



Identificação Espacial de Som Surround em Jogos

JOÃO PEDRO NUNES DE MAGALHÃES

Junho de 2021

Spatial Identification System of Surround Sound in Games

João Pedro Nunes de Magalhães

**Dissertation for obtaining the Master's Degree in
Informatics Engineering, Specialization Area in
Graphics Systems and Multimedia**

Supervisor: Filipe de Faria Pacheco

Porto, June 2021

Identificação Espacial de Som Surround em Jogos

João Pedro Nunes de Magalhães

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas Gráficos e Multimédia**

Orientador: Filipe de Faria Pacheco

Porto, Junho 2021

Resumo

Este documento foi desenvolvido no âmbito da unidade curricular “Tese/Dissertação/Estágio” (TMDEI) com o objetivo de descrever todo o processo de desenvolvimento do projeto “Identificação Espacial de Som *Surround* em Jogos”.

A realização deste projeto parte da falta de soluções que permitam pessoas com dificuldades auditivas de aproveitar melhor o seu *gameplay*.

O objetivo principal do projeto é a elaboração de um sistema que permita a um utilizador visualizar informação de localização de sons relevantes em jogos em que esta informação é disponibilizada através de som surround.

Será elaborada uma análise ao valor deste sistema a fim de constatar as diferentes oportunidades e abordagens bem como caracterizar a ideia de negócio.

O desenvolvimento do projeto percorreu as diferentes fases da engenharia de software, nomeadamente, análise, levantamento de requisitos, design, implementação e testes.

Palavras-chave: Som *surround*, Jogos, Inclusividade, Acessibilidade.

Abstract

This document was developed within the scope of the "Thesis / Dissertation / Internship" (TMDEI) curricular unit to define the entire development process of the project "Spatial Identification of Surround Sound in Games".

The realization of the project stems from the lack of solutions that allow people with hearing difficulties to enjoy more their gameplay.

The main objective of the project is the development of a surround sound identification system in games with an interface that allows the visualization of information of the sound location.

An analysis of the value of this system will be carried out to verify different opportunities and approaches as well as to characterize the business idea.

The development of the project went through the different phases of software engineering, namely, requirements analysis, design, implementation, and testing.

Keywords: Surround sound, Games, Inclusivity, Accessibility.

Acknowledgments

To my supervisor, Professor Filipe de Faria Pacheco, I appreciate the availability to discuss the issues that arose during the development of the project and the help provided to resolve them.

I thank all my friends and colleagues for all the weary nights spent without sleep and the effort put in.

I thank my parents for the support given throughout the course and the development of this dissertation and all the adversities that have arisen over time.

And most of all I thank my sister, for being available every time her little brother bothered her and for believing in me.

Index

1	Introduction	1
1.1	Context	1
1.2	Problem.....	2
1.3	Objectives.....	2
1.4	Contributions	3
1.5	Types of Hearing Loss.....	3
1.6	Document Structure	4
2	State of the Art	7
2.1	Sound in Games.....	7
2.2	Surround Sound	8
2.2.1	Primary Cues for Auditory Localization	10
2.2.2	Surround Sound Enhancement	11
2.3	Related Works	11
2.4	Existing Technologies	14
2.4.1	Audio/Music Frameworks.....	15
2.4.2	Audio Mixing Software	16
2.4.3	Frontend Technologies	18
2.4.4	Programming Language.....	19
2.5	Existing Solutions.....	20
2.5.1	Sonic Radar 3	20
2.5.2	THX Spatial Audio (Razer 7.1 Surround Sound Software)	21
2.5.3	Graphic representation.....	22
3	Value Analysis	25
3.1	New Concept Development Model	25
3.2	Analytic Hierarchy Process (AHP)	28
3.2.1	Stage 1 - Construction of the Hierarchy for the Criteria	28
3.2.2	Stage 2 - Alternatives and criteria comparison	29
3.2.3	Stage 3 - Relative Priority.....	30
3.2.4	Stage 4 - Consistency	31
3.2.5	Stage 5 - Evaluation regarding the criterion chosen	32
3.3	Value Proposition.....	33
3.4	Business Model CANVAS.....	34
3.5	Porter's Value Chain	35
4	Approach and Design Discussion	37
4.1.1	Spatial Identification System of Surround Sound in Games	37
4.1.2	Interface/Information System	38

5	Design and Implementation	39
5.1	Architecture	39
5.2	Functional and Non Functional Requirements	40
5.3	Implementation	41
5.3.1	Spatial Identification System of Surround Sound in Games	41
5.3.1	Interface/Information System.....	48
6	Testing and Evaluation	53
6.1	Hypothesis	53
6.2	Evaluation Criteria	54
6.3	Evaluation Methodology	54
6.3.1	Software Experience Questionnaire	55
6.3.2	System Usability Scale (SUS).....	56
6.4	Final Results	57
6.4.1	Software Experience Questionnaire Results.....	57
6.4.1	System Usability Scale Results	62
6.4.1	Hypothesis Validation.....	64
7	Conclusion	67
8	References.....	69
	Annex 1 - Software Experience Sample Survey	75
	Annex 2 - System Usability Scale Sample Survey	82
	Annex 3 - Software Experience Questionnaire Results	86
	Annex 4 - System Usability Scale Results.....	91
	Annex 5 - Session Guide	92

Figure List

Figure 1: Reference loudspeaker arrangement with loudspeakers L/C/R and LS/RS (ITU, 2012)	9
Figure 2: Dialogue in The Elder Scrolls V: Skyrim.....	12
Figure 3: Sound visualization in Fortnite Battle Royale.....	12
Figure 4: Gameplay from Assassin's Creed Valhalla.....	13
Figure 5: Gameplay from Cyberpunk 2077.....	13
Figure 6: Gameplay from Assassin's Creed IV: Black Flag.....	14
Figure 7: Angular vs React downloads per Year (Reis, 2020).....	19
Figure 8: Sonic Radar 3 used in Counter-Strike: Global Offensive.....	21
Figure 9: Window showing radar from THX Spatial Audio software.....	22
Figure 10: Examples of the possible graphic representations.....	22
Figure 11: Product Innovation Process (Koen, et al., 2001).....	26
Figure 12: The New Concept Development Model (Koen, et al., 2001).....	26
Figure 13: Hierarchical Decision Tree - AHP.....	29
Figure 14: Business Model CANVAS.....	35
Figure 15: Component Diagram.....	40
Figure 16: Use Case Diagram.....	41
Figure 17: level_meter file unaltered.....	43
Figure 18: Default Device Setup.....	44
Figure 19: Voicemeeter Banana Setup.....	45
Figure 20: level_meter prints both input channels output.....	46
Figure 21: 8x8 Matrix Setup.....	46
Figure 22: get_second_device method.....	47
Figure 23: Process of obtaining the loudest channel.....	49
Figure 24: Flow Chart with the logic behind the calculation of the right direction.....	50
Figure 25: Examples of the output from all the possible directions.....	51
Figure 26: System Usability Scale equation.....	56
Figure 27: System Usability Scale Questionnaire (Bangor, et al., 2008).....	56
Figure 28: Age and Sex question results.....	58
Figure 29: Hearing loss question results.....	58
Figure 30: Hours spent playing games on pc per week question results.....	59
Figure 31: Counter-Strike: Global Offensive results.....	59
Figure 32: Isolation issues question results.....	60
Figure 33: Identify the origin of sound question results with sound (1) versus without sound and with the software (2).....	61
Figure 34: Frames Per Second question results.....	62
Figure 35: Sus Question 1 Results.....	63

Table List

Table 1: World Health Organisation Grades of Hearing Impairment	3
Table 2: Comparison Virtual Surround Card vs Voicemeeter vs Voicemeeter Banana	17
Table 3: The Fundamental Scale - AHP (Saaty & Wind, 1980)	30
Table 4: Matrix of pairwise comparisons of the criteria - AHP	30
Table 5: Sum of columns from the matrix from Stage 2 - AHP	30
Table 6: Normalized Matrix Table from Stage 2 - AHP.....	31
Table 7: Relative Priority - AHP	31
Table 8: Random Consistency Index Table - AHP (Saaty, 1987).....	32
Table 9: Conversion of qualitative dimension to numerical value - AHP.....	32
Table 10: Qualitative score for each idea – AHP (Lalic & Palcic, 2009).....	33
Table 11: Final score for each idea – AHP	33
Table 12: SUS Questionnaire Average Results.....	63
Table 13: Conversion of answers into a numerical value.....	64
Table 14: t-test results obtained.....	64
Table 15: Questions related to the session with sound.....	88
Table 16: Questions related to the session using software without sound.....	89

Acronyms

Acronyms List

AHP	Analytic Hierarchy Process
FEI	Front End of Innovation
FPS	First-Person Shooter
HRTF	Head-Related Transfer Function
ILD	Interaural Level Difference
ITD	Interaural Time Difference
NCD	New Concept Development
NPPD	New Product and Process Development
MIR	Music Information Retrieval
SUS	System Usability Scale

1 Introduction

The main purpose of this document, which refers to Tese/Dissertação/Estágio (TMDEI) curricular unit, is to answer questions related to the system developed as well as all the components that make up the practical solution, that aims to use information obtained from surround sound to improve the interaction with people with different degrees of hearing loss.

1.1 Context

The evolution of technology allows for different disabilities to be less determining in the way the people affected live their lives. The gaming industry has been increasing in popularity with each year that passes and, consequently, so has the player base (Accenture, 2021). Video games allow the players to be anything or anyone they want, to run, jump or even fly. The player base includes a vast diversity of people, so people with disabilities also have a right to belong, even though video games are still looking for better ways to facilitate these players' gameplay (Bierre, et al., 2005). But to facilitate the adaptation, the industry uses different tools, both hardware or software, to reach every person.

The project developed within the scope of this dissertation looks to contribute to the diminishment of hearing-related disadvantages in video games for PC and the improvement of immersion for these types of users.

1.2 Problem

One of the key elements of communication between a game and the player is by sound, where important information is passed to the user about the main goal, it being either an enemy or a destination (Collins, 2008). The use of surround sound allows for a more immersive experience that allows more intensive and clearer gameplay. In a multiplayer First-Person Shooter (FPS) it can be the deciding factor between victory and defeat since auditory information often carries artifacts of the user surroundings (Grimshaw, 2004).

Knowing this it is understandable that deaf and hearing-impaired players end up having a disadvantage and less enjoy the walkthrough. The implementation of a system that will convey this crucial information for this type of user will help minimize this disadvantage and try to normalize the experience in-game.

1.3 Objectives

The main objective of this project is to develop a spatial identification system of surround sound in games that will allow users to visualize the information through eyesight instead of from hearing.

To achieve this main goal other objectives are defined:

- Implementation of a spatial identification system of surround sound in games.
- Implementation of an interface that conveys the information of the spatial origin of surround sound to the players.
- Testing Phase to validate the accuracy of the system implemented.

With the implementation of this project, it is expected that the system is accurate and assesses how it helps players with hearing disabilities to better their experience and gameplay in games.

1.4 Contributions

The development of this system will focus on improving the ability to identify the direction of the sound of deaf and hearing-impaired players, improving the overall experience and gameplay of these users. The system will allow each player to visualize the direction of sound through a radar while playing.

The process of developing an original idea and accomplishing it will allow the student to develop capabilities and improve his sense of responsibility and confidence.

1.5 Types of Hearing Loss

Hearing impairment refers to the loss of the ability to hear from a range from close slight impairment to total deafness. The World Health Organization classifies the severity of hearing loss in five grades (WHO, 2008) and is presented in Table 1.

Table 1: World Health Organisation Grades of Hearing Impairment

Grade of impairment*	Corresponding audiometric ISO value**	Performance	Recommendations
0 - No impairment	25 dB or better (better ear)	No or very slight hearing problems. Able to hear whispers.	
1 – Slight impairment	26-40 dB (better ear)	Able to hear and repeat words spoken in normal voice at 1 metre.	Counselling. Hearing aids may be needed.
2 - Moderate impairment	41-60 dB (better ear)	Able to hear and repeat words spoken in raised voice at 1 metre.	Hearing aids usually recommended.
3 - Severe impairment	61-80 dB (better ear)	Able to hear some words when shouted into better ear.	Hearing aids needed. If no hearing aids available, lip-reading and signing should be taught.
4 - Profound impairment including deafness	81 dB or greater (better ear)	Unable to hear and understand even a shouted voice.	Hearing aids may help understanding words. Additional rehabilitation needed. Lip-reading and sometimes signing essential.

* Grades 2, 3, and 4 are classified as disabling hearing impairment (for children, it starts at 31 dB)

** The audiometric ISO is calculated separately for each ear as the mean value of hearing loss for the four frequencies 500 Hz, 1000 Hz, 2000 Hz, and 4000 Hz.

This grade takes into account the biggest hearing loss from four frequencies; 500Hz, 1000 Hz, 2000 Hz, and 4000 Hz. This also indicates the existence of other types of deafness related to the frequencies. It is possible to single out people that have difficulty hearing bass sounds (low frequencies) or high sounds (high frequencies). It is also possible to single out people that are only deaf from one ear and not the other.

Another important aspect of hearing impairment is the age factor. A 5-year study conducted analyzing 2837 adults in the United States determined the incidence of hearing loss by age group. The information gathered goes as follows:

- Ages 21 to 34: 2.9%
- Ages 35 to 44: 6.4%
- Ages 44 to 54: 10.9%
- Ages 55 to 64: 25.1%
- Ages 65 to 84: 42.7%

The conclusion taken is that from ages 20 onward, the hearing ability will lessen through time and the percentage starts being more prominent from ages 40 onward (Nash SD, 2011).

1.6 Document Structure

This document is composed of seven chapters that compose the main body of the dissertation, followed by a chapter of References and a chapter of Annexes.

The first chapter has the main goal to introduce to the reader the main problem that it aims to fix while giving the proper context, objectives, and contributions.

In the second chapter, a gathering of information related to the theme of this project is carried out, to access the best way to approach the development and which steps to follow while developing.

The third chapter has the goal to evaluate the business side of the project using different tools that will make clear the best business decision to make. It also describes the business idea and its respective value.

The fourth chapter was written to evaluate and decide which was the best option regarding the design of the system. Alternatives are analyzed and compared to each other to decide the best way to implement the components.

The fifth chapter gives a thorough explanation of the process and decisions made in the development process. The functionalities and the needed means to achieve them are also described.

The sixth chapter evaluates the hypothesis that will serve as the main tool to verify the success of the system. Here are also discussed the results obtained from the questionnaires from the testing phase.

The seventh and final chapter will present the conclusions taken away from the development of the system presented in this dissertation.

After this chapter, there are two sections, the Reference section that enumerates the utilized references used during the writing process of this document and an Attachment section that complement the information presented in the document.

2 State of the Art

In this chapter, important topics related to the development of this project will be covered to better understand the need for software like this and how different games have approached the issue.

2.1 Sound in Games

In the early 1970s, games start to be commercially sold, and with the early decade classic Pong (1972), the first uses of sound in-games use different pitches and duration to differentiate between a ball bouncing off a wall, the ball hit by a player's paddle or a point being scored (Collins, 2008).

With the evolution of the industry, games started to introduce background music so players would be entertained during gameplay. By the time games start becoming 3D and no longer need to be played linearly, the sound is used to improve the sense of immersion of the player and is used to give clues in which the player relies upon to have success in his walkthrough (Collins, 2008).

This is where the introduction of surround sound comes as a helpful tool to improve even more the immersion of the player, enabling the player to experience sound coming from behind him and not just the usual left and right. This allows the player to react faster and more accurately to the environment set by the game. In First-Person Shooters, the use of surround sound is an essential tool for the success of any player since the gathering of information in real-time can be the difference between winning and losing (Grimshaw, 2004).

2.2 Surround Sound

Surround sound is a way to reproduce sound that uses several audio channels from speakers that surround the listener in a way that gives a better and more immersive experience (ITU, 2012). This system is mostly used in movie or home theaters and allows the user to recognize with more precision the origin of the sounds. Another use for this system is in videogames, especially in consoles, due to the equipment which allows the user to easily set it up with some sound columns, on the contrary to a computer.

The main goal for the development of this software is to localize the sound so the user, that is unable to get this information himself, can enjoy, at least, a little bit more the gameplay and “sort of” normalize the experience through the visualization on a radar in his cellphone or computer. To achieve this, the use of a surround sound card and a virtual audio mixer is crucial to identify the sound intensity originating from each channel of a surround system. After this gathering of information, a series of calculations are needed so when two or more channels are active at the same time the radar presented to the user will show a combined origin for that particular moment.

The most common configuration is the International Telecommunication Union’s (ITU) 5.1 surround, which means 5 full-range channels and 1 Low-Frequency Effects (LFE) channel, but several configurations exceed the 7-channel mark and even one configuration that requires 24 speakers called 22.2 surround sound (NHK World, 2011).

The 5.1 configuration requires 6 speakers, one located in front of the listener called Center(C), two to his left, one located at a 60° degree angle (L - Left), and one located between a 100° and 120° degree angle (LS - Left Surround), two to his right that mirror the left channels, one located at a 60° degree angle (R - Right) and one located between a 100° and 120° degree angle (RS - Right Surround) and a subwoofer whose position is not critical (ITU, 2012). The configuration shown in Figure 1 is the most common and recommended, although it is possible to use a 5.1 system with a different configuration.

