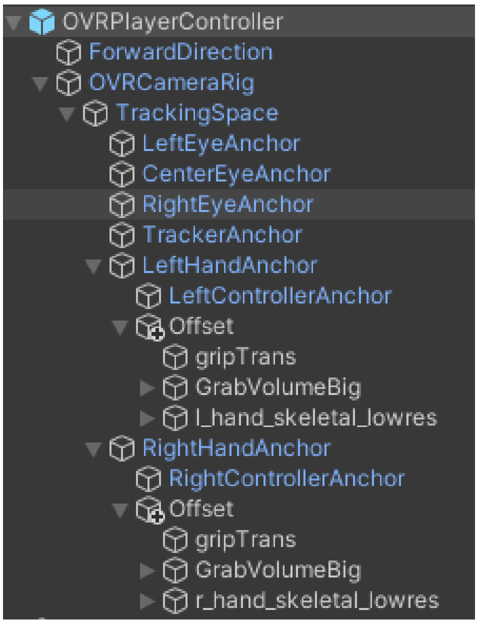

Figure 4.5: Prefab OVRPlayerController with correct order in the hierarchy of project

Upon establishing the initial framework, it became apparent that accommodating all the mini-games within a single scene would lead to suboptimal efficiency within the game. Consequently, a decision was made to partition each mini-game and the results menu into distinct scenes, thereby facilitating the optimization of asset and script processing. Subsequently, the development efforts focused on constructing a portal system and incorporating the SceneSwitcher script.

The portal system encompasses two fundamental game objects. The first object manifests as a plane, encompassing requisite materials, a collider, and the aforementioned script. The second object constitutes a pre-designed particle system prefab, specifically the *3D Games Effects Pack Free* [4] sourced from the Unity Asset Store. While the SceneSwitcher script does not boast an intricately intricate structure, it effectively capitalizes on the functionalities inherent to Unity's scene loading system. By assigning a unique index to each scene and correlating it with its corresponding portal, the script enables the seamless loading of desired scenes upon invocation (see Figure 4.6).

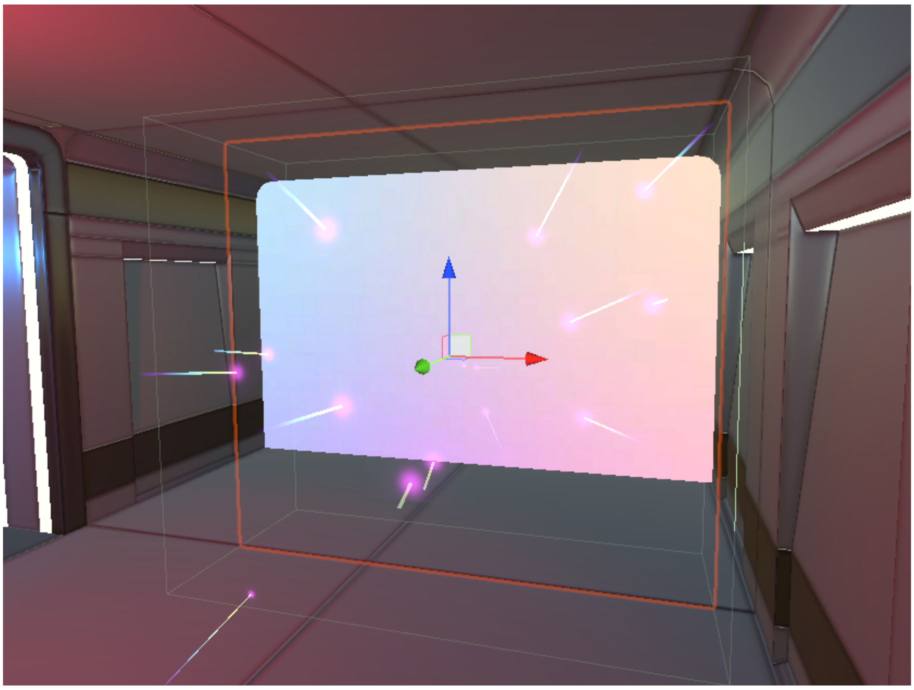

Figure 4.6: Portal system

The subsequent phase involves commencing the development of the initial scene in the game, specifically the Main Menu. As explained previously in section 3.5, this room is situated within the ship and encompasses a variety of objects designed to acquaint the player with interactive elements.

On one side of the room, there is a basketball court consisting of the floor, a basket with continuous diagonal movements that rebound off the defined boundaries, and a ball with a special bouncy material. On the other side, we find the main table, which includes a ball with unique elastic properties different from the basketball, a cube, the main map of the ship, and a bottle equipped with a collider at its base for performing flips (see Figure 4.7).

All of the mini-games have a relatively uniform design and mechanics, albeit with distinct focuses on visuospatial skills. This prompted the decision to adopt a consistent development approach while incorporating pertinent variations tailored to each game.

In the scene, each mini-game features an empty GameObject that serves as a container for its associated script manager. This script, without delving into intricate tech-

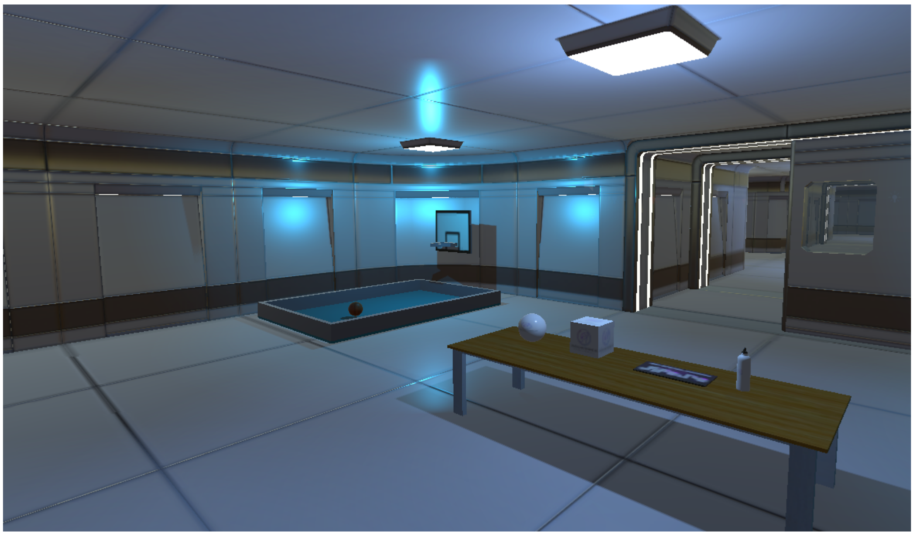

Figure 4.7: Interactable objects Main Menu

nicalities, serves as the primary control mechanism governing player-object interactions within the respective mini-game. It encompasses functionality such as managing the scoring system, tracking player successes and failures, and facilitating the computation of end-of-game scores.

