TÉCNICO LISBOA

Exploring the Impact of a Spatial Tracker on Object Localization in Video Games for Visually Impaired Individuals

Daniel de Barros Rodrigues

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Rui Filipe Fernandes Prada
Prof. Hugo Miguel Aleixo Albuquerque Nicolau

Examination Committee

Chairperson: Prof. Diogo Manuel Ribeiro Ferreira
Supervisor: Prof. Rui Filipe Fernandes Prada
Member of the Committee: Prof. Joana Carvalho Filipe de Campos

June 2023

This work was created using LaTeX typesetting language
in the Overleaf environment (www.overleaf.com).

Acknowledgments

I wish to extend my deepest gratitude to a group of special people who have made this dissertation possible.

Firstly, my heartfelt thanks go to my family for their unwavering support throughout my academic journey. To my parents and sister, thank you for your faith in me and for being there at every step of the way.

I am incredibly grateful to my close friends, who have also provided their unwavering support. André Fonseca, Filipe Valente and Tiago Almeida, your friendship has been invaluable. I reserve a special mention for Andreia Almeida and Samuel Gomes. Samuel, for his dedication and guidance, Andreia, for an unparalleled level of commitment, encouragement, insights and patience. Both of their contributions have been profoundly impactful throughout this journey.

Lastly, I extend my gratitude to my supervisors, Professors Rui Prada and Hugo Nicolau. Their guidance, expertise and understanding, especially considering my constraints as a working student, were essential in the completion of this thesis.

Thank you all for your unwavering belief, support and invaluable contributions to this significant accomplishment.

Abstract

As the use of virtual environments becomes more relevant nowadays, accessibility for individuals with visual impairments remains a significant concern. One major obstacle for these individuals is the difficulty in locating objects within virtual environments like video games.

Having this in consideration, the aim of this project is to explore the feasibility of designing a mechanism that improves object localisation for individuals with visual impairments in virtual environments using auditory and haptic stimuli as cues to assist the users. In order to validate the effectiveness of this mechanism, we developed a video game prototype using *Unity* graphic engine and game design techniques that enhance auditory and haptic perception.

Ten individuals with visual impairments participated in a three-phase summative user evaluation of the prototype. The initial phase introduced the prototype to the participants, followed by an instructive second phase where users were trained on the usage of the designed mechanism. The final phase echoed the first but now with the designed mechanism incorporated.

Upon analysis of the data garnered from these evaluations, it was observed that the developed mechanism did not yield a significant difference in the participants' ability to locate objects within the virtual environment, regardless of its presence or absence.

Keywords

Visual Impairment; Sound Localisation; Virtual Environments; Video Games; Accessibility

Resumo

À medida que o uso de ambientes virtuais se torna cada vez mais comum nos dias de hoje, a acessibilidade para indivíduos com deficiência visual continua a ser uma preocupação significativa. Um dos principais obstáculos para este grupo de indivíduos é a dificuldade em localizar objetos dentro de ambientes virtuais como os videos jogos.

Tendo isto em consideração, o objetivo deste projeto é explorar a viabilidade de desenvolver um mecanismo que seja capaz de melhorar o processo de localização de objetos para indivíduos com deficiência visual em ambientes virtuais recorrendo ao uso de estímulos sonoros e hápticos como pistas para ajudar os utilizadores. Para validar a eficácia deste mecanismo, nós desenvolvemos um protótipo utilizando o motor gráfico *Unity* e técnicas de design de jogos que aprimorem a percepção auditiva e háptica.

Dez indivíduos com deficiência visual participaram numa avaliação sumativa de três fases do protótipo. A fase inicial introduziu o protótipo aos participantes, seguida por uma segunda fase instrutiva onde os utilizadores foram treinados sobre o uso do mecanismo desenvolvido. A fase final replicou a primeira, mas desta vez com a integração do mecanismo desenvolvido.

Após a análise dos dados recolhidos destas avaliações, observou-se que o mecanismo desenvolvido não resultou numa diferença significativa na capacidade dos participantes em localizar objetos no ambiente virtual, independentemente da sua presença ou ausência.

Palavras Chave

Pessoas Invisuais; Localização Sonora; Ambientes Virtuais; Video Jogos; Acessibilidade

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

ANC Active Noise Cancelling

API Application Program Interface

AI Artificial Intelligence

BATE Beyond Accessibility To Efficiency

FT Fourier Transform

FPS First Person Shooter

HRTF Head-Related Transfer function

IFT Inverse Fourier Transform

ILD Interaural Level Difference

IVI Individuals with visual impairments

ITD Interaural Time Difference

LTI Linear Time-Invariant

PC Personal Computer

SDK Software Development Kit

SONAR Sound Navigation and Ranging

TTS Text To Speech

USB Universal Serial Bus

UART Universal Asynchronous Receiver/Transmitter

VR Virtual Reality

WHO World Health Organization

1

Introduction

Contents

In recent decades, the field of Information Technologies has seen remarkable development, as exemplified by the entertainment industry, which is supported by the great evolution in the area of virtual environments, with a greater focus on virtual reality environments and video games.[1]

This evolution results from a greater societal awareness of how these types of virtual environments can have a positive impact through their incorporation in essential areas, such as education or health [1, 2]. For example, the use of virtual environments in education can help students better understand complex concepts and engage in interactive and immersive learning experiences, leading to better learning outcomes [3]. Similarly, in the healthcare sector, virtual environments can be used to improve patient outcomes by simulating medical scenarios and procedures, providing more effective and safer training for medical professionals [4]. Overall, the potential benefits of virtual environments in various sectors are becoming increasingly recognised, leading to continued research and development in this field.

However, despite the increasing effort to make these environments more accessible, when we analyse their accessibility, it is easy to observe that there is still a long way to go in order to make this type of environment accessible to the entire population of Individuals with visual impairments (IVI) [5–8]. This lack of access can limit this group of individuals the ability to enjoy the benefits that it offers. Virtual environments, such as video games, are no exception [9, 10]. In fact, traditional video games are often inaccessible to these individuals, as they are highly dependent on visual cues. In 2002, through an analysis of the American census, it was concluded that "Popular game genres such as strategy, sports, and role-playing games lack accessible games for players with motor impairment or visual impairment", with 2% of the American population unable to play any type of game and 9% able to play but with reduced quality in terms of user experience [11].

In summary, through research and analysis of studies within this context, together with the previously acquired knowledge in the development of virtual environments, this thesis aims to develop a mechanism that facilitates access to virtual environments for IVI by trying to improve the process of locating objects in a virtual environment.

The proposed mechanism, which will be integrated into a video game prototype for validation, leverages auditory and haptic stimuli to improve user perception and interaction within a virtual environment. Its two key principles are based on delivering spatial information effectively. First, it incorporates stimuli signals to denote object presence and location. Second, it uses distance-based discretization of these signals to provide nuanced feedback about object proximity. With these principles in mind, the goal of the mechanism is to allow users to better comprehend their spatial relationship with virtual objects by modulating signal intensity with distance.

[1] https://www.businessinsider.com/video-game-industry-120-billionfuture-innovation-2019-9?op=1 (as consulted in February 2023)

1.1 Problem

There are already many promising studies in the field of virtual environment accessibility for IVI, which have demonstrated a strong potential in the ability of this group of people to make use of such environments. Lumbreras and Sánchez, for instance, demonstrated that this group of individuals have the capacity to create highly accurate mental maps of a virtual environment through the spatialization of sound known as *Spatial Audio* [12].

In another study, it was possible to observe that these individuals have a greater aptitude for sound localization tasks than those without visual impairments [13].

However, there are still challenges that need to be tackled in order to make these environments more accessible for these individuals, one of them is the spatial localisation within virtual environments like video games. Some research has shown that the use of sound can aid in spatial localisation, but object identification remains an issue. For example, visually impaired users may have difficulty identifying objects such as doors, stairs, or other navigational cues in the virtual environment [14–16]. Given this context, the work presented in this document focuses on the following problem:

It is difficult for people with visual impairments to locate and identify objects in a virtual environment.

1.2 Research Goal

In order to improve the accessibility of virtual environments like video games, it has been identified, based on previous studies and analysis, that it is necessary to develop mechanisms that facilitate the spatial localisation of objects in these types of environments. This was the basic assumption for the development of a technique called *Spatial Tracker*, whose main objective is to facilitate the localisation of objects in virtual environments for IVI. Within this context, the hypothesis presented is:

Is it possible to improve the process of locating objects in a virtual environment for people with visual impairments through the use of a *Spatial Tracker*?

For further details on the process behind this mechanism please refer to the following chapters (Chapter 3 and Chapter 4 respectively). The validation of the hypothesis was based on the analysis of user performance after the execution of experimental tests with the developed prototype.

1.3 Approach and Contributions

In order to execute this project successfully, a rigorous analysis of articles and scientific studies was required. These studies and articles encompassed several key topics like video games for IVI, sound localisation, spatial localisation, and finally, accessibility in virtual environments.

Based on the knowledge acquired from these contributions, the goal was to design and implement a video game prototype using a graphics engine where the previously mentioned mechanism of spatial tracking was incorporated.

The design of the spatial tracking mechanism was based on two core principles, both aimed at enhancing the user's perception and interaction.

The first principle incorporates auditory and haptic stimuli. These two types of stimuli were selected as they can effectively convey spatial information. Both cues provide signals to the user about the presence and location of objects.

The second principle uses the distance to a target object to discretize the intensity of stimuli signals. By adjusting the intensity of both auditory and haptic stimuli based on the user's proximity to virtual objects, the mechanism can deliver more nuanced and precise spatial information. As the object gets closer to user, the intensity of the auditory and haptic signals increases, providing a clear, intuitive indication of proximity. This gradation of stimuli based on distance intends to help users have a better understanding of their spatial relationship with virtual objects.

The choice of a video game within the virtual environments was established taking into account two criteria. Firstly, because it is a medium that facilitates learning, requires interaction and usually manages to promote player engagement. On the other hand, it is a medium that does not involve high costs, and one of the objectives of this study is to ensure accessibility, unlike other examples such as Virtual Reality (VR), where equipment is generally expensive and consequently less accessible to most of the users.

The video game prototype belongs to the First Person Shooter (FPS) genre, more specifically to the *Rail Shooter* subcategory. Additionally it was necessary to design a level that was composed of several sections with each one having distinct objectives.

After the implementation of the video game prototype, the next phase involved engaging in an iterative prototyping process with sighted users and making corresponding improvements and corrections to identified issues. Once it was ensured that the prototype was stable and there were no usability problems, we conducted experimental tests with IVI to validate the effectiveness of the developed mechanism. The effectiveness was evaluated through statistical analysis of the obtained results.

1.4 Document Structure

This document is composed of six chapters. In the introductory chapter (Chapter 1), the project's context is established, underlining the fact that despite the growing popularity of virtual environments, their accessibility remains restricted for certain individuals. Consequently, the primary focus of this work is to enhance the accessibility of virtual environments, with a particular emphasis on video games for IVI.

The second chapter (Chapter 2) presents a comprehensive analysis and discussion of the conducted research, exploring various approaches to accessibility and the development of video games for individuals with visual impairments (IVI). This chapter also reviews the state-of-the-art technology relevant to this document's theme, such as graphics engines and spatial audio frameworks, and concludes with the contributions of this research to the project's development.

The third chapter (Chapter 3) delves into the origins and detailed analysis of the spatial tracking mechanism developed and integrated into the video game prototype.

The fourth chapter (Chapter 4) centres on the conceptual ideas and design decisions that informed the prototype creation process, its implementation, and the integration of the developed mechanism. This chapter opens with a brief introduction to prototyping, followed by detailed information about each prototype component's implementation.

The fifth chapter (Chapter 5) focuses on the formative usability tests and the summative user tests conducted on the prototype incorporating the developed mechanism. This chapter details the apparatus and measures used in both evaluations and presents the participants' profiles, procedures, and results in the usability testing section. A similar structure is followed for the experimental testing, covering the participants' profiles, procedures, data collection methods, and the statistical analysis used to assess the results. Any significant results are discussed towards the end of the chapter.

Lastly, the sixth and final chapter (Chapter 6) discusses the results obtained and provides explanations for any observed outcomes. Suggestions for future work, based on these results, are also reflected upon.

2

Related Work

Contents

Accessibility in virtual environments is of great importance for creating inclusive experiences, by addressing the unique needs and challenges faced by IVI, we can ensure that the goal of this study is achieved. In order to fully comprehend these needs and challenges, it's crucial to thoroughly analyse topics such as accessibility in virtual environments, video games designed for those with visual impairments and technologies like sound and spatial localisation.

In the next section, we will begin by analysing the studies that has been done in the area of accessibility to virtual environments for IVI, especially in the field of video games.

2.1 Accessibility of Virtual Environments for Individuals with Visual Impairments

Virtual environments have become increasingly popular for entertainment, education and even therapeutic purposes. However, for IVI, accessing and navigating virtual environments can be a challenge. In recent years, there has been a growing effort to make virtual environments more accessible for IVI through the use of assistive technologies, such as audio cues and haptic feedback. In this section, solutions and challenges associated with the design and implementation of these environments for this specific audience are carefully analysed.

2.1.1 Game Accessibility for the Blind

A study was performed in order to provide a in-depth examination of the current state of game accessibility for IVI and explores the potential of audio description as a means to improve the gaming experience for this population.

Despite growing awareness and effort to make video games more accessible to IVI, there is still a significant gap in the availability of truly inclusive gaming experiences. The researchers Mangiron and Zhang (2016) [17] found that many games lack the necessary non-visual cues and interaction mechanisms that would make them accessible to players with visual impairments.

The study suggests that audio description has the potential to be a valuable tool for enhancing the accessibility of video games for IVI. Audio description provides a narrated explanation of visual elements within a game, helping to convey critical information that IVI would otherwise miss. The researchers suggest that game developers should consider factors such as the timing, content, and delivery of audio descriptions to achieve the optimal balance between accessibility and immersion.

A call for greater collaboration between game developers, accessibility experts, and the blind community in order to develop and implement effective audio description solutions. Additionally, the authors recommend the development of standardised guidelines and best practices for incorporating audio description into video games, ensuring consistency and quality across the industry.

2.1.2 Intelligent Virtual Cane

The problem with traditional virtual environments often rely heavily on visual cues, which are not accessible to visually impaired users. This limitation makes it difficult for them to navigate and interact with virtual spaces, which in turn restricts their ability to fully benefit from the opportunities that virtual environments provide for entertainment, learning, and social connection.

In order to overcome this kind of issues researchers attempted to create an intelligent virtual cane using the *Nintendo Wii* controller, which would enable visually impaired users to navigate virtual environments through non-visual means. The idea was to adapt a commonly available device, like the *Wii* controller, to serve as an assistive technology, translating the visual information of the virtual environment into haptic and auditory feedback that blind users could interpret.

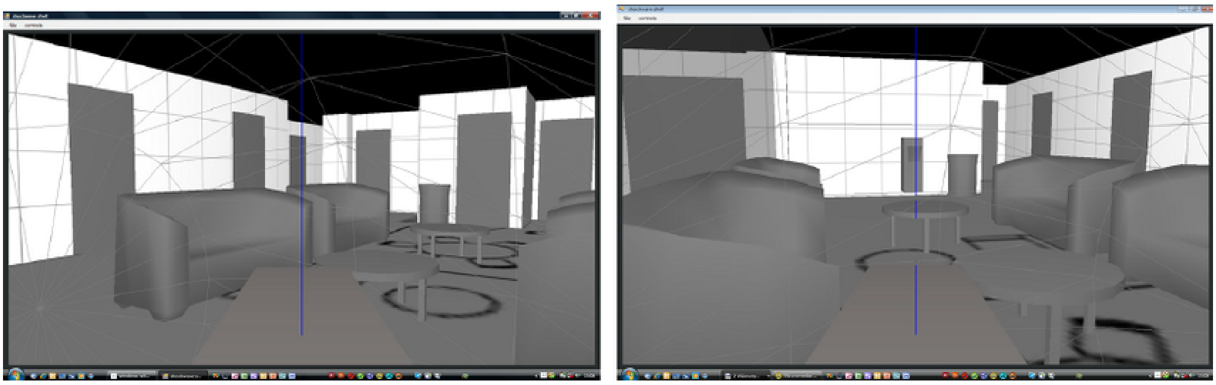


Figure 2.1: Intelligent Virtual Cane - Virtual Environment used for testing

To achieve their goal the researchers [18] designed a system that used the *Wii* controller to simulate the experience of using a traditional white cane, providing feedback on virtual objects and obstacles by using a combination of haptic feedback (vibrations) and sound localisation (auditory cues), the system aimed to convey spatial information about the virtual environment to the users, allowing them to navigate the space effectively. The study found that the intelligent virtual cane, was successful in providing an accessible means of navigating virtual environments for visually impaired users. The participants were able to use the virtual cane to detect obstacles, determine their distance, and navigate around them. Furthermore, the study highlighted the potential of combining haptic feedback and sound localisation as a means of providing spatial information to blind users in virtual environments.

2.1.3 NavStick

With the goal of enabling blind and visually impaired users to access and play video games, this study tried to address this challenge by providing these individuals with the ability to look around and explore virtual environments.

The solution proposed by the researchers [19] was to develop a system called *NavStick*, which would allow blind users to explore virtual environments by providing them the ability to "look around" using non-visual means. This system consisted of a standard game controller and a custom software interface that provided auditory and haptic feedback based on the player's actions. The system used a combination of spatial audio cues and vibration feedback to convey information about the virtual environment and the player's relative position within it. Two key elements facilitate the tool's functionality, the act of *scrubbing*, in which the player tilts the controller thumb stick circularly to survey different directions by receiving auditory feedback about what lies in that exact direction via ray casting technique. The feedback is provided using spatialized audio emanating from the identified object. The other element, is a circular data structure known as a *NavPie*, representing the player's immediate surroundings. Each slice or region of the *NavPie* represents a visible item from the player's perspective, with the slice's direction and arc length indicating the object's relative direction and angular extent.

As the player moves the thumb stick, *NavStick* announces the first object located in the given direction. The current iteration only accounts for objects directly visible from the player's position, treating the navigation system like a line-of-sight laser pointer. When players tilt or scrub in areas without objects of interest, *NavStick* can be configured to either remain silent or play a default tone, depending on the user's preference.

The study results showed that participants were able to use the *NavStick* to look around and understand the spatial layout of the environment, identifying key landmarks and obstacles. It was also demonstrated that players could effectively navigate and complete tasks within the game using the auditory and haptic feedback provided by the *NavStick* system.

2.2 Video Games for Individuals with Visual Impairments

Building on the previous discussion of accessibility in virtual environments for IVI, this section will delve into the realm of video games specifically designed for IVI. As we have observed, there are considerable challenges in making virtual environments accessible to visually impaired users. However, several efforts have been made to overcame these challenges by creating games that rely on auditory, haptic, and other non-visual cues to deliver a satisfying and engaging gaming experience.

In this section, we will analyse in detail three examples of video games for blind players: *AudioDoom*, *Blind Hero*, and *A Blind Legend*. Each of these games showcases unique approaches to creating accessible gaming experiences, employing innovative techniques and design principles to cater to the needs of IVI. By examining these games, we can better understand the potential and limitations of current accessible game design.

2.2.1 AudioDoom

This article begins with the premise that software lacks the necessary interfaces to be accessible without visual cues, thereby limiting its accessibility to groups of people who do not have access to such visual cues and depend on other types of cues, such as sound cues. The primary objective of the project was to identify how audio-based and spatial sound experiences can create spatial cognitive structures in the minds of IVI, aged between eight and eleven years old.

To enable cognitive testing in this group of individuals, a model was created to design hyperstories, which is essentially a story that occurs in a hypermedia environment (an extension of the term hypertext, which is a non-linear information medium that includes graphics, audio, video, plain text, and hyperlinks). From this model emerged *AudioDoom* [20], a prototype based on the classic game *DOOM* [1], which was used to test the ideas presented in the article through interactive hyperstories.

AudioDoom works by segmenting the navigable space into atomic environments, each containing interactive entities located in different voxels. These environments are interconnected, forming a corridor that gives a semantic and narrative connection to the game's storyline. Interaction with these entities, based on the type of entity, can be a single event or a chain of events aimed at achieving a specific goal.

The players navigate this virtual world by using a wireless ultrasonic joystick, known as *The Owl*. With this device, the player can interact with and move within different voxels. The system is designed to have a strong correlation between the haptic feedback from *The Owl* and the acoustic feedback from the game.

The testing procedure consisted of two sessions separated by two weeks, each consisting of a pre-defined set of actions: gaining confidence with the software by testing it at least five times, exploring the AudioDoom model, building the perceived structure using LEGO pieces, oral discussion about decisions and testing errors, re-exploring the model and finally repeat the process from step three (building the structure).

After analysing and discussing the results obtained from these tests, it was demonstrated that participants where able to mentally render navigable spatial structures through the use of spatialised sound.

[1] `https://slayersclub.bethesda.net/en` (as consulted in February 2023)

2.2.2 Blind Hero

The approach taken by the authors of this study is based on the ability of individuals with visual impairments to perceive sound cues. In this way, the substitution of visual stimuli for haptic stimuli is explored as a complementary means. To conduct this experiment, the authors selected the popular music rhythm game, *Guitar Hero* [2] in which players use a guitar-shaped game controller to simulate playing a guitar across numerous music songs. The game became popular for its engaging game play, which involves matching notes that scroll down the screen in time with the music.