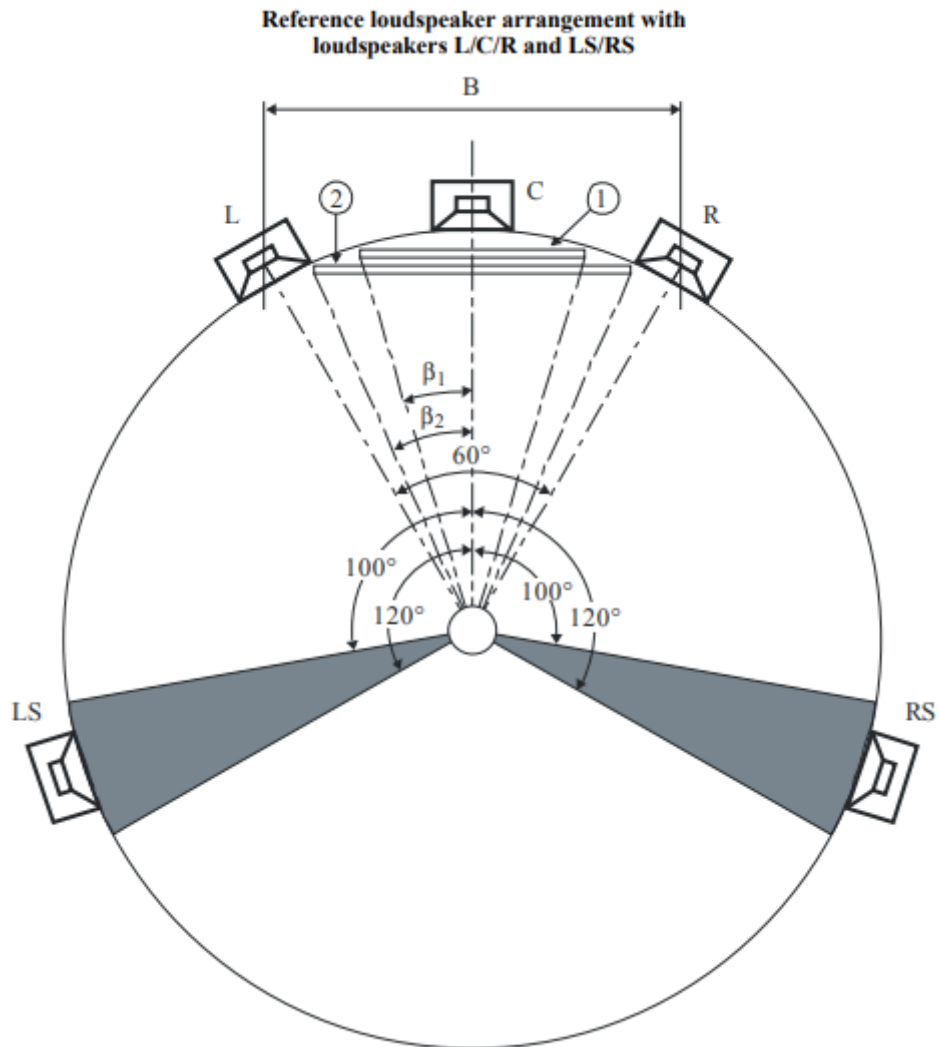


Figure 1: Reference loudspeaker arrangement with loudspeakers L/C/R and LS/RS (ITU, 2012)

Virtual Surround is a method that simulates the effects of true surround sound with fewer sources of sound. This technique uses different ways of tricking the human auditory system into thinking that the origin of the sound is coming from different angles by using frequency changes or slight delays (Santos, 2020). Most First-Person Shooter (FPS) games such as Counter-Strike: Global Offensive (CS: GO) (Valve Corporation, 2016), Overwatch (Messner, 2016), and Rainbow Six Siege (Park, 2020) use this kind of sound. Since most players, in every level of competitive play, use a standard stereo headset that cancels out other noises, this approach is the easier and most fruitful option.

2.2.1 Primary Cues for Auditory Localization

The location for sound needs to be calculated through external cues such as Interaural Time and Level Difference (ITD and ILD) and Head-Related Transfer Function (HRTF).

Interaural Cues

It has been known for a long time that one of the principal cues for sound localization is the time and level difference at the ears of the listener (Rayleigh, 1907). ITD represents the difference between the times the sound reaches both ears. So, when the distance from both ears is the same there is no difference in time and when the source is to the side of the listener the times will vary. One of the major problems with this cue is the cone of confusion. The problem exists when the ILD produced by a sound at one location and the ILD produced by a sound at any location on a cone centered on the interaural axis with the actual sound source falling onto its surface are the same. (Stefan Röttger, 2006).

ILD represents the difference in sound pressure level reaching both ears. This difference exists due to each ear being able to detect its own loudness. The head can cast an acoustic shadow, which consists of an area to the side of the head opposite from the source of sound in which the loudness of a sound is less because of blocked sound waves. The acoustic shadow allows for the changes in loudness and frequency going into each ear and works best for high-frequency sound (Hobbs, 2004).

Head-Related Transfer Function

HRTF describes how a sound wave is processed by the head, pinna (visible part of the ear that is outside the head), and torso and their diffraction and reflection properties before it hits the inner ear (Jens Blauert, 1997). In terms of gaming, HRTF sounds allow players to hear sounds from every direction instead of a single source. One of the advantages is that when paired with a headset, the quality of sound increases (Lundqvist, 2010).

2.2.2 Surround Sound Enhancement

One technology recently developed to better the simulation of surround sound experience is the Apple Spatial Audio. This software, released as part of the most recent firmware update for the AirPods Pro, through the use of Dolby Atmos's signals and the application of directional audio filters, the frequencies for each earphone are adjusted so that sound can be simulated better in a 3D space (Parsons, 2020).

Dolby Atmos is a software that was firstly developed for use in cinemas and adds height channels that allow sounds to be interpreted as three-dimensional (Morrison, 2019).

The main distinguishing factor from other software is not only the fact that it tracks the head movement through the use of gyroscopes and accelerometers in the AirPods but also how it tracks the position of the screen the user is watching it on, where the screen can belong to an iPhone or an iPad, placing the sound relatively to the screen (Parsons, 2020).

2.3 Related Works

Through the years, accessibility in games is becoming an important matter and the gaming industry is trying to satisfy deaf gamers using different techniques so they can feel more or less the same immersion has hearing gamers have (Craven, s.d.). Some of these features include:

- **Subtitles** - This allows the hearing-impaired players to not miss out on any narration or important conversation present in the game. Many times, these dialogues are a fundamental part of the progression of the game and without it, the player can come to a standstill. In the case of subtitles, there are some aspects to keep in mind such as the placement and appearance, so the player can see it clearly, the label of who is speaking, and also the importance of why the subtitles should be on by default for the player to not miss out on any of the story. All these little details allow for a better experience for a deaf player and do not affect negatively the experience of the average player. Figure 2 shows a simple example of this feature.



Figure 2: Dialogue in The Elder Scrolls V: Skyrim.

- **Sound Visualization** – Allows the player to know where specific sound effects are coming from. It is a helpful feature in open-world games that have audible gunfire or that allows enemies to sneak up on you. One of the games that allow the player to choose this option can be seen in Figure 3, indicating two different types of sound, gunfire to the west and footsteps in the northeast direction.



Figure 3: Sound visualization in Fortnite Battle Royale.

- **Map Navigation** – The game indicates visually the direction in which the player should go. This ensures that every player knows where they must proceed to achieve the next goal. This guidance is usually done using a compass on the top or bottom of the screen (Figure 4) or using indicators on the extremities of the screen, in Figure 5 case, it is on the left side of the screen.



Figure 4: Gameplay from Assassin's Creed Valhalla.

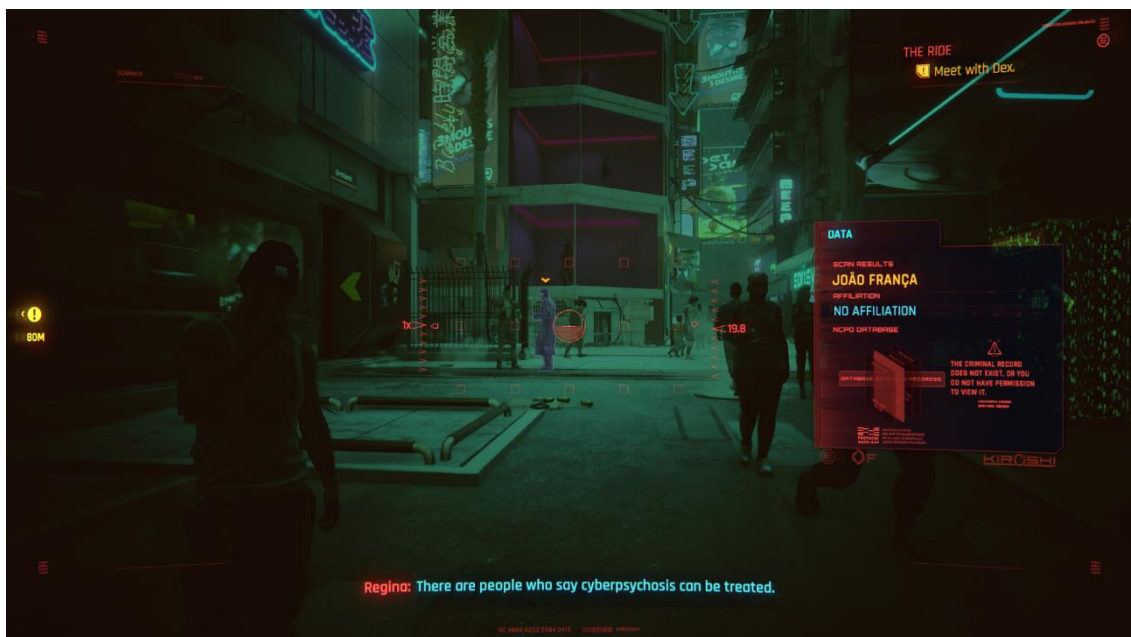


Figure 5: Gameplay from Cyberpunk 2077.

- **Stealth noise level indicator and enemy proximity** – In games that require stealth, audio hints often help the player understand if they were spotted or if they made too much noise, which helps in choosing between hiding or continuing the mission. Games use indicators like the one seen in Figure 6 to provide this information to the player. In Assassin's Creed IV: Black Flag they go from yellow which indicates that the enemy was able to hear the player's character and suspects something is near, to red which indicates that the enemy spotted the player and is now in attack mode.

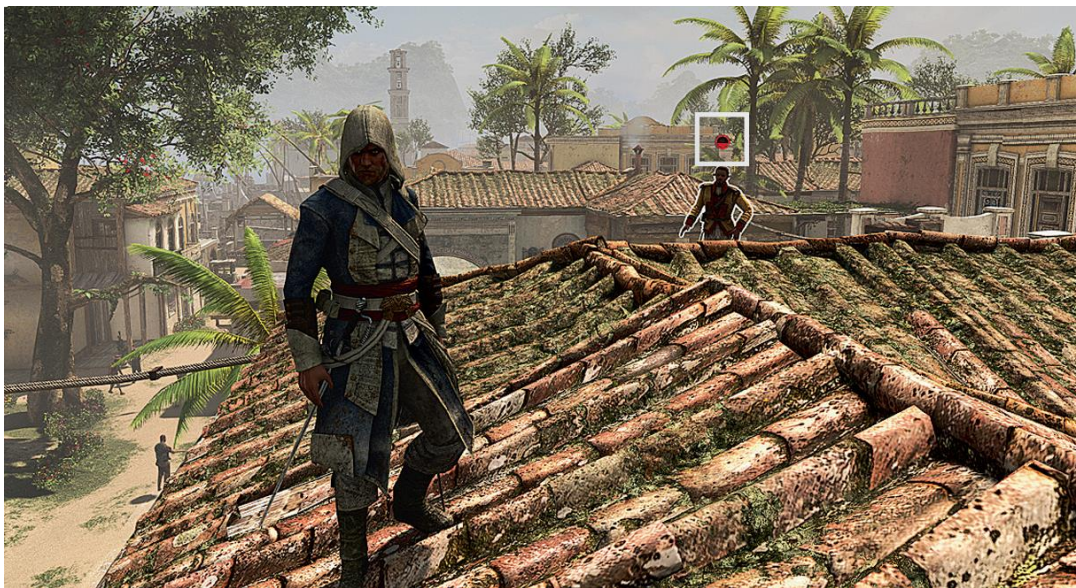


Figure 6: Gameplay from Assassin's Creed IV: Black Flag.

- **Controller vibration** – In console games, a simple vibration paired with a loud noise, such as an explosion, or a violent crash of your car can make the player more immersed in the game and can help detect better the location of the danger.

2.4 Existing Technologies

As mentioned before, to develop this software, a software that simulates an audio mixer and allows to analyze the different channel outputs will be crucial. In this section, a few of this type of software will be listed and compared.

2.4.1 Audio/Music Frameworks

The most important part of the project is how the audio will be obtained and manipulated. For this, different audio/music frameworks exist and allow access to several tools that facilitate the output processing of the sound.

CAMEL

Content-based Audio and Music Extraction Library (CAMEL) is a C++ framework that allows access to Music Information Retrieval (MIR) extraction functions without the restrictions of GUIs (Sanden, et al., 2010). MIR is the science of retrieving information from music.

It provides a series of tools that can be used in a wide range of analysis tasks, containing different extraction algorithms that can be used in prototyping and development stages.

CLAM

C++ Library for Audio and Music (CLAM) is a C++ framework that can be used in the audio and music realm as a research and development platform. It is composed of several algorithms ready to use and has a repository with all the classes that will act as data containers to use as input or output in the processing algorithms. It allows for rapid prototyping which can be used with Frontend technologies like Qt (Amatriain, et al., 2006).

Libstaudio

“libstaudio” serves as a standard audio API for C++. It contains five different methods that can print the input and output devices, generate a short melody that plays through the default output device, or, the most important for this project, measure the input volume coming from the microphone and constantly print the maximum captured during a time period (Somberg, et al., 2019).

Libsoundio

“libsoundio” is an API for C that allows features such as produce a sine wave into the default device or list all the devices, but the major issue is that it is not possible to obtain the decibel value of the output coming from any device which discouraged its use (Kelley, 2020).

Web Audio API

Another API that was considered was the “Web Audio API”, which is developed in Javascript, and would allow to show, on an HTML page, the values of the output and then it would be possible to draw a radar and show the information (MDN Web Docs, 2021). The main issue using this library would be translating the values into a readable number, which would require additional calculations.

CLAM vs CAMEL vs libstaudio vs libsoundio vs Web Audio API

CLAM and CAMEL offer many features regarding the treatment of sound, but CAMEL is best used in audio files and not with real-time input, and CLAM is best used with music files to identify chords, although it can be used on the microphone input.

Since “libstaudio” offers tools similar to “libsoundio” and Web Audio API, with the addition of measuring and obtaining the decibel level coming from the real-time input, the prior has the advantage. With this, the library chosen to aid in the retrieval of the information from the surround system is “libstaudio”.

2.4.2 Audio Mixing Software

One of the key elements for the development of this project is audio mixing software that will allow the manipulation and access to sound output from the sound card. In this subsection, some technologies will be described as well as the advantages and the disadvantages of their use.

Virtual Surround Card and Virtual Audio Cable

A virtual Surround Card allows to emulate a sound card and work like a real one. It can be used to record, change and broadcast sound from different applications. Two examples of this type of software are e2eSoft VSC (E2ESOFT, 2021) and Dante Virtual Soundcard (Audinate, 2021). Virtual Audio Cable is a driver that, as the name suggests, creates a virtual input and output that automatically sends all audio output to the input (VB Audio, 2020). By combining the two

applications, it is possible to manipulate in real-time the output coming from the game and to reroute the adjusted output, only being needed to obtain the output from the channels.

VB Audio – Voicemeeter

Voicemeeter is a free software for Windows that has the option to record audio while having the option to change the intensity of the output. It works up to a 7.1 system and provides an API that can be used in any language and can make use of all its features (VB Audio, 2020). It integrates three audio input devices (two physical and one virtual), and three audio output devices (two physical and one virtual). This allows to combine different application sounds or just single out one of them.

VB Audio – Voicemeeter Banana

Voicemeeter Banana is an upgraded version of Voicemeeter. It has the same characteristics and features but with the addition of new ones. The main improvement is that Banana integrates five audio input devices (three physical and two virtual), and five audio output devices (three physical and two virtual) (VB Audio, 2020).

Virtual Surround Card /Virtual Audio Cable VS Voicemeeter VS Voicemeeter Banana

The next table focuses on the comparison of technologies relative to audio mixing.

Table 2: Comparison Virtual Surround Card vs Voicemeeter vs Voicemeeter Banana

	Virtual Surround Card / Virtual Audio Cable	Voicemeeter	Voicemeeter Banana
Record Audio	Yes	Yes	Yes
Reroute devices	Yes	Yes	Yes
Additional mixing tools	No	Yes	Yes
Free	-	Free	Free
API	No	Yes	Yes
Virtual devices	-	One	Two

Since the Voicemeeter Banana software offers an API that allows getting information from each sound channel and easier use through code, it is the software that will be used for this project.

In addition, Voicemeeter Banana also provides additional software that allows the user to manipulate the audio in whatever way they would like such as the 8x8 Gain Matrix that allows to re-distribute audio channels into different multi-channel systems.

2.4.3 Frontend Technologies

Windows Forms

Windows forms is an open-source user interface framework for Windows developed by Microsoft. It uses C Sharp as the programming language and allows the creation of desktop apps with a drag-and-drop visual designer (George, 2020).