Notably, the utilization of ScriptableObjects merits mention, as they were employed to instantiate levels and introduce specific variations unique to each mini-game, with the exception of the initial one which does not rely on this approach. Another shared mechanic among all mini-games is the final stage. Upon completion, the player is presented with a floating canvas that displays the obtained scores, accompanied by a particle system that enhances the visual appeal. When the player decides to conclude a mini-game, a prescribed sequence of actions is required. Namely, approaching a door to activate a collider initiates the door's opening animation. Subsequently, the player is expected to navigate into an energy ball, thereby triggering another collider that seamlessly returns them to the Main Menu.

Now that a comprehensive contextual backdrop has been provided regarding the overarching development of the mini-games, it is appropriate to delve into a detailed examination of the distinct variations characteristic of each individual game.

- **Mini-game 1: Fix O'Clock**

  The initial mini-game developed in this project is *Fix O'Clock* (see Figure 4.8), which entails determining the precise moment when a gear aligns with a designated position marked by a white shape. The player's task is to press a button to halt the gear's rotation when it reaches this specific point. Successful alignment of the gears results in a score increment, while misalignment leads to a failure. Each successful hit increases the rotational speed of the gear. The game concludes once the player achieves 10 successes. *Fix O'Clock* serves as a simple yet effective exercise to enhance patients' spatial cognition abilities. Additionally, it targets the player's inhibition, planning, and processing speed.

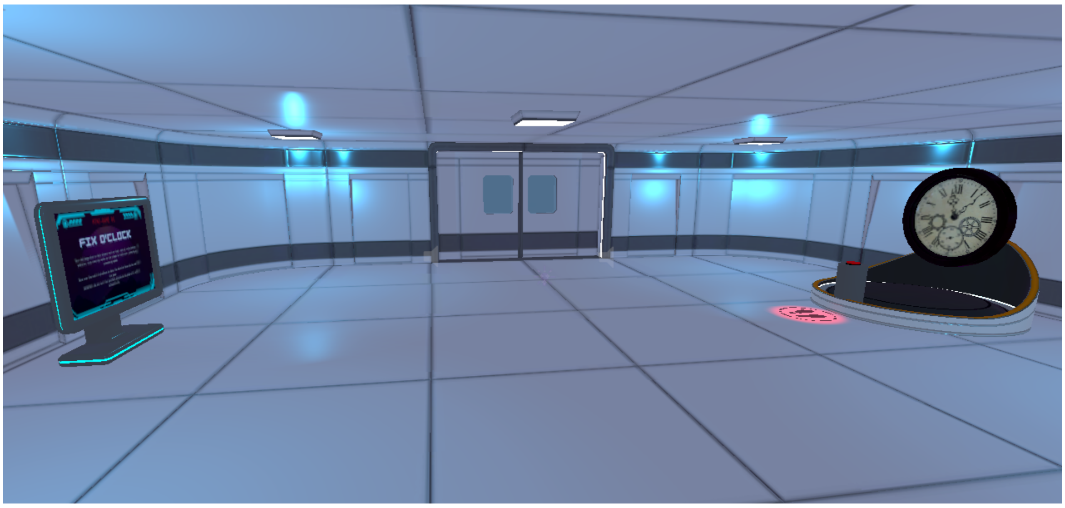

Figure 4.8: Fix O'Clock room

Unlike the other mini-games, this one does not utilize ScriptableObjects to manage its levels. This decision stems from the fact that achieving a successful hit does not significantly alter the models or game mechanics. It solely amplifies the speed of the rotating gear. Therefore, creating distinct levels for each hit would be an unnecessary complexity. After finalizing this decision, the subsequent steps involved creating the necessary elements to implement the test. Firstly, a VR button was developed with the aid of a tutorial found on YouTube. It allowed for a straightforward implementation process. Capitalizing on the button's concept, a platform was designed to initiate the game when the player positions themselves on it. Unity events were employed to trigger the start and stop of the game as the player enters and exits the platform, respectively (see Figure 4.9).

With the mechanisms in place to commence and conclude the mini-game, attention shifted towards defining the movement and collision behavior of the gears. The primary gear was programmed to execute continuous circular motion by designating it as a child object of a concealed central gear. This arrangement ensured the gear revolved perfectly around the established axis. A script governed the speed of the gear's rotation and received a reference to a GameObject representing the silhouette of the visible gear, which it should collide with. Regarding the white silhouettes, they were managed within the mini-game's general manager. Each successful hit triggers the activation of a distinct silhouette, chosen randomly to introduce variability and unpredictability (see Figure 4.10).

