INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Anxiety reducing through a neurofeedback serious game with dynamic difficulty adjustment

LUÍS MIGUEL ALMEIDA DA MATA
Outubro de 2016

POLITÉCNICO
DO PORTO

Anxiety reducing through a neurofeedback serious game with dynamic difficulty adjustment

Luís Miguel Almeida da Mata

Thesis to attain the master's degree in
Informatics Engineering,
Graphical Systems and Multimedia specialization

Supervisor: Doctor Filipe Pacheco

Júri:
Presidente:
[Nome do Presidente, Categoria, Escola]
Vogais:
[Nome do Vogal1, Categoria, Escola]
[Nome do Vogal2, Categoria, Escola] (até 4 vogais)

Porto, 24th November, 2016

To Cristina, for being who she is.

To my parents that are always there.

Abstract

Presently, society has to deal with a large number of mental issues. Anxiety disorder is a serious concern, affecting millions of people's lives and, although methods to tackle the problem currently exist, these main treatments are being linked to some issues and improvements must be found. One of the alternatives is Neurofeedback, a biofeedback treatment, completely non-invasive and showing impressive results so far. It uses a neuroheadset equipment to read the neural activity of the brain, giving the user visual feedback about it. The purpose this, is to train the users' brain in specific regions and frequencies, allowing the subjects to learn how to voluntarily control its neural activity, even outside of the session. Current applications using this method might be too simple, which can become tedious and disengaging. Serious games can help with these issues, since it can bring enjoyment and engagement while doing this type of treatment. The interest in games' capabilities in education has been increasing over the past years, since it has been proved that games are an excellent tool for education and skill learning. Joining these concepts of game and neurofeedback, this project aims to create a serious game prototype, applying the current treatment knowledge.

The development process of a new game with neuroheadset integration, capable of reading the neural activity of the user while playing and giving the appropriate feedback, will be described in the present document. Since studies proved that a good balance between challenge and skill increases the learning performance, a dynamic difficulty adjustment system is implemented within the game, allowing the game to adapt itself to each user's skill individually, and keeping the user in a challenging, motivating zone. At the end of the document, the results of pilot test on a few subjects are shown.

Keywords: Serious Games, Neurofeedback, Dynamic Difficulty Adjustment.

Resumo

Na sociedade actual o número de problemas relacionados com perturbações mentais tem sido cada vez mais relevante, sendo esse o caso da ansiedade. O distúrbio de ansiedade é um problema que atinge milhões de pessoas e, embora existam métodos para combater este problema, estudos comprovam que estes têm algumas lacunas que podem trazer outros problemas associados, sendo portanto necessário procurar melhorias aos métodos actuais. Uma das alternativas tem apresentado excelentes resultados e denomina-se *Neurofeedback*. Este é um tratamento de *biofeedback*, não-invasivo e que utiliza um equipamento *neuroheadset* para capturar a actividade neuronal, apresentando indicações visuais sobre o comportamento do utilizador. Isto é feito com o objectivo de treinar o cérebro do utilizador, em regiões e frequências específicas, para que este seja capaz de controlar voluntariamente a sua actividade neuronal. As aplicações actualmente utilizadas com este intuito podem se tornar aborrecidas e monótonas devido à sua simplicidade. Um jogo sério pode ajudar com estes problemas, uma vez que é capaz de trazer divertimento e motivação para este tipo de tratamento. O crescente interesse nas capacidades educativas dos jogos sérios, tem identificado estes como excelentes ferramentas para a educação. Este projecto pretende portanto criar um protótipo de um jogo sério, aplicando os conceitos de *neurofeedback*.

Neste documento, é apresentado o processo de desenvolvimento de um novo jogo com integração de um neuroheadset, capaz de identificar a actividade neuronal do jogador dando respostas adequadas. Uma vez que estudos comprovam que um bom balanço entre desafio apresentado e técnica do utilizador aumenta a capacidade de aprendizagem, foi implementado também um sistema de ajuste de dificuldade dinâmica, permitindo uma adaptação do jogo a cada indivíduo e mantendo este numa zona motivante de equilíbrio entre desafio e proficiência. No final serão apresentados os resultados de um teste piloto efectuado em alguns indivíduos.

Palavras-chave: Jogos Sérios, *Neurofeedback*, Ajuste de dificuldade dinâmico.

Acknowledgements

I would like to thank everyone that helped me get to the end of this long, but gratifying road. I would like to start by thanking Doctor Filipe Pacheco for guiding me through this process and giving very important feedback during the project's development. To Hugo Sousa, PhD student, for the invaluable insight provided, availability and the evenings working together.

Thanks to Cristina, who supported me at all times, in good and bad moments, always believing I was able to go further. Then, my family for always being there, aiding me in any way they can and making me who I am today. To my friends for providing truly fun and enjoyable moments.

I would like to thank all of you. It is because of you I am here today.

Luís Mata

Table of Contents

List of Figures

List of Tables

Abbreviations and Acronyms

Acronym list

BCI	Brain-computer Interface
CNS	Central Nervous System
DDA	Dynamic Difficulty Adjustment
DLL	Dynamic-link library
GBL	Game-based Learning
GPU	Graphics Processing Unit
HUD	Heads-up display
IDE	Integrated Development Environment
OS	Operating System
PNS	Peripheral Nervous System
QEF	Quantitative Evaluation Framework
SDK	Software Development Kit
UI	User Interface
UML	Unified Modelling Language

Symbol List

Hz	Hertz (Unit of frequency)
µV	Microvolt (Amplitude)

1. Introduction

The increase in the number of cases of depression, anxiety and other mental disorders (Anxiety and Depression Association of America, 2014a), makes it necessary to find more efficient ways to treat these types of problems. The current common treatments are mostly based on medication, which has clear drawbacks and limitations in several scenarios and limited long-term effectiveness (Demos, 2005). One recently used treatment uses a brain-computer interface (BCI), and is called neurofeedback, a brain "training" method, non-invasive and with surprising results so far. This treatment involves capturing the brainwave activity and provide feedback about the subject's brain behaviour, usually through some visual interface, teaching the user how to control its own brain activity and promoting changes at the cellular level. In spite of this, the current visual applications being used with this intent are mostly repetitive and monotonous (Connolly et al., 2013a). Joining this information with the purpose of a serious game, the product aims to test a tool with improved effectiveness, creating a new game with neurofeedback capabilities, making it possible for the user, in a healthy and motivating way, to tackle the problem.

1.1 Problem Identification

It is evident that mental health problems are an increasing major public health burden. In recent years, the rate of anxiety, depression and other mental disorders suffered a significant rise: Anxiety disorders are the most common mental illness in the U.S., affecting 40 million adults, meaning 18% of the population age 18 or older (Anxiety and Depression Association of America, 2014a). As reported by Eurostat, it is also possible to observe an increase in the cases of work-related stress, depression and anxiety issues reported in Portugal (20.6% in 2007 to 23.2% in 2013) and in the United Kingdom (29.3% in 2007 to 41.8% in 2013) (Eurostat, 2013). In the European region, 27% of adult population, between 18 and 65, had experienced at least one of a series of mental disorders in the past year (WHO, 2013). Teenagers and children are also affected: Anxiety disorders affect one in eight children, which increases the risk to perform poorly in school or engage in substance abuse (Anxiety and Depression Association of America, 2014a).

This is a problem that keeps increasing in today's society, therefore, existing treatments must be enhanced and new ways to improve people's health must be found (Bystritsky et al., 2013).

1.1.1 Purpose

The purpose of this project is to test the capability of a serious game prototype to help people reduce anxiety. There has been an increasing interest in serious games and their capabilities for teaching and behaviour change (Boyle, 2013). Knowledge of the brain is also growing and new tools and techniques are being created to deal with the society's increasing mental health problems. A possible anxiety treatment currently being used is called neurofeedback, but its current common visual applications are very simple and monotonous. Therefore, a combination of the two is highly promising (Ron-Angevin and Díaz-Estrella, 2009). This project is aimed to test a prototype usefulness to those who want to see their anxiety levels reduced, in an enjoyable and healthy way, as well as professionals that need an enhancement to the current existing applications.

1.1.2 Question at Hand

The problem identified can be treated and, although the success rate can vary depending on the individual being treated, multiple approaches have been effective (Anxiety and Depression Association of America, 2014b). Two tools are commonly used as treatment: dialogue and medication, multiple times joining both (Demos, 2005). But medication has, unfortunately, some problems: The use of anxiety reducing drugs is being linked to the increased risk of Alzheimer's disease, growing by 43-51% among older people (Gage et al., 2014). It is also problematic the risk of dependence of some drugs (Physicians' Desk Reference, 2014) as well as some known uncomfortable side-effects, for example, drowsiness, poor sleep and nausea (Longo and Johnson, 2000). It is also noted that medication does not cure, rather reduces the symptoms, to allow people to live normal lives (National Institute of Mental Health, 2015). The other mentioned method, therapy, can be very powerful, but most of the times it is not used without medication, and some conditions might not improve with it only (Demos, 2005). The chosen technique, neurofeedback is showing positive, permanent results, with no side-effects. Currently, neurofeedback common visual applications are very simple and monotonous, consisting of simple two-dimensional moving bars or growing balls (Ninaus et al., 2013), that can become monotonous and unchallenging.

Video games have shown that they can be powerful tools for learning or behavioural change, and can upgrade the current neurofeedback results. As such these two concepts will be merged in order to bring improvements to current techniques.

1.1.3 Point of View

This project considers two points of view: the developer and the user.

From the point of view of the developer, it is necessary to create a new serious game, capable of following the best practices of the neurofeedback current treatments. It is necessary to learn and understand current treatment procedures, identify and translate them into enjoyable game mechanics.

The user wants to verify the ability of a new tool, a video game, to help with the anxiety issue, reducing their anxiety state. This can be done with the support of a professional, or in an uncontrolled environment, where the perfect conditions might not be achieved for optimal results.

1.1.4 Assumptions

It is assumed that the equipment necessary for this project will exist: a computer with the minimum specifications necessary and the neuroheadset connected. The user should be able to use a computer, play a game using peripherals. He or she must not suffer from epilepsy or other related conditions and should be capable of learning a new skill in a healthy way (Demos, 2005). It is expected that this game is tested in two environments, as was stated before: a controlled, professional environment, with experts that need an enhancement to the current procedures, and a less controlled one, the common user that wants to try to reduce their anxiety, where the perfect environment conditions might not be met and positive results more difficult to achieve.

It will be tested if this game is feasible and can have at least the same efficacy as other applications while being more entertaining to the users. It is assumed that it works and that it uses neural activity analysis to obtain results as it happens with current treatment applications. It is not expected for results to appear in the first sessions. Like other neurofeedback treatments, it will be necessary from 10 to 40 sessions to achieve results (Gevensleben et al., 2010).

This is a purely academic project. This is not aimed to be a replacement for current treatments or methods, an appointment with a professional is always recommended. By testing the tool, the user is complying with these assumptions, and accepts that expected results might not be achieved.

1.1.5 Engineering Information

The information necessary for this project game design and development was acquired during the student’s academic course, together with bibliography and internet communities’ information. For the clinical solution chosen, existing research papers and bibliography were used as research, as well as the important insight of an area specialist. To get the expected results, after the development of the tool, it will be necessary to test on subjects.

Some information might not be perfectly accurate, since the current knowledge on the brain is still growing, and some of its behaviour cannot be precisely explained. In order to obtain the best solution possible, this project will be continuously updated with new information from adequate sources.

1.1.6 Concepts

Serious games are applications that "*were designed for a purpose going beyond entertainment*" (Djaouti et al., 2010). They can be especially important as a behaviour changing tool (Schuller et al., 2013). The promising neurofeedback research and current uses, is a key element to this project and these joined concepts can help with currently used applications' issues. For this purpose, a dynamic difficulty adjustment (DDA) will be applied, making changes in real-time to the game mechanics, to give accurate feedback to the user. This solution is thought to be better since the current applications with this type of functionality are either too expensive or too simple, that might become boring after multiple sessions (Ron-Angevin and Díaz-Estrella, 2009).

The technologies chosen for the game development were Unity 5 with C#. This selection considered the current development knowledge, the previous experience with the tool, as well as the ease of access and price. The large community and information available also played an important role in this selection, especially comparing to similar technologies, Unity has more information available.

The headset equipment chosen for this project is the Emotiv EPOC, a neuroheadset with Electroencephalography (EEG) reading capabilities. Compared with other technologies, EEG is more affordable, portable and easy to use, making it the most popular in the field (Wang et al., 2011). This specific equipment was select due to its community support, important to help the development of the product, and its accessibility compared with other products. As for the integration with the application, the company's equipment SDK will be used, to access the headset functionalities.

1.1.7 Inferences and Implications

The current solution is thought to be the best solution since there are proven results in the teaching capabilities of video games (Blunt, 2008), and research also shows very positive results about neurofeedback treatment (Demos, 2005). Despite this, there are still some limitations: biometric sensors or EEG headsets more specifically, are still either too expensive or not so accurate in the results, with different filtering capabilities, which can influence the final result of the tool. The headset used in this project, was selected taking into account its precision and affordability. It might not be the most accurate equipment in the market, but its accessibility is a very important factor in this selection. The information gathered is limited to the existing research knowledge of the brain functions and actions. In the future, more will be known about the brain and its activity, which means that this project can be improved and updated. The individuality can also pose a problem since the reason behind the anxiety issue can differ, making this game a general tool for most of the purposes but with the possibility of having exceptions. One more problem is that 10-40% of users fail to gain BCI control (Guger et al., 2003), for unknown reasons. Symptoms will not magically go away and users must decide whether or not the treatment is right for them. It is not advisable for those with learning disorders, as the appropriate candidate must be able to learn a new skill in a healthy way (Demos, 2005). Therefore, this is not a replacement of a professional, and an appointment is still necessary.

1.2 Value Analysis

"Create and test a serious game prototype capable of helping people who want to see reduced their anxiety issues in a healthy and engaging way. "

This value proposition overview message is directed to the external entities and defines the value the serious game can bring to the customer. It specifies the product as a serious game with neurofeedback capabilities, and its unique characteristics for the target customer, people who need a way to reduce their anxiety. This message helps determine if the user is reached, showing that the product has benefits that surpass the sacrifices, independently of the different customer value perceptions.

Table 1 - Benefits and sacrifices

Domain\Scope	**Product**
Benefit	-Permanent solution, with good results that can be seen outside the sessions -Painless, non-invasive method -Engaging and motivating, avoiding repetition as much as possible
Sacrifice	-Necessary equipment -Developing the product -Some exceptions to the success of this tool are expected.

Since there are other solutions available to the user for anxiety issues, the customer will judge the value of this project, based on a multi-criteria decision, identifying their priorities. This analysis is made using an analytic hierarchy process, with the benefits and sacrifices specified in the previous table, to decide if acquiring the product.

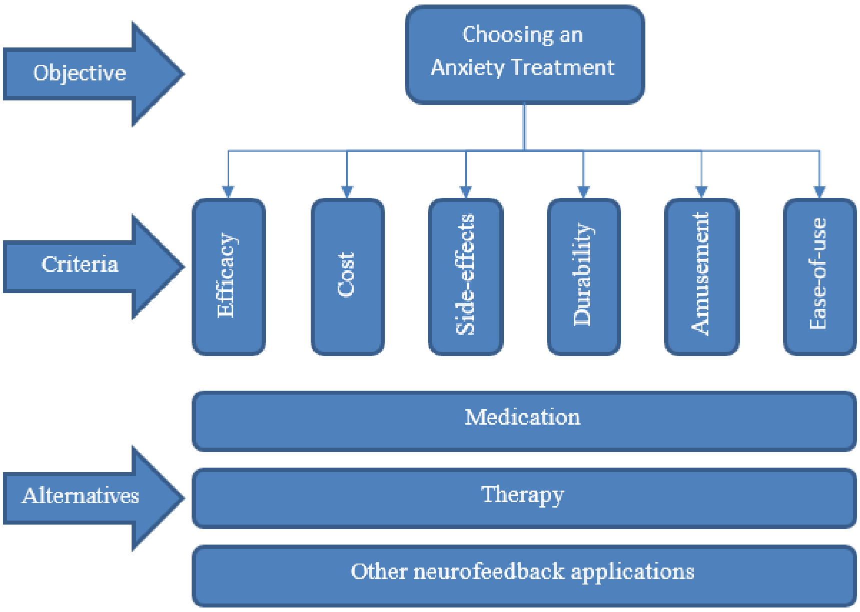


Figure 1 - Analytic Hierarchy Process

The user will have six parameters to consider to make the decision: Efficacy, cost, side-effects, durability, amusement and ease-of-use. The weights of each of these parameters can be classified by importance as can be seen in the next table:

Table 2 - Estimated value parameter importance.

	Efficacy	**Cost**	**Side-effects**	**Durability**	**Amusement**	**Ease-of-use**
Efficacy	1	3	3	1	7	7
Cost	1/3	1	1	1/3	5	5
Side-effects	1/3	1	1	1/3	5	5
Durability	1	3	3	1	5	7
Amusement	1/7	1/5	1/5	1/5	1	1
Ease-of-use	1/7	1/5	1/5	1/7	1	1

The most relevant parameters, according to the goals defined, are that the application works and helps the user. Therefore the efficacy and durability are the most relevant ones. Next comes the cost and the side-effects, since it is always expected that the user will have the least possible side-effects from a treatment and that this project can be accessed by the user with a sacrifice as small as possible. The amusement and ease-of use are also considered, although not as relevant, but still play a part in this selection process due to their influence in the final result.

1.3 Objectives

This project's main objective is to test the capabilities of a serious video game prototype, connected with a neuroheadset, to help improve the mental health of people with anxiety issues. It will be based on the neurofeedback principles, using a neuroheadset together with a serious game, so that the process is engaging and motivating.

It is expected the creation of a dungeon crawler action/adventure serious game, with DDA directly related with the neuroheadset input. With this input, a difficulty change is executed in real-time, "training" the user to be able to control the neural activity. It will have visual changes related with the player performance, so that the user can have feedback about its current brain activity, like other existing neurofeedback applications. Furthermore, it is also pretended to add specific features like random room and theme generation, to avoid boredom over multiple sessions.

All this process will be based on neurofeedback principles, which consist of "training" the brain to function properly and avoid the erratic behaviours.

1.4 Solution Identification

According to the problem identification and value analysis, it is possible to identify a potential solution. Considering what was said before, we can identify some key points to this project development.

The solution must take into account efficacy, price, durability, side-effects and other parameters as it was mentioned. It should solve the problem, by using the brain ability to improve its functionality and even its structure (Budzynski et al., 2009). For this, it is crucial to give direct, real-time feedback for the user to learn how to gain voluntary control over neural signals, keeping the user engaged, independently of its level, since a highly motivating feedback improves performance (Garris et al., 2002). A game can be a beneficial to this since it can maintain motivation over multiple sessions and traditional feedback screens are usually monotonous. However, highly complex games can distract, interfere with the learning outcome or even introduce stress (Ninaus et al., 2013), so a simple solution can have better results, and the trainee must always feel successful (Demos, 2005). For the equipment chosen, EEG is more affordable, portable and easier to use compared with other technologies, making it the one of the most popular in the field (Ninaus et al., 2013).

Considering these points, it is possible to formulate a possible solution: a simple, 2D serious game with dynamic difficulty adjustment (DDA) applied to the current neurofeedback techniques.

The neuroheadset is applied to the player's head, and communicates through Wi-Fi with the connected computer USB equipment. The following figure represents the hardware structure of this project.

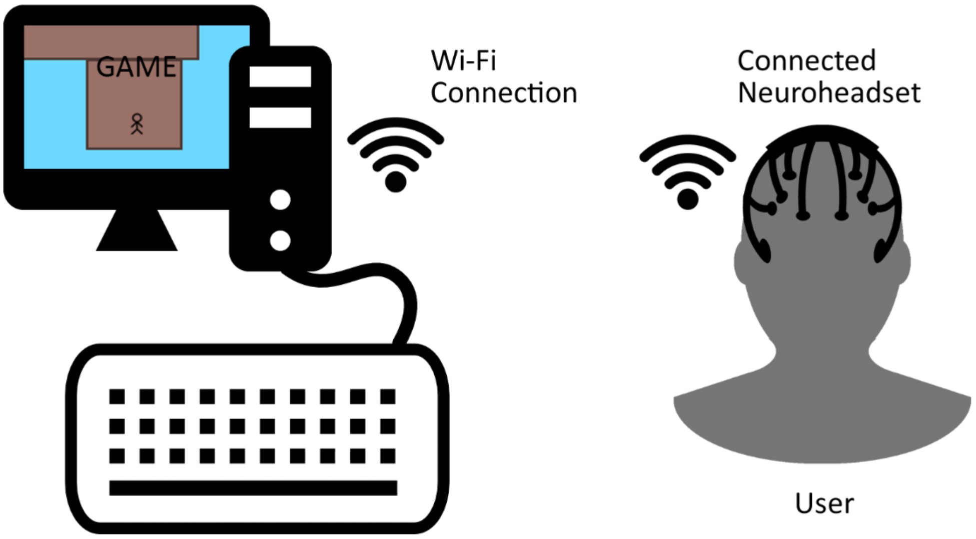


Figure 2 - Hardware Structure.

Inside the game, the headset SDK will be used to get real-time input of the current neural activity of the player. This way, the project software consists of the following structure:

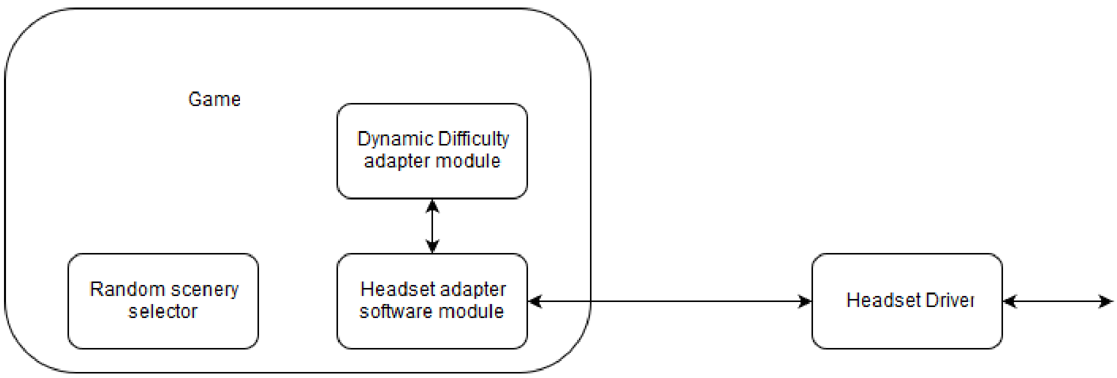


Figure 3 - Project's multiple features communication.

The game should be simple in order to avoid distracting the user, which could degrade the results. It will be necessary to have DDA to give accurate feedback on the player performance, as well as keep the difficulty just above the user level, so that it becomes challenging and not frustrating, increasing the user's motivation (this will be detailed in chapter 2.1).

This difficulty adjustment will be based on the input from the neuroheadset. It reads data and, for chosen neural frequencies, adjusts the threshold windows where we want to reward the user. This means that if the user increases or decreases, inside some specific frequencies, the expected channel's amplitude, he will be rewarded with the ability to damage the enemy. If the player maintains this state for some moments, the difficulty of the game increases, which will set the threshold window a bit smaller, making the game a bit harder. This way, whenever the user is able to maintain this state, he will defeat the enemy and can move to the next room where a harder enemy will appear. To avoid stress or disappointment, the player never loses, but can only advance if the desired conditions are met. When the player is not having the expected neural activity for some time, the difficulty is then reduced a little, by setting the threshold limit window to another value, motivating the user to go forward and keep trying.

The duration of the game can vary but will be set to be close to the duration of a neurofeedback treatment session. The game will have procedural map and theme generation capabilities each time the user starts the game, which includes different themes and different room structures. With this, the game is intended to be as non-repetitive as possible.

1.5 Project Evaluation

This project's assessment will help determine the quality and efficacy of the prototype developed. It will be evaluated if the game works properly with the neuroheadset integration, and uses the neurofeedback techniques having at least the same efficacy as other applications while at the same time, being more entertaining to the users.

This work will be assessed in two different scopes: software quality and software efficacy.

The software efficacy will be evaluated according to the prototype's ability to actually help reducing the disorder. A pilot test will be run on a small group of people, analysing the subject's evolution through the experience. To do this, the application's logging ability will allow the comparison of the headset readings in the beginning of the session with the results by the end of the session, verifying the progress of the brain activity during the experiment.