Qt

Qt is an open-source development platform designed for the creation of user interfaces and was developed by The Qt company. It uses C++ or Javascript as the programming language, together with several UI designing tools, it allows the creation of desktop and mobile apps (The Qt Company, 2020).

Angular

Angular is owned by Google and was acquired to enter the Web Apps development market. It was created due to the necessity to facilitate the creation of Web Applications and guarantee that they maintain responsive behavior. It was born based on JavaScript however in one of the updates it was changed to TypeScript (Google, 2020).

React

React is one of the most popular Web Apps technologies. It was created and maintained by Facebook and is based on JavaScript. With the success of React, a specific version was also created for the development of hybrid applications called React Native, whose main function is to optimize the user's usability concerning the device on which he runs the application (Facebook, 2020).

Angular VS React VS Windows Forms VS Qt

The following figure shows the number of downloads per year of each platform of both React and Angular. Since 2015, React has more downloads than angular and the number continues to grow.

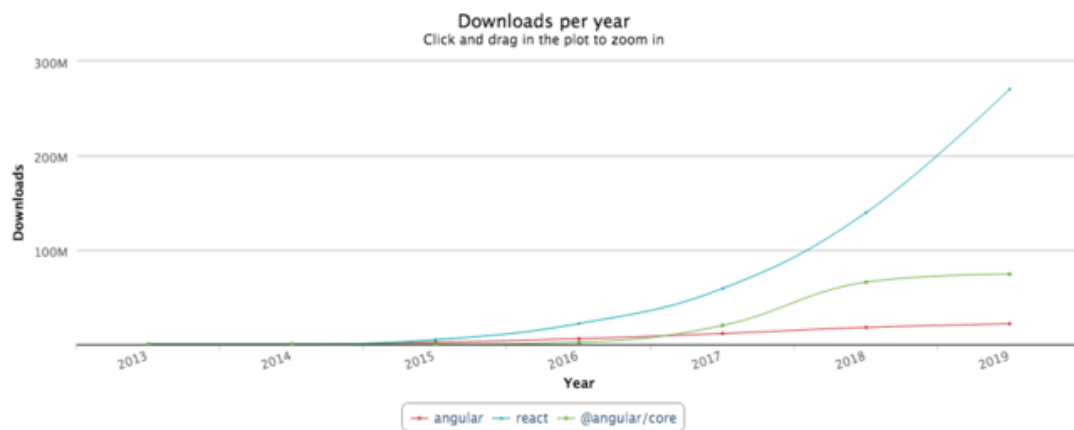


Figure 7: Angular vs React downloads per Year (Reis, 2020).

For a web browser solution, the framework chosen for this project would be React which allows for a better mobile cross-platform framework and because it has an easier learning curve, but to obtain the sound output from a sound card a low-level language will be needed and thus Windows Forms or Qt is a better choice.

2.4.4 Programming Language

JavaScript

Frequently known as JS, is a programming language characterized as dynamic, prototype-based, and multi-paradigm (Mozilla, 2020). It is the main client-side language in Web navigators that allows the creation of dynamic Web Pages.

C++

C++ is a low-level general-purpose programming language developed by Bjarne Stroustrup as a continuation of C, even having the nickname “C with Classes”. It is one of the most used

languages worldwide counting efficiency and performance as its best design features (Stroustrup, 2014).

C Sharp (C#)

C Sharp is an object-oriented programming language developed by Microsoft as part of the .NET platform. Being inspired by the C languages family, it has some similarities to C, C++, and Java, but has the goal to produce a more economical application concerning memory space.

JavaScript vs C++ vs C Sharp (C#)

The language will depend on the Frontend technology chosen. If React is chosen the language used for this project will be JavaScript due to the familiarity, if Windows Forms or Qt is chosen, the language used will be either C Sharp or C++, although Qt allows the use of Javascript a low-level language is recommended for this sort of system.

2.5 Existing Solutions

The objective of this section is to describe other solutions of spatial surround sound identification systems that translate in-game audio into visual information, which this project was based on as the idea for this thesis.

2.5.1 Sonic Radar 3

Sonic Radar 3 is a software that creates a screen overlay that presents in a visual form the sound cues in-game according to their position. It can be used by any player as long as the game supports 5.1 channels or more and the machine used is compatible with the software (ASUS, 2019). The goal is to reduce the disadvantage players with hearing difficulties face since most games rely on sound effects to give feedback to the player and help with the main goal (Aaron Marks, 2010). In Figure 8 the radar is placed in the top right corner and shows the direction of sound coming from the front of the player.

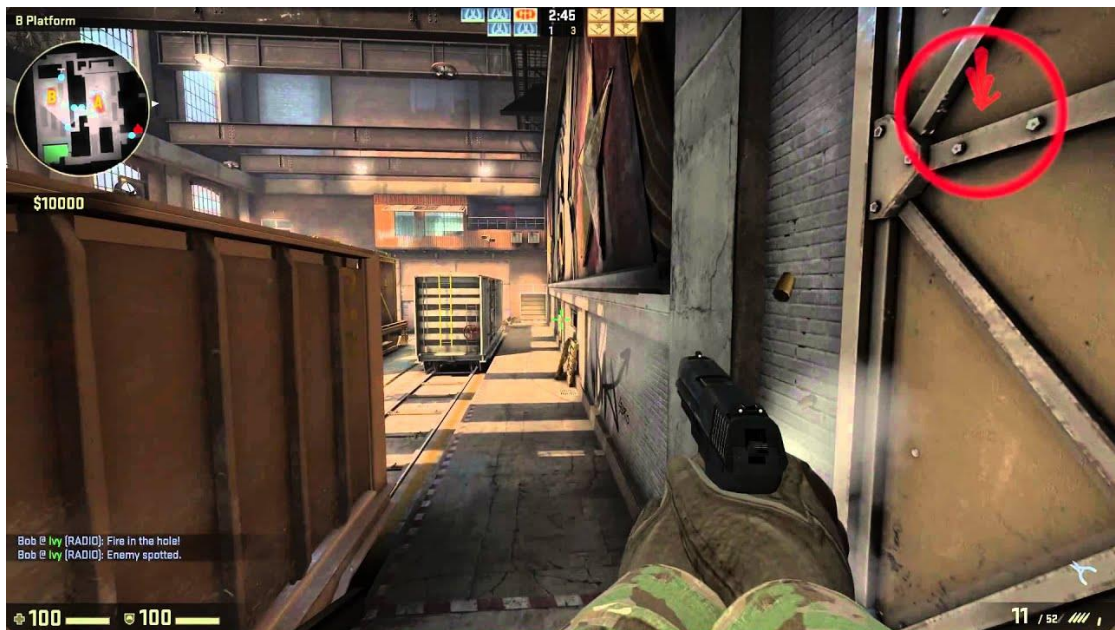


Figure 8: Sonic Radar 3 used in Counter-Strike: Global Offensive.

2.5.2 THX Spatial Audio (Razer 7.1 Surround Sound Software)

The THX Spatial Audio is a paid software made with Razer that is available for certain headsets from the brand and can be used in Windows. This software needs to be used with a 7.1 Surround sound headset for a better experience. One of the pages from the interface allows the user to see a radar that indicates the origin of the sound, and it also allows the user to choose from different profile settings, having one specific for gaming which enhances footsteps and gunshots in-game (Razer, 2021). This radar is what increases the value of this software for this study. shows the window where the radar is present.

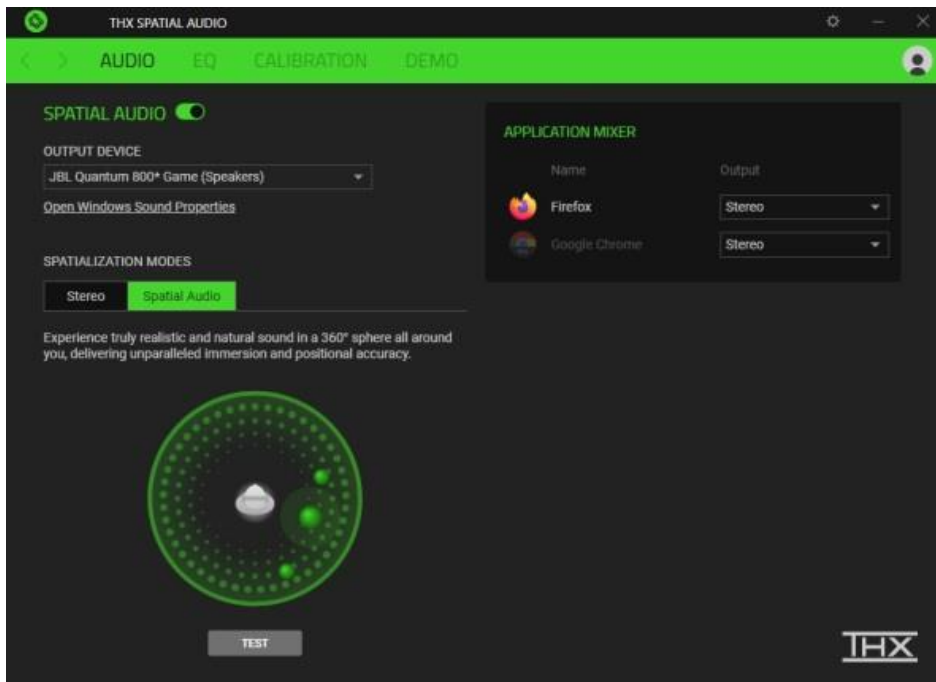


Figure 9: Window showing radar from THX Spatial Audio software.

2.5.3 Graphic representation

With the previous examples, three different options to indicate the direction of sound were identified; four-directional arrows, a text message with the direction, or a radar that uses different intensities. Figure 10 shows the first drafts drawn. In chapter 3 the best solution will be chosen according to certain criteria.

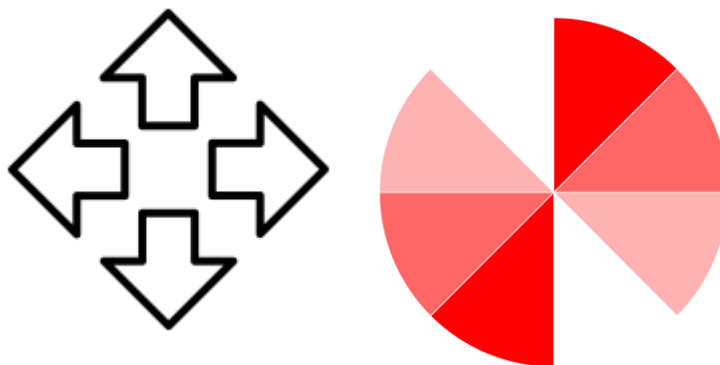


Figure 10: Examples of the possible graphic representations.

Furthermore, the possibility of using a vibrating helmet will be discussed, since this is a solution that would allow the player to focus on the game by using also their tactile system and not

divert their attention towards more visual indicators. The hardware would be composed of eight Bluetooth sensors that would be connected to the system that in turn would send a response to the helmet so the correct sensor would vibrate and thus indicating the direction to the user.

3 Value Analysis

This chapter has the goal to evaluate the business side of this project. A value proposition will be presented to highlight the advantages that come from the development of a spatial identification system of surround sound in games.

Tools will be used to help decide the course of action for the development such as the New Concept Development Model (NCD) that will help define the different opportunities and approaches that could be pursued and the Analytic Hierarchy Process (AHP) that will identify, with chosen and validated criteria, the best approach to take regarding the graphic presentation, thus complementing each other.

Finally, the Business Model CANVAS and Porter's Value Chain will be applied to the project to give a better perspective on the business idea and value, respectively.

3.1 New Concept Development Model

The innovation process of any product can be divided into three phases: Front End of Innovation (FEI), New Product and Process Development (NPPD), and commercialization phases.

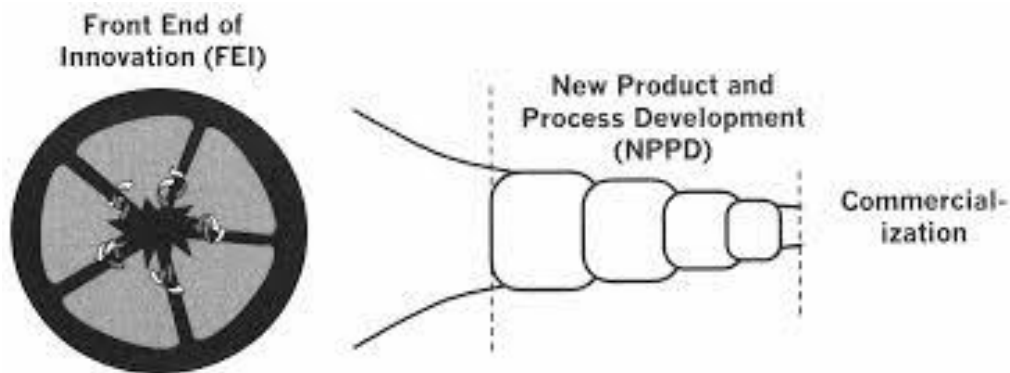


Figure 11: Product Innovation Process (Koen, et al., 2001)

The Front End of Innovation is composed of the activities that come before the more formal and structured New Product and Process Development phase. In the second phase, some marketing evaluations are made to obtain information about the market. In the commercialization phase, a business model is developed with the decisions made in the previous process. The shapes of the FEI and NPPD are meant to be circular and sequential, respectively. This shows the contrast between them where the FEI process is expected to be more fluid and iterate between the five elements and where the NPPD is meant to show a series of well-structured and chronologically-ordered steps (Koen, et al., 2001).

The New Concept Development Model has the goal to optimize the FEI process while providing insight and a common language/definition of the key elements of the Front End of Innovation.

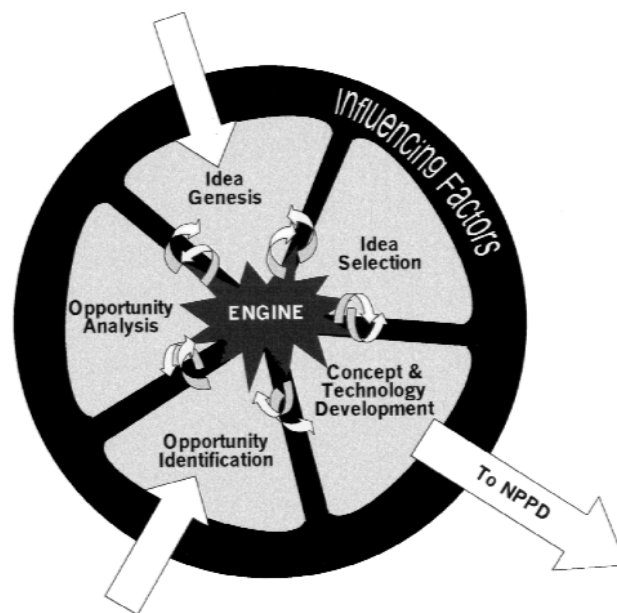


Figure 12: The New Concept Development Model (Koen, et al., 2001).

The NCD model (Figure 12) is composed of three key elements:

- The Engine characterizes the leadership, culture, and strategy of an organization.
- The internal area represents the five elements of the NCD.
- The Influencing Factors represent the outside factors that affect the decisions of the two inner parts. These same factors affect the whole innovation process seen in Figure 11.

This model, according to Koen, is composed of five key elements:

- **Opportunity Identification** – The lack of different options for hearing-impaired people in games is the main problem. To combat this situation, the implementation of software that allows the visualization of the origin of sound while gaming will add another alternative.
- **Opportunity Analysis** – After a careful analysis of the different opportune possibilities, the development of new software that can be used without restrictions was concluded to be the better option.
- **Concept & Technology Development** – In this component there were two options, either use existing software that has some restrictions for its use or create a new one that can be used in a more general way.
- **Idea Selection** – In this section, the final idea was chosen with the help of the Analytic Hierarchy Process (AHP) method. Having market saturation, interaction, and granularity as the three chosen criteria, the idea chosen was the one to develop a system that allows the user to visualize the origin of sound while gaming.
- **Idea Genesis** – In this section, the form of visualization of the information was the critical point. The information could be seen through a radar that highlights the direction of the sound (including sound intensities), through a message that appears on-screen, or a four-directional arrow.
-

3.2 Analytic Hierarchy Process (AHP)

The Analytic Hierarchy Process (AHP) is a measurement method used to quantify the relative importance of a set of activities, through qualitative and quantitative criteria. It allows to capture the main aspects of a decision and has a useful technique that allows maintaining consistency, reducing the partiality from the decision-making process (Saaty & Wind, 1980).

The AHP is made up of three main functions :

- **Structuring Complexity** – Deals with complexity with a hierarchical structure turning it into a homogeneous group of criteria.
- **Measurement on a Ratio Scale** – Each criterion will be assigned a weight in a ratio scale thus giving it each their importance.
- **Synthesis** – The final phase, where the information obtained is put together which facilitates the process of dealing with complex decisions (Forman & Gass, 2001).

Each criterion is determined according to the fundamental scale proposed by Saaty, ranged from 1 to 9, where 1 represents a comparison where the elements have equal importance, 5 represents strong importance and 9 represents absolute importance over another element (Saaty & Wind, 1980).

3.2.1 Stage 1 - Construction of the Hierarchy for the Criteria

In this stage, a hierarchical decision tree is made with the intent of highlighting the best idea of the three and studying the relationship between the criteria. With this objective, in this tree, the criteria need to be well structured as well as the goal and the alternative ideas.

