Escola Tècnica Superior d'Enginyeria ETSE-UV

MULTIMEDIA ENGINEERING DEGREE



DEGREE FINAL PROJECT

SLUMBER
FIRST-PERSON STRATEGY AND DECISION-MAKING VIDEO GAME

AUTHOR:
YASMINE GONZÁLEZ BIBED

TUTOR:
MANUEL PÉREZ AIXENDRI

SEPTEMBER, 2020

SLUMBER

Yasmine González Bibed

2020

Table of contents

1. Abstract 4
2. Introduction 5
3. Motivation and goals 6
4. State of the art 7
5. Planning and specification 20
5.1. System specification 20
5.2. Work methodology 21
5.3. Planning 23
5.4. Cost estimation 25
5.5. Feasibility study 30
5.6. Risks 35
6. Project development 37
6.1. Conceptual 37
6.1.1. Character background and world backstory 37
6.1.2. Requirements 38
6.1.3. Use case diagram 40
6.2. Analysis 42
6.2.1. Static point of view 42
6.2.2. Dynamic point of view 44
6.2.2.1. Interaction diagrams 44
6.2.2.2. State diagrams 45
6.2.2.3. Activity diagram 51
6.3. Design 54
6.3.1. Updated class diagram 54
6.3.2. Use case sequence diagrams 57
6.4. Implementation 64
6.4.1. Introduction 64
6.4.2. Main and Game Over screens 65
6.4.3. Common prefabs 68
6.4.4. Gameplay levels 73

7. Testing and results..... 93
7.1. Testing..... 93
7.2. Results 96
8. Outcome and conclusion. Future work..... 107
9. Annex 1. Video game storyboard..... 108
10. Annex 2. Video game script..... 125
11. References 147

1. Abstract

Since the 1970s, video games have become a humongous part of our lives, and as time passes by the possibilities that they allow are becoming bigger and bigger. The increasingly technology advancements give extremely valuable tools to developers that, combined with a solid engineering technical knowledge and an interesting concept, can end up not only being a form of entertainment, but a way to look within oneself and learn in the process.

In this sense, *Slumber* is a first-person strategy and decision-making video game develop for PC that aims to challenge the player in an adventure that requires him to be focused and survive, all while questioning his own personal values. This is presented in a video game that contains several levels in which the player has to advance through multiple events where he has to make decisions that will be key to the progression of the story.

This video game was planned and based on an original concept, to be later developed using a video game engine and all of the acquired multimedia engineering knowledge. After analyzing the results from the users who have played the game, an analysis regarding the players' decisions based upon relevant demographical statistics was carried out.

2. Introduction

In this document is presented all the relevant information related to the development and player analysis of *Slumber*, a first-person strategy and decision-making video game. This video game intends to challenge the player in many ways, both trying to figure out the way to survive as well as having to make decisions that question the player's own personal values. The video game has been developed using Unity 2019.2.2.f1 and has been given out to many people to play it once it was done. After getting all of the results from the people who have played it, conclusions have been drawn out and analyzed.

First, we will explain the concept behind the video game and how the idea came together. After that, we will analyze the state of the art of the video games and the video game industry, both technology and popularity wise.

Furthermore, we will describe the planning and specification of the video game development. In this sense, we will understand the specifications that for the video game, as well as the planning done before starting the development and how it ended up reflecting in reality. We will also describe the cost estimation of what the game would cost if a video game company were to make the video game, having in mind that it has actually been developed by myself with no other investment other than time and the cost of access to Internet. What is more, we will also look at the feasibility behind the video game and the risks of its development if it was a video game company that wanted to develop the video game.

After that, we will analyze the development of the video game in its four phases of development: conceptual, analysis, design and implementation. The background story and the backstory for the world will also be explained concisely. We will use UML diagrams to have a clear understanding of the video game as a system, and lastly, we will go over numerous details regarding the video game's implementation in Unity.

Lastly, we will examine the testing that has been done for the video game, as well as the results drawn out by the users who have played the game. We will also extract conclusions out of the entire development of the game as well as the results analyzed and see how the video game could possibly be used in the future.

The documents that contain the video game script and the video game storyboard have also been annexed to the document, as well as the reference used for this document.

3. Motivation and goals

Since I was a kid, I have always loved video games and, in fact, it was wanting to know how to create video games what led me to enroll in the Multimedia Engineering degree, as it was one of the optional subjects in the last year course. As such, I decided that it would be a great tribute to this fact making a more complex video game entirely by myself.

I wanted to use both of the engineering and more technical knowledge I have acquired during these past four years in the Multimedia Engineering degree, along with my own creativity and passion to give the best of me and create something I can be proud about. I also wanted to fully understand the process of making a video game, so the best way to do so was to create a video game myself.

When it came to what kind of video game I wanted to make, I knew that I did not want to create a video game that was merely based on entertainment, but that it acted kind of as a challenge to the player himself. Because of this, I came up with the concept behind *Slumber,* which is a video game that challenges the player's own personal values and requires him to go through a mental process to realize the truth underneath it all. I also wanted to draw out some conclusions upon all of the players' answers after having played the game to extract the general behavior of the players, based on a demographic statistical analysis.

Another relevant aspect that motivated me to develop this video game, was the wish to explore as much as I could the different possibilities when it came to the development using the Unity game engine. I wanted to see how all of the different mechanics and tools offered by Unity would allow me to make the videogame I had planned beforehand, looking at it from both an engineering and developer perspectives.

4. State of the art

Video games have been in our lives for quite a long time now. The videogame industry started to take off in the early 70's with home consoles and arcade videogames. However, since its beginning, the videogame industry has become an incredibly profitable entertainment market with a revenue of almost $135 billion in 2018[1], which represents a 10.9% growth from the previous year. Not only it has become one of the most profitable industries in the world, surpassing Hollywood's revenue[1], but it has come to be the leading form of entertainment today.

Even though video games are found in homes worldwide today, they actually got their start in the research laboratories of scientists. The first recognized example of a game machine was unveiled by Dr. Edward Uhler Condon at the New York World's Fair in 1940. The game, based on the ancient mathematical game of Nim, was played by about 50,000 people during the six months it was on display, with the computer reportedly winning more than 90 percent of the times.[2] However, the first game system designed for commercial home use did not emerge until nearly three decades later, when Ralph Baer and his team released the "Brown Box" in 1967.[2] The "Brown Box" was a vacuum tube-circuit that could be connected to a television set and allowed two users to control cubes that chased each other on the screen. It could be programmed to play a variety of games, including ping pong, checkers and four sports games.



Figure 1. Ralph Baer's Brown Box. National Museum of American History

[1] How Much Is the Gaming Industry Worth? https://techjury.net/stats-about/gaming-industry-worth/
[2] The History Of Gaming: An Evolving Community https://techcrunch.com/2015/10/31/the-history-of-gaming-an-evolving-community/

However, the “Brown Box” did not have a great success but it definitely set the stage up for its successors. In 1972, Atari became the first gaming company to really set the benchmark for a large-scale gamming community, with the release of Pong, which became the first real electronic video game. With these video games, gaming in arcades became really popular among young people. In addition, home gaming was still considered to be a privilege by the mid 70’s, but by the late 70’s video game consoles had already blown up and became really popular.

With the birth of Personal Computers (PC) came along a new stage for gaming. Early computers such as the Macintosh allowed users to co create their own game using code, as well to connect their devices with other players and be able to play together. However, the real revolution in gaming came when LAN networks, and later the Internet, opened up a way to multiplayer gaming, allowing gamers to compete and interact from different computers. This was a key step which set the stage for the large-scale interactive gaming that modern gamers currently enjoy.



Figure 2. Variety of gaming devices. images.collectorz.com

Between 1993 and 1996, Sega, Nintendo and Atari made a number of attempts to break into online gaming by using cable providers, but it wasn’t until the release of the Sega Dreamcast, the world’s first Internet-ready console in 2000, that real advances were made in online gaming as we know it today. Nevertheless, the console wasn’t successful, but it paved the way for the next generation of consoles, such as the Xbox, released in the mid-2000s. Since the early 2000s. Internet capabilities along with computer processor technology have improved at such a fast rate that every batch of games, graphics and consoles seem to be incredibly more powerful than the previous. In this moment, online gaming has become an indispensable part of the gaming community.

Since smartphones and app stores hit the market in 2007, gaming has undergone another unbelievable rapid evolution that has brought gaming into the mainstream pop culture. This huge shift in the gaming industry has not only widened gaming demographics but also pushed gaming to the forefront of media attention. As well as the rise of smartphones as a gaming device, PCs are starting to take over video game consoles as a preferred device to play heavier games, especially with the growth of digital distribution platforms, such as Steam.[3]

Video game players include a wide and diverse cross-section of audience of every age, gender and ethnicity, from kids to adults, to this day there are basically

[3] Steam https://store.steampowered.com/

videogames of every genre. According to the Entertainment Software Association, the average gamer is 35 years old and has been playing for around 13 years. In fact, and specially with the rise of smartphones and mobile gaming that allow access to casual and puzzle games, 65% of adults in the United States of America play videogames, being smartphones the most common way of doing so.[4]

The reason of video games' success among an audience of all ages relies on the incredibly humongous number of genres that can be found amongst video games. Along with the wide variety of genres (arcade, first-person shooters, platformers…), the flexibility of the device where they can be played also enhances its success. Making everything portable is certainly trendy these days, and the gaming industry has been keeping up with this trend. Even before smartphones hit the scenes, the gaming consoles themselves were made portable with the examples of systems like Nintendo DS, PlayStation Vita and many others. Video games can be played on home video game consoles, smartphones, PCs or gaming portable devices, which perfectly adapts to every person's needs.

Not only there are video games of every genre and subgenre, but there is also a lot of variety when it comes to storytelling. The narrative of videogames needs to find a balance between giving the player interactivity and constructing a solid plot that engages the player at the same time. In this area, Michael Mateas and Andrew Stern, creators of the awarded[5] 2005 video game Façade, cite in their work *Interaction and Narrative:*

"…game designers also experience a deep ambivalence. The ephemeral quality of gameplay, the experience of manipulating elements within a responsive, rule-driven world, is still the raison d'être of games, perhaps the primary phenomenological feature that uniquely identifies the computer game as a medium. Where gameplay is all about interactivity, narrative is all about predestination."

Mateas and Stern. Interaction and Narrative

When it comes to narration in video games, it is necessary to include elements that provide an explanatory background of the story and characters. These elements can be introduced to the player via embedded narrative or emergent narrative. The former includes pre-generated narrative content that exists prior to a player's interaction with the game, such as cutscenes and backstory, while the latter arises from the interaction of the player with the game and varies from each player. Game design involves employing and balancing the use of both embedded and emergent narrative. However, there are many ways a story can be told in video games, depending on the gameplay itself.

[4] 2019 Essential Facts About the Computer and Video Game Industry https://www.theesa.com/esa-research/2019-essential-facts-about-the-computer-and-video-game-industry/

[5] Mott, Tony (2010). 1001 Video Games You Must Play Before You Die. London: Quintessence Editions Ltd. p. 615. ISBN 978-1-74173-076-0.

Because of the nature of video games, which combines player interactivity and executing a story with a plot, we are able to find video games with nonlinear or linear gameplay. A video game with a linear gameplay presents a number of events to the player that must be cleared in the order they are presented, whereas in a nonlinear gameplay the player can complete them in a different order. A nonlinear video game reaches another level of interactivity, giving the player a bigger freedom and also allowing the inclusion of different paths to victory, different types of victory or optional subevents. In this regard, it is adequate to cite video games with multiple endings, which are video games that use linear story lines where the player faces different endings depending on his actions.

This kind of gameplay provides a deep focus on storytelling, rather than a gameplay driven by skill leveling. Instead, players delve profoundly into the many different ways a journey could unfold, telling a story with many different branches. This kind of videogames can be categorized into a “Choose Your Own Adventure” genre, which merges unites this idea with other subgenres like sandbox, action-adventure, role-playing, strategy…

Slumber falls into the category of this interactive “Choose Your Own Adventure” genre. In order to have a clear vision of the nature of the video game, it is relevant to briefly take a look at some video games that have similar feeling to them.

As for the first example, we are going to take a look at the 2018 adventure video game *Detroit: Become Human* developed by Quantic Dream (the same studio which has released well-known titles such as *Fahrenheit, Heavy Rain* and *BEYOND: Two Souls)* released for PC and PlayStation 4. This video game is a futuristic neo-noir thriller game from a third-person which is subject to a set and controllable perspective.[6] The premise of the video game sets the player in Detroit during the year 2038 where the city has been revitalized by the invention and introduction of androids into everyday life. However, when androids start behaving as if they are alive, events begin to spin out of control. The game takes the player into an emotional journey with choices that must be taken in order to achieve their ultimate goal, which can be defined in different ways depending on the values of the player. The plot of the game deals with a variety of mature themes that explore moral ground and each player decision affects what will happen.



Figure 5. PS4 cover for Detroit: Become Human

[6] Dayus, Oscar (19 June 2017). "How Real-World Issues, Including Terrorism, Affected Detroit: Become Human's Story". GameSpot.

The video game centers on three playable character who can die as the story continues without them, and using quick time events and dialogue decisions the story branches out depending on which choices the player makes. These choices can be



Figure 4. Screenshot of the gameplay of Detroit: Become Human. Don't Feed the Gamers

viewed in a flowchart during and immediately after a given chapter ad the player can rewind to certain points in the story to reshape decisions in the event of regret. Certain scenes feature countdowns, which force the payer to think and act quickly, as well as obtaining clues by highlighting and analyzing the environment. David Cage, the creator of the video game says the following[6] about the player's choices and how they affect the story itself:

"…it's not this kind of game where you get three different endings and that's it--there are many paths, many ways of playing the story, of traveling through this tree structure, leading to many different endings, but the goal for us was to give the feeling to the player that they are in control of their destiny, that they are telling their own story."

David Cage

Detroit: Become Human had a development budget of €30 million[7] and is based on Quantic Dream's refined technology, with performance capture of more than 250 actors for 513 roles, 35,000 camera shots, 74,000 unique animations, 5.1 million lines of code, and a script of between 2,000 and 3,000 pages.[8] Despite being on a different level of complexity and length due to the resources employed, the idea where *Slumber* falls is the same as the highly-acclaimed *Detroit: Becoming Human*, and thus give us a clear idea of the interaction of the player with the game and how his choices affect it.

[7] Audureau, William (14 January 2018). "Quantic Dream, un fleuron du jeu vidéo français aux méthodes de management contestées". Le Monde.
[8] Parker, Laura (16 February 2018). "A Videogame Developer Who Finds Power in Pathos". Wired.

Another video game that is worth taking a look to is *Gemini Rue*. This video game was released in 2011 for PC and was later ported to iOS and Android in 2013 and 2014, respectively, due to its success and great reception. The game is a cyberpunk adventure game developed entirely by Joshua Nuernberger, and published by Wadjet Eye Games.[9] The plot of the video game is a story about two tortured souls in a neo-noir future : Azriel Odin, a police officer with a dark past looking for a friend who has disappeared, and Delta-Six, a resident of a center facility who has had his memory erased and is undergoing weapon training.



Figure 6. PC cover for Gemini Rue

The player is free to switch between these two characters at any time, and the stories of the dual heroes converge and the game plunges into a memorable ending. Its gameplay is based on giving the player the ability to interact with a 2D environment and NPCs to solve puzzles and communicate with characters. In this sense, both *Gemini Rue* and *Slumber* are really alike, because both of them require interaction from the player with the environment in order to compile useful information to advance through the game.

Gemini Rue has been greatly acclaimed but many professionals of the video game industry, and amongst them, the magazine Wired declared the following:

"A dark, captivating story that evokes memories of Blade Runner and Shadowrun, with a thrilling final sequence and quite a few disturbing moments."

- Wired



Figure 7. Screenshot of Gemini Rue. Adventure gamers.

[9] "Gemini Rue". Wadjeteyegames.com. 2011-02-24.

One of the subgenres of video games that rely on the interactivity in the player in the same way that *Slumber* does are visual novels. Visual novels are a type of video game originated in Japan in the 90's, that features text-based story with narrative style of literature and interactivity along with static or sprite-based visuals, background music and voice acting. The biggest gameplay aspect of a visual novel is making choices, in a way that the player has to decide what the main character does in the form of options. The options can lead the player to different story plotlines, called routes, or they could lead the player to a bad end, which forces the player to try again. Visual novels involve many genres, such as fantasy, romance, horror, and action.

While this video game genre lacks in active gameplay elements, it is solely focused on storytelling and allows the player to fully engage himself in the plot of the game. Because of this, visual novels tend to have amazing storylines and plots. The main similarities the visual novel genre has with *Slumber* are the important role of decision making to achieve one of the possible endings of the story, as well as the fact that text is used for communication within the characters and the player. However, S*lumber* brings the player more interaction when it comes to the gameplay itself.

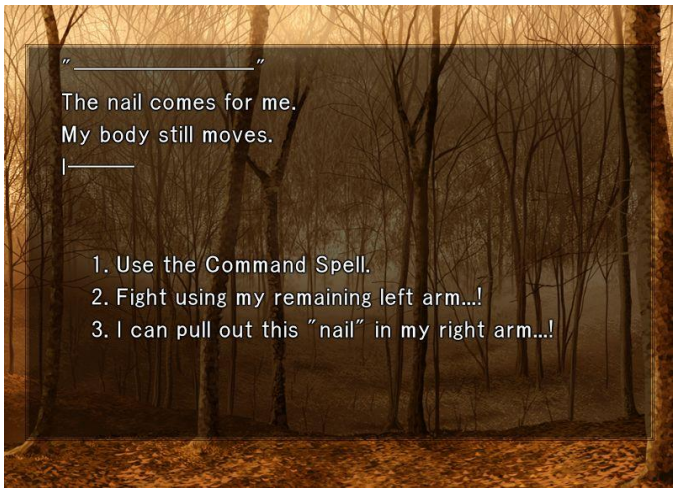


Figure 7. Screenshot from the 2004 visual novel Fate/Stay Night. Type-Moon.

In order to have a better understanding of the genre, it is convenient to review one of the most popular and representative visual novel, that will help to have a better understanding of the relation between the genre and *Slumber*. The game we are going to take a look at is called *Fate/Stay Night*, which is an eroge visual novel created by Type-Moon that was released in 2004. The plot circles around a young male named Emiya Shirou who becomes a participant in a battle called the Holy Grail War where seven chosen persons (masters) fight along historical heroes (servants) for a chance to obtain the Holy Grail that will grant the winner wishes. While Fate/stay night may seem like just an action story due to the premise, deep inside it is a story about grand ideals, and motivations of characters. The game offers three different routes, and each

of then differ vastly from the others giving the player more information in each route about the world where the story takes place.



Figure 8. PS Vita cover for Fate/Stay Night

The main thing that *Slumber* and visual novels like *Fate/Stay Night* have in common is the existence of bad ends. This bands ends lead to the main character to die and go back to a previous point in the story. In *Slumber*, there are many circumstances linked to the decisions the player makes that lead the player to die and restart the game. Due to the difference between the length of the visual novel, which is about 50 hours long, and *Slumber,* the player does not go back to a previous point but is taken to the main menu to start again. Another thing in common, is that both of the games push the player to explore different alternatives in order to achieve victory, survival in the case of *Slumber* and a good end in the case of the visual novel.

With that being said, *Fate/Stay Night* is one of the most successful visual novels in the market and, even though it was released in 2004 for PC, it has been ported to other platforms like PS2 in 2007 and PS Vita in 12 years after its original release. Also, the visual novel has been adapted to other media such as three anime adaptations for each of the three routes, manga and has also led the creator to release many other spin-offs and games for different platforms. Overall, the visual novel created a solid reputation and success for the creators, selling more than 751,488 copies.[10]

Contrary to common misconception, video game players are generally healthy, civically engaged, college-educated, and socially active. In fact, a lot of people find in video games a way to relieve stress and stimulate their minds, which reflect positively in their lives and general health. Not only video gaming is the main form of entertainment at the present, but since its beginning it has also been a way to provide education and instruction to the player. In that sense, video games act as an educational tool, providing an alternative to the traditional learning method. Using video games to educate and instructs allows the audience to have immerse themselves, which ultimately leads to a better understanding of the subject in particular. Another contribution to that is that video games allow storytelling and the creation of fantasy worlds, which contributes into the improvement of creativity and imagination of the player.

Video games have overpassed a massive improvement in the last two decades, thanks to the advances in technology which have allowed the development of much more realistic and intuitive video games, as well as breath-taking graphics and gameplays. The rise of artificial intelligence has fundamentally transformed the gaming experience, and it can be considered the most significant video game technology

[10] "2000-Present, Average Sales of 10,000 or More". Someanithing: Anime DVD/BD Sales in Japan.

advances of all time. Another humongous step up in gaming is the appearance of AR (Augmented-Reality) and VR (Virtual-Reality) video games, that have brought to the video gaming scene a fully immersive experience within the player. Furthermore, video games' graphics have abruptly improved in the last decade. This is due to the improvement of the hardware where the games are played. The games that came out in the 1990s and early 2000s do not have sophisticated textures, shaders or sprites that are present in today's games and, there have been significant strides made in the way these graphics are rendered on gaming platforms. Today we have the required hardware that allows for greater pixels and shading on a massive scale.

With the rise of the video game industry, there has been a great investment when it comes to tools designed for video game development. The use of these tools has reached a whole new level with the use of game engines, such as Unreal or Unity. A game engine is the software that provide game creators with the necessary set of features to build games quickly and efficiently. These frameworks are targeted for game development that supports and brings together several core areas. They allow the importation of art and assets, both in 2D and 3D, from other software such as Maya, Photoshop… and the assembly of those assets into scenes and environments.[11] They also permit the addition of lighting, audio, special effects logic, physics, animation, special effects…, a wide variety of tools in order to give game creators a powerful tool. In our case, the game has been developed using Unity3D, which is a multi-platform game engine that allows the creation of interactive high-quality 3D content. The Unity game engine was released back in 2005, aiming to make game development more accessible to game developers.

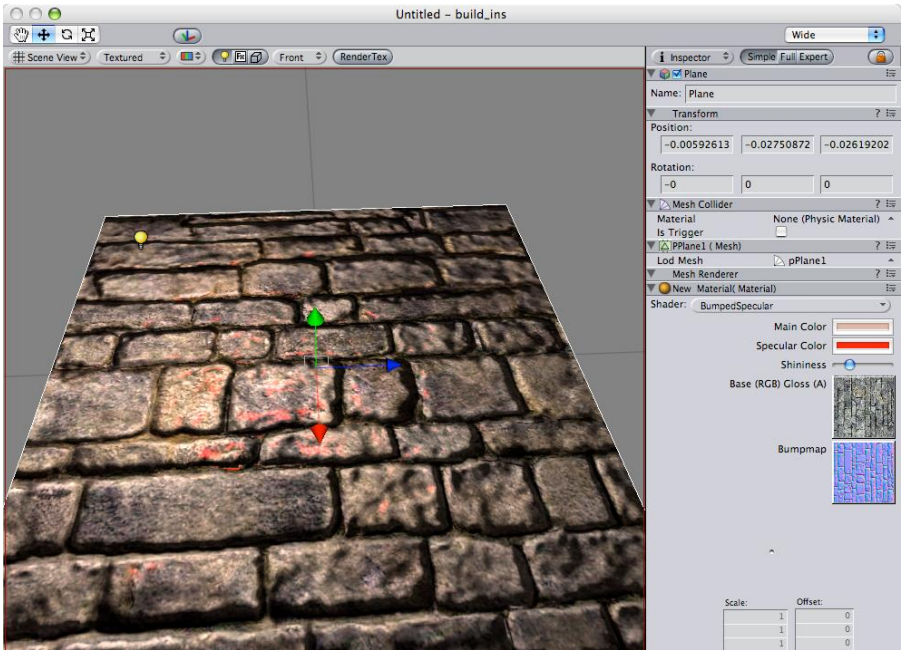


Figure 9. Unity editor in 2005

[11] "Game Engines - How Do They Work?" Unity, unity3d.com/what-is-a-game-engine.

Unity was first released for Mac OS X, later adding support for Microsoft Windows and multiple web browsers. Since its first launch, the Unity team has been constantly adding more features and improving its development environment to cover the needs of video game developers, as well as increasing the many possibilities that a video game can achieve. The version of the Unity editor that has been used for the development of the video game here described is Unity 2019.2.2f1.

Since its first released in 2005, Unity has allowed the development of many relevant video games that have both immensely popularity and quality, as well as a wide variety of genres and for all gaming platforms. Some examples of these video games are:

- Firewatch: it is a single-player first-person adventure game developed by Campo Santo and published by the developer in partnership with Panic. It was released in February 2016 for Microsoft Windows, OS X, Linux, for PlayStation 4 and Xbox One in September 2016, and for Nintendo Switch in 2018. The game received generally positive reviews, earning praise for its characters, story, dialogue and visual style. Firewatch has won the award for Best 3D Visual Experience at the Unity Awards 2016, Best Indie Game at the 2016 Golden Joystick Awards, Best Narrative at the 2017 Game Developers Choice Awards and Debut Game at the 2017 British Academy Games Awards. However, the presence of technical issues and the game's ending were both subjects of criticism. As of 2018, Firewatch has sold over 2.5 million copies across all platforms, being extremely successful for an indie video game.



Figure 10. Firewatch gameplay

- Ori and the Blind Forest: it is a platform-puzzle adventure video game developed by Moon Studios and published by Microsoft Studios. The title was released for Microsoft Windows and Xbox One on March 2015, and for Nintendo Switch on September 2019. Upon release, the game received really good critical acclaim, with players praising the game's gameplay, story, art style, action sequences, musical score, and environmental design. The developer stated that the game was profitable for the company within a few weeks after its initial launch. The video game was earned many awards, including Xbox Game of the Year at the Golden Joystick Award 2015 and Artistic Achievement at the 2016 British Academy Games Awards.

- Beat Saber: it is a virtual reality rhythm game developed and published by Beat Games. The game was released for PlayStation 4 and Microsoft Windows on May 2019, supporting most virtual reality headsets such as PlayStation VR, Oculus Quest, Oculus Rift and Windows VR headsets supporting both Steam VR and Oculus VR APIs. During its early access phase, Beat Saber received a lot of positive reviews, becoming the highest rated game on Steam less than a week after its early access release. The game had sold over a million copies by March 2019. The video game has been awarded with numerous awards, such as the Fan Favorite VR Game at the 2018 Gamer's Choice Awards, the Best VR/AR Game at the 2019 Game Developers Choice Awards, the 2019 Golden Joystic Awards, and the 2019 The Game Awards.

- Fate/Grand Order: it is an online free-to-play single-player RPG mobile game, developed by Delightworks and published by Aniplex (a Japanese anime and music production company owned by Sony Music Entertainment Japan). The game was first released in Japan on July 2015 for iOS and Android, and due to its popularity, an English and Chinese version of the game was released on June and November 2017, respectively. Fate/Grand Order grossed $982 million in 2017, making it the year's sixth highest-grossing mobile game. In 2018, Fate/Grand Order grossed $1.2 billion, making it the year's seventh highest-grossing free-to-play game. As of March 2019, the game has grossed more than $3 billion worldwide, and the total revenue of the game crossed $4 billion by the end of 2019. The video game is extremely popular, having received a total of 32 million downloads worldwide as of October 2018, including over 4 million downloads for the English version, and 7 million in September 2019.

Along with the use of the Unity game engine, some newer features and tools have been used for the making of the video game, such as Timeline, Terrain Tools and Cinemachine. While we will explore more deeply how they have exactly been used for the implementation of the game, it is relevant to briefly describe what each one of these features does and how powerful they are. Firstly, Timeline is a recent tool implemented

into the Unity development environment in 2018 that allows developers to include cutscenes during gameplay in a comfortable way.

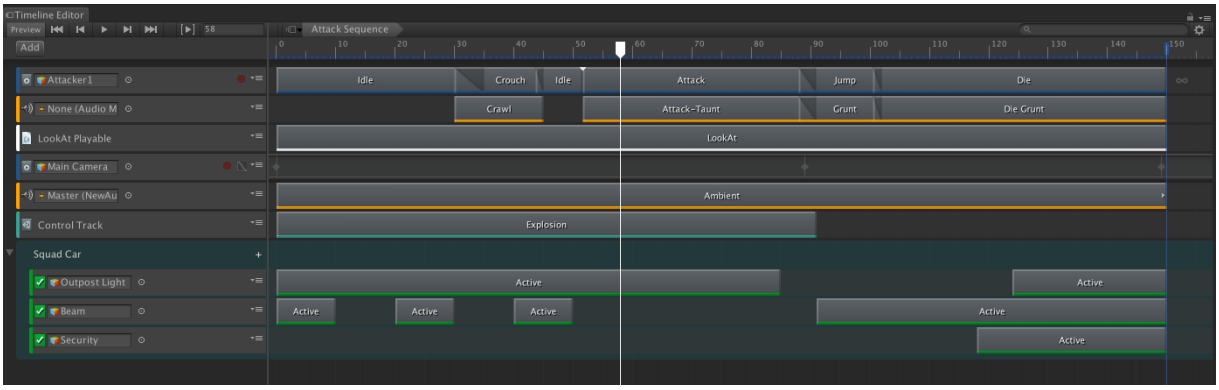
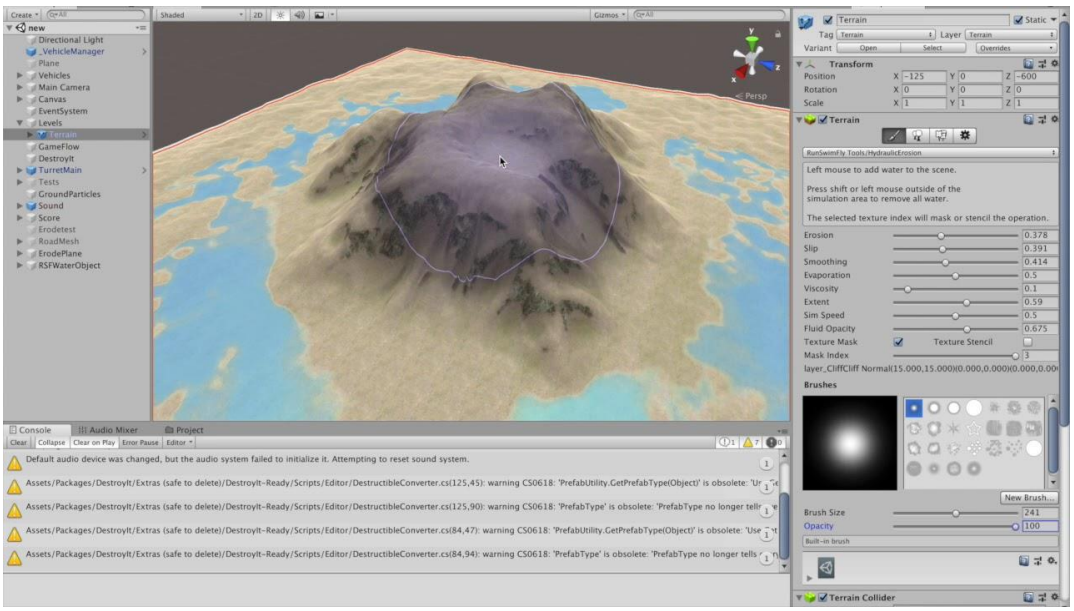


Figure 11. Timeline window in the Unity Editor

The Timeline tool appears in the form of a windows in the Unity Editor, that ca be used to create cinematic content, gameplay and audio sequences, as well as other effects, such as particle systems. This tool uses a Timeline asset, which stores the tracks, clips and recorded animations without being linked to any specific object in the scene and then the Timeline instance stores its links. For this reason, this is a powerful tool that allows to incorporate cinematics that will improve the overall player experience with the video game.

Secondly, the Terrain Tools is a new feature to Unity, that was added to the editor in 2019. This tool allows the developer to incorporate terrains as objects in the scene, that can be shaped and textured as to meet the developer needs. The terrain component provides tools to raise or lower the terrain, paint holes and textures, between other uses. This terrain is extremely useful to developers whose video game



requires of landscapes that the player has to explore, since it is easy to use and allows infinite possibilities in the making of a terrain.

At last, Cinemachine is a free package, available for any Unity project, that is composed of a set of tools that allow the developer to use dynamic and smart cameras to create cinematic camera shots, as well as camera behaviors in real-time. This tool allows the developer to integrate smart cameras in the scene, which will act in real-time accordingly to advanced behaviors, such as procedural targeting to a specific object, procedural framing, and blending between different smart cameras. This tool suite is immensely valuable for crafting beautiful and dynamic shots, that will hugely enrich the cinematic aspect of the game.

As we have seen, because of the extreme ease of video game creation that these frameworks grant creators, the amount of video games in the market is unmeasurable, which ultimately leads to a bit of an over-saturation. However, video games have become an essential part of our lives for quite some time now and they continue to evolve, grow along technology and bring us a charming feel of comfort.

5. Planning and specification

5.1. System specification

Before defining the specification of our system, we have to be aware that being the software itself a video game, its main purpose is entertainment. However, the game will not only grant entertainment to the player but will also require mental work from the player via gameplay, which will be mirrored in the outcome of the game. All of the actions of the player are granted to contribute to the overall experience with *Slumber*.

As a concept, a video game is mostly known as a way to entertain the player, however in basically every video game, there is also a sublayer of other substances that play a part in the way the game reflects into the player. Whether we are talking about a card game, a first-person shooter, or a sports video game, there is always an expectation of the player using a certain secondary skill in order to fully enjoy and make the most out of the experience with the game. With this in mind, we are able to comprehend why every individual may have a preference between one or another genre.

So, in the case of *Slumber* this sublayer that we are talking about is the mental and strategical work that the player is expected to have while playing the game. However, the game itself has not been concepted to be played once but more times than that. This means to try to fool the player at first, giving them some choices that may appear not so important, but that end up being crucial at the end of the game. Another aspect that is really relevant to this matter is that the player must enroll himself in the game, and the best way to do so is to make the video game appealing quality-wise, especially when it comes to the player's control of the main character, and the visual and audio features of the game.

Slumber is a single-player strategic and decision-making video game so the player is expected to drown by himself into the game. In fact, throughout the game the player is all by himself until he finally meets other people in the last level of the game, where he is required to make decision that may interfere between his own personal values. This ought to act as a way to make the player look within his own self. Also, while putting together the idea for the game, I chose to have the player all by himself making decisions that may feel meaningless playing the first, but end up being key at the end. This is a metaphor of life itself, how we are all alone making small decisions every day but, at the end it all counts, even the slightest of things that may appear insignificant.

To make this happen, the player is first prompted to a main screen where he is required to input his name. Once the game itself starts, the player appears in a desert island and has to go through different events, where, in order to advance, he has to make multiple choices that do not have an immediate effect in himself but that will be the key to survive at the end and/or in the future. These events are placed in different levels and require the player to explore the environment of each level where the events take place. Once all of the events have happened, a door that leads to the next level will appear. There is a total of 5 levels, and except from the first one, where the player

only has to advance straight forward until he finds the door, the player has to interact with the object that triggers the event and make the choice he believes is more suited to his own thoughts about the matter. At the end, the player may be able to survive if he has made the correct actions in the past, if not the case, he will not survive and will have to start again. The entirety of the script and the storyboard are annexed to this document.

5.2. Work methodology

The work methodology that the development of the video game has been followed the following schedule:

1) Pre-production
    1.1) System specification and requirement analysis
    1.2) Analysis: idea clarification, script and story board making, and character background & world building study
    1.3) System design
    1.4) Planning and work methodology establishment for the production stage

2) Production
    2.1) Level design
    2.2) UI development
    2.3) Animations
    2.4) Coding
    2.5) Sound and background music
    2.6) Testing and error fixes

3) Post-production: data compilation and statistical analysis study of people who have played the game

Firstly, in the pre-production phase, all of the decisions regarding the system, the conceptual idea of the game, the design of the game and how the production of the game is going to be carried were carefully crafted and thought all throughout. This is a key phase, as having the appropriate planning and a clearly set goal and project slope is of extreme relevance for the following phases of the video game development process. First off, the system specification and the requirement for the video game was specified. After that, a more creative phase started: developing the precise idea behind the video game, setting up the characters' background and building the world where the story will be taking place. Once this was all set and clear, the writing of the script and the story board began, in the order they have respectively been mentioned. The script for the video game contains all of the dialogue that is implemented via the dialogue system of the video game and represents the core that moves the story and the action of the video game.