To assess the quality of the software produced, a Quantitative Evaluation Framework (QEF) (Escudeiro and Bidarra, 2008) will be applied and used. The objects of evaluation will be defined and set considering the multiple aspects and characteristics of the game, getting an overall view of this project's prototype condition and performance.

As a support method to these, a questionnaire will also accompany the tests. The purpose is to get the overall subject's interest in this concept and satisfaction during the experience, as well as verifying the game quality, if defects or other problems were found, allowing to assess the project's feasibility.

1.6 Document Structure

This document was defined and organized to better give the reader an insight about the project. It was taken into account previous works as well suggestions from the people involved in it. The document is organized as the following map shows:

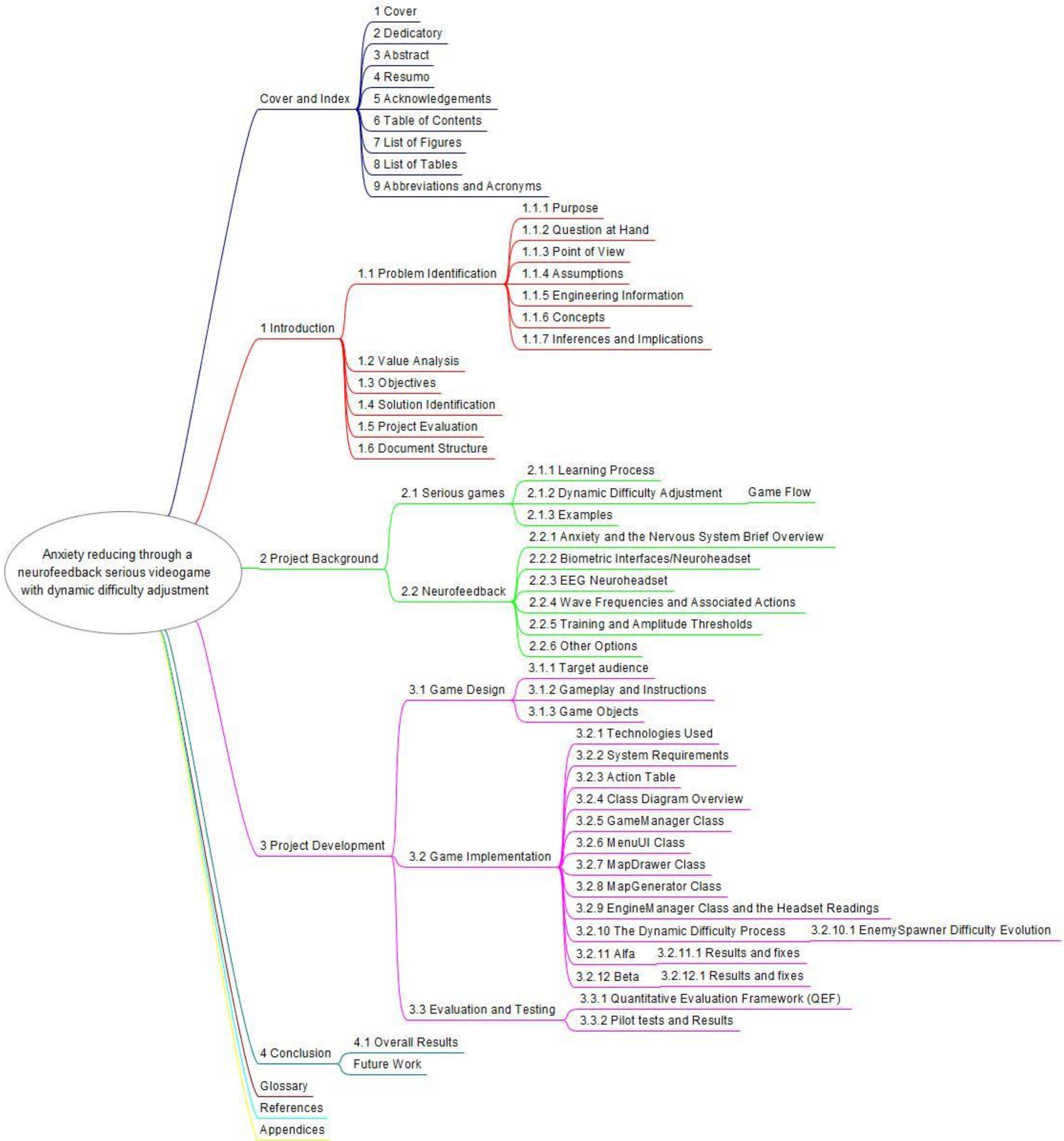


Figure 4 - Document Structure Mindmap

As it’s possible to see, the document is mainly split into introduction, project background, project development and conclusion. The introduction and conclusion chapters are self-explanatory. In the first one, where this chapter is included, the identified problem is presented, the goals of this study, development and the possible solution. The project background chapter will show the studied subjects and all the information thought relevant for this projects goals to be met. It will detail the knowledge got through the study of the two main topics of this work: serious games and neurofeedback.

The third chapter, project development, as the name implies, presents the implemented solution, the main aspects of the development, with special emphasis on specific features' behaviour, like the case of the dynamic difficulty adjustment implemented and the procedural map generation that this project will include. At the end of this chapter, it will be shown the evaluation of the work done as well as the results of the tests made, displaying how the experience was made as well as the results gotten from both the gameplay and the questionnaire supplied to the participants. The conclusion presents the overall results of this development and possible improvements to be made in the future.

2. Project Background

For this development to be possible, multiple technologies and methods were studied. Addressed by this project, we can identify two main categories: serious games and neurofeedback. This chapter serves to better understand what they are and what insights the previous studies on these subjects can bring to this development.

2.1 Serious Games

In recent years, computer gaming progressed steadily, becoming one of the most economically successful forms of human-computer interaction systems (Liu et al., 2009). Only in the United States, from 2009 to 2012, it was seen an annual growth of 9.7 percent (Siwek, 2014). Games have changed how we spend our free time becoming the most popular leisure activity. More recently, an interest in games designed for learning purposes and skill acquisition rose, the called serious games (Boyle, 2013).

Serious games are commonly defined as games designed for purposes other than pure entertainment (Djaouti et al., 2011). Albeit video games' initial purpose was to supply fun and leisure, it began to be clear that players were developing skills and gathering knowledge while playing. This way, there was an increase in interest over the past years, with specialists studying the use of a video game as an engaging new learning method.

As stated before, the primary purpose of serious games is not entertainment, enjoyment or fun (Michael and Chen, 2006). Education, defence, healthcare and others, are areas where games have been proved useful (Djaouti et al., 2010). Some studies show good results in serious games applied to education, since students using this system had significantly better scores (Blunt, 2008).

To establish a relation between which kind of games are most appropriate for supporting a specific kind of learning, multiple learning theories were studied.

2.1.1 Learning Process

The learning process has multiple skills involved, which can go from a simple recall of information to evaluating the learned material. A work developed by Bloom, Krathwohl and collaborators, found that "*most educational objectives could be placed in one of the three domains: cognitive, affective and psychomotor*" (Cullinane, 2009). The team also developed a taxonomy that has been used for decades as a framework for education and assessing achievements (Dettmer, 2005). This taxonomy allows the identification of different levels of expertise and in most literature, when referring to this taxonomy, typically refers to the cognitive domain (Cullinane, 2009). Its revised edition identifies six levels of the cognitive process:

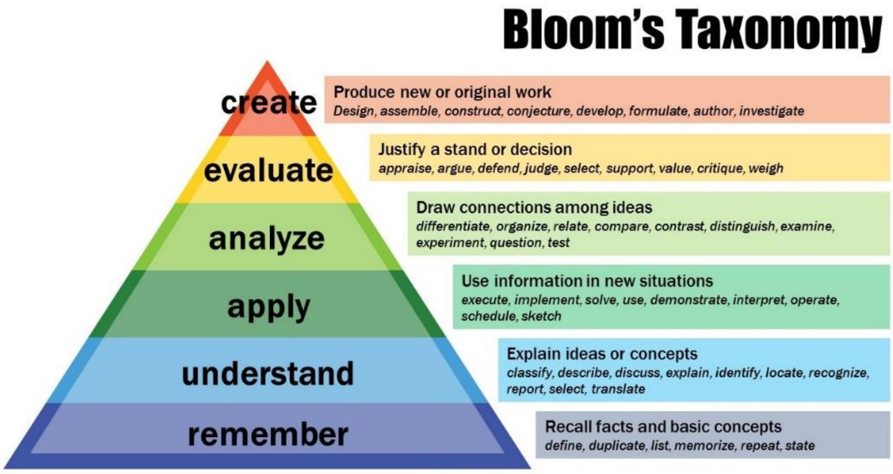


Figure 5 - Bloom's Taxonomy (Armstrong, 2015)

It is worth mentioning that this taxonomy is hierarchical, which means that it is only possible to reach higher levels when the objectives on the bottom have been fulfilled (Cullinane, 2009). This project, which is based on the cognitive domain, can roughly fit into the three base categories: recall and memorize the expected behaviour to repeat it, understanding and identifying how it happened and applying the knowledge to repeat it when desired.

There are some learning theories that have particular relevance to game learning. Behaviourism, Piagetian theory, Vygotsky's theory and Kolb's theory are some examples (Boyle, 2013).

One of the relevant theories for this project is Behaviourism, developed by Pavlov, Watson and Skinner, which is based on conditioning a fundamental mechanism for learning and, for neurofeedback games specifically, operant conditioning of brain activation. This theory is especially relevant for understanding how the reward system works in a serious game (Connolly et al., 2013a). In a behaviourist operant conditioning application, the user is presented with a challenge or question. If the pretended answer is given, the user is then rewarded.

Another relevant learning theory for this development is Vygotsky's, which stated that learners make the most progress when they are presented with tasks just beyond their current abilities

(Boyle, 2013). This will be clearer in the next chapter, where the dynamic difficulty adjustment (DDA) is presented and the concept of flow is detailed.

2.1.2 Dynamic Difficulty Adjustment

Traditionally, games have static difficulty levels set, and provide the player, at the beginning, a selection between "beginner", "medium" and "hard" difficulty. The difficulty is defined and then all the game unrolls based on the level selected. Despite this effort to keep the difficulty appropriate for the player's skill level, the experience and ability of each one is different, creating situations where the game becomes boring by being too easy, or frustrating by being too hard (Jennings-Teats et al., 2010). At the limit, the player eventually stops playing that game. Because of this, there has been an increasing interest in new methods for the games to get the difficulty adequate for the player (Missura, 2015).

DDA calculates the player's skill, based on their actions in-game, and makes real-time changes in the game variables, adjusting the elements to better match the player's current skill (Sutoyo et al., 2015), keeping the player inside the flow state, which will be explained in the next sub chapter.

The player performance in the game is evaluated and the system makes changes to game elements, making it harder or easier and getting close the adequate level of the player. This is particularly important for this development, since the game will adjust to the player's level, regarding the brain activity and reward when the desired conditions are met.

This way, by adjusting the difficulty of the game and keeping the player away from the boredom or the frustration, it is applied Vygotsky's learning theory insights, and maximized the player learning performance, by applying the concept of flow.

2.1.2.1 Game Flow

According to Csikszentmihalyi, we can define flow as "*the state in which people are so involved in an activity that nothing else seems to matter*" (Csikszentmihalyi, 1990). This definition is close to what previous studies stated as "*zone of proximal development*", in which the challenge is just outside the limits of the player's current skill level (Boyle, 2013).

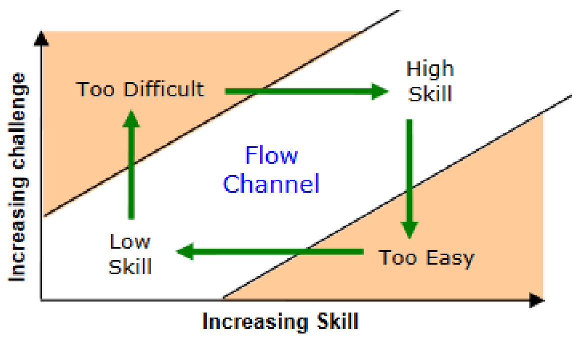


Figure 6 - Game Flow State (Hunicke and Chapman, 2014)

The point of this concept is to keep the player in a zone where a balance between challenge and ability lies. Furthermore, a feeling of control over the game activity is important as well (Csikszentmihalyi, 1990). To achieve this, and keep the player inside the "*flow*", the challenge must be adequate to his skills, and therefore, the previous described concept of DDA is applied. By interpreting and quantifying the player's current ability and skill, in this project's specific case, the neural activity readings (better detailed in chapter 2.2), it is possible to adjust to each individual abilities, and make changes in the game to keep the player in a challenging zone, neither excessively frustrating, nor easy. There are some known examples where dynamic difficult adjustment was used in videogames.

2.1.3 Examples

Crash Bandicoot

Figure 7 - Crash Bandicoot (Mulkerin et al., n.d.).

Crash Bandicoot, one very popular game for PlayStation, used a DDA system to help weaker players to be able to progress. They did this by making changes if the player died many times in the same place, like a boulder that moved a bit slower each time or setting a continue point closer to that place ("Making Crash Bandicoot – part 6," n.d.).

flOw

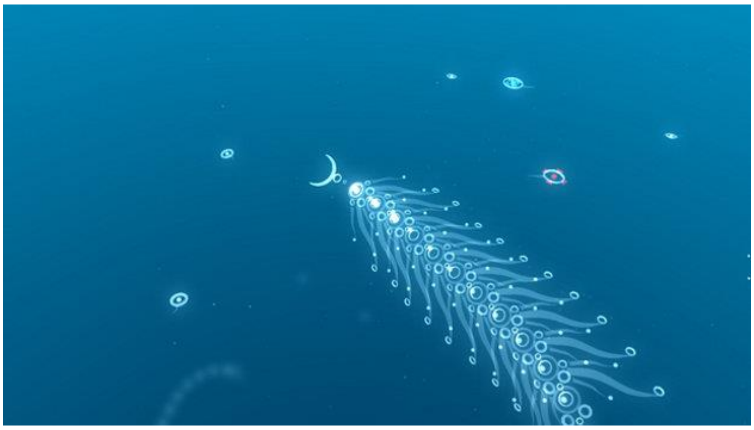


Figure 8 - flOw, award winning indie game ("flOw," n.d.).

Published in 2006, flOw's indie flash game creator stated that uses a dynamic difficulty adjustment system embedded in its own game design, allowing the player to choose when to go forward to a harder enemy or to go to a weaker enemy ("Joystiq interview," n.d.).

Madden NFL 09

Figure 9 - EA Sports Madden 09("Madden NFL 09," n.d.)

This also very popular game, Madden NFL 09 from EA Sports, has a system called Madden IQ, which works with an optional test at the beginning. Then, the game evaluates the player knowledge of the game and sets the difficulty accordingly ("Madden NFL 09 First Hands-On," n.d.).

2.2 Neurofeedback

Neurofeedback *"is a comprehensive system that promotes change at the cellular level of the brain and empowers the client to use his or her mind as a tool for healing."* (Demos, 2005).

2.2.1 Anxiety and the Nervous System Brief Overview

The nervous system is a complex network responsible for the communication between the brain and the various parts of the body ("Definition of Nervous System," n.d.). It is composed by the Central Nervous System (CNS) which consists of the brain and spinal cord, and the Peripheral Nervous System (PNS), made up of the somatic and autonomic nervous system (Mandal, 2010). The neuron is the base unit of communication within and between these parts, and this communication is achieved through a neurotransmission or synaptic transmission, which is an electrochemical activity, the movement of chemical or electrical signals across a synapse (Stufflebeam, 2008). The Electroencephalogram (EEG), is the graphical representation of the recorded electrical neurotransmission within the brain (Blinowska and Durka, 2006), and this representation is possible through the use of an EEG headset, an equipment capable of reading this electrical activity on the scalp, also commonly called neuroheadset (Emotiv, 2015). These EEG readings can be used to execute the neurofeedback training (Demos, 2005).

Historically, it is considered by many that Richard Caton's 1875 scientific research is the first key biofeedback event. Joseph Kamiya's experiment, in 1963, was also a success, where a volunteer was trained to recognize bursts of a specific brain wave. This was achieved by, each time the volunteer achieved the desired behaviour, verbally reinforcing it. This study demonstrated the typical biofeedback training loop: (a) Specific biologic activity is recorded; (b) Each time the desired state is achieved, trainee is rewarded; (c) Trainee becomes able to voluntary control a biological activity. All biofeedback activities are based on the process innovated by Kamiya (Demos, 2005).

Anxiety can be defined as one experienced emotion, a disquieting sensation that not everything is well and, because of this, it borders on the feeling of fear of a not well defined danger (Clemens, 2003). When facing a problem at work or before an important decision, it is common to feel some anxiety. But if the feeling remains, this can start to affect the behaviour or the performance, sometimes, even real physical symptoms (Crosta, 2015). An anxiety disorder has different types, and can present a variety of symptoms: restlessness, fatigue, irritability, muscle tension, sudden attacks of intense fear, even being afraid of social interaction, agoraphobia, etc. These symptoms can interfere with daily activities. (National Institute of Mental Health, 2015). Therefore, managing anxiety is an important task to everyone, and help might be necessary.

Through various years of study on this subject, it is now possible to map certain activities to specific brain regions. One great breakthrough in the area has been the early nineteenth century work of Korbinian Brodmann, where he was able to create a map, parcelling the cerebral cortex into "*microstructural and functional units"* (Zilles and Amunts, 2010).

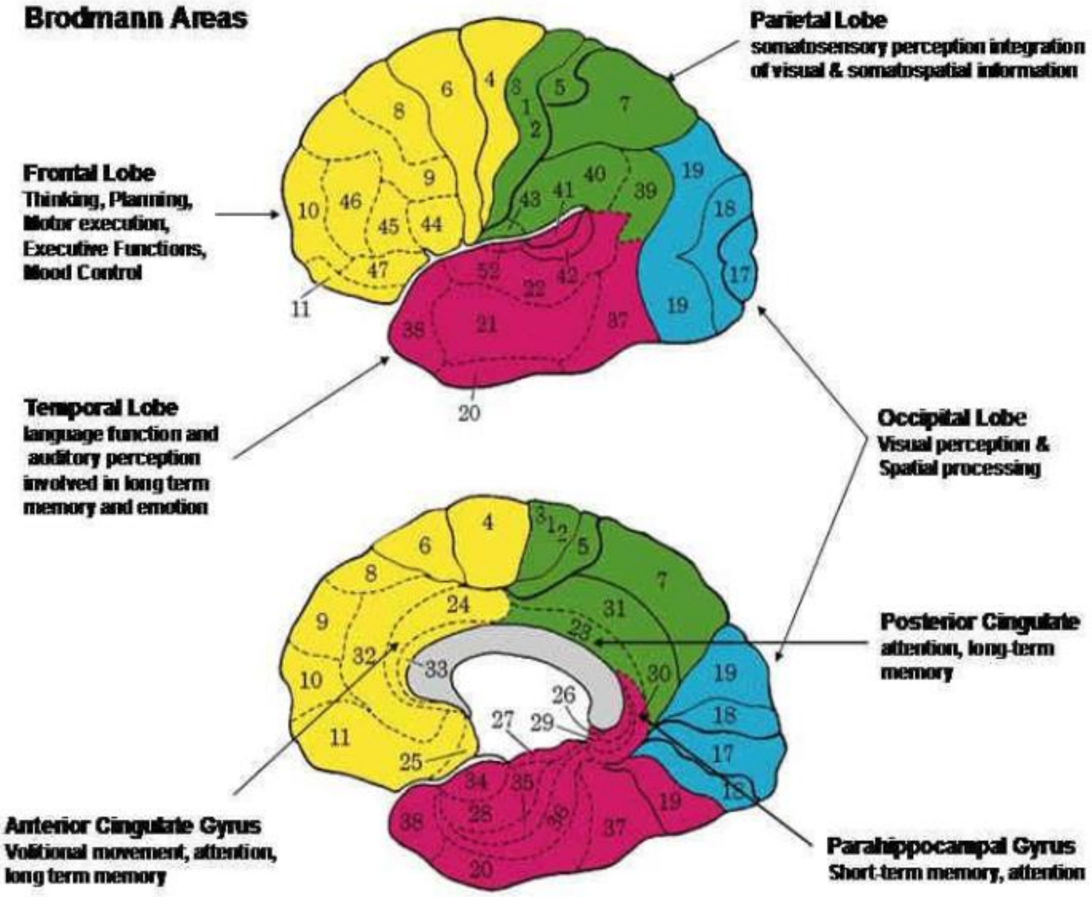


Figure 10 - Brodmann areas map example (Warner, 2013).

Neurofeedback is one type of biofeedback, which trains the brain by giving real-time audio/video feedback about its electrical scalp activity (Lofthouse et al., 2012). This way, the user receives direct feedback about its brain behaviour and can learn to gain voluntary control over neural signals (Connolly et al., 2013a). It focuses on the Central Nervous System instead of the typical biofeedback application, in which the target is the Peripheral Nervous System.

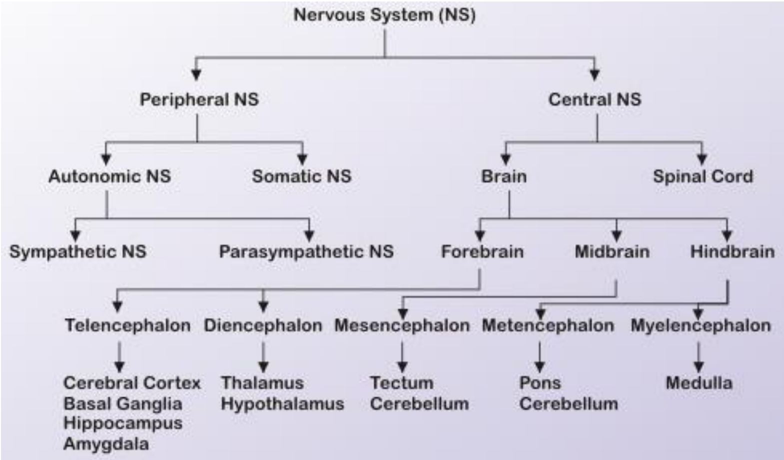


Figure 11 - Nervous system structure (Mandal, 2010).

This method has shown positive results in the treatment of multiple issues such as depression, anxiety, addictions, attention deficit/hyperactivity disorder, obsessive-compulsive disorder and others. Business executives, athletes, performers and many more can also benefit from peak performance training (Pop-Jordanova and Demerdzieva, 2010).

As stated before, neurofeedback is based on neural activity data reading, using EEG for this purpose, and so a biometric interface is necessary.

2.2.2 Biometric Interfaces/Neuroheadset

"Biometrics – the process by which a person's unique physical and other traits are detected and recorded by an electronic device or system" ("Biometrics Definition," 2015). They are unique measurable physiological, behavioural or anatomic characteristics. In the beginning of the nineteenth century, researchers studied the relationship between physical features and criminal tendencies. The fingerprint is a good example of a common biometric characteristic (Prabhakar et al., 2003).

A biometric interface is the equipment necessary to read a biometric characteristic. Currently, there are multiple types and purposes where they are being used. Authentication, for example, can use fingerprinting, hand geometry, voice verification, signature verification, retinal scanning, iris scanning and facial recognition. One other familiar use is the fitness bands, capable of reading heart rate.

A neuroheadset is, likewise, a biometric equipment, a brain-computer interface (BCI) capable of reading electrical signals from the brain.

2.2.3 EEG Neuroheadset

Some of the most frequent neural activity reading methods are electroencephalograph (EEG), Magnet Resonance Imaging (MRI) and Near-Infrared spectroscopy (NIRS) (Ninaus et al., 2013).

EEG was chosen for this project, due to its portability, cost and temporal resolution (Wang et al., 2011). It has electrodes positioned at specific locations to measure the electrical activity on the scalp.

For this project, the equipment is a 14 channel (plus 2 references) EEG headset called EPOC, from the Emotiv Company. According to the international 10-20 system (Oostenveld and Praamstra, 2001), a system providing letters and numbers to identify brain locations, the channels of this equipment are positioned at: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4 and P3/P4 for the CMS/DRL references (Emotiv, 2015).