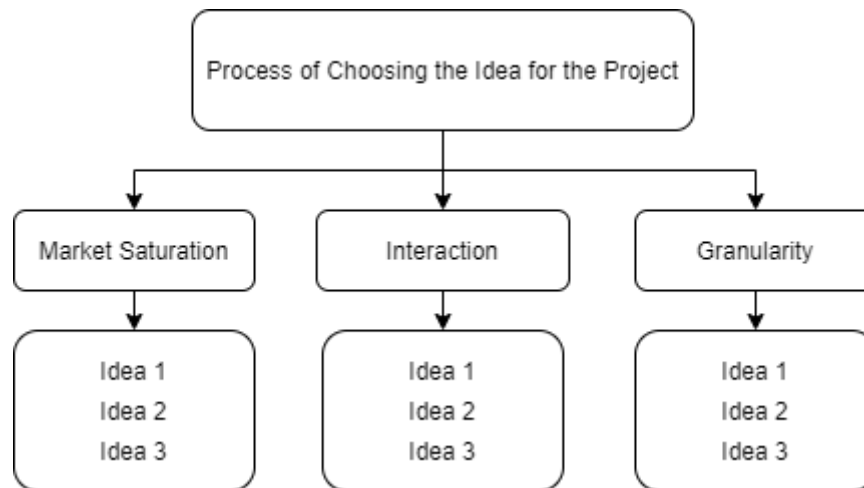


Figure 13: Hierarchical Decision Tree - AHP

The ideas shown in Figure 13 are the ones presented in the Idea Genesis element of the NCD model. Idea 1 consists of a system that shows information through a radar that highlights the direction of the sound, Idea 2 shows the information through a message shown on screen, and Idea 3 shows information through a four-directional arrow.

The three criteria chosen to decide which one of the three ideas thought initially are Market Saturation, Interaction, and Granularity.

- **Market Saturation** – Before developing an idea, it is better to analyze the market to assess the number of similar software already developed. An idea with fewer alternatives is more interesting to pursue.
- **Interaction** – Represents the human-interface interaction and the ease or difficulty the user has to analyze the information presented to them.
- **Granularity** – Indicates the amount of information the user can have access to while using the software.

3.2.2 Stage 2 – Alternatives and criteria comparison

This stage uses a “matrix of pairwise comparisons of the criteria” to define the priorities between the elements of each level of the hierarchy (Saaty, 1987).

To develop the comparison matrix, the Fundamental Scale proposed by Saaty will be used. This scale uses nine values which allow maintaining the consistency of the matrix.

Table 3: The Fundamental Scale - AHP (Saaty & Wind, 1980)

Intensity of Importance	Definition	Explanation
1	Equal importance	Two activities contribute equally to the objective
3	Weak importance of one over another	Experience and judgment strongly favor one activity over another
5	Essential or strong importance	Experience and judgment strongly favor one activity over another
7	Demonstrated importance	An activity is strongly favored and its dominance is demonstrated in practice
9	Absolute importance	The evidence favoring one activity over another is of the highest possible order of affirmation
2,4,6,8	Intermediate values between the two adjacent judgments	When compromise is needed

Using the criteria defined in the Hierarchical Decision Tree, the following comparison matrix was made.

Table 4: Matrix of pairwise comparisons of the criteria - AHP

Criteria	Market Saturation	Interaction	Granularity
Market Saturation	1	1/3	1/5
Interaction	3	1	1/3
Granularity	5	3	1

3.2.3 Stage 3 – Relative Priority

The goal in this stage is to give each criterion a value with the same measurement. For this, the matrix will be normalized by dividing each cell by the sum of the respective column.

Table 5: Sum of columns from the matrix from Stage 2 - AHP

Criteria	Market Saturation	Interaction	Granularity
Sum	9	13/3	23/15

Table 6: Normalized Matrix Table from Stage 2 - AHP

Criteria	Market Saturation	Interaction	Granularity
Market Saturation	1/9	1/13	3/23
Interaction	1/3	3/13	5/23
Granularity	5/9	9/13	15/23

After obtaining the normalized matrix, the relative priority of each criterion is calculated by calculating the mean value of each line from the normalized matrix.

Table 7: Relative Priority - AHP

Criteria	Market Saturation
Market Saturation	0.11
Interaction	0.26
Granularity	0.63

The conclusion taken from this stage is that Granularity is the most important criterion, followed by Interaction and then Market Saturation.

3.2.4 Stage 4 – Consistency

In this stage, the Consistency Ratio (CR) is calculated by dividing the Consistency Index (CI) by the Random Consistency Index (RI).

$$CR = \frac{CI}{RI}$$

The Consistency index is calculated using the next formula:

$$CI = \frac{\lambda_{max} - n}{n - 1}$$

Where **n** is the number of criteria and λ_{max} is the eigenvalue of the matrix. The following calculation obtains the eigenvector.

$$\begin{vmatrix} 1 & 1/3 & 1/5 \\ 3 & 1 & 1/3 \\ 5 & 3 & 1 \end{vmatrix} x \begin{vmatrix} 0.11 \\ 0.26 \\ 0.63 \end{vmatrix} = \begin{vmatrix} 0.32 \\ 0.79 \\ 1.95 \end{vmatrix}$$

With the eigenvector calculated the λ_{max} can be calculated by dividing each value from the eigenvector and divide it by the relative priorities and then calculating the average.

$$\lambda_{max} = \frac{0.32}{0.11} + \frac{0.79}{0.26} + \frac{1.95}{0.63} = 3.04$$

With the value of λ_{max} is now possible to calculate the Consistency Index value.

$$CI = \frac{3.04 - 3}{3 - 1} \approx 0.019$$

The value for the Random Consistency Index is obtained from the following table.

Table 8: Random Consistency Index Table - AHP (Saaty, 1987)

n	1	2	3	4	5	6	7	8	9	10
Random Consistency Index (RI)	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

The number of criteria chosen is three, therefore $n = 3$. It is now possible to calculate the Consistency Ratio (CR) using the initial formula presented.

$$CR = \frac{0.019}{0.58} \approx 0.033$$

Saaty proposed that a CR value above 0.1 should not be tolerated since the percentage of inconsistency would be too high. Since the value obtained is below 0.1 it is possible to assume that the relative priorities are acceptable.

3.2.5 Stage 5 – Evaluation regarding the criterion chosen

Until now the process validated the three criteria we are going to use to evaluate the three ideas. The next step will evaluate each idea individually using numerical values from a scoring scale by selecting the qualitative dimension that best fits (Lalic & Palcic, 2009).

Table 9: Conversion of qualitative dimension to numerical value - AHP

Qualitative Dimension	Poor	Fair	Good	Very Good	Excellent
Numerical Value	0.0	0.15	0.35	0.7	1.0

The qualitative dimension ranges from “Poor” to “Excellent”, the first value means the project doesn’t fulfill the criterion at any level, and the last value meaning it fulfills completely the

criterion. Using Lalic and Palcic’s following table, the calculation of the final grade for each idea can be made but first, the ideas need to be evaluated using the qualitative dimension.

Table 10: Qualitative score for each idea – AHP (Lalic & Palcic, 2009).

	Total	Criterion	Market Saturation	Interaction	Granularity
			Weight	0.11	0.26
Idea 1		Score	Very Good	Very Good	Excellent
Idea 2			Excellent	Good	Very Good
Idea 3			Good	Very Good	Good

Table 11: Final score for each idea – AHP

	Total	Criterion	Market Saturation	Interaction	Granularity
			Weight	0.11	0.26
Idea 1	0.889	Score	0.7	0.7	1.0
Idea 2	0.642		1.0	0.35	0.7
Idea 3	0.441		0.35	0.7	0.35

The final score for Idea 1 is calculated using the following process:

$$Idea\ 1 = 0.11 * 0.7 + 0.26 * 0.7 + 0.63 * 1.0 = 0.889$$

The calculation for the other ideas follows the same logic but using their numerical value.

Although the scores of the criteria don’t vary too much, we can see that the idea with the highest grade is Idea 1. This results from the fact that Idea 1 distances itself from the others in the most important criteria.

3.3 Value Proposition

A Value Proposition encapsulates the manner in which the product or services are made available and offered to meet the necessities of the customer (Kambil, et al., 1996).

The software that is going to be developed, has the goal to facilitate the gameplay and the overall experience for deaf or hearing-impaired users in a way that their experience becomes

come enjoyable. The value of this project focuses on giving the user clear information about the location of sound so they can also be capable of recognizing different kinds of sound and distances when playing video games and remove a possible handicap.

3.4 Business Model CANVAS

The Business Model CANVAS is a tool that facilitates the analysis and allows the development of new or existing business models on a single page (Osterwalder & Pigneur, 2010). It was developed by more than 200 consultants around the world.

It can be divided into two stages: the Back Stage that contains the more structural elements and the Front Stage that deals with more subjective and emotional elements. The Back Stage is composed of four segments and the Front Stage has five segments. Although they are divided, they still communicate with each other.

- **Customer Segments** – For whom are we creating value and who are the most important customers.
- **Value Propositions** – What value is delivered to the customer and what products and services are offered to each Customer Segment.
- **Customer Relationships** – What relationships each Customer Segment expects from the institution.
- **Channels** – How the Customer Segments are reached, and which are best.
- **Revenue Streams** – How the institution generates money from each Customer Segment.
- **Key Activities** – The activities that the institution must do to have success.
- **Key Resources** – The financial, physical, intellectual, or human resources that the institution requires.
- **Key Partners** – The network of partners or suppliers essential for the Key Resources and Key Activities.
- **Cost Structure** – The most important costs originated from this business model.

The CANVAS model for this project can be seen in Figure 14.

Key Partners	Key Activities	Value Propositions	Customer Relationships	Customer Segments
ISEP faculties; Voluntaries.	Software design; Software development; Software testing.	Implementation of a software that shows information about the location of sound in video games;	Personal assistance via e-mail.	Deaf Videogame players; Hearing-Impaired Videogame players.
	Key Resources			
	Audio Mixing Software.			
			Channels	
			Surveys.	
Cost Structure			Revenue Streams	
Development Costs; Promotion Costs.			Free software.	

Figure 14: Business Model CANVAS

3.5 Porter's Value Chain

Porter's Value Chain is a tool used on a product to increase its value. It's a process composed of a series of activities that can be divided between Primary and Support activities (Porter, 1985). The Primary activities are composed of the following:

- **Inbound Logistics** – All the processes related to receiving and storing material or parts internally.
- **Operation** – Changes needed to alter the input received into a product that can be sold.
- **Outbound Logistics** – All the processes related to the movement of the final product to the customer.
- **Marketing and Sales** – Processes used to convince the customer that the final product is better than that of competitors.
- **Service** – Processes needed to maintain the value of the product sold.

The Support activities are composed of the following:

- **Infrastructure** – Consists of the core systems that allow maintaining daily operations. This includes legal, accounting, administrative, and general management.
- **Technological development** – Relates to the general equipment and software needed to continue to transform the raw parts into a finished product.
- **Human resources management** – Consists of the processes that involve human management such as hiring, training, or firing personnel.
- **Procurement** – The processes needed to acquire the resources needed to do the operations.

For the project that is going to be developed the main operation consists of turning the sound output (Inbound Logistics) into visual information for the user to process (Outbound Logistics). The Marketing and Service activities will be related since the promotion and assistance for this software will both be made via e-mail.

The Support Activities for this project are almost nonexistent except for the Technological Development activity where audio/music frameworks, audio mixing technologies, frontend technologies are essential parts for the operation mentioned before, and the Procurement activity where the developer had to acquire a surround sound card to capture surround sound.

4 Approach and Design Discussion

This chapter will address the proposed and alternative designs for the system that is going to be developed. The system to be implemented can be divided into two elements that will communicate with each other, one that will obtain the audio coming from the game and process it, and the other that will take that information and show it to the user in a clear way.

4.1.1 Spatial Identification System of Surround Sound in Games

This system will interact directly with the sound card to obtain the audio output. Firstly, the software shall identify the sound card or cards that the computer has connected. For this project, ideally, the sound card should be a surround sound 5.1 or higher. The main restriction of this design is the need for a surround sound card or in alternative the use of a virtual sound card that would simulate the behavior of a real surround sound card. This could be achieved using the Virtual Audio Cable mentioned in section 2.4.2.

The use of anything less than a 5.1 sound system will not offer enough channels and thus not enough information. With a surround sound system, it is possible to show, at least, five directions or even ten if one is to consider the intervals between the five channels.

After obtaining information from the sound card and confirming it is a surround sound card, with the use of an audio/music framework and mixing software, the output from each individual channel can be obtained and processed to remove any extra “noise”. With the help of the mixing software, it is possible to only pick up sounds that have a certain level of frequency or

intensity meaning it is possible to ignore certain sounds that don't give crucial information to the player.

An alternative to the restriction of needing a surround sound card is using virtual surround sound. Since many games use HRTF calculations to simulate surround sound in-game, the reversed calculations could be applied to obtain the origin of the sound. This creates two problems; the first problem is that the application of these calculations would take too long to implement. The second problem, contrary to surround sound, the possibility of the calculations not being 100% accurate would add another variable to the hypothesis that this dissertation proposes to evaluate.

After the system converts the processed channel's output into degrees, it sends this information to the interface element to show to the user.

4.1.2 Interface/Information System

This system will be responsible to show the information to the player clearly and intuitively. As mentioned before in the Value Analysis chapter, three ways to show the information were evaluated, either using radar, a message on-screen, or four directional arrows. It was concluded, using validated criteria, that the best choice was a radar due to the information it can convey to the player.

Another issue to address in this part of the system is where the information is presented. The way chosen to do this is by showing the information on the browser so the player can either see it on a mobile device or a second screen. Another alternative way considered to show the information is by using an overlay on the computer screen itself while gaming.

One other alternative that could be used to give information to the user is using a helmet that would have between four and eight vibrating sensors that would indicate to the player the direction of the sound. This option would take more time and thus is not guaranteed to be developed.

5 Design and Implementation

This chapter will address the proposed and alternative designs of the solution developed, taking into account standards and rules, adopting the good practices of computer engineering. Firstly, the architecture will be presented followed by the functional and non functional requirements identified. After, the process of implementation will be described dividing it into two systems, the spatial identification system of surround sound, and the interface/information system.

5.1 Architecture

The first step to take is to conceptualize, with a high level, the way the different components will communicate with each other.

The architecture implemented follows the representation present in Figure 15 and is composed of the following components:

- The Backend Component is responsible for the generation of the key information that will be sent to the interface component.
- The Interface Component is responsible for converting the information into a readable radar that aids the player while gaming.

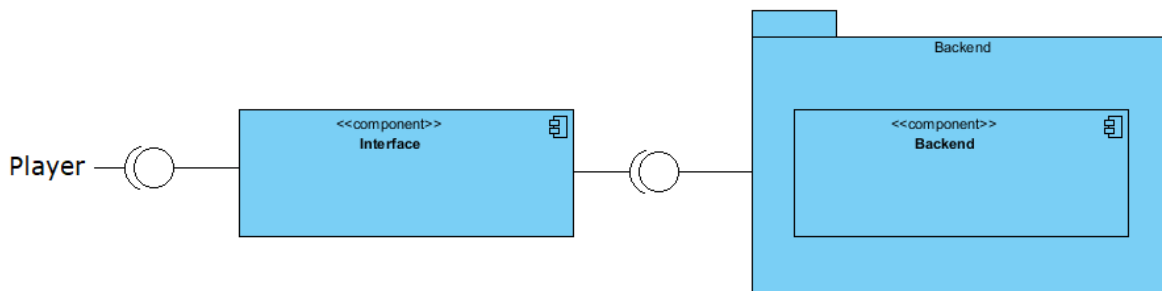


Figure 15: Component Diagram.

5.2 Functional and Non Functional Requirements

Functional requirements are those that explicitly describe the functionalities and services that the system must have. These specify how the system should react to a specific input, how it should behave in certain situations, and what the system should not do.

A use case describes a scenario of possible interaction with one user or another system.

For this project, the following use cases were identified:

- The player is able to initiate the radar.
- The player is able to identify the origin of the sound.

The use cases are separate since the first relates to the user opening the radar while the identification of the origin of sound relates to the understanding of the information shown to the user. The user to use the software needs to open the different applications that will be enumerated in section 5.3.1, which consists of the mixing software Voicemeeter banana and the tool 8x8 Matrix provided by Voicemeeter itself. After this, the user only needs to run the executable and the radar will be shown.

The following figure shows a graphic representation of the use cases listed before.

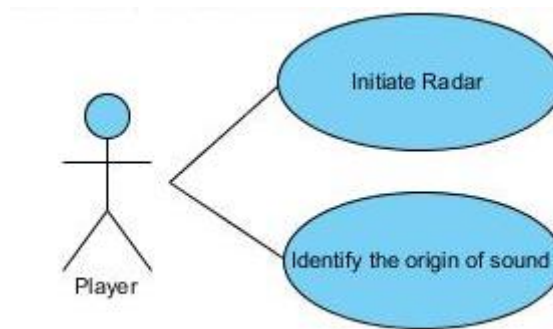


Figure 16: Use Case Diagram.

Non Functional requirements are the ones that cannot be described as a use case. Robert Grady presented the **FURPS** acronym to validate these requirements and consequently give more detailed information as well as a way to categorize them (Grady, 1992).

Relating to Usability the interface should be easy to understand and it should not have excessive information.

The software should also keep the number of errors to a minimum as well as maintain the CPU usage low so it doesn't interfere with video games. Another important aspect is that the software can run on any PC as long as the user has a surround sound card connected to it.