Secondly, in the production phase is where all of the material that has been created in the pre-production phase is implemented. Each of the levels of the video game were design in Unity, which require a great amount of time and effort in order to shape the scenes accordingly to the video game requirements, as well as the necessary items for it. In this sense, it is relevant to bring up the fact that due to the fact that the entire video game has been made with no budget at all, all of the assets that the video game could use were strongly limited. In the case that the video game was made by an actual company, the artists would be the one in charge of providing the assets needed for the matter, but as this was not the case, only the assets available for free were the ones who have been used.

After having all of the levels designed, the UI for the video game, such as the main screen, game over screen, the minimap at the corner of the screen that shows the player's position, the dialogue boxes for the dialogue system and the pause menu were implemented. Once we had all of the UI components implemented into the game, all of the animations that were necessary for the game were gathered. Once we had all of the characters with their own animation controllers, as well as the player's own animation controller, which is extremely important, we made sure that it was working properly. It was a hard task to make as much compatible as it was possible between the animations available for free on the Internet and what we actually needed for the game. This is a really important aspect of the video game, as we have to be careful that the animations that the player sees are not out of place at any time, as this would make the player get out of his character and lose concentration and immersion into the game.

When the animations of all of the characters were working properly, the coding phase began. This phase was centered in setting up all of the lower level requisites for the game to work as we want it to. This meant creating all of the triggers for each one of the events, setting up all of the variables needed to keep and use the information as needed, and overall, making sure that the video game gives the player what he is expected to receive at the proper time. At last, in this production phase all of the UI and Ambiental sound effects, and background music for each one of the levels and the menus were gathered and later implemented into the game. The sound effect department was also one hard to gather, as once again, due to the non-existent budget of the video game, the sound effects for the video game were restricted to what was available on the Internet for free. During this phase all of the available testing was done, trying to make sure that the video game works as expected.

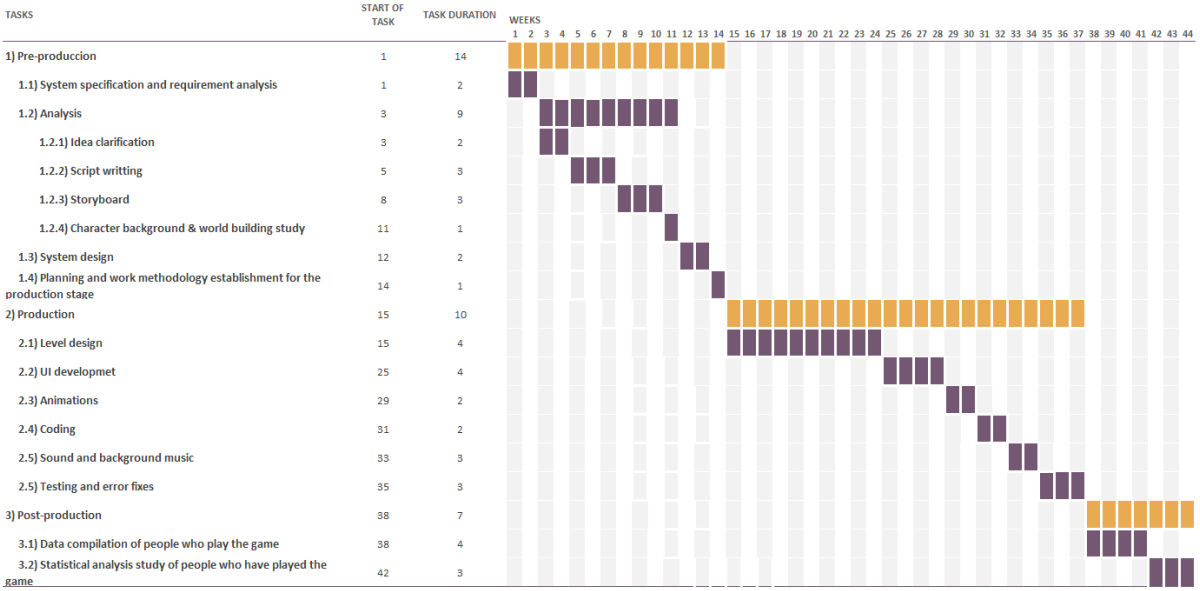
Once the production phase was over, we had *Slumber* ready to be played by other people. It was distributed to a numerous amount of people who tried it in their own computers. Each one of the people who tried the video game was asked to read a document before they start playing the game, where there was a thank you note for wanting to try out the game, and a brief description of the video game. After they had played the game as many times as desired, they were given a small survey which contained multiple relevant questions. Some of these questions were quality oriented,

while some others where directed towards the direction they had taken while making their in-game choices. At last, this recollected data was analyzed and the results were useful in order to draw conclusions both regarding psychological behaviors upon their choices and their overall impression and experience with the game.

5.3. Planning

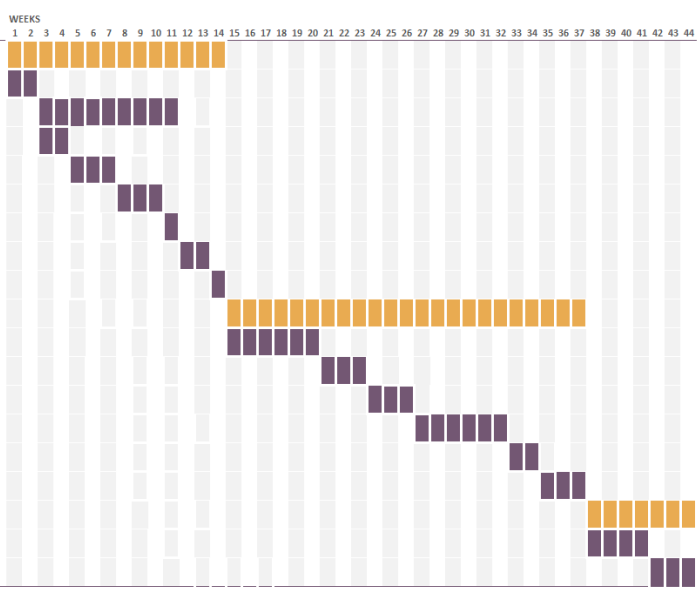
The schedule that was first proposed is the one shown below:

TASKS	START OF TASK	TASK DURATION
1) Pre-produccion	1	14
1.1) System specification and requirement analysis	1	2
1.2) Analysis	3	9
1.2.1) Idea clarification	3	2
1.2.2) Script writting	5	3
1.2.3) Storyboard	8	3
1.2.4) Character background & world building study	11	1
1.3) System design	12	2
1.4) Planning and work methodology establishment for the production stage	14	1
2) Production	15	10
2.1) Level design	15	4
2.2) UI developmet	25	4
2.3) Animations	29	2
2.4) Coding	31	2
2.5) Sound and background music	33	3
2.5) Testing and error fixes	35	3
3) Post-production	38	7
3.1) Data compilation of people who play the game	38	4
3.2) Statistical analysis study of people who have played the game	42	3



The pre-production stage was correct when stipulated, but the production stage suffered of a few variations in the lengths of each of the tasks. The schedule that represents the actual schedule for the project is the following one:

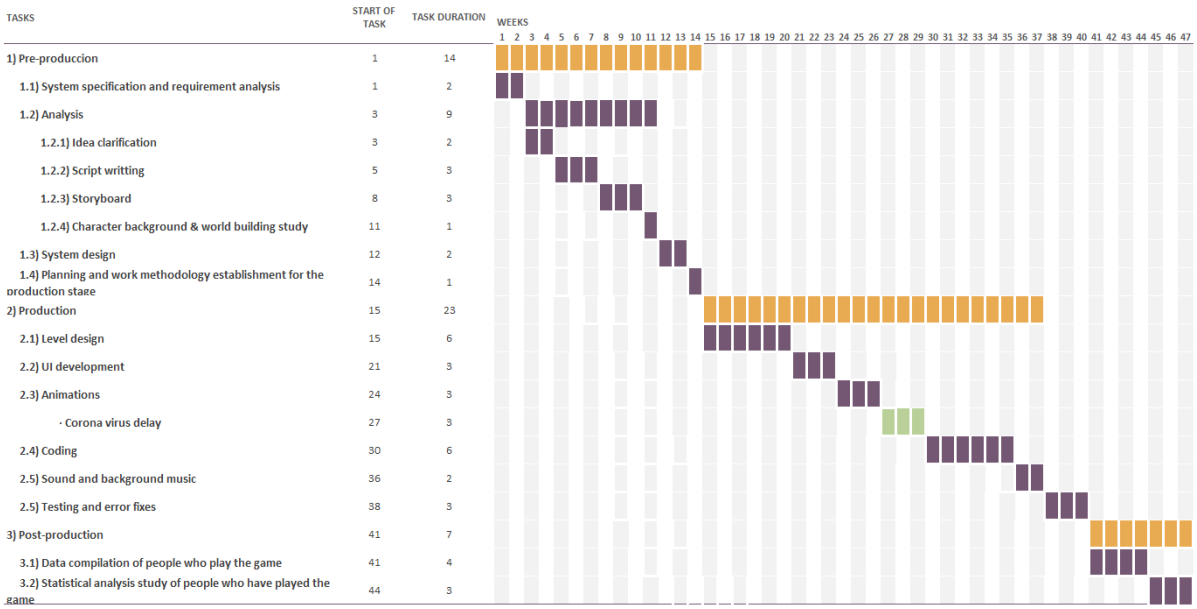
TASKS	START OF TASK	TASK DURATION
1) Pre-produccion	1	14
1.1) System specification and requirement analysis	1	2
1.2) Analysis	3	9
1.2.1) Idea clarification	3	2
1.2.2) Script writting	5	3
1.2.3) Storyboard	8	3
1.2.4) Character background & world building study	11	1
1.3) System design	12	2
1.4) Planning and work methodology establishment for the production stage	14	1
2) Production	15	23
2.1) Level design	15	6
2.2) UI development	21	3
2.3) Animations	24	3
2.4) Coding	27	6
2.5) Sound and background music	33	2
2.5) Testing and error fixes	35	3
3) Post-production	38	7
3.1) Data compilation of people who play the game	38	4
3.2) Statistical analysis study of people who have played the game	42	3



As we can see comparing both charts, both the pre-production and post-production stages had the same duration as planned as they actually were executed. Despite that, the production stage had the same duration in both the production and post-production stages with the difference that the duration of the tasks were not the ones proposed at first. While in the early schedule the level design had the longest duration, and the other tasks were shorter, being the UI development the second longest, the reality of their execution was different. At the end, both the level design and the coding took the same amount of time, being the other tasks much shorter in comparison. This reflects how the process of putting everything together and giving a sense to the video game itself took a bigger amount of time due to its complexity.

At last, it should also be mentioned that even though the schedule did not change as much when put in practice, due to the worldwide pandemic coronavirus breakthrough and spread, which has immensely affected both the personal and educational aspects of our lives, the chronogram was a bit altered. While it was going to take a total of 44 weeks, forcing a few weeks of delay for the project, which ended up taking about 47 weeks for its full development instead.

If we take in account the delay provoked by the coronavirus pandemic, the schedule looks this way:



This shows how something unexpected can affect the development process of any given project. This is the reason why we always must have a bit of a lose schedule in case something that may alter the planned schedule happens. It is of extreme importance to always be prepared for whatever may happen, take proactive measures, and react accordingly to it.

5.4. Cost estimation

When listing the costs that the development of the video game would need, we have to take in account the overall scope of the game in order to estimate the amount of work, time invested and resources that the development of the video game will require. In this context, we should start by stating the scope of the game and the departments that the game development should be divided into.

Having an average of four levels, each of them taking place in a relatively small world space, and not many props or characters, the work that needs to be done when it comes to art and modeling is not that overwhelming. Thus, the programming, testing and le*vel design are the busiest aspects in the game. We should also take in consideration the marketing and advertising of the game, which will take place after the development and the last stages of the game.

The game development work should at least be divided in the following departments. Note that the departments listed below correspond to the average departments being carried by the majority of video game developers[12]:

- Design. The design department takes care of the gameplay design, as well as the structure and main rules of the game. According to the scope and complexity of the game, we could consider having about 2 to 3 people working on the design department. Video game designers earn an average salary of $4,419 per month[13], so the design department would need an investment of about $8,838 to $13,257 per month.

- Art. The art department creates all of the art required for the video game, which needs to be coordinated and in constant contact with other departments, such as design, programming and level design. All of the 3D models, environments, animations, among other material needed for the development are provided to the rest of the departments by the art department. Having this in mind and the time that building all of the material needed, along with the possible changes that some of these models may need at some time, we could estimate a team of around 5 to 8 people working on the art of the game. Game artists earn an average of $4,459 per month[12], so the art department would require an investment of around $22,295 to $35,672 per month.

- Programming. The programming department will take care of all the programming needed for the development, which includes physics, management of the graphical content, gameplay, scripting and input processing. We could branch the department into some one or two lead programmers, who would be more experienced in video game programming and would take care of the codebase of the game as well as supervision. The rest of the programmers, about 5 to 6 people, could be less experienced and would follow the instructions given by the lead programmers. In this

[12] Moore, Michael E.; Novak, Jeannie (2010). *Game Industry Career Guide*. Delmar: Cengage Learning.
[13] Salaries - CareerExplorer. CareerExplorer, www.careerexplorer.com/

department we could also include an engine and tool programming sub-department, whose role would be to provide tools to the rest of the staff, as well as taking care of the pipeline and the maintenance of the software and computers. The more experienced video game programmers earn about $6,875 per month[12], while the average video game programmer salary is about $5,493[12]. This means that the programming department would need about $41,215 per month.

- Level design. The level design department work with development tools provided by the programming department, and centers around the conception of each level of the game. The average game level designer salary is about $4,137 per month[14], and considering of hiring about 2 to 3 people, the level design department would cost around $8,275 to $12,411 per month.

- Testing. The testing department ensures that the game falls within the expected quality levels. This department takes care of noticing and locating possible errors in the game and plays a critical role in the last stages of the game development. For this department we could have one or two people testing beta versions of the game, and maybe hire about 3 to 4 more people for the last tests of the game prior to its release. A game tester earns about $2,426 per month[15], so we would need to make an investment of about $2,426 to $4,852 during the development of the game, and then the quantity required would go up to about $9,706 for the last tests.

- Sound. When it comes to the sound department, the fact that the characters will communicate via dialogue boxes means that there is no need for voice actors. For this reason, the cost required by this department will be highly reduced, and will only center in background music and ambience sound effects. What's more, the use of free royalty music can also decrease even more the investment needed in the sound aspect of the game. With this in mind, the cost of the sound needed for the game will not be so greatly, as we will only need a foley sound artist for the sound effects, a music composer for the music score of the video game, as well as hiring a symphonic orchestra that records the said soundtrack for the game. Another alternative to having an actual composer for the music score and an orchestra that plays the music, is using license free music that is available on the Internet. At most, and taking the more expensive way, this aspect of the game would cost from $2,000 to $6,000, depending on factors such as the experience and prestige of both the music composer and the symphonic band[16].

- Marketing and advertising. We should think about the costs that would generate using several ways of advertising, such as television advertisements, print advertisements in magazines, online advertisements, as well as events and launch

[14] "Average Game Level Designer Salary." PayScale, www.payscale.com/research/US/Job=Game_Level_Designer/Salary.
[15] Salaries - CareerExplorer. CareerExplorer, www.careerexplorer.com/
[16] "How Much to Charge For Composing Music?" Film and Game Composers, 2 Oct. 2015, filmandgamecomposers.com/guides/how-much-should-i-charge-to-compose-music/.

parties. Due to the effortlessly easy and inexpensive way of marketing that social media and the Internet represent, the marketing aspect of the game will not need a lot of investment on it.

Along with the cost of the employees' salary, the cost of the working place should be taking in account as well. For the size of the personnel that has been considered, about 18 to 24 people, and the estimated duration of the entire development process, renting an office for about $1,500 monthly[17], would be fitting for the matter. Obviously, the cost of the rent for the place would differ depending on where it is actually located, but as the project doesn't really have an exact place where it must be made, then this cost can fluctuate mildly.

Another alternative to this matter, is that there may not be a need to rent a place for all of the personnel to work together, but instead we could consider having them working separately whether it is in separate offices by department or even having them work from their own homes. However, this would mean that the development process of the video game would require a secure and reliable infrastructure to coordinate the files and the working methodology between all of the parts involved in the development process.

There are no plans right now on adding any online feature to the game, so there is no need to include the cost of a domain and/ or cloud storage site. However, it would be interesting to create a website with the purpose of advertising the game so people can learn more about the game first-hand. Also, the fact of having a website enhances the overall image of the brand and the game itself, making it appeal more solid and stable.

[17] Find Your next Office. www.instantoffices.com/.

Another aspect we should be looking at is the manufacturing and distribution of the game. In this sense, we should decide if the game will be distributed only online or if it is also going to be produced by optical disc. This decision should be made looking at the profit that optical disc distribution would provide us or if the comparative profit is not going to be enough to cover the investment. In order to have a clear idea of this matter, we should take a look at progression of the sales of computer and video game sales from 2009 to 2017.

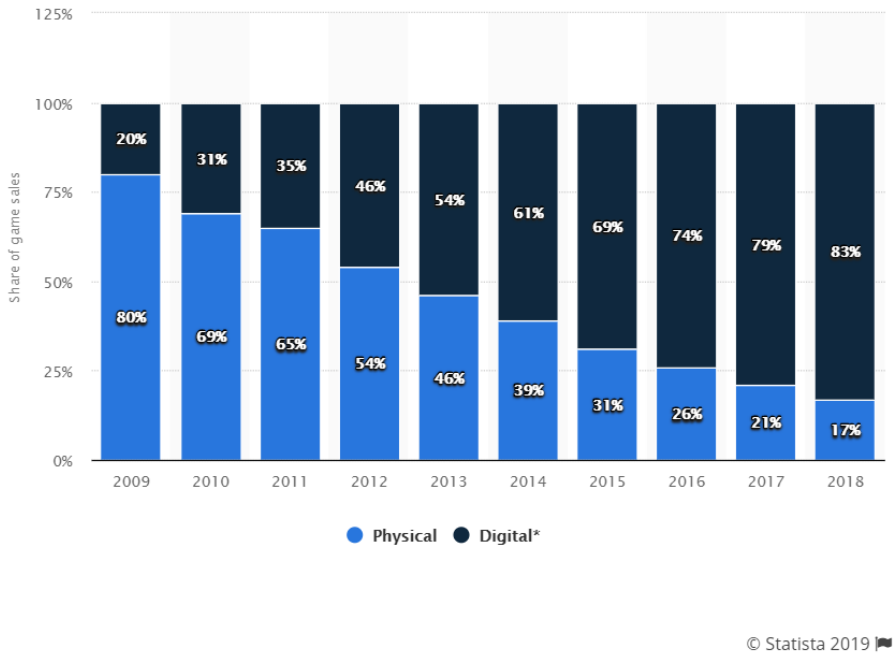


Figure 9. Breakdown of U.S. computer and video game sales from 2009 to 2017, by delivery format. Statista

As we can see in the previous graph[18], the way that video games are acquired have experienced a total turn-around. Digital games' sales have overpassed drastically those of the physical copies in the last years, due to the improvement that cloud and digital content went through in the last decade. Indeed, the gaming industry possesses a strong bond with cloud technology, which provides many benefits to gamers. Digital distribution of games allows players to have instant access to video games right after buying them without having to physically go to a store and get it.

Another benefit that players find in digital distribution of games is the fact that they can have access to new video games at midnight the same day they come out. This method encourages the most anxious fans to go for digital distribution of games as their preferable way to buy their video games. Game digital distribution platforms supply players digital wallets that appeal as a comfortable and trustful way to buy

[18] "U.S. Computer and Video Game Sales - Digital vs. Physical 2017." Statista, www.statista.com/statistics/190225/digital-and-physical-game-sales-in-the-us-since-2009/.

games. As well, game digital distribution platforms grant players with discounts on games more often than their physical counterparts, and most people simply buy physical copies of games for pure collection purposes.[19] This means that releasing physical copies of the game would only serve as another method of advertising, and the costs of it would not grant any benefit.

The cost estimation must also include the cost of licenses, from software licenses to distribution and release licenses. For starters, the main cost when it comes to licensing is the software needed for the development of the game itself. Each one of the computers that will be used for the development needs the licenses for the software that they are going to be using. Given the fact that the version of Unity that we will be using is the free one, the only software license we would need to pay for is the software for the modeling and art departments. If we suppose that the operating system is Windows, for an average of 10 computers, we would need an investment of $68 for Adobe Photoshop and another $283 for Maya, each one of them per month.

Lastly, we could also consider using a software to unify the pipeline of the development of the video game, and this way, make the project management easier and more efficient. This kind of software allow a more fluid communication and integration between all of the parts involved in the development process of the video game. The use of a software of that kind, such as Shotgun, would require an investment of about $30 to $50[20].

So, taking all of this into account, the approximate cost of the development of the video game would be around $105,000. This is a suitable cost, considering all of the aspects that need to be implemented into the game, and even more assuming we want the video game to have the best quality for its release available under an adequate budget.

However, all of the data recollected is applied in the supposed case a video game production company would want to hire a video game developing studio and release the video game under their trademark. In reality, the video game has been developed all by myself, with no cost other than the price of my laptop where I have developed the game, access to Internet, and a lot of effort put into it. All of the assets (character and props models, animation, background music, sound effects, etc.) and software used are available for free and have been acquired under zero cost.

[19] “Why Do Gamers Still Buy Physical Games?” GamesIndustry.biz, www.gamesindustry.biz/articles/2018-09-14-why-do-gamers-still-buy-physical-games.
[20] “Pricing.” Shotgun Software, www.shotgunsoftware.com/pricing/.

5.5. Feasibility study

As it has been mentioned before, the video game industry is one of the most profitable industries in the world right now. In fact, video games are the most popular and profitable form of entertainment.[21] In 2017, the gaming industry produced an estimated revenue of $116 billion, beating movie, music, TV and TV streaming services. In addition, the gaming industry is still growing with around a 10.7 percent growth in 2018.[22] Below is a graph illustrating these facts:

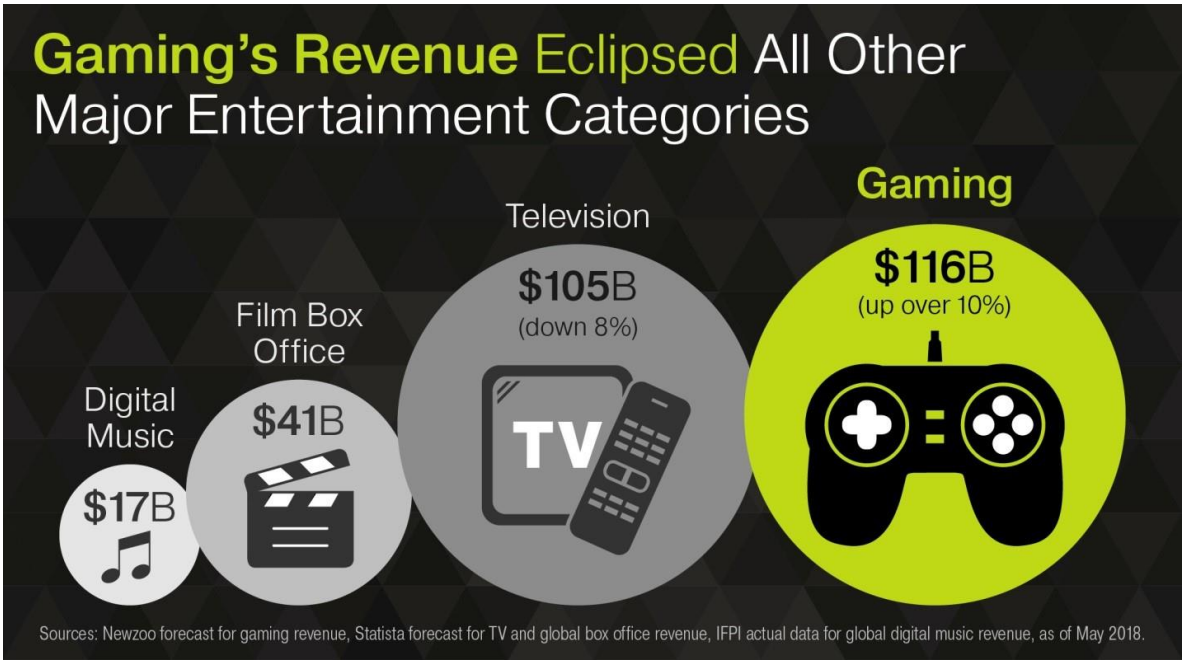


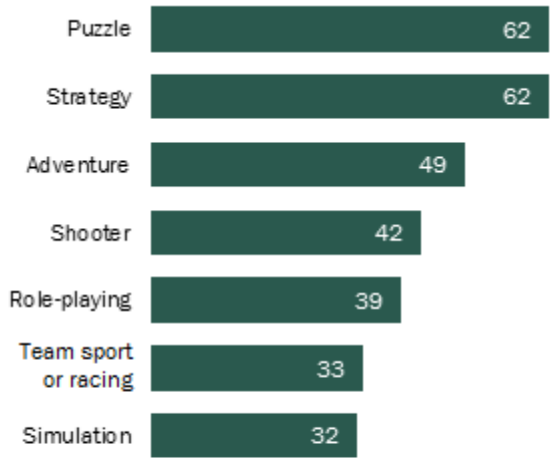
Figure 10. Gaming's revenue compared to other entertainment industries and its yearly growth. Statista, Newzoo, IFFPI.

With gaming now statistically dominating the entertainment stage as of today and the near future, we can absolutely confirm that it is profitable to invest on the development of a video game. However, the fact that it is such a profitable industry implies that there are many other game creators that compete in the same market. Luckily, the game market is substantially broken down into many genres, so we should take a look at the possible competitors of our video game in the market.

[21] "STATISTICALLY, VIDEO GAMES ARE NOW THE MOST POPULAR AND PROFITABLE FORM OF ENTERTAINMENT." Gamecrate. www.gamecrate.com/statistically-video-games-are-now-most-popular-and-profitable-form-entertainment/20087.
[22] "2018 Video Game Industry Statistics, Trends & Data - The Ultimate List." WePC.com, 1 Aug. 2019, www.wepc.com/news/video-game-statistics/.

In order for a video game to be successful, it needs to have a clear demographic target of the market which can be interested in the game and end up being a potential player. This is of extreme relevance, as we cannot simply make a video game and hope for it to be successful. Before starting the developing process of the game, we have to make sure that a video game has a good chance of being successful and provide revenue after its release on the market.

Slumber will have PC as its platform to being played, so we should compare how it performs against the other gaming devices, such as mobile phones, handheld gaming devices and home consoles, as its potential players will be amongst those that play video games of its genre and specific gaming platform. This was chosen because PC gaming has been exponentially increasing in the past years, being 52% of gamers that play on PC[23], and it is expected to increase even more as the time goes by. As of 2020, An increasing number of people are upgrading their PCs in order to play videogames, set to equate to a 10.3% sequential rise in sales across the industry year-on-year.[24] This is product of having so many video game digital distribution platforms, such as Steam, and the flexibility that playing in a PC brings to the user. As a fact, 89.3% of all households in the United States had a computer at home in 2016[25].



Note: Response options are not exhaustive and respondents were permitted to select all that apply. 5% report that they play none of these types of games.
Source: Survey of U.S. adults conducted March 13-27 and April 4-18, 2017.

Figure 11. Preferred genres by gamers

When it comes to the genre, *Slumber* mainly falls under the strategy and decision-making category so we should take a look at some demographical statistics regarding the category where *Slumber* will be a competitor. On the left is a graph representing the preferred genres among gamers.

As we can see, puzzle and strategy games are among the most popular type of video games played by gamers[26], as about six-in-ten adults who say they play video games often or sometimes report that they typically play

[23] "Video Game Statistics" Review42, 19 May 2020, review42.com/video-game-statistics/.
[24] Ridley, Jacob. "'Millions of New PC Gamers' Are Turning Their Home-Working PCs into Gaming Machines." Pcgamer, PC Gamer, 8 July 2020, www.pcgamer.com/pc-gaming-hardware-report-2020/.
[25] "Digest of Education Statistics, 2017." National Center for Education Statistics (NCES) Home Page, a Part of the U.S. Department of Education, nces.ed.gov/programs/digest/d17/tables/dt17_702.60.asp.
[26] Brown, Anna. "Who Plays Video Games? Younger Men, but Many Others Too." Pew Research Center, Pew Research Center, 7 Aug. 2020, www.pewresearch.org/fact-tank/2017/09/11/younger-men-play-video-games-but-so-do-a-diverse-group-of-other-americans/.

these types of games (62% each). This means that our video game has a higher chance to find its own place in the market as the audience it is aiming to is quite large. Apart from the most played genre and the gaming device where it will be played, we should also briefly study the economical financial status of our market target. Because of its nature, the video game requires mental work from the player to fully get themselves involved in the story and make the corresponding choices according to their own principles. This means that the video game will also required a level of mental complexity from the player, so even though there are not any particular aspects of the game that would not allow a minor to play it, it is more aimed to adults rather than teenagers who are not expected to have the mental maturity the game requires to fully experience it. Taking this in account, we can narrow the demographic target of the game to both females and males 18 to 54 years old, as there is never a gender implied in the game and it is a genre both genders tend to enjoy, as we can see in the graph below:

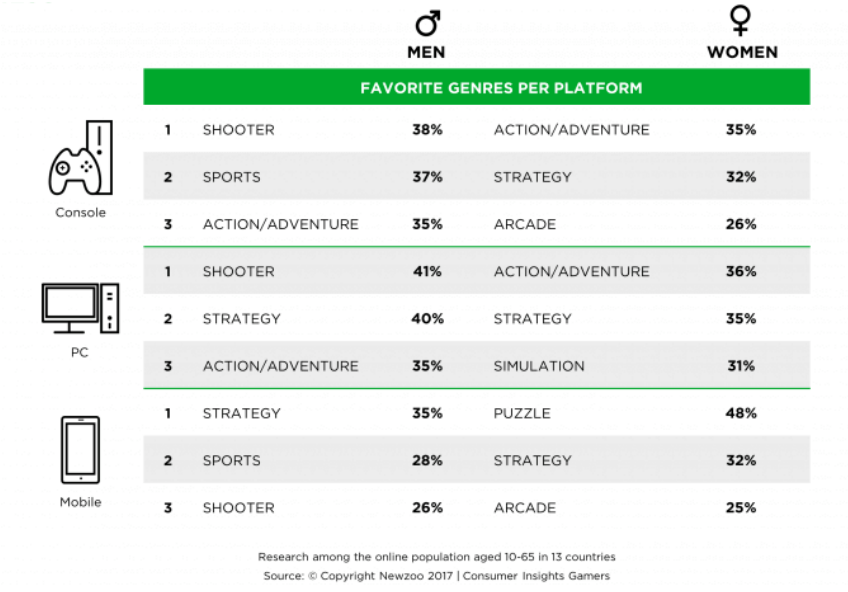


Figure 12. Preferred genres according to player gender

The market target of *Slumber* is a safe play, as it is people with a higher level of education the ones who usually play the genre. The fact these people are the main target of our video game makes the release of the video game greatly feasible, as their financial status is crucial to ensure that the video game will have good sales upon its release.

At last, we also should take in consideration the appeal of the game so that it attracts potential players who may get interested about it. It is not only important to carry an effective promotional campaign before and after its release, but we have to make sure that the video game looks appealing as a trademark. Every video game, even though it may fall in the same genre as other games, needs some special features that makes them stand out from other competitors. In the case of *Slumber*, this

differential factor comes in the form of challenging the player to put himself in the shoes of the main character, survive and try to discover what is going on while relying on its own decisions.

The mysterious vibe of the game is enhanced by the color palette chosen for the logo and the look of it. As we can see pictured below, on the left we have the logo for the video game, and on the right, the meaning of the color palette used for it.

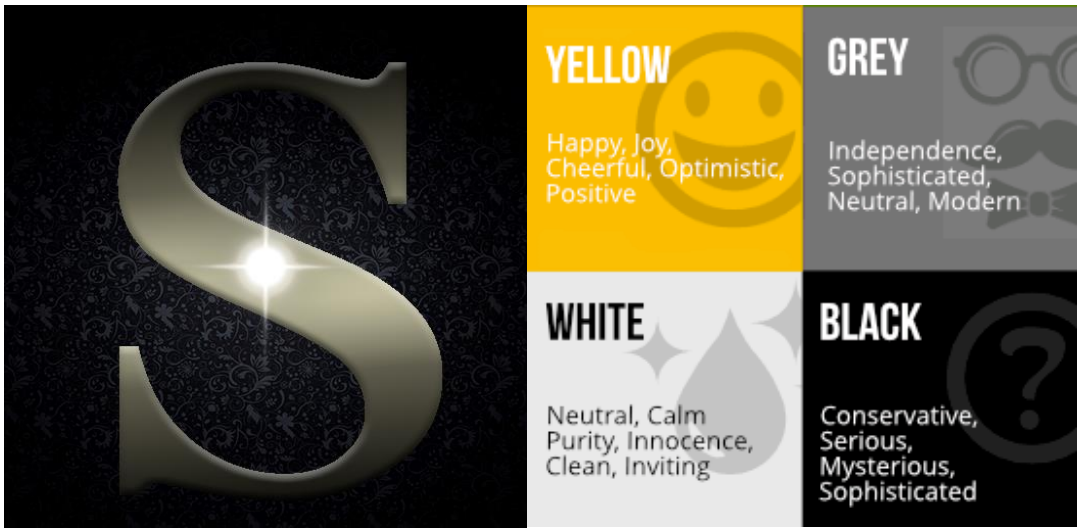


Figure 13. On the left, the logo for Slumber, on the right, the meaning behind the colors used

The colors chosen for the logo are meanly black and grey colors, which are colors that transmit a serious, mysterious and sophisticated feel to the image. These colors were selected because of the desire to truly let the player know that it is a mysterious game with a dark twist to it. This is also remarked by the fading colors which are taking over the gloom of the light that surges from the center of the letter S, as well as the name of the videogame itself, Slumber, which is truly mysterious and key to the story. This gloom of light means to represent the ray of hope that emerges from the entire adventure, which also reflects on the subtle use of the colors yellow and white. These colors that represent optimism, purity and innocence, have the goal of bringing some hope to the player, letting them know that within the darkness that surrounds the adventure, there's a way to overcome all of the issues they encounter and get to salvation. Overall, the use of these pictures want to appeal to the player at first sight, trying to catch their attention and make them want to jump into the adventure.

Having all of this in mind, we should also study the competitors in the same genre of our video game. In the first place, we must consider the previously mentioned *Detroid: Become Human* as one of the potential competitors of the game. Released in 2018 it has already sold more than 5 million copies overall as of Summer 2020[27], so that is yet another proof of how strong the market target of *Slumber* is. Another strong

[27] Reis, Eric. "Detroit: Become Human Reaches PC Sales Milestone." Game Rant, 5 Aug. 2020, gamerant.com/detroit-become-human-pc-sales/.

competitor for our video game is *Disco Elysium*, which is a role-playing video game developed and published by ZA/UM released in October 2019, that features an open world and dialogue-heavy gameplay mechanics, where the player has to assume the role of a detective and solve a murder case involving several decision he is required to make. Even though *Disco Elysium* does not have the massive success that *Detroit: Become Human* has, it has been praised in many different aspects and has been greatly received by video game players and critics, even being awarded a 10 out 0f 10 by GameSpot, their first perfect score since 2017[28].

There are a few things we must consider while looking at these facts. First of all, we should note that both of the video games mentioned before have been highly acclaimed and received by the players. This is an evident clue that tell us that the attention gained by these video games means that the video game has already a strong place in the market. The fact that there are competitors with our video game does not mean that the video game has a higher chance of not being successful, but rather, that there are potential players that are looking for more video games of the like. We should take advantage of these fact and release a video game that can be just as successful as the ones mentioned before. Another valuable information about these facts is that if we look at the release date of the games, we notice how recent they are. This is yet another sign that the genre has a strong spot in the market that we must take advantage of and exploit.