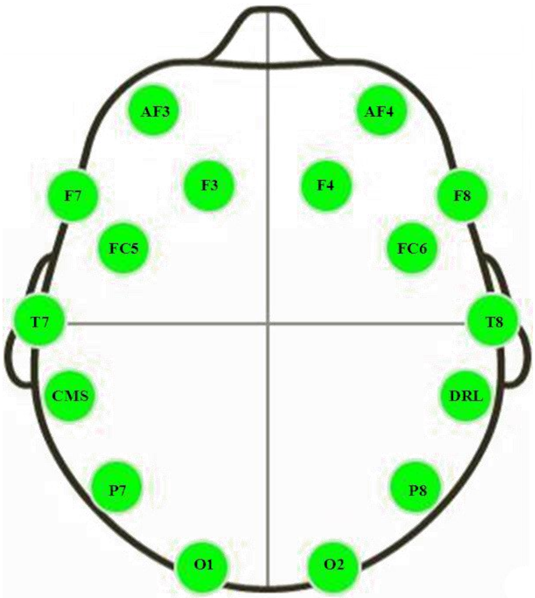


Figure 12 - EEG Electrode Positioning (Emotiv, 2015)

The other equipment looked into, PLX XWave (“PLX XWave - Brainwave to iPhone Interface,” n.d.) lacked functionality for this projects development, and MyndBand EEG Brainwave headset (“MyndPlay Ltd,” n.d.) seemed a good option but only had one sensor. Other equipment were also considered but lacked some of the pretended functionalities or accessibility.

About the selected headset, its specifications include, as stated, 14 channels with noise cancellation configuration, proprietary wireless 2.4GHz band, a signal resolution bandwidth between 0.2 to 43 Hz (Emotiv, 2015), enough to capture the most common frequency bandwidths: from 1 to 42Hz (Demos, 2005).

These frequency capture is enough to capture the most frequently emitted by the brain, and the ones used for the neurofeedback training.

2.2.4 Wave Frequencies and Associated Actions

Previous studies allowed associating characteristics with wave frequency. It is possible to identify the following bandwidths:

Table 3 - Common bandwidth frequencies (Demos, 2005) (“Understanding Frequencies,” n.d.).

Common Bandwidth Name	Frequency Range(Hz)	General Description of Characteristics
Delta	1-4	Sleep, repair, complex problem solving
Theta	4-8	Creativity, insight, deep states
Alpha	8-12	Alertness and peacefulness, readiness, meditation
Beta	13-21	Thinking, focusing, sustained attention
SMR	12-15	Mental alertness, physical relaxation
High Beta	20-32	Intensity, hyper alertness, anxiety
Gamma	38-42	Cognitive processing, learning

For this project’s objectives to be met, and according to the bibliography studied, the most relevant bandwidths were identified: alpha and beta. For these bandwidths, the selected equipment makes a distinction between the frequencies: alpha= 8-12Hz; lowbeta=12-16Hz; highbeta=16-25Hz.

Characteristics associated with alpha bandwidth are usually inner calm or peacefulness. For an anxiety reducing treatment, alpha can be downtrained (inhibited) in the anterior (frontal) region of the brain, and uptrained (rewarded) in the posterior region, only if high amplitudes are not present. Together with this, it might be relevant to look into the beta readings. Anxiety disorders frequently show that an asymmetry in the right hemisphere is indicative of anxiety (Warner, 2013). Because of that downtraining beta in the anterior right hemisphere, might help with this issue. High beta and theta frequencies are also associated with anxiety, it might be important to downtrain them as well, in order to help with the issue (Demos, 2005).

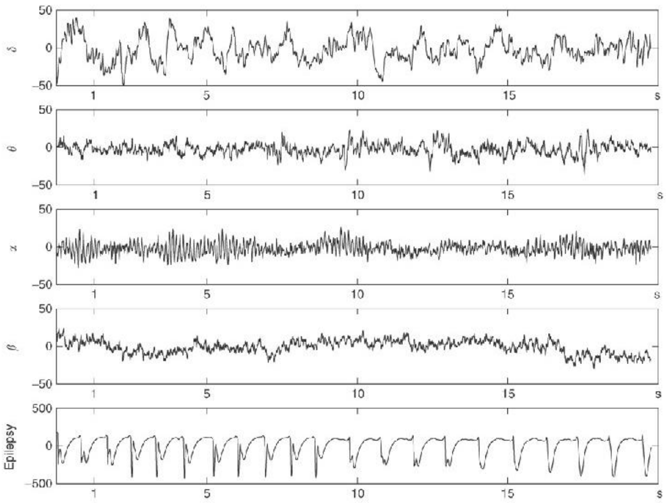


Figure 13 - “*Characteristic EEG rhythms, from the top: δ (0.5–4 Hz), ϑ (4–8 Hz), α (8–13 Hz), β (13–30 Hz). The lowest trace–EEG during epileptic seizure, note that the amplitude scale is an order of magnitude bigger.*” (Blinowska and Durka, 2006)

2.2.5 Training and Amplitude Thresholds

It is important to state that all biofeedback applications should reward and never punish, being very important to give a positive reinforcement to the trainee, each time he is on the right path to the expected behaviour (Demos, 2005). If the trainee's performance is not reinforced at appropriate time, the process might fail, and the user might not get any result in the learning process. It is also important to give an adequate challenge to the brain, so that it is possible to gradually improve, and for this, the previous described dynamic difficulty adjustment will be used.

The DDA will mostly work with threshold adjustment to reach the goal of the training. Since the amplitudes are what changes in neurofeedback training (Hammond, 2011), the point of the training is to teach the user to voluntarily control the brain activity's amplitude of specific frequencies.

A neurofeedback training usually consists of downtraining (inhibit) or uptraining (reward) a specific brain wave frequency amplitude. This training consists on rewarding the user, immediately when he has the pretended behaviour, and one or more thresholds are frequently used for this. The goal is to set limits in which the specific brain wave is pretended to be. But in the beginning the user probably will not know how to achieve this, and a trial-and-error learning will take place (Connolly et al., 2013a).

For the learning to happen, amplitude limits are set within a specific frequency, and, every time the user is able to reach the expected window, it receives positive feedback through visuals or sound (Demos, 2005).

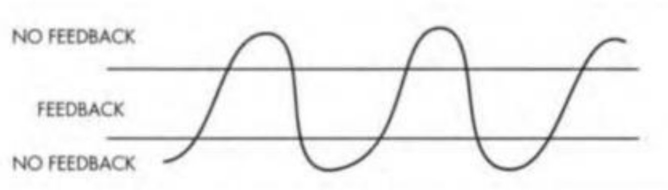


Figure 14 - Feedback Thresholds Bars (Demos, 2005)

In this project, it is addressed both alpha and beta frequencies and commonly, these frequencies amplitudes range from 20-100 µV for alpha and 5-20 µV for beta frequencies (Benbadis, 2015).

The visual reward of traditional graphical applications can become repetitive and lack the ability to keep the user motivated, therefore this development aims to test the possibility of joining a serious game concept with the neurofeedback treatment.

2.2.6 Other Options

Currently there are other applications that also apply the concept of neurofeedback game, like the case of Transparent Corporation's Mind Workstation ("Mind WorkStation," n.d.). This application has professional capabilities being considered a treatment tool and the visuals of the application fulfil the purpose with a fire to be set or a ball moving on the screen. Other currently available options, are the cases of NeuroPlus ("NeuroPlus Attention Training," n.d.) and BioExplorer ("BioExplorer," n.d.). It is hoped that this project can bring enjoyment as well as a relaxing procedure in the path to achieve the pretended results. Applying the concept of flow, as explained in the chapter 2.1.2, it is aimed to test the possibility of another option, if possible more accessible to the common user and with at least the same efficacy as other applications.

3. Project Development

The next chapter will present the development process of this application. It will cover the high level design and detail the features of the prototype construction.

3.1 Game Design

In this sub-chapter the game will be described, as well as its objects, instructions and other elements.

The game is a small, 2D adventure/action game. It is a 2D game with a top-down viewpoint, in which the player travels across randomly generated squared rooms battling the enemies that show up. It is set up in a fantasy environment and it tells the story of the main character. The ability to damage the enemy is directly connected to the headset readings of the player's brain activity, as it will be explained. The player has to beat one enemy in each room until all the rooms are cleared. Information about the current difficulty level, the remaining enemies to be defeated and the current enemy health are displayed in the HUD.

3.1.1 Target Audience

This game is aimed to all ages, from 10 to 65 years old, everybody can play this game assuming that they are healthy and don't suffer from any medical conditions. To attain results, the game should be played by anyone who is able to learn a new skill in a healthy way. The player might or might not be accompanied by a professional, and this together with the environment where the game is being played, might influence the final results.

The identified possible use cases of this project, are shown in the following figure:

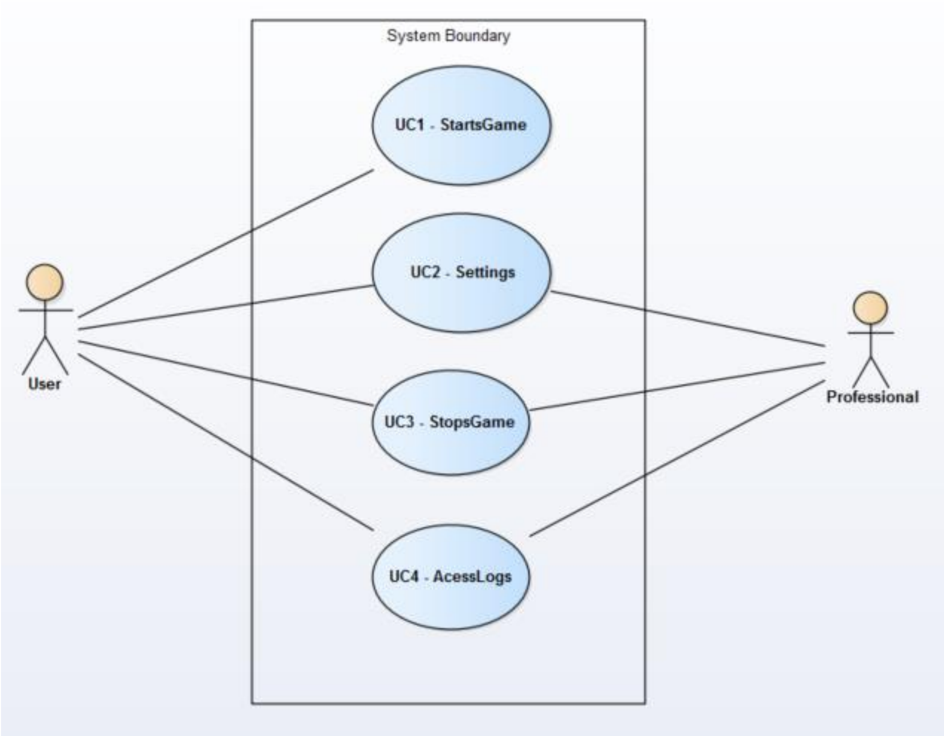


Figure 15 - Project's use cases.

The game is designed to be used in two environments: one where the player is alone using this application, and it is considered that he can access all the features, the second where a professional is accompanying the user, where the use is only able to play the game since the professional has to control the game execution and have other privileges.

3.1.2 Gameplay and Instructions

When the level starts an adjustment period occurs. The player reads a small short story of the main character and the instructions of the game, in which pressing the space key skips the messages. Then, a countdown of 20 seconds shows up and, during this period, the game reads the brain activity of the player, to later be able to properly adjust the difficulty to each person. The player uses the WASD or directional keys to control the movement and can roam around the rooms freely. After getting close to an enemy, the enemy starts chasing the player until the player is caught. At this point, a battle sequence starts. In this battle sequence, while the characters' battle animation plays, the player should focus and concentrate in getting the right mind-set, the right brain wave amplitudes. The application gets the current brain wave readings, makes the appropriate calculations and, if the player is having the pretended brain activity, will be rewarded. This is accomplished by allowing the player to hit the enemy reducing its health. If the player is able to maintain this state for a set period, the game then proceeds to increase the difficulty, which will be later on explained in detail.

Figure 16 - In-game fight sequence.

The rooms each have an enemy to defeat and the player can only damage the enemy when the desired behaviour is read. Direct feedback of its performance is supplied by watching the enemy's health progression. In the top left corner of the screen, it's possible to check the current difficulty level, as well as how many enemies are left to be beaten in order to reach the end. It is only possible to advance to the next room when the enemy in the current room is defeated. The enemy does no damage to the player, and it is possible to pause the game anytime by clicking with the mouse in the screen pause button, or by pressing "Esc" on the keyboard.

3.1.3 Game Objects

The game has a main character and various enemies.

Figure 17 - Main character and 1 enemy.

The player moves a character around a built architecture, where enemy characters exist. The game also has other objects but only as background and not relevant to the main objective.

3.2 Game Implementation

3.2.1 Technologies Used

For the game development, the key technologies used are Unity 5 with C#. As mentioned above, this choice was made taking into account the current and previous platform experience, the ease-of-use of this tool and programming language knowledge ("Unity - Game Engine," n.d.). The Unity graphical engine, considering the scope of this development, is simpler to learn and perfectly able to fulfil the objectives of this development, to achieve the desired result. The large community and information available also played a role in this selection, since it will help the developer through this process. Compared with Unreal Engine, which also has free access, no previous experience exists and with a steeper learning curve, more time would be necessary for this development. The fact that this game is intended to be kept simple and fun, the graphical engine of the Unreal Engine, one of its assets, is not very relevant, making Unity the best technology for these objectives ("Unreal Engine 4," n.d.). Similar technologies like CryEngine or Construct 2 were also looked into, but with identical results.

As for the integrated development environment (IDE), Visual Studio Community 2015 was used. This tool was opted instead of the default Unity 5 MonoDevelop, since there is a greater experience working with it, which gave an advantage over the other tools available. Both support C# development, but Visual Studio's debug features and its integration with version control systems were also important factors. It also has a plugin to better integrate with Unity 5 to help making the development smoother ("Visual Studio," 2016).

The engineering diagrams were made using a trial version of Sparkx Systems tool, Enterprise Architect. This software allows UML modelling as well as many other features. The class diagram, use case diagram and sequence diagrams that are present in this document were all made with this tool ("Enterprise Architect UML modelling tool," n.d.).

As explained previously in chapter 2.2, the neuroheadset chosen for this development was the Emotiv EPOC. The headset equipment provides a software development kit (SDK), which is used in this project to communicate with the headset. It provides methods and tools to access the headset capabilities, and in this way, access the necessary data to develop the proper behaviour behind the game. During this development, it was also necessary to use of the Emotiv Control Panel, to simulate the presence of a headset, while the equipment was not physically available, making it possible for the development to continue (Emotiv, 2015).

There are other technologies that are used in this development. One of them is the version control system, Git. This was not strictly necessary, but it was used during this development to easily back up the progress, and for the tools allowing to easily change a large part of the current development ("GitLab," n.d.).

As for the graphical design, the Image editor used was mostly GIMP, in order to draw the models and edit the sprites during development ("GIMP," n.d.). It has all the necessary tools to reach the goal, and it is open source, making it easily available. The sprites used in-game were created with the help of a free web tool, Character Generator ("Universal LPC Sprite Sheet Character Generator," n.d.).

3.2.2 System Requirements

For this game to run properly, a computer, with the common mouse and keyboard peripherals connected, and an Emotiv EPOC headset are needed. The headset must be properly connected and installed.

Table 4 - Minimum hardware requirements.

Minimum requirements	
OS	Windows 8.1
GPU	DX9 (shader model 3.0) or DX11 with feature level 9.3 capabilities
Memory	1 GB
Disk space	500 MB

It is also necessary to have a Microsoft Windows operating system installed. Since this is a prototype, it was only tested on Windows 8.1, so it is not guaranteed that it will work in other systems.

3.2.3 Unity *GameObjects* and *MonoBehaviour* methods

The Unity development platform works mostly with *GameObjects*, components and scripts. "*GameObjects are the fundamental objects in Unity that represent characters, props and scenery. They do not accomplish much in themselves but they act as containers for Components, which implement the real functionality.*" ("Unity - Game Engine," n.d.).

A script can be considered as a behaviour component, which can be attached to objects to execute specific behaviours. Other components include *rigidbodies*, *colliders* and even audio sources. These components are attached to some object in the scene, so that they are executed when the scene plays. Unity scripts, which are also *components*, derive from a class named *MonoBehaviour*, native to the Unity game engine system. A script is, in fact, a C# class, which allows the use of Unity's engine methods such as *Start* and *Update*. Here is a brief overview of the methods used throughout this particular game:

Start - This method is called when a script is enabled and is executed only once in the lifetime of the script. It is called on the first frame, just before the *Update* methods are called.

Update/ FixedUpdate – The *Update* method is called every frame and is where almost all the game behaviour is programmed. *FixedUpdate* is very similar, except that it runs a fixed framerate frame, and because of this, is commonly more used for behaviours that include a *rigidbody*, or some kind of physics.

Awake – *Awake* is executed before the *Start* method, when the script is being loaded, and not in the first frame like what happens with *Start*.

OnDestroy - As the name suggests this method is called when the object is going to be destroyed.

OnCollisonEnter – There are other methods that, like this one, are directly related with collision detection. This case in particular, is used when an object with a collider component, or a rigidbody, touches another collider/rigidbody.

These are the main methods, derived from the *MonoBehaviour* class, used during this development. The information here present was taken from Unity' documentation ("Unity - Scripting API," n.d.).

The following figure shows level 1's Unity *GameObjects* hierarchy present in the scene, followed by a succinct explanation of their use:

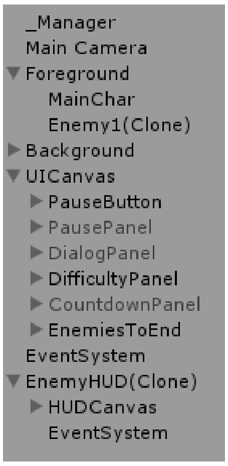


Figure 18 - Unity *GameObjects* hierarchy.

_Manager: This *GameObject* is crucial to the level. It is the object on which the *GameManager* and *MenuUI* classes are attached. There is no visual representation of this object in the scene.

Main Camera: The main camera with the script *SmoothCamera2D* attached, which allows the camera to follow the player movement smoothly.

Background: The parent object of the architecture of the level. Inside this object are placed the instantiated walls, floor and all the architectural elements of the game.

Foreground: The container of the player character and the various spawned enemies.

UICanvas: The canvas where the various heads-up display (HUD) elements are drawn: pause button and panel, dialogs, difficulty and remaining enemies' indicator.

MainChar: As the name implies this is the main character object. Attached to this object are a sprite and animation renderer, a *rigidbody* and a collider component. This object has also some behaviour that needs processing and, therefore, there are also some script components attached: *CharMovement2D*, responsible for the player movement and *CharStatsnSettings*, where stats like speed and attack power are stored. This will be better detailed further.

Enemy#: like the previous one, this is the enemy object. Since there are multiple enemies for each gameplay, this represents the instantiation of the current enemy. There is only one at a time in a room. This object has the *EnemyMovement2D* script attached, to move in the player direction and, likewise, *EnemyStatsnSettings* script, to control variables like speed and health points. In the hierarchy there is also an *EnemyHUD GameObject* which is the visual representation of the current enemy health. This will, like the enemy object itself, be instantiated by a script, which is why it has the "(clone)" indication, visible in the previous figure.

3.2.4 Action Table

To help in the development of this project, an action table was made:

Table 5 - Action table.

ID	Level	TRIGGER	OBJECT	ACTION	RESULT
1	1.1	Player	Enemy	Player touches enemy	Player and enemy stop moving. 1.2 starts
2	1.1	Enemy	Enemy	Enemy loses all HP	If not last room, a new enemy is spawned in the next room
3	1.1	Enemy	Enemy	Enemy loses all HP	Enemy dying animation plays and object disappears. 1.1 resumes.
4	1.1	Player	Enemy	Player gets near area around enemy	Enemy starts moving towards player.
5	1.2	Player	Timer	Player gets into the intended bandwidth	Timer to increase difficulty stops and timer to decrease difficulty starts.
6	1.2	Player	Timer	Player starts failing to get into the intended brain activity	Timer to increase difficulty stops and timer to decrease difficulty starts.
7	1.2	Timer	Difficulty	Timer to increase/decrease difficulty is reached	Difficulty level increases/decreases.
8	1.2	Difficulty	Enemy	Difficulty increases	Next enemy's health points will be higher and player will have more attack power.
9	1.2	Difficulty	Enemy	Difficulty decreases	Next enemy's health points will be lower and player will have less attack power.
10	1.2	Player	Enemy	All enemies are defeated	A message is shown indicating the player reached the end.

This action table is meant to showcase the actions that exist in the game, separated by 1.1 and 1.2 levels. This split means that although there is only one level present, it is considered to have two different moments: when the player is free to roam around the room, and when the player is locked in battle, 1.1 and 1.2 respectively. This table shows that, for example, when the player touches the enemy, the player stops moving and isn't allowed to roam anymore, starting the 1.2 sequence which will only be finished when the 3rd action is triggered. When the enemy loses all the health points, the level returns to the 1.1 sequence.

3.2.5 Class Diagram Overview

An overview of the final UML class diagram of the software can be seen below.

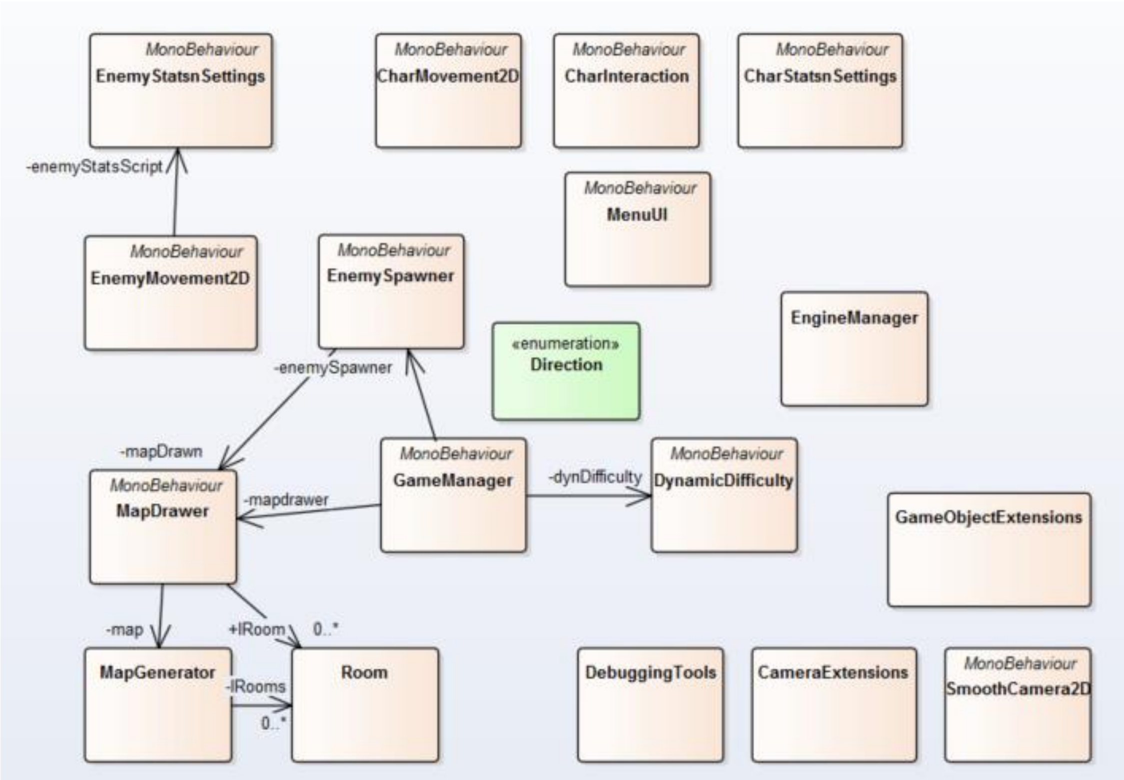


Figure 19 - Class diagram overview.

Some of these classes define specific behaviour that needs additional detail, explained in the following sub-chapters.

3.2.6 *GameManager* Class

One of the main classes of this game is named *GameManager*. This class is attached to the _*Manager GameObject*, having no visual representation in the scene. It is responsible for some of the main game features and game start flow control. This class file is where it is defined the enum *Direction*, used through all the application, to have a definition of “Up”, “Down”, “Left” and “Right”. The *GameManager* class is also where the log files of the application are defined and set.