To produce the haptic stimuli, a glove with small motors attached to the tip of each finger was developed to enable IVI to play the game, as the motors vibrated when a certain note was to be played. These motors, controlled via a Universal Serial Bus (USB) port using a USB to serial Universal Asynchronous Receiver/Transmitter (UART) interface chip set, are activated based on the game's instructions. Each motor corresponds to a button on the guitar controller, and buzzes when that particular button needs to be pressed. The software for *Blind Hero* was adjusted from the game *Frets on Fire* [3] to control when and how long each pager motor on the glove is activated. This change posed a timing challenge because players respond more quickly to visual cues than to touch-based ones. To address this, a delay was incorporated to give players enough time to react to the buzzing motors.

To test the concept, the performance of participants (including normally sighted participants) was evaluated, divided into groups of four, where three groups played *Blind Hero* while one group consisting only of normally sighted individuals played *Guitar Hero*. This was structured this way in order to follow the Beyond Accessibility To Efficiency (BATE) principle which consists in making technology accessible to individuals with disabilities, it emphasises that accessibility should not be an afterthought or an add-on, but rather an integral part of the design process from the beginning. The principle encourages designers and developers to focus on creating solutions that are not only accessible, but also usable, efficient, and effective for all users, regardless of ability.

Based on the tests performed, it was concluded that haptic perception can enhance game play and user experience for IVI in sound cues based games. The study found that haptic feedback provided a reliable and effective means for players to understand the game play mechanics and timing, and that players were able to adjust to different haptic patterns and integrate them into their game play strategies [21].

[2] https://support.activision.com/guitar-hero-live (as consulted in February 2023)
[3] https://www.fretsonfire.org/forums/ (as consulted in February 2023)

2.2.3 A Blind Legend

Blind Legend [4] is an audio game that was developed outside the context of research and was financed through a public crowdfunding campaign before being commercialised. It is considered one of the rare cases of great success in this genre and has been referenced on multiple websites dedicated to video games. The game's protagonist is aided by his daughter, who guides him through the virtual environment of the game. She provides directions on which way the player should move. The combat is based on sound cues, for instance, a specific sound cue is played to help the player identify the timing of actions such as attacking and blocking as observable in Figure 2.2 [5].

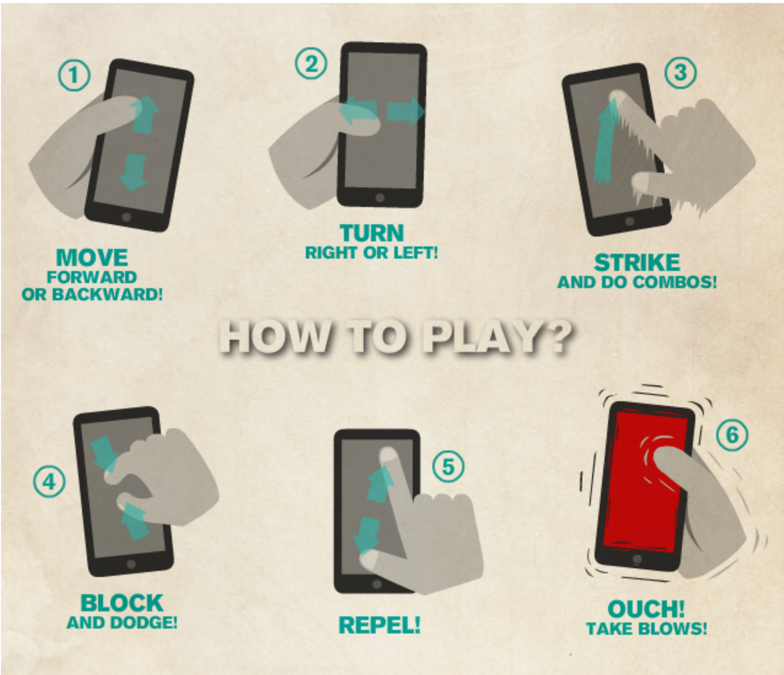


Figure 2.2: Game controls scheme

The game developers were highly skilled in integrating the interaction between the player and the virtual environment by using binaural sound along with a well-defined control scheme. *Blind Legend's* success lies in its ability to immerse the player in a completely audio-based experience, providing a unique and engaging game play experience that challenges traditional notions of video game design. The game has received widespread acclaim for its innovation and has been recognised as a pioneering example of the audio game genre.

[4] http://www.ablindlegend.com/en/home-2/ (as consulted in February 2023)
[5] https://www.ulule.com/a-blind-legend/ (as consulted in February 2023)

2.3 Graphic Engine

As we have seen, the development of accessible games for IVI involves leveraging innovative methods, such as audio cues, haptic feedback and spatial navigation to facilitate user interaction. While these strategies are effective, they only form part of the overall game development process. Another essential component lies in the underlying technology that powers these games: the graphic engines. Even though visually impaired players might not interact with games in the same visual way, graphic engines are still paramount for creating immersive and interactive game environments.

In recent times, the development of virtual environments and video games has become increasingly simplified due to the evolution and accessibility of graphic engines, whose variety and offer have gradually been increasing. *CryEngine*[6], *Unity*[7] and *Unreal Engine*[8] are considered the most popular graphic engines for developing three-dimensional virtual environments that do not require the acquisition of a professional license for software development and publication [9].

After a thorough evaluation of each engine, focusing on criteria such as ease of integration with third party spatial audio frameworks, simple management of controller motor vibration and comprehensive documentation, the following conclusions were reached, *CryEngine*, although one of the most powerful graphic engines, is also considered one of the most complex to use. Among the three graphic engines, *CryEngine* was the last to abandon the need for a professional license, which may justify the lower amount of documentation available compared to the other graphic engines. None of the main spatial audio frameworks offer exclusive Software Development Kits (SDKs) for integration with *CryEngine*. To utilise these frameworks, it would be necessary to resort to *middleware* such as *FMOD*[10] or *WWISE*[11]. Another option would be to use *CryEngine*'s own spatial audio tool *CrySpatial*, but the documentation for this is even more scarcer.

In contrast to the other graphic engines, *Unity* is the simplest to use and benefits from the existence of extensive documentation, making it one of the best-documented graphic engines and consequently one of the most used. *Unity*'s package technology simplifies the process of integrating most of the major spatial audio frameworks. A *Unity* package encapsulates a collection of assets, scripts, libraries and tools that developers can utilise to enhance their projects. To install these spatial audio packages, developers just need to download the respective package and import it into their project through the *Unity Editor*. The strength of this approach is that these spatial audio frameworks have been packaged into easy-to-install *Unity* packages. All of the major spatial audio frameworks such as *Occulus*, *Resonance Audio* and *Steam Audio* have *Unity* packages for their solutions.

[6] https://www.cryengine.com/ (as consulted in May 2023)
[7] https://unity.com/ (as consulted in May 2023)
[8] https://www.unrealengine.com/en-US (as consulted in May 2023)
[9] https://www.gamedesigning.org/career/video-game-engines/ (as consulted in February 2023)
[10] https://fmod.com/ (as consulted in February 2023)
[11] https://www.audiokinetic.com/en/products/wwise/ (as consulted in February 2023)

Although not as simple as *Unity*, *Unreal Engine* has gained notoriety and relevance over the past few years and is also well-documented, with the latest iteration, *Unreal Engine 5*, presenting impressive levels of visual fidelity. Alongside *Unity*, *Unreal Engine* supports *Resonance Audio* and *Steam Audio* frameworks via plugins. In *Unreal Engine*, plugins function as modular pieces of code that can be added to a project to extend the engine's core functionality. When a plugin like *Resonance Audio* or *Steam Audio* is installed, it allows developers to access and implement their spatial audio processing capabilities directly within *Unreal Engine*'s native development environment. However, similar to *Cry Engine*, *Unreal* only supports the *Occulus Spatializer* through *middleware*.

Having this in mind, the choice would be based on the graphic engine that is better documented and provides greater ease in the integration process of spatial audio frameworks. In this case, *Unity* and *Unreal Engine* would be the top choices, however, the decision was made to proceed with *Unity* since choosing *Unreal Engine* would ultimately require a better mastery of the *C++* language which would end up costing more time, whereas *C#* language was already mastered.

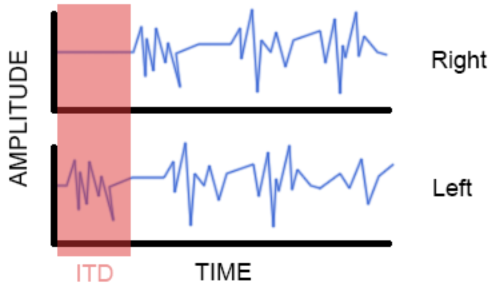
2.4 Spatial Audio

Spatial audio technology was developed to mirror the mechanisms of the human brain's sound localisation process. This process involves the interaction of sound waves with the human ears and their subsequent interpretation by our brain in the real world. By utilising this intricate information, it is possible to fabricate the illusion that sounds in a virtual environment originate from various directions. The term spatial audio is broadly used to delineate technologies striving to emulate this complex biological process.

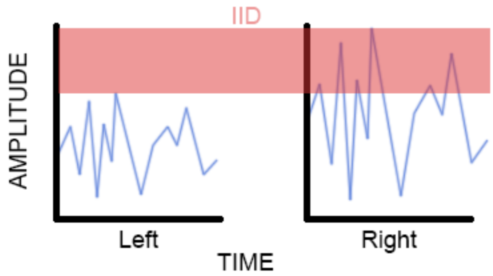
In the context of games designed for visually impaired players, the role of spatial audio becomes even more relevant. It offers a tool to help convey spatial information and enable navigation through the gaming environment, thus compensating for the lack of visual cues. By simulating the auditory cues that we naturally rely on for spatial orientation in the real world, spatial audio effectively bridges the gap between the virtual and physical world for IVI.

2.4.1 How it works

The implementation of this type of technology involves replicating several mechanisms of the human neuronal system, with the most relevant being Interaural Time Difference (ITD) Figure 2.3(a) which essentially consists of the difference in arrival time of a sound between two locations. Interaural Level Difference (ILD) Figure 2.3(b) consists of the difference in sound intensity between two locations. Lastly, Head-Related Transfer function (HRTF) Figure 2.4 consists of a function that describes how a sound located at a specific point reaches the ear. A pair of HRTF functions can be used to synthesise a binaural sound.



(a) ITD - *Interaural Time Difference*



(b) ILD - *Interaural Level Difference*

Figure 2.3: ITD and ILD Mechanisms

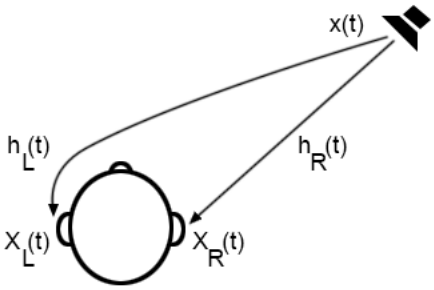


Figure 2.4: Head-Related Transfer Functions

2.4.2 Duplex Theory

The duplex theory [22], also known as the duplex theory of sound localisation, proposes that there are two mechanisms involved in sound localisation, ITD and ILD.

ITD [23] is a sound localisation cue that depends on the difference of the arrival time of a sound wave to each ear. This cue provides information about the horizontal location of the sound source which is only possible because sound waves travel through the air at a limited speed and because the two ears are separated by a non-zero distance. Due to this distance, the sound wave reaches one ear before the other, creating a time difference between the two auditory channels. The brain can interpret this time difference and use it to determine the horizontal location of the sound source. ITD is particularly useful for low-frequency sounds because their wavelength is longer, making it easier for the sound wave to wrap around the head so that it can reach each ear with slightly different times. The ITD can be approximately measured by the following formula Equation (2.1) [24]:

$$ITD \approx \left(\frac{a}{c}\right) \times \sin\ \theta \tag{2.1}$$

Where θ is the azimuth in radians $(0 \leq \theta \leq \pi)$ to the origin of the sound source, a is the distance between the ears (roughly the head diameter) and c is the speed of sound. However, it is important to note that this formula only provides an approximation of ITD and does not account for the complexities of the head shape, diffraction, and other factors that can influence ITD.

On the other hand, the ILD mechanism depends on the fact that the intensity of sound waves is attenuated during their propagation in air, so the ear closer to the sound source receives a stronger signal than the other ear. This happens as a consequence of higher-pitched sounds having shorter wavelengths, meaning that they are more easily blocked by the head and as a result, a greater intensity difference between the ears. This mechanism describes the difference in sound intensity that reaches each listener ear when a sound source is positioned at a certain angle of the head. The intensity difference can be used by the brain to determine the vertical position of sound sources (*i.e.*, whether the sound is coming from above or below the head). The following Equation (2.2) can be used to determine the ILD [25]:

$$ILD = 20 \times \log_{10}\left(\frac{Pr}{Pl}\right) \tag{2.2}$$

Where Pr is the sound pressure at the right ear, and Pl is the sound pressure at the left ear, assuming that the sound pressures at the ears are in the same units, typically Pascals (Pa).

The duplex theory suggests that the human brain combines information from both mechanisms to perceive the three-dimensional space location of the sound source.

2.4.3 Head-Related Transfer Functions

Despite the fact that the duplex theory demonstrates the crucial role of ITD and ILD in sound localisation, both can only address problems related to lateral localisation. One of the limitations of the duplex theory is the cone effect model, also known the cone of confusion [26]. In a scenario where two sound sources are positioned symmetrically to the right (as shown in the Figure 2.5), one in front and the other behind the head, both will produce the same ITD and ILD values. As an alternative, researchers have proposed the pinna filtering effect theory [27]. According to this theory, the shape of the human ear assists in shaping and filtering sound waves upon entry into the auditory canal, which can impact how sounds are perceived and localised. These modifications generate a frequency spectrum on the eardrum, which is used by the auditory nerves to identify the source of sound. This frequency spectrum can be described as a head-related transfer function HRTF [28, 29]. These functions possess a highly individual character, as the shape of the head and ears varies significantly from person to person [30, 31]. Measuring large sets of HRTF data is exceedingly time-consuming, and over the years, several methods have been suggested to compute individual HRTFs more rapidly while preserving high levels of precision [32, 33].

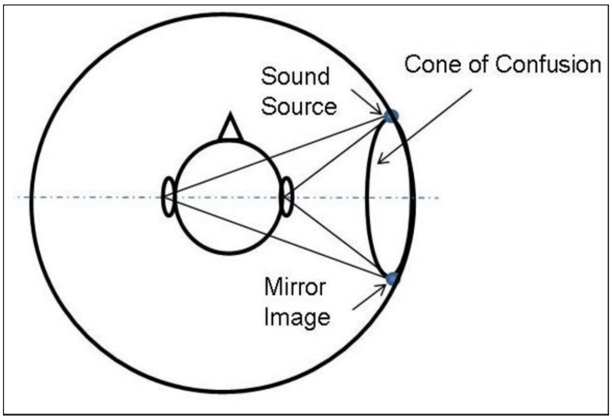


Figure 2.5: Cone of Confusion

2.4.3.A HRTF Measurement Principles

Under specific circumstances, HRTF functions can represent a Linear Time-Invariant (LTI) system that exists between a sound source at a specific point in space and a defined position in the listener's auditory channel. In other words, this applies to a static scenario. Equation (2.3) illustrates the basic principle for signal processing through an LTI system [34, 35]:

$$y(t) = x(t) \times h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)\, d\tau, \tag{2.3}$$

Where $y(t)$ is the output signal resulting from the convolution operation of the input signal $x(t)$ with the impulse response of the LTI system represented by $h(t)$. Similarly, the relationship between the output signal $\mathbf{Y}(f)$ and the input signal $\mathbf{X}(f)$ can be described as follows:

$$\mathbf{Y}(f) = \mathbf{X}(f) \cdot \mathbf{H}(f), \tag{2.4}$$

Where $\mathbf{H}(f)$ represents the transfer function of an LTI system, the transformation of a signal from time domain to frequency domain can be achieved through a Fourier Transform (FT) [34–36]. Using Equation (2.4), the transfer function of the LTI system can be effortlessly calculated by analysing the excitation signal and the system response in the frequency domain:

$$\mathbf{H}(f) = \frac{\mathbf{Y}(f)}{\mathbf{X}(f)}\ , \tag{2.5}$$

By applying the Inverse Fourier Transform (IFT), it is possible to obtain the impulse response of the system, $h(t)$.

$$h(t) = y(t) \times \textbf{IFT} \left\{ \frac{1}{\mathbf{X}(f)} \right\} \tag{2.6}$$

2.4.3.B HRTF Measurement Methods

We can identify three main categories for HRTF measurement methods [35, 37]:

- **Physical measurement methods:** Generally, these methods utilise a dummy head or a similar device to measure HRTFs. The stimulus is reproduced by a sound source at various locations around the head, and the resulting sound is recorded in the ears at a specific position. While this method is quite effective, it is also very time-consuming and requires precise calibration.

- **Numerical simulation methods:** These methods utilise mathematical models of the head and ears to simulate HRTFs. The HRTFs are calculated through numerical methods, modelling the shape and acoustic properties of the head and ears. While these methods are computationally efficient, they may not be as accurate as they cannot account for individual differences in HRTFs.

- **Psycho-acoustic methods:** These methods involve using human listeners to measure HRTFs. A stimulus is generated at different locations around the head, and the listener is requested to identify the sound's location, with HRTFs being obtained from participant responses. While this method is quick and easy to execute, it may not be very accurate for all listeners due to the limited diversity in HRTFs.

2.4.4 Application

The use of this technology for creating multimedia content has been increasing due to its ability to provide a more realistic and immersive experience for the user. In the video game industry, some producers are already incorporating this technology into their products[12][13], while others are referring to it as a selling point in their product promotion strategy[14][15]. This demonstrates its potential as a technology and the increase in popularity it has been receiving over the past few years. Several factors contribute to this growth, including the rising popularity of virtual environments and augmented reality technologies that heavily rely on spatial audio to create realistic and immersive environments. Additionally, the emergence of streaming platforms and mobile devices has made it easier to provide high-quality spatial audio content to a larger audience.

[12] https://playvalorant.com/en-us/news/game-updates/valorant-patch-notes-2-06/ (as consulted in February 2023)
[13] https://blog.counter-strike.net/index.php/2016/12/17260/ (as consulted in February 2023)
[14] https://hellblade.com/posts/development-diary-15-binaural-audio-test (as consulted in February 2023)
[15] https://tinyurl.com/gow-3daudio (as consulted in February 2023)

2.4.5 The reference APIs

After careful analysis, three Application Program Interfaces (APIs) were identified that provide the required technology without the need to acquire a license. The analysis also took into consideration the following criteria, compatibility, ease of integration, availability of examples and documentation and features relevant to the scope of this project.

- ***Oculus Spatializer***[16] is an audio plugin developed by *Meta*[17] for incorporating spatial audio into virtual reality experiences. It is compatible with major operating systems except for *Linux*, as well as game engines like *Unity* and *Unreal Engine* (through the use of *middleware*). The integration process is relatively easy, as the plugin comes with a user-friendly interface and easy-to-follow instructions. There are also comprehensive documentation and examples available on the *Oculus* developer website, which can help developers understand and use the plugin's features effectively. Some of the essential features of *Oculus Spatializer* include spatialization and audio propagation. Although it is a very solid API, it has a single problem which is the absence of the implementation of the occlusion effect that is present in the rest of the competition.

- ***Resonance Audio***[18] is an advanced spatial audio technology developed by *Google*. It allows developers to create immersive audio experiences in 3D environments. By using sophisticated digital signal processing algorithms, it can simulate real-world acoustics, making sound behave like it would in the physical world. This API is compatible with every relevant operating system and in addition, it supports game engines like *Unity* and *Unreal Engine*. Integrating *Resonance Audio* into a project is relatively straightforward, and there are numerous resources and examples available to developers. The SDK includes a comprehensive documentation library with step-by-step guides, reference materials and sample code. Some of the notable features of *Resonance Audio* include real-time spatialization, sound occlusion and obstruction and reverb modelling. However, despite the occlusion effect being implemented and calculated in real-time, it is quite basic, functioning like a switch. This means that the effect has only one intensity when active, which in certain situations impacts the user experience by seeming unrealistic.

- ***Steam Audio***[19] is another spatial audio technology that delivers high-quality 3D sound for gaming and VR experiences developed by *Valve*[20]. The software is designed to work with popular game engines like *Unity* and *Unreal Engine*, making it easy for developers to integrate into their projects. The API is compatible with all the major operating systems. However, the API documentation is limited and there are not many examples available online. Some of the features of *Steam*

[16] https://developer.oculus.com/resources/ (as consulted in February 2023)
[17] https://www.meta.com/ (as consulted in February 2023)
[18] https://resonance-audio.github.io/resonance-audio/ (as consulted in February 2023)
[19] https://valvesoftware.github.io/steam-audio/ (as consulted in February 2023)
[20] https://www.valvesoftware.com/ (as consulted in February 2023)

Audio include real-time sound propagation, which allows the API to simulate how sound waves interact with objects and environments in real time. The best feature is actually the ability to mark certain objects as ”dynamic geometry,” which causes occlusion, reverb and other acoustic effects to be recalculated as they move in the virtual space. Other features include HRTF spatialization, dynamic reverb, and occlusion. On the other hand, the worst aspect of this API is the need for the geometry system to require integration with the graphics engine.

2.5 Discussion

Through this analyses we can recognise the distinct challenges that virtual environments, particularly video games, present for IVI. We understand that the successful identification and localisation of objects within these environments rely on a multitude of variables and so, our project is set on forging a distinctive path.

In essence, the project aims to validate the proposed mechanism impact on the capacity of IVI to locate objects in virtual environments. To do this, we must ensure the user can efficiently learn to use the mechanism we develop. Taking into account the research indicating that video games can considerably enhance learning processes, we have opted to create a prototype in this type of virtual environment. However, this also introduces an array of accessibility-related challenges concerning the development of these environments, given the substantial dependence of video games on visual cues. To overcome these challenges, we plan to adopt innovative techniques to stimulate non-visual senses, utilising insights gained from the well-established works discussed throughout this chapter.