5.3 Implementation

In this section, the steps and decisions made during the development process will be presented. The process had in count the content presented in the state of the art and the approach and design discussion chapters.

5.3.1 Spatial Identification System of Surround Sound in Games

The first step in the development was obtaining the output of the surround sound card. A decision was made about writing the program in C++ due to its low-level capacities. With this set, a library named "libstdaudio" serves as a standard audio API for C++.

The decision to use “libstdaudio” was made after testing with different libraries. The first approach was made using the CLAM library, but it was soon concluded that the tools made available would not be helpful since the main goal of the framework is to work with chords from music or audio files, either by recording and showing the chord or by allowing to edit in post-production. Then the CAMEL library was tried but since the features were mostly related to music and could not be used with real-time input, this option was discarded.

The “libsoundio” was also a library tested with. It is a C library that was simple to use and allowed to obtain the name of the default devices and supported channels mapping which consists of rerouting different channels into one single audio track. This gave the ability to identify which channels were active but without knowing exactly what was the loudest.

The use of the Web Audio API allowed joining the information and the interface system the software is divided into a single system made with Javascript. The issue found was that the API works better with audio files instead of real-time input, and concern arose around the possibility of incurring excessive consumption of system resources such as memory and failing to present the visual data in a reasonable timeframe.

Since “libstdaudio” already offered a method that showed the decibel level of the different channels of the default input device, less work would be needed to implement the system.

The last method present in the *level_meter* file is shown in Figure 17. Firstly the default input device is obtained using the method *get_default_audio_input_device*. Then the device is initiated with the correct buffer, in this case, the *input_buffer*. This buffer represents an agglomerate of information from a period of time, the most important being the frames, which are a moment in time, and the channels.

After this, a loop statement is initiated circling the frames, and inside it, there’s another loop that iterates the channels from the device. There the value is obtained and stored in the *max_abs_value_input* variable if it is the higher recorded.

Finally, in a 150 milisecond interval the program uses the *gain_to_db* method to convert the value to a decibel number and then writes it to the command prompt. It also resets *max_abs_value_input* variable before doing the whole process again.

```
#include <iomanip>
#include <atomic>
#include <cmath>
```

```

#include <thread>
#include "audio"

float gain_to_db(float gain) noexcept {
    return gain > 0 ? std::log10(gain) * 20.0f : -
std::numeric_limits<float>::infinity();
}

int main() {
    using namespace std::experimental;
    std::atomic<float> max_abs_value_input = 0;

    auto device_input = get_default_audio_input_device();
    if (!device_input)
        return 1;

    device_input->connect([&](audio_device&, audio_device_io<float>& io) noexcept {
        if (!io.input_buffer.has_value())
            return;

        auto& in = *io.input_buffer;

        for (int frame = 0; frame < in.size_frames(); ++frame) {

            for (int channel = 0; channel < in.size_channels(); ++channel) {
                float abs_value = std::abs(in(frame, channel));

                if (abs_value > max_abs_value_input)
                    max_abs_value_input.store(abs_value);
            }
        }

        device_input->start();
        while(device_input->is_running()) {
            std::this_thread::sleep_for(std::chrono::milliseconds(250));
            std::cout << "input " << gain_to_db(max_abs_value_input.exchange(0)) << " dB\n";
        }
    }
}

```

Figure 17: level_meter file unaltered.

The thought process was to take the output and turn it into input to take advantage of the method last mentioned. The use of audio mixing software allows for this manipulation to be made. The software chosen was Voicemeeter Banana.

The default input and output devices need to be changed to Voicemeeter Input and Voicemeeter Output respectively (Figure 18). The Voicemeeter Input device also needs to be configured as Surround 5.1.

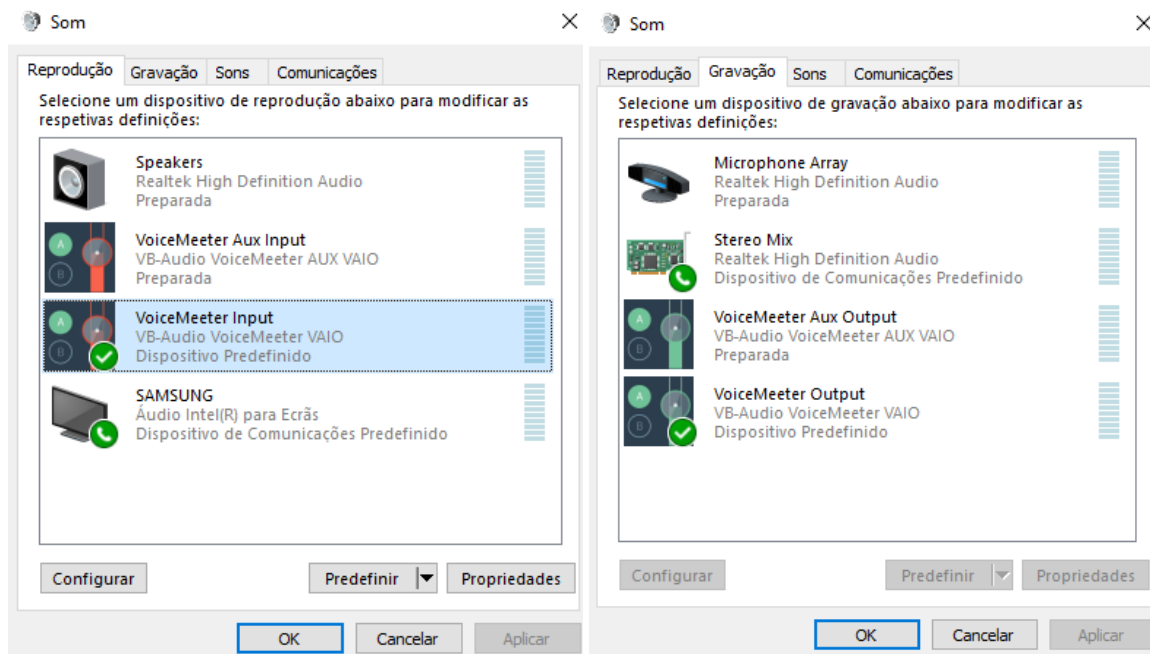


Figure 18: Default Device Setup.

This software is divided into three sections, the Hardware Input, the Virtual Input (Virtual 1 and Virtual 2), and the Hardware Output (A1, A2, and A3). The Hardware Inputs can be ignored since the goal is to route the output as input. In the Virtual Input section, the B1 and B2 buttons in both Virtual 1 and 2 must be active (green). For A1 (Hardware Output) the selection needs to be the speakers from the surround sound card (Figure 19).



Figure 19: Voicemeeter Banana Setup.

With this, if the executable *level_meter* (shown in Figure 17) will output the decibels from the output but it does not distinguish between the left and right channels, only outputting the highest one in that interval. So firstly, the division between channels was made in order to print both channel's highest decibels levels recorded in the interval (Figure 20).

```

device_input->connect([&](audio_device&, audio_device_io<float>& io) noexcept {
    if (!io.input_buffer.has_value())
        return;

    auto& in = *io.input_buffer;

    for (int frame = 0; frame < in.size_frames(); ++frame) {

        max_abs_value_input1.store(std::abs(in(frame, 0)));
        max_abs_value_input2.store(std::abs(in(frame, 1)));

        float abs_value1 = std::abs(in(frame, 0));
        float abs_value2 = std::abs(in(frame, 1));

        if (abs_value1 > max_abs_value_input1)
            max_abs_value_input1.store(abs_value1);

        if (abs_value2 > max_abs_value_input2)
    }
}

```

```

        max_abs_value_input2.store(abs_value2);
    }
});

```

Figure 20: level_meter prints both input channels output.

The next step was to apply this logic to the 5 channels from the output, but since the output is converted into input, only the Left (0) and Right (1) Channels can be presented. The solution was to route the output from the Left and Right channel to Virtual 1 and the Left Surround and Right Surround to Virtual 2. Using the 8x8 Matrix, a software installed when Voicemeeter Banana is installed, it is possible to route channels 5 and 6 (Left and Right Surround) from Virtual 1 as channels 1 and 2 (Left and Right) in Virtual 2, as seen in Figure 21.

With this setup, the two input devices need to be processed in the *level_meter* file.



Figure 21: 8x8 Matrix Setup.

Before the input device obtained was the default one, this time to obtain the second input a method needed to be created. The method to get the default input device and the method to obtain the various input devices is present in the *_wasapi_backend* header. The *get_second_device* method is a combined version of the two.

After some setup code, the script loops the device count creating the device list, and when it gets to the position of the desired input device, it stores it and finally returns it as an audio device.

```
static optional<audio_device> get_second_device(bool output_device) {  
  
    //Setup  
  
    vector<IMMDevice*> devices;  
    IMMDevice* device_2 = nullptr;  
    int count = 0;  
    for (UINT i = 0; i < device_count; i++)  
    {  
        IMMDevice* device = nullptr;  
        hr = device_collection->Item(i, &device);  
        if (FAILED(hr))  
        {  
            if (device != nullptr)  
            {  
                device->Release();  
            }  
            continue;  
        }  
  
        if (device != nullptr)  
            devices.push_back(device);  
  
        //Position of the Virtual 2 device  
        if (count==4)  
        {  
            device_2 = device;  
        }  
        count++;  
    }  
  
    try  
    {  
        return audio_device{ device_2, output_device };  
    }  
    catch (const audio_device_exception&)  
    {  
        return nullopt;  
    }  
}
```

Figure 22: get_second_device method.

In the *level_meter* method, it is now possible to get both devices and apply the logic used in the default device in the second device. The output now consists of the decibels of the Left, Right, Left Surround, and Right Surround channels.

5.3.1 Interface/Information System

The second part of the project focuses on presenting the information to the user in a simple and aesthetically pleasing way. The first option was to develop a radar in Javascript which allows for has the upper hand on the aesthetic part and connect it to the first part of the project with sockets and establish a connection through TCP/IP.

Upon the start of the development of this part, the IXWebSocket C++ library was chosen as a method of communication (Sergeant, 2021). The main code offers an example that creates a simple server that calls a method every time the client sends a request. The idea was to merge the already created *level_meter* with the main code from this library, but this was proven to be a bigger task than anticipated because the merge meant conflicts with dependencies.

Following this, the solution was to try to create an interface in C++ using a graphic interface framework like Qt. But when trying to open the project created for the first part, we faced the same issues that occurred with the IXWebSocket library, and since time was starting to run out, a decision was made to print the radar in the command prompt.

This has the advantage of keeping the CPU usage to a minimum and thus not having a negative impact on the FPS (Frames Per Second) recorded when in-game. Since the first part of the program can only obtain the values from four channels, this means the maximum number of directions that can be presented is eight, and since this is doable with print commands, this decision is justified.

The first step was to store the values every time the 150 milliseconds passed and then compare them between each other to find which channel was the loudest. These values are also reset before the next iteration and the console is cleared giving the impression of the radar only changing the arrow direction. The minimum sound decibel level chosen for the system was 0, which means the system recognizes any minimal sound input.

```
while(device_input->is_running() /*&& device_input_B2->is_running()*/) {
    std::this_thread::sleep_for(std::chrono::milliseconds(150));

    system("CLS");
    L = max_abs_value_input1;
    R = max_abs_value_input2;
    BL = max_abs_value_input5;
    BR = max_abs_value_input6;

    std::string direction = "none";
    float higher_db = 0;
```

```

if (L > 0) {
    higher_db = L;

    direction = "L";
}

if (R > 0)
{
    if (R > higher_db)
    {
        higher_db = R;
        direction = "R";
    }
}

if (BL > 0)
{
    if (BL > higher_db)
    {
        higher_db = BL;
        direction = "BL";
    }
}

if (BR > 0)
{
    if (BR > higher_db)
    {
        higher_db = BR;
        direction = "BR";
    }
}

```

Figure 23: Process of obtaining the loudest channel.

The program then computes the loudest channel but first takes into account if the adjacent channels have a decibel level no lower than 5 decibels. If this occurs the print will be the direction in between. This process is shown in a flow chart in Figure 24.

For example, if the Left channel is the loudest, the program then checks if the Right and Left Surround channels are within a 5-decibel level of the Left channel. If the Left Surround channel is within the 5 decibels, then the direction presented will be 90 degrees the Left and if the Right channel is within the 5 decibels the direction presented will be directly in front. If both channels are within the interval the highest one will be the chosen. If neither of the Right or Left Surround channels is within 5 decibels the direction present will be the front left which is approximately 45 degrees to the left. The same logic is applied to the other channels.

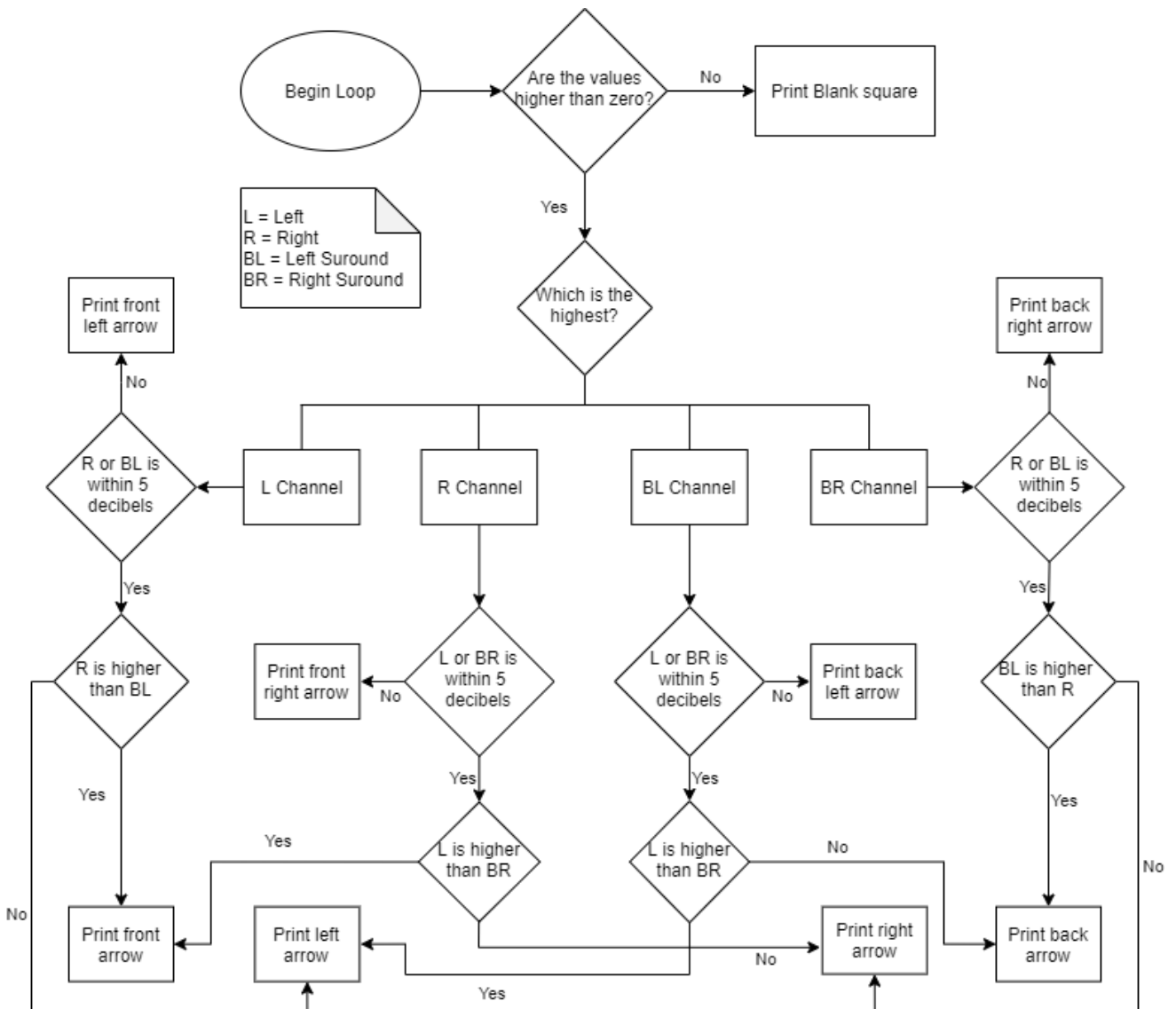


Figure 24: Flow Chart with the logic behind the calculation of the right direction.

The output also writes the direction of the arrow above the square as can be seen in Figure 25.