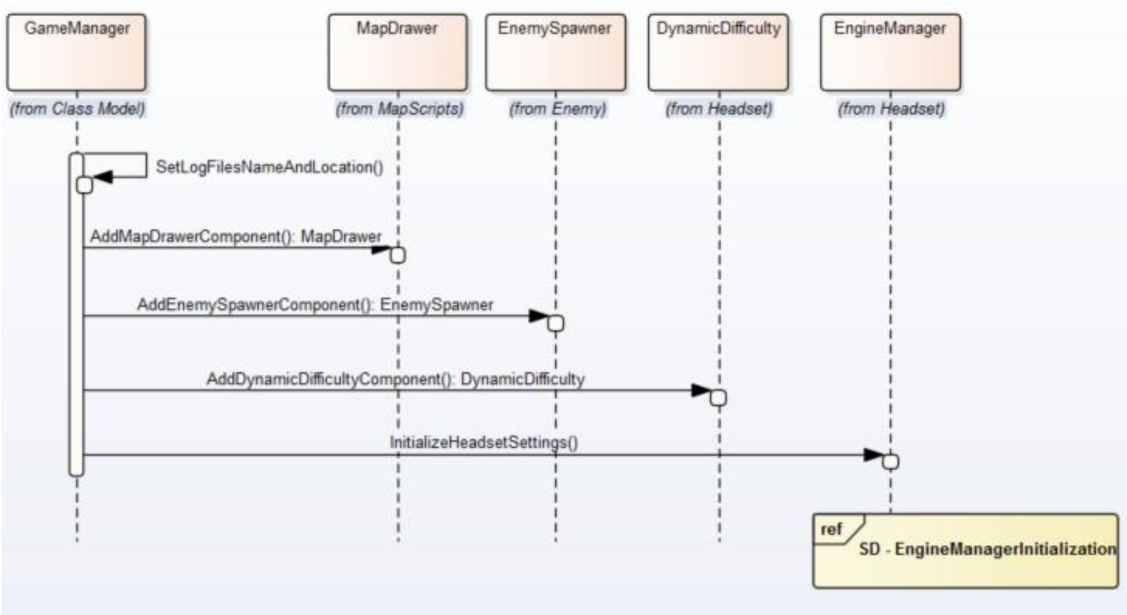


Figure 20 - Sequence Diagram: GameManager Start

GameManager, after setting the log files name and location, makes the necessary calls to instantiate *MapDrawer*, *EnemySpawner* and *DynamicDifficulty* classes. After this, it is set the environment to allow the start of the neuroheadset readings.

When the *Update* method of this class runs for the first time, it is executed a call to the dialog showing static method, present in the *MenuUI* class, which will show the brief introduction to the player.

3.2.7 *MenuUI* Class

This is the class responsible for the UI elements of the game. It is composed mainly of static methods, which are used by other classes, and controls for example the dialog showing and the information about the current difficulty and enemies remaining.

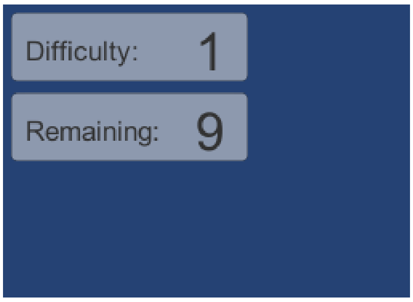


Figure 21 - Information panels.

It uses the *Awake* method to allow finding the UI panels in the hierarchy before anything else, so that when another class needs to make a call on these UI elements, these are already available to be used and edited.

Figure 22 - Dialog example.

During this class's execution, the *Update* method is used to control the dialog showing, presenting a new message or removing if the specified key was pressed. This method is also responsible for toggling the game pause state, controlling this action by the press of the screen corner button, or the defined "Esc" key.

3.2.8 *MapDrawer* Class

Instantiated by the *GameManager* script, the *MapDrawer* class is the entry point to the map's architecture generation and is responsible for the actual drawing of the objects in the scene. The first goal of this class is to properly load the necessary assets from the disk, storing them in *GameObjects'* variables to be used later. These assets are randomly chosen, between three different themes for the game floor, and another three randomly picked wall themes. Then a new *MapGenerator* object is created*,* returning a list of *Room* objects that will be used to instantiate the elements in the correct positions and sizes afterwards. To better understand this, the *Room* class has to be explained first. This class is mostly an information storage class, where the attributes of a room are set. Width, height, origin, entry door position, exit door position are established in this class. The class's constructor is in charge of randomly generating the room width and height, within set maximum and minimum limits, setting the room default centre point, origin position and the default entry and exit door positions.

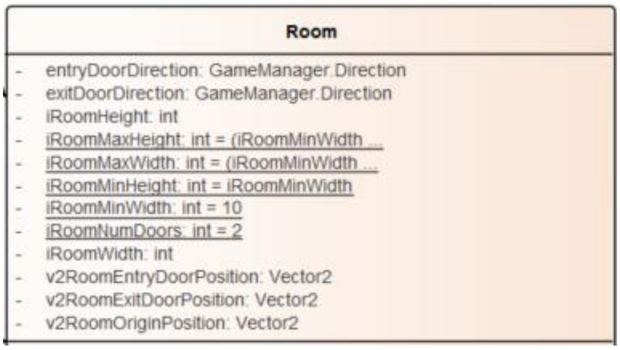


Figure 23 - Room class attributes.

As stated before, the *MapGenerator* constructor will return a list of *Room* objects, which will allow to correctly draw the architecture. The *MapDrawer*'s *InstantiateMapArchitecture* method loops through all the rooms present in the list returned and, for each room, instantiates the floor, the walls' blocks and the doors, putting the new instantiated scene objects into the correct hierarchy parent, the *Background*.

3.2.9 *MapGenerator* Class

For a better understanding of the previously described *MapDrawer* class's process, the *MapGenerator* class should be explained and perceived. As mentioned before, this game's map is randomly generated at the start of the game, so that each time the user plays the game, the map has some differences in the room sizes and room placement. The different sizes are a product of the previously shown *Room* class constructor, but the *MapGenerator* class sets the remaining attributes of the room.

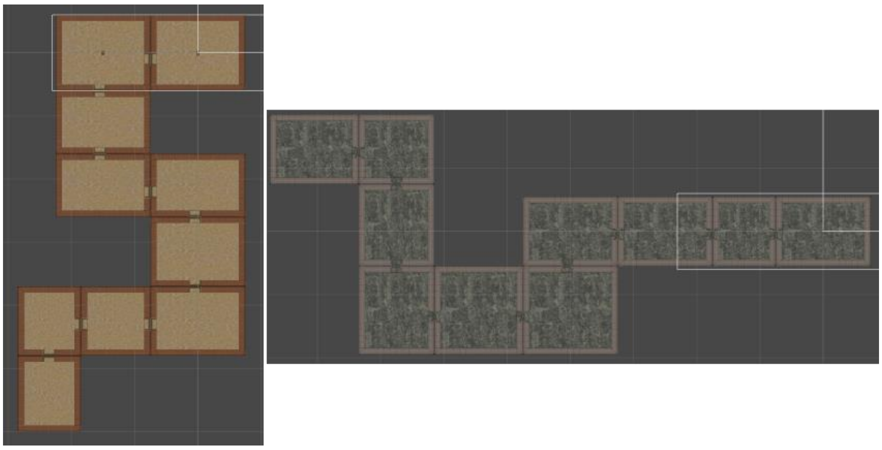


Figure 24 - Randomly generated maps example.

The actual rooms' map positioning is based on a simple 2 dimensional array, called *arrayWorldMap*, which allows keeping track of where the current rooms are located and where the new rooms can be placed. This constructor starts by creating a new, default *Room*, and setting its position in the centre point of the *arrayWorldMap*. This is our starting position for the map generation.

Then, the method *GetRandomExitDoorPosition* is called, passing the *arrayWorldMap* and current array positions as parameters. The method is the one responsible for checking inside *arrayWorldMap*, where the free positions for the next room to be placed are. It checks the adjacent array positions and the neighbouring 2 positions, to avoid getting into dead ends as can be seen in the next figure:

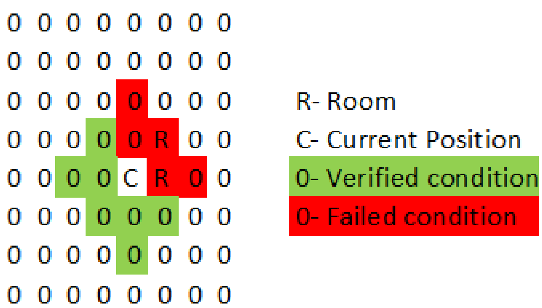


Figure 25 - Exit door verification example.

At this point, the *RoomExitDoorDirection*, will define where the next *Room* will be placed.

The next step is to get the attribute values of the *Room* object created, in order to define an origin, an entry and exit door position values, and ending by adding the *Room* object into a list that is returned. This closes the generation of the first *Room*, since no additional computation is necessary.

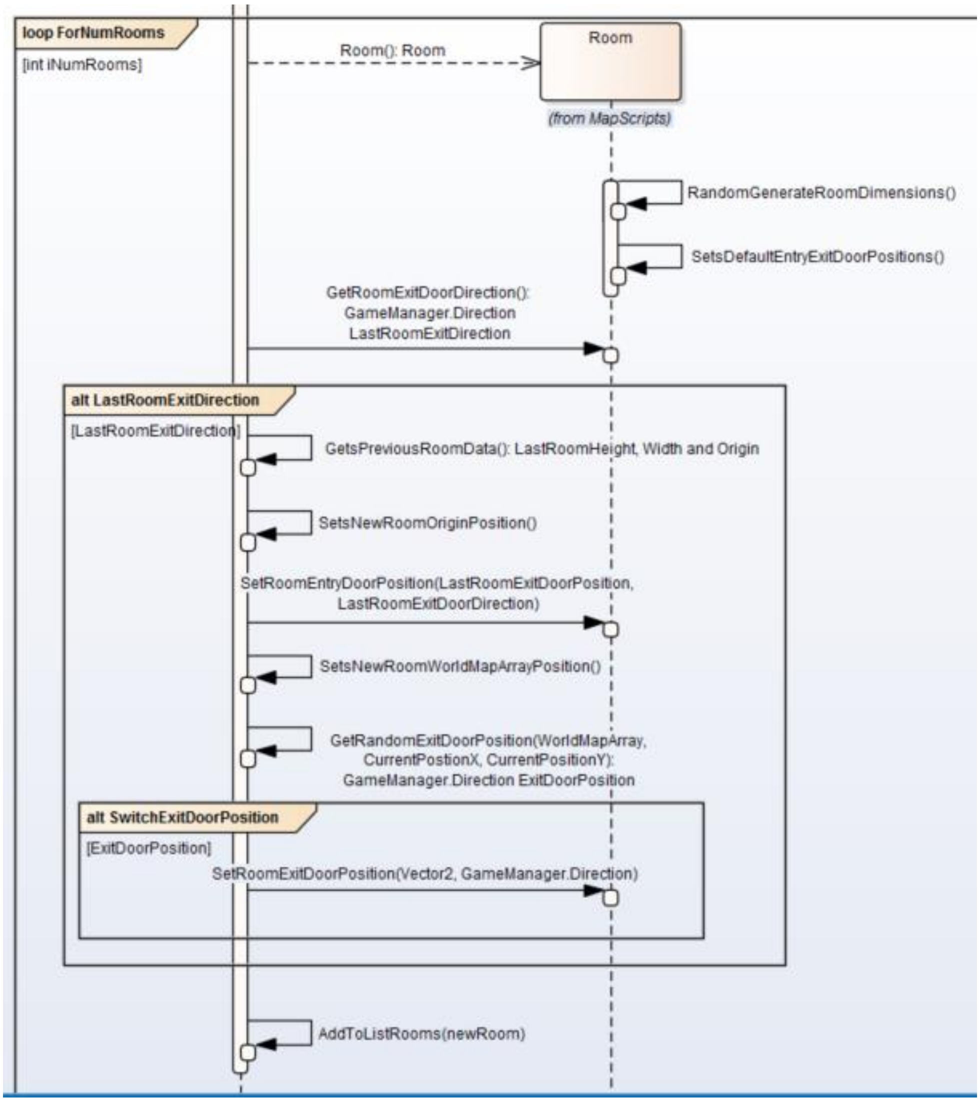


Figure 26 - Partial sequence diagram of *MapDrawer*.

The remaining *Room* objects are generated with a very similar procedure. As can be seen in the sequence diagram above, with the main difference between first *Room* and the following, is the verification of the previously placed *Room* exit door position, setting the new object origin and entry door position correctly. Finally, the *Room* is stored into a *Room* list that is, as mentioned in the previous chapter, returned to the *MapDrawer* for a correct element placement in the scene. This list will contain all the generated rooms and their respective defined attributes: the origin point, exit and entry door positions as well as the width and height of the room, all the necessary information for the *MapDrawer* to instantiate the game map architecture.

3.2.10 *EngineManager* Class and the Headset Readings

The *EngineManager* class, unlike most of the other classes present in this project, is not a derivation of Unity's *MonoBehaviour*. Due to this reason, its process start is controlled in a different way, being solely managed by other classes' interaction, and not by Unity's event functions *Start*, *Update* or other. This class is the largest of the project, with multiple attributes and methods of which the most relevant will now be detailed.

First, a method named *InitializeHeadsetSettings*, invoked by the *Start* method of *GameManager,* sets some necessary values to the class variables that will afterwards be used for the headset interaction. The creation of a new instance of *EmoEngine*, the interface with the Emotiv EPOC SDK dll, is done at this moment, allowing later access to the headset functionality. The equipment's event listeners connected/disconnected are also defined during this process, finishing the actual connection to the headset. All this process is done using the equipment's library provided functions, which allow access to the headset functionality. At this point, the neuroheadset is ready to be used by the game.

On the first execution of the *DynamicDifficulty* class, a call to the *EngineManager StartReadingsProcedure* is done. This process starts by designating the headset channels that are going to be used and starts a new thread. The whole channel reading process is done in a parallel thread, allowing the game engine to have more free resources to process the graphics and dynamic difficulty calculations, while the *EngineManager* class is making calculations using the headset returned values. As stated on chapter 2.2, this application will opt for uptraining alpha bandwidth in posterior regions, downtraining alpha in anterior area. As for the beta readings, split by the Emotiv SDK methods into low beta and high beta, is downtrained in the right hemisphere anterior region, trying this way to get the expected results, as supported by the studied bibliography.

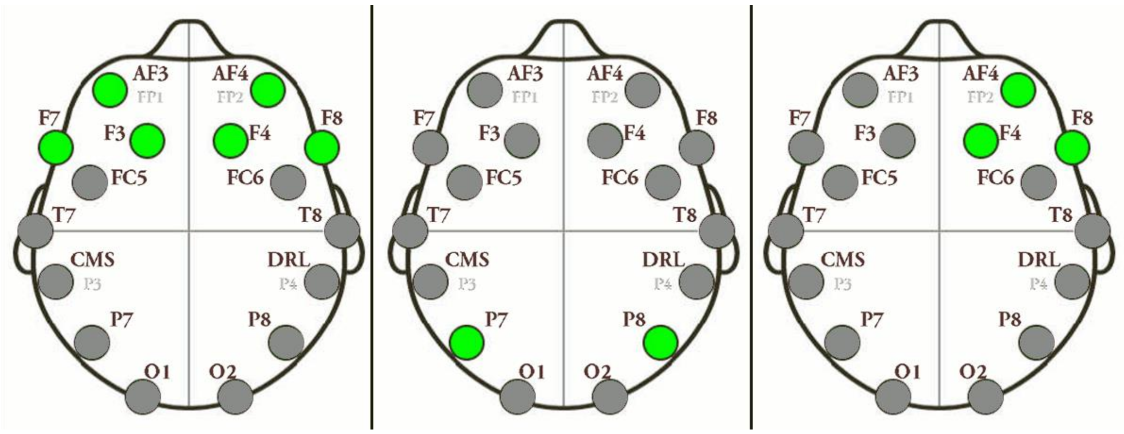


Figure 27 - Application's action regions: alpha downtrain | alpha uptrain | beta Downtrain.

The first method performed by the new thread adjusts the limits of the bandwidth waves. This means that it gets and stores the headset readings, in periods of half a second, to later evaluate and calculate the difficulty values and bandwidth amplitude limits, process that will be further explained in the next chapter. For the duration of the adaptation period, currently set 20 seconds, every half-second the application gets the average values got, for the previously specified channels, using the *IEE_GetAverageBandPowers* function, part of the Emotiv SDK dll. According to the own function definition, this returns the values with 0.5 seconds step size and 2 seconds window size. These values are then stored into an array for each wave type, and these are used for the dynamic difficulty computations. The execution control of this cycle is done manually by the use of timers.

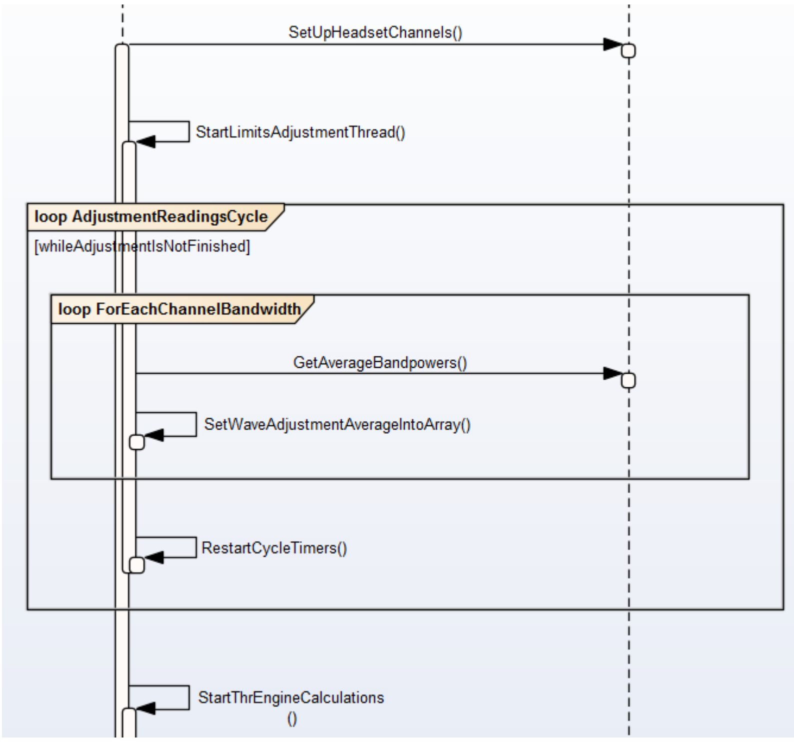


Figure 28 - Adaptation period sequence diagram of *EngineManager*.

When the specified adaptation period finishes, a Boolean flag indicates that it should now proceed to the readings method, *ThrEngineCalculations*. This method, which also runs in the same thread as the previous one, is very similar. It cycles every 0.5 seconds, getting the average values the headset channels registered, and storing them into the appropriate arrays. All of this process is done while logging these values to a text file, set in the *GameManager* class, to later have access to the values read and averages, allowing the analysis of the results.

It is also worth mentioning that since these methods work on a different thread, all the shared attributes between these and main thread methods are controlled with special attention to avoid race conditions or other thread issues, locking the variables before accessing them. When the game is stopping, either by closing or quitting, it is called *RequestThreadStop* that finishes the thread execution, disconnecting the *EmoEngine* instance.

The *EngineManager* class also has, aside from the common getters and setters functions to control attribute's access, the average and standard deviation methods, that are used from the main thread, to get the respective average or standard deviation values, of the registered readings. The dynamic difficulty process does this.

3.2.11 The Dynamic Difficulty Process

The *DynamicDifficulty* class, invoked by the *GameManager Start* method, is where all the difficulty management process is done. The *Start* process of this class only does some common actions: initializes the timer variables, the bandwidth limits and loads the sounds that are going to be played. When the *Update* method runs for the first time, a call is then made to the previously mentioned *EngineManager* method *StartReadingsProcedure*, which will begin the thread that will initialize the headset functions with the purpose of adapting the bandwidth limits. This will serve as "preparation" for the calculations that follow and it is executed only once, verifying if the thread is already running, avoiding multiple thread creation each time the *Update* method runs. Also in the *Update* method, it is checked if the adaptation period has finished, getting this information from the *EngineManager* class. When the game's adaptation countdown period finishes, *StartDynamicDifficulty* gets the average values of the period, storing them for later use. The purpose of these values calculation is to verify if the player is within the correct brain activity or not and to take appropriate action. This process also registers the obtained standard deviation values, since 1/4 of this value is going to be used as difficulty increment/decrement, value chosen during testing as thought to be the most adequate.

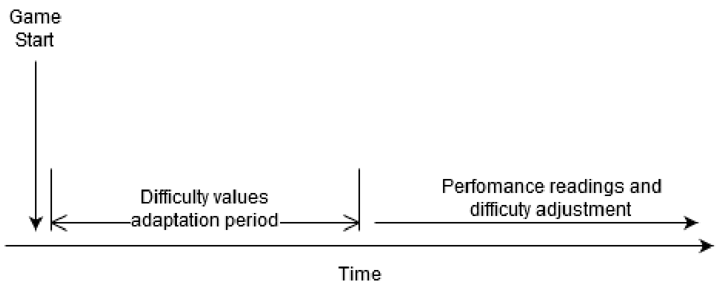


Figure 29 - Difficulty adjustment execution periods.

This way, using an adaptation period where neural activity is being read and average values recorded, it is expected that the training's bandwidth amplitude limits and amplitude value increments/decrements will be suitable to each person, since it's directly related to the user's normal brain activity, and the use of small increments/decrements will keep the game challenging. This information is logged into a file for later analysis.

After this execution, the user's personal bandwidth average is stored, in different variables by region and wave type for later access, creating the necessary conditions for the appropriate training: uptrain alpha in the posterior region, downtrain alpha in the anterior region, and downtrain beta in the anterior right hemisphere.

It is now possible for the actual game to start and with it the user's suitable difficulty adjustment. The *StartDynamicDifficulty* function, after verifying that the adaptation period has finished, starts Unity's *InvokeRepeating* method, for the *CalculateDynamicDifficulty*, the next step in this execution timeline. *InvokeRepeating* allows to repeatedly make a call to a method every specified cycle time, in this case, once each 0.5 seconds. This was the cycle chosen because the headset average readings only run once each 0.5 second as well, not being necessary to run this method with a shorter cycle. The *CalculateDynamicDifficulty* function is responsible for verifying the current performance of the player and, if he is achieving the results for the set time, increases the difficulty or decreases it if the player is not getting the expected development. The initial execution sequence can be seen in the following partial sequence diagram:

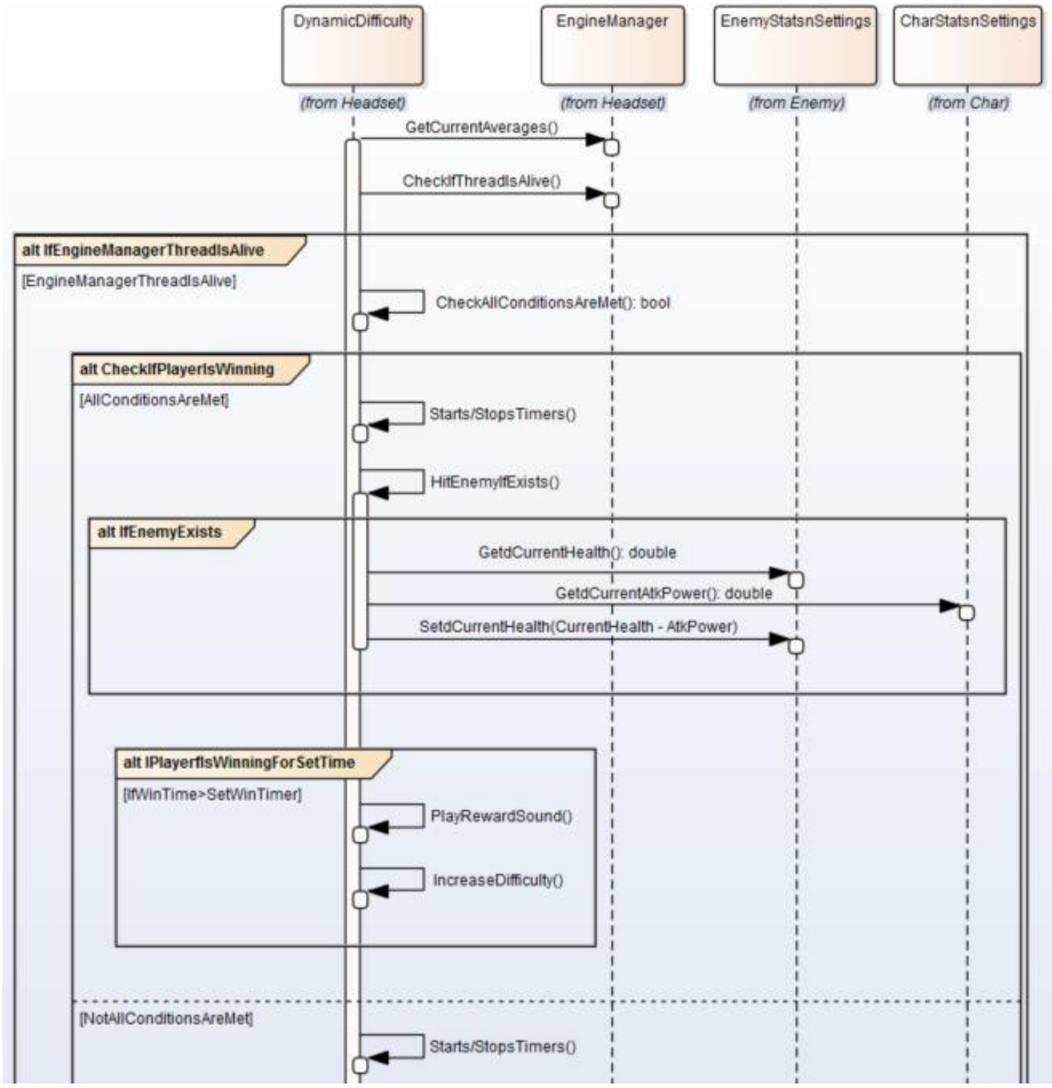


Figure 30 - Calculation of the dynamic difficulty partial sequence diagram.