Drawing inspiration from the *NavStick* study (section 2.1.3), our proposed mechanism will make use of the potential of spatial audio technology. However, our focus will be different. Rather than focusing on navigation, as was the objective of the previous study, we aim to optimise the user's ability to identify and locate objects within the environment through the use of spatial audio technology alongside the creation of a diverse range of meticulously crafted or chosen sound effects designed to better assist users in discerning the placement of different elements within the game environment.

Similar to *Blind Hero* project (section 2.2.2), our approach also integrates haptic and audio stimuli. Nevertheless, our methodology presents a unique variation in its implementation of this combined feedback since we advocate the use of a discrete range of intensity levels for the auditory and haptic feedback signals, which are designed to be simple, discernible and easy for the user to grasp.

Regarding software, after careful examination of the most popular graphic engines and spatial audio APIs, *Unity* emerged as our primary choice due to its extensive documentation and widespread use in similar developments. In the early stage of integrating a spatial audio solution with the graphics engine, the *Resonance Audio* API was our initial choice due to its simplicity and ease of integration.

However, its limitations regarding occlusion implementation required a switch to the *Steam Audio* API during the development process, which offered superior real-time sound effect recalculations. This shift in technology and its intricate details are thoroughly discussed in Chapter 4.

3

Prototyping the Spatial Tracker Mechanism

Contents

Bateman and Boon said “Game Design is the process of coordinating the evolution of the design of a game” [38], while game development is considered the process of creating video games from a concept to the finished product, both are integrated processes and aim to create good experiences. According to Carlos Martinho and his colleagues [39], a good experience should be enjoyable and engaging for the player while providing a sense of achievement and/or mastery. The authors discuss that this is achieved through careful design of game mechanics, aesthetics and narrative. Another emphasised aspect is the importance of player feedback, interaction and play testing in refining game design to ultimately achieve a good experience. As such, the success of this project is inherently dependent on the development process, and therefore, the design phase of both the spatial tracking mechanism and the prototype is crucial to the final outcome. With an understanding of the basic requirements needed to begin the prototype development process, it was essential that the development of the mechanism took place prior to the prototype to guarantee that its development would not be constrained by any potentially restrictive decisions. Nonetheless, it’s important to bear in mind the requirements and needs of the prototype to ensure that integrating the mechanism into the prototype remains feasible. With this context in mind, the following sections will analyse in detail the various approaches that were undertaken until the final definition of the spatial tracker mechanism was achieved.

3.1 Spatial Tracker Mechanism

The knowledge acquired from the analysis and study of the works described in Chapter 2 was fundamental for the development of this mechanism. In the initial phase, the basic concept behind this mechanism arose from the idealisation of a system that stimulates both auditory and haptic senses. IVI rely heavily on these stimuli in their daily activities, making them more adept at performing tasks of this nature [20, 21]. Therefore, this system should be capable of integrating mechanisms of spatial localisation alongside mechanisms of sensory stimulation. With this idea in mind, the process of reflecting the design of the mechanism was initiated. During this process, the idea of using Sound Navigation and Ranging (SONAR) as a metaphor for the mechanism emerged. This technique utilises sound waves to locate and identify objects that are submerged underwater. The active SONAR device emits a sound wave that travels through the water until it encounters an object. Upon hitting the object, the sound wave reflects back to the SONAR device, which then detects the reflected wave. The device measures the time it takes for the sound wave to travel to the object and back, using this information to calculate the distance to the object [40]. The intention was to develop a similar behaviour to locate objects in virtual space that could be easily integrated into a future prototype, which led to the first approach.

3.1.1 First Approach: Using Ray Casting for Collision Detection

In *Alien Isolation*[1], a game from 2014 developed by Creative Assembly[2], the player has access to a tool that functions as a motion tracker as can be observed in Figure 3.1(a). This tool enables the player to locate the enemy and it operates by measuring the distance between the enemy and the player, with the tool's signal frequency increasing as the enemy gets closer to the player. Similar to the SONAR, the tool emits a signal every few seconds that reflects off the nearest objects and returns to the tracker, thus showing the enemy's location on the tool's screen. By integrating this concept with that of SONAR, the first approach to the spatial tracking mechanism emerges. The mechanism would be available to the user as an object that, similar to the sound wave of a SONAR, casts multiple rays in a 360° range every frame using the Ray casting technique. This technique involves casting a ray from a starting point in a specific direction and checking if it intersects with any objects in the scene [41]. The range of the rays would be, as shown in Figure 3.1(b):

(a) *Alien Isolation* Motion Tracker

(b) Ray casting technique

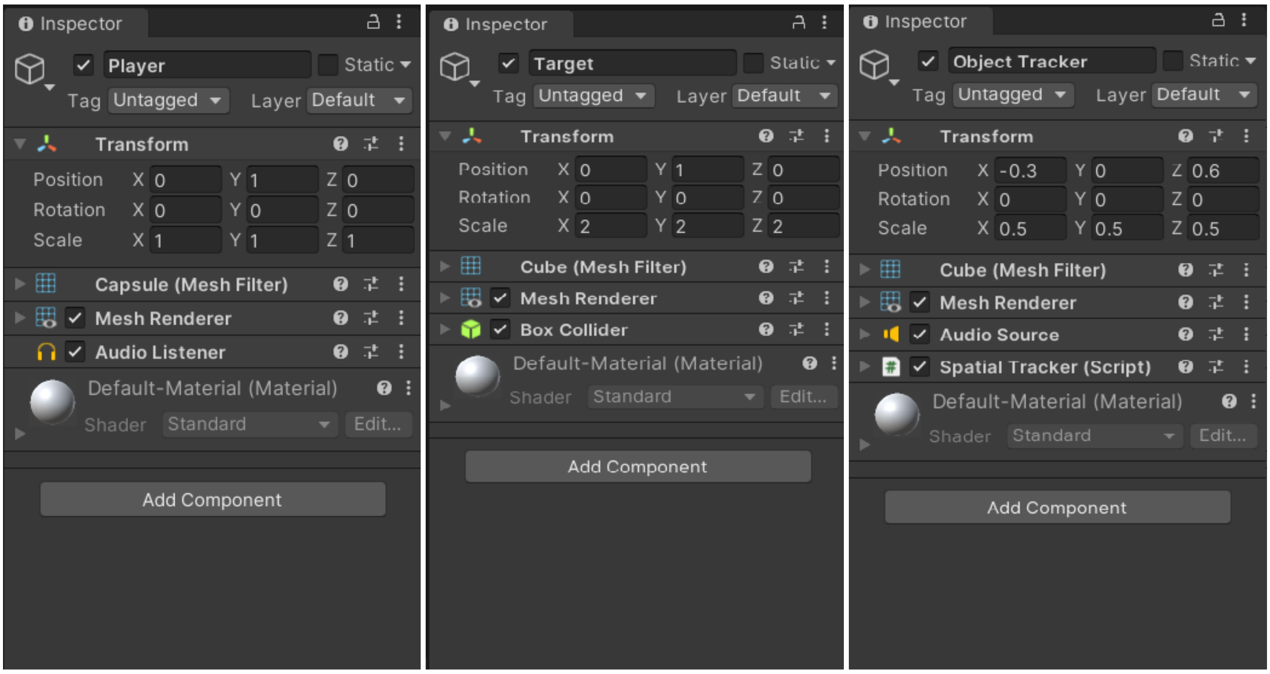
Figure 3.1: First approach using ray casting technique base on *Alien Isolation* tool

If a ray hits a target, the mechanism can determine the target's position and calculate the distance. A beep is emitted in a loop based on the distance, with the frequency increasing or decreasing if the distance increases or decreases, respectively. If the target gets too close, the beep becomes continuous instead of intermittent. If there are no targets in the scene, no sound is emitted. Similarly, the haptic stimulus is provided by the controller's motors vibration, with intensity increasing gradually as the target gets closer to the user.

To validate the idea, a simple proof of concept was made in *Unity*, consisting of a scene made up of three *GameObjects*. In *Unity*, a *GameObject*[3] can represent characters, environments, cameras, lights

[1] https://www.feralinteractive.com/en/switch-games/alienisolation/ (as consulted in February 2023)
[2] https://www.creative-assembly.com/ (as consulted in February 2023)
[3] https://docs.unity3d.com/Manual/class-GameObject.html (as consulted in February 2023)

and more. It is a container for all other *Unity* components, such as scripts that define the *GameObject*'s behaviour. Thus, in the scene, there was a *GameObject* representing the player, which had another *GameObject* as child called **ObjectTracker** and finally a *GameObject* called **Target** (refer to Figure 3.2).



(a) *Player GameObject* **(b)** *Target GameObject* **(c)** *ObjectTracker GameObject*

Figure 3.2: First Approach *GameObjects*

The **Player** *GameObject* also incorporated a component called *AudioListener*[4], responsible for capturing and outputting sound in a *Unity* scene, typically attached to a *GameObject* called *Main Camera*.

The **ObjectTracker** incorporated a component known as *AudioSource*[5], which is responsible for playing sound within the scene. This component allows various parameters such as volume, pitch and audio clip looping to be specified. As the player moved within the scene, the *AudioSource* played the configured audio clip, updating its intensity based on the distance to the target. The audio clip is set within the *AudioSource* settings and can be played, stopped, paused and manipulated in various ways using scripts or animations.

In addition to *AudioSource*, the **ObjectTracker** also included a script known as *ObjectTracker.cs* which was responsible for performing several functions, casting rays every frame to detect the **Target** *GameObject*, playing the corresponding sound stimulus (by increasing the sound intensity as the target approached) upon detection of the **Target** and enabling the haptic stimulus (by increasing the intensity of the controller motor's vibration as the target approached) upon detection of the **Target**.

[4] https://docs.unity3d.com/ScriptReference/AudioListener.html (as consulted in February 2023)
[5] https://docs.unity3d.com/Manual/class-AudioSource.html (as consulted in February 2023)

Finally, the **Target** *GameObject* included a *BoxCollider*[6], which is a 3D representation of a rectangular box used for detecting collisions with other objects within the scene. *Box Colliders* are frequently employed for static objects like walls and floors, as well as dynamic objects such as boxes and crates. When the *ObjectTracker* casts rays and they intersect the *Box Collider* of the **Target** *GameObject*, this collision is detected, leading to the reproduction of the stimuli, as detailed in Algorithm 3.1 and Algorithm 3.2.

Algorithm 3.1: ObjectTracker - *Update* method

```
begin
    for ray = 1, 2, ..., 360 do
        direction ← Calculate ray direction based on the current angle
        // verify if target was hit
        if Raycast(ObjectTracker.position, direction, out hit) then
            // if was hit update the stimuli signals
            UpdateStimuliSignal(Player.position, target.position)
```

Algorithm 3.2: ObjectTracker - *UpdateStimuliSignal* method

```
Data: Player position pp, Target Position tp
begin
    newDistance ← Distance from pp to tp
    if newDistance < previousDistance then
        // If the target is getting close update the signals
        // Increasing its intensity
        audioSource.volume ← audioSource.volume + 0.5f
        controller.vibration ← controller.vibration + 0.5f
    else
        // Otherwise decrease it
        audioSource.volume ← audioSource.volume − 0.5f
        controller.vibration ← controller.vibration − 0.5f
    previousDistance ← newDistance
```

However, following the testing of the created concept, this approach was set aside due to the inherent drawbacks of using ray casting to detect collisions. This method can be computationally intensive, particularly if multiple rays are fired simultaneously, which can lead to performance issues and slowdowns in the software. As a result, it can have a negative impact on the overall user experience. Hence, a second approach was devised.

[6] https://docs.unity3d.com/Manual/class-BoxCollider.html

3.1.2 Second Approach: Using Sphere Colliders for Collision Detection

The second approach, although based on the methodology previously presented, differs mainly in two aspects: the use of a sphere collider instead of employing the ray casting technique and the discretization of both the sound signal and vibration intensity. Replacing ray casting with sphere colliders required updating the **ObjectTracker** *GameObject* to add the *SphereCollider* component.

SphereCollider[7] is one of several types of colliders in *Unity*, representing a collision detection primitive in the form of a sphere, refer to Figure 3.3. They detect collisions between the sphere and other objects in the virtual environment. When a collision is detected, *Unity* sends a message to a method responsible to handle the collision event. In addition to being easy to configure and use, sphere colliders are versatile and can be applied to a wide variety of objects. Compared to the ray casting technique, sphere colliders can be more efficient in certain situations, particularly when detecting collisions with larger or more complex objects. This happens since sphere colliders provide a simplified representation of the object's shape, using a single point which is the centre and a radius, this simplification reduces the complexity of calculations compared to ray casting techniques, which require tracing a ray through the object's geometry. Additionally, when dealing with large or complex objects, using a single sphere collider that encompasses the entire object can be more efficient than using multiple ray casts, since this will reduce the number of collision checks required and avoids the need for complex calculations involving the object's geometry.

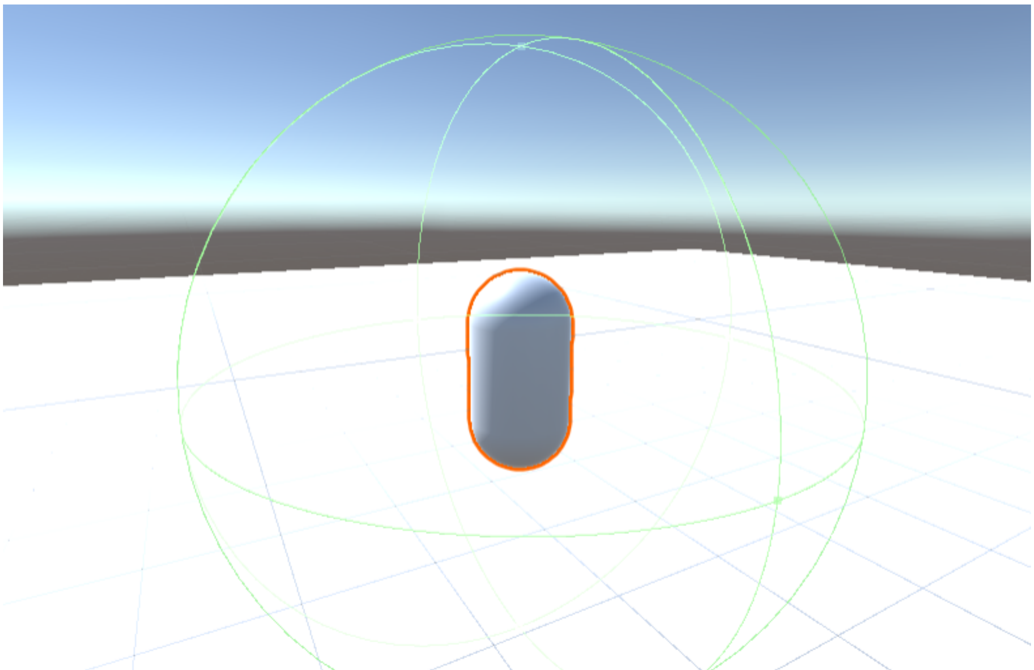


Figure 3.3: Visualisation in *Unity* Editor of the *Sphere Collider* Component

[7] https://docs.unity3d.com/ScriptReference/SphereCollider.html (as consulted in February 2023)

When integrated into the *Spatial Tracker* mechanism, the user gradually receives both stimuli as the target objects enter inside the sphere collider range. After updating the demonstration used in the previous approach (refer to Algorithm 3.3), it became apparent that it was necessary to discretize the auditory and haptic stimuli to make them easier for users to interpret. As a result, we decided to discretize the intensity of both stimuli on a scale with five levels of intensity, ranging from very low to very high (very low, low, medium, high, very high). A five-point scale is simple and easy to understand, and it can help the user better understand the range of intensities. Furthermore, the use of more intensity options allows for greater discrimination in data collection. The scale takes into account the distance from the target to the user, with each level representing a percentage range (as shown in Figure 3.4) that represents how far or close the object is to the user. Then the sound and haptic signals are updated to match the desired intensity, pseudo-code is illustrated in the following Algorithm 3.4.

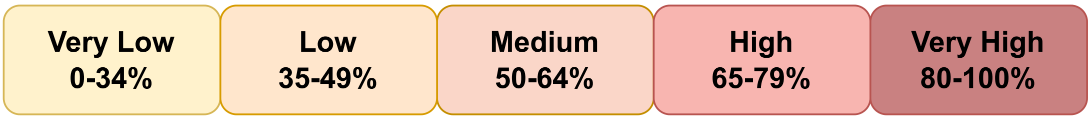


Figure 3.4: Discretion of auditory and haptic stimuli in a scale of five levels of intensity

Algorithm 3.3: ObjectTracker - *Update* method

```
begin
    // Verify if there is a intersection between the ObjectTracker and the Target
    if ObjectTracker.SphereCollider.bounds.Intersects(Target.BoxCollider) then
        // if there is update the stimuli signals
        UpdateStimuliSignal(Player.position, target.position)
```

To validate the performance of the new approach, the previous demonstration was updated and informal play-testing sessions were performed with four participants at different times and locations. Since the four participants were not visually impaired they were informed that they would need to use a blindfold during the session.

The demonstration consisted of a scene where a **Target** *GameObject* was randomly generated in one of four predefined positions. When the *Player* object was within a *distancePercentage* of less than 40%, they needed to press the space bar to destroy the **Target**. The player navigated automatically through the scene using a looping rail system in a quadrangular movement.

The goal of these play-testing sessions was to receive direct feedback from users regarding the mechanism's performance and, consequently, to understand if the mechanism helped them locate the **Target** object. No metrics were collected during these sessions, but each session was monitored. At the end of the session, participants were asked whether it had been difficult to locate the target and if they had any suggestions or ideas to present. The sessions revealed that half of the participants were unable

Algorithm 3.4: *SpatialTracker - UpdateStimuliSignal* method

```
Data: Player position pp, Target Position tp
begin
    newDistance ← Distance from pp to tp
    distancePercentage ← newDistance / distanceOnCollision
    // If the target is getting closer, update the signals
    if newDistance < previousDistance then
        previousDistance ← newDistance
        // Validate the according intensity
        if distancePercentage > 80% and distancePercentage ≤ 100% then
            signalSound ← Very Low Intensity Signal
        else
            if distancePercentage ≥ 65% and distancePercentage < 80% then
                signalSound ← Low Intensity Signal
            else
                if distancePercentage ≥ 50% and distancePercentage < 65% then
                    signalSound ← Medium Intensity Signal
                else
                    if distancePercentage ≥ 35% and distancePercentage < 50% then
                        signalSound ← High Intensity Signal
                    else
                        signalSound ← Very High Intensity Signal
```

to locate the target. The drawback of this method is that it relies on the size of the sphere collider. During the play-testing sessions, it was identified that if the radius was configured with a certain range it could be too small and the **Target** object would spawn outside of the collider range, making them invisible to the player. On the other hand, in a future scenario with multiple objects simultaneously in the scene and with a relatively larger collider radius, it could cause confusion at the stimulus level if the collider intercepts several objects at the same time.

Due to these issues, it became clear that there was a need to develop a new approach where the mechanism would not be dependent on the use of the *SphereCollider* component.

3.1.3 Final Approach: Avoid Handling Collision Detection

After conducting the play-testing sessions described in the previous approach, it became evident that there were areas for improvement and this final approach aims to solve those problems. For instance, the previously implemented discretization was insufficient in helping the user differentiate the intensity of the stimuli with the defined scale. In summary, two main issues needed to be addressed at this stage: improving the discretisation of the intensity scale for both stimuli and establishing an effective methodology for object detection in the virtual environment.

In order to improve the scale discretization, the solution adopted was to reduce the number of intensity levels to three instead of five which simplifies and turn the different levels of intensity more evident to the user, as shown in Figure 3.5.