[28] Wildgoose, David (4 November 2019). "Disco Elysium Review - Pure Dynamite". GameSpot.

5.6. Risks

About the risks that developing the video game may have, they could refer to different aspects and distinct stages of the developing process. For every one of these potential risks that the video game may go through, we should have a contingency plan in order to reduce as much as possible the damage and the loss of money that they could generate.

In the first place, we should consider any kind of issue that the departure of an employee would mean to the developing process. If this were to happen, the developing process would inevitably be slowed down, because it would require to look for a new person to take the place of the worker who has left the developing team. Not only it would mean a relevant loss of time, but it would also require an increasement to the overall budget of the game. There is not really a contingency plan for this; the only things that can be done in order to reduce as much as possible the impact that this may have, is not have a tight schedule while planning the developing process, and taking all of this in account, and also, try to make sure that everybody is working properly and happily in their own assigned tasks. Obviously, we should consider that everyone working on the project is human and there is always the possibility that something unexpected happens in their personal life and thus, they are forced to leave the working team. These are both proactive measures that we should take while planning the schedule for the development of the game.

Secondly, and also kind of attached to slightly the same aspect of the personnel, is that one of the workers of the personnel can leak information about the game to third parties who may steal the idea or some concepts related to it. This could be dangerous specially upon the release of the game, so we should be prepared and have both proactive and reactive plans for it. As for the proactive part, we can make sure that the information of the game is as much restricted to the respective worker as possible, which can be done by having restrictions when it comes to the material they are given for their respective tasks. Also, each one of the workers must sign an official document where they assure that they will not be leaking any information to anybody outside the project. Even though, these measures are not totally reliable, as it is ultimately down to the worker to release any kind of information. So, taking this in account, we should also consider a more reactive measure, which would be trying to change some of the aspects of the video game that are leaked, in case it ends up being a part of another project outside *Slumber*. This risk would also mean a loss of time for the developing process, but it is still manageable if these mentioned measures are executed.

Another potential risk is that the main office or some of the offices where the game is being developed, goes through a natural disaster. This is an unlikely risk, but there is always the chance that it may happen. Another risk related to this one, is that one or more computers, as well as material that is being used for developing the video game may go to waste or break. A proactive measure against this risk is having a bit of a lose

budget in case one of these risks happen, and take the reactive measures, such as repairing the damage, as fast as possible.

Having any part of the development process delayed, whether it is a checkpoint of a task or a task that is taking longer than expected to be fulfilled, is also something that we should really must be careful of. The best way to ensure that all of the tasks are performed within the established duration for the task, it is of extreme relevance to take the proactive measure of checking the work of every person working on the development of the video game. This way we make sure that the project is following the schedule planned and the risk of anything being delayed is greatly lowered.

We should also consider risks that are more related to the market. One of the risks that we should take in account is the fact that, once there is a set date for the release of the video game, another popular video game such as a sequel of a greatly popular saga or a hyped video game may set its release as well. There is not really a proactive measure that can be planned for this case, but instead, the most appropriate reactive measure would be postponing a bit the release date of *Slumber* once the market is not as irradiated by this new video game. The second risk that can happen in this aspect, is that a new gaming device gets release, which would take the spotlight in the video game market. However, the release date for consoles are usually set with enough time, so the re-schedule for the release of our video game should not present much of a problem. The good thing about these is that they can be easily solved, but nevertheless should be taking in account.

Lastly, and also related to the market, the last risk that we should be careful of, is the video game not having a good reception upon its release. In order to prevent that, the most important proactive measure would be to have a solid and strong marketing plan, like the one proposed before, for its release. The main idea is to hype the gaming community about the video game so that it has enough people interested on it, as it is a newcomer in the video game market. Even though we have a strong marketing plan backing up the release, there is always the possibility that the video game does not get the expected public reception, so we should also have a reactive measure in case this happens. These measures would be to have a lose planed budget for the marketing aspect, so that we can invest more in marketing for the video game, as well as considering lowering the price in the way of sales and discounts, which are normally attractive to the potential buyers[29]. The goal is to have as many people as possible giving *Slumber* a try, and make them feel happy about their experience so that they recommend it to other players and the sales end up generating incomes for the project.

[29] Santini, Fernando de Oliveira, et al. “An Analysis of the Influence of Discount Sales Promotion in Consumer Buying Intent and the Moderating Effects of Attractiveness.” Revista De Administração, Elsevier, 6 Sept. 2016, www.sciencedirect.com/science/article/pii/S0080210716304022.

6. Project development

6.1. Conceptual

6.1.1. Character background and world backstory

In this point, we will explain the background of the characters involved in the story, as well as the backstory that has been thought for the video game.

The player goes through and induced dream, which is a creation by the current government. The government wants to create a utopian society and in order to do so, they are putting every human through this dream. Only the ones who make it alive are considered to be worthy of keeping in the society, while the rest of the humans are discarded and do not belong in the society the government wants to build. The video game's name itself, slumber, is a clue to the player to what is actually going on.

The video game covers the player's journey in this induced dream, which will have to go through several levels and discover the truth behind this chain of events. The player will have to make various decisions which will allow him to get a glimpse of what is going on and how to stop it. However, even if the player ends up surviving, the video game only covers a part of this induced dream, leading him to the start level again but with a new door which leads into another journey. In this sense, the game adopts a cyclic shape. This shape is intended and its purpose is to give the player a feeling of enclosure.

The characters that appear in the video game are the following:

- Player: the role taken by the player. He is a regular person which has been put by the government in an induced dream in order to analyze if he has the correct ideals required to live in the utopian society that is being put by the government. He does not remember anything prior to the dream, which is caused by the process of being put in the induced dream. The name of the player will be chosen by the actual player of the video game and will be used in-game.

- Girl: a 11 year-old girl which has been put in an induced dream. She has survived all the levels and got to the white room. However, she did not take the pearl in the gloomy room. She also did not drink any water from the lake nor stored it, and thus is suffering from dehydration when the player meets her. Only the player can help her to survive, otherwise she will die.

- Pregnant woman: a 27 year-old woman who is pregnant and has been put in an induced dream. She has survived al the levels and got to the white room, but she didn't take the pearl nor eaten any fruit. The woman is starving when the player meets her. If the player chooses to give her the fruit, she will die because it is poisoned. However, if she does not receive any fruit from the player, she will not die because of it.

- Doctor: a 43 year-old man who is hired by the government. In the game he acts as the main antagonist and his job is to judge and keep track of the people

who go through the induced dream. He is put in an induced dream as well, however because the process is different, he does not lose any of his memories. He looks like a reliable person on the outside, but on the inside he is a heartless selfish man who only cares about himself.

He works for the government because he is sure that all human beings should act the same way and have the same ideals and education. Because of such belief, he is determined to eradicate everybody who do not act exactly as the government dictates. His ideals are correctness, accuracy and impeccability.

6.1.2. Requirements

Slumber is a single-player video game and thus, the inputs that the video game will be receiving are driven by the actions of the player. In order to achieve this, the video game must be both visually appealing to the player, as well as easy to understand. Even though the game requires mental effort from the player when it comes to making choices in the several events that take place in the game, the player must know at all times how to advance through the game. In that sense, it is important that the player feels comfortable using the controls of the main character of the game, and the interaction with the environment and the graphical user interface.

We should take all of this in consideration while taking in account the requirements of the video game. Below are listed the functional and non-functional system requirements:

- Functional requirements:

FR1: Player will be able to control the main character of the video game through the use of keyboard and will be able to interact with the UI elements through the mouse.

Evident

FR2: Player will be able to interact with the environment through gameplay. By interacting with the environment, the player will activate events that will require decision making strategies.

Evident

FR3: Game will have several levels in which the player is expected to advance though the events. The player will be prompted with clues via internal dialogue and on-screen messages in order to do so.

Evident

FR4: Game will engage the player and make him invest some time in making decisions throughout the game. The decisions will be what ultimately will lead to a good or a bad end.

Evident

FR5: All dialogue will be portraited and given to the player via dialogue boxes. Apart from the dialogue boxes, there will also be clues portrayed to the player via on-screen messages.

Evident

FR6: User interface must be comfortable and intuitive to the player.

Evident

FR7: Game will require identification (name) from the player before starting the game for the first time. That name will be used as the name of the main character for the rest of the game activity.

Evident

FR8: Player will not be able to go back to levels which he has advance through already.

Evident

FR9: Game will use royalty free background music.

Hidden

FR10: Player will be able to exit the game at any time during gameplay. In order to do so, he will be required to use the exit button via UI

Evident

FR11: Player will be able to look at an information screen that will give him information about the controls of the game. This screen will be overlay the current viewport of the player.

Evident

FR12: Player will be able to access and change settings of the game accessing the settings screen from the main screen.

Evident

- Non-functional requirements:

NFR1: Game will load within seconds.

NFR2: Game will have 3D models for the environment and the characters.

NFR3: Characters (main character and NPCs) will not be based on real life people and are expected to appeal some degree of realism.

NFR4: Game should run on PC (Windows).

NFR5: Game will be developed in Unity3d.

NFR6: Game will not require any payment in-game.

NFR7: Game will be programmed using C#.

NFR8: Game should run smoothly within an adequate amount of FPS.

NFR9: Low-end computers must also be able to play the game. This means that the system requirements, both software and hardware wise, should not be too high.

6.1.3. Use case diagram

Now that we have a clear idea about the video game, its requisites and several specifications, we will briefly discuss all of the functionalities of the system, as well as all of the possibilities of user interactions with the video game. All of these will be presented in the form of UML diagrams, which will make it simple and clear to understand.

First of all, we should specify the actors of the system. As this is a video game, the ones involved in the use of it will be the software of the video game, the gaming device, in our case, the personal computer and its peripheral devices such as keyboard, mouse, screen and sound output, and the player himself. Obviously, the player will be the main

actor at all cases, and the only one who can actively interact with the video game, but we also have to make sure the computer meets the requirement so that the player can experience the game as it is supposed to be experienced. The player will make use of the computer peripheral devices in order to interact with the video game.

Once starting up the game, the player will be prompted to a main screen before getting into the game itself. In this main screen he will be able to exit the game, change some settings and start playing the game. Before starting the game, the player is prompted to enter his own name which will be used for the rest of the game. Once the player starts the game, he will not be able to go back to the main screen, but nevertheless, will be able to exit the game at any time.

With this in mind, below are the use cases of the video game that make up the conceptual stage of the project:

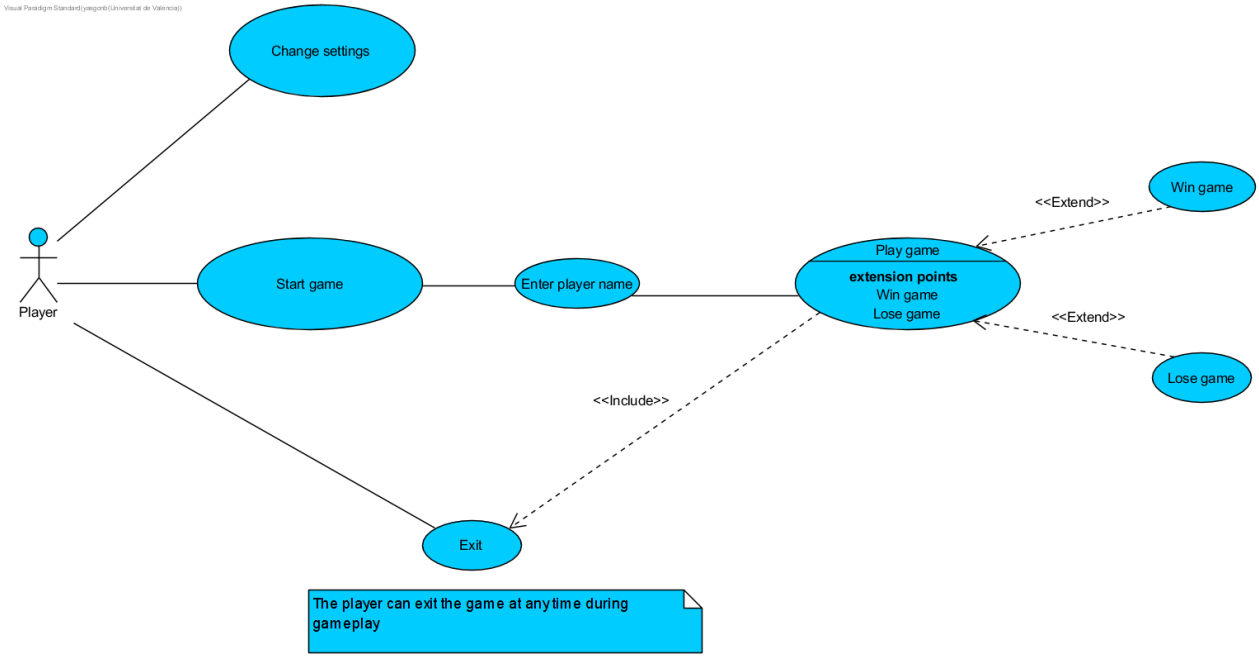


Figure 14. Use case diagram

6.2. Analysis

6.2.1. Static point of view

In this static point of view phase, we will analyze the classes involved in our system, as well as the object instantiated by those classes and how they are connected to one another. In order to do so, we will take a look at their corresponding UML diagrams, so it is easier and clear for the matter.

Here is the class diagram for our system:

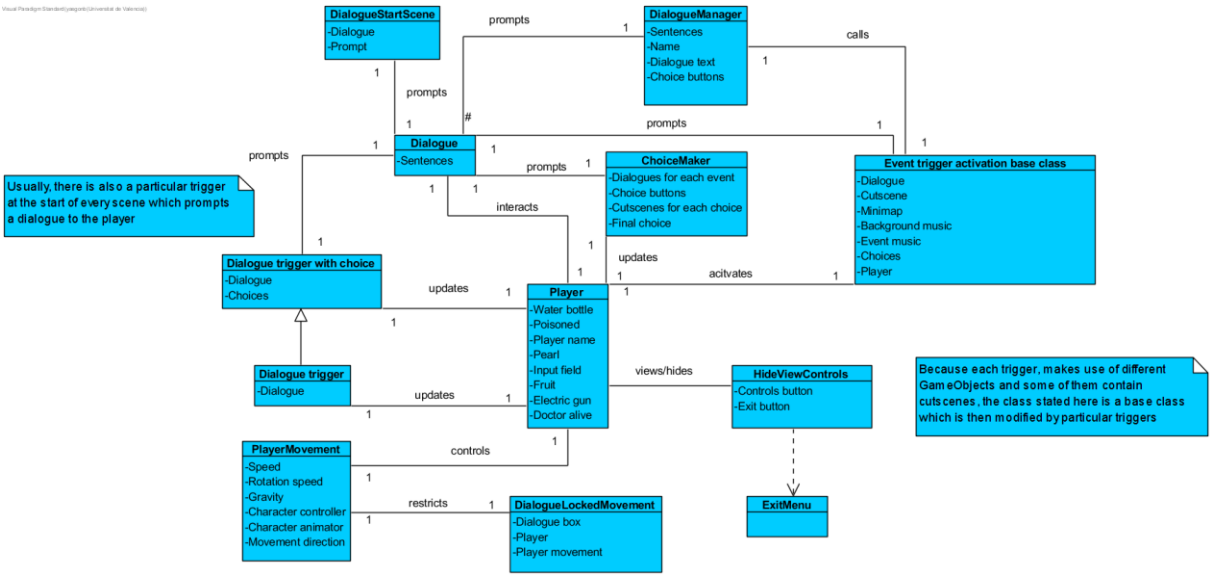


Figure 15. Class diagram

In this diagram, we have included the main classes involved in making the game work as it is expected. We have to keep in mind, that the entire game uses about 40 different scripts, as there are many events that do not use the same number of game objects and/or do not behave the same way. For simplification, in the diagram we have only included base classes that are normally the ones implemented into the events, for example a base class for all of the event triggers.

Another thing to notice about the diagram is that because of Unity's nature and the use of GameObjects, as well as the nature of the game, the great majority of the objects have a one-on-one connection. This is caused because the game is driven by the player's interactions with the environment and the UI elements, and as a system, the game simply reacts to their actions. We will go over each of the classes, their attributes and their methods in the design class diagram.

Next up is the object diagram of our system. This object diagram shows different instances of objects that implement some of the classes pictured in the class diagram. In the diagram, we have supposed the case that the player "Yasmine" is in the "Pearl

event" of the last level. In this event, the player has a dialogue with the doctor character, and if they have made the right decisions in the previous levels, the player will be asked to make a choice about who to sacrifice.

As we can observe in the Player instance, the player has made the right decisions, and thus, the corresponding dialogue with choices will be prompted. The "Choice maker" object will keep track of the player's decisions, while the "Dialogue manager" and "dialogue" show the corresponding dialogue to the player. At last, the "Player movement" object is restricted by the "Dialogue locked movement" so that the player is unable to move while the dialogue is showing on screen.

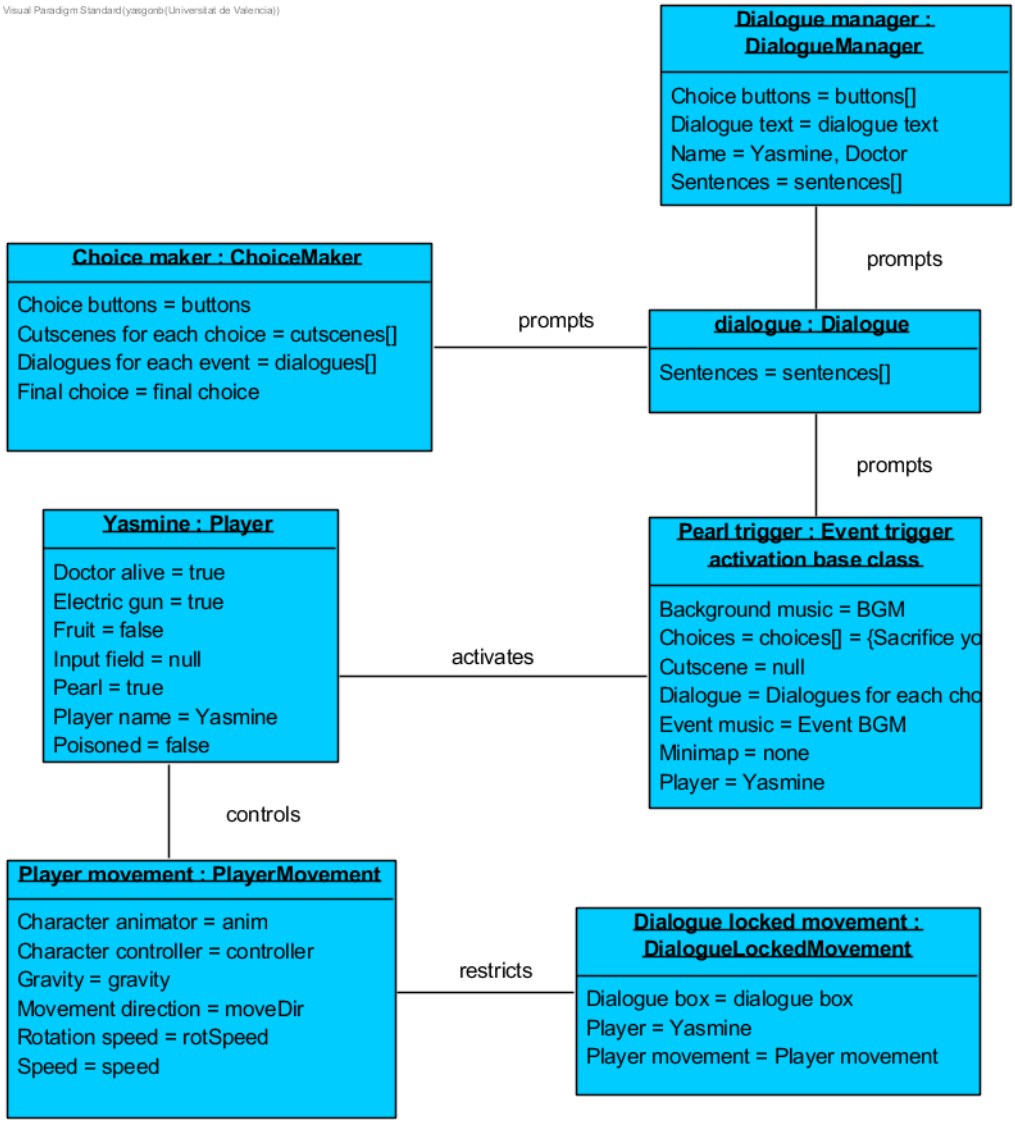


Figure 16. Object diagram

6.2.2. Dynamic point of view

Now, we will look at the system with a dynamic point of view. In this sense, we will use interaction, state and activity diagrams to have a better understanding of our system.

6.2.2.1. Interaction diagrams

The interaction diagrams that will be useful to understand our system are thee system sequence diagram and the communication diagram. Firstly, the sequence

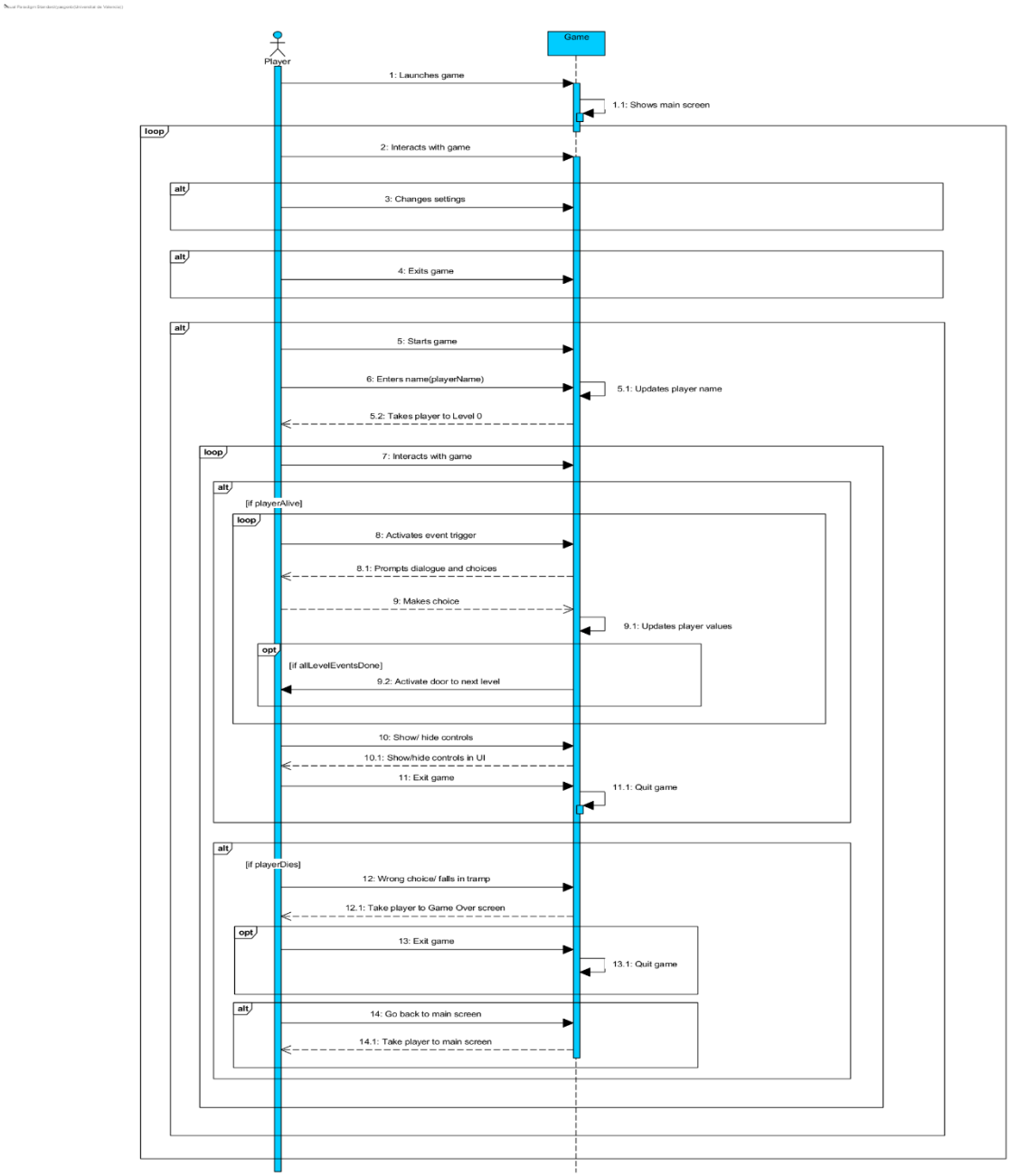


Figure 17. System sequence diagram

diagram shows object interactions arranged in time sequence of the execution of our system. The one for our particular case is the one pictured below:

The system sequence diagram pictured above represents how the player interacts with the system, which in our case is the game itself. The system sequence diagrams displays since the game launches the game via the executable file, until he either loses or wins the game, according to the decisions made. After the game has been launched, the diagram shows the possibilities that the player has within the main screen of the game, which is either exit the game, change game settings or start playing the game.

After the player enters his name and starts playing the game itself, a play game loop has been considered in order to clearly show how the system works. In this loop, the player interacts with the game, and interacting with it will ultimately lead him to either winning or losing the game. Winning the game will take the player to the last level which is a cutscene followed by being taken to the main screen, and losing the game will take the player to the game over screen. Lastly, in this loop we have considered the player quitting the game as well, as he can do it at any time while playing.

The other diagram relevant to the interaction point of view of our system, is the communication diagram. In this diagram is reflected the messages exchanged by the player and the system, which in our case is the game itself. The communication diagram is the following:



Figure 18. Communication diagram

6.2.2.2. State diagrams

The most important state diagrams that our game will use can be classified into two different categories. On one hand, we have the animation controller diagrams, that are used by Unity to play the corresponding animation according to the character's state. These are controlled by script in *Slumber*. On the other hand, we also have a few state diagrams that will be helpful to portrait some of the game's mechanics system-wise.

The first of these last-mentioned state diagrams is the state diagram that models the availability of the player to move the character while playing. In order to have a better control of the event triggers and also to have the player keeping as much focused to the dialogues as possible, the player will only be able to move the character with the game controls as long as there's not a dialogue active on screen. This means that the player will only able to move the character as long as there is no interaction with the dialogue via the UI elements. The player will start being able to move and interact with the environment, and will be restricted when there is a dialogue on screen until he is done with the dialogue.

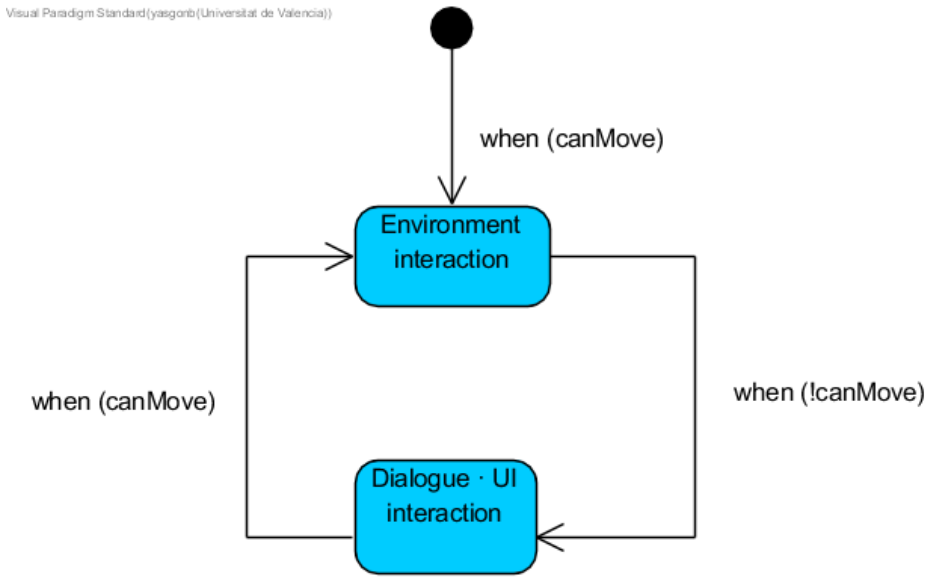


Figure 19. Player movement state diagram

The second one of the state diagrams is the one that represents whether the player's props are visible or not. Throughout the game, the player will have to interact with these objects, which are a part of the GameObject of the player itself. These objects will not be visible unless the player makes the correct choices, as stated in the following state diagram:

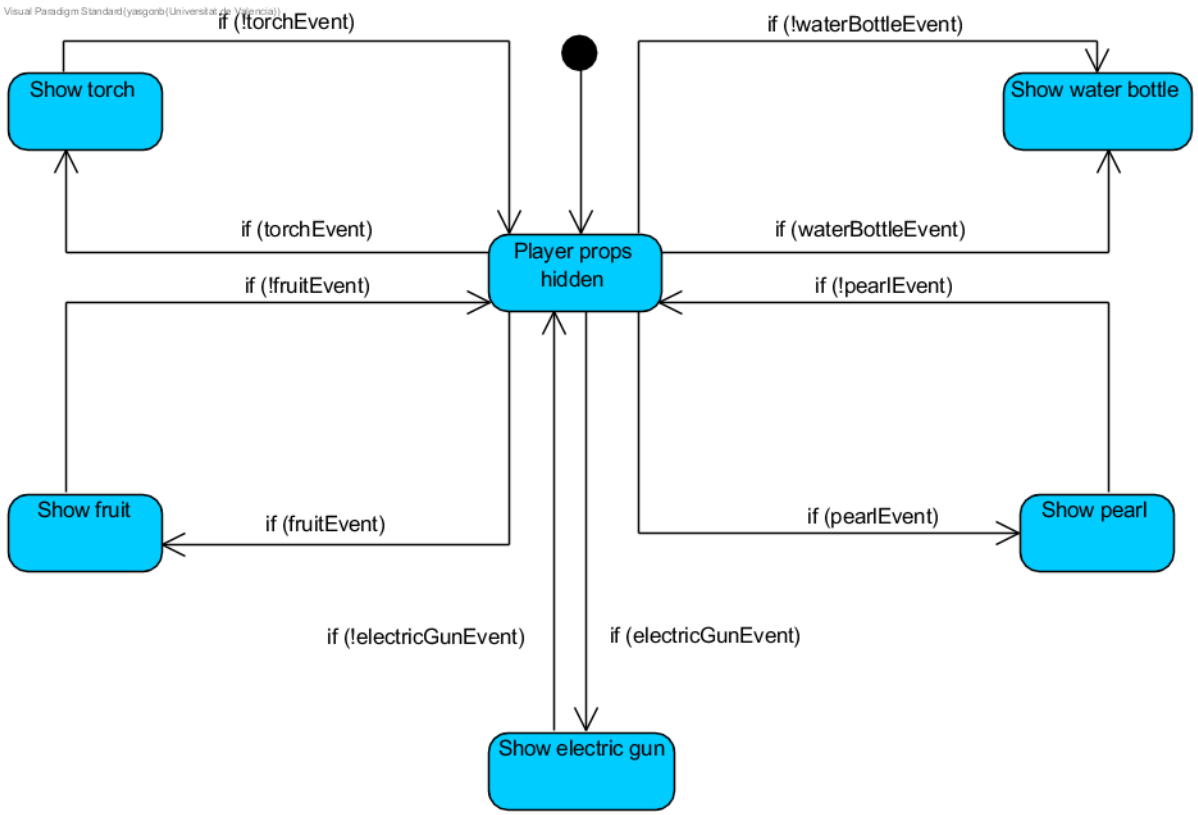


Figure 20. Player props state diagram

Now we will take a look at the characters' animator controller state diagrams. As it has been mentioned before, these diagrams are used by Unity to play the corresponding animation for each character. The video game has 4 characters, so each one of them has their own animator controller.

The first one we will be looking at is the animator controller state diagram for the doctor.

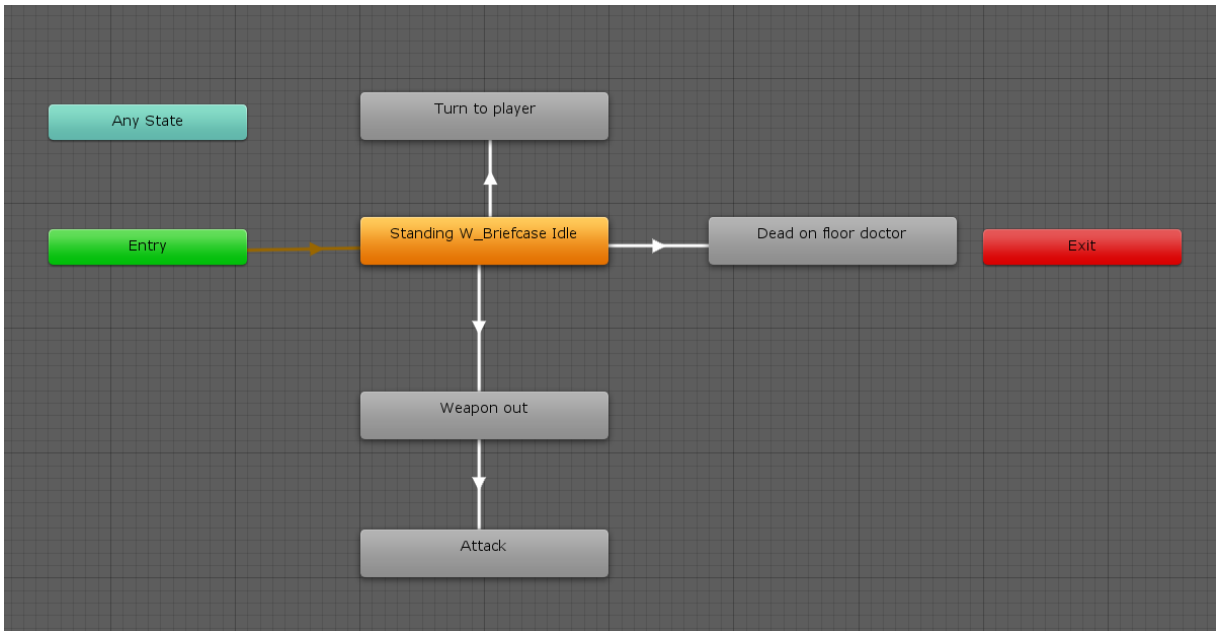


Figure 21. Doctor animator controller state diagram

The doctor animator controller uses a total of five animations. Its default state is the "Standing with Briefcase Idle" animation, which is the one that has the doctor idle standing animation that is playing most of the times.

The next character controller animator diagram we will be looking at is the one corresponding to the girl character. It is the following:

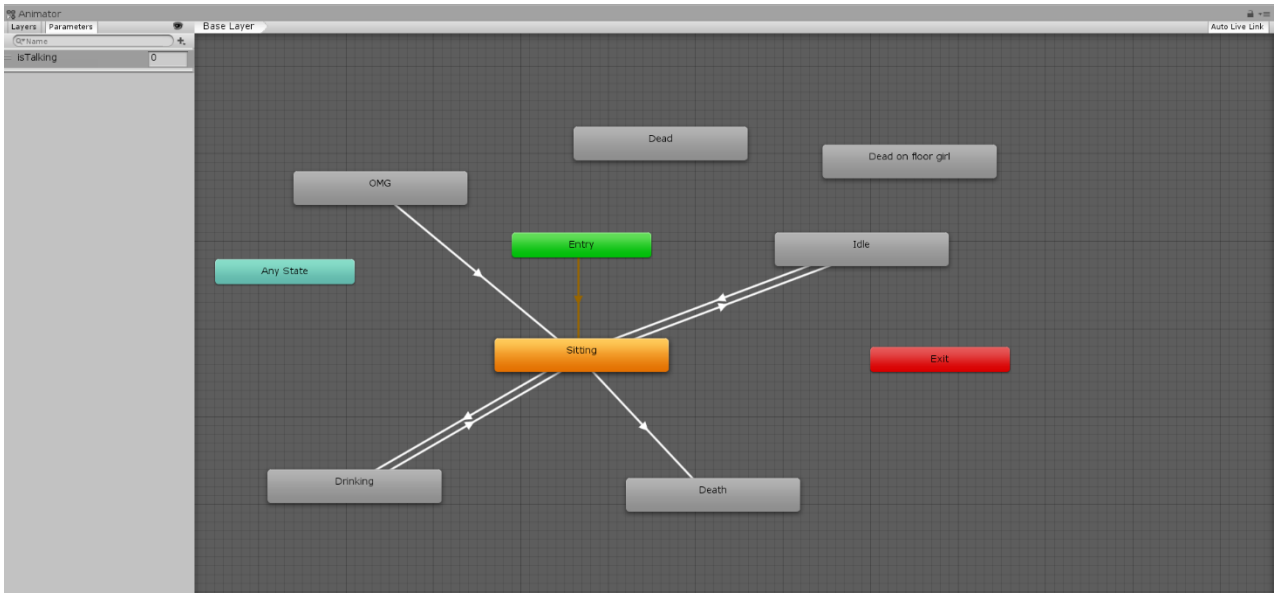


Figure 22. Girl animator controller state diagram

In this state diagram, there is a total of seven animation states and the default state for the girl character is the “Sitting” state which corresponds to the girl’s sitting animation that is playing normally. In this case, this state diagram uses a variable “isTalking”, which is controlled by script and changes the animation in case the girl is talking, in this case, to the player.