First, the current averages registered by the *EngineManager* are got, with a specific function for each bandwidth type, where the corresponding array is first locked not to be accessed by other threads, and afterwards the average is calculated from the current array values. It then verifies of the value obtained is the maximum/minimum, solely for the purpose of logging this information.

Afterwards, it validates if the player is meeting the necessary conditions, by comparing the current averages got for each specific bandwidth type, with the previously set amplitude limits, obtained during the adaptation period of the game, the *StartsDynamicDifficulty* method. If the player is, for at least 10 seconds, in the right path and achieving a good performance with all the bandwidth types in the rewarding zone, the difficulty is increased. Uptraining bandwidth types is achieved by adding the defined value change increment to the bottom amplitude limit. For the bandwidths to downtrain, the opposite happens: subtracted the value change to the maximum rewarding limit. This way, the reward window will get a bit further away from the player's beginning average values.

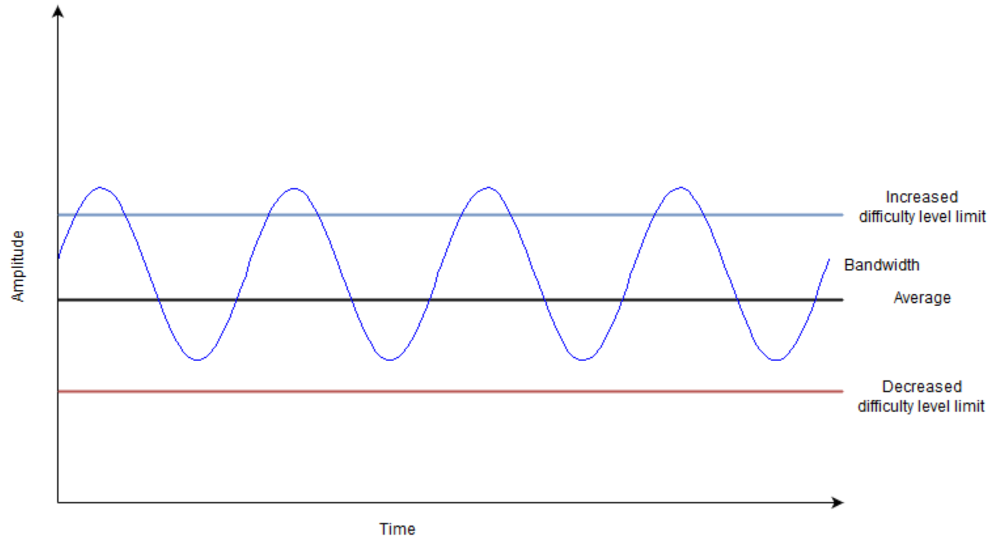


Figure 31 - Limits setting for uptrain bandwidths.

In the previous figure's example the black line represents the average value resulting of the adaptation period for a specific bandwidth. When the blue line, which represents the brain activity of the player, is above that average for the set time of 10 seconds, the limit will then change to the "Increased difficulty level limit", which is based on the mentioned standard deviation, being this new the limit for the player to be rewarded (in the figure's example, the new black line). On the other hand, if the player's bandwidth cannot stay above the average line for 20 seconds, meaning is not being able to get into the reward zone at all, then this means that the limit is set too high for the player, and then it is decreased to better suit the player performance.

The timing differences, 10 seconds and 20 seconds for increasing and decreasing respectively, were chosen through testing and are thought to be the ideal timers to achieve better results by avoiding player's frustration and not letting the game get too easy, allowing the player to finish it too quickly.

This is expected to have good results, since all these values are based on the initial adaptation period of when the game detects the player's normal brain activity, thus making this adjustment completely personal and suitable for each user.

Due to the current neurofeedback principles, all this process is done with visual and audio indication to the player. When the player is able to stay in the reward zone for 10 seconds, a bonus sound plays. Opposite to this, if the player levels down for not being able to get into the compensation zone, a penalty sound is played. Each time the player gets into the reward zone, hitting the current enemy does the visual feedback. It is verified the current enemy health and then reduced by the current player power. These enemy health points and player power values are different according to the situation.

3.2.11.1 *EnemySpawner* Difficulty Evolution

To better challenge the players during the various enemy encounters, another difficulty variation is implemented. During the game's progression, the enemy's health points and player's damage power change, so that the last enemy is harder to beat than the first one. A constant increase on the challenge is intended but, unlike the previous process, where game elements are directly affected according to the player's performance, this computation only indirectly alters the duration of each battle, since the quantity of the enemy's health points will not influence any of the neurofeedback's reward window values.

These calculations are done in another class not yet mentioned, the *EnemySpawner*. This class is responsible for the enemy instantiation in the rooms, keeping control if the enemy is still alive and spawning a new one when the last one dies, until there are no more enemies to beat. In this particular project, different enemies were not set but this functionality would be easily implemented through class heritage. The particularity about the current spawning process is the enemy health points and character attack power definition. Each time an enemy is spawned in the scene, its starting health points value and player attack power are set differently, depending on how many enemies already spawned and the current player performance.

There are attribute values that are used for these calculations: *dPowerValue* and *CurrentDifficultylevel* are used for both enemy health and player power definition. *dBaseAtkPower* is also used for the character power while *iTotalEnemiesSpawned*, *dBaseHealth* and *dEnemyHealthDivisor* are used to set the enemy's health value.

Each time an enemy spawns, its health is set according to the following equation:

$$EnemyStartingHealthPoints = \frac{((dBaseHealth \times iTotalEnemiesSpawned \times CurrentDifficultylevel)^{dPowerValue})}{dEnemyHealthDivisor}$$

This equation was achieved through experimentation and with the clear objective of setting small increments in the game's challenge. Some of the variables in the equation are constant, like the case of Enemy's *dBaseHealth* value, defined to 50, the *dPowerValue,* set to 1.25, and *dEnemyHealthDivisor* with the value of 2. These values are not used as constant in the equation, to better allow changes and adaptation through the tests, for example: the player is currently in the 6th difficulty level and the 4th enemy is going to be spawned. In this case, the equation is:

$$EnemyStartingHealthPoints = \frac{((50 \times 4 \times 6)^{1,25})}{2}$$

According to this example, the 4th enemy would have 3531.4 health points. The following chart shows this evolution. In the appendix section it is also possible to find the full-calculated results table.

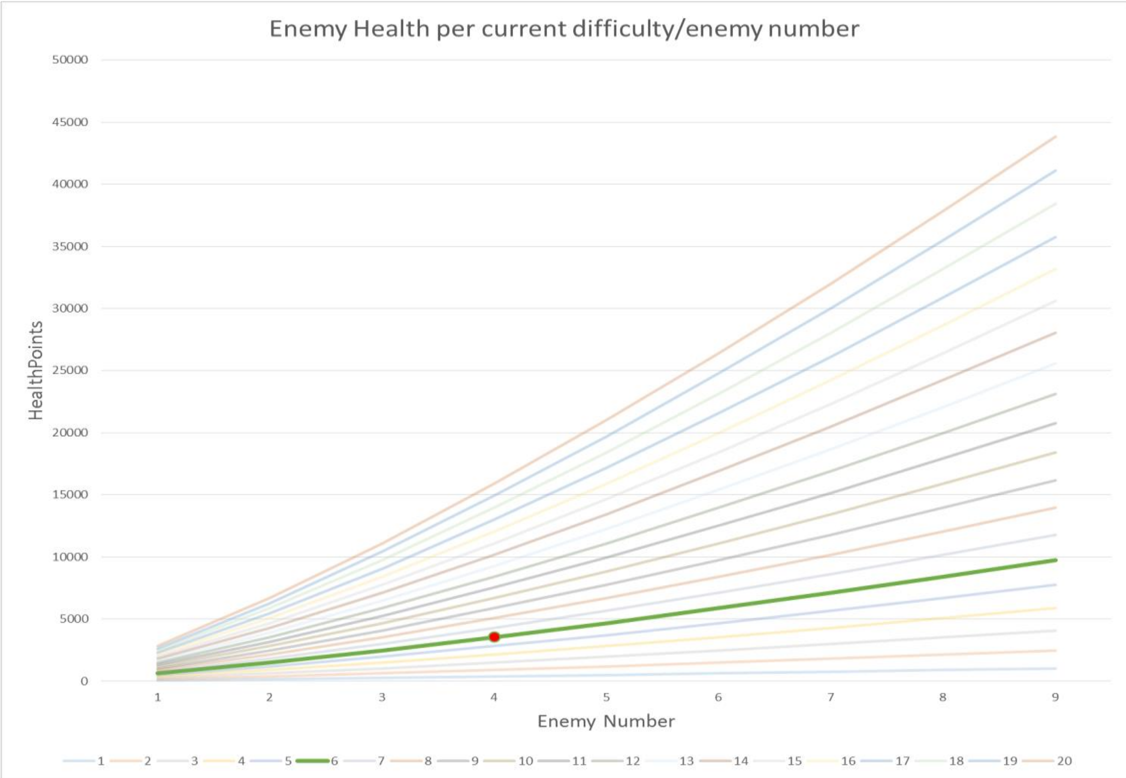


Figure 32 - Enemy health evolution chart with current example highlight.

This way, both the current player performance and the enemy number will have an influence in the enemy health points.

The other mentioned change, related with the time it takes for an enemy to be beaten, is the character attack power, or damage ability. This value also has a variation according to the player performance and it was the method found to try and balance the game, since the increase in the enemy health would create an impossible barrier for the player to surpass.

This is also done when an Enemy is spawned, using the following equation:

$$PlayerAttackPower = CurrentDifficultyLevel \times dBaseAtkPower \times dPowerValue$$

As with the previous case, the *dBaseAtkPower* is fixed at 10, and *dPowerValue* is the same as before, 1.25. For a player currently in the difficulty level 6, for example, the player would have 75 points of attack power. Since the *DynamicDifficulty* class is the one responsible for actually controlling if the enemy is taking damage or not, as explained due to being the class controlling the player’s current performance, the verification if the enemy is getting damage is done twice per second. Therefore, by putting both examples together, if the current difficulty level is at 6 and the enemy is the number 4, the enemy would have 3531.4 health points and the player 75 of attack power, which would take around 23.5 seconds to beat the enemy, considering that the player doesn't leave the reward zone and is always doing damage to the enemy. Following the same calculations, the longest battle would be with the last enemy of the level, with a level 20 current difficulty, taking 87.67 seconds of rewarding behaviour for the player to be able to surpass this enemy.

A proper balancing of the game is expected through the use of these features, making it an engaging experience, adapting properly to each and every person, giving everyone an enjoyable time while training the brain for a more healthy activity. This work's evaluation was done through the use of a quality evaluation framework and running pilot tests to better assess the efficacy of the game.

3.3 Evaluation and Testing

3.3.1 Quantitative Evaluation Framework (QEF)

To better assess this project's quality level and educational capability, a QEF approach (Escudeiro and Bidarra, 2008) was applied, in order to get an overview of the game completion level according to the initially defined ideal objectives. The criteria behind the selection of this process considered the previous experience and knowledge of the framework. This framework's application allowed a better issue tackling on this project, since it was possible to, during the development, verify the development's current state and deviation of the defined objectives. This way it was possible to identify and attempt an early solving of possible critical limitations, increasing the final product's overall quality. The QEF is based on a three-dimensional space: pedagogic, ergonomic and technical. These dimensions were thought to be the best suited for this project, because of its objectives being very close to the typical educational software goals.

Table 6 - Dimensions and related factors of the QEF.

Pedagogic	Learning
	Motivational
Ergonomic	Content Quality
	Navigation
	Adaptability
Technical	Deployment
	User Interaction

The first dimension, pedagogic, has two different factors which aim to characterize the learning and motivational requirements. The learning factor has the goal of assessing the efficacy of the application regarding the actual neurofeedback training. For this reason, self-learning and self-assessment of the player, real-time feedback, behaviour rewarding, and difficulty adjustment are considered requirements, since all of these factors are critical to this project's success. As for the motivational factor, it is intended for this application to be fun and challenging, motivating the player as much as possible over multiple sessions.

The second dimension considered, ergonomic, addresses the content quality, adaptability and navigation features of the game. This dimension's requirements consist of evaluating the graphical and written content of the game, so that the game is available to the target with high quality. The navigational aspects of the game are also considered, since this brings a better experience to the player.

Finally, the third dimension is the technical dimension, where factors like deployment and user interaction are looked into. The game's initial goal was to allow the player easily install and play the game. The homogeneity of the menus and graphical elements of the game, the predictability of the game elements can help the player experience by not creating any frustrating issue, such as an unintuitive menu that would force the player to search the desired option. All these features were thought with the goal of creating a product capable of providing the user the best possible experience while fulfilling the project's set objectives in mind. This document's full main table can be found in the appendix section.

3.3.2 Pilot Tests and Results

As mentioned in the introduction chapter, there was another scope of evaluation of this project, as such, a pilot test with real subjects was run. Having the equipment available for two weeks, this was intended to not only help with the QEF usage and application assessment, but also to evaluate the results of the neurofeedback session. The test was run on 9 subjects, with ages ranging from 20 to 60 years old, of which 7 were male and 2 were female. These individuals were chosen in order to get an overview of the target audience, and to confirm that the game would not harm people's health. Both the subject's and the equipment's availability were taken into account as well, since they have also played a major role in the selection of participants and the tests. The tests were run in the most comfortable environment possible for the subject, with special focus on keeping the user concentrated on the game and without any major distractions. Verbal explanations were kept to a minimum during the experience, since it was made an effort to keep these as infrequent as possible because all the subject's movements and actions could create artefacts in the headset readings.

The player started by answering a few questions about the overall perception of his anxiety and previous experience with neuroheadsets and games. 56% of the subjects consider themselves anxious but only 22% assumed that they felt anxiety the day the test was run. About the previous experience with these types of applications, 78% of the individuals assume they regularly play games and 67% had never used a neural headset, being this their first experience with this type of equipment.

Table 7 - The results of the questionnaire before playing.

	Yes	No
Did you feel anxiety today?	78%	22%
Do you consider yourself an anxious person?	56%	44%
Do you usually play videogames?	78%	22%
Have you ever tried a neural headset?	33%	67%

After these first questions, the equipment was set up as well as possible. The following figure shows that, despite the efforts to avoid this, some sensors contact quality were always average. The optimal conditions were not achieved since multiple factors existed that influenced the contact quality. Half the subjects had medium to long hair, which can bring complications to the headset contact quality, for example.

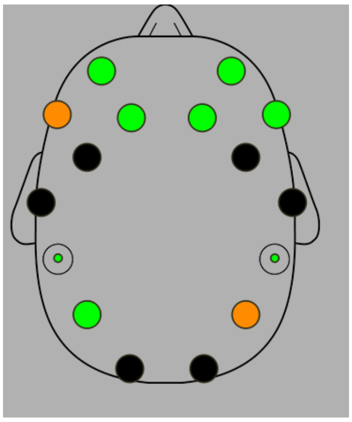


Figure 33 - Subject's typical contact quality. Green – Good; Orange – Poor; Black – Not connected.

The artefacts in the headset readings brought some problems analysing the results, especially with two of the tests made, which had multiple moments where the readings were not accurate, becoming harder to evaluate these. This can be justified, not only with the poor contact quality of the equipment in those regions but with excessive movement or talking as well.

The individuals played the game, while it registered into log files their neural activity. With these files it was possible to better assess the project's performance. 78% of the subjects were able to finish the game which means that, at least for some periods, they were able to damage the enemies that were in front of them by achieving all the necessary conditions, otherwise they would not progress.

Despite this, most of tests revealed that the project's goal was very hard to achieve with just one session. For the purpose of the following analysis, the conditions for the player to be able to advance in the game will be joined in three categories. It will be referred as alpha posterior, aimed to uptrain, alpha and beta anterior to downtrain. In the optimal solution, the subject would be able to increase the alpha value in the posterior region and decrease both alpha and beta frequencies in the anterior region.

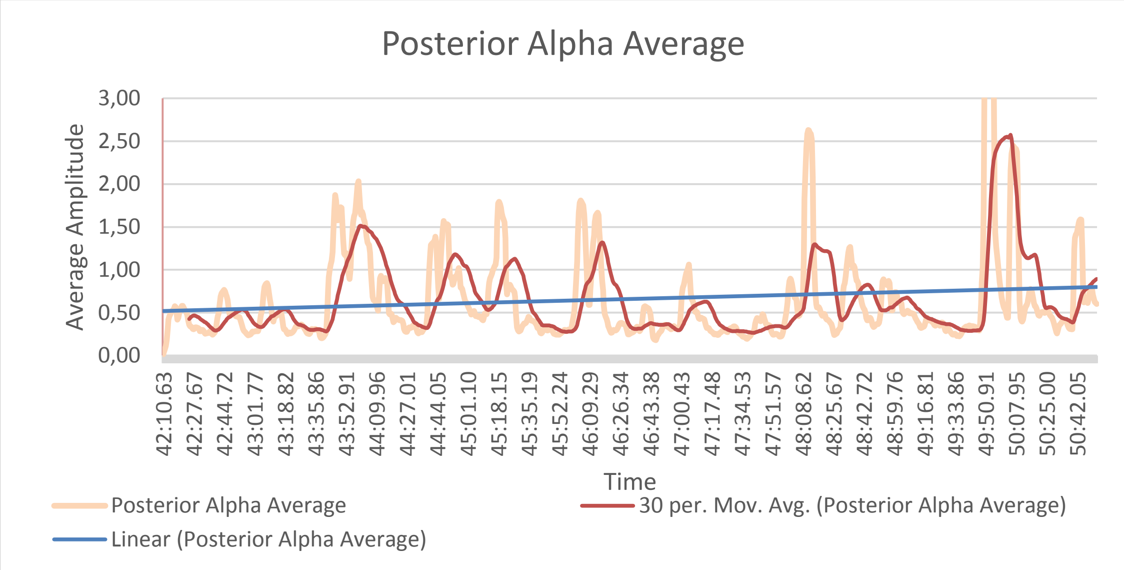


Figure 34 - Successful posterior alpha evolution of one of the subjects.

Of the 9 subjects only 1 was not able to increase the posterior alpha activity. The above figure, shows one of the examples where the posterior alpha average was increased during the session, which fulfilled the goal. In orange is the representation of the average readings, in red the trend line of moving average for 30 periods, for the chart to be more readable. In blue is the linear trend line showing the overall evolution. This was the behaviour registered for most of the subjects were the test was run. The same didn't happen with the other two conditions, since both alpha and beta in the anterior region of the brain, showed opposite results. While alpha was read throughout all the anterior region, beta was only got from the right hemisphere, obtaining similar results for both these bandwidth frequencies.

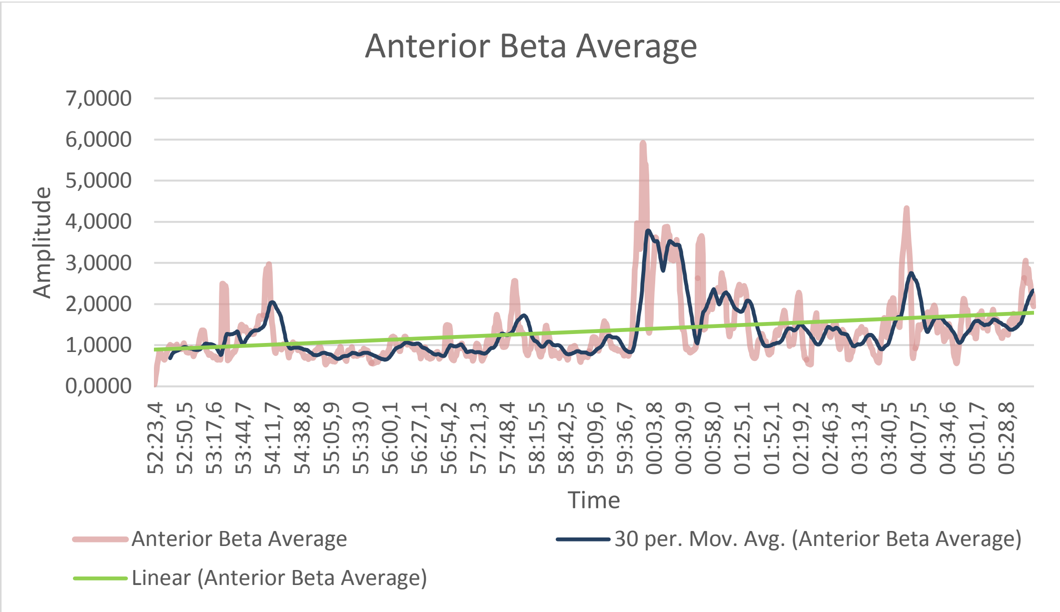


Figure 35 - Anterior beta average evolution example.

The above example of one of the subjects shows exactly this, a gradual increase in the beta frequency activity in the posterior right hemisphere of the brain. This was not the intended result of the application, but further training could provide a different answer. One of the possible reasons for this to have happened is the presence of artefacts, or noise, in the headset readings. Since these two bandwidths are supposed to be downtrained, and their amplitudes reduced, if at some point noise is registered by the headset, the values gotten in that period will be very high amplitudes, completely the opposite of the rewarding situation.

One of the subjects though, was able to get the opposite results. This individual was able to successfully decrease both alpha and beta frequencies activities in the anterior region, failing only to increase the alpha bandwidth frequency in the posterior region.

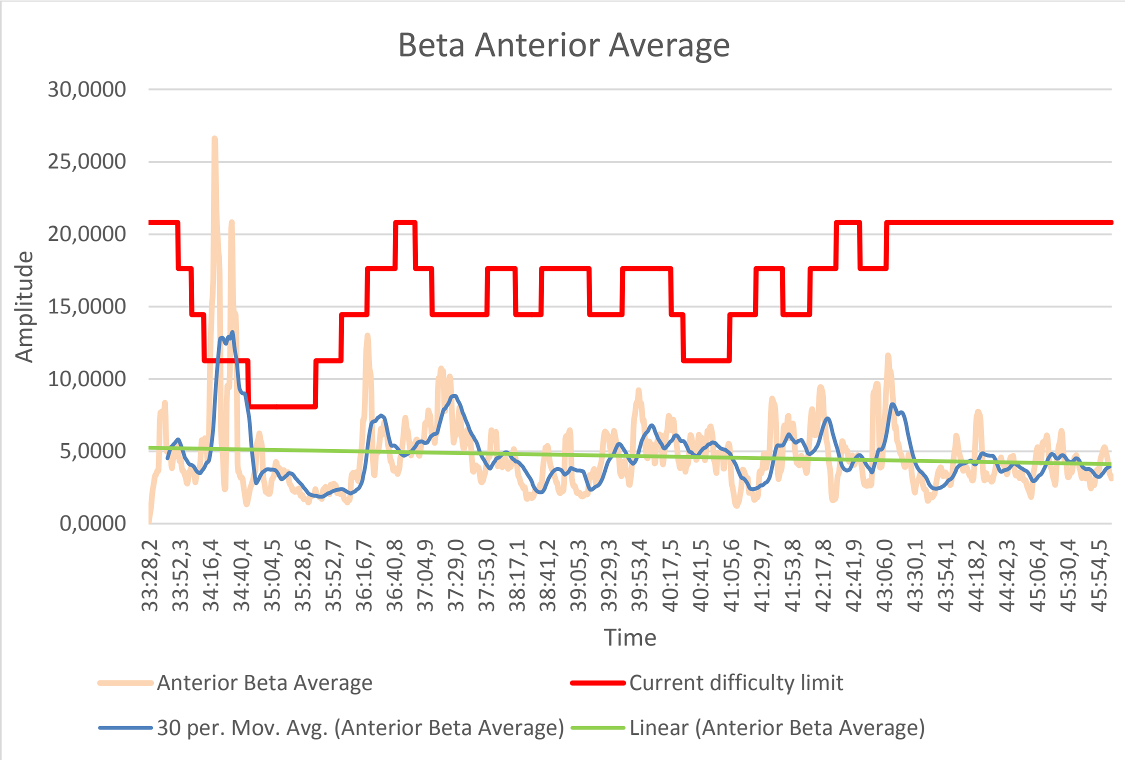


Figure 36 - Subject's beta anterior average readings.