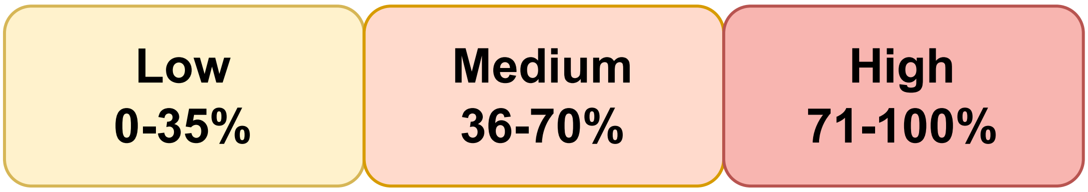
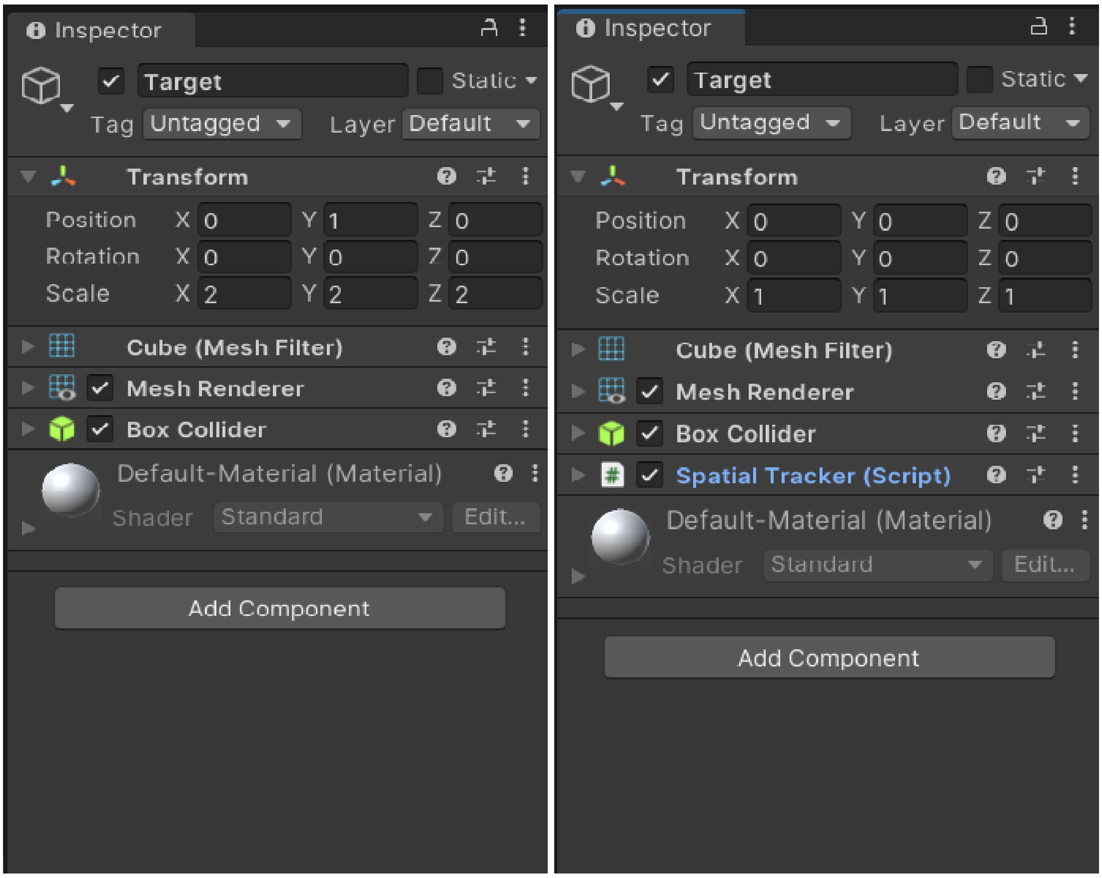


Figure 3.5: New discretisation of auditory and haptic stimuli in a scale of three levels of intensity

In regards to object detection, there were two topics in previous approaches that needed to be addressed.The first topic was the preconceived notion that the **Player** *GameObject* had to identify the other objects in the scene. The second topic was the centralisation of stimulus production in a single object, such as the **ObjectTracker** *GameObject*.

Therefore, in previous approaches, the **Player** *GameObject* served as the parent of the **Object Tracker**. The latter was responsible for detecting target objects in the virtual environment and producing auditory and haptic stimuli based on their distance from the user. However, this method had a flaw when dealing with complex scenarios that involved multiple **Target** *GameObject*. Collision management was necessary to identify the closest object, which posed a problem. The solution to this issue involved delegating the responsibility of informing the player that a new target is present within the scene to the **Target** *GameObject* itself. The only requirement for this approach is that the **Target** *GameObject* must know the user's location to calculate the distance to the **Player** *GameObject* each frame. In this way, the **Target** *GameObject*, once generated, has access to the user's position and possess a new version of the *ObjectTracker.cs* script that utilises the new logic presented in Algorithm 3.5 to activate both stimuli. On the other hand, the **ObjectTracker** is removed from the **Player** *GameObject*, and a new **TargetAudioTracker** *GameObject* is created, which has an *AudioSource* that plays the stimuli calculated by the **Target** *GameObject*s, creating the illusion that the player is detecting the objects in the scene. The changes made to the **Target** *GameObject* can be observed in Figure 3.6.



(a) Target *GameObject* (before) **(b) Target** *GameObject* (after)

Figure 3.6: Target*GameObject* before and after the update

Algorithm 3.5: *SpatialTracker - UpdateStimuliSignal* method

Data: Player position pp, Target Position tp
begin
  newDistance $\leftarrow$ Distance from pp to tp
  distancePercentage $\leftarrow \frac{newDistance}{distanceOnCollision}$
  `// If the target is getting closer, update the signals`
  if *newDistance* $<$ *previousDistance* **then**
    previousDistance $\leftarrow$ newDistance
    `// Validate the according intensity`
    if *distancePercentage* $> 70\%$ **then**
      signalSound $\leftarrow$ Low Intensity Signal
    else
      if *distancePercentage* $\geq 35\%$ *and distancePercentage* $< 70\%$ **then**
        signalSound $\leftarrow$ Medium Intensity Signal
      else
        signalSound $\leftarrow$ High Intensity Signal

4

Game Design and Implementation

Contents

After designing the *Spatial Tracker* mechanism, it was important to define certain components of the game prototype, such as the game's theme, type, actions, and others. To assist with this task, it was used at least one type of design document. We ended up choosing a *High Concept* document (appendix A), which is commonly used in the game design field[1 2] [42]. This document can help transform vague or incomplete ideas into well-defined plans and focus on the core elements of the game. Additionally, it facilitates the detection of design errors, which are easier to correct before any code implementation. Having this in mind, in the next sections will be described the establishment of the game world, the creation of the *High Concept* document and every relevant step of the prototype development process.

4.1 Game World

The presence of a story and a cohesive universe can determine the quality of a player's experience in a game. A study published in the scientific journal "Games for Health Journal" in 2012 found that a well-developed and engaging video game narrative helps players suspend disbelief and reduce counterarguments, enable the story experience as a personal experience, and create players' affection for characters [43], another study suggests that transportation into a narrative world is a key factor in media enjoyment. When people become immersed in a story, they are more likely to experience pleasure and satisfaction [44, 45]. Therefore, to increase the engagement with this work we've developed a a game world set in a distant future where a race called the *Nauds* invades planet Earth once again. The player assumes the role of a super soldier who suffered injuries that left him blind during the last *Naud* invasion but, thanks to technological advancements, can now fight the enemies and save humanity.

4.2 High Concept

The *High Concept* document encapsulates the core ideas of the game in a concise document (up to four pages). It begins with a captivating sentence summarising the game idea and includes descriptions of gameplay, game flow and environment. Its objective is to communicate the game's enjoyment factor through easily readable content. Topics include the game's premise, design goals, target audience, genre, selling points and target platform.

Through *High Concept*, it was possible to establish that the action genre would be the best option for this project since it is a genre that helps improve the learning process [46]. It was also established that within the variety of sub-genres, the *Rail Shooter* genre was selected for the prototype, using an FPS perspective. This genre consists of a type of video game where the player follows a predetermined

[1] https://www.gamasutra.com/blogs/ErinRobinson/20141124/231401/Gravity_Ghost_Crafting_a_HighConcept_Document.php) (as consulted in February 2023)

[2] https://blog.unity.com/products/speed-limit-a-game-dev-diary) (as consulted in February 2023)

path while facing enemies or performing actions that appear in front of them or in specific areas in the game environment. A popular example of a *Rail Shooter* is *Sega's* classic "*House of the Dead*" [3]. The following two aspects were taken into account in making this decision:

- **Character movement -** When choosing this genre, the player's navigation in the game environment was not a concern since there was no need to develop mechanisms to allow navigation for IVI. This would be a complex process since the focus of this project is on locating objects in virtual space rather than navigating it.
- **Discretization of interaction areas -** Using the game "*Blind Hero*" (section 2.2.2) as reference, where musical notes are generated in five distinct sections and approach the player until they reach a specific area where the player must press the corresponding note on the controller to obtain points. In the context of this project, the same can be done to clearly define three distinct sections (as shown in Figure 4.1) where the objects which the player can interact with, will be generated. These sections were named "Lanes". This way, when enemies or health kits and ammunition are generated, they "slide" to the player's position, similar to what happens with notes in *Blind Hero*. While objects (enemies, health kits, etc.) slide towards the player, looped sound effects are played. In combination with the spatial audio framework, these sounds assist in distinguishing the object and its position in the scene.

Figure 4.1: Lanes layout

In summary, it was determined that from a first-person perspective, the player wakes up in their recently attacked space station where they have to eliminate all the *Nauds* that appear in front of them in order to achieve the final objective, which is to complete the level designed for this prototype.

The player is equipped with a tracker (*Spatial Tracker* mechanism) designed to help them understand

[3] https://hod.sega.jp/ (as consulted in February 2023)

the location of the *Nauds* in the scene, that is, in which lane they are situated. The player can also collect a weapon, life and ammunition that appear in the lanes to survive the *Nauds*' attack.

The game prototype comprises a level with four sections, each with its own objective (see section 4.3.4 more details). Upon completing the challenges of a section, the player automatically advances to the next section through a gate. The following two subsections will examine the actions and challenges the player faces in this prototype.

4.2.1 Player Actions

It was defined that the actions a player could execute in the game prototype include collecting health kits, picking up ammunition, firing and reloading a equipped weapon. To perform the described actions, the player simply has to press the button assigned for each action, as outlined in section 4.3.4.

4.2.2 Player Challenges

As mentioned earlier, each section has different objectives, thereby posing unique challenges for the player. Throughout the level, the player will be tasked with identifying and collecting a certain number of health kits, identifying and collecting a specific amount of ammunition, overcoming a target shooting challenge, and finally, eliminating all the *Nauds* that are attacking their space station in order to complete the level. When firing the weapon, the player must consider the weapon's range, as it only reaches enemies up to a medium distance. If the enemies are further away, they won't be hit by the shot (fig. 4.1).

4.3 Prototype Development

With the completion of the prototyping of the *Spatial Tracker* mechanism and the establishment of the basic concepts for the prototype, we can now move on to the implementation phase of the prototype. The prototype is a limited representation of a design concept. According to Jesse Schell, "prototyping is an iterative process in which preliminary models or versions of a product or system are created with the aim of validating ideas, functionalities, and user interaction flows". It is a widely used technique in the development of games and other iterative projects [47]. Prototypes are used to obtain users feedback, identify usability problems and other design issues, as well as communicate ideas and concepts.

The development of this project was a lengthy process, with most of the development work being done in October 2022. Usability tests were conducted in the same month, which contributed to the establishment of the final version. Throughout the development process, several informal play-testing sessions were conducted with blindfolded individuals, as they provided quick feedback and allowed for constant improvement of the project despite not belonging to the group of IVI. The development of the project was divided into the following phases: audio feedback system, player, collectable objects, controller scheme, lanes (which include the object spawner and picker), enemy, and finally, level design. The next sections will detail the development process of each of these phases, as well as the tools used and the decision-making process.

4.3.1 Audio Feedback System

It should be noted that this project relies heavily on auditory stimuli, demanding the use of a wide variety of audio clips. Some of these clips were provided by open source platforms, while others were custom-crafted using *Audacity* software[4].

To establish the audio feedback system, it was determined that each object emitting audio feedback (such as the player, enemies, health kit, etc.) should be structured with an **Audio** *GameObject*. This *GameObject* would be composed of child *GameObjects*, each representing a distinct sound effect, for example, the **Footsteps** *GameObject* for player footstep sounds (fig. 4.3). Each of these *GameObjects* is equipped with an *AudioSource* component that outputs to an *AudioMixer*[5]. The *AudioMixer* facilitates the adjustment of various audio elements like volume, pitch, and other effects. It constructs audio structures by directing and grouping sources to specified buses, applying effects, and then routing them to the game's output, as can be seen in Figure 4.2. Each **Audio** child *GameObject* includes an *AudioController.cs* script with methods such as *PlayAudio()* and *StopAudio()*, among other methods which may vary between *GameObjects*. For instance, another common method in these *GameObjects* is responsible for selecting an audio clip, sometimes randomly, from an array of configured audio clips. Additionally, each child usually possesses a *ResonanceAudioSource* component for spatial audio features like HRTF, which assist the player in identifying the location of the sound and consequently help determining the lane the objects are in.

Figure 4.2: *AudioMixer* composition for this project

[4]https://www.audacityteam.org/ (as consulted in February 2023)
[5]https://docs.unity3d.com/Manual/AudioMixer.html (as consulted in February 2023)

4.3.2 Player

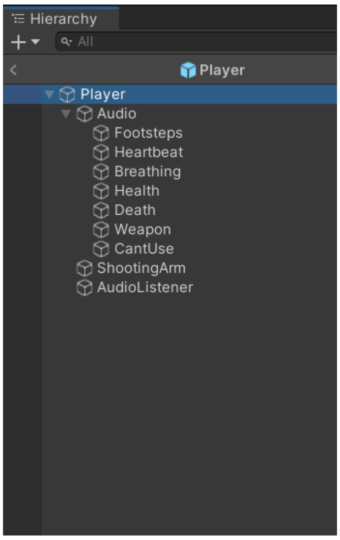


Figure 4.3: Player *GameObject prefab* structure

In this section, we address the development process of the Player. The **Player** is represented by a *Prefab*[6] that contains child *GameObjects* (refer to Figure 4.3). A *prefab* is an object that serves as a template for other objects. It is used to instantiate objects that have the same properties and components defined by the *prefab*, allowing for the reuse of elements in different parts of the project[7].

A script called *PlayerController.cs* was created and is responsible for having informative and control variables, such as whether the player has gained or lost life, and interaction methods directly assigned to the input scheme, such as picking up objects or shooting weapons.

To achieve the final version, we created a health system that includes mechanisms for heart rate, breathing, life recovery and death. Furthermore, mechanics for footstep sounds and weapon handling were also integrated.

4.3.2.A Footsteps

The *GameObject* **Footsteps** follows the structure mentioned in section 4.3.1. This mechanic was developed using four sound clips for each foot, totalling eight distinct sound clips. These sounds were generated by manipulating open source audio. Essentially, the method implemented in *FootstepsAudioController.cs* alternately plays the sound clip for the footsteps, if starting with the left foot then it switches to the right foot and vice versa. This approach produces a more natural and less mechanical sound effect.

[6] https://docs.unity3d.com/Manual/Prefabs.html (as consulted in February 2023)
[7] https://www.gamasutra.com/blogs/MarkFerrari/20140722/221715/Prefabs.php (as consulted in February 2023)

4.3.2.B Health System

The health system comprises four mechanisms: heart rate, breathing, life recovery and death. It was established that the player's life will be represented by an integer value ranging from 0 to 100, where zero denotes the player's death and 100 means the player is in perfect health. An enumerated list of states was created to indicate the player's health status composed by the following elements *Damage*, *PickUp*, *Recover* and *None*. This allows for the corresponding sound to be played when the health status changes.

Therefore, two new *GameObjects* Heartbeat and Breathing were created as children of the **Audio** *GameObject*. For both mechanisms, three distinct audio clips have been used, which will be played according to the player's current life value, as observable in Figure 4.4. For instance, one of the audio clips represents a low heart rate, while another represents an intermediate heart rate, with the remaining representing a high heart rate. The same logic applies to the breathing mechanism. The idea behind these two mechanisms is to allow the player to perceive whether they are in good health or at risk of dying.

The concept behind the life recovery mechanism, is to play sound effects when the player gains or loses life (refer to Section 4.3.3 for more details). If the player recovers their health, a relief sound effect is played. If they lose health, a suffering sound effect is played. If the player loses life and has a value equal to or less than 0, then the player dies and the corresponding sound effect (that sounds like agony) is played by the **Death** *GameObject*.

Unlike the other children of the **Audio** *GameObject*, all the *GameObjects* belonging to the health system do not have the *ResonanceAudioSource* component because, in this particular case, it was not intended to have the functionality of spatial audio but rather to create the sensation that all sound effects are experienced in the first person perspective.

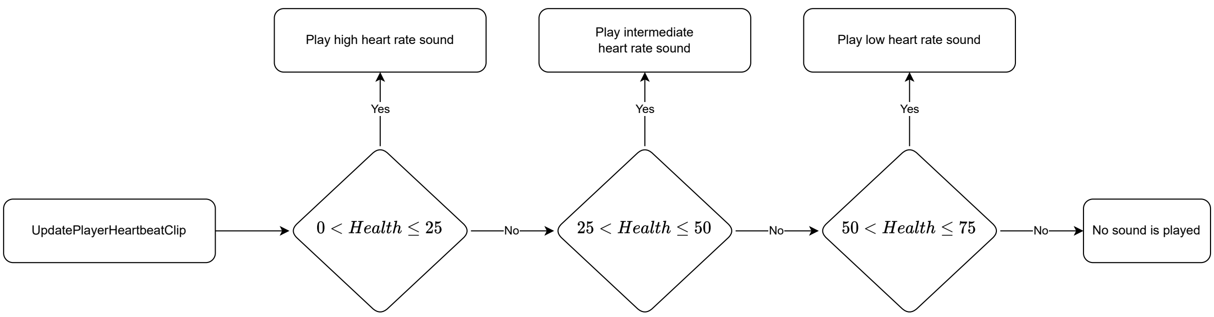


Figure 4.4: Heart rate sound stimuli flow diagram

4.3.2.C Weapon

This *GameObject* was developed to manage the sound effects related to the weapon equipped by the player. For instance, if the player shoots or reloads the weapon, the respective sound effect is played.

4.3.3 Pickable Objects

With the **Player** complete, it was time to implement the objects that the player can pick up in the scene, namely health kits, ammunition and a pistol.

4.3.3.A Health Kit and Ammunition

A *prefab* was created for both object with a *ResonanceAudioSource* component attached.

The **Health Kit** *prefab* contains a script named *HealthKitController.cs* with a method named *GainLife()*, which is executed whenever the player picks up a health kit. This method adds 20 life points to the player, informs *PlayerController.cs* that the health status has changed and prompts the *PlayerAudioController.cs*, which in turn triggers the *HealthKitAudioController.cs* to play the corresponding sound effect for the *Recover* state. Ultimately, the method self-destructs the *prefab* (refer to Figure 4.5).

For the **Ammunition**, a script called *AmmoController.cs* was created where the main method *ReloadWeapon()* verifies if the player has a weapon equipped, then it increases the number of bullets to 15 units, regardless of the current number of bullets in the weapon, and subsequently triggers the *AmmoAudioController.cs* to play the reload sound effect.

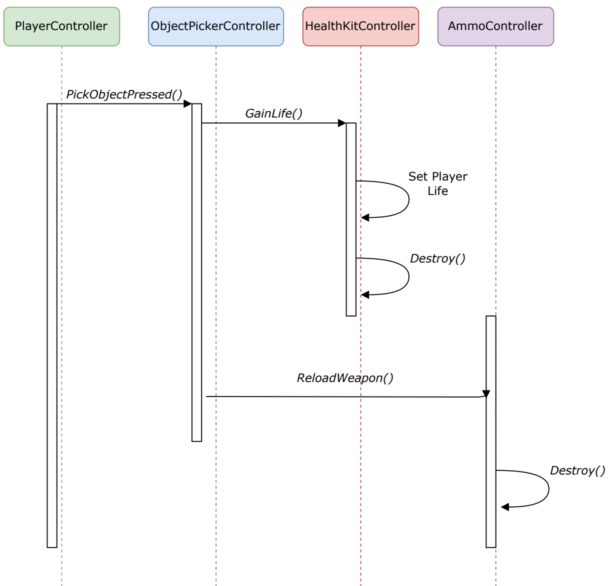


Figure 4.5: Picking Objects flow diagram

4.3.3.B Pistol

The pistol posed one of the most significant challenges during implementation. The mechanics of shooting and killing enemies were crucial for the prototype, as it was imperative for the IVI to be able to clearly identify where they were shooting.

Initially, the plan was to use the **ShootingArm** *GameObject* to shoot in three different directions left, right and front. However, this approach required precise timing to hit enemies sliding towards the player, which could lead to potential frustration.

Therefore, a different approach was adopted, which involves having, in addition to the **Pistol** *prefab*, three **Invisible Gun** *prefabs*, one for each lane. Upon instantiating the **Pistol**, three **Invisible Gun** *prefabs* are also instantiated, each in a specific position in their respective lanes. The idea is that when a player fires the pistol, the *ray cast* and sound effect are not produced by the pistol itself, but by the invisible gun corresponding to the lane in which the player intends to shoot. For example, if the player shoots to the right, the *ray cast* and the respective sound effect are activated by the invisible gun placed to the right of the player. The method *ShootPressed()* triggers a sequence of internal triggers, as can be observed in fig. 4.6. If the shot hits an enemy, the enemy takes a predefined amount of damage points and the weapon's ammunition is decreased by one unit. If the weapon runs out of ammunition, a sound effect is played, to signal that the weapon is out of ammunition.

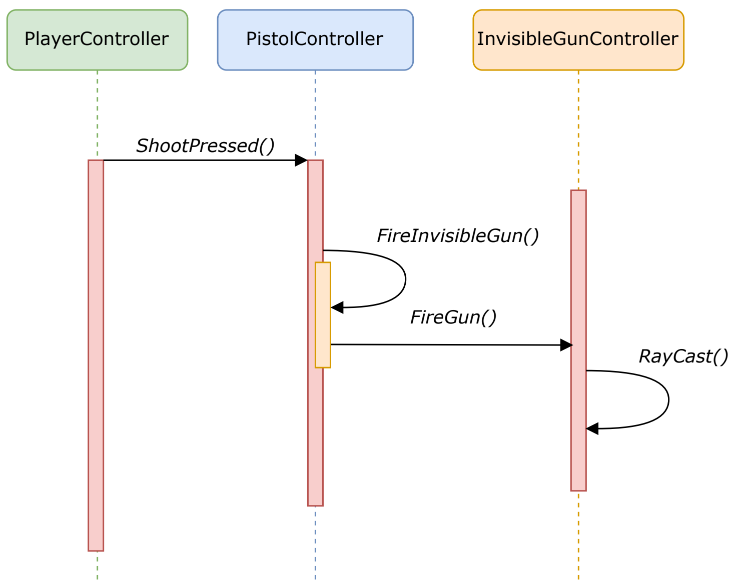


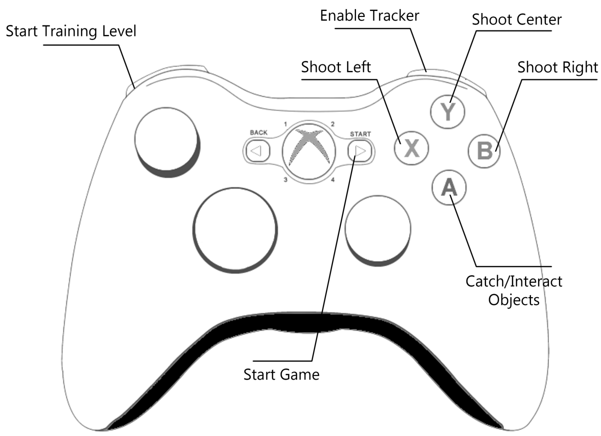
Figure 4.6: Shooting flow diagram

4.3.4 Control Scheme

In order to enable interaction between **Player** and **Pickable Objects**, it was important to define and establish the control scheme for the prototype. For this project, an *Xbox* 360 controller was used, as illustrated in Figure 4.7(a), and the control scheme shown in Figure 4.7(b) was defined.