Following up, we will take a look at the woman animator controller state diagram, which is pretty similar to the girl character’s animator controller state diagram.

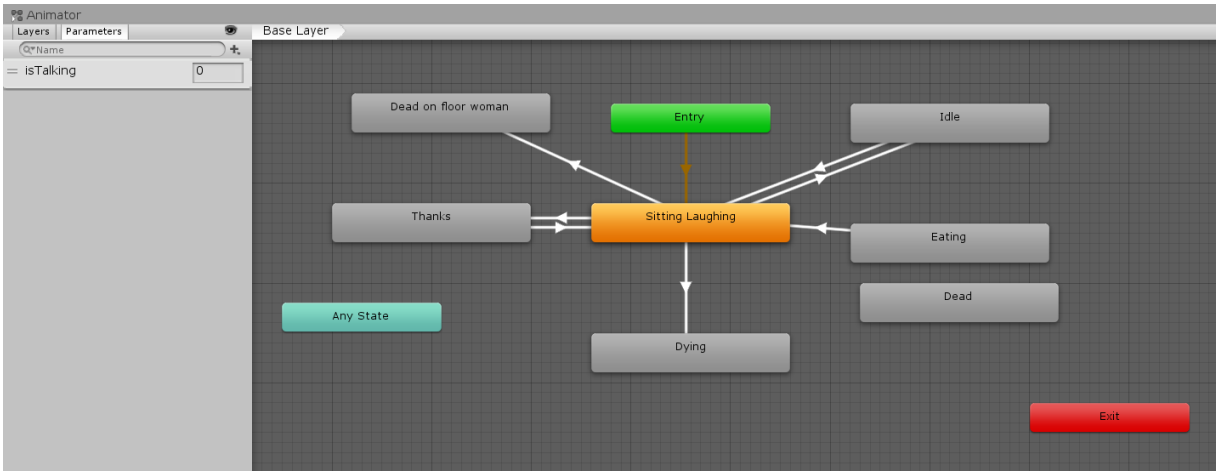


Figure 23. Woman animator controller state diagram

As we can see, both of the diagrams look really similar. In this case, the “Sitting Laughing” is the default animation state of the woman, while the “isTalking” variable, also controller by script, changes the animation playing to the “Idle” animation state once the player is interacting with the woman.

Lastly, the most relevant animator controller state diagram is the one that controls the main character’s animations. It has the following state diagram:

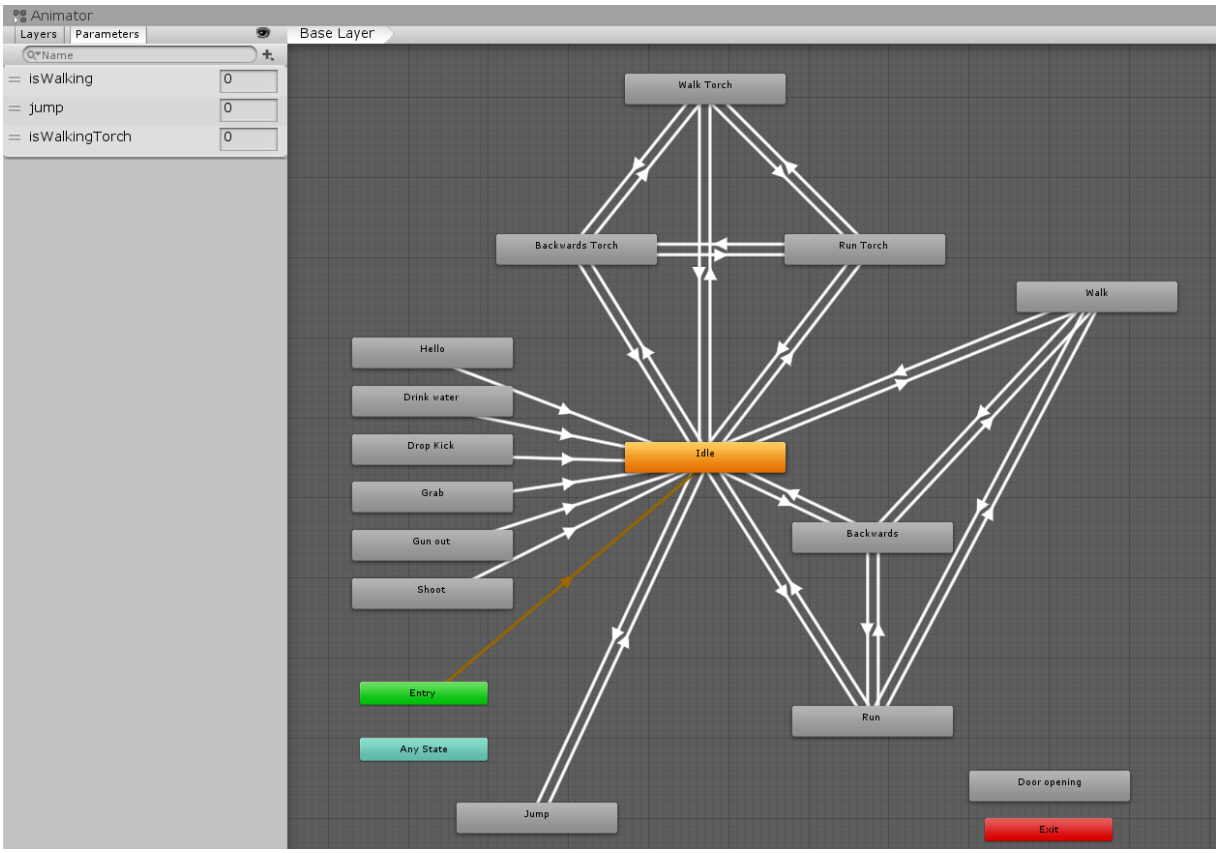


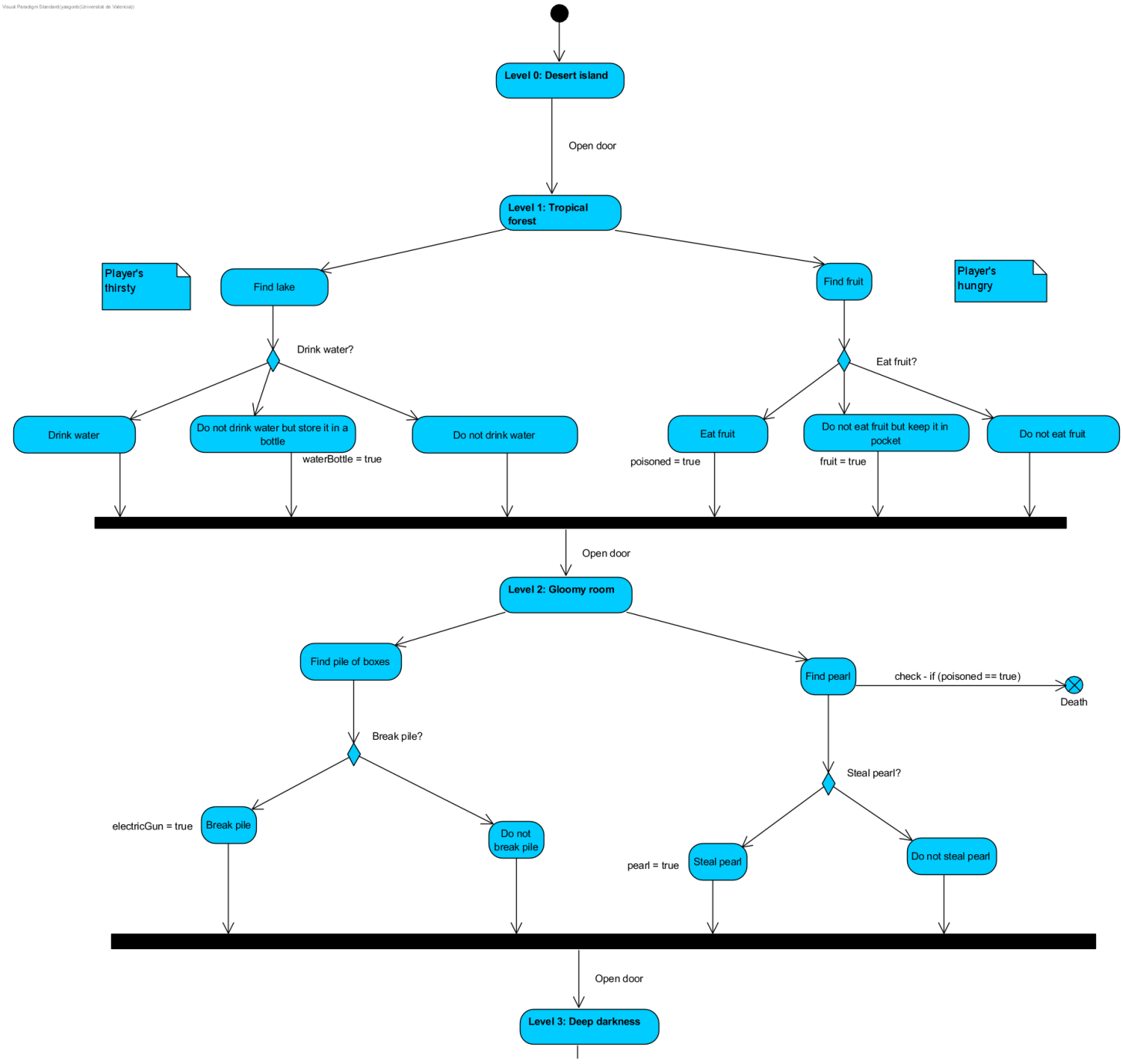
Figure 24. Player animator controller state diagram

The player animator controller state diagram is more complex than the ones we have seen before due to the fact that it has to reflect all of the several animations that correspond to the multiple events and actions that the player has to confront throughout the game. The player animator controller has a total of 15 different animation states, being all of the controlled through script. In order to do so, the animations are called via script or they change the values of some of the animator controller's parameters.

There are three parameters defined for this animator controller, one for when the character is walking, a second on for when the character is jumping and a last one for when the character is walking with a torch, which is only possible in the third level. The "isWalking" and "isWalkingTorch" parameter's value is extremely important, as its value will make the character either run (value equals 2), walk (value equals 1), walk backwards (value equals -1) or stay still (value equals 0). Each of the value will play the corresponding animation to "Run", "Walk", "Backwards", and go back to the "Idle" animation state, respectively.

6.2.2.3. Activity diagram

The last diagram we will be analyzing from this dynamic point of view is the activity diagram. This diagram pictures the game flow where we can clearly see the different outcomes of the game, accordingly to the player's decisions:



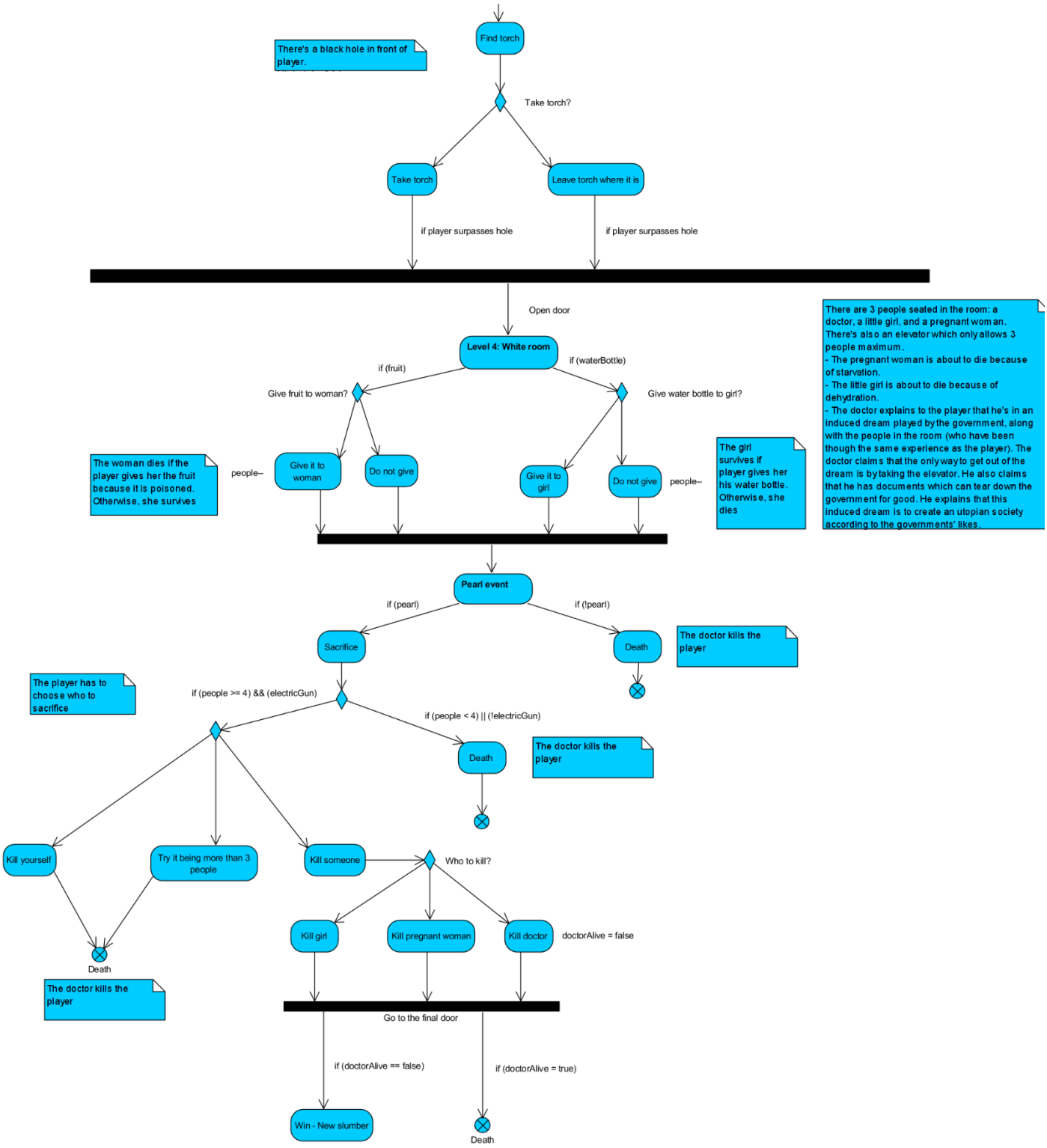


Figure 25. Game flow activity diagram

This activity diagram reflects the entirety of the game flow. In it we can see the events of each level and the different choices the player has to make. When the game itself starts, the player is taken to “Level 0”, where he only has to find the door which will led him to the next level. The following level, “Level 1: Tropical forest” has two different events, which the player will have to find and make the corresponding choices. The two events are regarding the water of the lake, and an exotic fruit. Once the player has gone through both events, the door that leads to the next level will appear.

The player will cross the door and will be taken to the next level, which is “Level 2: Gloomy room”. In this level, there are also two events, regarding a pile of boxes that contain an electric gun and a pearl placed on a table, in which the player will have to decide on. If the player has eaten the fruit back in the previous level, he will die once the pearl event triggers. Once the player plays both events and survives, the door that leads to the next level will appear.

After going through the door, the player will be taken to the next level, “Level 3: Deep darkness”. This level consists of a single event, triggered by a torch, and the player will have to find the door placed at the end of a dark corridor. However, there is a hole in the center of the corridor, so the player must be careful not to fall and get to the door. Once he does so, he will be taken to the next level.

“Level 4: White room” is the level where the player meets the other three characters, and the key to being able to win. At first, the player will have to interact with the woman and the girl, and his actions will decide if they are killed or saved. If both of them survive, then the player will be asked by the doctor if he either wants to sacrifice himself, sacrifice someone else or try going through the door being more than three people. Otherwise, the doctor will proceed to kill the player.

The only way to win the game and be able to cross the door that leads to salvation is to kill the doctor, but in order to do so, both the woman and the girl must be saved. At last, if the player manages to survive and cross the door, he will be taken to the fifth level “Win – New slumber” and, consecutively, to the main screen. If at any time the player either dies, gets kills or falls down the trap, the player will be taken to the game over screen.

6.3. Design

In the design phase we will take a look at the updated class diagram with all of the methods of the classes of our system, as well as the sequence diagrams for each one of the use cases of our system.

6.3.1. Updated class diagram

As we have mentioned, the updated class diagram is the following:

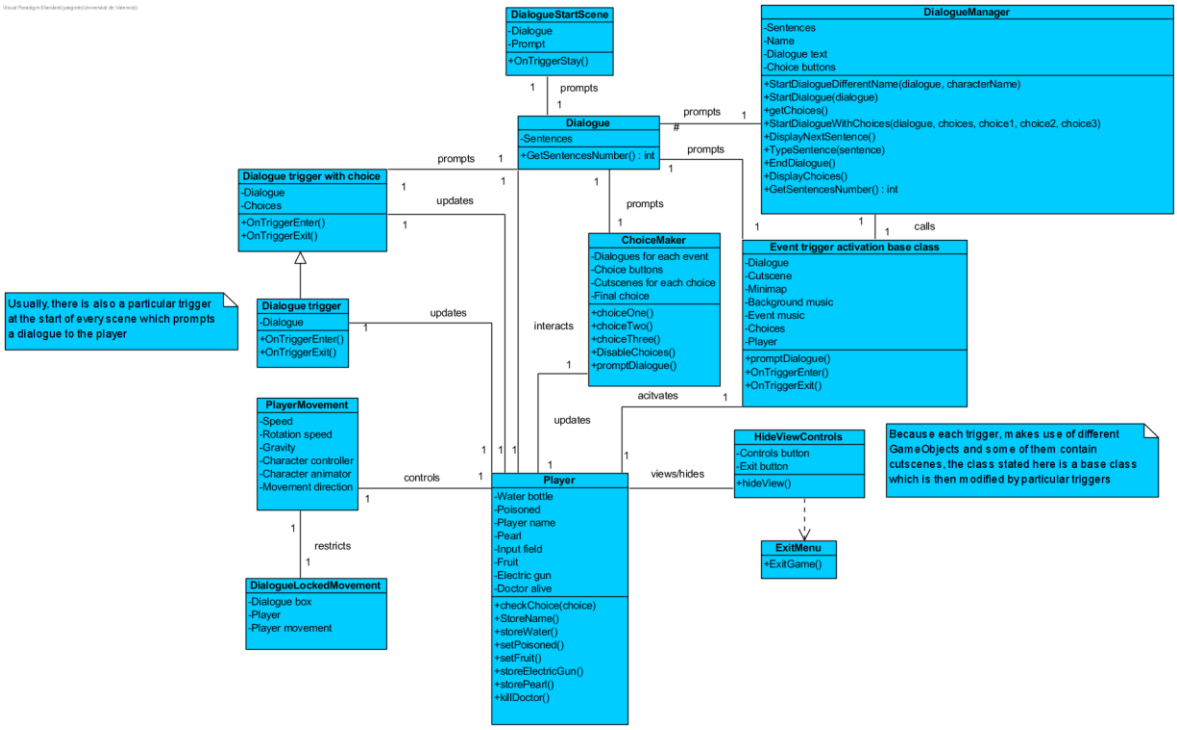


Figure 26. Updated class diagram

This class diagram now shows all of the methods for each of the relevant classes, as explained before. We will go through each of the class in order to truly understand what their role in our system is.

We will start by talking about the Dialogue class. This class has a string[] sentences attribute, and is used by the Dialogue Manager to prompt each dialogue to the player. Each dialogue in the game corresponds to a single character and has a number of sentences. The number of sentences of each dialogue can be retrieve by the use of the method GetSentencesNumber().

The class DialogueManager is the class whose role is to control the flow of the dialogues prompted. This class is called by the even trigger classes when there is the need to display a dialogue to the player and has the attributes for the sentences displayed, the name of the character who talks, the text of the dialogue and the choice buttons that the player will be interacting with. This class uses many methods, being three of them to start the dialogue with a different name for the character talking other than the actual player's name, to start dialogue with the player's name as the one who is talking, and to start a dialogue with choices for the player to make. All of the choices' dialogue end with a player's reflection on the issue, so there is no need to pass a different character name.

Some of the other methods of this class are getChoices(), which returns the choices of the dialogue, DisplayNextSentence() which is a recursive method that is called every time the player presses the continue button on the dialogue box and keeps on reading the dialogue. TypeSentence(sentence) is also another recursive method called by the class itself to show the dialogue to the player in the way that the sentences are displayed in a type-machine way, with some motion to it. The method EndDialogue() is also a recursive method that the class calls in order to end the dialogue when there are no more sentences left, and lastly, the GetSentencesNumber() is a method that returns the number of sentences of the dialogue held by the sentences attribute.

Next up, the DialogueStartScene is a class that is called when a level starts. This class has a dialogue and a prompt attribute, being the dialogue the one that is going to be displayed to the user, and the prompt, a reference to the prompt message that appears on screen. This class uses the Unity OnTriggerStay() function to do so. The Dialogue trigger is a class that displays a dialogue to the player when its trigger is activated through Unity's OnTriggerEnter() function, and that destroys itself on the OnTriggerExit() function, so it is triggered only once. The Dialogue trigger with choice is a variant of the Dialogue trigger function which implements the dialogue with choices, instead of being only the dialogue text.

The Event trigger activation base class is a class that has many variants, as it has been mentioned before, because of the use of different amount of GameObjects and mechanics involved, so it is a base class representing the different instances and variations. This class has as attributes: a dialogue, a custcene, a reference to the minimap, a reference to the background music, a reference to the particular event music, the choices for the player to make, and a reference to the player. The class uses Unity's OnTriggerEnter() function in order to activate the corresponding event's objects and mechanics, and the OnTriggerExit() function to destroy itself, so that the player does not activate the trigger more than once.

The Player class contains all of the player's variable values throughout the game, these values are extremely important and are updated as the player advances in the game and makes his decisions. Each of the attributes are related to those key objects that will decide whether a player is able to win or not, and they all have their own

method to set their state. These methods are called based upon the player's decisions in each of the corresponding event.

Next up, the PlayerMovement class is the class that allows the player to move the main character. It has attributes regarding the speed, rotation speed and gravity, which are values deliberately set so that the movement of the character is smooth and allows the player a comfortable use of the character. The class also has references to the character controller for the use of animations and movement, and the movement direction which is an input from the player. The DialogueLockedMovement is a class that has references to the dialogue box, the player's GameObject and the player's movement script, so that the player is not able to move the character while a dialogue is active on screen.

Lastly, the ChoiceMaker class is an extremely important class as it is the one in charge of updating the player's variables values when they are changed after a player's decision upon an event. This class has references to the UI buttons for each of the choices using the method for the selected choice button and once these are clicked the class prompts the corresponding dialogue and updates the player's variables. After the player has made its decision, the class disables the choice's UI buttons using the class' DisableChoices() method.

As for the design patterns that have been used in this case, taking in account that most of the object instances of the classes, the expert pattern is the most relevant design pattern that has been put in practice. In this pattern, there are several classes which are expert to a particular mechanic. For our video game implementation, most of the classes are expert in their respective gimmick. Because of this design pattern, the classes have an easy access to the variables in each case.

Another design pattern that has been applied to the classes of our system is the structural adapter pattern. Even though there has not been an interface implicitly implemented, there is a base class for the even trigger classes, which are used throughout the game. This design pattern is extremely beneficial for our case because it allows the use of the same structure for the great majority of the mentioned events, being both beneficial for development purposes while being easier to understand for the player as there is an implied pattern to follow.

6.3.2. Use case sequence diagrams

In this point we will look at the sequence diagrams for each of the use cases of our system that we analyzed in the analysis phase. We will analyze each of the use of case pictured in the use of case diagram for our system:

- **Change settings**:

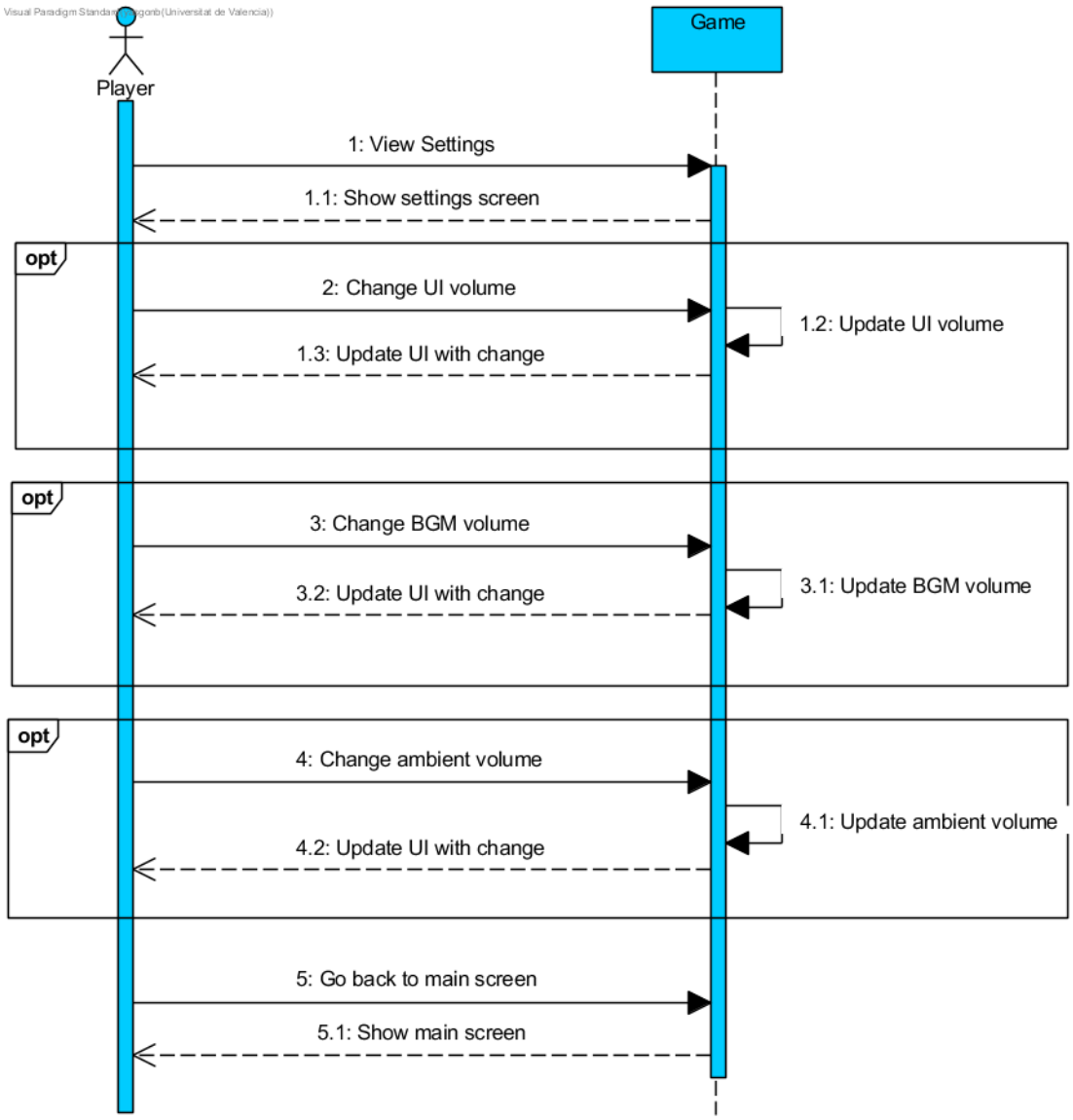


Figure 27. Change settings use of case sequence diagram

Actor	Player
Description	The player will be able to change several settings, such as the UI volume, the ambient volume, and the sound effects volume. The player will also be able to go back to the main screen.
References	FR12.
Pre-condition	Player must have launched the executable that launches the game.
Post-condition	Player will have or have not changed some settings about the game.
Regular flow	1. Player launches the game. 2. Player enters the settings screen. 3. Player changes some settings about the game. 4. Player goes back to the main screen.
Alternative flow	1. Player launches the game 2. Player enters the settings screen 3. Player doesn't change any settings and goes back to the main screen.

- **Start game:**

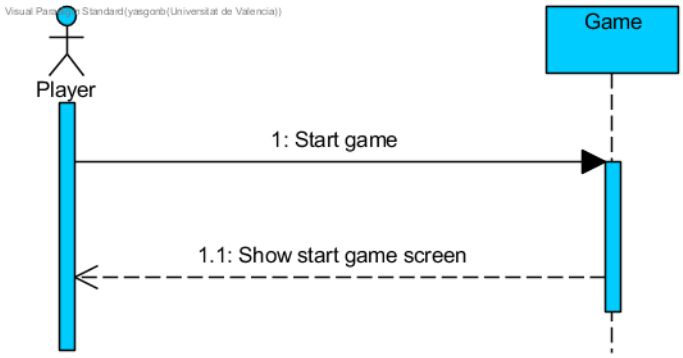


Figure 28. Start game use of case sequence diagram

Actor	Player
Description	The player starts the game and is taken to a screen where he has to enter his name.
References	FR1, FR6, FR7.
Pre-condition	Player must have launched the executable that launches the game and entered his name.
Post-condition	Player will have started the game.
Regular flow	1. Player launches the game. 2. Player is taken to a screen where he has to enter his name.
Alternative flow	1. Player launches the game. 2. Player quits the game.

- **Enter name:**

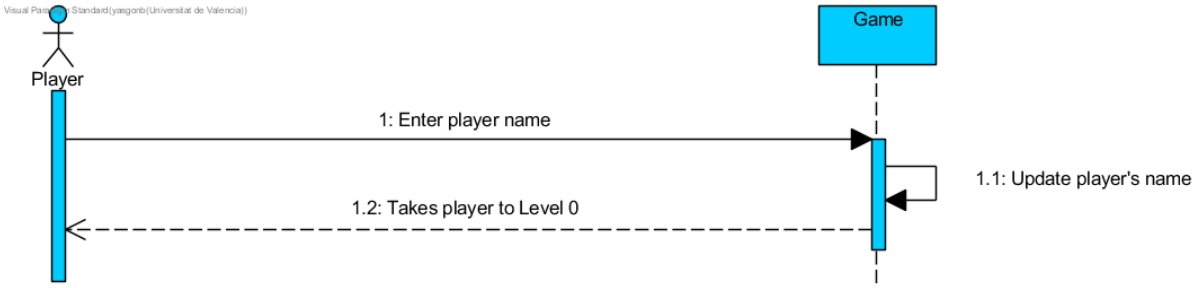


Figure 29. Enter name use of case sequence diagram

Actor	Player
Description	The player enters his name and is taken to the first level of the game.
References	FR3, FR8, FR9, FR11.
Pre-condition	Player must have launched the executable that launches the game and entered his name.
Post-condition	Player will have started the game.
Regular flow	1. Player enters his name. 2. Player is taken to the first level of the game.
Alternative flow	1. Player does not enter his name. 2. Player goes back to the main screen and exits the game.

Alternative flow 2	1. Player enters his name. 2. Player goes back to the main screen and exits the game.

- **Exit game:**

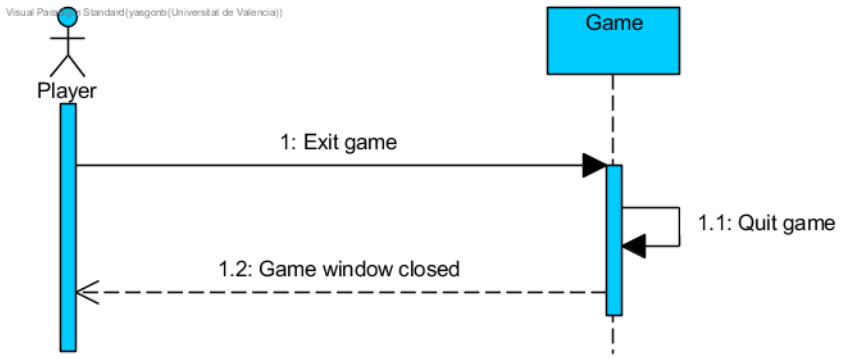


Figure 30. Exit game use of case sequence diagram

Actor	Player
Description	The player clicks the exit button and quits the game.
References	FR10.
Pre-condition	Player must have launched the executable that launches the game.
Post-condition	Player will have quit the game.
Regular flow	1. Player launches the game. 2. Player clicks the exit button. 3. The game window is closed.
Alternative flow	None.

- **Play game:**

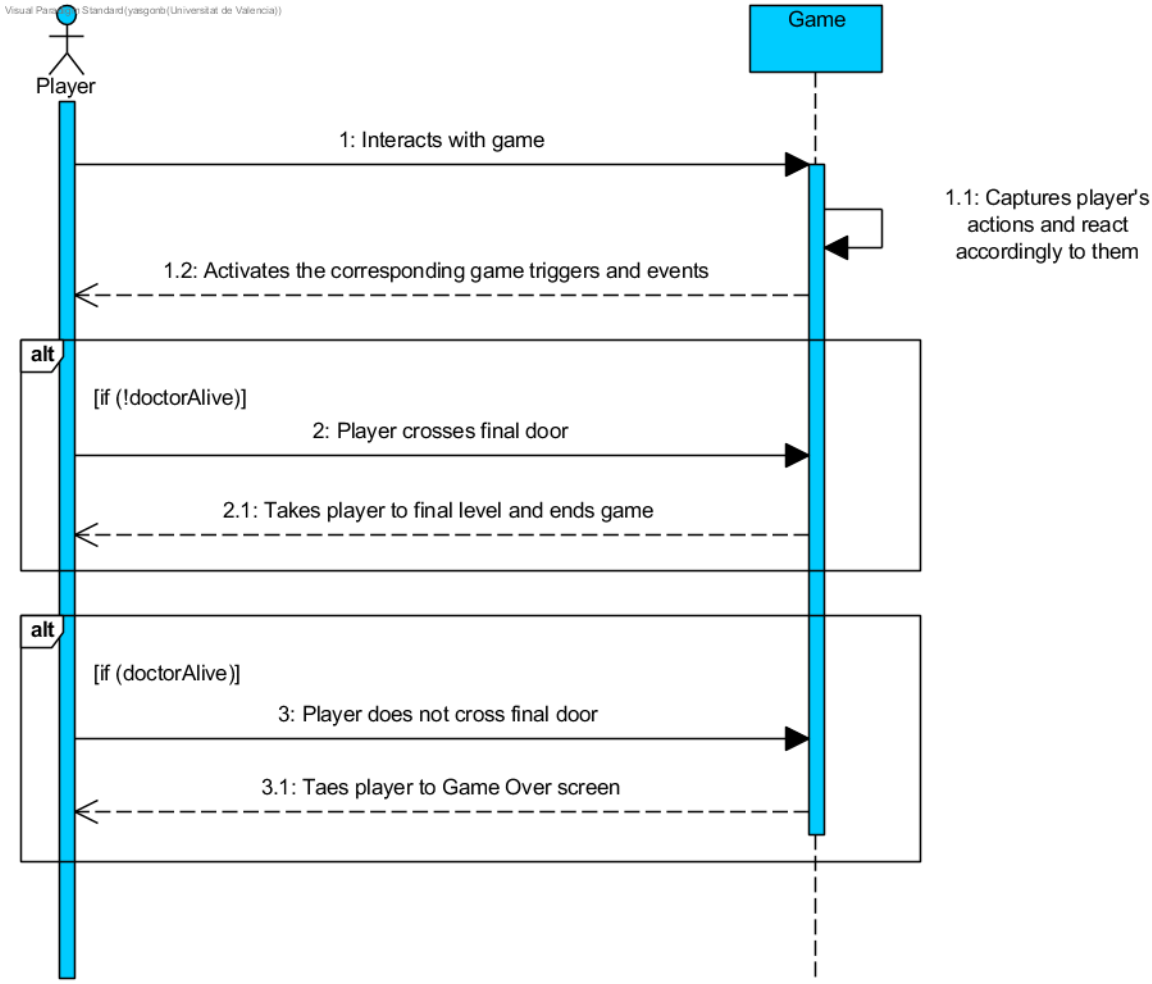


Figure 31. Play game use of case sequence diagram

Actor	Player
Description	The player interacts with the game and advances through the levels.
References	FR1, FR3, FR4, FR5, FR8.
Pre-condition	Player must have launched the executable that launches the game and have entered his name.
Post-condition	Player will be advancing through the game.
Regular flow	1. Player plays the game and interacts with the environment and the UI.
Alternative flow	None.