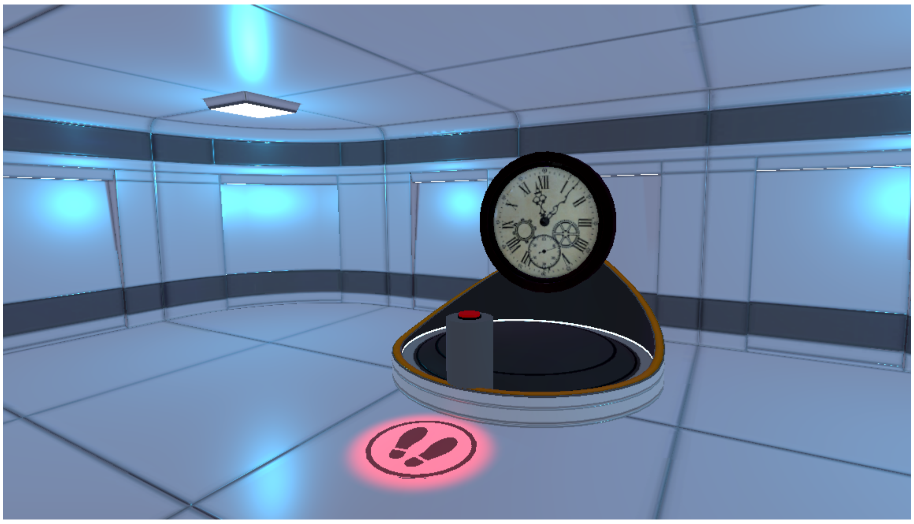

Figure 4.9: Objects that compose Fix O'Clock

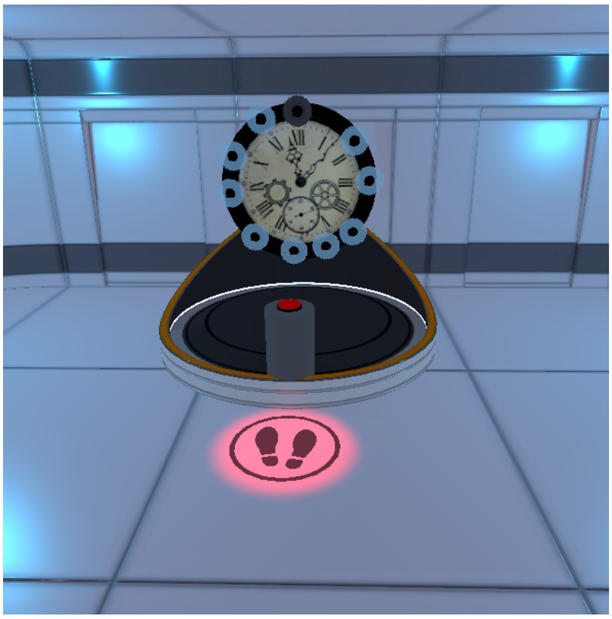

Figure 4.10: Positions shapes of gear

Upon completing the development of mini-game 1 and addressing any remaining issues, a significant omission was detected within the video game—the absence of proper audio management. This oversight became apparent in two crucial instances. Firstly, during testing of the mini-game, It experienced uncertainty in discerning whether actions were correct or not, as feedback was solely provided at the conclusion of each test. Secondly, upon transitioning to the main menu scene, the background music from the preceding mini-game continued to play. These factors, coupled with the desire for effective audio management, prompted the implementation of an AudioManager across all game scenes.

The AudioManager was introduced to facilitate seamless control over various audio elements within the video game (see Figure 4.11). This entailed managing the narrator's explanatory voice and the background music, which contributed distinctive nuances to each test. By integrating the AudioManager, the process of audio management became streamlined and uncluttered. It is crucial to highlight that the successful implementation of this manager was achieved through the aid of a YouTube tutorial, which was included within the aforementioned playlist.

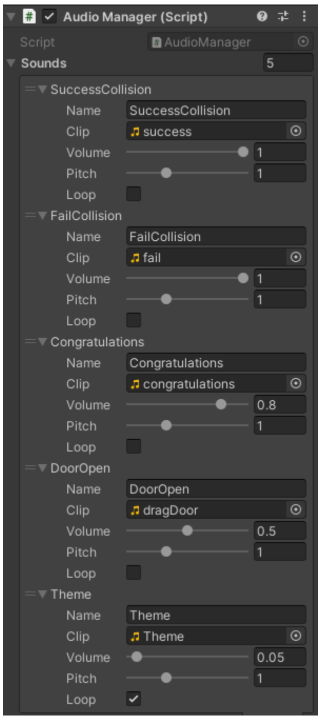

Figure 4.11: Audio Manager

- **Mini-game 2: Shadow Puppets**

  This particular mini-game (see Figure 4.12) revolves around the task of identifying the correct silhouette among the four options presented on the buttons, based on the pieces displayed in the accompanying 3D model. The player's objective is to select the button that corresponds to the shadow representation of the figure. Upon pressing a button, the mini-game provides audio feedback to indicate whether the player's selection is correct or incorrect. With a total of five levels, the complexity of comprehending the figure progressively increases. The primary focus of this mini-game is to enhance spatial rotation and visual imagery skills.

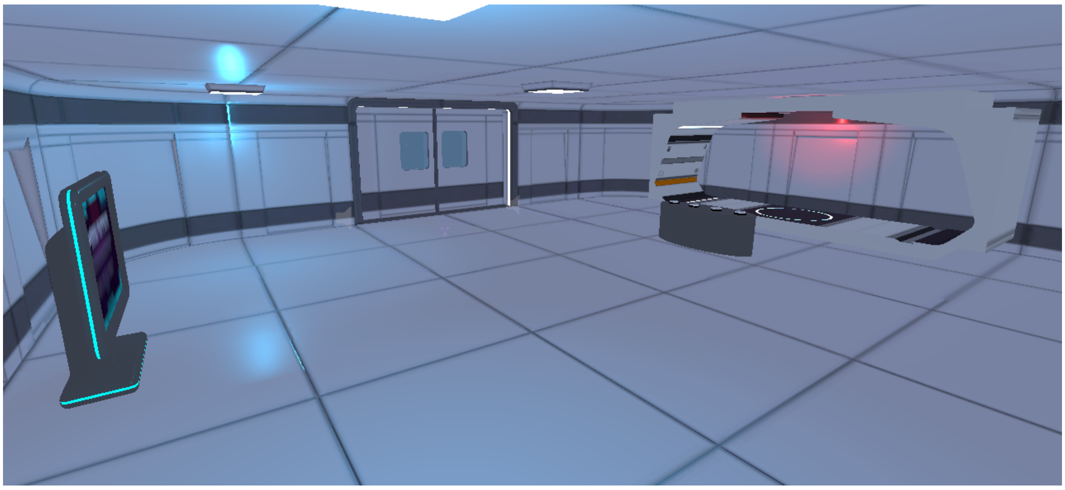

Figure 4.12: Shadow Puppets room

After providing an overview of the basic functionality of *Shadow Puppets*, let's delve into its implementation and the constituent elements. Firstly, the VR button designed in the previous mini-game is utilized with certain modifications. Specifically, a child object with a mesh renderer component has been incorporated into the collider of each button. This child object is responsible for applying the updated 2D silhouette to its corresponding button (see Figure 4.13). Each level is equipped with an array of four materials and another one that is the correct. Through this arrangement, the manager of the mini-game is able to randomize the button positions, offering the player an infinite number of combinations.