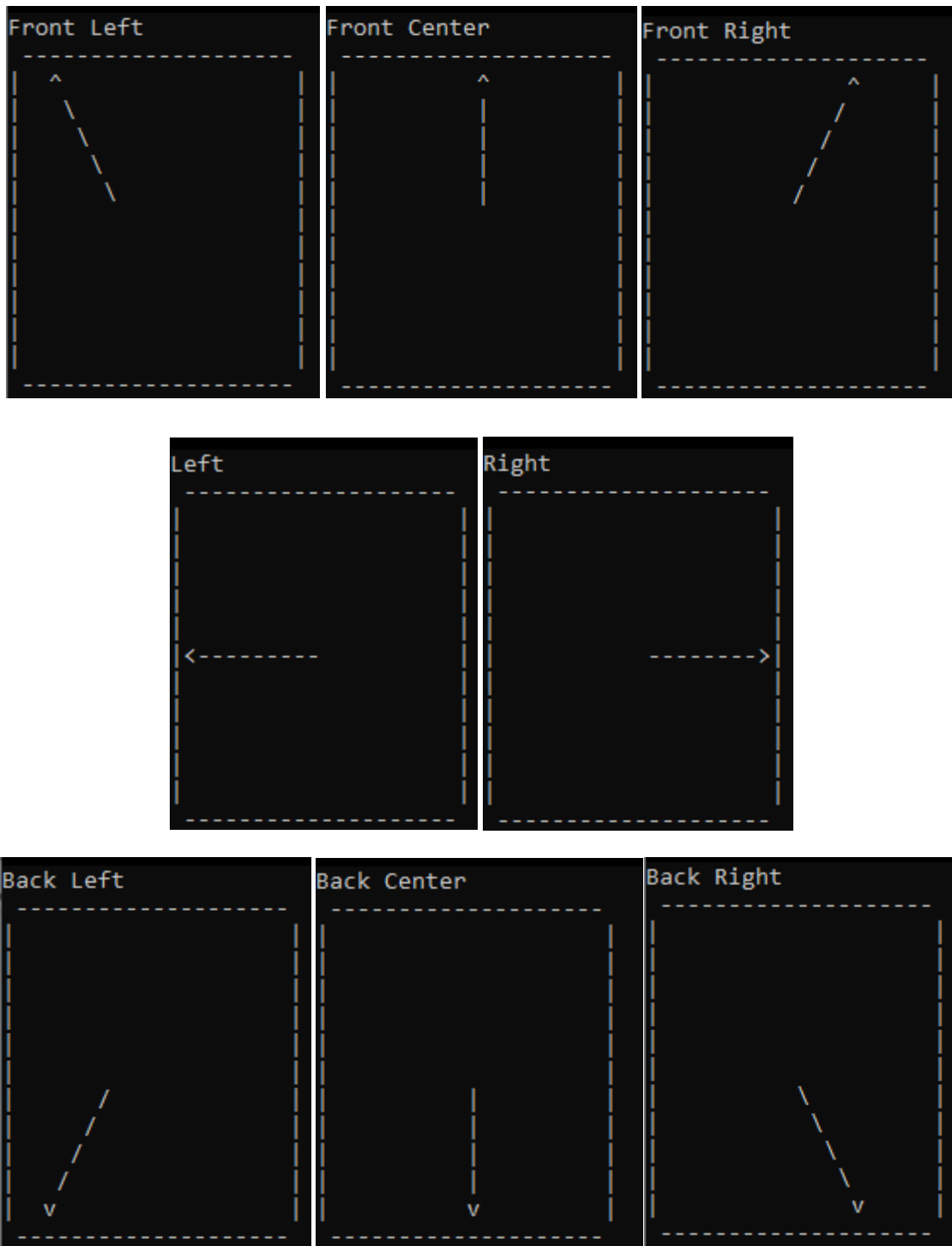


Figure 25: Examples of the output from all the possible directions.

6 Testing and Evaluation

Every project should be tested and evaluated to prove the effectiveness of the spatial identification system of surround sound software in games. For this, it's necessary feedback from the target users. In this chapter are presented the hypothesis that will be tested, the different evaluation criteria, evaluation methodologies, and the hypothesis tests that will serve to evaluate the final project. In the end, a conclusion will be drawn taking into consideration the results obtained.

6.1 Hypothesis

To verify if the software developed contributes to the gaming experience of deaf and hearing-impaired people, the hypothesis to test is if this sort of software improves the performance of the target audience and their respective gameplay.

- **Null Hypothesis (H_0)** - The implementation of a simple system of spatial surround sound visualization does not improve or contribute to the overall gaming experience of deaf and hearing-impaired people.
- **Alternative Hypothesis (H_1)** - The implementation of a simple system of spatial surround sound visualization does improve or contribute to the overall gaming experience of deaf and hearing-impaired people.

The way to validate the hypothesis is through a t-test, which is a method that can be used in a small sample size of results from a normally distributed audience (Britannica, The Editors of Encyclopaedia, 2020).

6.2 Evaluation Criteria

Considering the evaluation of the final solution, the following criteria were defined to assess the quality of the software:

- **Usability** – Will measure the quality of the human-computer interaction.
- **Accuracy** – Will measure the precision of the calculation of the point of origin of the sound made by the software.
- **Relevance** – Will measure the importance of this software to deaf or hearing-impaired gamers.

6.3 Evaluation Methodology

For this project, two different evaluation methodologies will be used, the system usability scale (SUS) and a pilot study questionnaire.

The Pilot Study is a tool used to test a system in a development environment before the final release, and, although it does not guarantee the success of the system, it increases its likelihood. It also provides valuable insights for future work (Teijlingen & Hundley, 2001).

In addition to the SUS questionnaire, another questionnaire will be created to assess the user experience when using this software and the likelihood of future use in real life. Both will be used in the Pilot Study.

The ideal target audience intended for this project would be people with some degree of hearing loss. Since the testing experience should be equal to every tester and with the current COVID-19 situation, a decision was made to change the target audience, allowing anyone to test the software.

The user will play the FPS game, Counter-Strike: Global Offensive. There will be two sessions, one without sound while the software runs on a second monitor and the other with sound using

a surround sound setup or using a headset. They will play 5 rounds of practice against bots on any difficulty.

The user will also do the setup needed in order to use the software while playing. For this, a guide was created and all users must follow it so all of them have the same experience and consequently evaluate the same aspects. This guide can be consulted in the Annexes chapter.

6.3.1 Software Experience Questionnaire

The Software Experience Questionnaire is made up of five pages. The first page starts by gathering information about the user, such as age, sex, and if there is any level of difficulty hearing, then obtains information regarding the degree of familiarity the user has with Counter-Strike: Global Offensive and the number of hours they play weekly. The answers for the age question range from less than twenty years, twenty to forty years, or more than 40 years. These ranges were chosen according to the information gathered in section 1.5 (Types of Hearing Loss).

The questions on the second page inquire the user about the setup they used when testing the software. For example, if they use a headset or a surround sound setup, what is the refresh rate and resolution of the monitor, and if they had any kind of visual or audio issues regarding isolation from the outside world, such as a television playing in the background or noise from the street, respectively.

The following two pages contain the same questions, but one is dedicated to the experience using sound and the other to the experience without it. The first question is the most important asking the user if they were able to identify the origin of sound and to what degree which directly relates to the accuracy criteria. The following questions are related to the immersion during the testing session, and the last question relates to the performance of the computer, which inquires the user about the average frames per second they had during the session.

The final page allows the user to give any kind of feedback they desire about the session or on how to improve the software. An example of the questionnaire will be present in the Annexes chapter.

6.3.2 System Usability Scale (SUS)

The System Usability Scale (SUS) is a worldwide used questionnaire that has the purpose to evaluate the usability of a certain system (Lewis, 2018). The most used version consists of a 10-question form that alternates between a positive and a negative tone (Figure 27).

The SUS score is obtained by subtracting 1 from the score in odd-numbered questions, the ones with a positive tone, and for the even-numbered questions, the ones with a negative tone, by subtracting the question score from 5. After adding all the numbers, multiply the sum by 2.5 to obtain the standard SUS score. This system requires all the question ratings so when a question is left empty, it should be given a score of 3 (the halfway point of the 5-point scale) (Lewis, 2018).

Another way to obtain the standard SUS score is with the following equation:

$$SUS = 2.5 (20 + \text{SUM}(SUS01, SUS03, SUS05, SUS07, SUS09) - \text{SUM}(SUS02, SUS04, SUS06, SUS08, SUS10))$$

Figure 26: System Usability Scale equation.

Through the use of the same questions, the score obtained from this methodology has more impact since it can be compared with other scores and thus gives a more reliable answer to the usability issue.

The System Usability Scale Standard Version		Strongly Disagree					Strongly Agree				
		1	2	3	4	5	1	2	3	4	5
1	I think that I would like to use this system frequently.		0	0	0	0	0				
2	I found the system unnecessarily complex.		0	0	0	0	0				
3	I thought the system was easy to use.		0	0	0	0	0				
4	I think that I would need the support of a technical person to be able to use this system.		0	0	0	0	0				
5	I found the various functions in this system were well integrated.		0	0	0	0	0				
6	I thought there was too much inconsistency in this system.		0	0	0	0	0				
7	I would imagine that most people would learn to use this system very quickly.		0	0	0	0	0				
8	I found the system very awkward to use.		0	0	0	0	0				
9	I felt very confident using the system.		0	0	0	0	0				
10	I needed to learn a lot of things before I could get going with this system.		0	0	0	0	0				

Figure 27: System Usability Scale Questionnaire (Bangor, et al., 2008).

6.4 Final Results

In this section, the results obtained from the questionnaires of the pilot study will be analyzed and then discussed considering the criteria defined in section 6.2.

In total, twenty-two people participated in the testing phase. The participants were composed of mostly friends and family, which facilitated the recruitment and the session planning, and avoided extra contact with people than the student is not normally in contact with, thus preventing unnecessary exposure to the COVID-19 virus.

Every participant played the game firstly with only in-game sound and then with the software and without in-game sound. After, each one of the testers answered the two questionnaires mentioned before in Google Forms. This tool allows the exportation of the results to an Excel sheet where it is easier to handle the answers. The tables with the results from the questionnaires will be present in the Annexes chapter.

In this analysis, the goal is to verify if the hypothesis tested is valid or not.

6.4.1 Software Experience Questionnaire Results

This section will be divided into four sections, firstly the results from the first page of the questionnaire will be analyzed to summarize the average tester, secondly the second page, which relates to the setup used. The third section will be a comparison of the answers between pages four and five, which inquires the user about the in-game experience. The last section will describe the feedback obtained.

6.4.1.1 Tester

The data obtained from the first page indicates that the average age group was the 20-40 years old with 77.3%, while the rest is almost evenly divided by the users with less than 20 years and more than 40 years. Most of the testers were also Male with a 59.1 percentile (Figure 28).

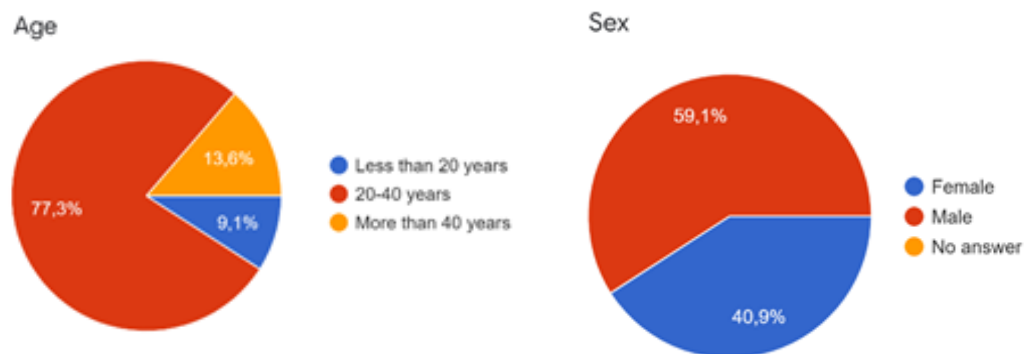


Figure 28: Age and Sex question results.

The third question inquires the tester about hearing loss. With the user not being obliged to have hearing loss to test the system, the majority of testers did not have any degree of hearing loss with only 13.6% responding that they have some difficulties as can be seen in Figure 29.

Do you have any hearing difficulties while using stereo audio?

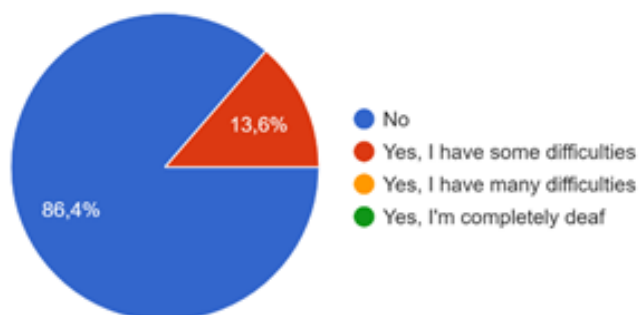


Figure 29: Hearing loss question results.

The remaining questions ask about the average number of hours played on pc per week and the degree of familiarity with playing Counter-Strike: Global Offensive (CS:GO). In the first question, the most common answer was more than 20 hours, and 27.3% answered that they do not play games at all. This gives us a range of users that range from people that experience every day with games and people that do not even approach them (Figure 30).

How many hours do you spend playing games on pc per week?

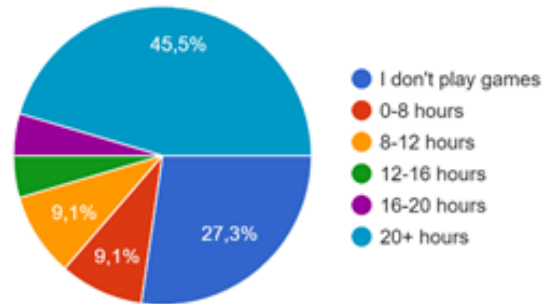


Figure 30: Hours spent playing games on pc per week question results.

The question about CS:GO records a percentile of 27.3 much like the percentile recorded in the previous question. With this is safe to assume that the users that said that they play games during the week, have experienced or are avid CSGO players.

Have you played Counter-Strike: Global Offensive?

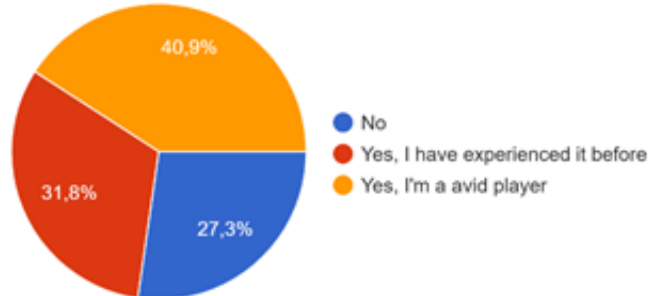


Figure 31: Counter-Strike: Global Offensive results.

6.4.1.2 Setup

The majority of the users (63.4%) used a headset while testing with sound, which is usually the standard way of playing games. The remaining used a surround sound system. The majority of monitors used while testing on average had a 60Hz refresh rate and used a 1920x1080 resolution which is the most common, with only some exceptions using one that had 120Hz or more. The data graphics related to these questions are present in the Attachment chapter.

In this section is also asked about any isolation problem during the session, specifically an audio or visual one. In Figure 32 it is possible to see that out of the twenty-two users, only two answered that they had audio-related issues.

Did you experience any isolation issues while playing?

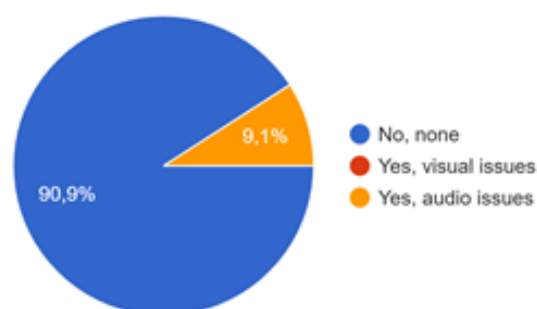


Figure 32: Isolation issues question results.

6.4.1.3 Comparison between in-game experiences with and without sound

The first question is the most important, which asks the user if they were able to identify the origin of sound while playing. The average response when playing with sound shows that all of the users were able to always or most of the time identify it. When testing without sound and using the software, the majority of the users were able to identify it sometimes, with some of the users being able to identify it most of the time.

From the 14 users that answered that they could identify the origin of sound all the time, six of them changed the answer to "Yes, most of the times" when using the software and the rest answered with "Yes, sometimes" which indicates a reduction in the ability but also means that the user still could identify if to some degree.

The 3 participants that responded that they had some degree of hearing loss answered this question with "Yes, most of the times" in the session with sound, and when using the software, they answered with "Yes, sometimes". This does not indicate a big drop off in the ability to identify the sound which is a great indicator of the success of the purpose of the radar. Figure 33 shows the graphics that contain the data from this question on pages 3 and 4, respectively.

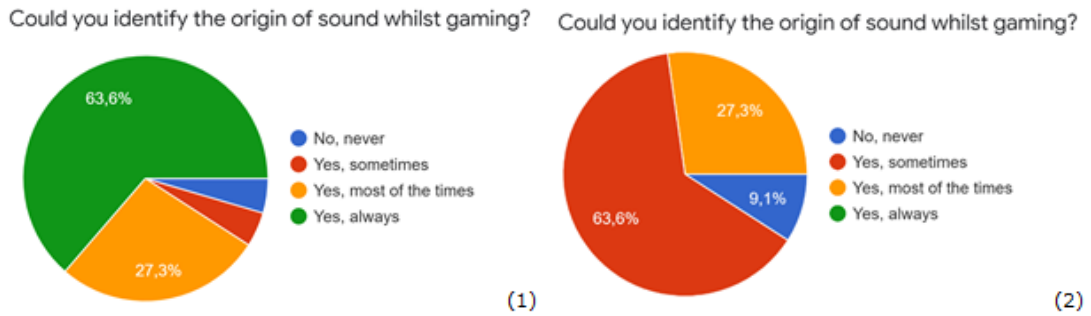


Figure 33: Identify the origin of sound question results with sound (1) versus without sound and with the software (2).

With this is important to reference that the percentage of people that agreed that they felt disoriented, increased when using the software. But the majority still could use the radar to guide themselves.

The question related to the aesthetic of the radar shows that the answers are split with twelve people agreeing that it was aesthetically pleasing and the remaining disagreeing, which indicates that there is room to improve visually.