As can be seen in the previous chart, this subject was able to successfully decrease the activity of the beta frequency in the anterior right region. The red line shows the current amplitude limit set as result of the dynamic difficulty adjustment. In this specific bandwidth type, the higher the player current level, the lower the value of the limit, because this bandwidth was being downtrained. Since this bandwidth type was only one of the conditions, it can be seen situations like the final moments of this experiment, where the player's current amplitude is well below the set limit, but the application difficulty isn't changed. This happened since one of the other conditions was not being met, as can be proved with the following chart:

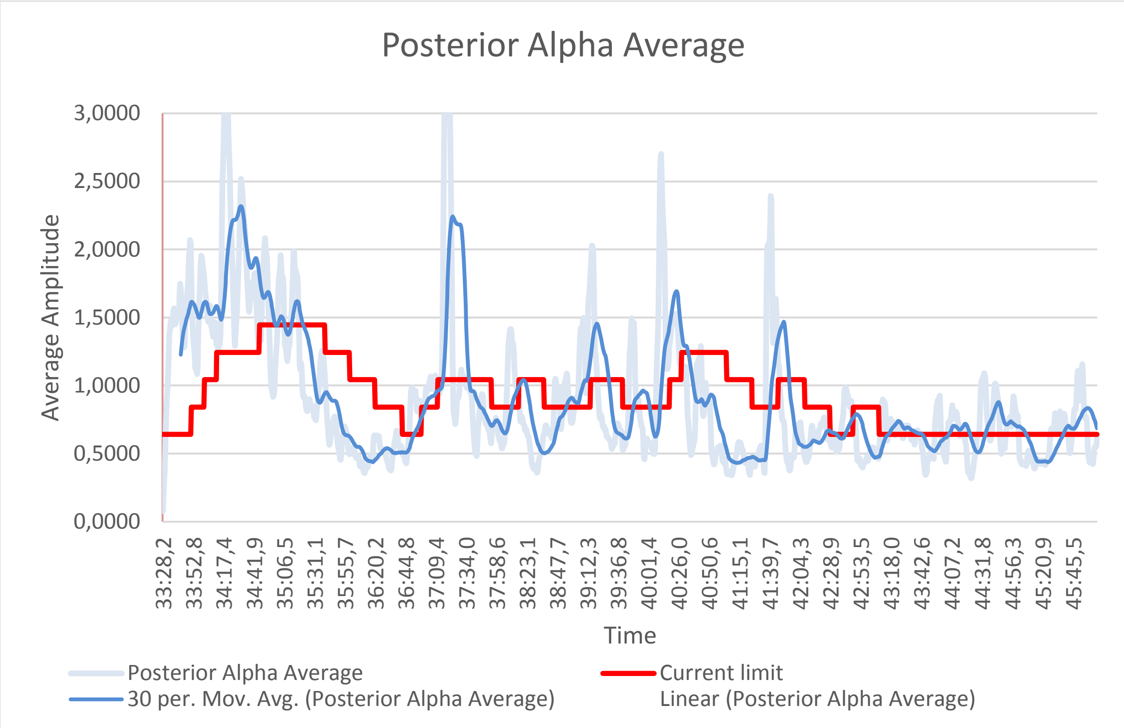


Figure 37 - Same subject's posterior alpha.

In the same time period as on the previous chart, the subject was not fulfilling the necessary conditions regarding the posterior alpha and therefore, the game took no action to increase the difficulty, which is the correct procedure.

At the end of each test, the second part of the questionnaire was presented to the subject, with questions related with their experience and with the game performance. Some of the main questions showed all the subjects were happy with the game's duration being a challenging experience to them. 67% felt that their neural activity had direct influence in the gameplay. Also 67% found the experience fun, but the same number felt frustration at some point during the game which might connect with the neurofeedback training being unsuccessful in this session, where the brain's activity didn't decrease, but increased instead. Regarding the player's ability to better control the anxiety after this experiment, 78% don't feel that they are able to do so, but all of the subjects believe that with more sessions that would be possible.

Table 8 - Questionaire results' summary.

	Yes	No
Did you find any error during the execution?	0%	100%
Were the presented messages simple and easy understandable?	100%	0%
Did you play only once?	89%	11%
Did you feel you controlled the character's actions in the game?	56%	44%
Did the characters have the expected behaviour?	89%	11%
Do you think the game with neural activity concept original?	100%	0%
Do you think the audio and video improved your experience?	78%	22%
Do you consider the experience fun?	67%	33%
At any moment, did you feel frustration while playing?	67%	33%
Was the experience challenging?	100%	0%
Does the game allows self-learning?	67%	33%
Was the game a permanent challenge?	100%	0%
After this experiment, do you think you are better able to control your anxiety?	22%	78%
Do you think your results could improve with more sessions?	100%	0%
Do you think the scenery variation would help you to play again?	78%	22%
Were you able to damage an enemy?	100%	0%
Did you feel that your brain's activity had a role in the enemy's damage?	67%	33%
Were you able to reach the end of the game?	78%	11%

With the help of this questionnaire it was possible to get feedback from actual subjects regarding the game. Part of this questionnaire was aimed to answer some of the requirements present in the QEF table, of which a summary can be observed in the next table.

Table 9 - Resulting QEF's summary.

q	**D**	Q_i	**Dimension**	Q_j	p_{ij} (factor's *j* weight in *i*'s Dim*)* [0,1]	**Factor**
77%	0,51	70,80	**Pedagogic**	72,54	0,59	Learning
				68,29	0,41	Motivational
		79,56	**Ergonomic**	100,00	0,24	Content Quality
				58,93	0,41	Navigation
				90,00	0,35	Adaptability
		62,87	**Technical**	40,63	0,38	Deployment
				76,22	0,63	User Interaction

According to the input from the questionnaire, together with the developer evaluation, it was reached a value of 77% completion of the initial objectives. This value is considered positive since the main features of the game, necessary for this project, were successfully implemented. Some low priority requirements were not implemented, since the main objective of this project was to create a prototype, but were thought initially as part of an ideal solution. Some examples of these requirements are the installation package and update, a help button or even a second language.

A particular requirement that should be looked into with care, is *"PL11- The player progresses through multiple sessions"*, which would fit the studied neurofeedback procedures. This was not applicable on this scope, since only a few tests were run and the lack of availability of the equipment prevented additional sessions. Although this, all of the subjects indicated in the questionnaire that they believe that with more sessions, their results would improve, which is a very positive reaction. Other relevant requirements, "*PL08- Learner gets rewarded for pretended behaviour"* and *"PL09- The game adjusts difficulty correctly to the player performance"*, cannot be translated to a question, but can be proved with the application analysis showed before. These requirements are thought to be 100% fulfilled, since it became evident that the game was able to adjust to the current player performance, and the player was rewarded when the expected conditions were met. The full questionnaire results and QEF table can be found in the appendix section. After these tests, some conclusions can be made and future work identified.

4. Conclusion

In this chapter it will be discussed the project's overall objective completion by looking into the work developed and possible future improvements.

4.1 Overall Results

The main goal of this project was the creation of a serious game prototype with dynamic difficulty and neurofeedback capabilities, and that was successfully accomplished. The game was created and works according to the main objectives initially defined, having neuroheadset integration, scenery variability and dynamic difficulty adjustment. The game follows the understood neurofeedback current procedures giving adequate feedback according to the players' results, as could be verified by the tests.

For a large extent of this project's duration, the game was developed by using the equipment's control panel. This was the solution found until the equipment was available, allowing the developer to progress with the application's construction, simulating the headset's data through the control panel. According to the project's QEF, the game requirements were achieved by 77% with the considered most relevant objectives being accomplished.

Having the equipment available for two weeks, the tests made on 9 individuals showed positive aspects but other factors revealed inconclusive, since diverse results were observed. It was verified that the application worked as devised, doing no harm to the subjects, but its actual anxiety reducing capability is yet to be demonstrated. The players were able to have the expected neural activity during periods of time, being rewarded when the conditions were met and, according to the bibliography, more sessions and training would make possible to observe the first results in anxiety reducing. With 100% of the tested people believing that with more sessions, they would be able to better control their anxiety, it was possible to verify that the application was very well received, being an original concept. On the downside, the tests' registered results showed that for some subjects, the expected behaviour was not achieved, being unable to properly decrease both alpha and beta bandwidth amplitudes in the anterior region of the brain for most cases. One of the main reasons that can justify this, is the equipment's noise registered during the gameplay. Since the decrease in the alpha and beta's amplitudes in the anterior region was the rewarded behaviour, the created artefacts momentarily increased the values by a significant number, even if the player was in his brain's activity rewarding zone at that moment. This put at risk the game's calculations related with the rewarding zone, since, from the game's point of view, the player did output a major increase in the amplitudes, due to the values that were being registered. One other possible reason for this behaviour, with the subjects not being able to reduce the anterior alpha and beta's activity, is the frustration created at some point during the game, with 68% of the player's reporting they experienced that feeling at least once during the gameplay. The dynamic difficulty was, according to the personal feedback obtained, very demanding after some point, indicating that this should be on improvement point. Despite this, the first neurofeedback sessions usually involve a lot of trial-and-error (Connolly et al., 2013b) so it was expected for the individuals to not achieve results in the first sessions.

4.2 Future Work

Future work will involve further improvements and testing. The main goal of this development was the creation of an application with adequate neurofeedback capabilities, which was successfully achieved, but a consolidation of this work is necessary.

The dynamic difficulty adjustment could be refined and improved, since the frustration most of the subjects felt during the experience is probably related with the difficulty's increments being too high and a smoother increase might have helped with this issue. To tackle this, more testing should be made with small adjustments to the dynamic difficulty module, evaluating the feedback of both the application results as well as the subjects' experience. One other identified improvement that would help the application's user attain better results, would be the inclusion into the game mechanics of the bandwidth's upper limits, meaning that for the uptraining frequencies not only the lower limit would exist but also an upper amplitude limit. This would reduce the game's exposure to artefact readings, since with this limit's implementation, read values that exceeded the normal brain's activity amplitudes would not reward the player. This would clear the uptraining frequencies game's readings of artefacts, giving more accurate feedback.

The graphical components of the game could bring another dynamic to the game. Subjects' opinion is that the character animations are good but "could be better" being this one of the points to improve in the future, to bring more fun and enthusiasm to the gameplay. Better character animations and various enemies, which could be implemented through heritage of the existing classes, should be included into the game to create a more immersive experience. The same can be said about the random room generation, where elements like the positioning of different objects, obstacles, inside the rooms or random spawn positions where one or more enemies could spawn in different spots of the room. The scenery variation could be further enhanced if a theme shuffling process is implemented, creating different themes between rooms inside the same level. All this could improve the current game's dynamic and a more engaging experience, increasing the motivation not only over multiple sessions but on the first gameplay as well.

Regarding the initial objectives, some of the QEF stated requirements should be implemented to get closer to the ideal solution: the availability of a help button, a helpful message when the player is being unable to progress, aiming to avoid his frustration, these features would increase the final product's quality.

Afterwards, additional testing is necessary. On a first phase with the same number of subjects with similar conditions, to verify the improvements made with current main issues. If the results are satisfactory and a positive feedback is received, a new phase of testing is necessary. This time with a larger group over multiple sessions, it would be possible to assess the actual capability of the game in helping anxiety issues, studying and evaluating the results got from these experiments.

In the future, the project's procedures which were object of study should be kept updated, especially the rapidly increasing knowledge of subjects such as the brain and serious games. The knowledge of the brain processes and behaviours is augmenting but there is still a large part unknown, and constant update on the currently applied methods is necessary. The methods used in this project should be regularly enhanced and updated, to follow the most recent findings. The game development is also a discipline with an increase of interest in recent years and new or better processes might be created, allowing the players a more engaging and fun experience. Due to the mentioned reasons, this project should be considered as a "work-in-progress", and kept updated properly according to the most accurate knowledge of its subjects.

For this work to be possible, a big effort was made by the developer, with multiple challenges addressed, learning about a complex new field, neurofeedback and the brain. It was a very exciting and rewarding experience. The new insights about the brain behaviour and neurofeedback, together with the experience of developing a full video game with a specific peripheral integration, are experiences that will not be forgotten.

Glossary

Analytic hierarchy process – Structured technique for dealing with complex decisions, providing a comprehensive and rational framework for structuring a problem, representing and quantifying its elements, for relating those elements to the goals and evaluate alternative solutions (Saaty, 2008).

Artefacts (EEG) – "*All signals that appear in the EEG record which don't come from the brain."*(GuruvaReddy and Narava, 2013).

Electroencephalography (EEG) – "*The record of the electric signal generated by the cooperative action of brain cells*" (Blinowska and Durka, 2006).

International 10-20 System – A system using letters and numbers to identify scalp electrodes positions (Oostenveld and Praamstra, 2001).

Value Proposition – A value proposition is a message that resumes the company's products that have value to the customer. It should target what is the product, the target customer, the value provided and why the product is different from the competition.

References

Anxiety and Depression Association of America, 2014a. *Facts & Statistics* [WWW Document]. URL http://www.adaa.org/about-adaa/press-room/facts-statistics (accessed 1.6.16).

Anxiety and Depression Association of America, 2014b. *Treatment* [WWW Document]. URL http://www.adaa.org/finding-help/treatment (accessed 1.6.16).

Armstrong, P., 2015. Bloom's Taxonomy.

Benbadis, S.R., 2015. Normal Awake EEG: Overview, Waveform Description, Clinical Correlation.

BioExplorer [WWW Document], n.d. URL http://www.cyberevolution.com/ (accessed 10.13.16).

Biometrics Definition, 2015. . Dictionary.com.

Blinowska, K., Durka, P., 2006. Electroencephalography (EEG), in: Akay, M. (Ed.), Wiley Encyclopedia of Biomedical Engineering. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Blunt, R., 2008. Does Game-Based Learning Work? Results from Three Recent Studies. Adv. Distrib. Learn.

Boyle, E., 2013. Psychological Aspects of Serious Games, in: Psychology, Pedagogy and Assessment in Serious Games, Advances in Game-Based Learning. IGI Global, p. 522.

Budzynski, T.H., Budzynski, H.K., Evans, J.R., Abarbanel, A., 2009. Introduction to Quantitative EEG and Neurofeedback: Advanced Theory and Applications. Academic Press.

Bystritsky, A., Khalsa, S.S., Cameron, M.E., Schiffman, J., 2013. Current Diagnosis and Treatment of Anxiety Disorders. P T 38, 30–38, 41–44,57.

Clemens, N.A., 2003. A Psychodynamic Perspective on Anxiety: J. Psychiatr. Pract. 9, 385–387. doi:10.1097/00131746-200309000-00007

Connolly, T.M., Hainey, T., Boyle, E., Baxter, G., Moreno-Ger, P., 2013a. Psychology, Pedagogy and Assessment in Serious Games, Advances in Game-Based Learning. Information Science Reference.

Connolly, T.M., Hainey, T., Boyle, E., Baxter, G., Moreno-Ger, P., 2013b. Psychology, Pedagogy and Assessment in Serious Games, Advances in Game-Based Learning. IGI Global.

Crosta, P., 2015. Anxiety: Causes, Symptoms and Treatments. Med. News Today.

Csikszentmihalyi, M., 1990. Flow: The Psychology of Optimal Experience. Harper & Row, p. 303.

Cullinane, A., 2009. Bloom's Taxonomy and its Use in Classroom Assessment. ResearchGate 1.

Definition of Nervous System [WWW Document], n.d. . Merriam-Webster. URL http://www.merriam-webster.com/dictionary/nervous+system (accessed 9.26.16).

Demos, J.N., 2005. *Getting started with neurofeedback*, First. ed. W. W. Norton & Company.

Dettmer, P., 2005. New blooms in established fields: Four domains of learning and doing. Roeper Rev. 28, 70–78. doi:10.1080/02783190609554341

Djaouti, D., Alvarez, J., Jessel, J.-P., Rampnoux, O., 2010. Origins of Serious Games.
Emotiv, 2015. Emotiv EPOC specifications.
Enterprise Architect UML modeling tool [WWW Document], n.d. URL http://www.sparxsystems.com/ (accessed 10.12.16).
Escudeiro, P., Bidarra, J., 2008. Quantitative Evaluation Framework (QEF). ResearchGate 16.
Eurostat, 2013. *Persons reporting a work-related health problem by sex, age and type of problem*.
flOw [WWW Document], n.d. . Playstation. URL https://www.playstation.com/en-us/games/flow-ps3/ (accessed 9.24.16).
Gage, S.B. de, Moride, Y., Ducruet, T., Kurth, T., Verdoux, H., Tournier, M., Pariente, A., Bégaud, B., 2014. *Benzodiazepine use and risk of Alzheimer's disease: case-control study*. BMJ 349.
Garris, R., Ahlers, R., Driskell, J.E., 2002. Games, Motivation, and Learning: A Research and Practice Model. Simul. Gaming 33, 441–467. doi:10.1177/1046878102238607
Gevensleben, H., Holl, B., Albrecht, B., Schlamp, D., Kratz, O., Studer, P., Rothenberger, A., Moll, G.H., Heinrich, H., 2010. Neurofeedback training in children with ADHD: 6-month follow-up of a randomised controlled trial. Eur. Child Adolesc. Psychiatry 19, 715–724. doi:10.1007/s00787-010-0109-5
GIMP [WWW Document], n.d. . GIMP. URL //www.gimp.org/ (accessed 10.12.16).
GitLab [WWW Document], n.d. URL https://about.gitlab.com/ (accessed 10.12.16).
Guger, C., Edlinger, G., Harkam, W., Niedermayer, I., Pfurtscheller, G., 2003. How many people are able to operate an EEG-based brain-computer interface (BCI)? IEEE Trans. Neural Syst. Rehabil. Eng. 11, 145–147. doi:10.1109/TNSRE.2003.814481
GuruvaReddy, A., Narava, S., 2013. Artifact Removal from EEG Signals. Int. J. Comput. Appl. 77, 17–19. doi:10.5120/13543-1175
Hammond, D.C., 2011. What is Neurofeedback: An Update. J. Neurother. 15, 305–336. doi:10.1080/10874208.2011.623090
Hunicke, R., Chapman, V., 2014. AI for dynamic difficulty adjustment in games. ResearchGate 2.
Jennings-Teats, M., Smith, G., Wardrip-Fruin, N., 2010. Polymorph: Dynamic Difficulty Adjustment Through Level Generation. Presented at the Proceedings of the 2010 Workshop on Procedural Content Generation in Games.
Joystiq interview: Jenova Chen - flOw creator [WWW Document], n.d. . Engadget. URL https://www.engadget.com/2006/09/18/joystiq-qanda-jenova-chen/ (accessed 9.24.16).
Lofthouse, N., Arnold, L.E., Hurt, E., 2012. Current Status of Neurofeedback for Attention-Deficit/Hyperactivity Disorder. Curr. Psychiatry Rep. 14, 536–542. doi:1007/s11920-012-0301-z
Longo, L.P., Johnson, B., 2000. *Addiction: Part I. Benzodiazepines—Side Effects, Abuse Risk and Alternatives*. Am. Fam. Physician 61, 2121–2128.
Madden NFL 09 First Hands-On [WWW Document], n.d. . GameSpot. URL http://www.gamespot.com/articles/madden-nfl-09-first-hands-on/1100-6191473/ (accessed 9.24.16).
Madden NFL 09 : Madden 2009 - EA Games [WWW Document], n.d. URL http://www.ea.com/madden-09 (accessed 9.24.16).

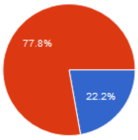
Making Crash Bandicoot – part 6 :: All Things Andy Gavin, n.d.
Mandal, A., 2010. What is the Nervous System? [WWW Document]. News-Medicalnet. URL http://www.news-medical.net/health/What-is-the-Nervous-System.aspx (accessed 10.1.16).
Michael, D., Chen, S., 2006. Serious Games: Games that educate, train and inform. Thomson Course Technology PTR.
Mind WorkStation [WWW Document], n.d. URL https://www.transparentcorp.com/products/mindws/index.php (accessed 10.13.16).
Missura, O., 2015. Dynamic Difficulty Adjustment. Rheinische Friedrich-Wilhelms-Universität Bonn.
Mulkerin, T., Jun. 23, 2016, 21, 107, n.d. “Crash Bandicoot” is coming back — and PlayStation dropped hints about it for a long time [WWW Document]. Tech Insid. URL http://www.techinsider.io/crash-bandicoot-remastered-games-ps4-2016-6 (accessed 9.24.16).
MyndPlay Ltd [WWW Document], n.d. URL http://store.myndplay.com/products.php?prod=28 (accessed 9.29.16).
National Institute of Mental Health, 2015. *Anxiety Disorders* [WWW Document]. URL http://www.nimh.nih.gov/health/topics/anxiety-disorders/index.shtml (accessed 1.14.16).
NeuroPlus Attention Training [WWW Document], n.d. URL http://www.neuro.plus/#pricing (accessed 10.13.16).
Ninaus, M., Witte, M., Kober, S.E., Friedrich, E.V.C., Kurzmann, J., Hartsuiker, E., Neuper, C., Wood, G., 2013. Neurofeedback and Serious Games, in: Psychology, Pedagogy and Assessment in Serious Games, Advances in Game-Based Learning. IGI Global, p. 522.
Oostenveld, R., Praamstra, P., 2001. The five percent electrode system for high-resolution EEG and ERP measurements. Clin. Neurophysiol. 112, 713–719. doi:10.1016/S1388-2457(00)00527-7
Physicians’ Desk Reference, 2014. *Xanax (alprazolam)* - Drug Summary [WWW Document]. URL http://www.pdr.net/drug-summary/Xanax-alprazolam-1873.31 (accessed 1.14.16).
PLX XWave - Brainwave to iPhone Interface [WWW Document], n.d. URL http://www.plxwave.com/ (accessed 9.29.16).
Pop-Jordanova, N., Demerdzieva, A., 2010. Biofeedback Training for Peak Performance in Sport - Case Study. Maced. J. Med. Sci. 3, 113–118. doi:10.3889/MJMS.1857-5773.2010.0098
Prabhakar, S., Pankanti, S., Jain, A.K., 2003. Biometric recognition: security and privacy concerns. IEEE Secur. Priv. Mag. doi:10.1109/MSECP.2003.1193209
Ron-Angevin, R., Díaz-Estrella, A., 2009. Brain–computer interface: Changes in performance using virtual reality techniques. Neurosci. Lett. 449, 123–127. doi:10.1016/j.neulet.2008.10.099
Saaty, T.L., 2008. Decision making with the analytic hierarchy process. Int. J. Serv. Sci. 1, 83. doi:10.1504/IJSSCI.2008.017590
Schuller, B.W., Dunwell, I., Weninger, F., Paletta, L., 2013. Serious Gaming for Behavior Change: The State of Play. IEEE Pervasive Comput. 12, 48–55. doi:10.1109/MPRV.2013.54

Stufflebeam, R., 2008. Neurons, Synapses, Action Potentials, and Neurotransmission - The Mind Project [WWW Document]. URL http://www.mind.ilstu.edu/curriculum/neurons_intro/neurons_intro.php (accessed 10.2.16).
Sutoyo, R., Winata, D., Oliviani, K., Supriyadi, D.M., 2015. Dynamic Difficulty Adjustment in Tower Defence. Procedia Comput. Sci. 59, 435–444. doi:10.1016/j.procs.2015.07.563
Understanding Frequencies [WWW Document], n.d. . Brain-Trainercom. URL https://brain-trainer.com/answers/about-the-brain/understanding-frequencies/ (accessed 10.5.16).
Unity - Game Engine [WWW Document], n.d. URL https://unity3d.com/ (accessed 10.12.16).
Unity - Scripting API [WWW Document], n.d. URL https://docs.unity3d.com/ScriptReference/ (accessed 10.12.16).
Universal LPC Sprite Sheet Character Generator [WWW Document], n.d. URL http://gaurav.munjal.us/Universal-LPC-Spritesheet-Character-Generator/# (accessed 10.12.16).
Unreal Engine 4 [WWW Document], n.d. URL https://www.unrealengine.com/what-is-unreal-engine-4 (accessed 10.12.16).
Visual Studio [WWW Document], 2016. . Vis. Studio. URL https://www.visualstudio.com/ (accessed 10.12.16).
Wang, Q., Sourina, O., Nguyen, M.K., 2011. Fractal dimension based neurofeedback in serious games. Vis. Comput. 27, 299–309. doi:10.1007/s00371-011-0551-5
Warner, S., 2013. Cheat Sheet for Neurofeedback 28.
WHO, 2013. *Data and Statistics*. World Health Organization.
Zilles, K., Amunts, K., 2010. Centenary of Brodmann's map — conception and fate. Nat. Rev. Neurosci. 11, 139–145. doi:10.1038/nrn2776

Appendices

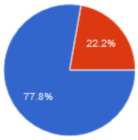
Appendix A. Questionnaire’s results.