It is worth mentioning that each level in the mini-game includes the storage of a corresponding 3D model. These models are equipped with a parent transform, allowing precise positioning within the game environment. Additionally, a simple script is attached to each model, enabling a slow rotation to enhance the player's comprehension of the figure. During gameplay, as the player progresses through the levels, the current model is destroyed and a new model is instantiated to present the next challenge. This dynamic process ensures a fresh and engaging experience with each level, promoting continuous learning and adaptation (see Figure 4.14).

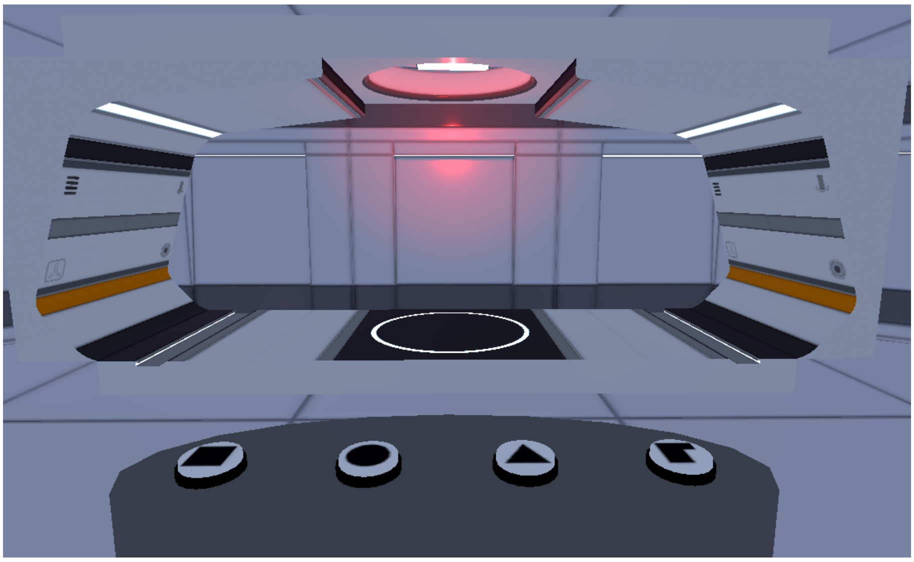

Figure 4.13: Buttons that compose Shadow Puppets first level

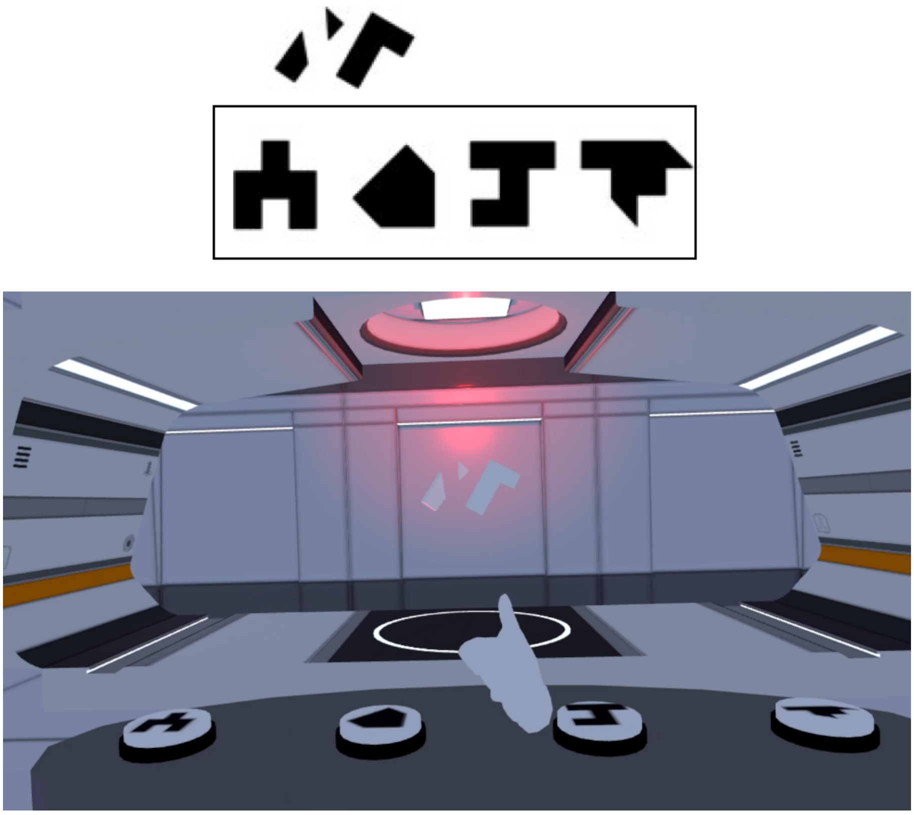

Figure 4.14: Shadow Puppets second level in-game

- **Mini-game 3: Drag The Cubes**

  Now let's proceed to discuss mini-game 3 (see Figure 4.15), which stands out as the most intricate and challenging among all. The objective of this mini-game is to mentally visualize the arrangement of a set of cubes based on a series of instructions. The player is granted a brief period to contemplate the outcome, and once they have reasoned it out, they must physically manipulate and position the cubes in the adjacent empty matrix according to their interpretation. *Drag The Cubes* comprises four levels that progressively escalate in difficulty. After the player has made the necessary adjustments to the cube positions, they must press the button again to verify if their solution is correct. It is important to note that even if the cube positions are correct, if the player exceeds the allotted time, the attempt will still be deemed unsuccessful. This test effectively enhances spatial visualization abilities and encourages patient planning skills.