The integration of the *Xbox* controller into *Unity* was straightforward thanks to the new input system released by *Unity*[8]. To integrate it, only one script called *InputManager.cs* was needed, which would execute the desired methods according to the player's input, as shown in fig. 4.8. In addition to implementing the controls for the *Xbox* controller, support for keyboard was also added to facilitate play-testing and debugging.

(a) *Xbox* 360 Controller



(b) Control Scheme

Figure 4.7: *Xbox* 360 Controller and Control Scheme

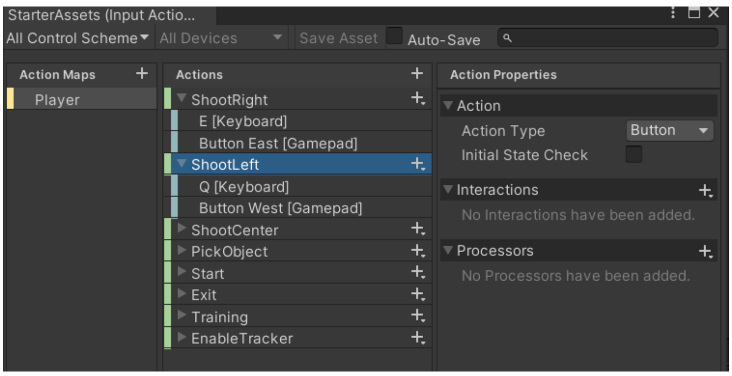


Figure 4.8: *Unity* new input system

[8] https://blog.unity.com/technology/introducing-the-new-input-system (as consulted in February 2023)

4.3.5 Lanes, Object Spawner and Object Picker

With the implementation of **Player** and **Pickable Objects** completed, it was crucial to devise a mechanism for sliding objects through the scenario like described in section 4.2. This involved defining and programming the concept of lanes. As a result, it was necessary to develop a method for generating and collecting the pickable objects, leading to the creation of an object spawner to randomly produce those objects in the lanes, and an object picker to facilitate the player's collection of these generated objects.

4.3.5.A Lanes

To implement the lane concept, a *prefab* was created initially consisting only of a *BoxCollider* and a *LaneController.cs* script. The **Lane** was designed to be a conveyor that transports pickable objects and enemies to the player. To achieve this, *Unity*'s native *OnTriggerStay(Collider other)*[9] method was used, which executes once per frame while there is a collision between *other* and the script owner. However, to make this mechanism work, colliders and the *Rigidbody*[10] component had to be added to all pickable objects. This component allows *GameObjects* to move and physically collide with other objects. It is used to apply the laws of physics such as force, velocity, acceleration and torque, as well as respond to external forces such as gravity.

To complete the implementation of the **Lane** *prefab*, the **Pickable Objects** had to be destroyed if they were not picked up, otherwise, they would continue to exist in the scene and play sound effects. Therefore, the native *Unity OnTriggerExit(Collider other)*[11] method was added to the controller script of all **Pickable Objects**. This method executes when the *other* object stops colliding with the script owner. With this new behaviour the pickable object will self-destruct whenever it stops colliding with the **Lane**.

Finally, three instances of the lane were created, one to the left of the player, one in the centre, and one to the right as shown in Figure 4.9.

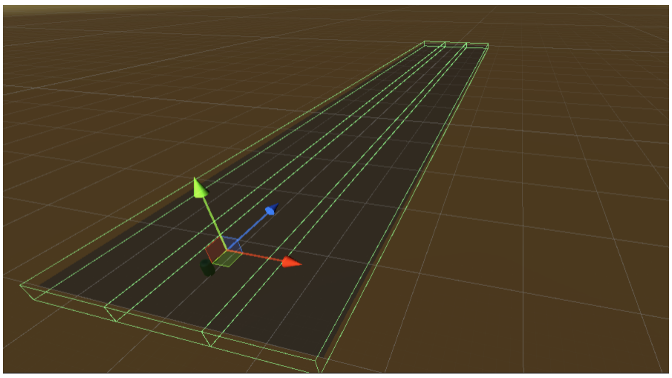


Figure 4.9: Lane instances

[9] https://docs.unity3d.com/ScriptReference/Collider.OnTriggerStay.html (as consulted in February 2023)
[10] https://docs.unity3d.com/ScriptReference/Rigidbody.html (as consulted in February 2023)
[11] https://docs.unity3d.com/ScriptReference/Collider.OnTriggerExit.html (as consulted in February 2023)

4.3.5.B Object Spawner

After creating the lanes, it was necessary to develop a mechanism for generating pickable objects randomly in each lane. This is where the concept of an **ObjectSpawner** was introduced, which involves having three spawning points, one in each lane. To achieve this, a *GameObject* called **SpawnPoint** was added to the **Lane** *prefab*, consisting of only one *transform*. An **ObjectSpawner** *prefab* was then created with an *ObjectSpawnerController.cs* script. The script contains a *SpawnObjects()* method, which randomly selects a lane and then selects a random object (health kit, enemies, etc) to be instantiated at the corresponding **SpawnPoint** position.

4.3.5.C Object Picker

The idea of creating an object picker arose to avoid overloading the **Player** with handling all the processes involving interaction with game environment. Although logic could be added to the *PlayerController.cs* to handle object picking, having a separate *prefab* to handle this feature seems more effective, especially when it comes to health kits and ammunition, which will be instantiated multiple times.

With this in mind, a *prefab* consisting of three *GameObjects*, one for each lane located opposite to the spawning points, was created. The *prefab* is composed of a script which uses the *ray casting* technique on each child and checks if there is a hit with a **Health kit** or **Ammunition**. If it does, the respective methods previously referred to in Section 4.3.3 are executed. Since there was only one weapon (**Pistol**) available, it didn't make sense to have this mechanic applied to that object, so the player automatically picks up the weapon as soon as it is instantiated. To notify the player that they have picked up the weapon, the reload sound effect is played.

4.3.6 Naud

Another important component that was missing from the prototype was the definition of an *Naud* enemy. It was established that the enemy would be instantiated like the other pickable objects and would have less health than the player (30 instead of 100). As such, it can be eliminated with only three shots. However, it would move quickly through the scenario, and if it managed to get to a certain distance from the player without being defeated, it would self-destruct, causing a high amount of damage. It only takes two hits to cause the player death and defeat, forcing the player to start the challenge again from the beginning. Based on this concept, an *Naud* enemy *prefab* was created containing a *BoxCollider* and a *Rigidbody*. This *prefab*, similar to the **Player**, is the parent of an **Audio** *prefab*, whose functionality is to manage the enemy's sound effects like described in section 4.3.1. The *Rigidbody* component allows the enemy to be moved through the lane, and the collider manages collision detection.

With the *Naud* enemy complete, it was time to incorporate the *Spatial Tracker* mechanism into the prototype. A new *prefab* called **EnemyTrackerAS** was developed. Two distinct scripts were developed and added to this *prefab*, called *EnemyTrackerController.cs* and *RumbleController.cs*. The former is responsible for producing sound stimuli, while the latter is responsible for haptic stimuli, with both taking into account the distance to the player.

The *PlayEnemyTrackerAudio()* method was developed to be executed on each fixed framerate frame, and it determines which sound effect to play and what the intensity of the motor vibration will be, following the logic established in the final approach (Chapter 3). To help the player clearly understand which lane the enemy was in, it was established that the same strategy would be followed for the shooting sound effect and associate the object with the lane instead of the player. In this sequence, a *prefab* called **ETAudio** was created at the same position as the **InvisbileGuns** and made a child of the **Lane** *prefab*. Hence, when an enemy is instantiated, the *Spatial Tracker* sound effects are played by the **ETAudio** assigned to the lane in which the enemy was generated.

4.3.6.A Roar and Attack

A *GameObject* named **Roar** was created to play a roaring sound effect on a loop that is produced by the enemy when it slides towards the player. While moving, the enemy controller script validates the distance to the player, and if it is less than or equal to a defined threshold, the *DoDamage()* method from the **Attack** *GameObject* is triggered to cause damage to the player and destroy the *GameObject* itself.

It should be noted that after developing these components, it was observed that the roar effect was not being perceived as intended. After investigating, it was discovered that this was due to the way the *ResonanceAudio* framework's sound wave simulation algorithm works. This algorithm assumes that sound sources are in fixed positions in the virtual space, and when a sound source moves, the algorithm has difficulty dealing with the position change, and may generate strange effects such as sound interruptions, delays, and inconsistencies in sound spatialization.

To fix this problem, the *ResonanceAudio* team recommends that sound sources be fixed in position, with listeners moving around them instead of the other way around[12]. Therefore, at this stage, there were two options: to look for an alternative or update all the work done so far to reverse the movement logic, that is, pickable objects would become static objects and the **Player** *prefab* would be the only movable object.

Before making changes, it was decided to reevaluate the other spatial audio frameworks. It was found that *Steam Audio* had a feature that allowed certain objects to be marked as "*dynamic geometry*" which causes occlusion, reverb, and other acoustic effects to be recalculated as it moves in virtual space solving the problem on hands. The only problem is that this would require updating all objects that used *ResonanceAudio* to use the new framework. To make a decision, *Steam Audio* was only applied to

[12] https://resonance-audio.github.io/resonance-audio/develop/unity/developer-guide (as consulted in February 2023)

the **Enemy** *prefab* in order to validate that it would have better results, and if successful, the change would be applied to all other objects. Since the testing was successful the *ResonanceAudioSource* was removed from and replaced with a script called *SteamAudioSource.cs* with HRTF enabled settings. For the **Enemy** *prefab* in particular, it was also necessary to add the script *SteamAudioDynamicObject.cs* and *SteamGeometry.cs*.

4.3.6.B Health System

Just like in the **Player** *prefab*, a health system needed to be implemented. The *GameObjects* **Hit** and **Pain** are used to play sound effects when the player hits the enemy. The **Hit** *GameObject* plays a sound when the player's hits the enemy, while the **Pain** *GameObject* plays a sound of the enemy's pain when hit. Some adjustments were necessary to balance the volume of both sound effects so that they do not overlap, making it difficult for the player to understand what is happening. For this reason, the sound of the **Hit** is played first and only when it ends is the sound of the pain played. Both *GameObjects* consist of *SteamAudioSource.cs* script and their respective audio controller scripts (*HitAudioController.cs* and *PainAudioController.cs*).

The **Die** *GameObject* functions similarly to the **Death** object (described in Section 4.3.2) in the **Player** *prefab*, and is used to play sounds when the enemy dies, either because its life has reached a value below 0 or because it was able to hit the player and self-destruct.

4.3.7 Level Design

Having completed the development of all the essential resources for the prototype, including the integration of the *Spatial Tracker* mechanism, it was time to start designing the level that would be used for usability testing. The idea is to have four distinct rooms with distinct purposes, where each room represents a corridor with access to other corridors through a door, as shown in Figure 4.10.

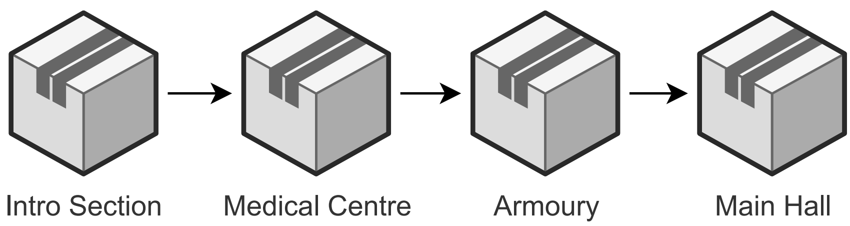


Figure 4.10: *Unity* new input system

4.3.7.A Room 1, Under Attack

In the first room, no objects besides the player are instantiated. The player starts injured, having only 20 health points, due to the events preceding the prototype experience (refer to Section 4.1). This way, the player can become familiar with the sound stimuli of footsteps, breathing and heart rate. The player exits the room upon colliding with the **Door** *prefab*, which triggers the loading of the next room.

4.3.7.B Room 2, Medical Centre

The second room, known as the medical centre, randomly spawns health kits in lanes 1 and 3. The purpose of this room is to assist the player in comprehending and utilising health kits, as well as the benefits of restoring health. Therefore, the **Door** *prefab* is not created until the player has completely regained their health, which requires collecting four health kits.

4.3.7.C Room 3, Armoury

The objective of the Armoury room is to teach the player on how to use the pistol and interact with ammunition. Therefore, the room is composed of three phases.

In the first phase, a **Pistol** is instantiated and automatically picked up by the player. In the second phase, similar to room two, ammunition kits are instantiated in lanes 1 and 3. The instantiating of ammunition only stops when the player picks up the second ammunition kit. Then, the third phase begins, which consists of a small target shooting challenge against three dummy robots (randomly instantiated in total, one for each lane).

To create this room, it was necessary to create the **DummyRobot** *prefab*, which is very similar to the *Naud* enemy, the only differences being its static nature and distinct sound effects upon being hit and destroyed. The third phase only ends when the third dummy robot is destroyed.

4.3.7.D Room 4, Main Hall

The last room in the level, known as the main hall, is the room that most closely resembles a normal level in a full version of the game. In other words, this is the room where players can interact with all the resources created for the game prototype. The challenge of this room is to defeat all the instantiated enemies. Once all enemies have been eliminated, the level ends.

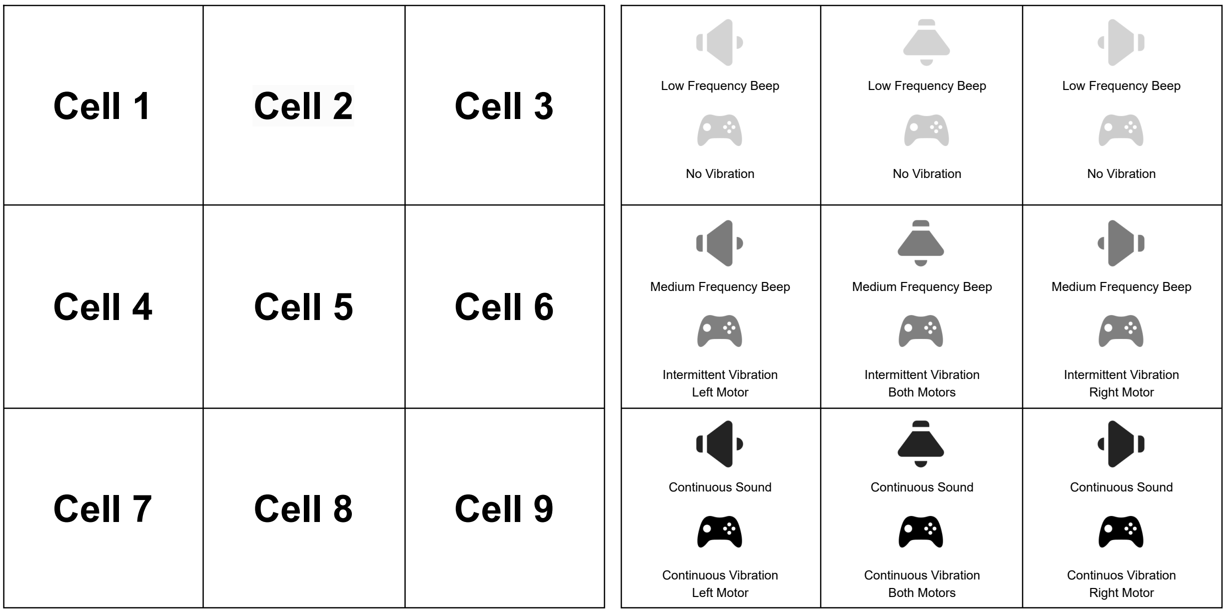
This phase of the project not only required the creation of new *prefabs* but also resulted in many changes to some objects. The **Object Spawner** *prefab* was the most affected, as new specific methods had to be implemented to instantiate objects as described in this section. For example, the methods *MedicalcenterSpawnObjects()* and *ArmorySpawnObjects()* were created. In addition to the **Door** *prefab*, it was necessary to create a **LevelManager** *prefab* to manage the setup and transition between levels.

This *prefab* is composed of a *LevelManagerController.cs* script and is the parent of a **SFX** *GameObject* responsible for playing the ambient music in each level. It is composed of several child *GameObjects*, one for each room, with a *SteamAudioSource.cs* and respective audio controllers script.

4.3.8 Training Level

With the prototype ready for usability testing, a mechanism was needed to help users learn how the *Spatial Tracker* mechanism works. With this idea in mind, it was established that a specific level should be created where the participant could focus its entire attention on the functioning of the *Spatial Tracker* mechanism.

Using the already developed resources, a new scene was created in *Unity* with a player and nine new *GameObjects* with a *BoxCollider*, each of which represents a cell of a grid as shown in Figure 4.11(a). The idea was to illustrate to the player how the discretization of the stimuli (sound and haptic) was established in the layout.



(a) Cell Setup **(b)** Stimuli at each cell

Figure 4.11: Training Level Setup

Therefore, a new *prefab* called **TrainingSpawner** was developed, which randomly selected a cell from the grid and communicated with the **Object Spawner** to instantiate an enemy in that cell so that the corresponding *spatial tracker* mechanism stimuli for that position were played. For example, if cell seven was selected, a continuous sound was played, and the vibration was only on the left motor and continuous. Figure 4.11 illustrates the established discretisation.

In addition to this new object, functionalities had to be added to existing objects such as *InputManagerController.cs*, *LevelManager.cs*, and even the script of the *Spatial Tracker* mechanism itself. *EnemyTrackerController.cs*, *EnemyAudioController.cs*, and **ObjectSpawner** had to be updated because they were all used in the same way as in the other scenes, except that in this scenario, the enemy did not produce any sound effects and those updates had to be made. Two sound warnings were also created to indicate to the player when the test started and when it ended.

4.4 Final Remarks

The final project did not deviate much from the originally established concept (refer to Section 4.2 for more details). The most challenging part was searching for open-source assets, especially sound effects that made sense to be assigned to certain *game objects*, For instance, for the sound effect of the health kit, we selected one close to a siren so that the player can associate the sound they hear with a health kit and regaining health. Some had to be manipulated using *Audacity* software to achieve the desired effect, which was a very time-consuming trial and error process. Changing from the *ResonanceAudio* framework to *SteamAudio* also had a significant impact on the project's development timeline because many objects had to be updated to use the new framework. Other than the points mentioned above, the project's development proceeded normally according to to the planned schedule. The source code has been made available, along with a release of the game for consultation on the *GitHub*[13] platform.

[13] https://github.com/drodriguesist/spatial-tracker-thesis/releases/tag/v1.0.0 (as consulted in May 2023)

5

Evaluation

Contents

To validate the effectiveness and usability of the *Spatial Tracker* mechanism developed, a series of user tests were conducted. These tests aimed to assess the users' ability to understand and utilise the mechanism, as well as their level of immersion and enjoyment while playing.

Furthermore, experimental tests were carried out later to evaluate the impact of the mechanism on players' performance in locating objects in the virtual environment and overall experience. The results of these tests will be presented and analysed in detail in this chapter, providing insights into the strengths and weaknesses of the mechanism, as well as potential areas for future improvements.

5.1 Apparatus

For both evaluation processes, a laptop computer with *Windows 10* operating system, 4*GB RAM*, and an *Nvidia GeForce GT 320M* graphics card was used to run the prototype developed in *Unity*. Additionally, the native *Windows Sonic* spatial audio solution was enabled. An *Xbox* 360 controller connected via USB was also used and for audio listening, *Sony WH-1000x M3* headphones equipped with Active Noise Cancelling (ANC) were utilised, allowing participants to have high sound isolation and thus promoting full focus on the tasks while simultaneously receiving optimal audio feedback from the prototype.

5.2 Measures

Throughout the execution of both experiments, several metrics were collected via the application prototype and stored in log files. This allows for cross-referencing the information with participants' feedback. The purpose of this data collection is to evaluate the player's performance in locating and identifying objects within the game scene, and their ability to complete the level before and after using the *Spatial Tracker* mechanism. This is assessed, for instance, through the number of enemies eliminated or the number of successful hits on enemies.

The data collected includes details about the current section of the level in which the player is present, when the player leaves a section, and when an object is generated. It also captures the instances when the player gains or loses health, when and which lane the player shoot, when enemies are hit, and finally, when enemies die.

5.3 Formative Usability Tests

As previously mentioned, after implementing the prototype in *Unity*, it is crucial to ensure that it can be used correctly by users in general. This is where usability tests come into play for this evaluation. The prototype can be assessed in the following interaction steps:

1. The player identifies and collects all health kits
2. The player understands they pick up the weapon
3. The player identifies and collects all ammo boxes
4. The player completes the target shooting challenge
5. The player completes the main hall section

5.3.1 Participants Profile

Studies suggest that usability tests conducted with about 6 participants are usually sufficient to identify most issues in a project [48]. However, the more participants included, higher the probability of detecting more specific and hard-to-find problems.