- **Win game:**

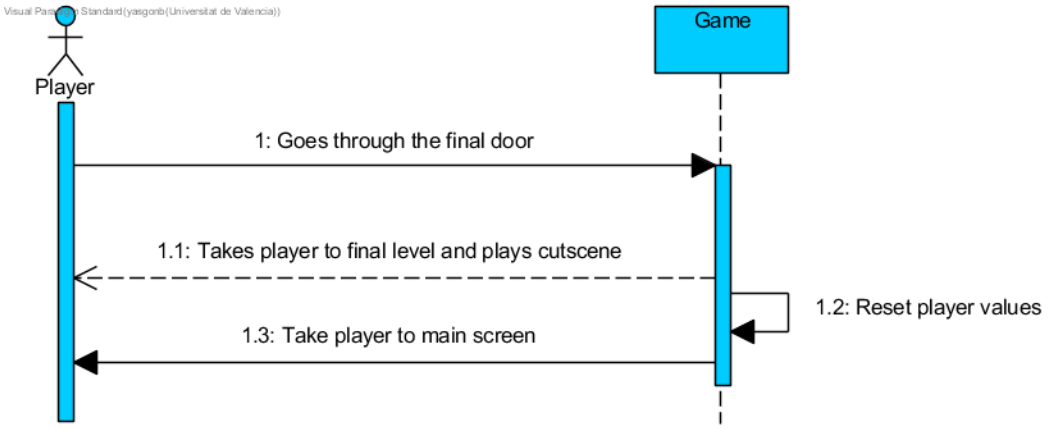


Figure 32. Win game use of case sequence diagram

Actor	Player
Description	The player manages to survive and win the game.
References	FR1, FR3, FR4, FR5, FR8.
Pre-condition	Player must have chosen the correct choices and not have fallen in the hole set in the fourth level.
Post-condition	Player will have won the game, seen the last cutscene and is taken to the main screen.
Regular flow	1. Player plays the game and survives. 2. Player is taken to the last level, where the cutscene is played. 3. Player is taken to the main screen.
Alternative flow	1. Player loses the game.

- **Lose game:**

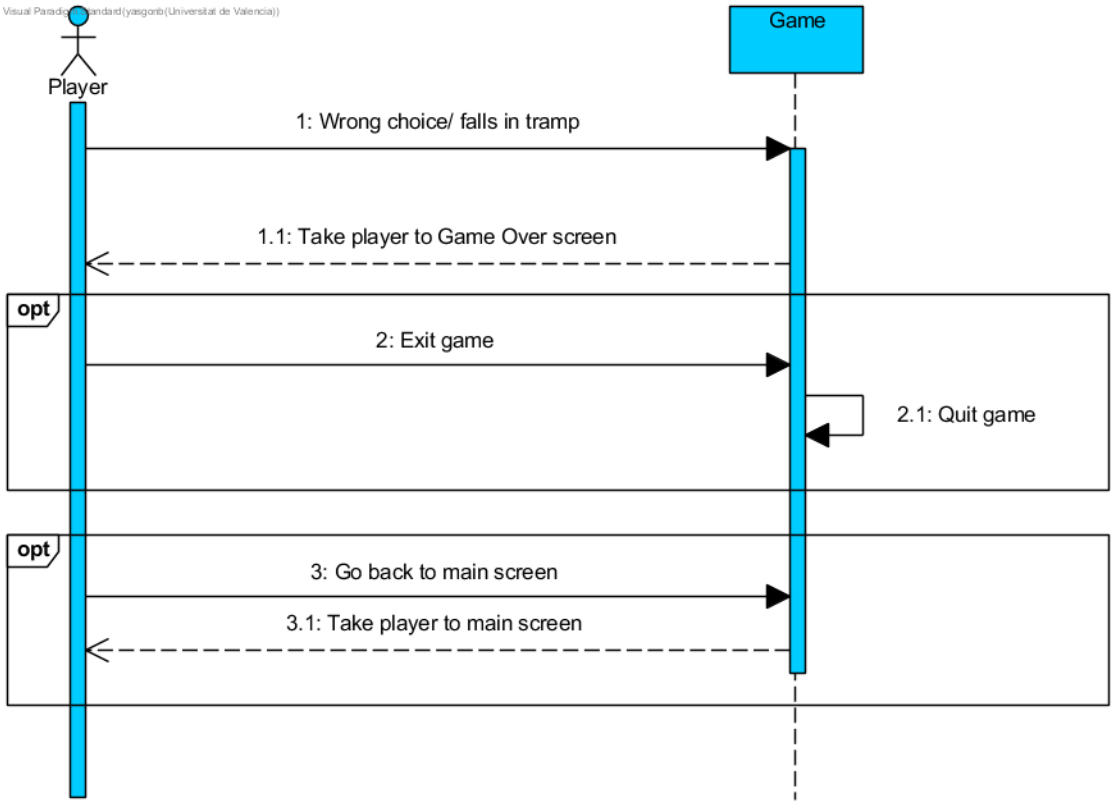


Figure 33. Lose game use of case sequence diagram

Actor	Player
Description	The player interacts with the game and advances through the levels.
References	FR1, FR3, FR4, FR5, FR8.
Pre-condition	Player must have launched the executable that launches the game and have entered his name.
Post-condition	Player will be advancing through the game.
Regular flow	1. Player plays the game and dies or gets killed. 2. Player is taken to the main screen.
Alternative flow	1. Player survives and makes the right decisions.

6.4. Implementation

6.4.1. Introduction

As for the implementation of the video game in Unity, we will go over some relevant aspects and for that matter, and also, we will take a look at the tools and assets that have been use for the development of the video game.

For starters, the video game is composed of 8 Unity scenes, which correspond to different levels and screens. There are scenes for the main menu screen and the game over screen, scenes for each of the levels where the player will be able to interact with the environment, and the last level which is the scene where the player is taken if he wins. These can be seen in the picture below:

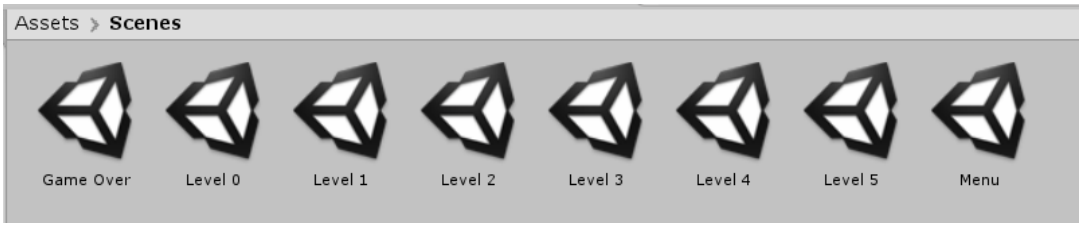


Figure 34. Unity scenes that make up the game

It is also worth mentioning that all of the scenes have been designed by myself with the use of free GameObject prefabs provided by the Unity Asset Store, as well as the models and animations provided by Mixamo[30]. The same applies to the sound effects and music used for the levels, which have been extracted from royalty free websites such as FreeSound[31] and Freeplay Music[32] for zero cost. While choosing the music, each of the levels has their own background music playing, which is suited to enhance the tone of each level. Due to the fact that there has been no money investment on any asset for the development, there has obviously been some limitations when it came to the assembly and composition of the game, however I tried my best to use the assets that would showcase what I had in mind while designing the game. Also, all of the levels use real-time lighting.

Another relevant aspect of the game is that, in order to enhance the immersion feeling on the player, the game has to be played all through at once. This is a decision based upon the goal of having the player experience the game in real-time so that he can not pause the game, leave and then resume where he left it, but rather experience the adventure from start to finish. This way the player does not have the option to think about his actions and decisions off-gameplay, but is forced to do so while playing the game. Nevertheless, the player has the possibility to exit the game at any time, but he will have to start over the game the next time he plays it.

[30] Mixamo, www.mixamo.com/.
[31] Freesound, freesound.org/.
[32] "Freeplay Music: Welcome: The Best Music Library on the Planet!", freeplaymusic.com/.

Now we will go through each of the scenes that make up the game to cover the most important tools that have been used in every one of the scenes.

6.4.2. Main and Game Over screens



Figure 35. Game Over screen

The Game Over scene corresponds to the Game Over screen. This screen is shown to the player every time he loses the game by dying, being killed or falling down the trap. This screen allows the player to either go back to the main menu or exit the game. In addition, when the player clicks the button to go back to the main menu, the player's variables are reset.

As we can observe in the picture above, the Game Over scene is composed of the main camera, the canvas which holds the UI elements, the EventSystem prefab necessary for the user interaction with the UI, the player's variables prefab, and the corresponding background music. The canvas game object implements the text "Game Over" and the two buttons to go back to the main menu and exit the game, respectively.

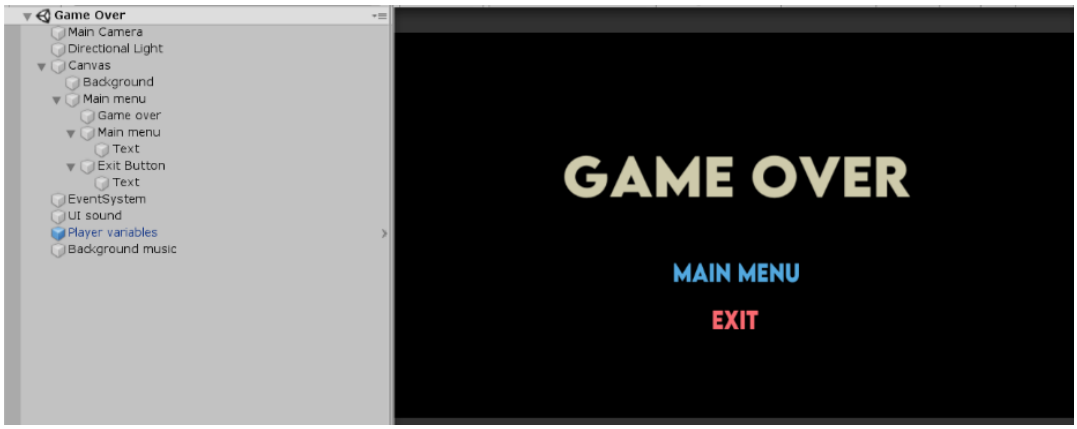


Figure 36. Hierarchy of the Game Over scene



Figure 37. Main menu screen

-Next, we will take a look at the Menu scene, which corresponds to the main menu screen. This screen is the one that is first shown to the player when the game is launched. As it has been mentioned before, it allows the player to either exit the game, change some game settings or start the game.

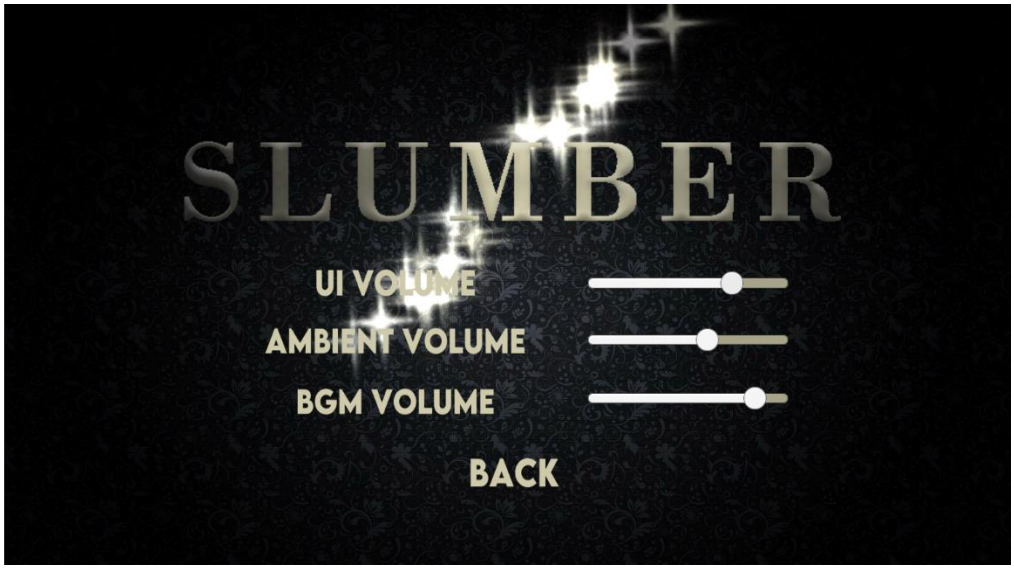


Figure 38. Settings screen

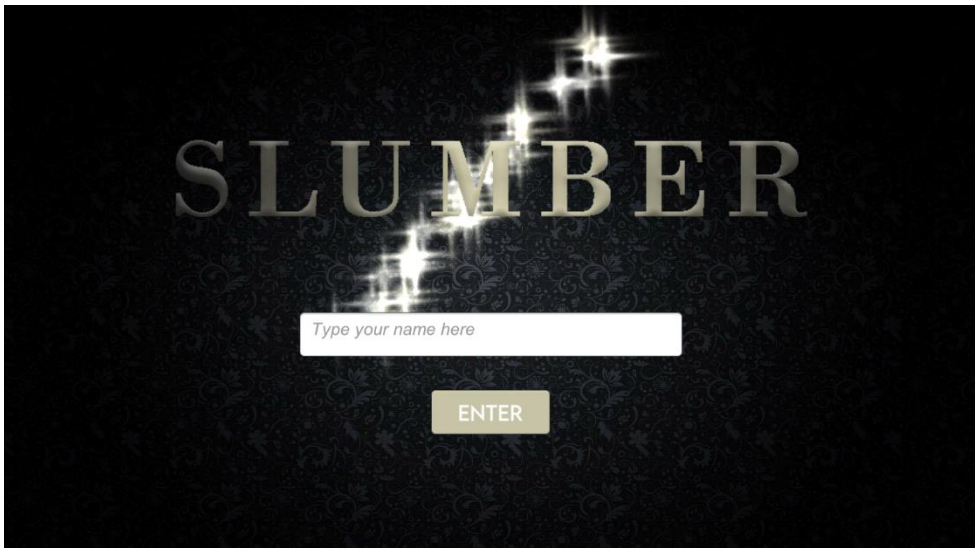


Figure 39. Start game screen

As pictured above, the settings screen allows the player to change the volume of different aspects of the in-game sounds, as well as going back to the main screen.The start game screen asks the player to enter his name, which will be used throughout the

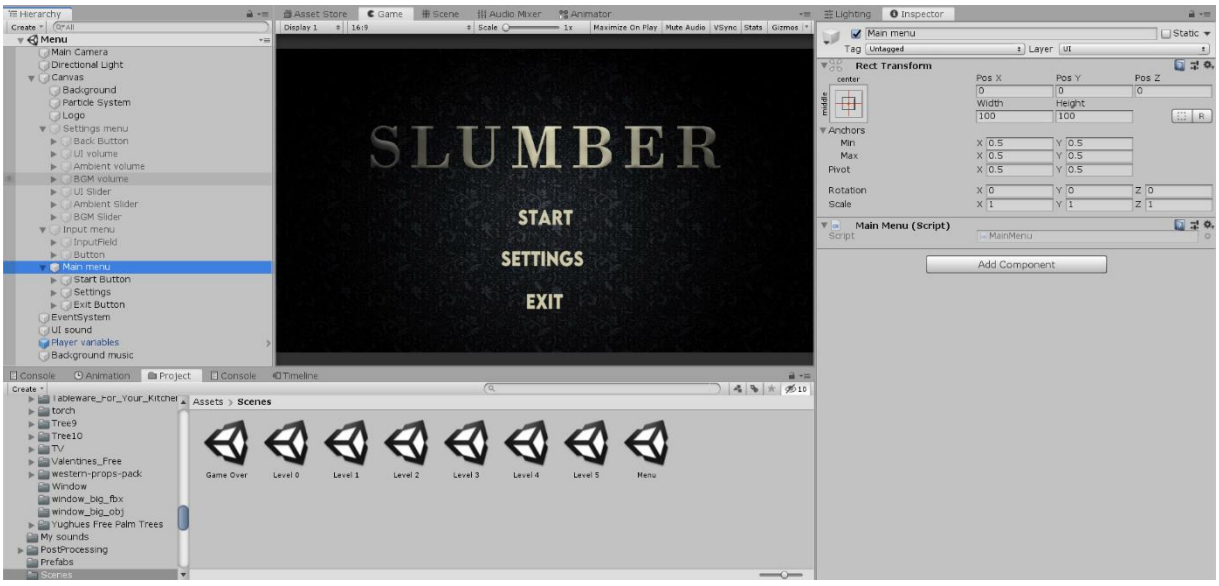


Figure 40. Menu scene

game by the dialogue manager. Once the player clicks the enter button, he will be taken to the first proper level of the game.

The Menu scene is composed by the main camera, two game objects for the clicking sound of the buttons and the background music, and a canvas GameObject. The canvas has a background picture with a particle system that gives a sparkling effect to the screen, and the logo of the video game. This scene uses three buttons for the different possibilities the player has. If the player presses the start button, the main

screen's buttons will be hidden, and the start game screen will be shown to the player. The same applies to the settings screen. This is controlled via script.

6.4.3. Common prefabs

Before continuing with the rest of the scenes of the game, it is relevant to take a look at the common prefabs prefab for the rest of the scenes of the game. We will start explaining the the player prefab and its controller. The character that the player will be using all throughout the game is the following:

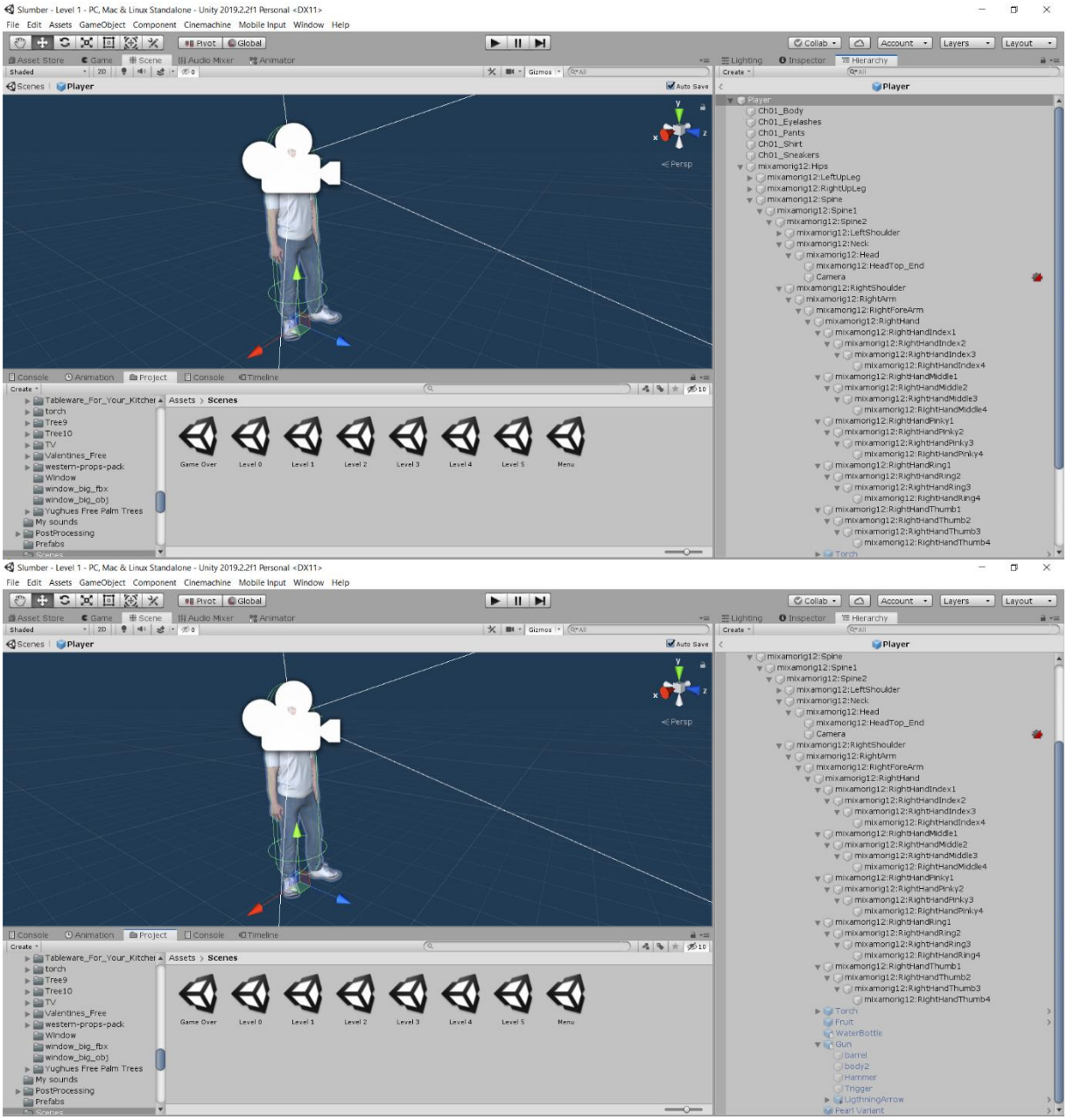


Figure 41. Prefab for the player's character

The player's character prefab is made up of numerous bones, which are used by the animations. What is important implementation-wise is that the player has a camera attached to its head. This is in order to create a realistic effect as the camera captures the player's head movement when it moves, adding up to the immersion that we aim to create for the player. This camera component has a Cinemachine script attached to it, so it can be the brain that allows the use of virtual cameras for the cutscenes that are played during the game.

The camera GameObject has a Post-Processing profile attached to it. The Post-Processing profile is a filter that is applied to the camera in order to alter the look of the game view of the given player camera. In our case, the Post-Processing profile set for the camera relies on the use of anti-aliasing for a clearer view of the game, and some other features, such as fog, ambient occlusion and color grading, that aim to enhance the mysterious vibe of the game. Lastly, the player also has attached to its hand the several props that may be used based on the player's decisions. These props are hidden and will be activated and visible in the appropriate case.

Regarding the player's movement, as we have explained before when looking at the system classes, the PlayerMovement script is in charge of moving the player according to the player's input. It is also worth mentioning, that the game uses the player's keyboard input to move the player, while the interaction with the UI elements is done with the player's mouse. The PlayerMovement is attached to the player character's prefab.

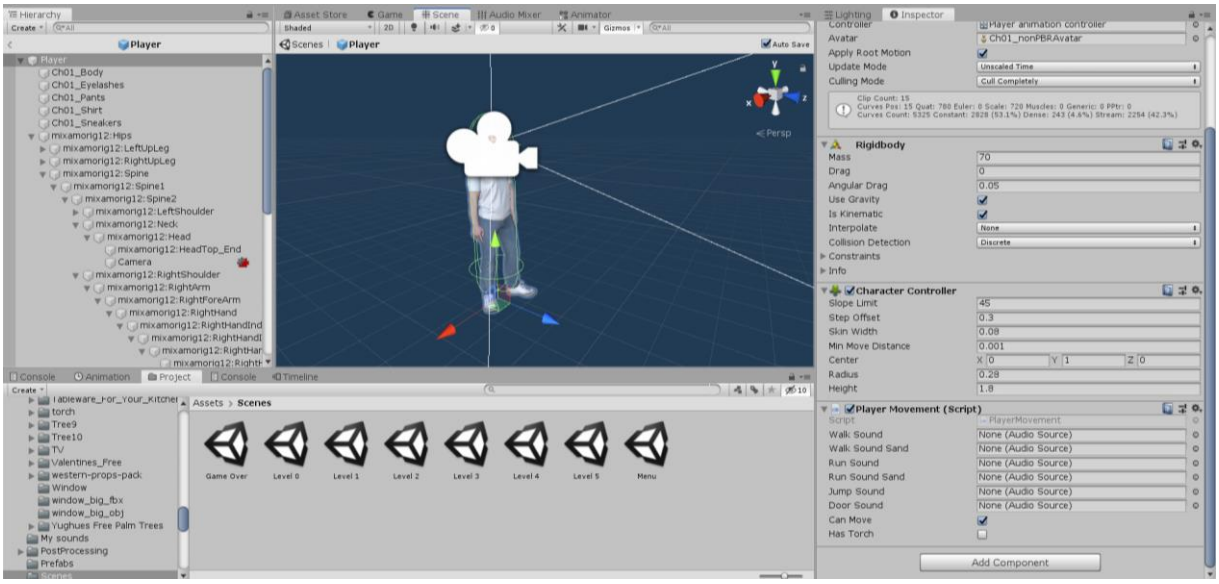


Figure 42. Prefab for the player's character with the PlayerMovement script attached

```
// Update is called once per frame
void Update()
{
    if (canMove){
        if (controller.isGrounded){
            if (Input.GetKey(KeyCode.W)){
                if (hasTorch){
                    anim.SetInteger("isWalkingTorch", 1);
                } else {
                    anim.SetInteger("isWalking", 1);
                }

                moveDir = new Vector3(0, 0, 1);
                moveDir *= speed;
                moveDir = transform.TransformDirection(moveDir);

                if (SceneManager.GetActiveScene().name == "Level 0"){
                    if (walkSoundSand.isPlaying == false)
                    walkSoundSand.Play();
                } else {
                if (walkSound.isPlaying == false)
                    walkSound.Play();
                }
            }
            if (Input.GetKeyUp(KeyCode.W)){
                if (hasTorch){
                    anim.SetInteger("isWalkingTorch", 0);
                } else {
                    anim.SetInteger("isWalking", 0);
                }

                moveDir = new Vector3(0, 0, 0);
            }
            if (Input.GetKey(KeyCode.S)){
                if (hasTorch){
                    anim.SetInteger("isWalkingTorch", -1);
                } else {
```

Figure 43. Snippet of the PlayerMovement script

The second prefab we will look at is the canvas prefab.

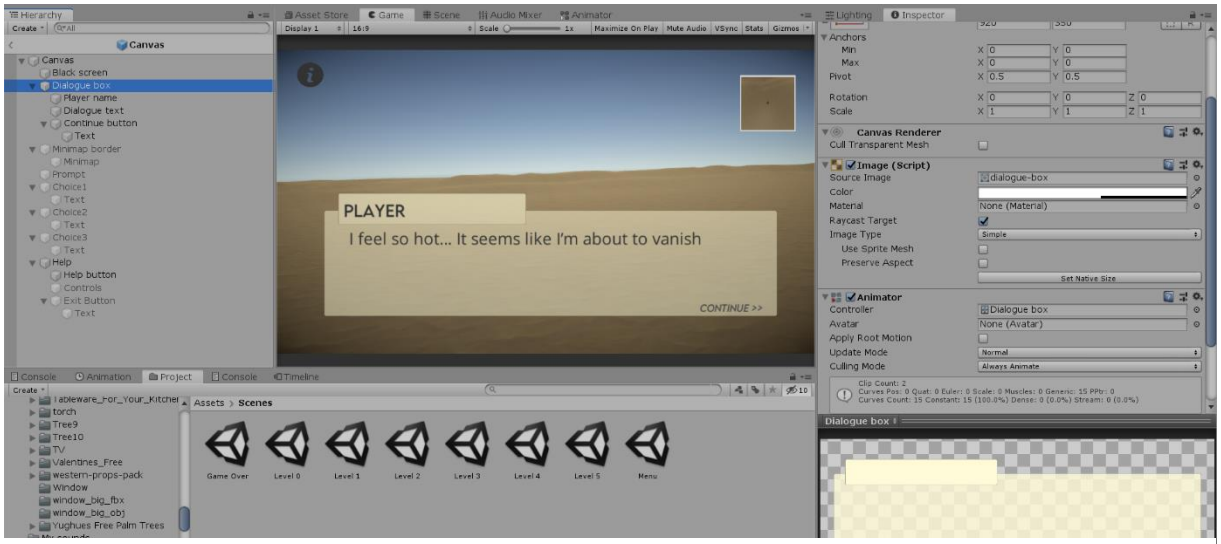


Figure 44. Canvas prefab

This prefab contains all of the useful UI elements for the player during gameplay. The Dialogue box prefab has includes a black layer that has an animation that plays when changing scenes as a fade out effect and the dialogue box where the dialogues will be shown to the player. The Dialogue box prefab is made of a dialogue box texture which has been made using Photoshop and the text objects where the dialogued text is displayed, as well as the player's name, and the continue button that are clicked by the player to display the next sentences. This button is linked to the DialogueManager class, which we explained previously.

A snippet of the DialogueManager script that manages the method to display the next sentence and the way the text is shown to the player can be seen below:

```
public void DisplayNextSentence(){
    if (choices){
        if (sentences.Count == 1 && choices){
            DisplayChoices();
        }
    }

    if (sentences.Count == 0){
        EndDialogue();
        return;
    }

    string sentence = sentences.Dequeue();
    StopAllCoroutines();
    StartCoroutine(TypeSentence(sentence));
}

IEnumerator TypeSentence (string sentence){
    dialogueText.text = "";
    foreach (char letter in sentence.ToCharArray()){
        dialogueText.text += letter;
        yield return null;
    }
}

void EndDialogue(){
    animator.SetBool("isOpen", false);
}

void DisplayChoices(){
    choice1Button.SetActive(true);
    choice2Button.SetActive(true);
    choice3Button.SetActive(true);

    choice1String.text = choice1Text;
    choice2String.text= choice2Text;
```

Figure 45. Snippet of the DialogueManager script

Another prefab that is included in the canvas prefab is the the Minimap prefab. This minimap consists of a small square that can be seen at the top right of the screen. This minimap shows the player's position at all times, and act as an aid to the player when it comes to exploring the environments.

Lastly, the canvas prefab also contains the Help prefab which is composed of a help symbol sprite that is positioned at the top left of the screen. This prefab is available to the player at all times, and when clicked, it shows the following picture which overlays on top of the player's view:

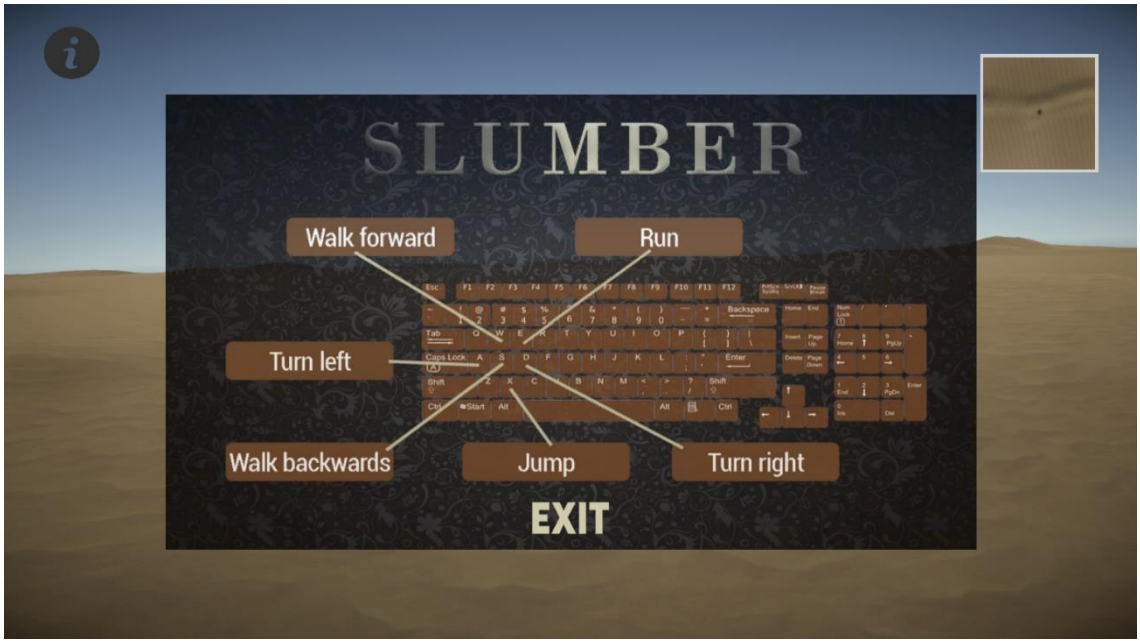


Figure 46. Help picture overlay

This picture acts as a way to help the player know the keys that allow him to control the player's movement by using the keyboard, as well as allowing him to quit the game. As the game is made to be played more than once due to its intended difficulty at first, I decide that it was best to have this as a help to the player and not include any kind of tutorial level or something of the like. This way, the player can simply look at the keys for the player movement the first time he plays the game or if he forgets about them, and not have to go through a tutorial level every time he plays since the beginning.

The other common prefab to all of the gameplay levels is the Meta prefab.

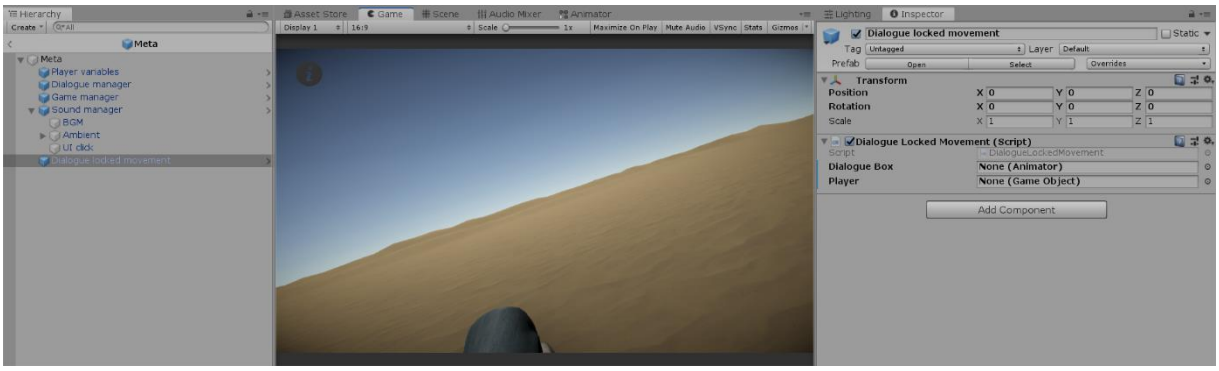


Figure 47. Meta prefab

The Meta prefab contains some of the most key prefabs of the system, which allow the video game flow to progress as expected. First of all, the Player variables prefab is a GameObject that has the script for the Player class, which contains all of the values for the player variables, and that we have explained before. After that, is the Dialogue manager Game Object which has the DialogueManager script attached to it and is called every time a dialogue is prompted to the player.

The Game manager prefab is the one that prompts the first cutscene and afterwards dialogue in each level. The Sound manager prefab is the one in charge to play the corresponding background music for each level and also has GameObjects for each of the necessary sound effects for the game. Lastly, the Dialogue locked movement prefab is a GameObject that has the DialogueLockedMovement script attached to it. As we have explained before, it restricts the player to be able to move its character while a dialogue is on screen.

6.4.4. Gameplay levels

In this point we will go over each one of the levels that compose the game, pointing out the most relevant aspects of each. First, it is important to note that while designing the levels, I have tried to give each level a special tone to it so that the player is also entertained while playing the game and, as a whole, enriches the player's experience with *Slumber*.

The first scene that the player is taken to after entering his name is Level 0 which is the scene that corresponds to the first actual level of the game "Level 0: Desert island". This level is referred as Level 0 as it does not represent any kind of challenger or a decision to make, but rather, the player is conducted to walk straight forwards until he finds the door that takes him to the next level. This level is necessary as it sets a path to the player, while introduces him to the game's gimmicks.