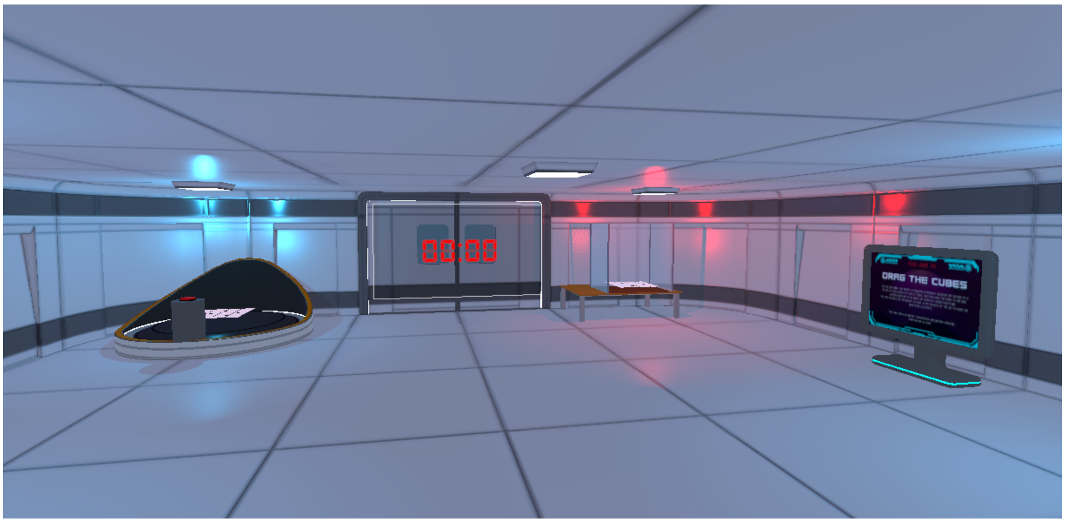

Figure 4.15: Drag The Cubes room

The mini-game comprises three primary components. On the left side, the player will encounter the instructions for each level, specifying the necessary moves to be executed. Adjacent to the instructions, there are the 3D cubes displayed along with their initial positions on the board. Finally, there is a button provided to allow the player to verify their solutions (see Figure 4.16).

The central component of the mini-game contains a countdown timer, providing the player with a visual representation of the remaining time for each level. Positioned on the right-hand side, there is a table showcasing a vertical array of cubes. These cubes are interactive objects that the player can manipulate and strategically position on the adjacent board (see Figure 4.17).

Having provided a comprehensive explanation of the different components and functionalities of the mini-game, let's delve into its development. Each level con-

Figure 4.16: Drag The Cubes first level in-game left side

Figure 4.17: Drag The Cubes first level in-game right side

sists of various elements, including an array that represents the cells of the matrix corresponding to the colored cubes, a material for the instructions, a model indicating the initial positions of the cubes, and an array for the interactive cubes. Both the correct cells and the cubes are managed through arrays, considering that the number of cubes varies across levels. Furthermore, to handle the positioning of the cubes within the board cells, a script called *Socket* has been implemented. This script enables the cubes to be placed precisely at the center of the designated cell when dropped, as well as allowing the player to pick up a cube and change its position within a cell (see Figure 4.18).

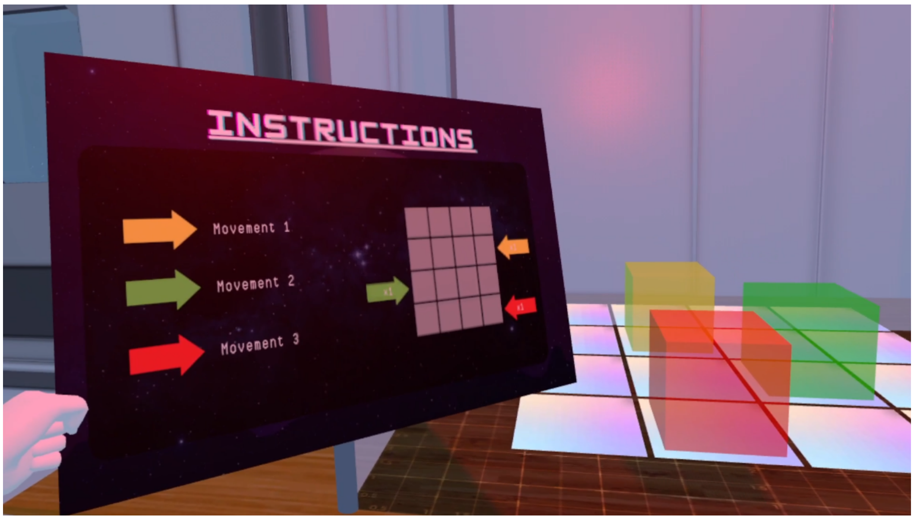

Figure 4.18: Example resolution first level

The timer implementation in the mini-game was achieved with the assistance of a tutorial found on YouTube. While following the tutorial, I made minor modifications to tailor it to the specific needs of the game. These modifications, however, were relatively insignificant and did not substantially alter the functionality of the timer.

- **Mini-game 4: Throw Your Choice**

  Within the mini-game room (see Figure 4.19), there are four recessed areas in the floor, each housing a 2D rotating geometric figure suspended in space. However, only one of these figures is the correct solution, and it is the player's task to identify and select it. To make their selection, the player must retrieve a ball and carefully place it into the recess they believe contains the correct figure. Upon successfully identifying the correct hole, doors will open, revealing a 3D object for the player to interact with and gain a deeper understanding of its geometry. *Throw Your Choice* is composed of four progressively challenging levels, and the player is given a limited amount of time to explore and manipulate the 3D model. The primary objective of this mini-game is to stimulate the player's reasoning abilities and enhance their spatial rotation skills.