In this test, 13 sighted individuals participated, ranging in age from 23 to 62 years old ($M = 35.23$, $\sigma = 11.01$), with 8 participants being men and the remaining 5 women. It is worth noting that 61.5% of the participants play video games with a frequency of one to three hours per week [49], and 53.8% were familiar with the *Rail Shooter* genre.

Table 5.1: Participants Profiles

Id	Age	Gender	Play Frequency	Rail Shooter Experience
P1	30	Male	Casually	Yes
P2	62	Male	Don't Play	No
P3	40	Male	Casually	Yes
P4	25	Male	Casually	Yes
P5	33	Male	Casually	Yes
P6	35	Male	Casually	Yes
P7	23	Male	Casually	Yes
P8	29	Female	Don't Play	No
P9	34	Male	Casually	Yes
P10	23	Female	Don't Play	No
P11	33	Female	Don't Play	No
P12	48	Female	Don't Play	No
P13	35	Female	Casually	No

5.3.2 Procedure

The usability testing procedures were conducted individually for each participant and carried out in an isolated room with the necessary equipment, as previously mentioned in Section 5.1. The process began with a brief introduction explaining that the test would take place in three distinct phases and providing all the essential information for playing, including the control scheme and corresponding in-game actions. The first phase followed, with the participant testing the game without the *spatial tracker* mechanism enabled. After completing the second phase, the training process with the mechanism began, and the participant was informed that this new mechanism would be activated and how it worked. Once the participant completed the training phase with the mechanism, they began the third and final phase, where they played the prototype with the *spatial tracker* mechanism enabled. After the participant finished the last phase of the testing process, demographic data, such as age, gender, and experience with video games and the *Rail Shooter* genre, were collected, followed by questions regarding the prototype's usability which can be consulted in appendix B, with the goal of identifying potential issues that could later impact the experimental tests.

The usability tests aimed to assess, in particular, both the first and third phases, whether the player could collect all health kits in the medical centre section, identify that they picked up the weapon, collect all ammo boxes, overcome the target shooting challenge in the armoury section, and finally, complete the last section of the level.

5.3.3 Results and Findings

The results of the questionnaires related to the usability tests can be found in appendix B.4 section. These allowed the identification of the following set of issues.

The most commonly reported problem by several participants occurred in the medical centre section and involved difficulty in understanding that they had to collect more than one health kit and, as a result, could not advance to the next section. Another issue arose as a consequence of the previous problem and consisted on a lack of feedback when picking up objects and for all other controls, as participants noticed that the game state did not change so they tried using other buttons to see if the game state changed. At the same time, it was reported that after completing the main hall section, the ending was abrupt, and there was no clear indication that the level had ended. Issues were also identified during the armoury section where a *bug* was found. Basically, due to the *bug* the third dummy robot was not being generated, leaving the player stuck in the armoury section, thinking they were walking and nothing was happening.

Table 5.2: Summary of the issues found in the usability tests and solutions proposed to fix the problems

Problem	Solution
The end of the game is abrupt there is nothing indicating that the player has returned to the main menu	Add an auditory message to indicate the end of the level
There is no feedback on the player's actions, such as when attempting to shoot without a weapon, or trying to pick up objects when there are none in the scene	Add sound effects for when the player attempts to shoot without a weapon and pick up objects when they are not present in the scene

Difficulty in understanding that it was necessary to pick up health kits until health was fully restored	Reduce the number of health kits to be picked up in the medical centre section from 4 to 2.
Bug in which the third dummy robot was not instantiated, creating a loop where the player could not progress in the game	Fix the bug that occurred due to a *Null Pointer Exception*
Difficulty in identifying which section the player is in	Add an auditory message for each section to identify it

To fix the abrupt ending and the difficulty of identifying in which section the player is, new audio messages were created using Text To Speech (TTS) technology and then edited in *Audacity* to create the effect of an Artificial Intelligence (AI) robotic voice. So, when the player enters a new section, the facility's AI informs the player which section they are in. Once the player reaches the end of the Main Hall section, a similar warning is played to inform that the level has been completed. On the other hand, if the player dies at the hands of the enemies, a message is played, warning that the player has died. The messages are available in both English and Portuguese languages.

Regarding the *issue of a lack of feedback on the player's actions*, a new sound effect was created, which plays whenever the player attempts to perform an action they are currently unable to do, such as pressing the shoot key without having a weapon equipped or picking up health or ammunition when none of these objects are in the scene.

Lastly, the *bug with the dummy robot* was fixed, and in the medical centre section, *the number of health kits to be picked up was reduced from four to two*, and the volume of the heartbeat and breathing sound effects was increased.

5.4 Summative User Studies

After conducting the usability tests and implementing the respective changes in order to fix the problems identified, it was time to proceed with summative user studies to assess participants' performance and the impact of the *spatial tracking* mechanism in locating the enemies during the main hall section.

5.4.1 Participants Profile

To validate the impact of the *spatial tracking* mechanism, the participation of IVI was necessary. The selection criteria consisted on having participants older than 16 years of age, blind or with reduced visibility for at least 5 years, and without any hearing problems. Thus, the execution of the experimental tests involved 10 participants in total as shown in table 5.3, half of whom were men and the other half women, aged between 26 and 60 years old ($M = 40.7,\ \sigma = 13.60$). All participants were introduced by the Raquel and Martin Sain Foundation [1], where 50% do not usually play video games regularly, 40% play between 1 to 3 hours per week, and 10% claim to play 7 or more hours per week [49].

Table 5.3: Participants Profiles

Id	Age	Total Blindness Onset	Gender	Play Frequency	Rail Shooter Experience
P1	63	60	Male	Regularly	No
P2	39	39	Male	Don't Play	No
P3	40	40	Female	Casually	No
P4	35	25	Male	Casually	Yes
P5	60	4	Female	Casually	No
P6	40	40	Female	Casually	No
P7	38	6	Male	Don't Play	Yes
P8	37	27	Male	Don't Play	No
P9	32	22	Female	Casually	No
P10	26	26	Female	Don't Play	No

5.4.2 Procedure

The experimental tests were conducted in two sessions held on October 27^{th} and 28^{th} 2022, at the facilities of the Raquel and Martin Sain Foundation, where an isolated room was provided to avoid interference during the tests. The structure of the summative user tests is similar to that used in the formative usability tests, with an introductory phase added to the summative user tests to allow participants to familiarise themselves with the application prototype, for more detail refer to the guide available in appendix D.

The beginning of each session with the participants was characterised by a simple introduction to the experience. This was followed by an explanation of the session's objective, and all necessary information was provided to the participants so they could perform the tests, including a brief description of the virtual environment, control scheme, and the stimuli (sound and haptic) they would receive from the developed prototype. Unlike the usability test sessions, participants were allowed to interact with the application prototype at this initial stage and clarify any questions they had. The goal is to let participants familiarise themselves with the prototype.

Once the introductory phase was completed, the tests continued with the participant testing the prototype with the *Spatial Tracker* mechanism deactivated, and having to rely on the spatialised audio cues only. As soon as the participants finished the test, they moved on to the next phase, where they would learn how the *Spatial Tracker* mechanism worked through the training level. The participants were informed that they did not need to perform any actions and should focus only on the stimuli produced by the prototype. Once the training with the mechanism was completed, the session proceeded to the next phase where the participant tested the prototype with the active mechanism.

[1] http://www.fundacao-sain.pt/ (as consulted in February 2023)

At the end of the session, participants were surveyed through two questionnaires, one for participant profile characterisation and another for satisfaction, those can be consulted in appendix C.

5.4.3 Data Collection and Processing

The data collected during the summative user tests was processed and can be found compiled in Table 5.4, where each metric has the following meaning:

- **Shots -** Represents how many shots on target the participant had in the Main Hall section
- **Ammo -** Reveals each time the participant tried to shoot but did not have ammo at that time.
- **Killed -** Counts the number of enemies killed by the participant in the Main Hall section
- **Level -** Indicates whether the participant manage to complete the level or not
- **Time -** This metric depicts the time spent by the participant to complete the test, either by reaching the end of the level or being killed by the enemies.
- **Score -** The score is calculated based on the number of accurate shots (20%), enemies killed (20%), amount of ammunition collected (20%), number of health kits picked up (20%), and whether the level was completed (20%) or not. The maximum score is 100 points, and the minimum is 0 points.

It should be noted that in this section, if the player reaches the end of the level, a total of six enemies are generated, so the maximum number of enemies that the participant can kill is six.

Table 5.4: Data collected from summative user tests

ID	Shots (B)*	Shots (A)†	Ammo (B)	Ammo (A)	Killed (B)	Killed (A)	Level (B)	Level (A)	Score (B)	Score (A)	Time (B)	Time (A)
P1	13	15	0	0	3	3	1	1	80	83	05:01	03:15
P2	7	11	4	13	2	3	0	1	36	71	03:30	03:15
P3	14	10	28	27	4	3	1	1	85	70	03:37	03:07
P4	5	2	7	11	0	0	0	0	30	36	08:23	04:08
P5	0	3	0	0	0	0	0	0	0	14	04:45	05:36
P6	0	6	1	17	0	0	0	0	10	41	04:27	03:05
P7	8	0	16	24	2	0	0	0	37	0	07:59	02:29
P8	3	5	46	0	1	0	0	0	41	17	05:39	08:10
P9	8	16	19	3	2	5	0	1	50	90	03:12	01:33
P10	0	3	2	14	0	1	0	0	0	8	00:48	03:14

* *B stands for **B**efore using the Spatial Tracker*
† *A stands for **A**fter using the Spatial Tracker*

5.4.4 Data Analysis

In order to determine if there were significant differences in participant performance with the use of the *Spatial Tracker* mechanism, we began by analysing the time participants took to perform the test with the prototype under both conditions, with and without the mechanism.

In the first phase, when the *Spatial Tracker* mechanism was disabled, participants took between 48 and 503 seconds (8 minutes and 23 seconds) to complete the test, on average, participants took 285.1 seconds, that is 4 minutes, 12 seconds ($M = 285.1,\ \sigma = 120.61$). In the third phase, with the *Spatial Tracker* mechanism active, participants took between 93 and 490 seconds (8 minutes and 10 seconds) and on average took 211.2 seconds, that is, 3 minutes and 32 seconds ($M = 211.2,\ \sigma = 102.47$). Given that only four participants managed to complete the level, conducting a statistical analysis on the "Time" spent variable, before and after using the *Spatial Tracker* mechanism may not yield robust and reliable results. Small sample sizes can lead to low statistical power, which complicates the task of identifying significant differences or drawing meaningful conclusions.

To validate whether the mechanism helped participants and had an impact on their performance, we used the Kolmogorov-Smirnov (non-parametric) and Shapiro-Wilk (parametric) normality tests to determine if the collected metrics followed a normal distribution, consult Table 5.5 using the *SPSS* statistics software[2]. The Shapiro-Wilk test is an objective test and provides a quantitative measure of the normality of distributions regardless of the sample size, which is important considering the sample size of this experiment and is considered stronger than the Kolmogorov-Smirnov test for small samples up to about 50 observations [50].

Table 5.5: Results from Kolmogorov-Smirnov and Shapiro-Wilk tests

	Kolmogorov-Smirnov			Shapiro-Wilk		
	Statistic	**df**	**Sig.**	**Statistic**	**df**	**Sig.**
Shots (B)*	.170	10	.200	.904	10	.242
Shots (A)†	.178	10	.200	.919	10	.352
Ammo (B)	.237	10	.120	.822	10	.027
Ammo (A)	.185	10	.200	.899	10	.214
Killed (B)	.236	10	.121	.868	10	.094
Killed (A)	.292	10	.015	.789	10	.011
Level (B)	.482	10	$<$.001	.509	10	$<$.001
Level (A)	.381	10	$<$.001	.640	10	$<$.001
Time (B)	.150	10	.200	.949	10	.655
Time (A)	.314	10	.006	.824	10	.028
Score (B)	.145	10	.200	.921	10	.364
Score (A)	.191	10	.200	.910	10	.281

* *B stands for **B**efore using the Spatial Tracker*
† *A stands for **A**fter using the Spatial Tracker*

As some of the metric samples do not follow a normal distribution and the data sample size is less

[2] https://www.ibm.com/spss (as consulted in February 2023)

than 30 samples, a non-parametric test was used for the variables "Shots" and "Killed". Considering that the participants are the same in both measurements (with and without the *Spatial Tracker* mechanism), the samples are dependent, and due to the sample size, the most appropriate test is the Wilcoxon Signed-Rank Test.

5.4.4.A Wilcoxon Signed-Rank Test - Shots

The application of the test showed that there was no statistically significant evidence to reject the null hypothesis that the median of the differences between "Shots" (shots on target) before $(M = 5.6,\ n = 10)$ and after using the mechanism $(M = 7.1,\ n = 10)$ is equal to zero,$Z = -0.768$ and $\rho - value = 0.443$ (two-sided test), see figure and table fig. 5.1 table 5.6. Therefore, we can observe that there is no significant difference in the "Shots" score before and after the mechanism was used.

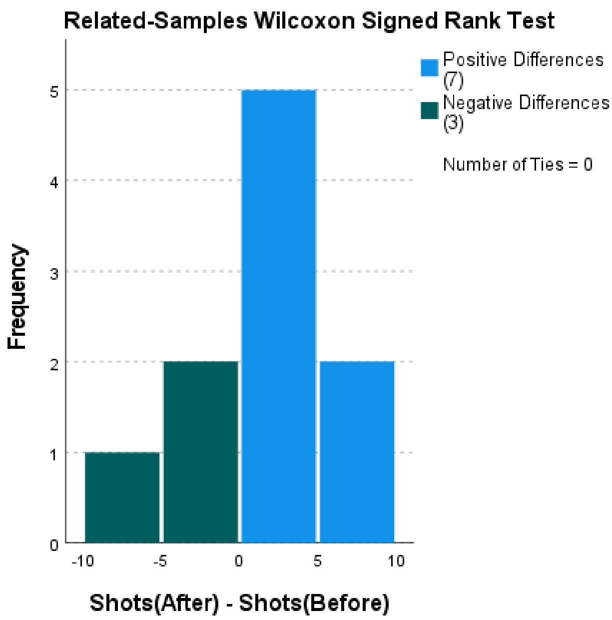


Figure 5.1: Differences between Shots(**B**efore) and Shots(**A**fter) using the mechanism

Table 5.6: Wilcoxon Signed-Rank Test with Shots variable

Test Statistics	
Wilcoxon Signed-Rank Z	-0.768*
Asymp. Sig. (2-tailed)	0.443

* *Based on negative ranks*

5.4.4.B Wilcoxon Signed-Rank Test - Killed

Applying the same test to variable “Killed” showed that similar to the “Shots” variable, there is no significant evidence to reject the null hypothesis that the median of the differences between “Killed” (enemies killed) before ($M = 1.4,\ n = 10$) and after using the mechanism ($M = 1.8,\ n = 10$) is equal to zero, $Z = -0.108$ and $\rho - value = 0.914$ (two-sided test), as illustrated by fig. 5.2 and table 5.7. Thus, we can conclude that there is no significant difference in the “Killed” score before and after the use of the mechanism.

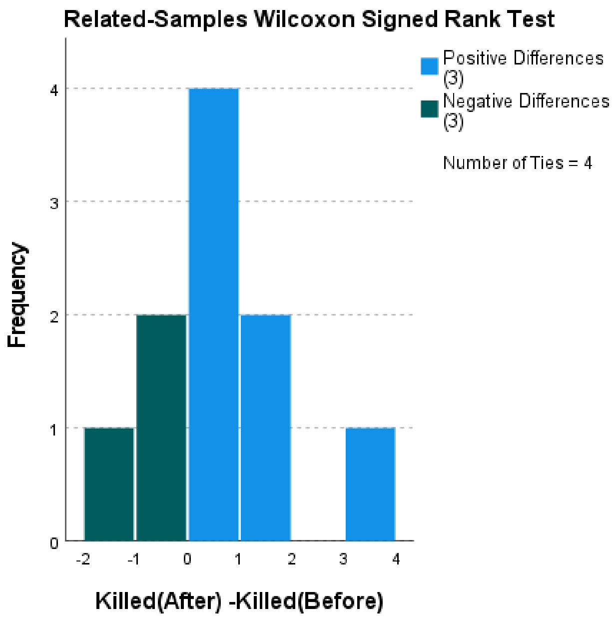


Figure 5.2: Differences between Killed(**B**efore) and Killed(**A**fter) using the mechanism

Table 5.7: Wilcoxon Signed-Rank Test with Killed variable

Test Statistics	
Wilcoxon Signed-Rank Z	-0.108*
Asymp. Sig. (2-tailed)	0.914

* *Based on negative ranks*

5.4.4.C Wilcoxon Signed-Rank Test - Score

The variable “Score” was also subjected to the Wilcoxon Signed-Rank test to validate if there was any significant difference between the use and non-use of the mechanism. As occurred in the previous cases, it was found that there is no significant evidence to reject the null hypothesis that the median of the differences between "Score" before ($M = 36.9,\ n = 10$) and after using the mechanism ($M = 43,\ n = 10$) is equal to zero, $Z = -0.764$ and $\rho - value = 0.445$ (two-sided test), as illustrated by fig. 5.3 and table 5.8. Thus, we can conclude that there is no significant difference in the "Score" before and after the use of the mechanism.

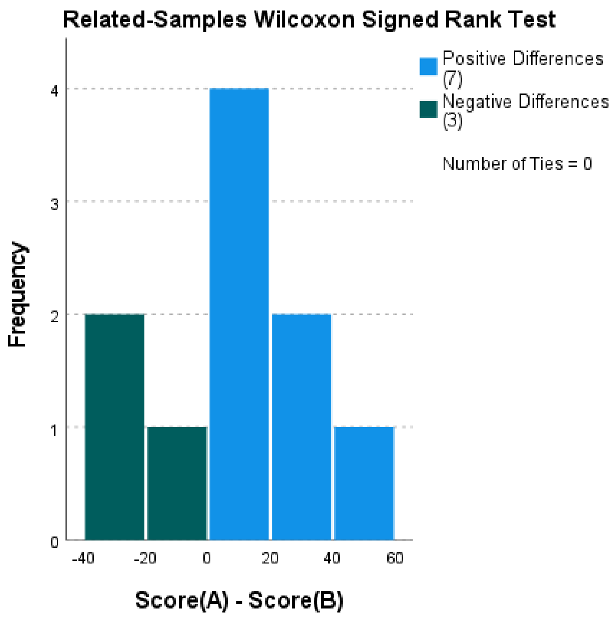


Figure 5.3: Differences between Score(**B**efore) and Score(**A**fter) using the mechanism

Table 5.8: Wilcoxon Signed-Rank Test with Score variable

Test Statistics	
Wilcoxon Signed-Rank Z	-0.764*
Asymp. Sig. (2-tailed)	0.445

* *Based on negative ranks*

5.4.4.D McNemar Test - Level

It was also necessary to evaluate whether there was a significant impact of using the mechanism on the participants' ability to complete the level or not. Since the samples of "Level" are paired, the participants are measured twice (with and without the mechanism) and the variable is binary categorical ($Yes = 1,\ No = 0$), the most appropriate test to apply is the McNemar test. table 5.9 and table 5.10 shows the results of the McNemar test, the $\rho - value$ of the test is 0.500. Since the value is not less than the significance level of 0.05, we do not have sufficient evidence to reject the null hypothesis that the difference between the proportions of equal and different outcomes follows an approximately symmetrical McNemar distribution, and consequently, there is no significant difference in the proportion of participants who completed the level before and after using the mechanism.

Table 5.9: McNemar Test Cross Tabulation on variable “Level”

		Level(B)†		**Total**
		0	**1**	
Level(A)*	**0**	6	0	6
	1	2	2	4
Total		8	2	10

* *B stands for **B**efore using the Spatial Tracker*
† *A stands for **A**fter using the Spatial Tracker*

Table 5.10: Chi-Square Tests for McNemar Test

	Value	**Exact Sig. (2-sided)**
McNemar Test		0.500*
N of Valid Cases	10	

* *Binomial distribution used*

Taking into account the results obtained in the previous tests, the question arose whether we could observe any correlation between the sample of the variable “Ammo” and the variables “Level” and “killed”, which could raise questions about the prototype’s design regarding the mechanics of reloading the weapon with ammunition, since the “Ammo” score indicates the number of times the player attempted to shoot but was out of ammunition, this may have influenced the player’s performance and is probably reflected in the number of enemies killed and their ability to complete the level or not. Since correlation tests only evaluate the statistical relationship between two variables, it does not imply a causal relationship between them.

To assess the relationship between the two variables, we can use Spearman or Pearson correlation tests. The main difference between the two tests is that the Pearson test evaluates the linear correlation between the variables, while the Spearman test assesses the rank-order relationship between the variable values. For these reasons, the Pearson test is more suitable for variables that have a linear distribution and requires the variables to have a normal distribution, while the Spearman test is more appropriate for variables that show an increasing or decreasing rank-order relationship. As depicted in table 5.5, not all variables follow a normal distribution, so it was established that the Spearman test would be used for this evaluation.

5.4.4.E Spearman Test - Ammo and Level

To perform this evaluation, the original data was reorganised. As can be seen in table 5.11, after applying the Spearman test, we can observe a coefficient of $R = 0.00$ and a $\rho - value\ of\ 1.000$, which is greater than the significance level of 0.05 ($\rho > 0.05$). We can conclude that there is no statistically significant relationship between the number of times the participant tried to shoot without ammunition and whether or not they completed the level.