Sentiu ansiedade durante o dia de hoje?



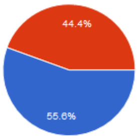
Sim	2	22.2%
Não	7	77.8%
Other	0	0%

Costuma jogar videojogos?



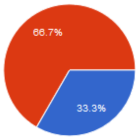
Sim	7	77.8%
Não	2	22.2%
Other	0	0%

Considera-se uma pessoa ansiosa?



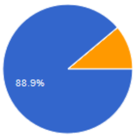
Sim	5	55.6%
Não	4	44.4%
Other	0	0%

Alguma vez usou um headset neuronal?



Sim	3	33.3%
Não	6	66.7%
Other	0	0%

A aplicação foi rápida a iniciar?

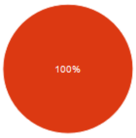


Sim	8	88.9%
Não	0	0%
Other	1	11.1%

O estilo do interface gráfico era coerente?

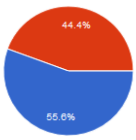
Sim	9	100%
Não	0	0%
Other	0	0%

Detetou algum erro na execução?



Sim	0	0%
Não	9	100%
Other	0	0%

Sentiu que controlou as ações do personagem no jogo?

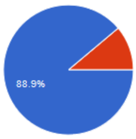


Sim	5	55.6%
Não	4	44.4%
Other	0	0%

As mensagens apresentadas eram simples e fáceis de entender?

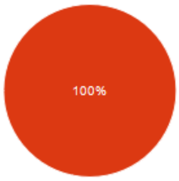
Sim	9	100%
Não	0	0%
Other	0	0%

Os personagens do jogo tiveram o comportamento esperado?



Sim	8	88.9%
Não	1	11.1%
Other	0	0%

Sentiu que o jogo pudesse ofender alguma caracteristica em particular (grupo étnico ou género)?

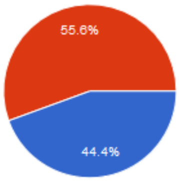


Sim	0	0%
Não	9	100%
Other	0	0%

Jogou apenas uma vez?

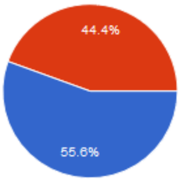
Sim	8	88.9%
Não	1	11.1%
Other	0	0%

Os menus eram intuitivos?



Sim	4	44.4%
Não	5	55.6%
Other	0	0%

O tema e conteúdo gráfico do jogo eram originais?

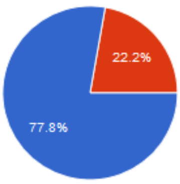


Sim	5	55.6%
Não	4	44.4%
Other	0	0%

Acha o conceito de jogo com actividade neuronal original?

Sim	9	100%
Não	0	0%
Other	0	0%

Acha que o áudio e o vídeo do jogo melhoraram a sua experiência?

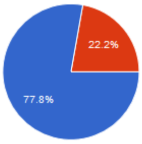


Sim	7	77.8%
Não	2	22.2%
Other	0	0%

As instruções eram claras?

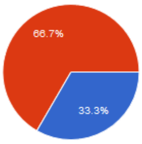
Sim	9	100%
Não	0	0%
Other	0	0%

As instruções eram necessárias?



Sim	7	77.8%
Não	2	22.2%
Other	0	0%

A variação de cenário motivou-o?

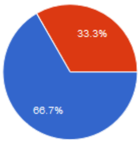


Sim	3	33.3%
Não	6	66.7%
Other	0	0%

O objetivo do jogo foi claro desde início?

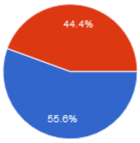
Sim	9	100%
Não	0	0%
Other	0	0%

O jogo permite auto-aprendizagem?



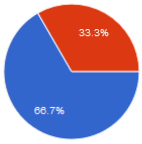
Sim	6	66.7%
Não	3	33.3%
Other	0	0%

Conseguiu auto-avaliar os seus resultados em tempo real?



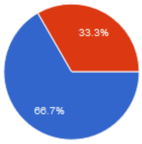
Sim	5	55.6%
Não	4	44.4%
Other	0	0%

Considerou a experiência divertida?



Sim	6	66.7%
Não	3	33.3%
Other	0	0%

Em algum momento sentiu frustração enquanto jogava?



Sim	6	66.7%
Não	3	33.3%
Other	0	0%

A experiência foi desafiante?

Sim	9	100%
Não	0	0%
Other	0	0%

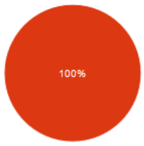
O progresso do jogo era intuitivo?

Sim	9	100%
Não	0	0%
Other	0	0%

O jogo foi um desafio constante?

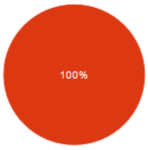
Sim	9	100%
Não	0	0%
Other	0	0%

A duração foi demasiado curta?



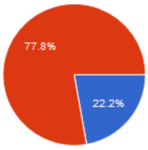
Sim	0	0%
Não	9	100%
Other	0	0%

A duração foi demasiado longa?



Sim	**0**	0%
Não	**9**	100%
Other	**0**	0%

Depois desta experiência de jogo, acha capaz de melhor controlar a sua ansiedade?

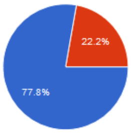


Sim	**2**	22.2%
Não	**7**	77.8%
Other	**0**	0%

Acha que os seus resultados poderiam melhorar com mais sessões?

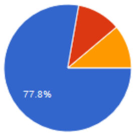
Sim	**9**	100%
Não	**0**	0%
Other	**0**	0%

Acha que a variabilidade dos cenários ajudaria a que voltasse a jogar?



Sim	**7**	77.8%
Não	**2**	22.2%
Other	**0**	0%

Conseguiu chegar ao final do jogo?



Sim	**7**	77.8%
Não	**1**	11.1%
Other	**1**	11.1%

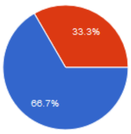
Conseguiu provocar dano a um inimigo?

Sim	**9**	100%
Não	**0**	0%
Other	**0**	0%

Acha o headset confortável de usar?

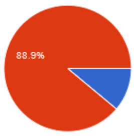
Sim	**8**	88.9%
Não	**1**	11.1%
Other	**0**	0%

Sentiu que a sua atividade cerebral teve influência no dano ao inimigo?



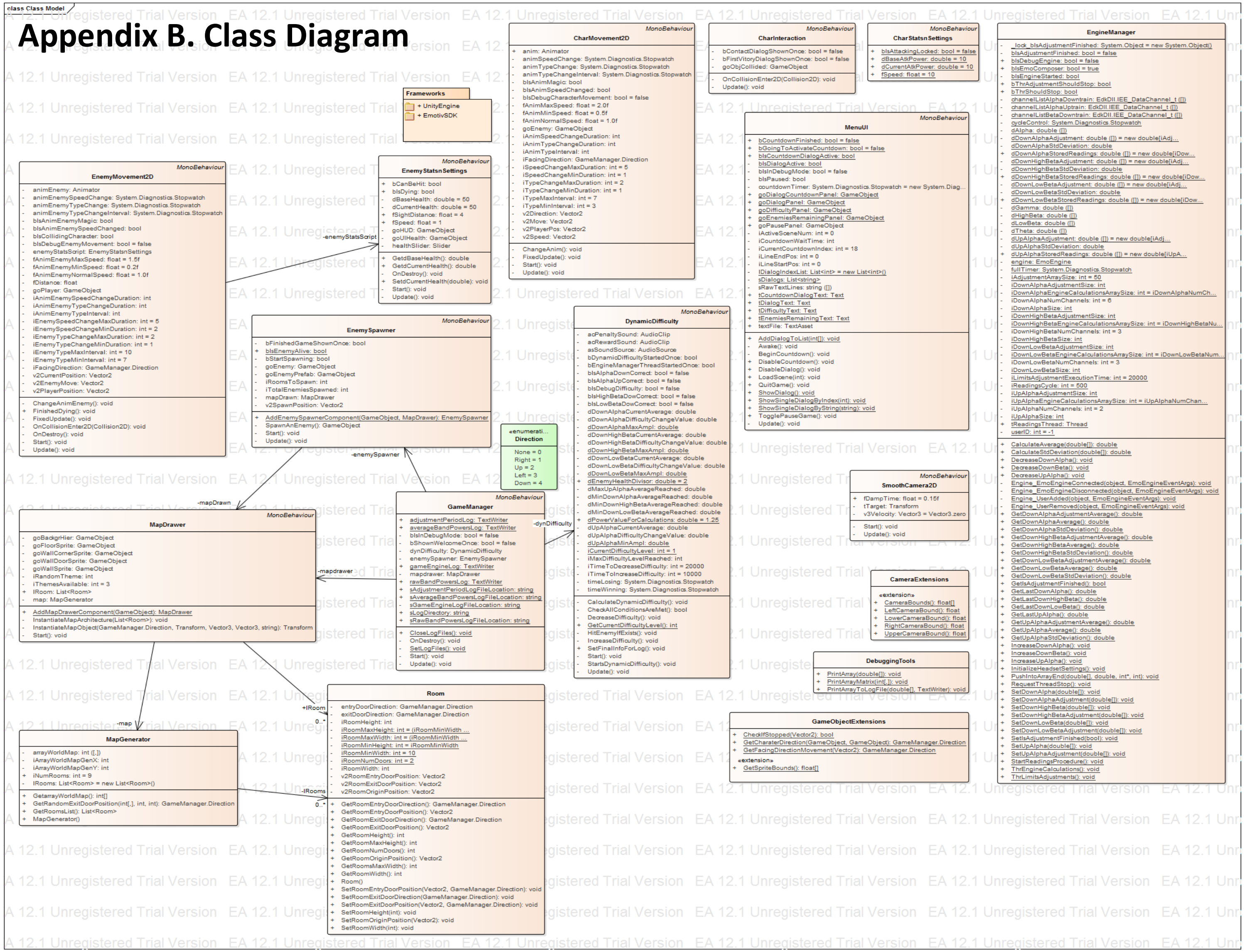
Sim	**6**	66.7%
Não	**3**	33.3%
Other	**0**	0%

Sentiu que o headset o perturbou?



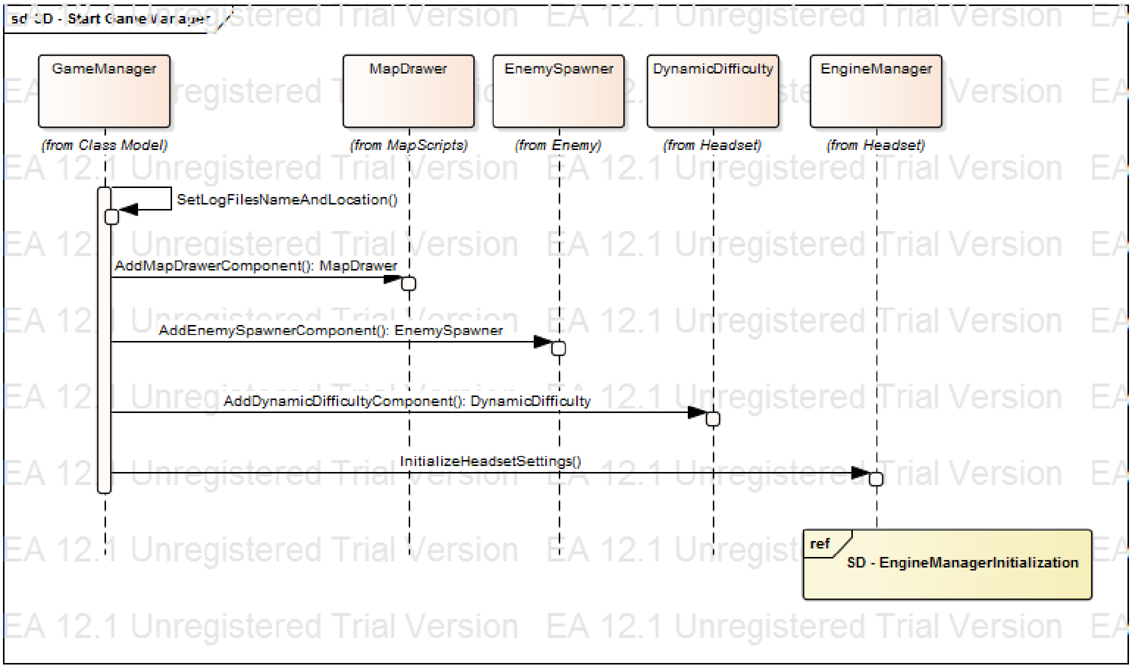
Sim	**1**	11.1%
Não	**8**	88.9%
Other	**0**	0%