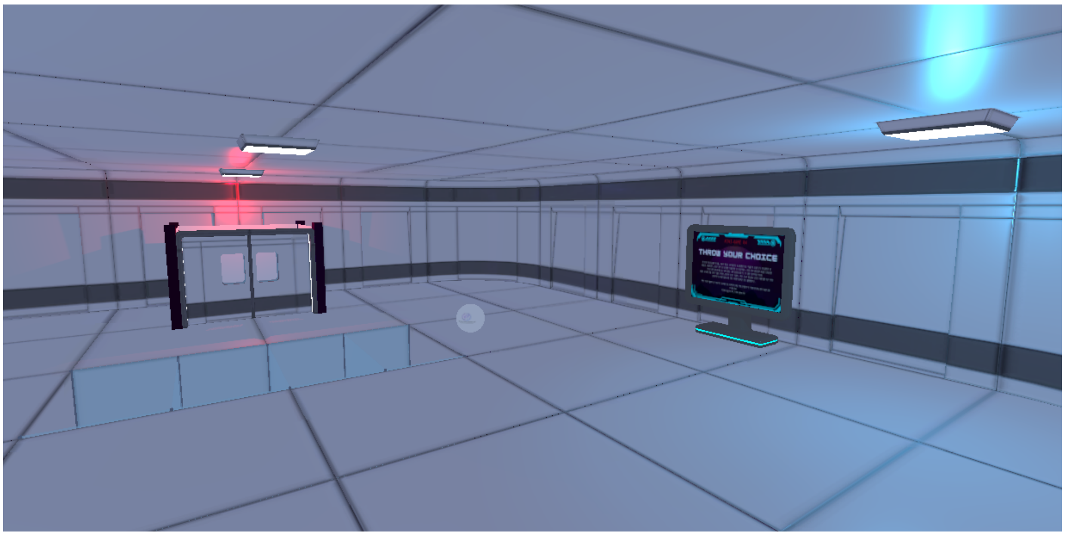

Figure 4.19: Throw Your Choice room

The mini-game consists of two primary areas that players will engage with (see Figure 4.20). The first area encompasses the holes situated on the floor, where 2D plans of various geometric figures can be observed (see Figure 4.21). These plans provide visual representations of the figures for the player to examine. The second area is the container, accessible through a door with a glass panel. From this vantage point, players must carefully observe the contents of the container, discerning the specific geometric figure contained within. The goal is to accurately identify the figure in order to make the correct choice and progress in the mini-game.

The development process for *Throw Your Choice* shares similarities with *Shadow Puppets* mini-game. Consequently, the mini-game manager for *Throw Your Choice* also receives a 3D model, an array of materials, and a single material designated as

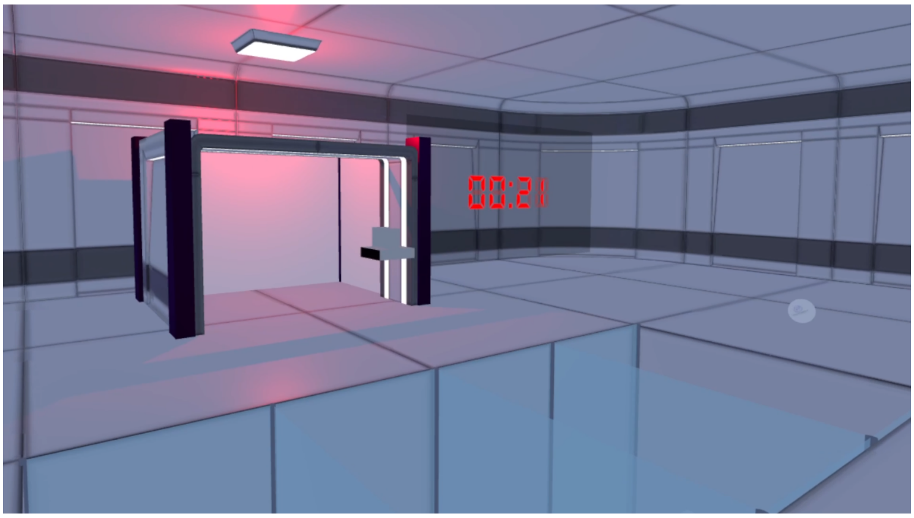

Figure 4.20: Throw Your Choice success first level in-game

Figure 4.21: Throw Your Choice first level in-game selection

the correct choice. These elements are randomized in the same manner as previously described. However, unlike mini-game 2, this mini-game does not require the player to press buttons. Instead, the player must physically pick up the ball and drop it into the correct hole (see Figure 4.22), enhancing the interactive experience. Additionally, once the correct hole is chosen, the player is able to manipulate the emerged figure, facilitating a comprehensive visuospatial exercise. As the development of the timer and the design of animations for doors and objects have been previously explained, I will omit these details in this section, considering them unnecessary for the present discussion.

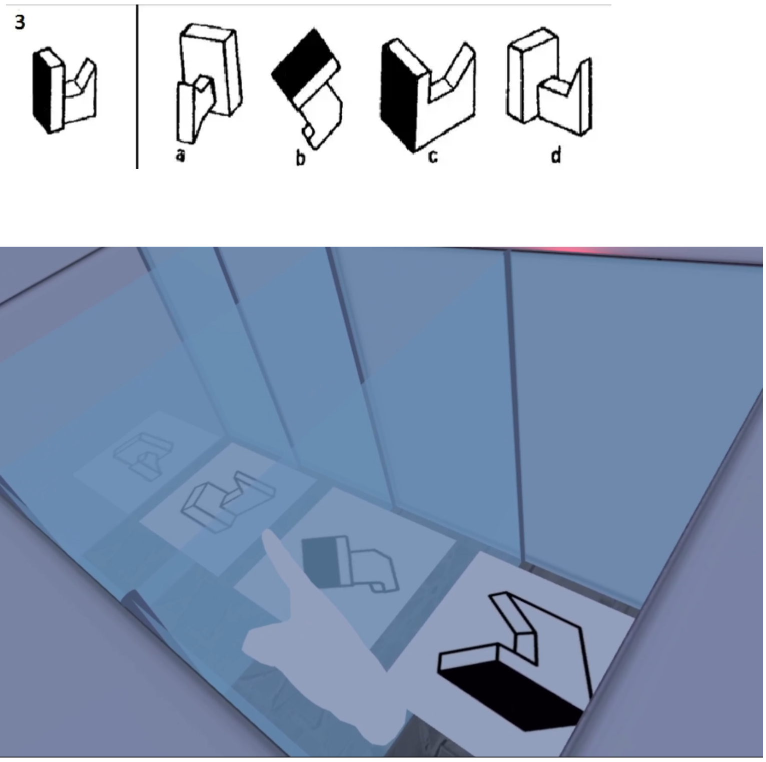

Figure 4.22: Image in-game level 4

With a thorough understanding of the unique aspects and development details of each mini-game, the *Main Menu* and the *Audio Manager*, we can now shift our focus to the development of the *Results Menu*.

The visual design of the *Results Menu* room was achieved by utilizing a pre-existing scene from the *Oculus Integration* package and making necessary modifications to align it with the desired final outcome of the project. Upon entering this room, the player's movement is restricted, as only the camera functionality was implemented while excluding the character controller to minimize unnecessary movements. With this clarification, there are two interfaces available for player interaction. The first interface consists of a compact floating panel containing buttons that can be activated by direct interaction with the player-controlled input devices (see Figure 4.23). Once the appropriate button is pressed, the player will transition to the designated scene seamlessly, facilitated by the triggering of the Unity SceneLoader event associated with the respective scene.