Table 5.11: Spearman Correlation between Ammo and Level variables

			Ammo	**Level**
Spearman's rho	**Ammo**	**Correlation Coefficient**	1.000	.000
		Sig. (2-tailed)	.	1.000
		N	20	20
	Level	**Correlation Coefficient**	.000	1.000
		Sig. (2-tailed)	1.000	.
		N	20	20

5.4.4.F Spearman Test - Ammo and Killed

Similarly to the previous test, when the Spearman test was applied to the data in table table 5.12, the results revealed a correlation coefficient $R = 0.221$ and a $\rho - value = 0.348$, which is greater than the significance level of 0.05 with a 95% confidence interval. In this way, we can assert that there is no significant correlation between the two variables.

Table 5.12: Spearman Correlation between Ammo and Killed variables

			Ammo	**Level**
Spearman's rho	**Ammo**	**Correlation Coefficient**	1.000	.221
		Sig. (2-tailed)	.	.348
		N	20	20
	Level	**Correlation Coefficient**	.221	1.000
		Sig. (2-tailed)	0.348	.
		N	20	20

Lastly, it is important to validate if the participant's profile can impact the results obtained with the use of the mechanism. Therefore, we selected only the participants who usually play with at least a casual frequency (at least 1 to 3 hours per week) which constituted 50% of the participants, thus filtering our sample, and ran the Wilcoxon Signed-Rank Test again for the same variables (since there were still samples that did not have a normal distribution) in order to validate if there was a significant difference.

5.4.4.G Wilcoxon Signed-Rank Test - Shots Filtered

The test application once again demonstrated that there was no significant statistical evidence, with $Z = 1.214$ and a $\rho - value$ of 0.225 (two-sided test). Please refer to the fig. 5.4 and table 5.13. As the $\rho - value$ is higher than the significance level of 0.05, we can conclude that there is no significant difference in the "Shots" (shots on target) score before and after the mechanism was used, even with the filtered data.

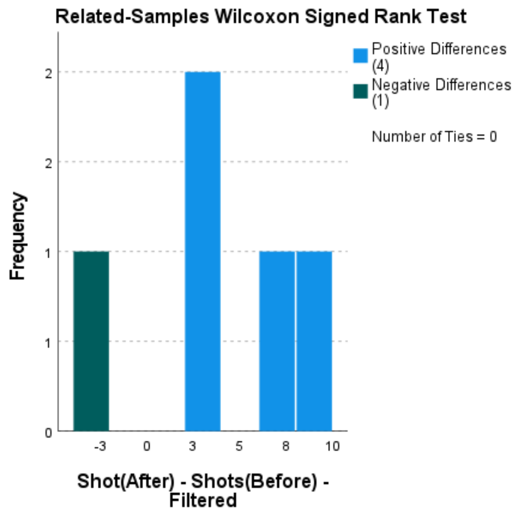


Figure 5.4: Differences between Shots(**B**efore) and Shots(**A**ter) using the mechanism

Table 5.13: Wilcoxon Signed-Rank Test with Shots variable

Test Statistics	
Wilcoxon Signed-Rank Z	-1.214*
Asymp. Sig. (2-tailed)	0.225

* *Based on negative ranks*

5.4.4.H Wilcoxon Signed-Rank Test - Killed Filtered

The results obtained after applying the Z-test with $Z = 0.447$ and $\rho - value = 0.655$ (two-sided test), as shown in Figure 5.5 and Table 5.14, which show that there is no statistically significant evidence to reject the null hypothesis. Therefore, since the $\rho - value$ is greater than the significance level of 0.05, we can conclude that there is no significant difference in the "Killed" (enemies killed) score before and after the mechanism was used even after data filtering.

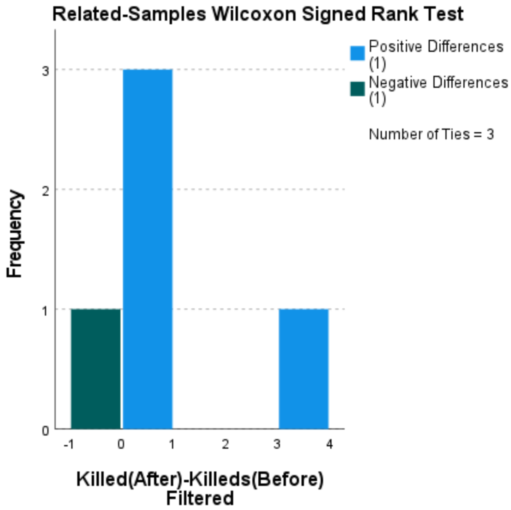


Figure 5.5: Differences between Killed(**B**efore) and Killed(**A**ter) using the mechanism

Table 5.14: Wilcoxon Signed-Rank Test with Killed variable

Test Statistics	
Wilcoxon Signed-Rank Z	-0.447*
Asymp. Sig. (2-tailed)	0.655

* *Based on negative ranks*

5.4.4.I Wilcoxon Signed-Rank Test - Score Filtered

The application of the test to the variable "Score" once again demonstrated that there is no statistically significant evidence to reject the null hypothesis with $Z = 0.273$ and $\rho - value = 1.095$ (two-sided test), as shown in Table 5.15. Therefore, since the $\rho - value$ is greater than the significance level of 0.05, we can conclude that there is no significant difference in the "Score" before and after the mechanism was used, even after data filtering.

Table 5.15: Wilcoxon Signed-Rank Test with Score variable

Test Statistics	
Wilcoxon Signed-Rank Z	-1.095*
Asymp. Sig. (2-tailed)	0.273

* *Based on negative ranks*

5.4.4.J McNemar Test - Level Filtered

Similarly, the McNemar test was also applied to the filtered data. Table 5.16 and Table 5.17 show the results of the test, with a $\rho - value$ of 1.000. This value is greater than the significance level of 0.05, and thus we do not have enough evidence to reject the null hypothesis that the difference between the proportions of equal and different results follows an approximately symmetric McNemar distribution. In other words, this means there is no significant difference in the proportion of participants who completed the level before and after using the *Spatial Tracker* mechanism.

Table 5.16: McNemar Test Cross Tabulation on variable "Level" Filtered

		Level(B)		**Total**
		0	**1**	
Level(A)	**0**	2	0	2
	1	1	2	3
Total		3	2	5

Table 5.17: Chi-Square Tests for McNemar Test - Data Filtered

	Value	**Exact Sig. (2-sided)**
McNemar Test		1.000*
N of Valid Cases	5	

* *Binomial distribution used*

5.4.4.K Participant Feedback

As previously mentioned, a satisfaction questionnaire (please refer to appendix C) was also administered to the participants consisting of ten questions whose responses were given on a five-point Likert scale, where point 1 means that the participant Strongly Disagree and point 5 means the participant Strongly Agree. The idea was to evaluate the user satisfaction after using the *Spatial Tracker* mechanism. After an evaluation of the descriptive statistics such as the mean and standard deviation (see table 5.18 of the responses, it is possible to observe that the majority of the participants (80%) felt emotionally connected to the game (Question 8) and in Question 10 ("Would you play the game again?") all participants liked the game enough to play it again. Similarly, Questions 2 and 3 ("How much effort did you put into playing the game?") and ("Did you feel that you did your best?") had relatively high means, suggesting that the participants were engaged and motivated during the experiment with the *Spatial Tracker* mechanism.

The objective was to ensure that the interaction with the prototype was satisfactory for the participants, suggesting that it did not affect the results obtained during the experimental tests. To test if the interaction with the prototype was satisfactory, a one-sample T-test was used, where the mean of all responses was considered as an estimate of the overall satisfaction of the participants with the prototype interaction. The following hypotheses were formulated:

Table 5.18: Descriptive Statistics for participants feedback

	N	Mean	Std. Deviation
Q1	10	4.40	.843
Q2	10	4.20	1.033
Q3	10	4.50	.707
Q4	10	3.90	1.370
Q5	10	2.70	1.829
Q6	10	4.60	.699
Q7	10	4.10	.738
Q8	10	4.20	.789
Q9	10	4.70	.483
Q10	10	5.00	.000

Null hypothesis $(H0)$**:** the population mean is equal to 4 (which is the value of a agreeable response on the scale used by the questionnaire).

Alternative hypothesis $(H1)$**:** the population mean is different from 4.

The results of the One-Sample T Test (Table 5.19) show a $\rho - value = 0.137$ (one sided) and $\rho - value = 0.275$ (two sided) both higher than a significance level of 0.05, so we can't reject the null hypothesis and can conclude that there isn't statistical evidence to assert that the population mean is significantly different from 4 (the value of a agreeable response on the scale used by the questionnaire). This suggests that the interaction of the participants with the prototype can be considered satisfactory. Another factor that supports this idea is that several participants asked after the experience if they could have access to the prototype in order to play again.

Table 5.19: One-Sample Test for the questions means with Test Value $= 4$

	t	df	Significance		Mean Difference	95% CI*	
			One-Sided p	Two-Sided p		Lower	Upper
Mean	1.163	9	.137	.275	.2300	-.217	.677

* *CI **C**onfidence **I**nterval*

It is important to note that in the final question made to the participants in the end of each session (see the guide appendix D), three participants mentioned that the mechanism ultimately introduced more inputs, causing more confusion than necessarily helping. One of these three participants even stated that the haptic stimulus (controller vibration) was distracting and that perhaps only the auditory stimulus would be sufficient or provide the ability to toggle only one of those inputs.

5.4.5 Results and Findings

Based on the analysis of the results obtained from the experimental tests, it is not possible to verify the research goal proposed by this study, "Is it possible to improve the process of locating objects in a virtual environment for people with visual impairments through the use of a *Spatial Tracker*?"

Through the results of this study, suggest that the use of the mechanism was not effective in improving the process of locating objects in the virtual environment for IVI, as there was no significant difference in participants performance with and without the mechanism.

However, it is essential to consider as well the effectiveness of the baseline prototype (without the *Spatial Tracker* mechanism) in order to interpret the results accurately. If the baseline prototype already provides a satisfactory level of performance in locating objects the addition of the tracking mechanism may not have resulted in significant improvements which could also justify the results obtained. In this case, that satisfactory level of performance could be attributed to the effort that was put into designing and implementing the baseline prototype. To address this concern and obtain a deeper understanding, further analysis and evaluation could be conducted.

Additionally, the qualitative feedback from the participants regarding their experience with the baseline prototype could also provide some insights on this topic. By collecting their impressions, preferences, and suggestions, we could gather more information on aspects that might be improved or enhanced in future iterations. As referred in section 5.4.4.K, some participants reported that the tracker mechanism ended up causing more confusion than helping locating the objects, which supports the theory of an already efficient baseline prototype, nevertheless, there was not enough reports that can confirm that this was the case.

Considering these factors and conducting further investigations can help clarify whether the tracking mechanism was ineffective or if the baseline prototype already provided a high level of efficiency, thus influencing the results of the study.

It was analysed that the participants' profile and their experience with virtual environments may have justified the absence of a significant difference in the results obtained. However, even after filtering the data obtained from participants who played casually or regularly, it was not possible to verify a significant difference in the interaction with the prototype with and without the mechanism.

On the other hand, there are some reasons that may justify the absence of a significant impact from the mechanism besides the ones already provided. One of them is related to the process of learning the mechanism, it is possible that participants did not receive sufficient training with the *Spatial Tracker* mechanism to use it more effectively. The number of times a individual should train to learn a new tool varies widely and depends on a number of factors, including the complexity of the tool, the person's previous experience and skills, their learning style, and the type of training provided.

The Ebbinghaus forgetting curve [51] suggests that learning is better retained with repeated, spacedout sessions. This means that the training phase with the mechanism should be repeated over time, with breaks in between, rather than being crammed into one intense session. A study coonducted by Donovan and Radosevich [52] supports this theory. The authors tested different training schedules and found that a moderate level of spacing between sessions (sessions distributed over time) was most effective for retention.

Another reason that may justify the absence of an impact is the sample size. A small sample may affect the reliability of the results since there is a higher risk that the sample is not representative of the general population, which may have resulted in very high confidence intervals for the statistical data obtained in the experimental tests, making it more difficult to identify significant differences in the stipulated metrics. The Central Limit Theorem [53] supports this idea that small samples can influence the reliability of the results obtained. It states that when the sample size increases, the sample means tend to follow a normal distribution around the population mean, making large samples more representative of the population. On the other hand, small samples may not adequately represent the population, leading to larger confidence intervals and less reliable results [54]. According to the 2001 portuguese census, there were approximately 163,569 visually impaired individuals, constituting about 1.6% of the total resident population[3]. As of 2021, the World Health Organization (WHO)[4] estimated that at least 2.2 billion people worldwide have a visual impairment or blindness, equating to approximately 28% of the global population. These figures underscore the complexity behind securing significant sample sizes for this demographic group.

Since the test with the *Spatial Tracker* mechanism was carried out after the test without the mechanism reveals a learning effects issue [55]. This means there's the potential for participants' performance to improve over the course of an experiment, simply due to repeated exposure to the tasks involved, rather than as a result of the specific conditions being tested. This issue should have been counterbalanced, for example, through a random selection of which option (with or without mechanism) would be tested first.

Lastly, the design of the prototype may have influenced the results as well. Although the participants' feedback regarding the prototype is positive, it does not invalidate the fact that the prototype may have design issues. It was identified during the experimental tests that some participants did not complete the level because they had run out of ammunition. It was attempted to verify if there was any relationship between the number of times the participant tried to fire without ammunition and the number of enemies killed, as well as the completion of the level, but in both cases, no correlation was identified. However, this was a topic that raised the question of whether the ammunition loading mechanic were automatic, would participants have performed differently in the experimental tests?

[3] https://www.pordata.pt/db/portugal/ambiente+de+consulta/tabela (as consulted in May 2023)
[4] https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment (as consulted in May 2023)

6

Conclusion

Contents

In this document can be found a use-case aimed to help IVI locate objects in a game environment. The research goal proposed by this study was whether a *spatial tracking* mechanism could improve the process of object location in virtual environments for IVI. To address this question, research of the current state of the art in accessibility for virtual environments was undertaken, along with an examination of existing video games for IVI. Additionally, potential technological solutions were assessed to facilitate these individuals' access to these environments.

To test this, we created a game prototype that incorporated the developed *Spatial Tracker* mechanism.

It was established that the experiment would consist of three phases in order to evaluate the impact of the mechanism on the participants performance locating objects in the game environment. The initial phase required participants to use the prototype without the mechanism in place. This was followed by a training phase involving the *Spatial Tracker*. Finally, in the last phase, participants interacted with the prototype again but this time with the mechanism enabled. Once the first stable version of the prototype was completed, usability tests were performed to identify possible usability problems.

The evaluation with visually impaired users was carried out with 10 participants, and the analysis of the results obtained suggests that the use of the developed mechanism did not have a significant impact on the process of object location in virtual environments for IVI. Notably, the baseline prototype, which doesn't incorporate the *Spatial Tracker* mechanism, may have already provided satisfactory performance in object location. This could make the addition of the tracking mechanism less impactful, thereby justifying the lack of significant difference in the participants performance. Also, the qualitative feedback from a small group of participants indicates that the tracker mechanism might caused more confusion, but in general participants may not have received sufficient training to use the spatial tracking mechanism effectively. Alongside, the small sample size which may affect the reliability of the results, as it might not be representative of the general population.

6.1 System Limitations and Future Work

As mentioned in section 5.4.5, during the experimental evaluation, some other considerations arose, particularly in terms of game design choices, such as the weapon reload mechanic. Would there be better performance if the pistol reload system was automatic? Considering the potential relief of cognitive and task burden for the players, the introduction of an automatic reload mechanic could allow players to focus more on game play and interaction within the game environment.

Another question raised by the analysis of experimental results is whether the introduction of a phased training with the mechanism over time could enhance participants' performance. The gradual exposure allows participants to familiarise with the mechanism, understand its functionalities and adjust to the interaction with the game environment. While there's no specific data supporting this idea, it's reasonable to assume that gradual exposure could lead to better adaptation and skill mastery. However, this is a hypothesis that requires further empirical studies for confirmation. These studies should compare participant performance across different training modalities, providing concrete evidence about the benefits of phased training with the *Spatial Tracker* mechanism.

On the other hand, a methodological limitation was identified in the experimental procedure, which could also justify the absence of significant differences. This is particularly related to the learning effect problem, as referred to in section 5.4.5.

Although the experimental results revealed that the impact of the developed mechanism was not significant, there are several intriguing topics that have emerged as a consequence of the work developed in this thesis which could be further explored. For instance, it would be worth examining the interaction of participants (both with and without the mechanism) with different devices, such as mobile devices. Such study could offer significant advantages, such as enhanced accessibility, portability, lower cost and hardware requirements, as well as leveraging the established development ecosystem. This would facilitate making the functionality available to a broader number of users while taking advantage of available tools and resources.

It would also be important to evaluate the use of the mechanism in performing a predetermined set of tasks with varying levels of complexity, such as a variant of the training level, where the first challenge for the participant would be to locate an object in the grid, and the other, more complex, would involve locating two or more objects. This could allow the evaluation of the mechanism's effectiveness in dealing with increasing challenges and verify its ability to improve participants' performance in more complex tasks. In addition, by conducting an assessment across a predetermined set of tasks, a more comprehensive understanding of the mechanism's capabilities can be achieved, identifying potential limitations or areas for improvement.

Another idea that was considered during the design phase of this project was the use of multiple *spatial tracking* mechanisms, where each mechanism would have a specific functionality associated with it, and it would be up to the player to decide when and which mechanisms would be active at any given moment.

Finally, it would be interesting to evaluate whether the integration of ambisonic sound effects with the mechanism could also improve its effectiveness. The addition of ambisonic sound effects into the *Spatial Tracker* could provide a more immersive for participants, making interaction with the mechanism feel more realistic and natural. This might aid in spatial perception and object location in the game environment. By assessing the integration of ambisonic sound effects, it would be possible to investigate how they affect the effectiveness of the mechanism in terms of accuracy, speed and ease of object location.

In conclusion, while the current study did not unequivocally demonstrate a significant impact on enhancing the object location process in the virtual environment, the development of the *Spatial Tracker* lays the groundwork for promising future investigations. It represents a step forward in research surrounding interface usage in video games for IVI and the improvement of accessibility for this group of individuals. The mechanism explored in this study provides a foundation from which further enhancements can be made and invites additional studies to evaluate its effectiveness in several distinct contexts. Despite the initial findings, the potential implications of this work are far-reaching and indicative of an exciting trajectory in the field.

Bibliography

[1] R. Sharma, S. Lajoie, and A. Dube, *Game Design for Mathematics Education*, 10 2022, pp. 25–37.

[2] D. A. Lieberman, *Interactive Video Games for Health Promotion: Effects on Knowledge, Self-Efficacy, Social Support, and Health*. Routledge, 1997, pp. 103–120, chapter 6. [Online]. Available: https://doi.org/10.4324/9780203811047

[3] M. Sailer and L. Homner, "The gamification of learning: a meta-analysis," *Educational Psychology Review*, vol. 32, no. 1, pp. 77–112, 2020. [Online]. Available: https://doi.org/10.1007/s10648-019-09498-w

[4] W. C. McGaghie, S. B. Issenberg, E. R. Cohen, J. H. Barsuk, and D. B. Wayne, "Does simulation-based medical education with deliberate practice yield better results than traditional clinical education? a meta-analytic comparative review of the evidence," *Academic Medicine*, vol. 86, no. 6, pp. 706–711, 2011. [Online]. Available: https://doi.org/10.1097/ACM.0b013e318217e119

[5] G. R. White, G. Fitzpatrick, and G. McAllister, "Toward accessible 3d virtual environments for the blind and visually impaired," in *Proceedings of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts*, ser. DIMEA '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 134–141. [Online]. Available: https://doi.org/10.1145/1413634.1413663

[6] S. Maidenbaum, S. Levy-Tzedek, D.-R. Chebat, and A. Amedi, "Increasing accessibility to the blind of virtual environments, using a virtual mobility aid based on the "eyecane": Feasibility study," *PLOS ONE*, vol. 8, no. 8, 08 2013. [Online]. Available: https://doi.org/10.1371/journal.pone.0072555

[7] S. M. Trewin, M. R. Laff, A. Cavender, and V. L. Hanson, "Accessibility in virtual worlds," in *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 2727–2732. [Online]. Available: https://doi.org/10.1145/1358628.1358752

[8] S. Maidenbaum, S. Abboud, G. Buchs, and A. Amedi, "Blind in a virtual world: Using sensory substitution for generically increasing the accessibility of graphical virtual

environments," in *2015 IEEE Virtual Reality (VR)*, 2015, pp. 233–234. [Online]. Available: https://doi.org/10.1109/VR.2015.7223381

[9] N. I. Ghali, O. Soluiman, N. El-Bendary, T. M. Nassef, S. A. Ahmed, Y. M. Elbarawy, and A. E. Hassanien, *Virtual Reality Technology for Blind and Visual Impaired People: Reviews and Recent Advances*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 363–385. [Online]. Available: https://doi.org/10.1007/978-3-642-23363-0_15

[10] M. Piçarra, A. Rodrigues, and J. a. Guerreiro, "Evaluating accessible navigation for blind people in virtual environments," in *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3544549.3585813

[11] B. Yuan, E. Folmer, and F. C. Harris, "Game accessibility: a survey," *Universal Access in the Information Society*, vol. 10, no. 1, pp. 81–100, 2011. [Online]. Available: https://doi.org/10.1007/s10209-010-0189-5

[12] M. Lumbreras and J. Sánchez, "Interactive 3d sound hyperstories for blind children," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '99. New York, NY, USA: Association for Computing Machinery, 1999, p. 318–325. [Online]. Available: https://doi.org/10.1145/302979.303101

[13] M. Ohuchi, Y. Iwaya, Y. Suzuki, and T. Munekata, "A comparative study of sound localization acuity of congenital blind and sighted people," *Acoustical Science and Technology*, vol. 27, pp. 290–293, 09 2006. [Online]. Available: https://doi.org/10.1250/ast.27.290

[14] G. Wersényi, "Virtual localization by blind persons," *Journal of the Audio Engineering Society*, vol. 60, no. 7/8, pp. 568–579, July 2012.