The level looks like the following in the Unity editor:

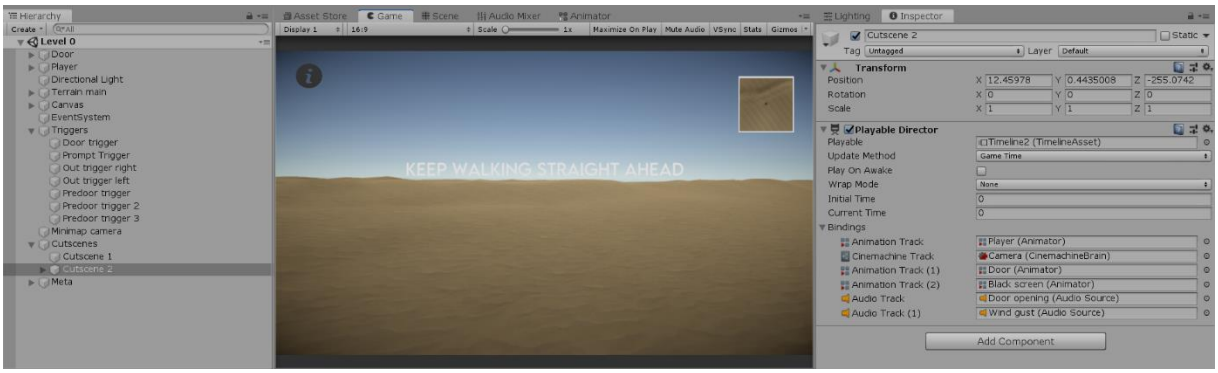


Figure 48. Level 0 as seen in the Unity editor

As we can see in the picture above, the scene contains several triggers that conduct the player to walk straight forward until he reaches the door. The level has been designed with a terrain and with the use of Unity terrain tools for the terrain bumps and the sand textures, we have been able to recreate a desert. Also, to make sure the player goes the right way and finds the door, apart from the dialogue and prompts in the way, we have also included triggers that prompt an on-screen message if the player goes out of the correct way.

Another thing to notice is that this level has a different walking sound for the player than the rest of the levels, as we have considered the fact that the player is walking on sand. Once the player reaches the door, he is prompted to press a key and then a cutscene plays, followed by the player being taken to the next level. The picture below is an in-game screenshot of that moment:



Figure 49. Screenshot of the game with showing a prompt to the player

After the player presses the ‘Q’ key, a cutscene is played:

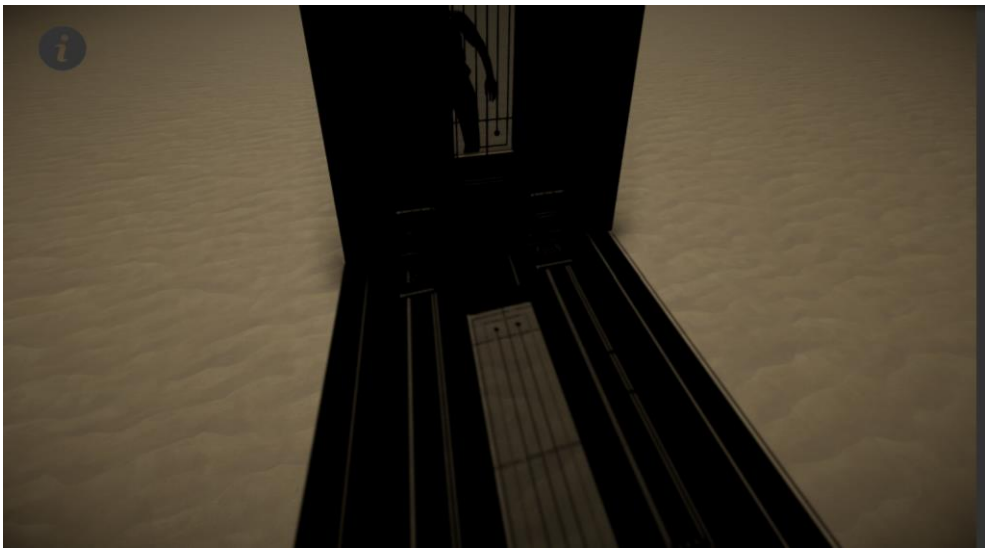


Figure 50. Screenshot of the game while the cutscene is playing

In order to play the cutscene, Unity’s Timeline as well as the Cinemachine package have been used. In the following picture we can see the Timeline window and the several tracks that play in this particular cutscene:

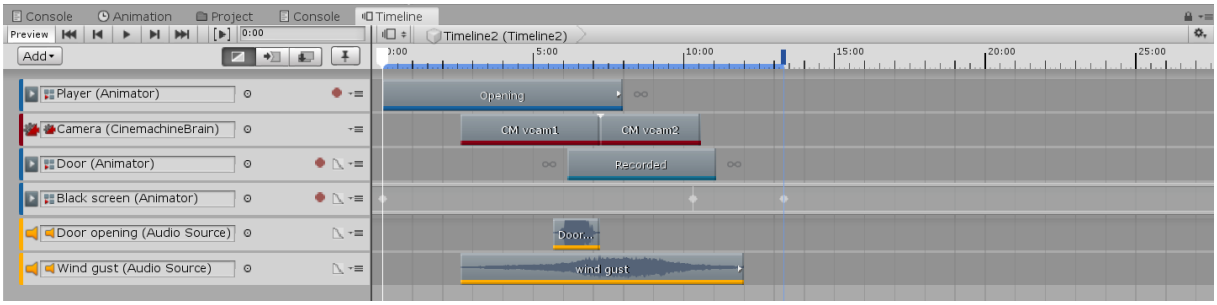


Figure 51. Timeline window for the Door opening cutscene of Level 0

As we have explained before, the Timeline tool allows the developer to place different tracks so that they are played once activated via script. In our case, the trigger that plays the cutscene was activated when the player pressed the ‘Q’ key. As we can see picture above, this cutscene uses two audio tracks for the door opening and wind gust sounds, as well as animation tracks for the player opening the door, the door opening and the black screen that acts as a transition screen to the next level. At last, a Cinemachine track is also included in the timeline. This track makes use of two virtual cameras which are positioned in different positions and change the view of the player while the cutscene is playing. The use of these virtual cameras give an immense improvement visual-wise, as it is truly enriching to the cutscene to have a camera shots from different camera angles playing in real-time.

The next scene we will be exploring is the corresponding to the second proper level, or “Level 1: Tropical forest”. This scene is the biggest and heaviest visual-wise, making use of three Unity terrains. Below are various screenshots of the editor view of the level:

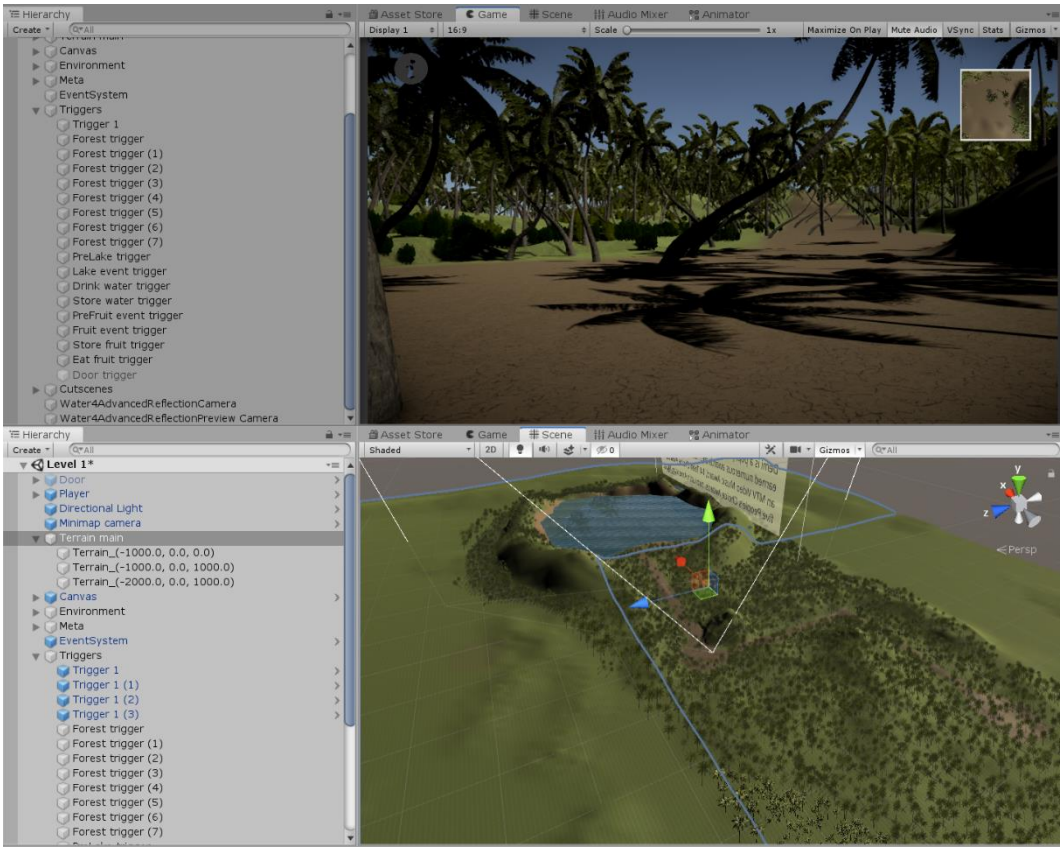


Figure 52. Editor window showing Level 1 Game and Scene views, as well as a part of the hierarchy view

The design of the level for this scene has made heavy use of the Unity terrain tools, as well as the Water prefab available for free in the Asset store. The Unity terrain tool has been used to give the level the feel and look of a tropical forest. In order to do so, I have used the tool to make the adequate bumps in the terrain, as well as placing numerous trees all around the level. The number of trees included in the game have been carefully adjusted so that there is a balance between the visual quality of the game and the cost of it, so that it meets the requirement of the video game being able to be played by lower-end computers. This is of extremely importance as all of the trees and nature details, such as bushes, are rendered in real-time and have wind applied to them. This also means that the shadows are being calculated in real-time so the field of view for the trees rendering has been thoughtfully taken care of.

This is what the Terrain window, as well as the different assets used for the implementation of the nature, looks in the Unity editor:

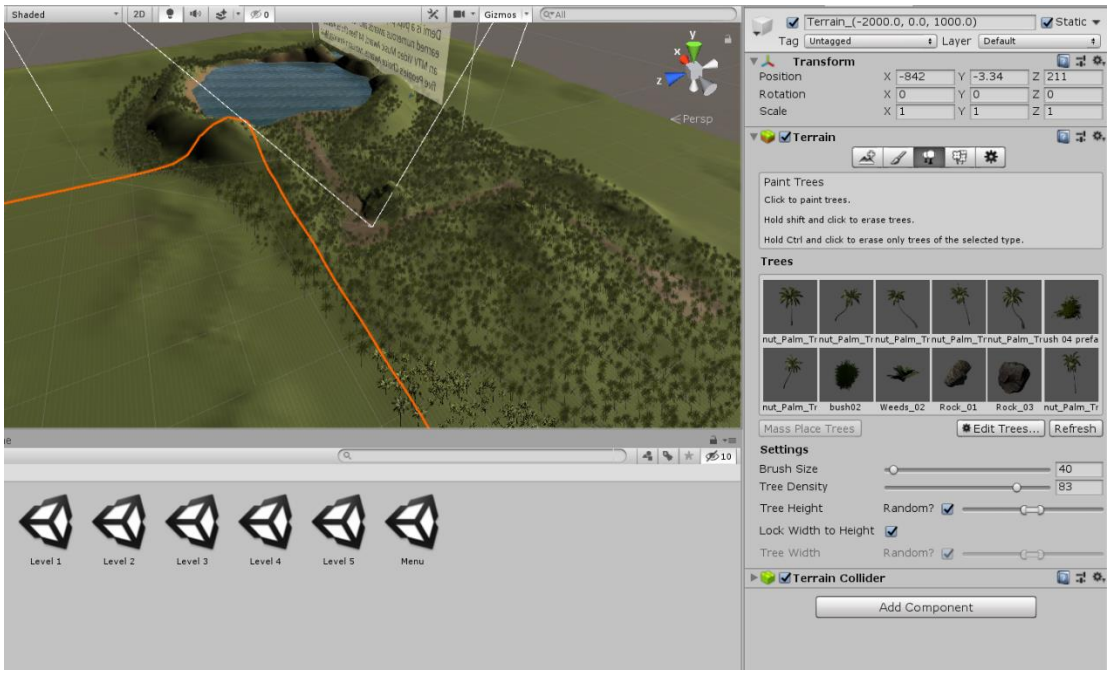


Figure 53. Terrain window in Level 1

As for the Water prefab its variables have been adjusted so that it looks like an actual lake, mainly using the Fresnel waves available with the prefab.

This level implements two different events until the door to the next level shows. This is taken care of by the DoorAppearManager GameObject which has a script that manages its appearance once both of the events are played by the player and the events can be made in the other the player prefers. The events use a Box collider as a trigger when the player makes contact with them. This mechanic is used for the triggering of all of the events in the game.

One of them is the Lake event, which prompts a cutscene and then a dialogue where the player is asked to make a decision. In this case, the cutscene also uses the Timeline and Cinemachine tools, but this time as the camera has to go back to the player, a auxiliary camera has been assigned to the player as well as a Camera management prefab that changes between the cameras once the cutscene is done playing. This prefab is also used in the rest of the levels that require this change.

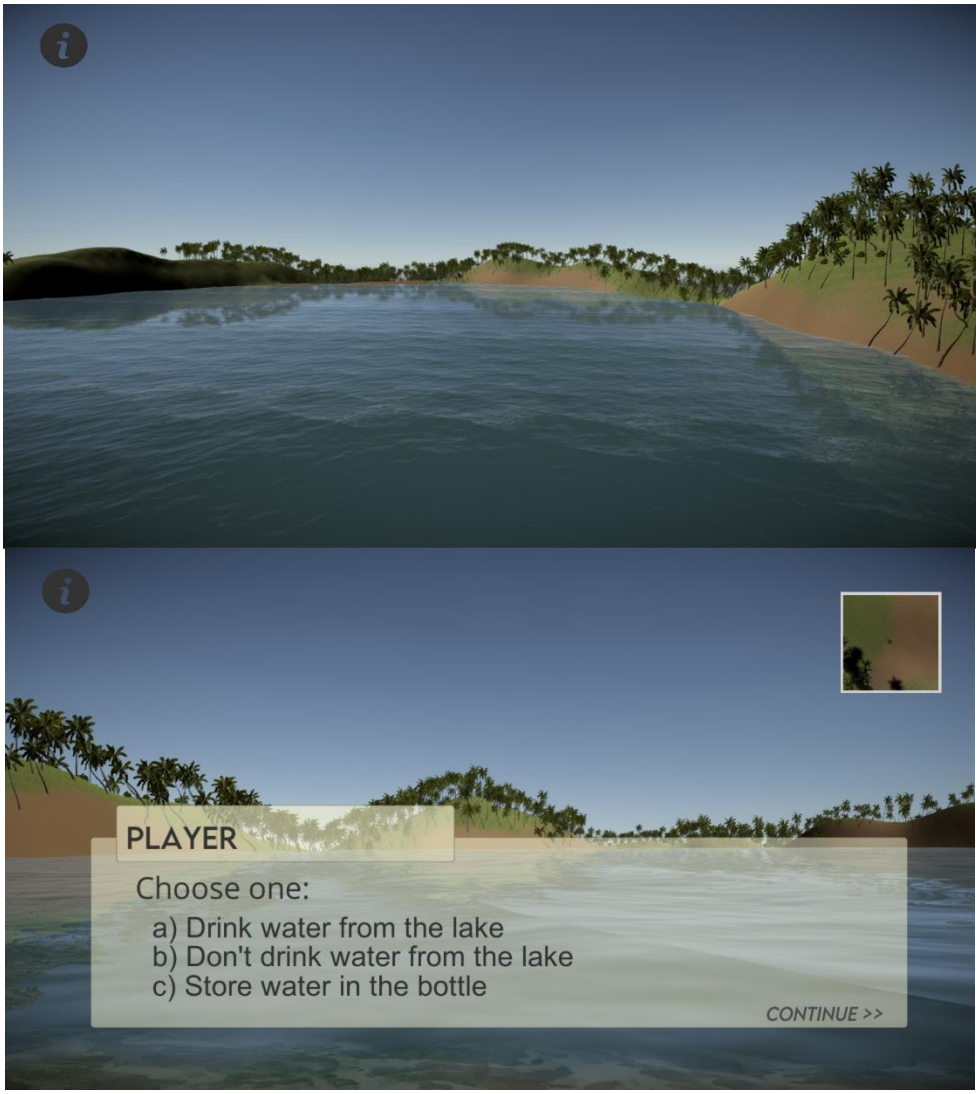


Figure 54. In-game screenshots of the Lake event cutscene and decision

The other event that takes place in this level is the one regarding the Fruit event which does not have a cutscene but has the same mechanics as all of the choice-based events in the game.

Also, taken in account the dimensions of the level, while designing the level and with the use of the Terrain tool we have made clear that there are two ways that the player should go through. This has also been enforced through the dialogues as well as including many triggers that prompt a dialogue if the player tries to go through the forest.

Cutscenes using Timeline and Cinemachine are both played for the appearance of the door as well as when the player crosses it.

The next scene we will be taken a look at is the corresponding to the third proper level: “Level 2: Pirate room”. While designing this level, I wanted it to look as a pirate room, so it is filled with many props, such as wooden boxes, crates, tables, bottles, and such. In this scene, the player is also expected to find and interact with two events. These two events correspond to the Box and Pearl events. These events follow the same mechanic as the ones explained before, the player enters the collider, is prompted to a dialogue with a reflection on the matter and then is left with the decision to make. The game will record the player’s actions by updating the player’s variable as well as playing the appropriate animation and afterward dialogue.

Below is a screenshot of the scene view of the scene as well as an in-game screenshot of the level:

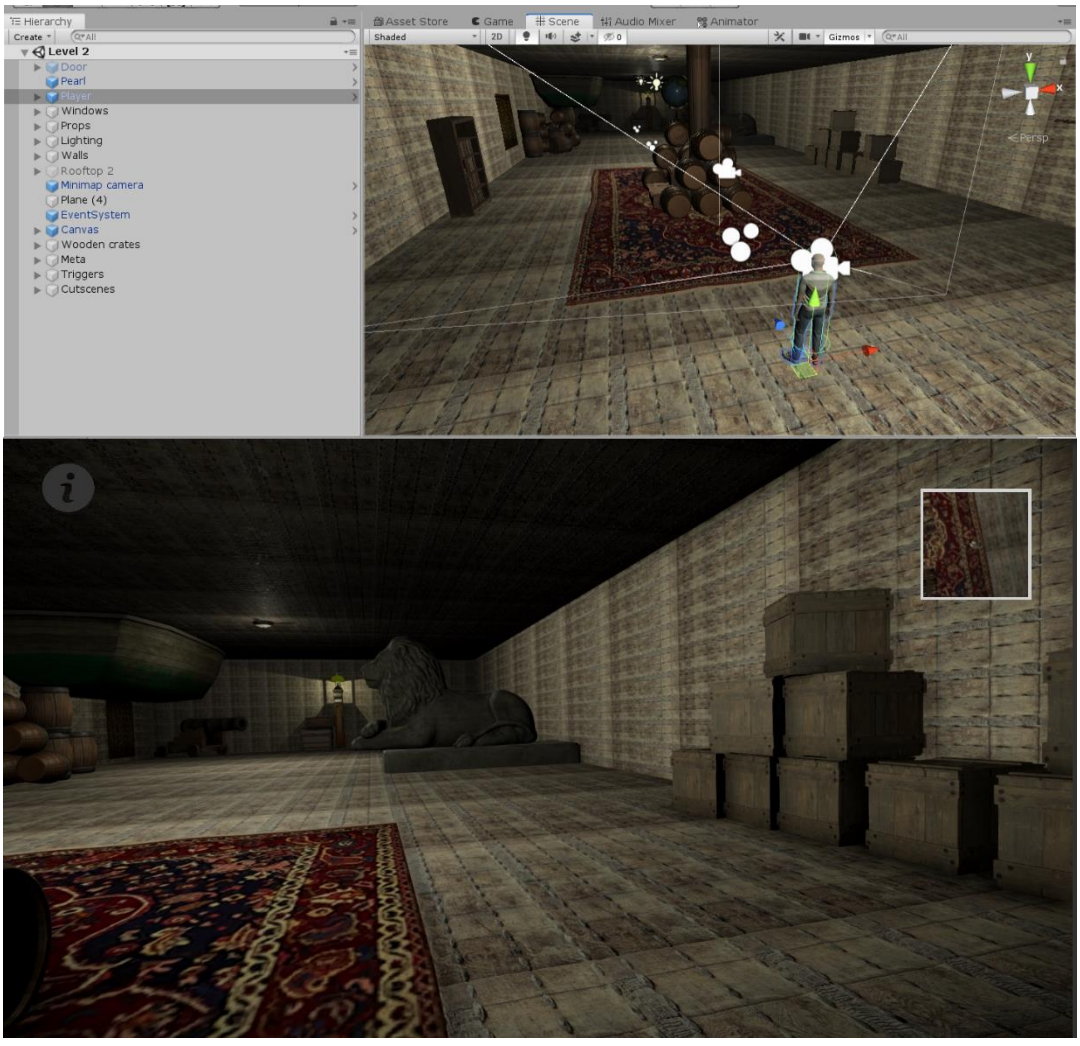


Figure 55. Editor window screenshot and In-game screenshots of Level 2

One special feature about this level is that in the Box event, when the player decides to break the boxes and triggers it, the boxes are indeed being broken in real-time while the animation of the player kicking the boxes is playing. This is done by attaching a script to each of the wooden crates that upon the trigger activation are replaced by their broken prefab counterparts with a mesh collider attached to them, so that it uses Unity's physics to actually break. This gives the player a sense of realism that is really enriching gameplay-wise.



Figure 56. In-game screenshots of the player's decision of breaking the pile and finding the gun afterwards

As for the pearl event, as it has been mentioned before, once the player activates the trigger, if he had eaten the fruit in the previous level he will die because of the fruit's poison and will be taken to the game over screen.

And just like the previous scene, once both events are played by the player, the door leading to the next level will show up and a cutscene that warns the player about it will be played. Another cutscene will be played as well once the player crosses the door.



Figure 57. In-game screenshot of the cutscene that plays once the door appears

The following scene correspond to the fourth proper level of the game: “Level 3: Pure darkness”. This level is composed a single decision-making event, but it also presents another challenge to the player, as there is a hole in the floor that can lead to the player dying and losing the game.

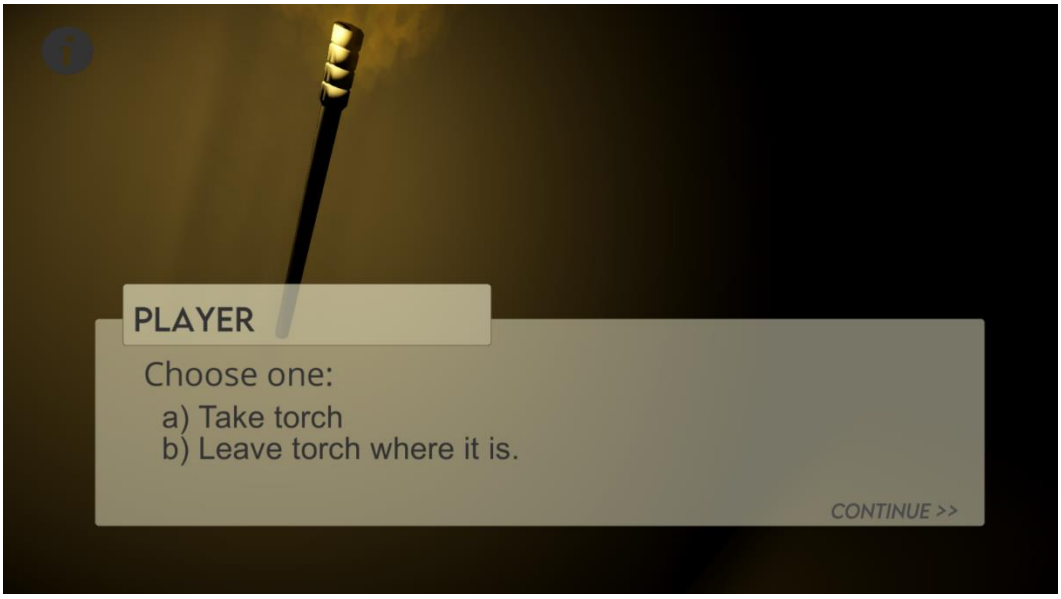


Figure 58. In-game screenshot of the torch event decision

A thing to note about this level is that the torch is made up of a prefab for the wooden stick and a particle system that acts as the fire and has been made by myself. This is the same prefab object that is attached to the main character’s prefab.

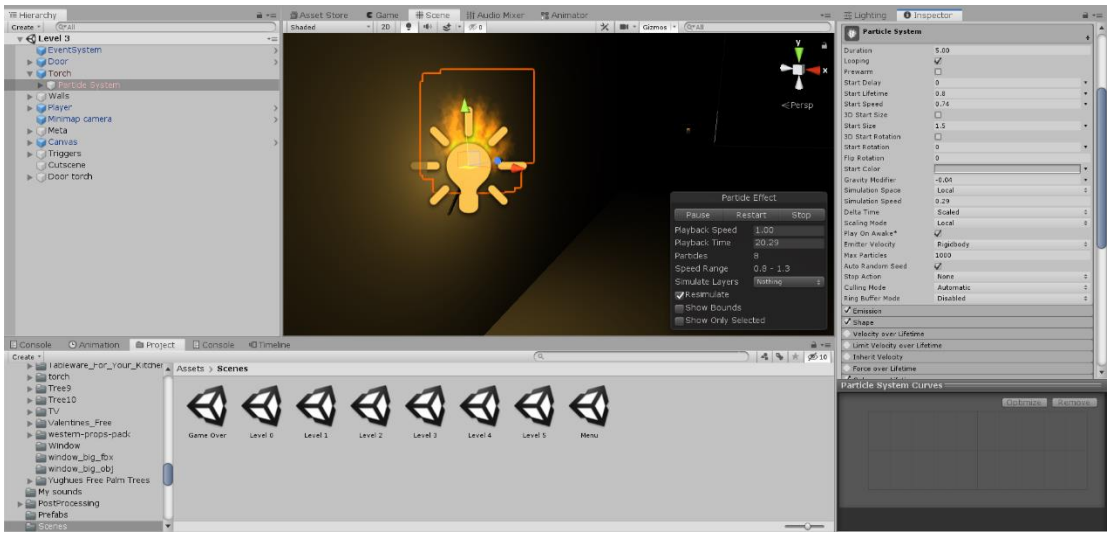


Figure 59. Editor window view of the particle system for the torch fire



Figure 60. In-game screenshot of the player using the torch

Just like the previous levels, if the player makes it to the door at the end of the corridor, he will be prompted to open the door and will be take to the next level.

The last proper level of the game is the one implemented by the Level 4 scene which corresponds to “Level 4: White room” of *Slumber*. This is the level where all of the previous decisions made by the player will be taken in account and where the player will find out the key clues about the story. The player is first introduced to three of the characters, and is expected to get close to the man standing there by the use of dialogues.

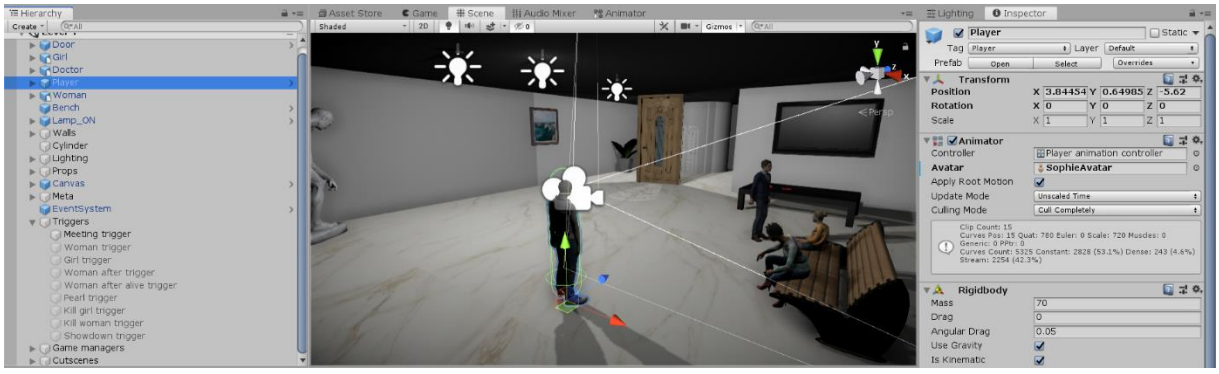


Figure 61. Editor window view of Level 4

While designing this level I wanted it to have a strict and straight-ahead vibe to it so I tried to include objects with sharper angles to them. Also, the shadows in the level have been creating positioning the lights in a certain way so that it creates a double shadow effect. This effect is a subtle clue that means that the characters here are not to be trusted, and thus, that is the reason of the double shadow.

As we can see in the last picture above, this last level has more triggers because the player has to interact multiple times with the doctor, the woman and the girl. Each interaction is triggered by a GameObject with a trigger on it that implements the Event trigger base class with several variations according to the particular event.



Figure 62. In-game screenshot of the player looking at the characters

Another thing that is worth explaining, and specially regarding the name of the other characters, is the use of the Dialogue manager and the dialogues triggered by the events. The dialogue system for the game uses public variables for both the name and the sentences displayed for the given dialogue.

As we have explained before, each dialogue has a number of sentences and a name of the character who talks, in the way that if there's a conversation between two characters, every time the conversation switches to the other player it means a new dialogue. The decision to have the name and the sentences as public is that is easier and more comfortable to have the dialogue's text on the Unity editor while implementing the game, instead of having Unity read it from an external file or via script. If the public name text component does not have anything written on it, then the player's name will be shown, otherwise the name written on the public name text will be the one that shows.

These are some snippets of the Woman trigger GameObject's inspector view, which has a WomanTrigger script attached to it. This WomanTrigger script corresponds to a variation of the Event trigger base class presented on the class diagram for our system. This GameObject has all of the necessary dialogues for all of the options that the player is able to choose, as well as references to some other necessary objects for the animations and management of the game flow.

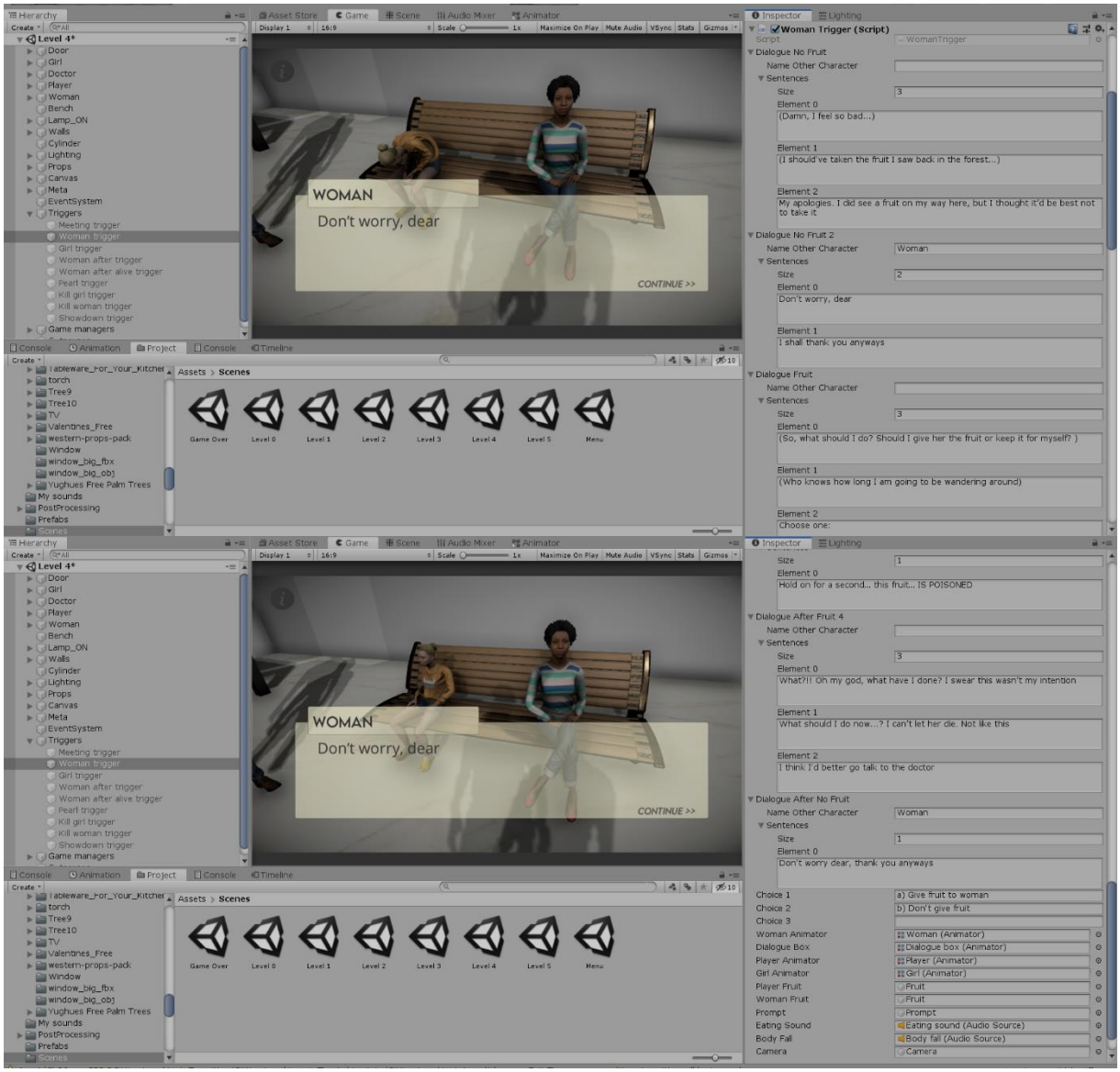


Figure 63. Editor inspector view of the Woman trigger GameObject

Here is also a snippet of the DialogueManager script, in order to understand what is being called by the dialogue triggers:

```
public void StartDialogueDifferentName (Dialogue dialogue, string characterName){
    animator.SetBool("isOpen", true);
    choices = false;

    //nameText.text = dialogue.name;
    nameText.text = characterName;

    sentences.Clear();

    foreach (string sentence in dialogue.sentences){
        sentences.Enqueue(sentence);
        numSentences++;
    }

    DisplayNextSentence();
}

public void StartDialogue (Dialogue dialogue){
    Debug.Log("name: "+dialogue.nameOtherCharacter);
    animator.SetBool("isOpen", true);
    choices = false;

    if (dialogue.nameOtherCharacter != "Doctor" && dialogue.nameOtherCharacter != "Man in suit" && dialogue.nameOtherCharacter != "Woman" && dialogue.nameOtherCharacte
        nameText.text = PlayerValues.name;
    } else {
        nameText.text = dialogue.nameOtherCharacter;
    }

    sentences.Clear();

    foreach (string sentence in dialogue.sentences){
        sentences.Enqueue(sentence);
        numSentences++;
    }

    DisplayNextSentence();
```

Figure 64. Snippet of the DialogueManager script

Ultimately, in order to win, the player has to save both the woman and the girl, so that he is presented the options on who to sacrifice. As only three people are supposed to be able to go through the door, at this point and if the player has both the pearl and the electric gun, the player is able to choose to sacrifice the doctor, which is the key to survive.