The second interface entails the primary floating canvas, which engages with the player through the use of rays emitted from the controllers. This canvas presents the

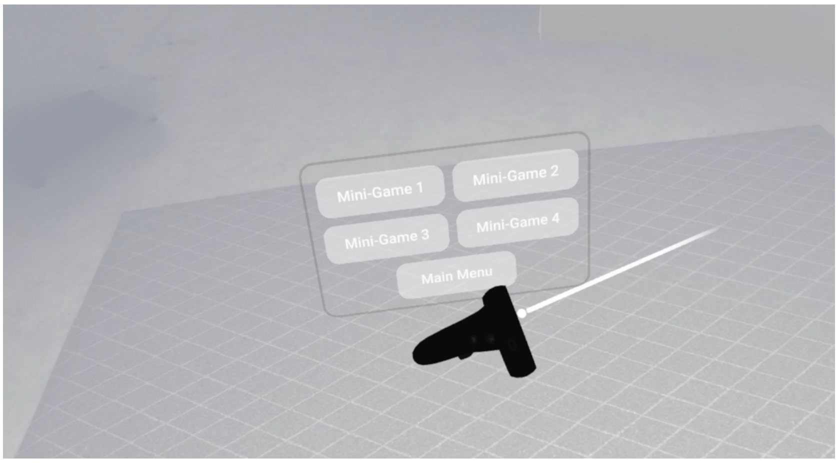

Figure 4.23: Image of floating panel

player with some information, including four toggle buttons representing each mini-game (see Figure 4.24). When the player directs the controller towards one of these buttons and presses the trigger, an additional floating canvas materializes on the right-hand side. This secondary canvas displays the scores achieved in each mini-game (see Figure 4.25).

Figure 4.24: Image of primary canvas

In order to oversee these overall scores, PlayerPrefs was utilized to store and consistently update the integer values corresponding to each mini-game, ensuring their persistence across multiple player sessions.

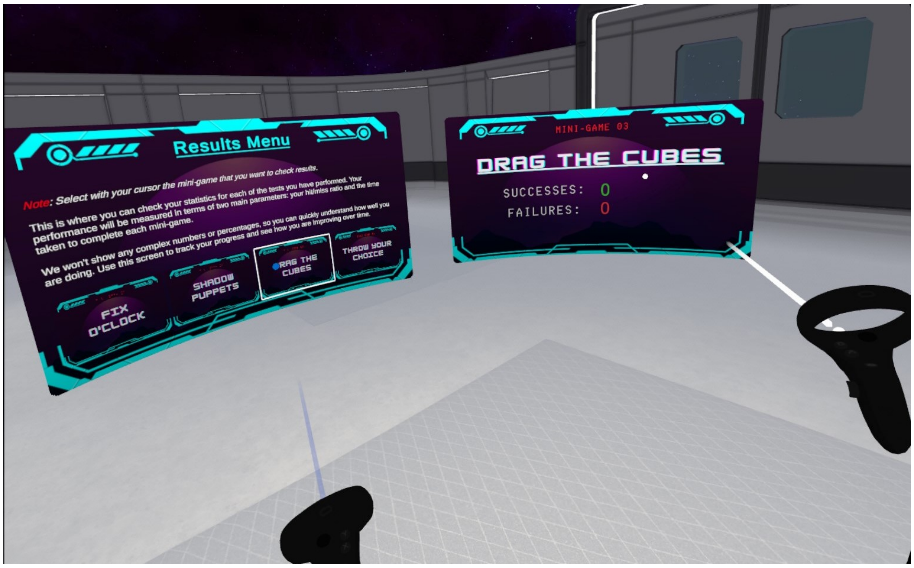

Figure 4.25: Image of punctuation canvas

In concluding this section and the entirety of the developed work, the final stages involved meticulous bug fixing and polishing. Without delving into excessive detail, visual bugs pertaining to scene lighting and the accompanying Skybox were addressed to achieve the intended lighting and ambiance. Additionally, adjustments were made to the main character, specifically fine-tuning parameters related to height, in order to establish a more authentic sense of scale and connection between the player and the virtual environment. Lastly, the build was refined and prepared, and a logo was designed to capture the distinctive essence of the project's identity (see Figure 4.26).

SPATIAL MASTERMIND VR
IMPROVING VISUOSPATIAL ABILITY

Figure 4.26: Logo design

4.2 Results

In accordance with the objectives outlined in section 1.2 of this final degree project, it can be affirmed that all of them have been successfully achieved, including certain additional objectives that were not explicitly mentioned in that section.

The primary aim of this project was to develop a virtual reality (VR) game that focuses on enhancing visuospatial skills in children and individuals who face challenges in this particular area. This objective was effectively realized through the creation of an immersive and engaging environment, where players actively interact with a variety of games and puzzles integrated within the virtual realm. Additionally, novel techniques were incorporated into educational settings, further augmenting the overall learning experience.

Additionally, the objective focused on exploring the possibility of transferring the enhanced problem-solving skills obtained through gameplay into noticeable improvements in mathematical and reading proficiencies. However, to validate this objective, an extended duration of study, a diverse range of participants, and a thorough examination of the gathered data and statistical analysis are required. As a result, the confirmation of this objective is currently pending, awaiting further investigation and evaluation.

The main objective of this study was to create a demo version that effectively showcases the accomplishment of the mentioned objectives. The demo comprises a virtual reality game consisting of four unique mini-games, each tailored to enhance crucial visuospatial skills. It is important to mention that playable build of the project, is currently accessible only on Windows operating systems. To provide a comprehensive presentation of the results, a full-length gameplay video has been included. This video offers a detailed demonstration of every aspect of the game, allowing viewers to gain a thorough understanding of its features and complexities.

The link to access the build, gameplay and source code of the project can be found in the Appendix A section.

CHAPTER 5

Conclusions and Future Work

Contents

In this chapter, the conclusions of the work, as well as its future extensions are shown.