[15] A. Kolarik, S. Cirstea, S. Pardhan, and B. Moore, "An assessment of virtual auditory distance judgments among blind and sighted listeners," *The Journal of the Acoustical Society of America*, vol. 133, p. 3287, 05 2013. [Online]. Available: https://doi.org/10.1121/1.4805400

[16] J. Guerreiro, Y. Kim, R. Nogueira, S. Chung, A. Rodrigues, and U. Oh, "The design space of the auditory representation of objects and their behaviours in virtual reality for blind people," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 5, pp. 2763–2773, 2023. [Online]. Available: https://doi.org/10.1109/TVCG.2023.3247094

[17] C. Mangiron and X. Zhang, *Game Accessibility for the Blind: Current Overview and the Potential Application of Audio Description as the Way Forward*. London: Palgrave Macmillan UK, 2016, pp. 75–95. [Online]. Available: https://doi.org/10.1057/978-1-137-56917-2_5

[18] L. Evett, D. Brown, S. Battersby, A. Ridley, and P. Smith, “Accessible virtual environments for people who are blind: Creating an intelligent virtual cane using the nintendo wii controller,” *Proceedings of the Seventh International Conference on Disability, Virtual Reality and Associated Technologies*, 01 2008.

[19] V. Nair, J. L. Karp, S. Silverman, M. Kalra, H. Lehv, F. Jamil, and B. A. Smith, “Navstick: Making video games blind-accessible via the ability to look around,” in *The 34th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 538–551. [Online]. Available: https://doi.org/10.1145/3472749.3474768

[20] M. Lumbreras and J. Sánchez, “Interactive 3d sound hyperstories for blind children,” in *Proceedings of the ACM CHI Conference*. ACM, 1999, pp. 318–325. [Online]. Available: https://doi.org/10.1145/302979.303101

[21] B. Yuan and E. Folmer, “Blind hero: Enabling guitar hero for the visually impaired,” in *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility*, ser. Assets ’08. New York, NY, USA: Association for Computing Machinery, 2008, p. 169–176. [Online]. Available: https://doi.org/10.1145/1414471.1414503

[22] J. W. Strutt, “On the perception of the direction of sound,” *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 83, no. 559, pp. 61–64, 1909. [Online]. Available: https://doi.org/10.1098/rspa.1909.0073

[23] W. Yost, *Fundamentals of Hearing: An Introduction: Fifth Edition*. Brill, 2021.

[24] J. C. Middlebrooks and D. M. Green, “Sound localization by human listeners,” *Annual Review of Psychology*, vol. 42, no. 1, pp. 135–159, 1991, pMID: 2018391. [Online]. Available: https://doi.org/10.1146/annurev.ps.42.020191.001031

[25] J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization*. The MIT Press, 10 1996. [Online]. Available: https://doi.org/10.7551/mitpress/6391.001.0001

[26] T. Letowski and S. Letowski, “Auditory spatial perception: Auditory localization,” 05 2012.

[27] D. W. Batteau and H. E. Huxley, “The role of the pinna in human localization,” *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 168, no. 1011, pp. 158–180, 1967. [Online]. Available: https://doi.org/10.1098/rspb.1967.0058

[28] A. D. Musicant and R. A. Butler, “The influence of pinnae-based spectral cues on sound localization,” *Journal of the Acoustical Society of America*, vol. 75, no. 4, pp. 1195–1200, Apr 1984. [Online]. Available: https://doi.org/10.1121/1.390770

[29] D. R. Begault, E. M. Wenzel, and M. R. Anderson, “Direct comparison of the impact of head tracking, reverberation, and individualized head-related transfer functions on the spatial perception of a virtual speech source,” *Journal of the Audio Engineering Society*, vol. 49, no. 10, pp. 904–916, Oct 2001.

[30] R. So, B. Ngan, A. Horner, K. Leung, J. Braasch, and J. Blauert, “Toward orthogonal non-individualized head-related transfer functions for forward and backward directional sound: cluster analysis and an experimental study,” *Ergonomics*, vol. 53, no. 6, pp. 767–781, 2010. [Online]. Available: https://doi.org/10.1080/00140131003675117

[31] P. M. Hofman, J. van Riswick, and A. J. van Opstal, “Relearning sound localization with new ears,” *Nature Neuroscience*, vol. 1, pp. 417–421, 1998. [Online]. Available: https://doi.org/10.1038/1633

[32] D. N. Zotkin, R. Duraiswami, and N. A. Gumerov, “Regularized hrtf fitting using spherical harmonics,” in *2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2009, pp. 257–260. [Online]. Available: https://doi.org/10.1109/ASPAA.2009.5346521

[33] A. Kan, C. Jin, and A. van Schaik, “A psychophysical evaluation of near-field head-related transfer functions synthesized using a distance variation function,” *The Journal of the Acoustical Society of America*, vol. 125, no. 4, pp. 2233–2242, 2009. [Online]. Available: https://doi.org/10.1121/1.3081395

[34] A. Oppenheim, A. Willsky, and S. Nawab, *Signals & Systems*, ser. Prentice-Hall signal processing series. Prentice-Hall International, 1997.

[35] S. Li and J. Peissig, “Measurement of head-related transfer functions: A review,” *Applied Sciences*, vol. 10, no. 14, 2020. [Online]. Available: https://doi.org/10.3390/app10145014

[36] R. Bracewell, *The Fourier Transform and Its Applications*, ser. Circuits and systems. McGraw Hill, 2000.

[37] R. O. Duda and W. L. Martens, “Range dependence of the response of a spherical head model,” *The Journal of the Acoustical Society of America*, vol. 104, no. 5, pp. 3048–3058, 11 1998. [Online]. Available: https://doi.org/10.1121/1.423886

[38] B. Bates, *Game Design: The Art & Business of Creating Games*, ser. Game Development Series. Prima Tech, 2001.

[39] C. Martinho, R. Prada, and P. Santos, *Design e Desenvolvimento de Jogos*. Portugal: FCA Editora de Informática, Lda., 2016.

[40] R. Urick, *Principles of Underwater Sound*. Peninsula Publishing, 1983.

[41] E. Lengyel, *Foundations of Game Engine Development*, ser. Foundations of Game Engine Development. Terathon Software LLC, 2016, no. vol. 1.

[42] T. Fullerton, *Game Design Workshop: A Playcentric Approach to Creating Innovative Games, Fourth Edition*, 4th ed. A K Peters/CRC Press, 2018. [Online]. Available: https://doi.org/10.1201/b22309

[43] A. S. Lu, T. Baranowski, D. Thompson, and R. Buday, "Story immersion of videogames for youth health promotion: A review of literature," *Games for Health Journal*, vol. 1, no. 3, pp. 199–204, Jun 2012. [Online]. Available: https://doi.org/10.1089/g4h.2011.0012

[44] M. C. Green, T. C. Brock, and G. F. Kaufman, "Understanding media enjoyment: The role of transportation into narrative worlds," *Communication Theory*, vol. 14, no. 4, pp. 311–327, 2004. [Online]. Available: https://doi.org/10.1111/j.1468-2885.2004.tb00317.x

[45] T. Dubbelman, "Playing the hero: How games take the concept of storytelling from representation to presentation," *Journal of Media Practice*, vol. 12, no. 2, pp. 157–172, 2011. [Online]. Available: https://doi.org/10.1386/jmpr.12.2.157_1

[46] R.-Y. Zhang, A. Chopin, K. Shibata, Z.-L. Lu, S. M. Jaeggi, M. Buschkuehl, C. S. Green, and D. Bavelier, "Action video game play facilitates "learning to learn"," *Communications Biology*, vol. 4, no. 1, p. 1154, 2021. [Online]. Available: https://doi.org/10.1038/s42003-021-02652-7

[47] J. Schell and J. Schell, *The Art of Game Design: A book of lenses*, 1st ed. CRC Press, 2008. [Online]. Available: https://doi.org/10.1201/9780080919171

[48] C. W. Turner, J. R. Lewis, and J. Nielsen, *Determining usability test sample size*. CRC Press, 2015.

[49] E. S. Association. (2022) 2022 essential facts about the video game industry. Entertainment Software Association. [Online]. Available: https://www.theesa.com/2022-essential-facts-about-the-video-game-industry/

[50] P. Royston, "An extension of shapiro and wilk's w test for normality to large samples," *Applied Statistics*, vol. 31, no. 2, pp. 115–124, 1982. [Online]. Available: https://doi.org/10.2307/2347973

[51] H. Ebbinghaus, *Memory; a contribution to experimental psychology*. Teachers College, Columbia University, 1909.

[52] J. J. Donovan and D. J. Radosevich, "A meta-analytic review of the distribution of practice effect: Now you see it, now you don't," *Journal of Applied Psychology*, vol. 84, no. 5, pp. 795–805, 1999. [Online]. Available: https://doi.org/10.1037/0021-9010.84.5.795

[53] S. G. Kwak and J. H. Kim, “Central limit theorem: the cornerstone of modern statistics,” *Korean journal of anesthesiology*, vol. 70, no. 2, pp. 144–156, 2017. [Online]. Available: https://doi.org/10.4097/kjae.2017.70.2.144

[54] G. M. Sullivan and A. R. Artino Jr, “Analyzing and interpreting data from likert-type scales,” *Journal of graduate medical education*, vol. 5, no. 4, pp. 541–542, 2013. [Online]. Available: https://doi.org/10.4300/JGME-5-4-18

[55] E. B. Goldstein and J. R. Brockmole, *Sensation and Perception*, 10th ed. Cengage Learning, 2016, chapter 1.

A

High Concept

The nauds are invading again. They left you blind and your hearing will be our salvation. Hunt them with your weapon, restore life to survive and the built-in tracker will aid you finding our enemies. It's time to show that we command our FATE and deliver. . . **ANNIHILATION**!

A.1 Features

- Be ready for a war where you'll have to rely on your hearing as you embody a blind elite soldier in a new and unique single game play experience.
- Play in a *On Rail Shooter* structure where you'll have to fight your enemies to escape the fate of certain death.
- From a FPS perspective with a fast paced action you'll have to act fast in order to survive a high diversity of nauds and adapt to their characteristics in order to overcome them.
- You'll be able to acquire powerful weapons, useful trackers and Hi-Tech Shields to support you in your darkest hour.
- Great feeling of progression with many levels to explore and rewards to earn
- Curated and customized selection of sound effects in order to provide the most immersive sound experience.

A.2 Overview

A.2.1 Player Motivation

Kill or be killed, the fate of human race depends on it. Prove your valour as an Elite Soldier be fast and precise, rely on your hearing that can mean the difference between death and life.

A.2.2 Genre

FATE ANNIHILATION is a 3D *On Rail Shooter* game, with elements from the FPS genre such as fast paced action, interaction and exploration.

A.2.3 Target Audience

- Visually impaired players that love fast paced games with a lot of interaction.
- FPS lovers looking for a new flavor.

A.2.4 Competition

- *Shade of Doom*
- *AudioQuake*

A.2.5 Unique Selling Points

- Lots of enemies to fight and rewards to obtain.
- Interaction and fast paced action are key elements.
- The variety of tools provide players different strategies.
- Intuitive progression.
- Use your skill to advance through the level.
- Choose your own path.

A.2.6 Hardware

- Target platform is Personal Computer (PC).
- Game controller is required.

A.2.7 Design Goals

- Fast paced: An action game where you'll need to be fast to survive.
- Diverse: High diversity of enemies which will require the player to come up with different strategies.

B

Formative Usability Tests

B.1 Demographic Questions

1. What is your age?
2. What is your gender?
   (a) Male
   (b) Female
   (c) Rather not say
   (d) Other
3. How often do you play video games?
   (a) Don't play
   (b) 1 to 3 Hours per Week
   (c) 7 or more Hours per Week
4. Are you familiar with Rail Shooters?
   (a) Yes
   (b) No

B.2 Answers

Table B.1: Participants Answers

Id	1	2	3	4
P1	30	a)	b)	a)
P2	62	a)	a)	b)
P3	40	a)	b)	a)
P4	25	a)	b)	a)
P5	33	a)	b)	a)
P6	35	a)	b)	a)
P7	23	a)	b)	a)
P8	29	b)	a)	b)
P9	34	a)	b)	a)
P10	23	b)	a)	b)
P11	33	b)	a)	b)
P12	48	b)	a)	b)
P13	35	b)	b)	b)

B.3 Usability Questions

1. Was it easy to identify and pick up the health kits?
   (a) Strongly disagree
   (b) Disagree
   (c) Neutral
   (d) Agree
   (e) Strongly Agree
2. Was it easy to identify and pick up all of the ammo boxes?
   (a) Strongly disagree
   (b) Disagree
   (c) Neutral
   (d) Agree
   (e) Strongly Agree
3. Did you complete the target shooting challenge?
   (a) Yes
   (b) No
4. How difficult do you rate the target shooting challenge?
   (a) Strongly disagree
   (b) Disagree
   (c) Neutral
   (d) Agree
   (e) Strongly Agree
5. Did you complete the main hall section?
   (a) Yes
   (b) No
6. How difficult do you rate the main hall section?
   (a) Strongly disagree
   (b) Disagree
   (c) Neutral
   (d) Agree
   (e) Strongly Agree
7. Did the sound effects help you better understand the game environment?
   (a) Strongly disagree
   (b) Disagree
   (c) Neutral
   (d) Agree
   (e) Strongly Agree
8. Did you ever have any problems understanding what to do next?
   (a) Yes
   (b) No
9. If so, when?
10. Do you have any suggestions to improve the game play of the prototype?

B.4 Usability Answers

Table B.2: Participants answers to usability questions Q1 to Q8

Participant	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
P1	e)	d)	a)	d)	a)	b)	e)	b)
P2	c)	b)	a)	e)	b)	a)	e)	b)
P3	e)	e)	a)	b)	b)	b)	e)	b)
P4	e)	e)	a)	e)	a)	c)	e)	b)
P5	b)	c)	a)	e)	a)	b)	e)	a)
P6	d)	d)	a)	d)	a)	c)	e)	b)
P7	e)	e)	a)	d)	a)	c)	e)	b)
P8	b)	a)	b)	a)	b)	a)	c)	a)
P9	e)	e)	a)	e)	a)	d)	d)	a)
P10	b)	a)	a)	b)	b)	a)	d)	a)
P11	d)	d)	a)	e)	b)	a)	e)	a)
P12	a)	a)	a)	a)	b)	a)	c)	a)
P13	e)	d)	a)	b)	a)	b)	e)	b)

Table B.3: Participants answers to questions Q9 and Q10.

Participant	Question 9	Question 10
P3	N. A. *	O final é um pouco abrupto. Algo a indicar que o jogo terminou e o jogador regressou ao menu ajudava a tornar o final da experiência menos confuso. Mas de resto está muito bem conseguido
P5	Foi dificil identificar as áreas do nível em que estava	Devia de haver instruções na primeira utilização e maior feedback nas ações
P8	Tive dificuldade em entender que tinha de ganhar vida mais do que uma vez e na parte de matar os inimigos	Indicar sonoramente os objectivos do que fazer a seguir tipo mensagens de voz
P9	Na parte de ganhar vida não tinha percebido que era necessário apanhar mais do que uma vez vida para progredir no jogo	N. A.
P10	Não entendi que precisava de apanhar vida para passar ao próximo nível	N. A.
P11	Não percebia bem o que tinha de fazer em cada parte	Criar instruções sonoras para cada desafio
P12	Quando deveria disparar e apanhar os objectos	Mais informação sonora sobre o que está a acontecer

* *N.A. stands for **N**ot **A**pplicable.*

C

Summative User Tests

C.1 Demographic Questions

1. What is your age?
2. How long have you been blind??
3. What is your gender?
    (a) Male
    (b) Female
    (c) Rather not say
    (d) Other
4. How often do you play video games?
    (a) Don't play
    (b) 1 to 3 Hours per Week
    (c) 7 or more Hours per Week
5. Are you familiar with Rail Shooters?
    (a) Yes
    (b) No

C.2 Demographic Answers

Table C.1: Participants Answers

Participant	1	2	3	4	5
P1	63	60	a)	c)	b)
P2	39	39	a)	a)	b)
P3	40	40	b)	b)	b)
P4	35	25	a)	b)	a)
P5	60	4	b)	b)	b)
P6	40	40	b)	b)	b)
P7	38	6	a)	a)	a)
P8	37	27	a)	a)	b)
P9	32	22	b)	b)	b)
P10	26	26	b)	a)	b)

C.3 Satisfaction Survey Questions

1. Did the game capture your attention?
   (a) Strongly Disagree
   (b) Disagree
   (c) Neutral
   (d) Agree
   (e) Strongly Agree
2. How much effort did you put into playing the game?
   (a) Strongly Disagree
   (b) Disagree
   (c) Neutral
   (d) Agree
   (e) Strongly Agree
3. Did you feel like you gave your best?
   (a) Strongly Disagree
   (b) Disagree
   (c) Neutral
   (d) Agree
   (e) Strongly Agree
4. Did you lose track of time?
   (a) Strongly Disagree
   (b) Disagree
   (c) Neutral
   (d) Agree
   (e) Strongly Agree
5. Did you feel the need to understand what was happening around you?
   (a) Strongly Disagree
   (b) Disagree
   (c) Neutral
   (d) Agree
   (e) Strongly Agree
6. Did you find the game challenging?
   (a) Strongly Disagree
   (b) Disagree
   (c) Neutral
   (d) Agree
   (e) Strongly Agree

7. How well do you think your session went?
    (a) Strongly Disagree
    (b) Disagree
    (c) Neutral
    (d) Agree
    (e) Strongly Agree
8. Do you feel emotionally connected to the game?
    (a) Strongly Disagree
    (b) Disagree
    (c) Neutral
    (d) Agree
    (e) Strongly Agree
9. How much did you enjoy the game?
    (a) Strongly Disagree
    (b) Disagree
    (c) Neutral
    (d) Agree
    (e) Strongly Agree
10. Would you play the game again?
    (a) Strongly Disagree
    (b) Disagree
    (c) Neutral
    (d) Agree
    (e) Strongly Agree

C.4 Answers Survey Questions

Table C.2: Participants Answers

Participant	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
P1	e)	c)	d)	c)	a)	e)	e)	e)	e)	e)
P2	e)	e)	e)	e)	a)	e)	d)	e)	e)	e)
P3	e)	e)	e)	e)	a)	e)	e)	e)	e)	e)
P4	e)	e)	d)	b)	e)	e)	d)	e)	e)	e)
P5	c)	d)	e)	a)	a)	d)	c)	d)	e)	e)
P6	d)	d)	e)	e)	d)	e)	d)	d)	e)	e)
P7	e)	e)	c)	e)	e)	d)	d)	d)	d)	e)
P8	c)	d)	e)	c)	d)	e)	c)	c)	d)	e)
P9	e)	b)	e)	d)	d)	c)	e)	c)	d)	e)
P10	d)	e)	d)	c)	a)	e)	d)	d)	e)	e)

D

Guião para Testes com Utilizadores

Bem vindo a esta experiência, o meu nome é Daniel Rodrigues e sou aluno do Instituto Superior Técnico e estou a fazer uma dissertação que pretende melhorar a experiência de pessoas com deficiência visual ou visibilidade reduzida com ambientes virtuais. Para isso desenvolvi um protótipo de um jogo que vai ser testado nesta sessão. A sessão é composta por quatro fases:

- Breve introdução e primeiro contacto com o protótipo
- Teste com protótipo
- Treino com uma mecânica desenvolvida no âmbito da tese
- Novo teste com protótipo e nova mecânica

A sessão terá uma duração de cerca de 15 a 30 minutos sendo que poderá interromper ou desistir a qualquer momento. Vamos então começar . . .

D.1 Introdução Protótipo

Este é um protótipo de um jogo de ação mais especificamente de um *Rail Shooter* em que cada objeto é caracterizado por um som específico, ao longo do jogo os inimigos e objetos (como vida e munição) vão aparecer à sua frente, o jogador pode disparar para a esquerda pressionando o botão X, centro pressionando Y e direita pressionando B. Para interagir com objetos como apanhar vida ou munição, o jogador deve pressionar A. A munição não é infinita e devido ao alcance da arma os inimigos não podem ser atingidos enquanto não estiverem a uma distância intermédia do jogador. Os inimigos só morrem quando atingidos com três tiros. No protótipo há dois resultados possíveis, o jogador morre ou completa com sucesso o nível, ambos sinalizados por uma mensagem sonora. Pode experimentar o protótipo, para dar início basta carregar no botão de *Start*. Qualquer dúvida ou questão é só perguntar.

D.2 Teste com protótipo

Feita a introdução e primeiro contacto com o protótipo vamos agora dar início ao teste, pode carregar no botão de *Start*.

D.3 Treino

Nesta fase o jogador não precisa de realizar qualquer ação. Basta ouvir e sentir a vibração. Quando o treino terminar será sinalizado por uma mensagem sonora. Para dar início ao treino carregue no botão LT.

D.4 Novo Teste com protótipo

Novo teste com protótipo com a nova mecânica incorporada. Pressione RT para ativar a mecânica e depois pressione *Start* para iniciar o teste.

Terminamos a sessão de testes, espero que tenha gostado e agradeço a sua participação. Tem alguma sugestão ou comentário a fazer sobre a experiência?

Gostaria de lhe fazer apenas algumas perguntas: Usar questionario demográfico e de satisfação.