In the session, while using the software, the number of users that thought the experience was unnaturally increased, as well as the number of people that disagreed with being deeply concentrated in the game. This issue is probably related to the fact that the user played the game with another monitor, contrary to the usual experience that is playing only with a single monitor, which also increases the probability of the user to disperse their attention. In either session, the majority of users still were fully occupied with playing the game.

In terms of performance, it is possible to access that four out of the twenty-two users had a decrease in frames per second. Still, no user registered frames below 25 which would mean that the game was unplayable with the software running. With these results, the software is viable and causes next to no downgrades in performance. Figure 34 shows these results.

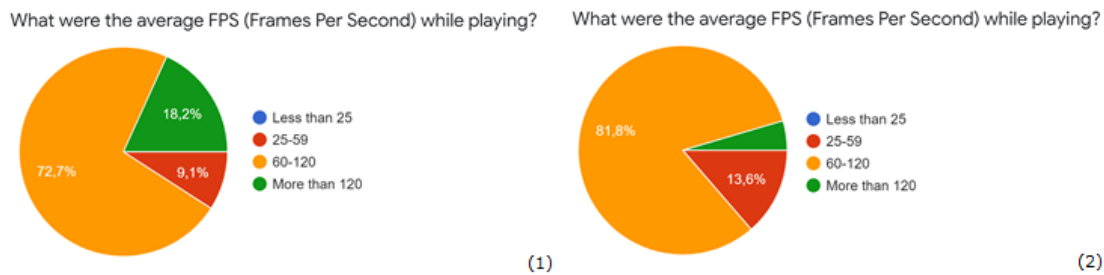


Figure 34: Frames Per Second question results.

6.4.1.4 Feedback

The section of the questionnaire designed for the feedback was not obligatory, having only six responses. Three of the answers argue that when playing with no sound is better to use the software than nothing which can relate to the relevance criteria mentioned before. The three other opinions advise that the radar should be placed somewhere on the main screen, nulling the need for a second monitor.

6.4.1 System Usability Scale Results

As mentioned before, the SUS Questionnaire allows us to attribute a numerical value to the evaluation of the usability of any system. To obtain this value, the mean value of each question will be determined and then used in the equation shown in Figure 26.

After applying the calculations, the system got a score of 70. With the average score being around 68, this means that 70 is above average. It is a score equivalent to a B- grade (Sauro, 2011).

This grade is seen as a passing grade but not a perfect one, but since this system is in its early stages it is an acceptable score. One of the reasons the grade might not be higher is because also some of the questions might not be logical to the tester when related to the system presented. For example, a question such as question number one: "I think that I would like to use this system frequently." is less relevant to most of the testers, that did not have any kind of hearing loss, since the software is not targeted to them.

The mean score for each question can be seen in Table 12. Odd-numbered questions should be as high as possible, while even-numbered questions should be as low as possible, with a score of one meaning “Strongly Disagree” and a score of five meaning “Strongly Agree”.

It is possible to observe that the only question that has a bad result is question number eight which indicates that the system was awkward to use, which most likely relates to the use of a two-monitor setup for the use of the system.

Table 12: SUS Questionnaire Average Results.

Intensity of Importance	Mean Score
1 - I think that I would like to use this system frequently.	3.27
2 - I found the system unnecessarily complex.	1.77
3 - I thought the system was easy to use.	4.23
4 - I think that I would need the support of a technical person to be able to use this system.	2.32
5 - I found the various functions in this system were well integrated.	3.77
6 - I thought there was too much inconsistency in this system.	2.36
7 - I would imagine that most people would learn to use this system very quickly.	4.23
8 - I found the system very awkward to use.	2.59
9 - I felt very confident using the system.	3.41
10 - I needed to learn a lot of things before I could get going with this system.	1.86

This is also one question that is important to single out since it relates directly to the relevance criteria named in section 6.2. The result obtained in this question was 3.27 which indicates that more often than not the users would use the software. Figure 35 shows the scores for this question.

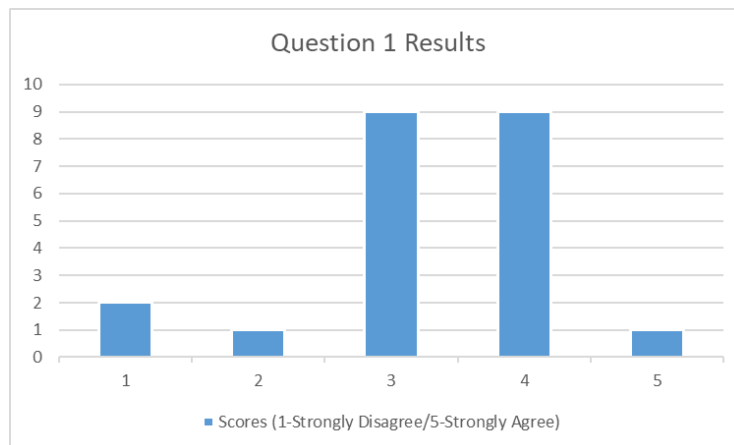


Figure 35: Sus Question 1 Results.

6.4.1 Hypothesis Validation

As said in section 6.1, a t-test will be done to validate the hypothesis. The type that will be applied is a one-tail one-sample mean t-test. A one-tail test is appropriate if there is only a need to determine the difference in a single direction (Britannica, The Editors of Encyclopaedia, 2020).

The values that are going to be used in this test in the response to the question “Could you identify the origin of origin whilst gaming?”. Since the question was answered with text, firstly they will be converted into numerical values. Table 13 shows this conversion.

Table 13: Conversion of answers into a numerical value.

Answer	Numerical Value
No, never	1
Yes, sometimes	2
Yes, most of the times	3
Yes, always	4

The data set was obtained from question one of page four of the Software Experience Questionnaire, which is related to the session without sound but with the software. The test will evaluate if the mean average of the data set is above 1.9, thus signifying that the average user was able to identify the origin of sound sometimes.

Firstly, the value of alpha chosen was 0.005 which is the most common and indicates a 5% probability of rejecting a null hypothesis that is true. The t-value obtained from the data set will depend on the alpha value and number of answers. In this case, the value is 1.740. For the null hypothesis to be rejected the t-value needs to be higher than 1.740 and the p-value needs to be less than 0.05 (Britannica, The Editors of Encyclopaedia, 2020).

Table 14: t-test results obtained.

t-test: One Sample	Score
Mean	2.18
Variance	0.35
Observations	22
Hypothesized Mean	1.9
Degrees of freedom	21
Test Statistic	2.25
P(T<=t) one-tail	0.02
Critic t one-tail	1.72

Table 14 shows that the mean value obtained from the data set was ≈ 2.18 with a t-value of ≈ 2.25 which is higher than 1.740 thus rejecting the null hypothesis. The recorded alpha value was of ≈ 0.02 which is also lower than 0.05 meaning it also rejects the null hypothesis. It is possible to conclude that there is enough evidence to infer that on average the user could identify sound while playing using the software.

7 Conclusion

In this chapter, we will discuss the objectives accomplished, the limitations, future work, and make a final assessment of the validity of use for the spatial identification system of surround sound software in games.

The objectives established at the beginning of the elaboration of this document were all achieved. In the phase of development, the implementation of a spatial identification system of surround sound in games and a complementing interface that conveys the information obtained were achieved. In the testing phase and with the use of evaluation methodologies the accuracy of the system was validated. The hypothesis present in section 6.1 was also validated with the results presented which also indicates the usefulness of the software.

With this said, the software has some limitations to its use. The developed software in its current state has the need for mixing software that allows the user to route the several surround sound output channels as two different inputs. The user also needs to use two monitors simultaneously.

This software was created with the intent to show hearing-impaired people information that they otherwise would not be able to obtain. Although the software developed achieves this goal, several improvements can be made that would benefit the user. The most important and necessary step to take would be to integrate the interface with the game, for example by showing the radar on the same screen as the game as an overlay, in order to reduce the effort and increase the ability of the player to identify the origin of sound which was not possible due to the time limitations in this project. The user should also be able to run the software without

having to install various applications and configure them. One step to solving this problem is to make a unified setup that would make automatic the installation of the final system.

Another way to increase the success of the software, as mentioned in section Comparison between in-game experiences 6.4.1.3, is to improve the interface visually. For example, in addition to the arrows and text shown in the current output, by color-coding the directions the user would have another way to absorb the information.

A final possible improvement would relate to the sound level that is shown by the software. Adjustments could be made so that the direction of the sound will only show louder sounds, by changing the decibel level measured by the system, or show only certain sounds, such as gunshots, by obtaining only certain frequency levels. Since the software shows every sound, if video games allowed to ignore the sound that comes from the player like it is possible to ignore SFX sounds or music, the use of the software would be an easier process.

In conclusion, the software developed was proven to aid the users that tried it, with some recommending its use when playing without sound. With the validation of the hypothesis, the software is worth exploring even further.

8 References

Aaron Marks, J. N., 2010. *Aaron Marks Special: Function of Game Sound Effects*. [Online]
Available at: <https://designingsound.org/2010/10/15/aaron-marks-special-function-of-game-sound-effects/>
[Accessed 3 June 2021].

Accenture, 2021. *Gaming: The next super platform*. [Online]
Available at: <https://www.accenture.com/au-en/insights/software-platforms/gaming-the-next-super-platform>
[Accessed 23 June 2021].

Amatriain, X., Arumi, P. & Garcia, D., 2006. CLAM: A Framework for Efficient and Rapid Development of Cross-Platform Audio Applications. In: *MM '06: Proceedings of the 14th ACM international conference on Multimedia*. New York, NY, USA: Association for Computing Machinery.

ASUS, 2019. *Sonic Radar 3 UWP introduction*. [Online]
Available at: <https://www.asus.com/en/support/FAQ/1038250/>
[Accessed 28 May 2021].

Audinate, 2021. *Dante Virtual Soundcard*. [Online]
Available at: <https://www.audinate.com/products/software/dante-virtual-soundcard>
[Accessed 25 June 2021].

Bangor, A., Kortum, P. T. & Miller, J. T., 2008. An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction*, pp. 574-594.

Bierre, K. et al., 2005. *Game Not Over: Accessibility Issues in Video Games*, San Francisco, California : s.n.

Britannica, The Editors of Encyclopaedia, 2020. *Student's t-test*. [Online]
Available at: <https://www.britannica.com/science/Students-t-test>
[Accessed 24 June 2021].

Collins, K., 2008. *Game Sound: An Introduction to the History, Theory, and Practice of Video*. London, England.: The MIT Press.

Craven, C., n.d. *Deaf and Hard of Hearing Accessibility Guide*. [Online]
Available at: <https://caniplaythat.com/basic-accessibility-options-for-deaf-hoh-players/>
[Accessed 16 June 2021].

E2ESOFT, 2021. *Virtual Sound Card*. [Online]
Available at: <https://www.e2esoft.com/vsc/>
[Accessed 25 June 2021].

Facebook, 2020. *React Versions*. [Online]
Available at: <https://github.com/facebook/react/releases>
[Accessed 15 February 2021].

Forman, E. H. & Gass, S. I., 2001. THE ANALYTIC HIERARCHY PROCESS—AN EXPOSITION. *Operations Research*.

George, A. D., 2020. *Desktop Guide (Windows Forms .NET)*. [Online]
Available at: <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/overview/>
[Accessed 15 February 2021].

Google, 2020. *Introduction to Angular concepts*. [Online]
Available at: <https://angular.io/guide/architecture>
[Accessed 15 February 2021].

Grady, R. B., 1992. *Practical Software Metrics for Project Management and Process Improvement*. NJ, USA: s.n.

Grimshaw, M., 2004. *SOUND AND IMMERSION IN THE FIRST-PERSON SHOOTER*,
Wolverhampton: s.n.

Hobbs, E., 2004. *A Roundup of Strange Science for the Month*. [Online]
Available at: <http://www.unmuseum.org/soearch/over0104.htm>
[Accessed 10 February 2021].

ITU, 2012. *Multichannel stereophonic sound system*. [Online]
Available at: https://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.775-3-201208-1!!PDF-E.pdf
[Accessed 5 February 2021].

Jens Blauert, R. A. B., 1997. *Spatial Hearing: The Psychophysics of Human Sound Localization* by Jens Blauert. *MIT Press*.

John M.Lasak, P. A. T. D. L., 2014. *Hearing Loss: Diagnosis and Management. Primary Care: Clinics in Office Practice*, March, pp. 19-31.

Kambil, A., Ginsberg, A. & Bloch, M., 1996. RE-INVENTING VALUE PROPOSITIONS. *Stern School of Business*.

Kelley, A., 2020. *libsoundio*. [Online]
Available at: <https://github.com/andrewrk/libsoundio>
[Accessed 20 June 2021].

Koen, P. et al., 2001. Providing Clarity and A Common Language to the “Fuzzy Front End”. *Research-Technology Management*, pp. 46-55.

Lalic, B. & Palcic, L., 2009. ANALYTICAL HIERARCHY PROCESS AS A TOOL FOR SELECTING AND EVALUATING PROJECTS. *International Journal of Simulation Modelling*.

Lewis, J. R., 2018. The System Usability Scale: Past, Present, and. *International Journal of Human-Computer Interaction*, pp. 577-590.

Lundqvist, J., 2010. *In a real game situation, is HRTF-enhanced game audio preferable over regular stereo game audio?*, Luleå: s.n.

MDN Web Docs, 2021. *Web Audio API*. [Online]
Available at: https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API
[Accessed 20 June 2021].

Messner, S., 2016. *How Overwatch uses sound to pinpoint threats you can't see*. [Online]
Available at: <https://www.pcgamer.com/how-overwatch-uses-sound-to-pinpoint-threats-you-cant-see/>
[Accessed 18 February 2021].

Morrison, G., 2019. *Dolby Atmos: Why it's cool, how it works and how to get it*. [Online]
Available at: <https://www.cnet.com/news/dolby-atmos-why-its-cool-how-it-works-and-how-to-get-it/>
[Accessed 15 February 2021].

Nash SD, C. K. K. R. K. B. N. F. H. G. P. J. T. T., 2011. *The prevalence of hearing impairment and associated risk factors: the Beaver Dam Offspring Study..* s.l.:s.n.

NHK World, 2011. *22.2 Multichannel Audio Format Standardization Activity*. [Online]
Available at: <http://www.nhk.or.jp/strl/publica/bt/en/fe0045-6.pdf>
[Accessed 5 February 2021].

Osterwalder, A. & Pigneur, Y., 2010. *Business Model Generation*. s.l.:John Wiley & Sons.

Park, M., 2020. *Ubisoft will rebuild how sound works in Rainbow Six Siege*. [Online]
Available at: <https://www.pcgamer.com/ubisoft-will-rebuild-how-sound-works-in-rainbow-six-siege/>
[Accessed 18 February 2021].

Parsons, T., 2020. *Apple spatial audio: what is it? How do you get it?*. [Online] Available at: <https://www.whathifi.com/advice/what-is-apple-spatial-audio> [Accessed 15 June 2021].

Porter, M. E., 1985. *COMPETITIVE ADVANTAGE: Creating and Sustaining Superior Performance*. United States of America: s.n.

Rayleigh, L., 1907. *On our Perception of Sound Direction*, s.l.: s.n.

Razer, 2021. *THX SPATIAL AUDIO*. [Online] Available at: <https://www.razer.com/p/RZ11-13000-0001> [Accessed 15 June 2021].

Reis, J., 2020. *ANGULAR VS REACT: A COMPARISON OF BOTH FRAMEWORKS*. [Online] Available at: <https://www.imaginarycloud.com/blog/angular-vs-react/> [Accessed 15 February 2021].

Saaty, R. W., 1987. THE ANALYTIC HIERARCHY PROCESS-WHAT IT IS AND HOW IT IS USED. *Mathematical Modelling*, 9(3–5), pp. 161-176.

Saaty, T. L. & Wind, Y., 1980. Marketing Applications of the Analytic Hierarchy Process. *Management Science*.

Sanden, C., Befus, C. R. & Zhang, J. Z., 2010. Camel: A Lightweight Framework for Content-Based Audio and Music Analysis. In: *AM '10: Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound*. New York, NY, USA: Association for Computing Machinery.

Santos, S., 2020. *Is Surround Sound Good For Gaming?*. [Online] Available at: <https://www.bluecinetech.co.uk/is-surround-sound-good-for-gaming/> [Accessed 18 February 2021].

Sauro, J., 2011. *Measuring Usability with the System Usability Scale (SUS)*. [Online] Available at: <https://measuringu.com/sus/> [Accessed 21 June 2021].

Somberg, G., Davidson, G. & Doumler, T., 2019. *A Standard Audio API for C++: Motivation, Scope, and Basic Design*, s.l.: s.n.

Stefan Röttger, E. S., 2006. *Mismatch negativity on the cone of confusion*, Berlin: s.n.

Stroustrup, B., 2014. *Bjarne Stroustrup - The Essence of C++*. Edinburgh, Scotland: s.n.

Teijlingen, E. R. v. & Hundley, V., 2001. The importance of pilot studies. *Social Research Update*, Volume 35.

The Qt Company, 2020. *Qt 6.0*. [Online]
Available at: <https://doc.qt.io/qt-6/index.html>
[Accessed 15 February 2021].

Valve Corporation, 2016. *Holiday Season never sounded so good..* [Online]
Available at: <https://blog.counter-strike.net/index.php/2016/12/17260/>
[Accessed 18 February 2021].

VB Audio, 2020. *Voicemeeter User Manual*. [Online]
Available at: https://vb-audio.com/Voicemeeter/Voicemeeter_UserManual.pdf
[Accessed 22 June 2021].