5.1 Conclusions

Throughout the course of my degree program, I have embarked on a comprehensive journey of video game development, encompassing the entire process from conceptualizing a simple idea to the creation of a playable beta version. This educational experience has provided me with valuable opportunities to work collaboratively with different types of people, as well as different game engines, continuously learn new programming languages, explore novel tools, and gain numerous other insights and skills. In essence, I have immersed myself in emerging technologies, rapidly adapting to them within the constrained time frame of three months.

Therefore, the completion of this project has represented a twofold accomplishment for me. Firstly, it signifies the successful utilization of VR technology, despite lacking any prior professional or academic exposure to this field. It has been a momentous achievement to delve into the realm of virtual reality and leverage its potential in developing a serious game that integrates didactics, psychology, and innovative technologies. The interdisciplinary nature of this endeavor has captivated my interest and highlights the tremendous research prospects within this domain. Moreover, I take great satisfaction in having undertaken this project predominantly as an individual endeavor. Serving as both the video game designer and developer, I have demonstrated the culmination of knowledge and skills acquired throughout my degree program. This accomplishment further reinforces my readiness to contribute effectively to the field of video game development.

In conclusion, this undertaking has not only been a test of technical proficiency, but also a testament to my adaptability and passion for exploring new frontiers. It has been a fulfilling experience that has strengthened my abilities and set the stage for future ventures within the realm of game development and beyond.

5.2 Future work

Considering the restricted time frame allocated for this project, the accomplished outcome entailed a playable beta iteration of the developed video game, surpassing the predetermined expectations. Originally planned to consist of five mini-games, the project had to be scaled down to four due to time constraints.

To continue the project's development, the immediate focus would be on creating the *Maze Master* mini-game. This game entails navigating a 3D labyrinth, requiring players to strategically plan their route, maintain spatial awareness, and effectively visualize their path. By providing players with an exploded map of the maze, they can mentally map out the correct path, fostering crucial skills such as spatial visualization and planning.

An important aspect to consider for future improvement would be upgrading the hardware, as the current Oculus Rift S may be considered somewhat outdated. Transitioning to a more contemporary device, such as the Oculus Quest 2 or, if feasible within the available budget, the Apple Vision Pro, would enhance the overall experience.

In terms of accessibility, the project configuration already accommodates players who may have difficulties with reading or comprehension by incorporating audio-based instructions. However, an additional improvement could involve implementing subtitles throughout the game, ensuring a seamless experience for individuals with hearing impairments. Furthermore, as with any project, there is room for enhancing the graphics by addressing glitches, optimizing performance, and rectifying visual bugs.

Lastly, although the following proposal is somewhat aspirational, given the limitations of time and resources, it would be intriguing to explore the implementation of dynamic difficulty for each mini-game. This adaptive feature would tailor the gameplay experience to the individual capabilities of players, providing a fully personalized and guided improvement journey. This aspect holds particular significance in the field of education, as video games offer the advantage of adaptability to cater to the unique needs of each student, surpassing the limitations of traditional learning materials. While these future developments may remain speculative due to time and financial constraints, they represent intriguing avenues for further enhancing the project's potential and effectiveness.

Bibliography

[1] Audacity. Audacity | audio editor free source. https://www.audacityteam.org/.

[2] Blender. Blender | free and open source 3d creation suite. https://www.blender.org.

[3] Canva. Canva | free-to-use online graphic design tool. https://www.canva.com/.

[4] Creepy Cat. 3d games effects pack free. https://assetstore.unity.com/packages/vfx/particles/3d-games-effects-pack-free-42285.

[5] DesignEvo. Designevo | free logo creator. https://www.designevo.com/es/.

[6] Barking Dog. 3d free modular kit. https://assetstore.unity.com/packages/3d/environments/3d-free-modular-kit-85732.

[7] Germán Fabregat Llueca. Guía para la redacción de las memorias. http://mermaja.act.uji.es/itis/IS31/guia_memoria.pdf.

[8] Epic Games. Sketchfab | website used to view and share 3d content online. https://sketchfab.com.

[9] Gestionet. Serious games: what are they, examples and types? https://gestionet.net/serious-games-blog/.

[10] Google. Google documents. https://docs.google.com.

[11] Mary Hegarty. Visual-spatial ability test. https://www.sciencedirect.com/topics/medicine-and-dentistry/visual-spatial-ability-test.

[12] Krita. Krita | digital painting and illustration software based on the libraries of the kde platform. https://krita.org/es.

[13] Microsoft. Github desktop | simple collaborations from desktop. https://desktop.github.com.

[14] Microsoft. Visual studio (2019). https://visualstudio.microsoft.com/es/downloads/.

[15] NeuronUP. Neuronup | 8 actividades para rehabilitar las habilidades visoespaciales. https://www.neuronup.com/producto/#actividades.

[16] Meta Oculus. Oculus integration package. https://assetstore.unity.com/packages/tools/integration/oculus-integration-82022.

[17] Overleaf. Overleaf. https://es.overleaf.com.

[18] Meta Platforms. Oculus rift s and rift system specifications and minimum requirements. https://www.meta.com/es-es/help/quest/articles/getting-started/getting-started-with-rift/rift-s-minimum-requirements/.

[19] Meta Platforms. Oculus rift s sdk. https://developer.oculus.com/rift/?locale=es_ES.

[20] Gantt Project. Ganttproject | free project management application. https://www.ganttproject.biz.

[21] Elsa Punset Bannel. Una mochila para el universo. https://www.planetadelibros.com/libro-una-mochila-para-el-universo/65485.

[22] Philip S. Dale. Jigsaw and hidden shapes test. https://www.researchgate.net/publication/259696711_Why_do_spatial_abilities_predict_mathematical_performance.

[23] Shutterstock. Turbosquid | 3d models for professionals. https://www.turbosquid.com/.

[24] Unity Technologies. Unity (2019). https://unity.com/es.

[25] Unity Technologies. Unity asset store. https://assetstore.unity.com.

[26] Youtube. Tutorials playlist from youtube. https://youtube.com/playlist?list=PLbVz61Qpdybe4fhrHI4xoKOGqY0LncHHh.

Appendix A

Source Code and Gameplay

Given the presence of numerous scripts, it is deemed more beneficial to provide a link to the GitHub repository rather than displaying select lines of code in this section. Moreover, there is a playable build and a complete gameplay of the project uploaded in Google Drive.

-Spatial Mastermind VR Repository
-Spatial Mastermind VR Build
-Spatial Mastermind VR Gameplay