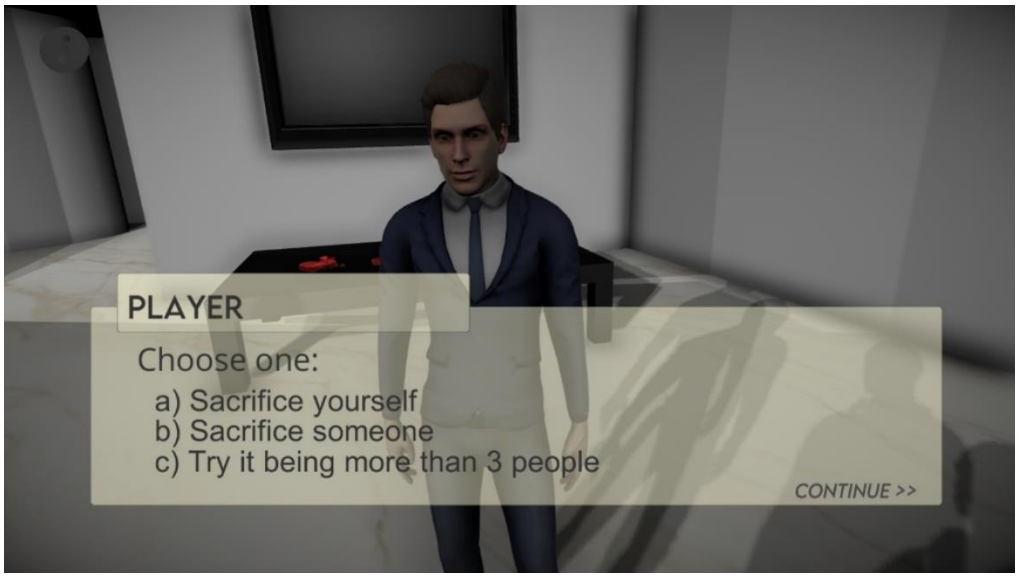


Figure 65. In-game screenshot of the decision on who to sacrifice

In order for this option to trigger the player has to have the pearl on him, does not have to give the poisoned fruit to the woman, and has to give the water bottle to the girl. These two conditions must be met so that the doctor asks the player who to sacrifice.

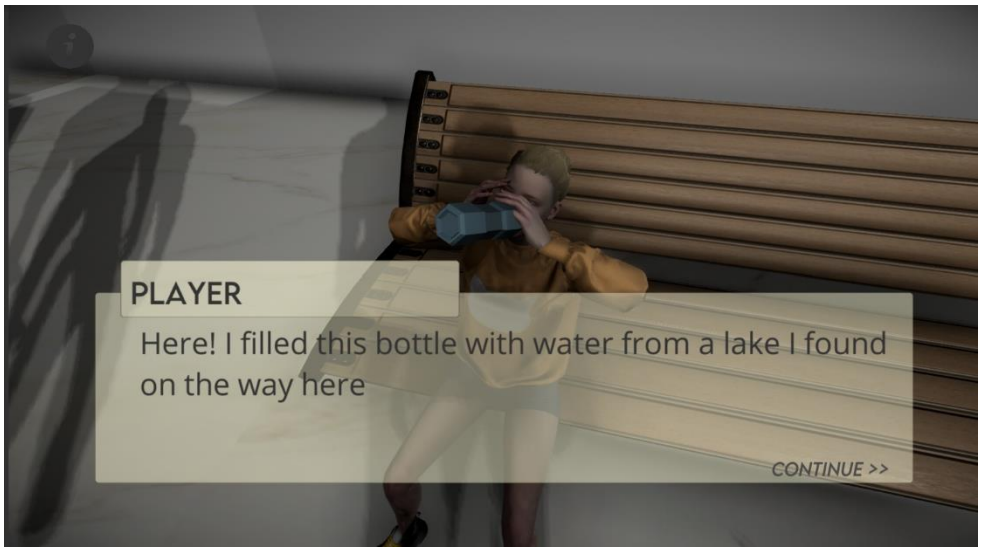


Figure 66. In-game screenshot of the girl drinking water from the water bottle the player gave her



Figure 67. In-game screenshot of the woman dying because the player gave her the fruit

However, if the player was not able to save both the woman and the girl, he will be prompted to go to the door, but will be killed by the doctor from his back. If the player has the option to choose on who to sacrifice but decides to sacrifice himself, he will be killed where he is standing by the doctor. If the player chooses to sacrifice either the girl or the woman, he will have to go to them and kill them with the electric gun. If the player does not have the electric gun on him, he will be simply be killed by the doctor as well.



Figure 68. In-game screenshot right before the doctor kills the player from behind



Figure 69. In-game screenshot of the doctor killing the player where they having a conversation

In the case the player has the pearl, the electric gun and has managed to save both the woman and the girl, and chooses to sacrifice the doctor, the player will be prompted to kill the doctor. Once he does so, he will be cued to go the door and cross it through a dialogue.

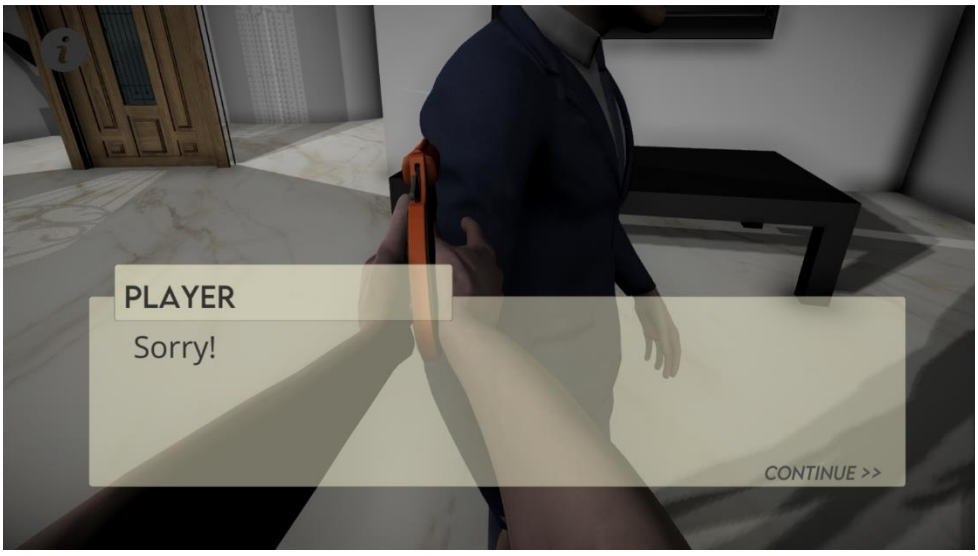


Figure 70. In-game screenshot of the player killing the doctor



Figure 71. In-game screenshots of the player being prompted to cross the door and the transition to Level 5

Once the player crosses the door, and a cutscene plays, he will be taken to the last scene, which is “Level 5: Win – New Slumber”.

This last scene has the same look of Level 0 where the player was first taken at the beginning of the adventure, but this time instead of having the dialogue of the player thinking to himself to find a way to get out of here, a cutscene plays and the player is taken back to the main screen. The background music for this level is also different from the one used for Level 0. While the other level had a more energetic and adventurous feel to it, this time the music has a feeling of emptiness and dissonance. This is the end of the adventure for now, the enclosure to this cycling shape of the game, as a metaphor of life itself. As the ending is not closed, it leaves the door open for a sequel.

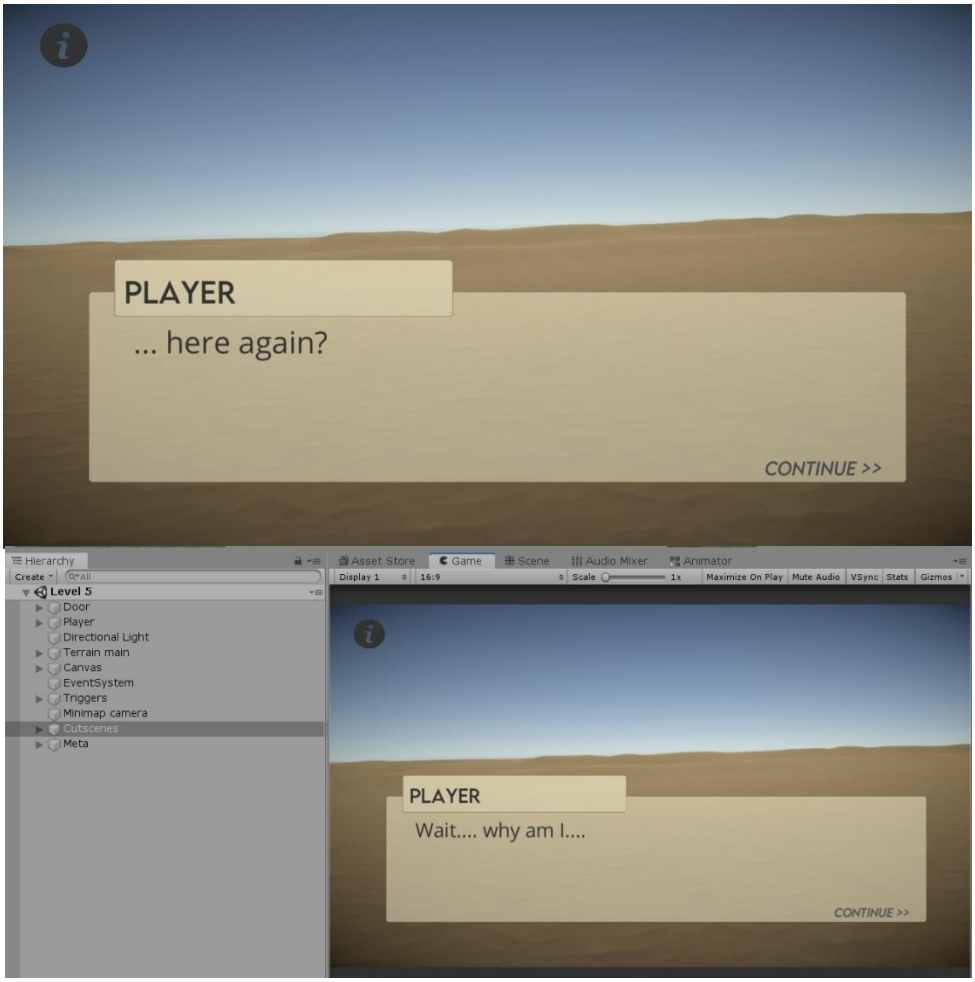


Figure 72. Screenshot of the editor window and an in-game screenshot with the last dialogue

At last, when the development of the game was finished, the game was build to create the executable files and the other necessary files for the game. Once built, the game has a size on disk of about 1 GB. Regarding the build settings for the game performance-wise, the audio and visual quality was set to the highest, as throughout the development this was always kept in mind. The build for the game also has set an executable icon that shows up when it is launched.

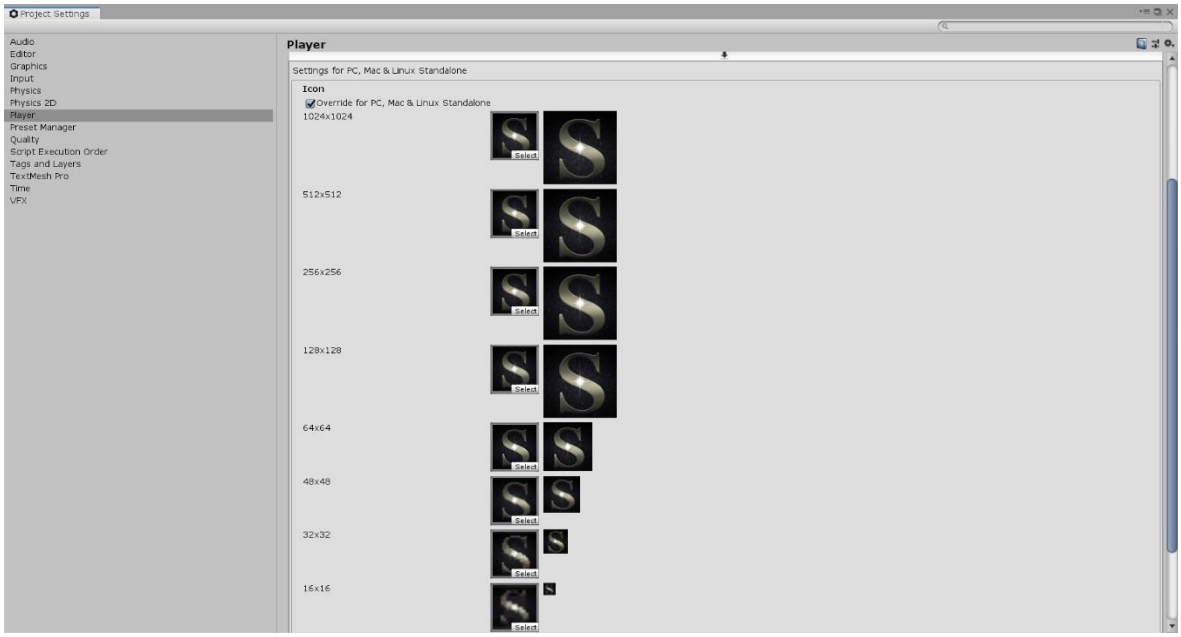


Figure 73. Icon for Slumber

7. Testing and results

7.1. Testing

When it comes to the testing for *Slumber*, there are two different aspects about it. On one hand, we have the testing I have made myself while developing the game, and the other hand we have the testing made by the players who were given the executable files via the Internet to play the video game. This way we know if the video game can be played without any issue in other computers.

Regarding the testing I have made myself, I have mainly worked on unit tests. These unit tests have been put in practice in several levels, regarding each of the mechanics involving the player's control, such as the player movement and the interaction with the game's UI, and each one of the events and the outcomes regarding the player's decisions. For these unit tests I have mainly used the Unity console to make sure the variable values were correct and the required GameObjects were active. These unit tests were carried out during the development of each level, as well as once the development for the video game was concluded. Along with these unit tests once the development for the video game was finished, a system testing was carried out as well. This unit test has been repeated as many times as required to make sure the system gives the player the expected outcome upon his actions.

When it comes to the testing made by other people other than me, the proper way to do so would be to have a team of Quality Assurance trained people to carry out the proper tests, but as this is not possible in our case, the testing has been made by the players themselves. This testing can be looked at from two different perspectives, one is to make sure that the players understand and respond to the game as expected, and the other perspective would be analyzing their decisions within the game itself.

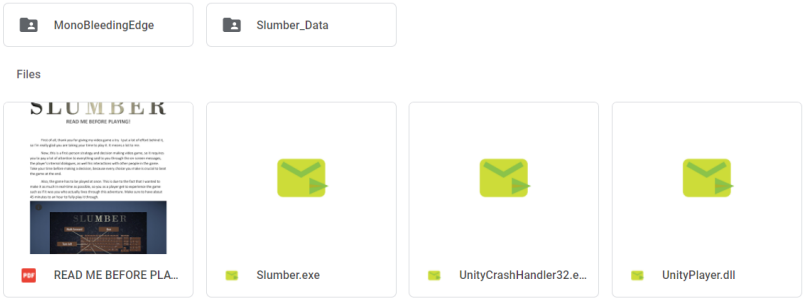


Figure 74. Folder with all the files distributed to the player

Once the game was finished and ready to be tried by other players, the executable file for the game was given to many people to try the game. Along with the executable file and the necessary files to play the game, a "Read me before playing" document was given as well. This document gave the player an introduction to the game, as well as a brief understanding about the game mechanics. The document is the following:

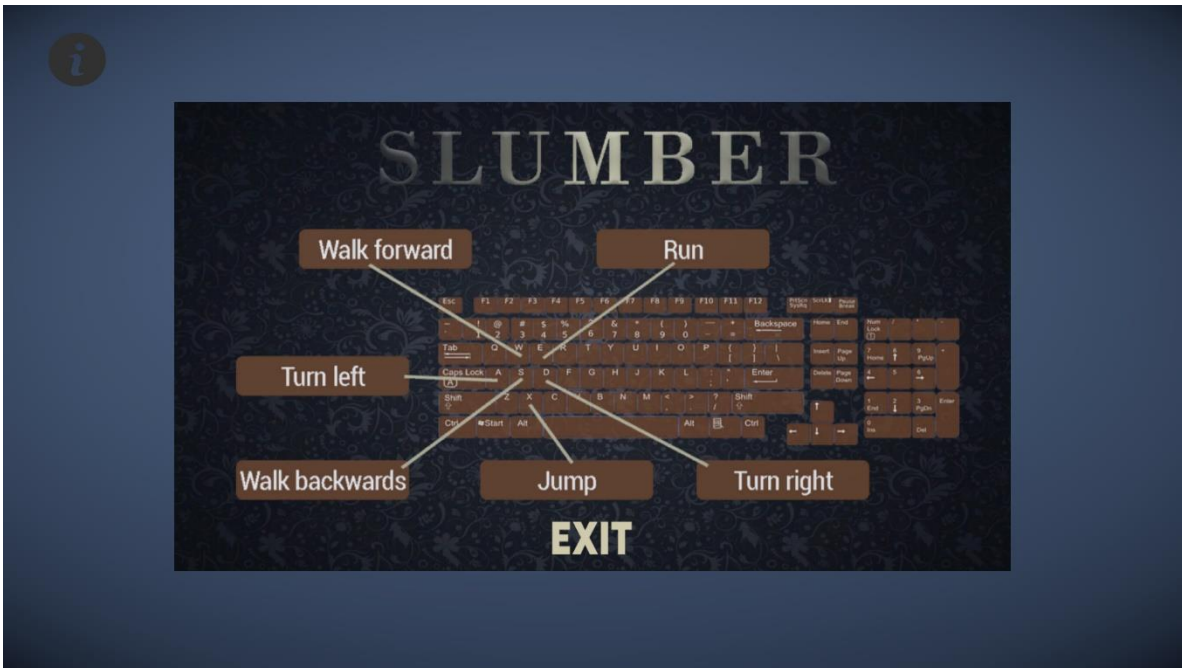
SLUMBER

READ ME BEFORE PLAYING!

First of all, thank you for giving my video game a try. I put a lot of effort behind it, so I'm really glad you are taking your time to play it. It means a lot to me.

Now, this is a first-person strategy and decision making video game, so it requires you to pay a lot of attention to everything said to you through the on-screen messages, the player's internal dialogues, as well his interactions with other people in the game. Take your time before making a decision, because every choice you make is crucial to beat the game at the end.

Also, the game has to be played at once. This is due to the fact that I wanted to make it as much in real-time as possible, so you as a player get to experience the game such as if it was you who actually lives through this adventure. Make sure to have about 45 minutes to an hour to fully play it through.



However, you can quit the game any time you want by clicking the (i) icon on the top left corner, and then the EXIT button. However, you will have to start the game again if you do that. So be careful!

Lastly, the game is pretty hard to beat on your first try, so you may want to play it for a second time if you weren't able to beat it at first. Don't let it discourage you, as it is much easier to beat on your second play because you have a lot more of information (if you have been paying attention, that is)

That's it! I hope you enjoy it, and make sure to let me know once you're done playing with the game so I can give you a small survey with a few questions about your experience with the game which will help me immensely.

Thank you once again!

- Yasmine

The players were also told to give me a reply once the were done playing the game, and a survey was given to them. This survey asked them brief questions about their name, gender and age, their overall enjoyment of the game and the decisions made on their first time playing the game. The survey was made so that it was not too overwhelming to the player but that at the same time allowed me to draw out conclusions upon the player's decisions and their experience with Slumber.

Below is a snippet of the survey:

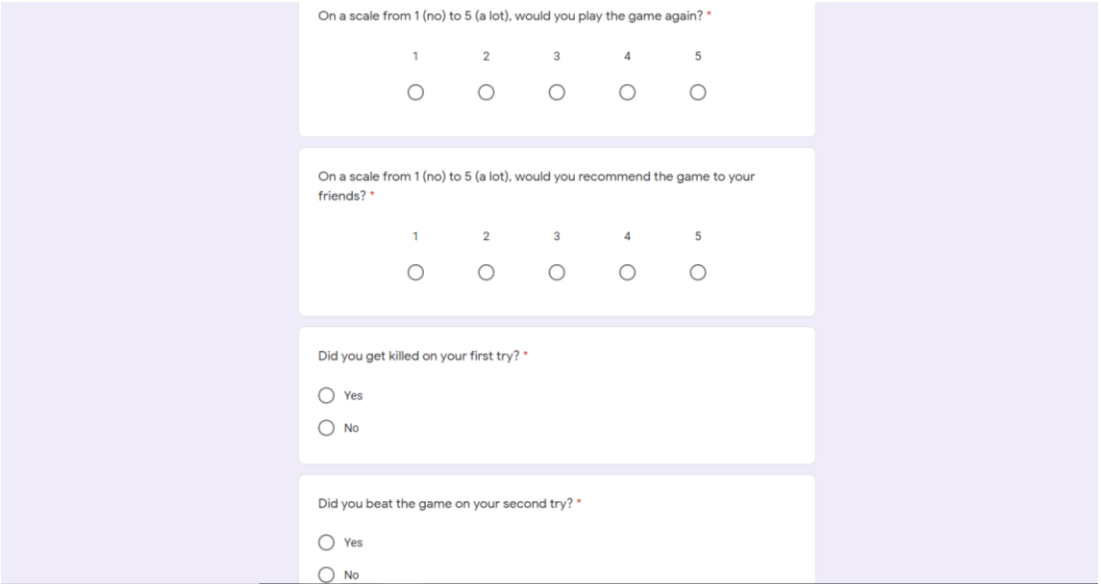


Figure 73. Snippet of the survey given to the players

7.2. Results

In this point we will analyze the results obtained with the players' surveys. As we have mentioned before, we will study them from two different perspectives, regarding the choices they have made within their first play and their experience overall with *Slumber.*

A total of 24 people have tried the game, so we have a proper amount of people who have tested the game in order to estimate the results and be able to make conclusions out of them. The players who have tried the game correspond to an age range that goes from 18 to 52 years old, from 8 different countries and being the genders of them almost completely balanced. The graphics below portray these facts:

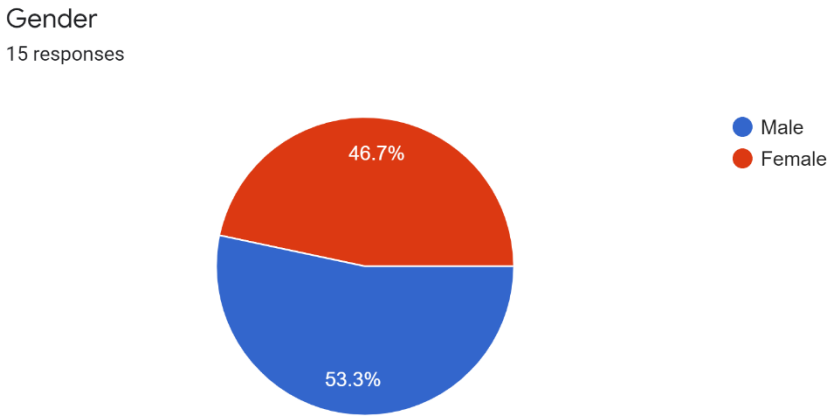


Figure 74. Gender of the players

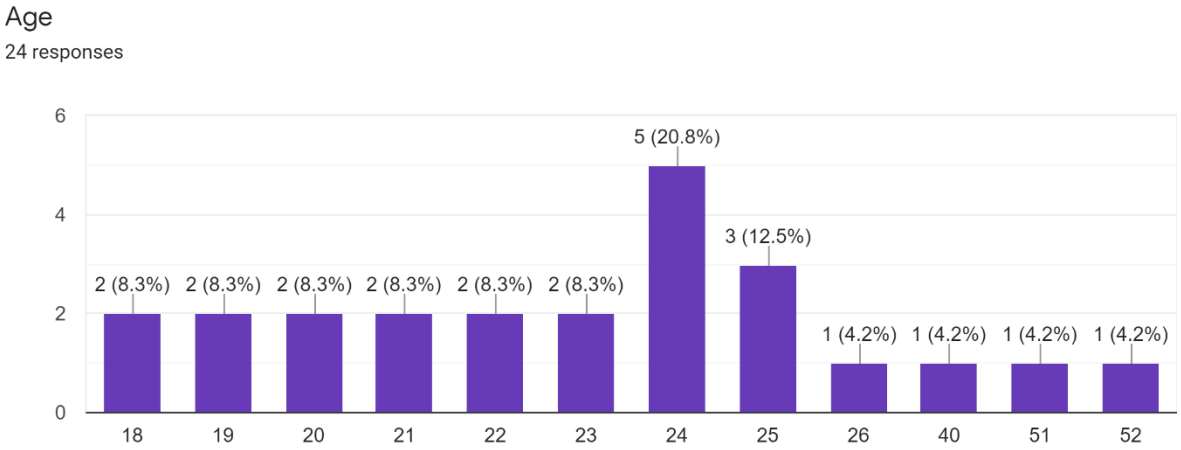


Figure 75. Age of the players

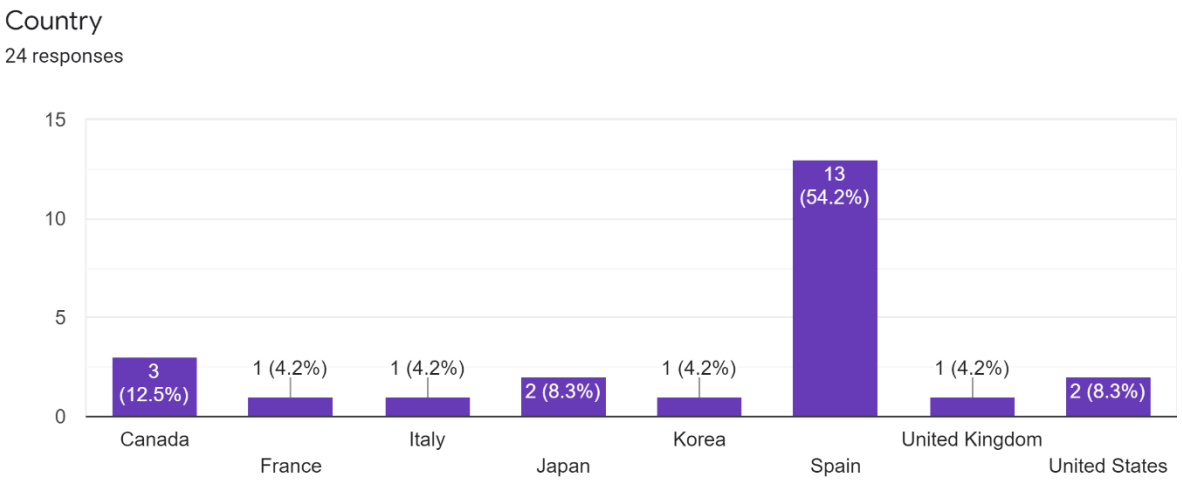


Figure 76. Countries where the players are from

Along with this demographic information, the players were also asked about their computer specs, so that we could estimate that the requirement of the video game of being able to be played by lower-end computers was met. The players who have tried the game have mid-end and high-end computers, but none of them has had any issues while playing the game, so we can estimate that the video game could also be played in lower-end computers without any kind of issue.

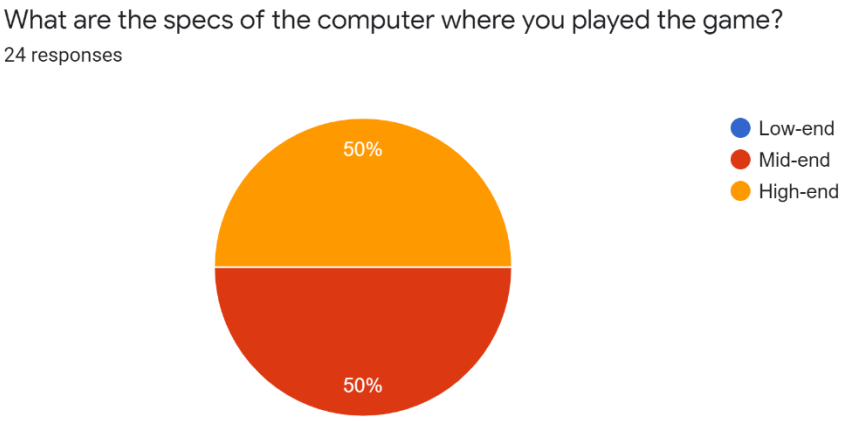


Figure 77. Specs of the computers where the players played the game

When it comes to the players' overall enjoyment of the game, these have been their responses:

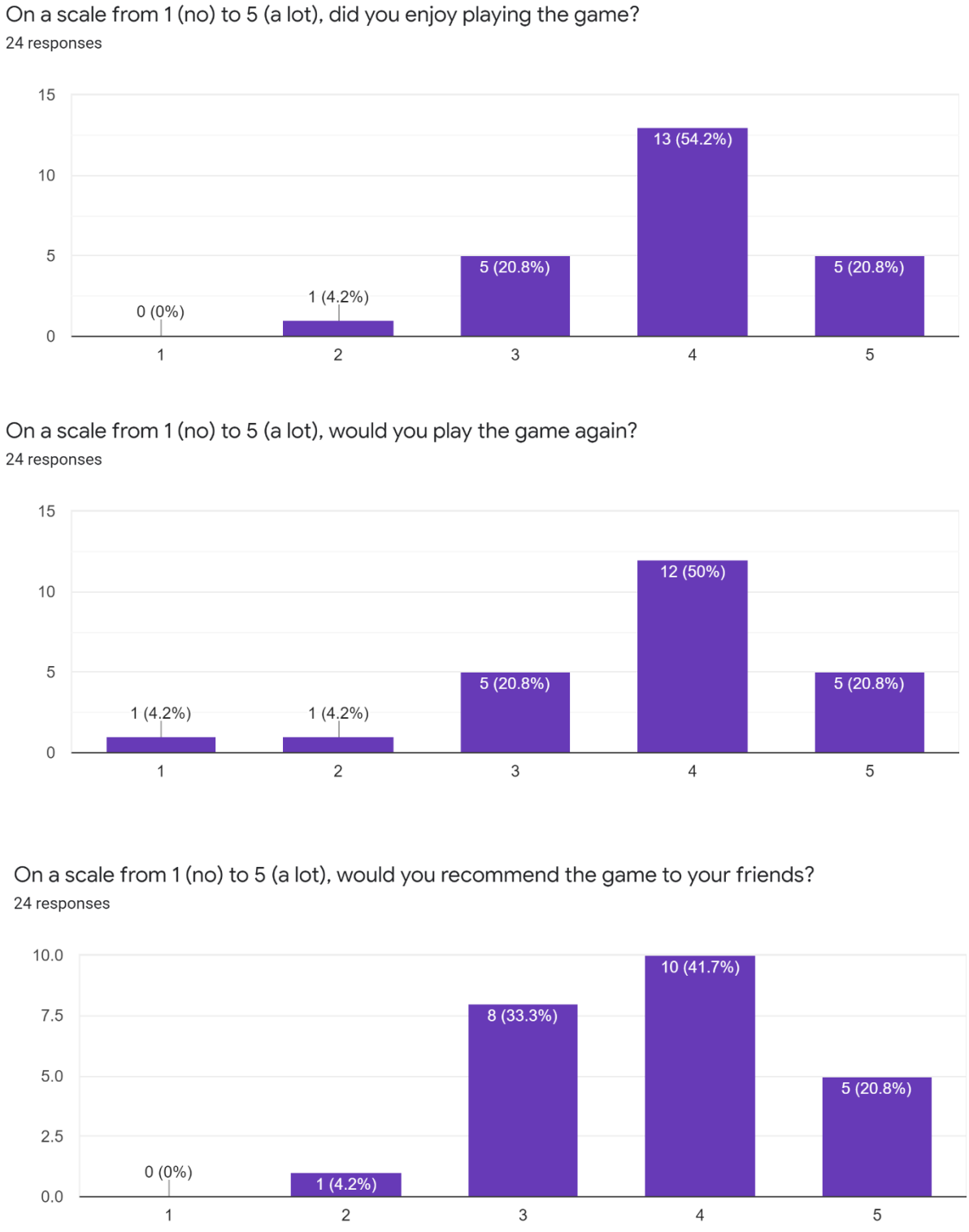


Figure 78. Players' answers about their overall enjoyment with Slumber

So as it is shown in the graphs above, the great majority of the players have had an excellent experience with the game. Most of them have really enjoyed playing the game and would recommend it to a friend. When it comes to playing the game again, as it is a video game that is really story-heavy and story-based, it is understandable that some players would not want to play it again, mainly due to the fact that they have probably figured out already the story and the mystery behind the adventure. However, there are some players that have really liked the story and maybe have not tried all of the possible routes, so they plan to play it again in the future. These are some of the optional comments that the players have left with their survey response:

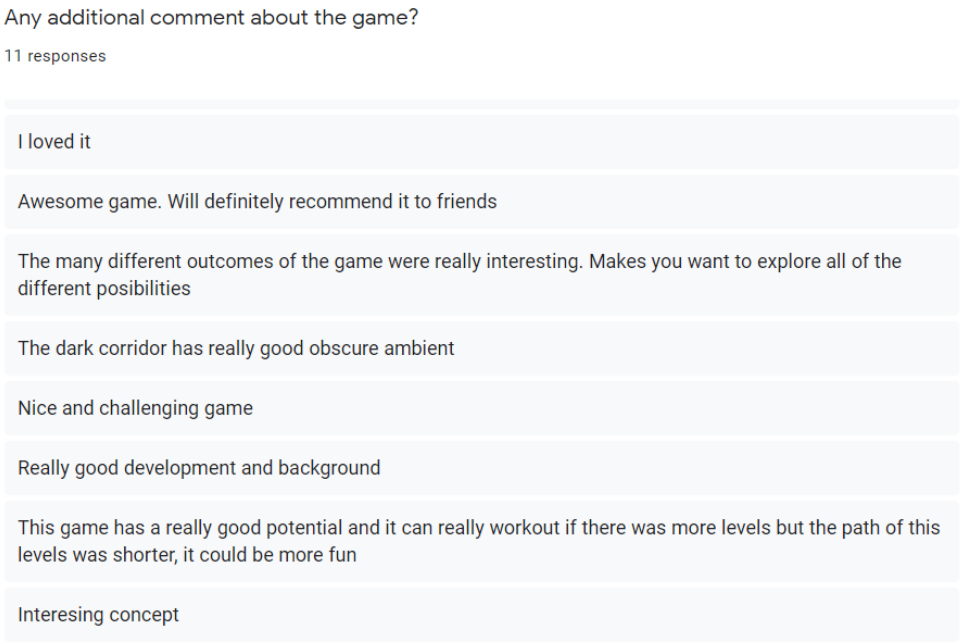


Figure 79. Some optional comments about the game left by the players

Also, when the players were given the files to play the game, they were only required to play the game once before doing the survey, but if they wanted they could play as many times as they wanted before doing the survey. So, related to the amount of times that the players have played the game before doing the survey we have collected the following data:

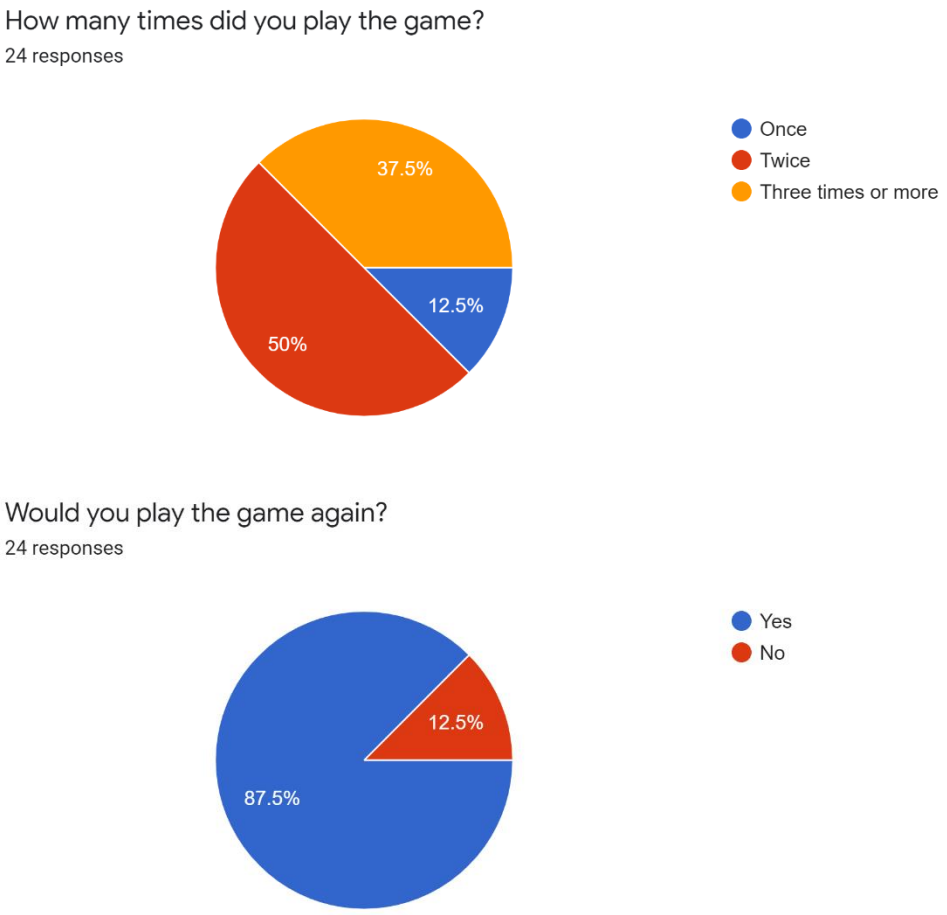


Figure 80. Players' answers concerning the times they have played the game and if they would play it again

So, as we can see in the graphs, the great majority of the players would play the game again, so thanks to this we can estimate that the game appeals to be interesting to the players overall. Also, half of the players who have played the game have played the game twice before doing the survey, while only a really small amount of the players have only played it only once, and the rest has played it three times of more. With this data, we can estimate that as it was planned, the game is really hard to beat at first, but it is easier to beat on the second try.