WHO, 2008. Potential health risks of exposure to noise from personal music players and mobile phones including a music playing function. 23 September.

Annex 1 – Software Experience Sample Survey

25/06/2021

Pilot Study Questionnaire

Pilot Study Questionnaire

Spatial Identification of Surround Sound in Games

*Obrigatório

1. Age *

Marcar apenas uma oval.

- Less than 20 years
- 20-40 years
- More than 40 years

2. Sex *

Marcar apenas uma oval.

- Female
- Male
- No answer

3. Do you have any hearing difficulties while using stereo audio? *

Marcar apenas uma oval.

- No
- Yes, I have some difficulties
- Yes, I have many difficulties
- Yes, I'm completely deaf

4. How many hours do you spend playing games on pc per week? *

Marcar apenas uma oval.

- I don't play games
- 0-8 hours
- 8-12 hours
- 12-16 hours
- 16-20 hours
- 20+ hours

5. Have you played Counter-Strike: Global Offensive? *

Marcar apenas uma oval.

- No
- Yes, I have experienced it before
- Yes, I'm a avid player

Setup Characteristics

Related to the setup used during testing

6. Which of the following sound setups were used while testing with sound? *

Marcar apenas uma oval.

- Headset
- Surround Sound Setup

7. What is the refresh rate of your monitor? *

Marcar apenas uma oval.

- 60Hz
- 120Hz
- 144Hz or more

8. What is the resolution of your monitor (in pixels)? *

Marcar apenas uma oval.

- 1280 × 1024
- 1600 × 1200
- 1680 × 1050
- 1920 × 1080
- 1920 × 1200

9. Did you experience any isolation issues while playing? *

Marcar apenas uma oval.

- No, none
- Yes, visual issues
- Yes, audio issues

Avançar para a pergunta 10

TESTING WITH SOUND

Related to the testing experience using sound



10. Could you identify the origin of sound whilst gaming? *

Marcar apenas uma oval.

- No, never
 Yes, sometimes
 Yes, most of the times
 Yes, always

11. It was aesthetically pleasing *

Marcar apenas uma oval.

- I agree
 I disagree

12. It felt like an unnatural experience *

Marcar apenas uma oval.

- I agree
 I disagree

13. I felt disoriented while playing *

Marcar apenas uma oval.

- I agree
 I disagree

14. I was deeply concentrated in the game *

Marcar apenas uma oval.

- I agree
 I disagree

15. I was fully occupied with the game *

Marcar apenas uma oval.

- I agree
 I disagree

16. What were the average FPS (Frames Per Second) while playing? *

Marcar apenas uma oval.

- Less than 25
 25-59
 60-120
 More than 120

Avançar para a pergunta 17

TESTING WITHOUT SOUND

Related to the testing experience without sound



17. Could you identify the origin of sound whilst gaming? *

Marcar apenas uma oval.

- No, never
 Yes, sometimes
 Yes, most of the times
 Yes, always

18. It was aesthetically pleasing *

Marcar apenas uma oval.

- I agree
 I disagree

19. It felt like an unnatural experience *

Marcar apenas uma oval.

- I agree
 I disagree

20. I felt disoriented while playing *

Marcar apenas uma oval.

- I agree
 I disagree

21. I was deeply concentrated in the game *

Marcar apenas uma oval.

- I agree
 I disagree

22. I was fully occupied with the game *

Marcar apenas uma oval.

- I agree
 I disagree

23. What were the average FPS (Frames Per Second) while playing? *

Marcar apenas uma oval.

- Less than 25
 25-59
 60-120
 More than 120

**FINAL
THOUGHTS**

This section gives the user the option to give any kind of feedback they desire about the session, the relevance of this software or on how to improve it.

24. Please provide any additional feedback you may have?

Este conteúdo não foi criado nem aprovado pela Google.

Google Formulários

Annex 2 – System Usability Scale Sample Survey

25/06/2021

System Usability Scale (SUS)

System Usability Scale (SUS)

The answers range from 1 (Strongly Disagree) and 5 (Strongly Agree).

*Obrigatório

1. 1 - I think that I would like to use this system frequently. *

Marcar apenas uma oval.

- 1
 2
 3
 4
 5

2. 2 - I found the system unnecessarily complex. *

Marcar apenas uma oval.

- 1
 2
 3
 4
 5

3. 3 - I thought the system was easy to use. *

Marcar apenas uma oval.

- 1
 2
 3
 4
 5

4. 4 - I think that I would need the support of a technical person to be able to use this system. *

Marcar apenas uma oval.

- 1
 2
 3
 4
 5

5. 5 - I found the various functions in this system were well integrated. *

Marcar apenas uma oval.

- 1
 2
 3
 4
 5

6. 6 - I thought there was too much inconsistency in this system. *

Marcar apenas uma oval.

- 1
 2
 3
 4
 5

7. 7 - I would imagine that most people would learn to use this system very quickly.

*

Marcar apenas uma oval.

1

2

3

4

5

8. 8 - I found the system very awkward to use. *

Marcar apenas uma oval.

1

2

3

4

5

9. 9 - I felt very confident using the system. *

Marcar apenas uma oval.

1

2

3

4

5

10. 10 - I needed to learn a lot of things before I could get going with this system. *

Marcar apenas uma oval.

- 1
- 2
- 3
- 4
- 5

Este conteúdo não foi criado nem aprovado pela Google.

Google Formulários

Annex 3 – Software Experience Questionnaire Results

Age	Sex	How many hours do you spend playing games on pc per week?	Do you have any hearing difficulties while using stereo audio?
20-40 years	Male	20+ hours	Yes, I have some difficulties
20-40 years	Male	20+ hours	No
20-40 years	Male	20+ hours	No
20-40 years	Male	20+ hours	No
20-40 years	Male	12-16 hours	No
20-40 years	Male	20+ hours	No
20-40 years	Male	20+ hours	No
Less than 20 years	Female	I don't play games	No
20-40 years	Female	I don't play games	No
20-40 years	Female	0-8 hours	No
20-40 years	Male	20+ hours	No
20-40 years	Female	8-12 hours	No
20-40 years	Female	I don't play games	No
Less than 20 years	Female	I don't play games	No
More than 40 years	Female	0-8 hours	No
More than 40 years	Male	I don't play games	Yes, I have some difficulties
More than 40 years	Female	I don't play games	Yes, I have some difficulties
20-40 years	Male	20+ hours	No
20-40 years	Male	20+ hours	No
20-40 years	Female	20+ hours	No
20-40 years	Male	8-12 hours	No
20-40 years	Male	16-20 hours	No

Have you played CS:GO?	Which of the following sound setups were used while testing with sound?	What is the refresh rate of your monitor?	What is the resolution of your monitor (in pixels)?	Did you experience any isolation issues while playing?
Yes, I'm a avid player	Surround Sound Setup	60Hz	1920 × 1200	No, none
Yes, I have experienced it before	Headset	60Hz	1920 × 1200	No, none
Yes, I'm a avid player	Headset	60Hz	1920 × 1080	Yes, audio issues
Yes, I have experienced it before	Headset	60Hz	1920 × 1080	No, none
Yes, I'm a avid player	Headset	144Hz or more	1920 × 1080	No, none
Yes, I'm a avid player	Headset	60Hz	1920 × 1080	Yes, audio issues
Yes, I'm a avid player	Headset	60Hz	1920 × 1080	No, none
No	Headset	60Hz	1920 × 1080	No, none
No	Surround Sound Setup	60Hz	1920 × 1200	No, none
Yes, I have experienced it before	Surround Sound Setup	60Hz	1920 × 1080	No, none
Yes, I'm a avid player	Surround Sound Setup	60Hz	1920 × 1080	No, none
Yes, I'm a avid player	Surround Sound Setup	60Hz	1920 × 1080	No, none
Yes, I have experienced it before	Surround Sound Setup	60Hz	1920 × 1080	No, none
No	Surround Sound Setup	60Hz	1920 × 1080	No, none
No	Surround Sound Setup	60Hz	1920 × 1080	No, none
No	Headset	60Hz	1920 × 1080	No, none
No	Headset	60Hz	1920 × 1200	No, none
Yes, I have experienced it before	Headset	60Hz	1920 × 1080	No, none
Yes, I'm a avid player	Headset	60Hz	1920 × 1200	No, none
Yes, I have experienced it before	Headset	60Hz	1920 × 1080	No, none
Yes, I have experienced it before	Headset	60Hz	1920 × 1080	No, none
Yes, I'm a avid player	Headset	120Hz	1920 × 1080	No, none

Table 15: Questions related to the session with sound.

Could you identify the origin of sound whilst gaming?	It was aesthetically pleasing	It felt like an unnatural experience	I felt disoriented while playing	I was deeply concentrated in the game	I was fully occupied with the game	What were the average FPS (Frames Per Second) while playing?
Yes, most of the times	I agree	I agree	I disagree	I agree	I agree	60-120
Yes, always	I agree	I disagree	I disagree	I agree	I agree	25-59
Yes, always	I agree	I disagree	I disagree	I agree	I agree	25-59
No, never	I agree	I disagree	I disagree	I agree	I agree	60-120
Yes, always	I agree	I disagree	I disagree	I agree	I agree	More than 120
Yes, most of the times	I agree	I disagree	I disagree	I agree	I agree	More than 120
Yes, always	I agree	I disagree	I disagree	I agree	I disagree	More than 120
Yes, always	I agree	I disagree	I disagree	I agree	I agree	60-120
Yes, always	I agree	I disagree	I agree	I agree	I agree	60-120
Yes, always	I agree	I disagree	I disagree	I agree	I agree	60-120
Yes, sometimes	I agree	I disagree	I disagree	I agree	I agree	60-120
Yes, always	I agree	I disagree	I disagree	I disagree	I agree	60-120
Yes, always	I agree	I disagree	I disagree	I agree	I agree	60-120
Yes, always	I agree	I disagree	I agree	I agree	I agree	60-120
Yes, always	I agree	I disagree	I agree	I agree	I agree	60-120
Yes, most of the times	I agree	I disagree	I agree	I agree	I agree	60-120
Yes, most of the times	I agree	I disagree	I disagree	I agree	I agree	60-120
Yes, always	I agree	I disagree	I disagree	I agree	I agree	60-120
Yes, always	I agree	I disagree	I disagree	I agree	I agree	60-120
Yes, always	I agree	I disagree	I disagree	I agree	I agree	60-120
Yes, most of the times	I agree	I disagree	I disagree	I agree	I agree	60-120
Yes, most of the times	I agree	I disagree	I disagree	I agree	I agree	More than 120

Table 16: Questions related to the session using software without sound.

Could you identify the origin of sound whilst gaming?	It was aesthetically pleasing	It felt like an unnatural experience	I felt disoriented while playing	I was deeply concentrated in the game	I was fully occupied with the game	What were the average FPS (Frames Per Second) while playing?
Yes, sometimes	I disagree	I disagree	I agree	I disagree	I agree	60-120
Yes, most of the times	I agree	I disagree	I disagree	I agree	I agree	25-59
Yes, sometimes	I disagree	I agree	I agree	I agree	I agree	25-59
Yes, sometimes	I agree	I disagree	I disagree	I agree	I agree	60-120
Yes, most of the times	I agree	I disagree	I disagree	I disagree	I agree	25-59
Yes, sometimes	I agree	I agree	I disagree	I disagree	I agree	60-120
Yes, most of the times	I disagree	I agree	I agree	I agree	I agree	60-120
Yes, sometimes	I agree	I agree	I disagree	I agree	I disagree	60-120
Yes, sometimes	I disagree	I disagree	I disagree	I agree	I agree	60-120
Yes, sometimes	I agree	I disagree	I disagree	I agree	I agree	60-120
Yes, sometimes	I disagree	I disagree	I disagree	I agree	I agree	60-120
Yes, most of the times	I agree	I disagree	I disagree	I disagree	I agree	60-120
Yes, sometimes	I disagree	I agree	I agree	I agree	I agree	60-120
Yes, most of the times	I disagree	I agree	I disagree	I agree	I agree	60-120
No, never	I agree	I agree	I agree	I agree	I agree	60-120
Yes, sometimes	I agree	I agree	I agree	I agree	I agree	60-120
Yes, sometimes	I agree	I agree	I agree	I agree	I agree	60-120
Yes, sometimes	I agree	I agree	I agree	I agree	I agree	60-120
Yes, most of the times	I agree	I disagree	I disagree	I agree	I agree	60-120
Yes, sometimes	I disagree	I agree	I disagree	I agree	I agree	60-120
No, never	I disagree	I agree	I agree	I agree	I disagree	60-120
Yes, sometimes	I disagree	I disagree	I disagree	I agree	I agree	More than 120

Please provide any additional feedback you may have?
Sem som é preferível ter o radar para nos conseguirmos orientar que não ter nada.
A experiência com o uso do radar foi vastamente superior à experiência sem qualquer indicador espacial e/ou sonoro.
Na minha humilde opinião, eu diria que jogar com um radar é melhor do que não ter nenhum auxílio, sendo assim, gostei da experiência.
In game visual windows of the terminal
Se houvesse a possibilidade de colocar o radar no ecrã do jogo, penso que seria mais útil em termos de utilizador de modo a ficar mais concentrado num só ecrã.
An in-game overlay would be better

Annex 4 – System Usability Scale Results

I think that I would like to use this system frequently.	4	3	2	3	4	1	1	3	5	3	4	4	4	3	4	4	4	3	3	3	3	4
I found the system unnecessarily complex.	1	2	3	2	1	1	3	1	2	2	2	1	1	2	3	2	1	1	1	2	3	2
I thought the system was easy to use.	5	4	4	4	5	5	3	4	4	5	3	5	4	4	4	5	5	4	4	4	4	4
I think that I would need the support of a technical person to be able to use this system.	2	3	3	1	2	5	3	2	3	2	2	1	1	3	4	2	2	2	2	2	2	2
I found the various functions in this system were well integrated.	5	4	2	5	5	3	5	3	4	4	4	4	4	4	4	4	3	3	3	4	3	3
I thought there was too much inconsistency in this system.	3	3	4	2	2	5	2	2	2	2	2	1	1	2	4	3	3	2	2	2	1	2
I would imagine that most people would learn to use this system very quickly.	5	4	3	5	5	4	4	4	5	5	3	5	4	4	5	4	5	4	4	4	4	3
I found the system very awkward to use.	2	3	3	2	1	4	1	3	2	2	4	1	2	3	4	2	3	2	4	2	3	4
I felt very confident using the system.	4	4	2	3	4	2	3	3	4	5	3	5	3	3	4	3	4	3	4	3	3	3
I needed to learn a lot of things before I could get going with this system.	1	2	1	3	1	1	2	3	1	2	3	1	1	2	4	2	2	1	2	2	2	2

Annex 5 – Session Guide

Guide – Session with sound

- 1- Open Counter-Strike: Global Offensive (CS:GO), click on the play button.
- 2- On the next window in the dropdown menu select "practice with bots", choose the Dust II map and click on the green button that says "GO".
- 3- Play 5 rounds and when finished close the game.

Guide – Session without sound using the software

Requirements: -Voicemeeter Banana installed.

-Surround sound card.

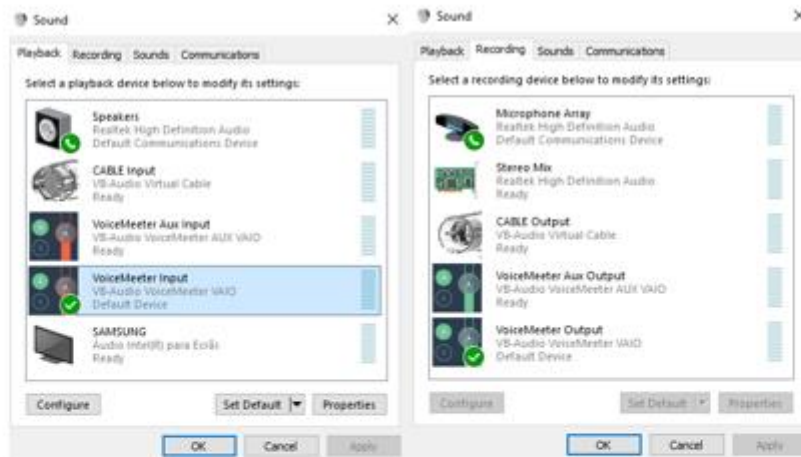
- 1- Open Voicemeeter Banana e configure the slot "Hardware Out" with the hardware speakers of the surround sound card.
- 2- In the "B1" and "B2", the buttons B1 and B2 should be activated (green color). Software should look like the following figure.



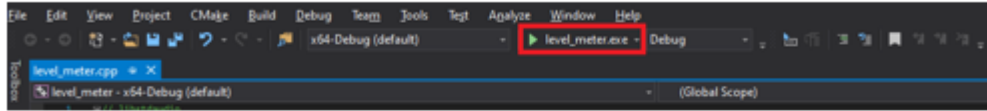
- 3- Open the "8x8 Output Matrix" software and configure it according to the figure below (This application comes with the installation of Voicemeeter Banana).



- 4- Opens sound configurations and define "Voicemeeter Input" and "Voicemeeter Output" as the default devices (The devices are also automatically installed with Voicemeeter Banana).



- 5- Open Visual Studio and select the folder that contains the project named "Radar".
- 6- After, run the executable named "level_meter.exe".



- 7- After this, the command prompt will be opened and it will display a white square without the indication of sound (preferably move the window to the second monitor so it does not get obstructed by the game).

- 8- Repeat steps 1 to 3 of the session with sound guide.