Appendix B. Class Diagram

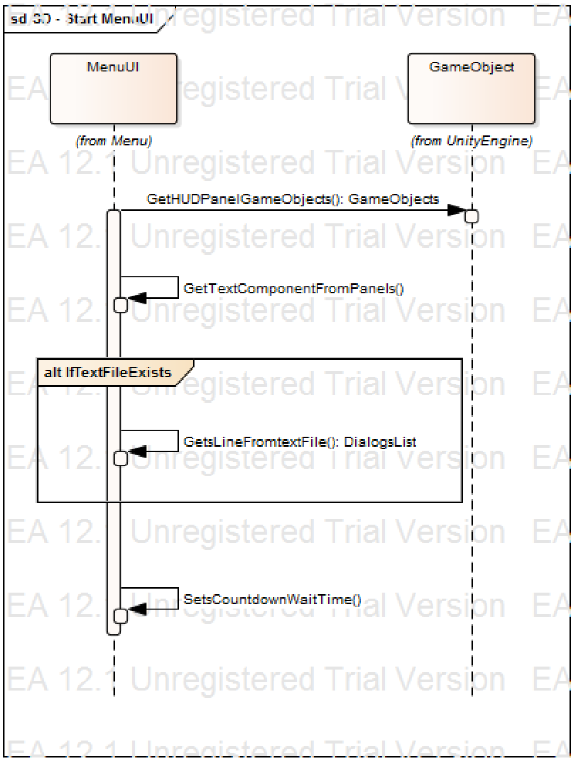


Appendix C. Sequence Diagrams

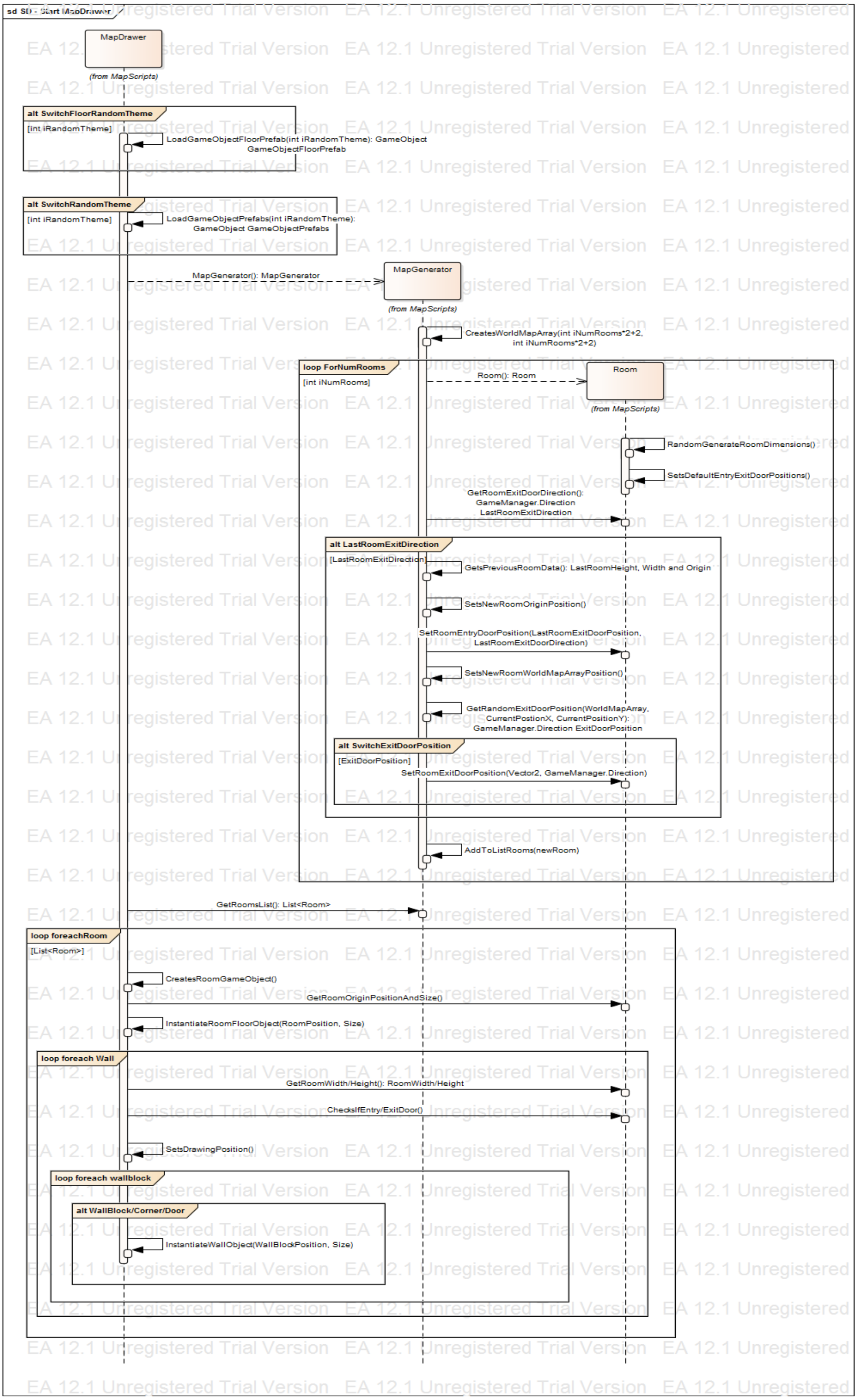
C.1. *GameManager Start*



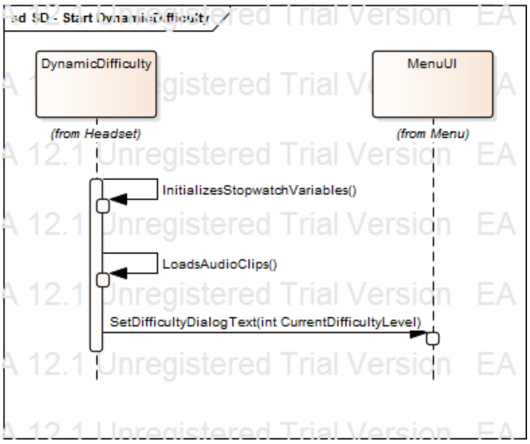
C.2. *MenuUI Start*



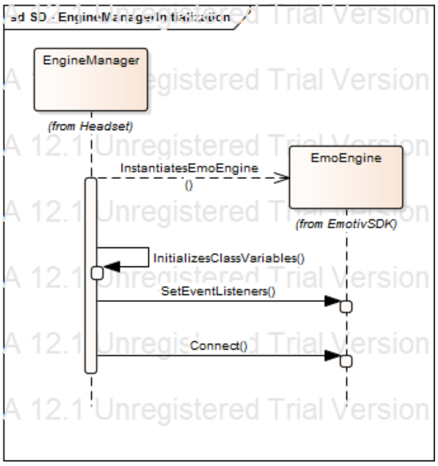
C.3. *MapDrawer Start*



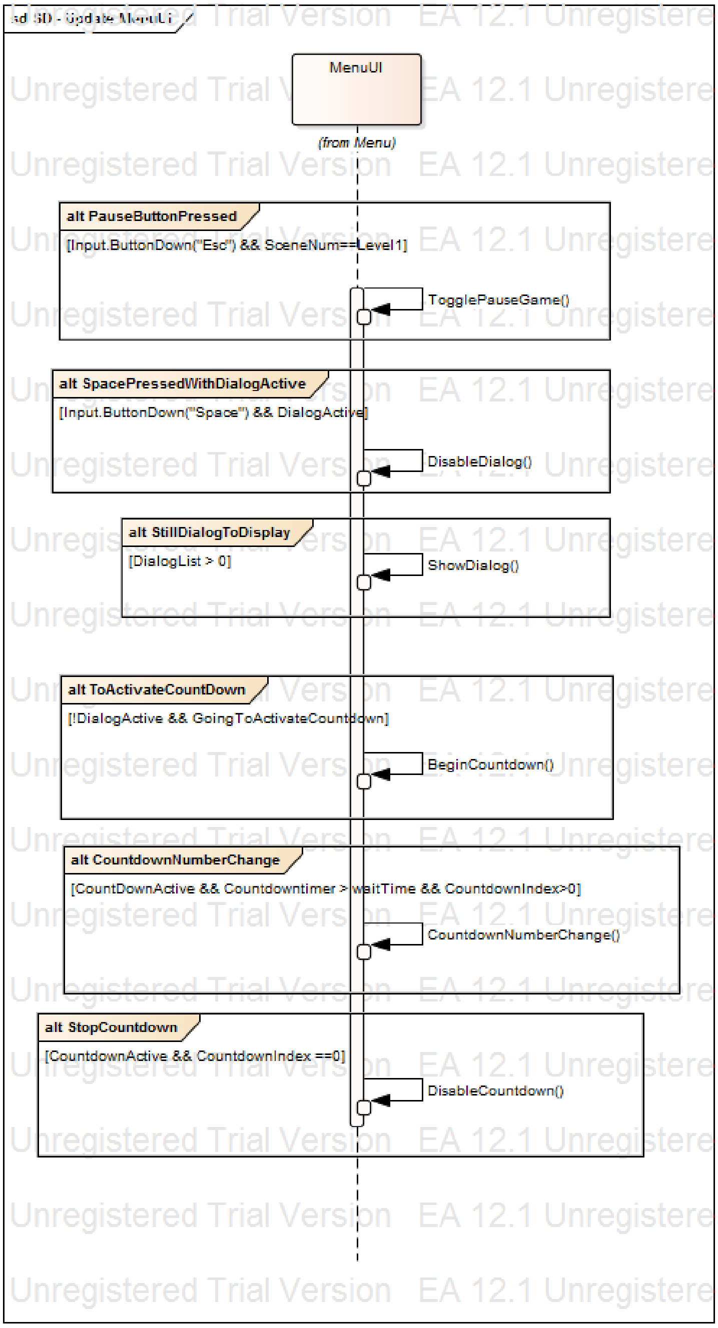
C.4. *DynamicDifficulty Start*



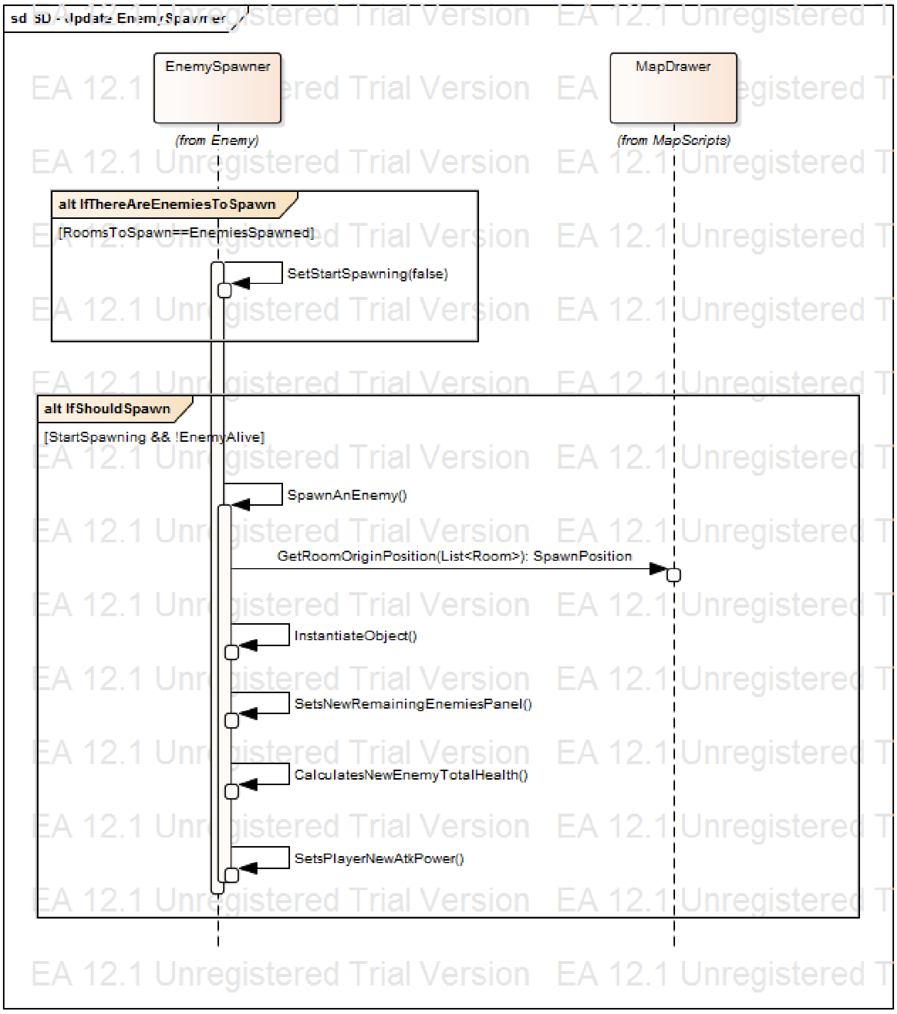
C.5. *EngineManager* Initialization



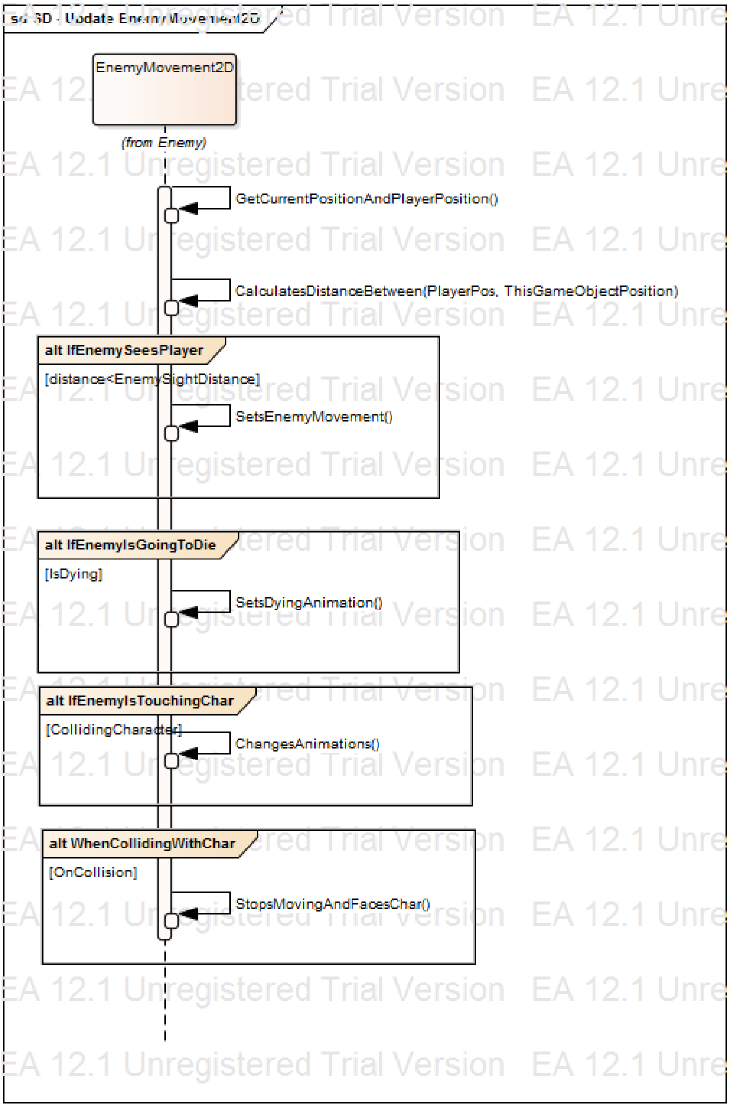
C.6. *MenuUI Update*



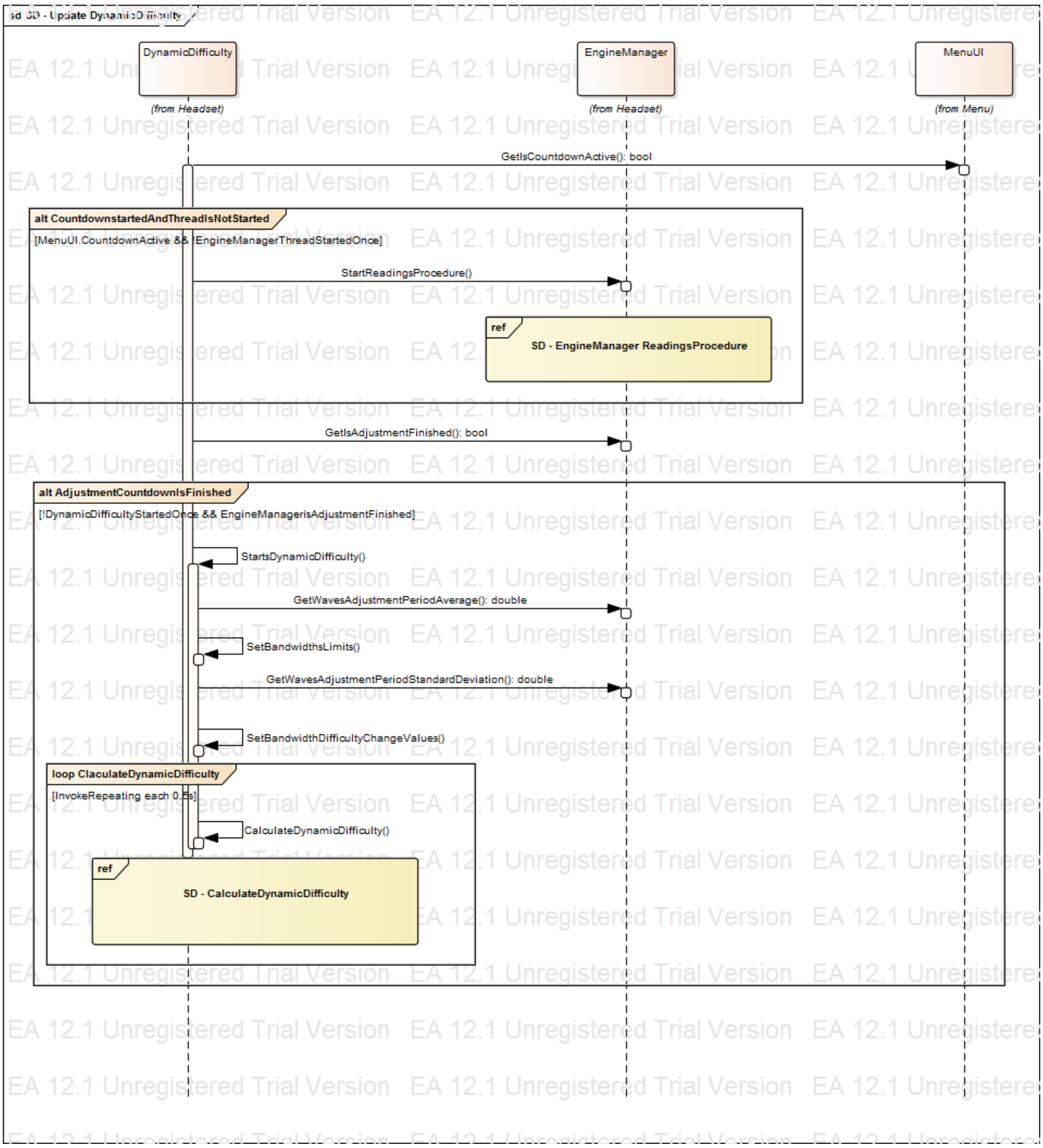
C.7. *EnemySpawner Update*



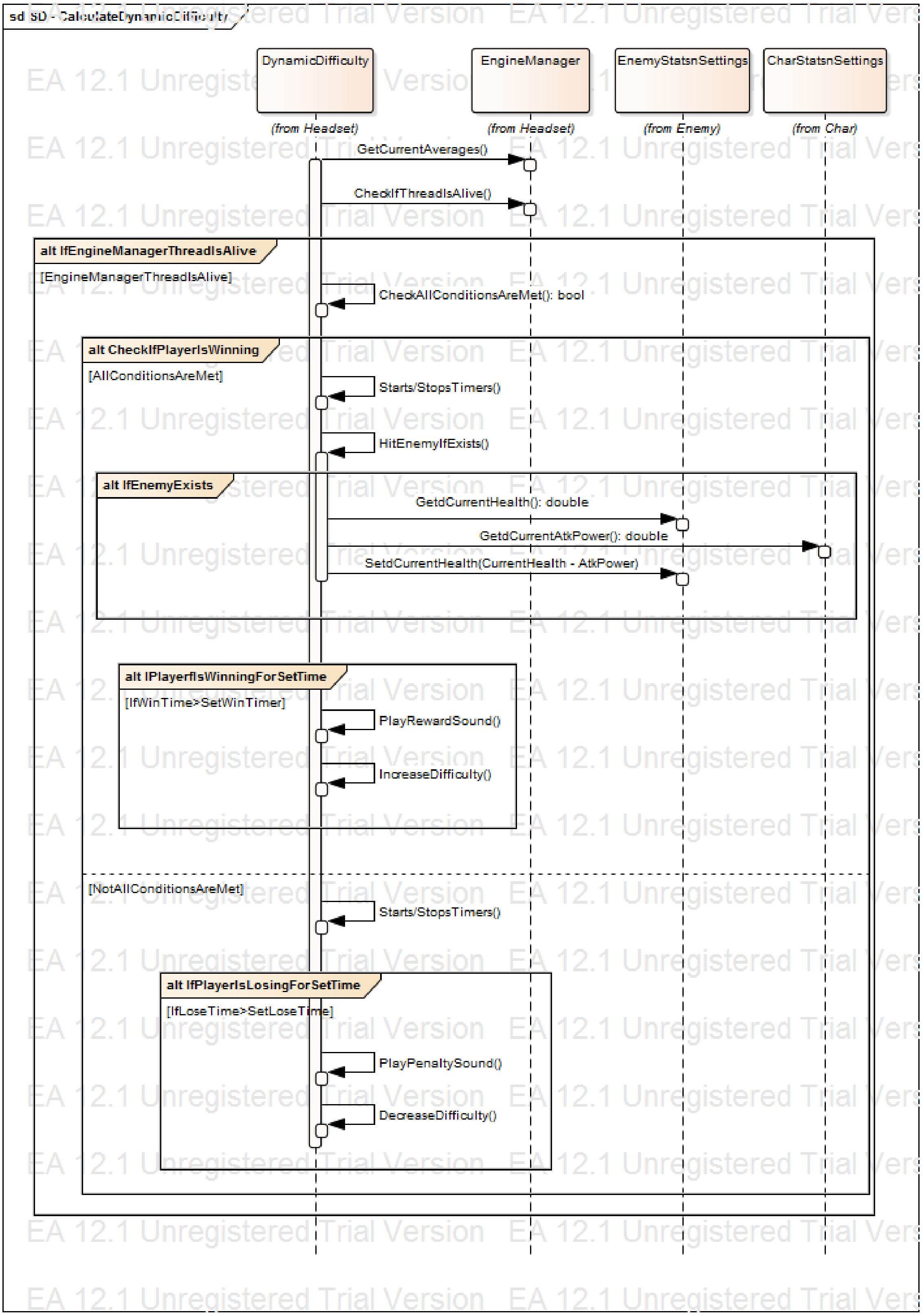
C.8. *EnemyMovement2D Update*



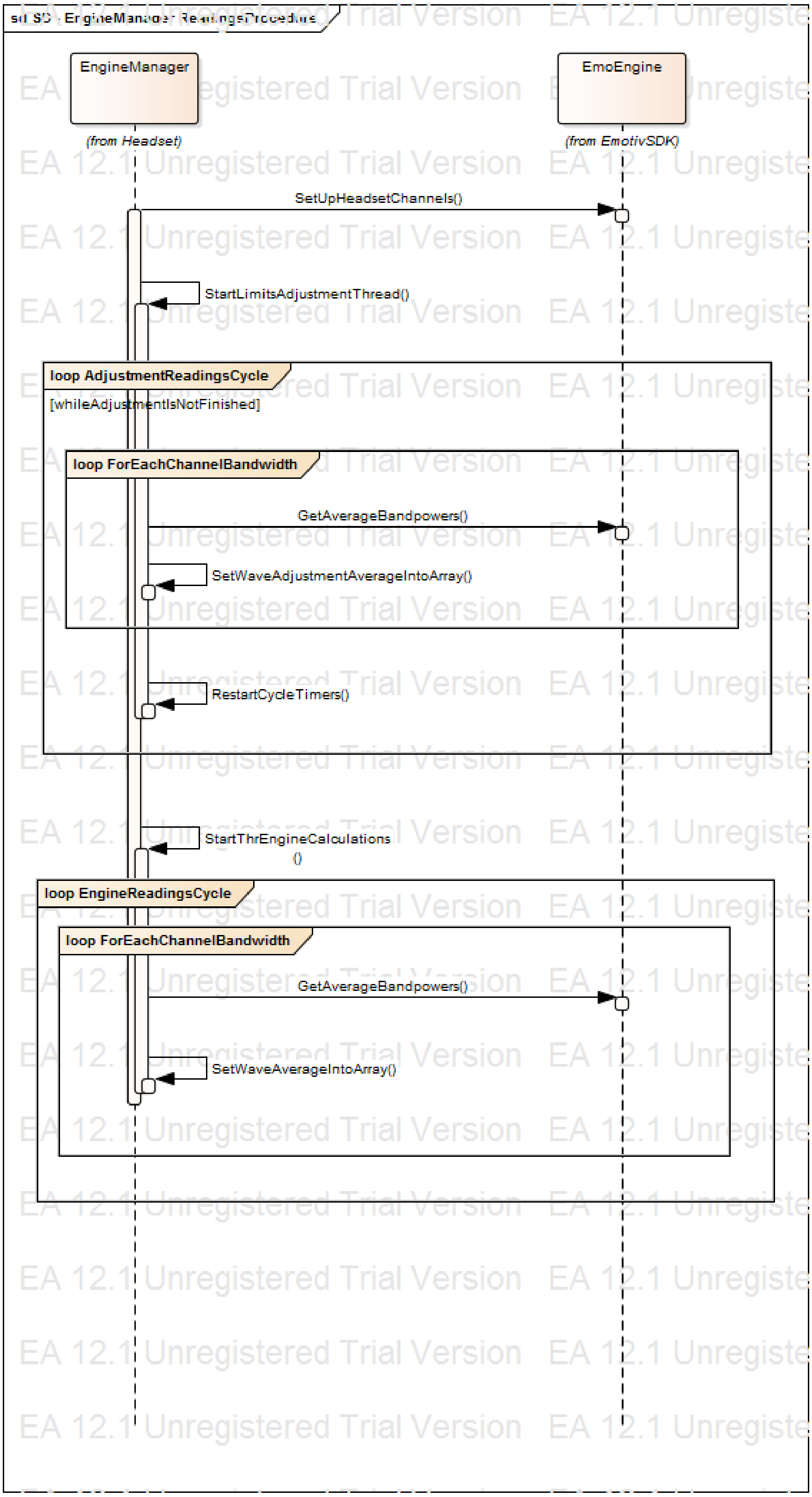
C.9. *DynamicDifficulty Update*



C.10. *CalculateDynamicDifficulty*



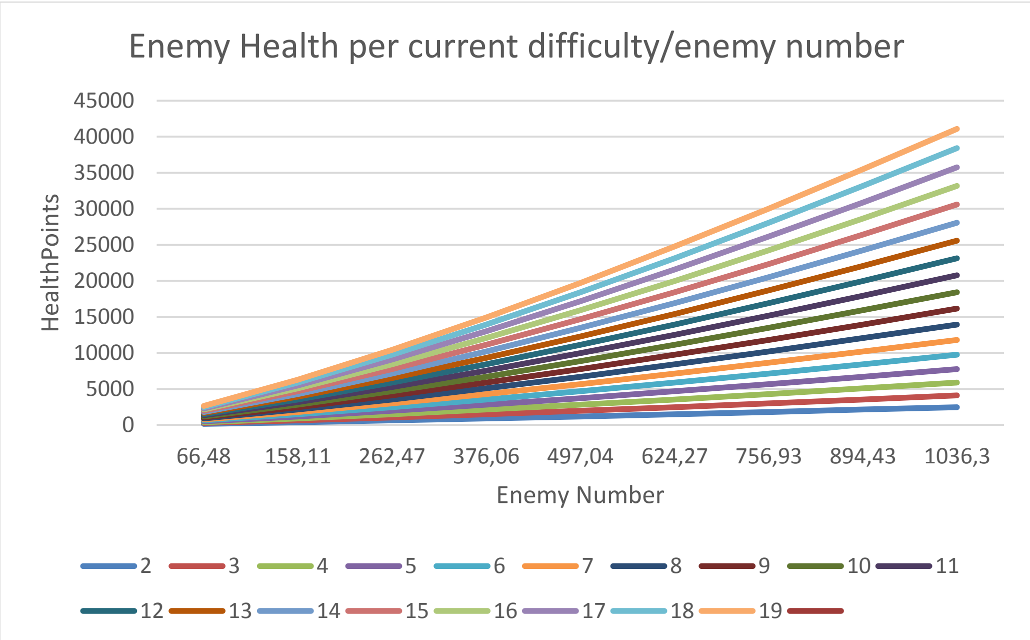
C.11. *EngineManager ReadingsProcedure*



Appendix D. Enemy Health Points Evolution Table

Legend: **Green** - Current difficulty level; **Blue** - Current enemy

	1	2	3	4	5	6	7	8	9	AtkPower
1	66	158	262	376	497	624	757	894	1036	12,5
2	158	376	624	894	1182	1485	1800	2127	2465	25
3	262	624	1036	1485	1962	2465	2989	3531	4092	37,5
4	376	894	1485	2127	2812	3531	4282	5060	5862	50
5	497	1182	1962	2812	3716	4668	5659	6687	7748	62,5
6	624	1485	2465	3531	4668	5862	7108	8399	9731	75
7	757	1800	2989	4282	5659	7108	8618	10184	11799	87,5
8	894	2127	3531	5060	6687	8399	10184	12034	13943	100
9	1036	2465	4092	5862	7748	9731	11799	13943	16154	112,5
10	1182	2812	4668	6687	8839	11101	13460	15905	18428	125
11	1332	3167	5258	7534	9957	12506	15163	17918	20760	137,5
12	1485	3531	5862	8399	11101	13943	16906	19977	23145	150
13	1641	3903	6479	9283	12269	15410	18685	22079	25581	162,5
14	1800	4282	7108	10184	13460	16906	20498	24222	28064	175
15	1962	4668	7748	11101	14673	18428	22344	26403	30591	187,5
16	2127	5060	8399	12034	15905	19977	24222	28622	33162	200
17	2295	5458	9060	12981	17158	21549	26129	30875	35772	212,5
18	2465	5862	9731	13943	18428	23145	28064	33162	38422	225
19	2637	6272	10412	14918	19717	24764	30026	35480	41108	237,5
20	2812	6687	11101	15905	21022	26403	32014	37830	43830	250



Appendix E. QEF Table

Legend: **Blue** – Input of the developer; **Grey** – Not implemented

q	D	Qi	Dimension	Q_j	p_{ij} (peso do Factor *j* na Dim *i)* [0,1]	Factor	pr_{jk} (peso do Requisito *k* no Factor *j*) {2, 4, 6, 8, 10}	Requirements	pc_k % de cumprimento do Requisito *k*) [0,100]
77%	0,51	70,80	**Pedagogic**	72,54	0,59	Learning	10	PL01- The goal is well defined and explained early in the game	100
							10	PL02- The game promotes self-learning	50
							10	PL03- The game promotes self-assessment	50
							10	PL04- Learner assessment relates with game scoring	100
							10	PL05- Player has access to his progress	100
							10	PL06- The game promotes real-time assessment	50
							10	PL07- Learner gets direct, intuitive feedback about performance	50
							10	PL08- Learner gets rewarded for pretended behaviour	100
							10	PL09- The game adjusts difficulty correctly to the player performance	100
							10	PL10- The player learns to better control his anxiety by the end of the session	25
							8	PL11- The player progresses through multiple sessions	100
							8	PL12- Game has an appropriate duration	100
							6	PL13- Game presents helpful message in case player is not progressing as expected	0
				68,29	0,41	Motivational	6	PM01- Items presented contribute to motivate user	75
							10	PM02- Levels presented contribute to avoid repetition and motivate the user	75
							8	PM03- The game experience is original	100
							10	PM04- Audio enhances gameplay	75
							10	PM05- Video enhances gameplay	75
							10	PM06- The game is simple and avoids distractions	0
							8	PM07- Graphics and pictures enhances gameplay	75
							10	PM08- The game is fun to play	50
							10	PM09- The game is challenging	100

q	D	Q_i	Dimension	Q_j	p_{ij} (peso do Factor *j* na Dim *i*) [0,1]	Factor	pr_{jk} (peso do Requisito *k* no Factor *j*) {2, 4, 6, 8, 10}	Requirements	pc_k % de cumprimento do Requisito *k*) [0,100]
77%	0,51	79,56	**Ergonomic**	100,00	0,24	Content Quality	8	EC01- Game language is adequate for target audience	100
							10	EC02- Game presents no offensive content or stereotypes in terms of gender, race, religion or culture	100
							10	EC03- Written content is free of grammatical and syntactical errors.	100
							6	EC04- Speed of communication between player and game is adequate	100
				58,93	0,41	Navigation	10	EN01- It is possible to exit the game anytime	50
							10	EN02- It is possible to pause anytime during gameplay	100
							2	EN03- Easy to switch languages from the menu	0
							10	EN04- Help button provided	0
							8	EN05- Pause button is provided during gameplay	100
							6	EN06- It is possible to personalize the player character before playing	0
							10	EN07- User interface is quick with progress information whenever relevant	100
				90,00	0,35	Adaptability	2	EA01- A session level selection is available	0
							8	EA02- Tutorial with good explanation is provided at the beginning	100
							2	EA03- Available in two languages	0
							8	EA04- Tutorial shows a clear goal	100
							10	EA05- Game instructions are clear, accurate and concise	100
							10	EA06- The player can use the game without manual	100

q	D	Q_i	Dimension	Q_j	p_{ij} (peso do Factor *j* na Dim *i*) [0,1]	Factor	pr_{jk} (peso do Requisito *k* no Factor *j*) {2, 4, 6, 8, 10}	Requirements	pc_k % de cumprimento do Requisito *k*) [0,100]
77%	0,51	62,87	**Technical**	40,63	0,38	Deployment	8	TD01- Installation is easy and simple	0
							8	TD02- There is a clear and simple entry point to the game	0
							6	TD03- A quick start is available	75
							10	TD04- The game is stable and executes systematically without failures	100
							6	TD05- The game can be easily updated	0
							10	TD06- The game immediately gives message in case of equipment malfunction	50
				76,22	0,63	User Interaction	2	TU01- Control reassignment available through the menu	0
							10	TU02- Theme and room positioning is automatically generated	100
							6	TU03- Interaction with the menu is intuitive	50
							10	TU04- All menu present the same design experience	100
							10	TU05- All menu present the same navigation experience	100
							8	TU06- The player controls the game actions	50
							6	TU07- Game menus are easily accessible	50
							10	TU08- Characters in the game behave as player expects	75
							10	TU09- Player was able to interact with the game through the headset readings	75
							10	TU10- All actions must be made by user interaction. Automatic actions must not exist	75