Also, related to the amount of times that it usually requires to beat the game and whether it is hard to beat on the first time playing, the players have answer the following questions:

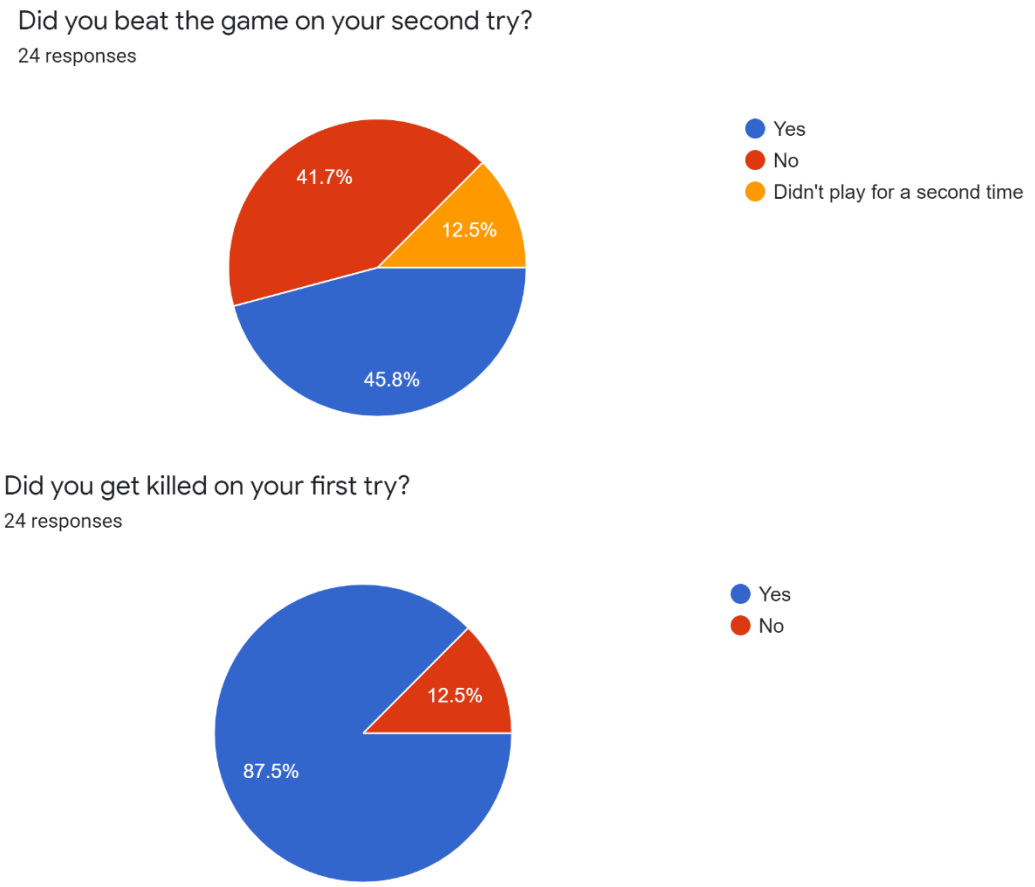


Figure 81. Players' answers about beating the game on their second try and whether if they were able to beat the game at first

The players' answers let us estimate that, just like it was planned, the game is quite hard to beat at first, but a considerable amount of the players were able to beat it the second time, while some others needed more times to do so. Only the 12.5% of players were able to beat the game at first, in this sense, it is also worth noticing that this is the same amount of players that only played the game once so we can guess that the rest of the players who have played the game more than once have tried beating it when they lost at first.

Furthermore, we would go over the players' answers upon their decisions on their first time playing the game.

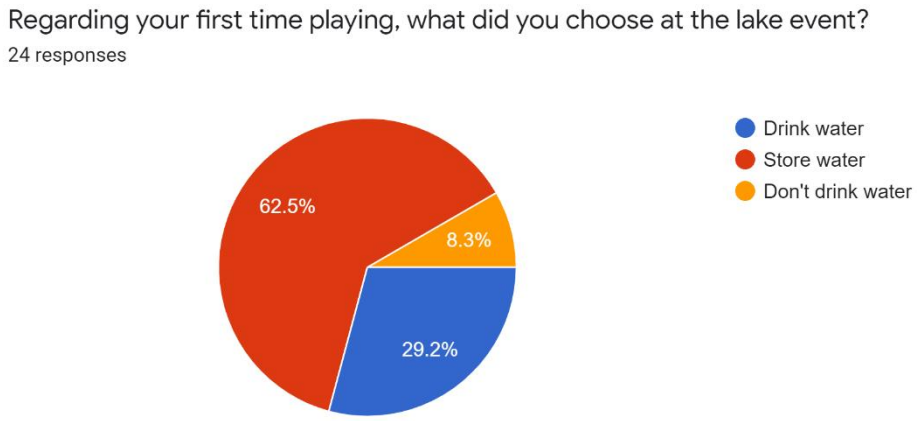


Figure 82. Players' answers about the Lake event

For starters, most of the players chose to store the water in the water bottle at the Lake event. This event was planned to see if the players would be cautious enough to keep the water for the future, be impulsive and simply drink the water or if they would be too unadventurous and would not do anything with the water. As we can see in the graph, the majority of the players were cautious and decided to store the water in the bottle.

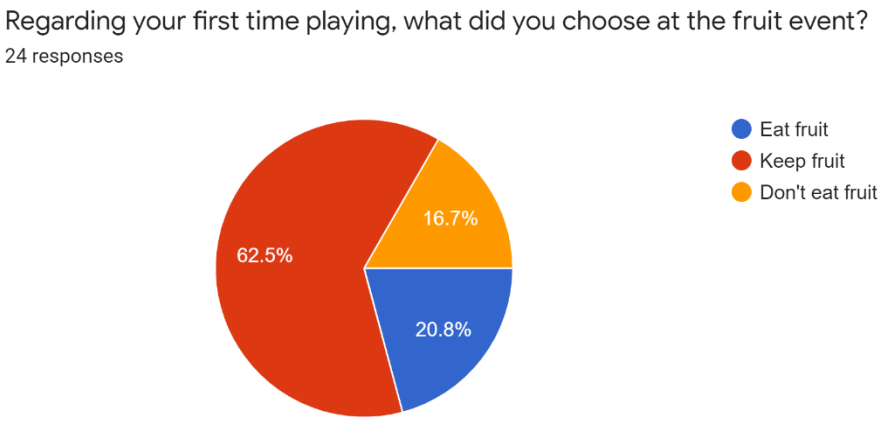


Figure 83. Players' answers about the Fruit event

The Fruit event followed the same premise of the Lake event, and that was to see if the players would choose a possibly risky option or not. The results were pretty similar to those of the Lake event, as most of the players were cautious and decided to keep the fruit. The rest of the players either ate the fruit or did not do anything about it. In this case, the riskiest option was a more popular option than the one regarding the water of the lake.

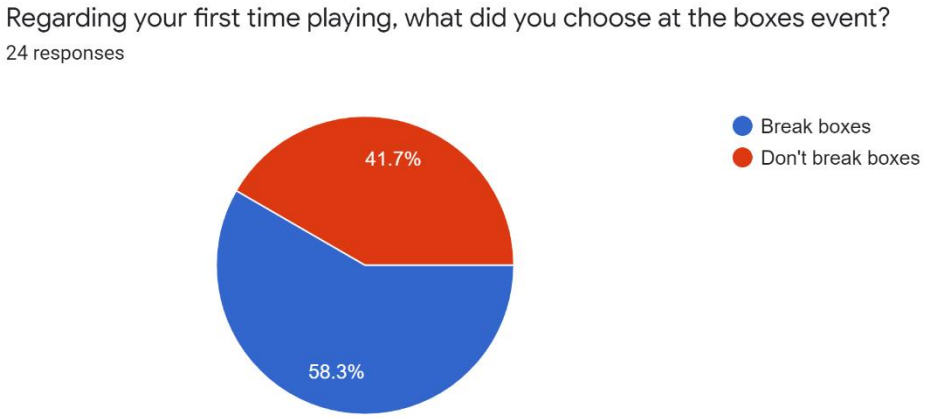


Figure 84. Players' answers about the Boxes event

In the case of the Boxes event, most of the players decided to break the boxes. The premise of this event was also to see if players would take the risk of breaking the boxes being in a place that looks like belongs to someone else. The answer upon this response is quite balanced, but the majority of players decided to break the boxes.

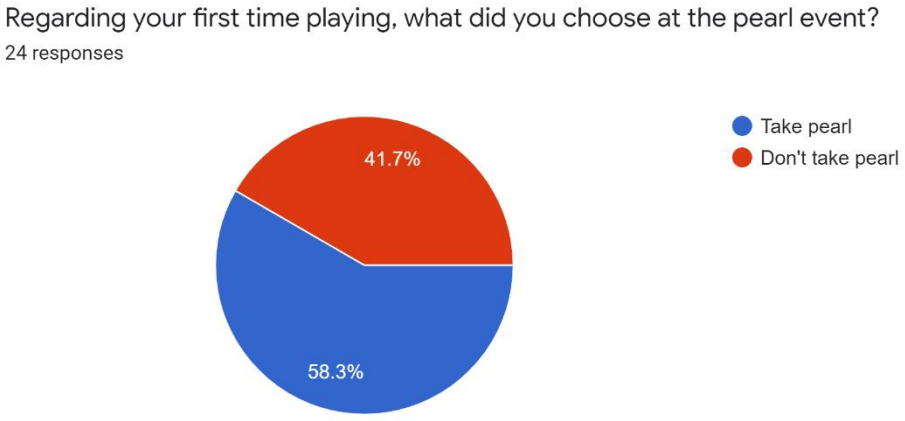


Figure 85. Players' answers about the Pearl event

Regarding the Pearl event, most of the players decided to take the pearl. This event followed the same premise as the previous one, so the majority of the players chose to steal the pearl not knowing if it belonged to someone. Surprisingly, both the steal pearl and the break boxes options have the same amount of people who chose it.

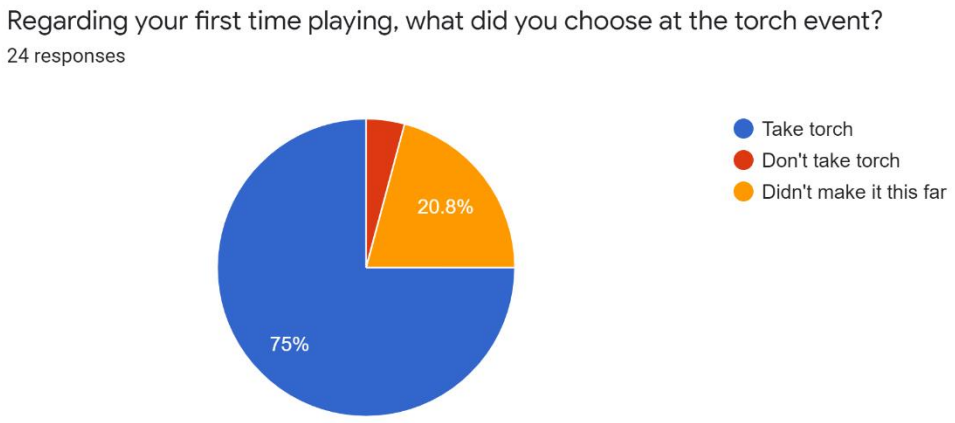


Figure 86. Players' answers about the Torch event

The Torch event's premise is related to being selfish and taking the torch of leaving it where it is and risking your own life in case someone comes after you. In this case, 75% of the players were selfish enough to take the torch, a vague amount of people chose not to take it and the rest did not make it this far. The people who did not make it this far responds to the amount of people who chose to eat the fruit at the Fruit event, and as we can see, they both have the same percentages.

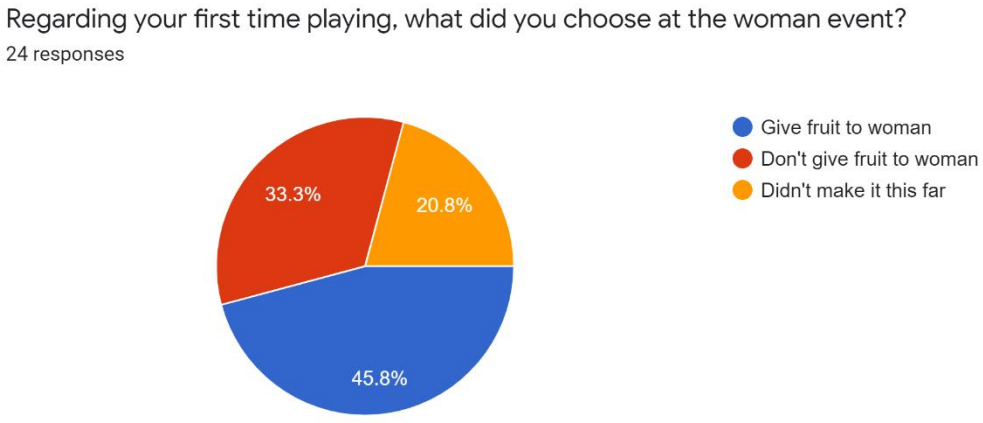


Figure 87. Players' answers about the Woman event

The first event of the last level, the Woman event, had the majority of players giving the fruit to the woman. In this event, I wanted to measure if people would be generous enough to give the fruit to the woman, or if they would prefer to be selfish and keep it for themselves. Also, the amount of people who did not make it this far, is the same that ate the fruit back at the forest.

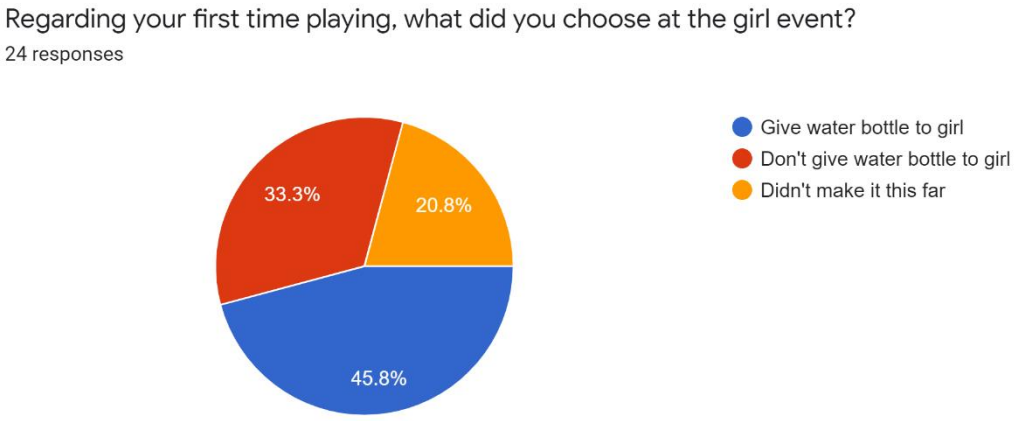


Figure 88. Players' answers about the Girl event

About the second event of the last level, the Girl event, it had the same premise as the Woman event and it was to see whether the players would be generous or selfish. This event had the exact same amount of people who chose to be selfish and generous, being the majority of players generous enough to give the water bottle to the girl.

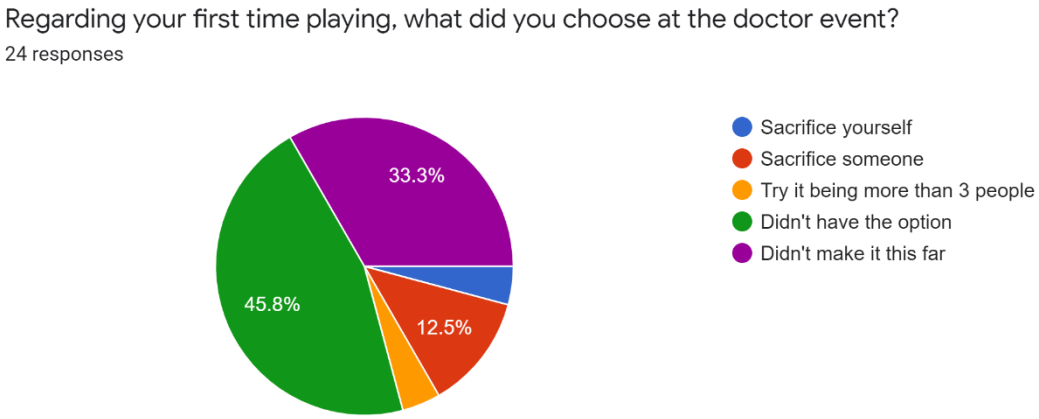


Figure 89. Players' answers about the Doctor event

The last major event of the game, the Doctor event, gave the players the options to either sacrifice themselves, sacrifice someone or try going through the door being more than three people. Most of the people did not have this option, as they were killed by the doctor as they were not able to save both the girl and the woman. Only the 12.5% of players chose the option to sacrifice someone.

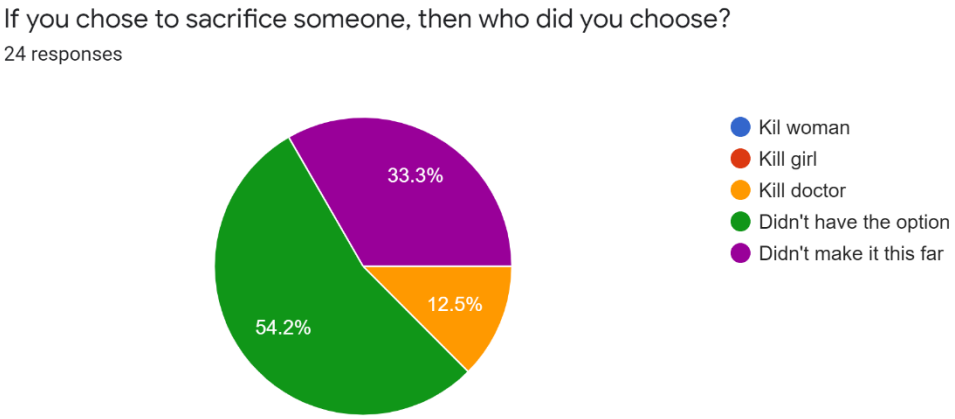


Figure 90. Players' answers about who they chose to sacrifice

As who they chose to sacrifice the ones who chose to sacrifice someone, all of the 12.5% of players who were able to beat the game decided to kill the doctor, and none of them chose to kill either the woman or the girl. This shows the affection that people tend to have with either women that are pregnant like the one in the game and also kids. The rest of the players either did not save both the woman and the girl or they died before that.

8. Outcome and conclusion. Future work

To conclude this document, we will take a look back at both the implementation of the game as a whole, the results drawn out from the players answers to the survey and the how the making the video game will reflect in the future.

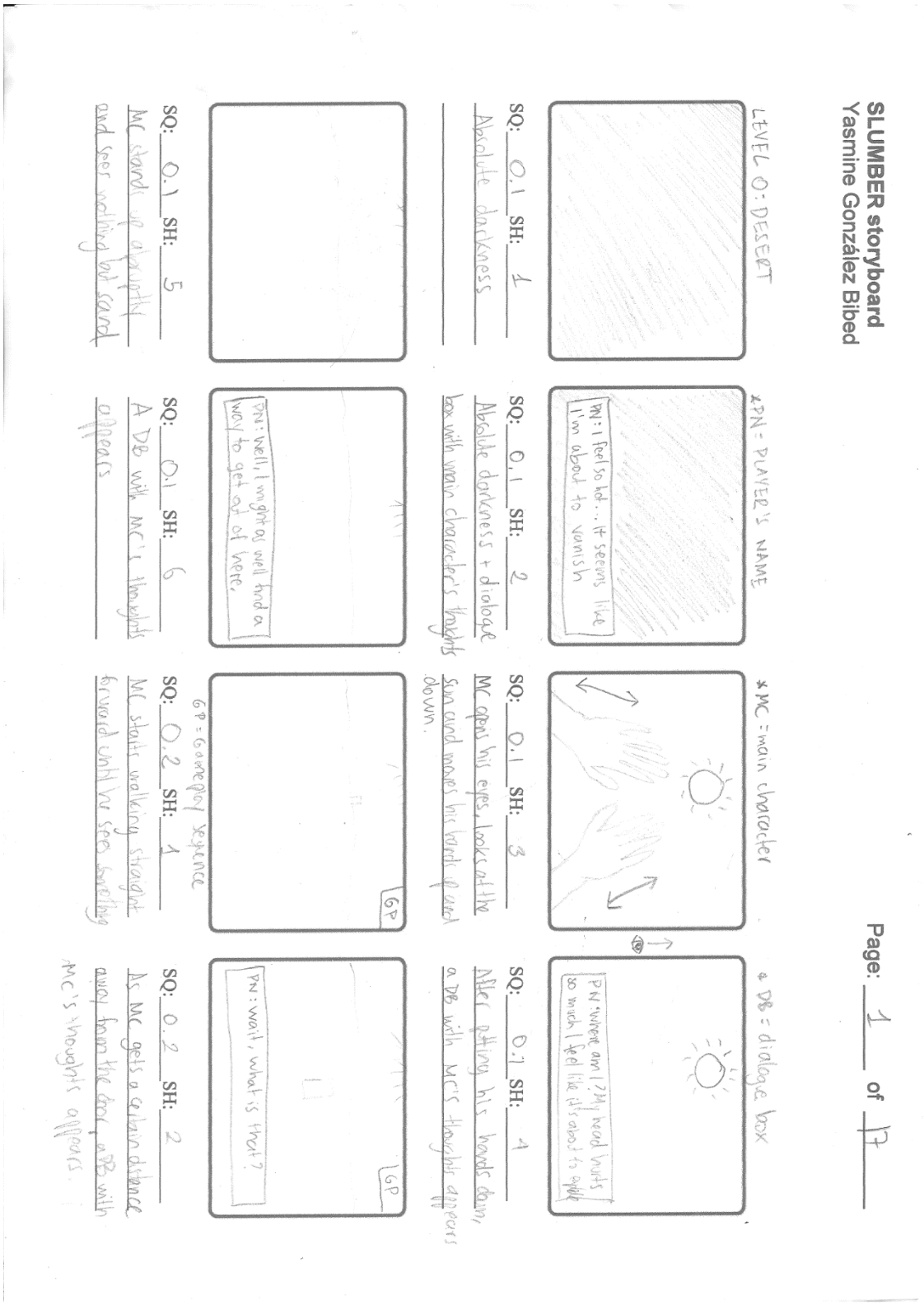
The first thing to emphasize about the development of the game, is that with the development of *Slumber* I have realized and confirm how powerful game engines are and the many tools it provides to game developers in order to implement his ideas. However, in this sense it is also worth mentioning that in my case, there has been some limitations because I had no real budget for the development of the video game. This led me to do some minor adjustments to the script and the storyboard, but nevertheless, I was absolutely able to implement the idea I had at first for *Slumber.* Also, with the development of the video game I have noticed that when writing the script and drawing the storyboard for it, there are many things that are not taking in account and have to be included while implementing it. These are mainly clues about the game controls and several aids to the player.

Furthermore, the players' answers to the survey allowed me to have an estimation of the players' thoughts about the game, their experience with it and to analyze their answers from a more psychological perspective. *Slumber* had a great reception from the players and the experience with the game was really positive. In addition, the players' responses to how the game was proposed, its difficulty and the story was outstanding, as the feedback received shows. Regarding the players' decisions, in general, we can say that most players were cautious, risky upon other people's belongings, and had a generous and kind attitude towards women and girls.

As for the future, having worked in this project for so long I am completely sure that I would be able to work in a project of the same nature without any problem at all, as I have learned many things about video game development that will surely be of use in the future. Regarding the future of the video game itself, it has an open ending which was left like that on purpose, so the video game could perfectly have a sequel to it. This is an open door for a future project.

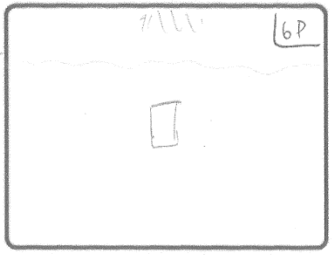
Overall, it was a great and satisfactory experience to be able to develop a video game entirely by myself, using the knowledge I have acquired from the Multimedia Engineering degree and my own skills and ideas. By doing so I was able to fully comprehend the process of making a video game, since the first ideas about it until the last touches of the development as well as analyzing people's reaction and experience with the video game.

9. Annex 1. Video game storyboard

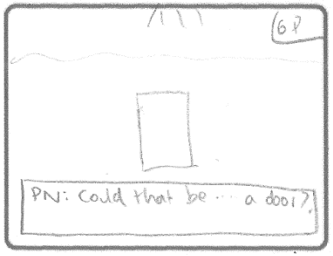


SLUMBER storyboard
Yasmine González Bibed

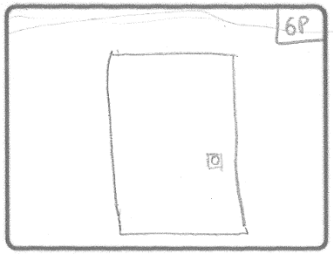
Page: 2 of 17



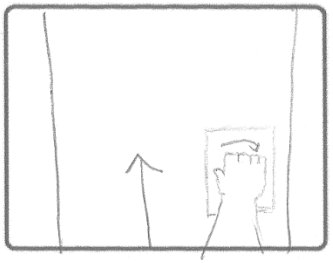
SQ: 0.1 SH: 3
MC keeps on walking



SQ: 0.2 SH: 4
MC gets closer to the door and a DB appears



SQ: 0.2 SH: 5
MC gets closer to the door

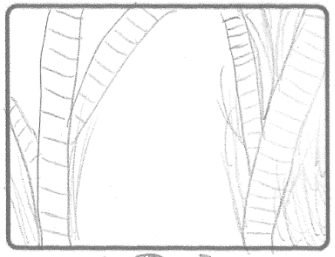


SQ: 0.2 SH: 6
MC opens the door and goes through it

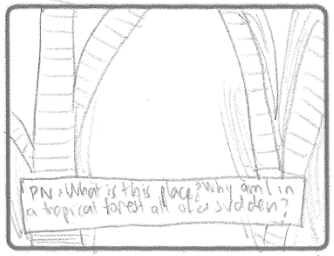
LEVEL 1: TROPICAL FOREST



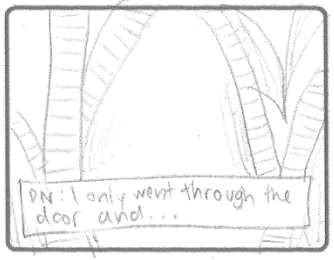
SQ: 1.1 SH: 0
Transition screen



SQ: 1.1 SH: 1
MC finds himself in a tropical forest. He looks to both sides



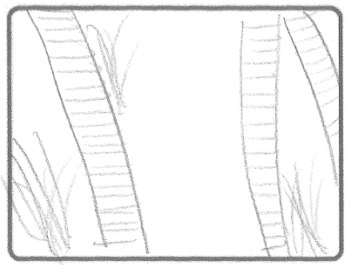
SQ: 1.1 SH: 2
A DB with MC's thoughts appears



SQ: 1.1 SH: 3
The DB is updated with the rest of MC's thoughts

SLUMBER storyboard
Yasmine González Bibed

Page: 3 of 17



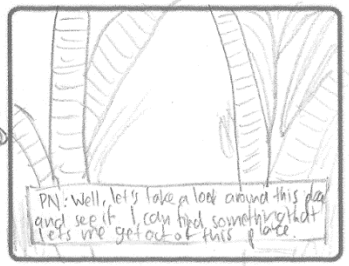
SQ: 1.1 SH: 4
MC looks back at the door, but it's not there



SQ: 1.1 SH: 5
DB showing MC's thoughts appears



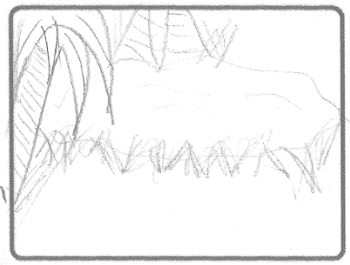
SQ: 1.1 SH: 6
DB updates



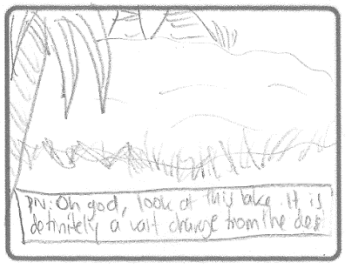
SQ: 1.1 SH: 7
MC turns again. The DB is updated at the same time

1) LAKE EVENT

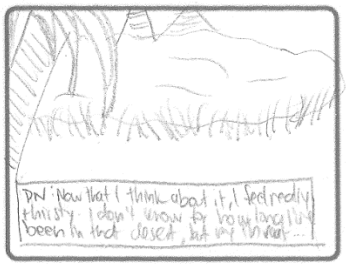
Player is free to explore the land there are 3 events



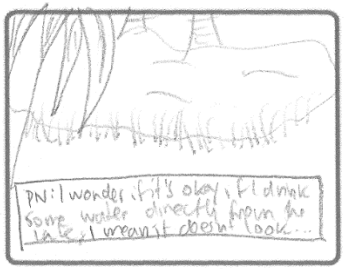
SQ: 1.2 SH: 1
MC finds a lake. He is feeling thirsty



SQ: 1.2 SH: 2
DB with MC's thoughts appears

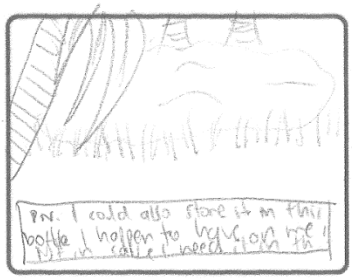


SQ: 1.2 SH: 3
DB updates.

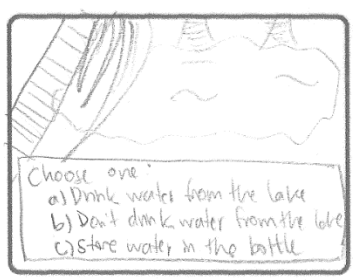


SQ: 1.2 SH: 4
DB updates.

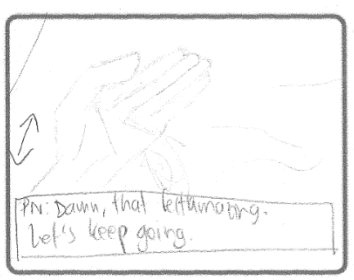
SLUMBER storyboard
Yasmine González Bibed



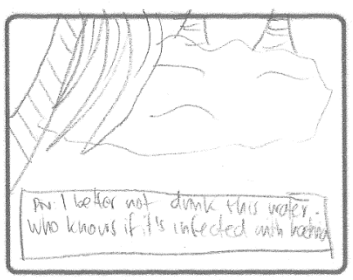
SQ: 1.2 SH: 5
DB updates.



SQ: 1.2 SH: 6
DB updates (bigger one)

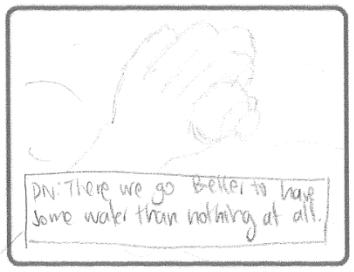


SQ: 1.2 SH: 7.a
MC gets closer to the lake and drinks water

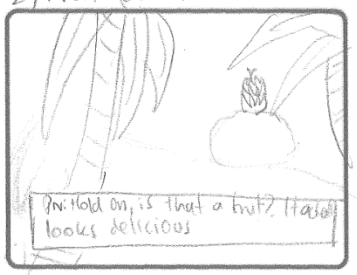


SQ: 1.2 SH: 7.b
MC doesn't drink any water. DB updates

2) Fruit event



SQ: 1.2 SH: 7.c
MC gets closer to the lake and fills up water bottle



SQ: 1.2 SH: 1
MC finds a fruit and he's hungry. A DB appears



SQ: 1.3 SH: 2
DB updates



SQ: 1.3 SH: 3
DB updates

SLUMBER storyboard
Yasmine González Bibed

Page: 5 of 17



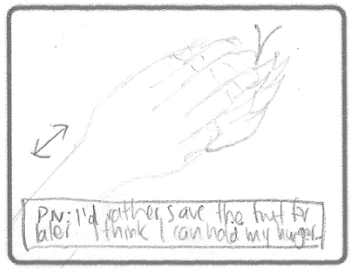
SQ: 1.3 SH: 4.a

MC takes the fruit and eats it
DB showing MC's thoughts appear.



SQ: 1.3 SH: 4.b

MC decides no to eat the fruit. DB showing MC's thoughts.

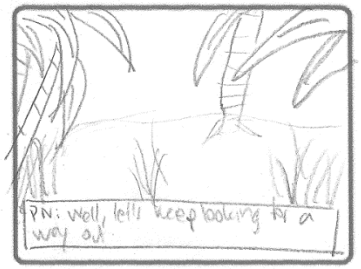


SQ: 1.3 SH: 4.c

MC takes the fruit and puts it in his pocket. DB showing MC's thoughts

LEVEL 2: GLOOMY ROOM

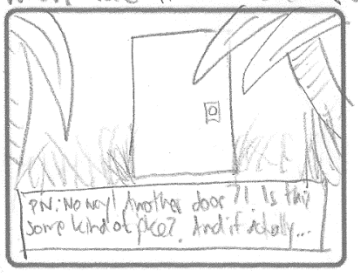
-After the two events have taken place



SQ: 1.4 SH: 1

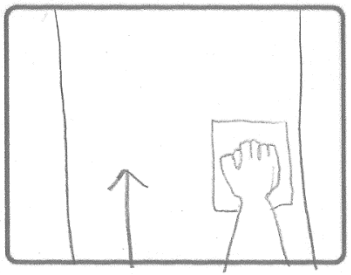
After the two events have taken place
DB with MC's thoughts appear
A door appears.

-When MC finds the door ←GP



SQ: 1.5 SH: 1

When MC gets closer to the newly appeared door, a DB appears.



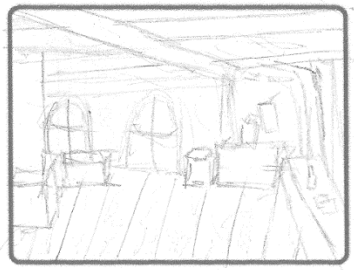
SQ: 1.5 SH: 2

MC goes through the door



SQ: 2.1 SH: 0

Transition screen



SQ: 2.1 SH: 1

MC finds himself in a gloomy and dark wooden room

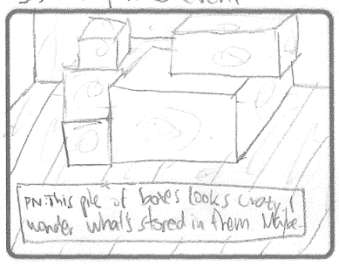
SLUMBER storyboard
Yasmine González Bibed

Page: 6 of 17

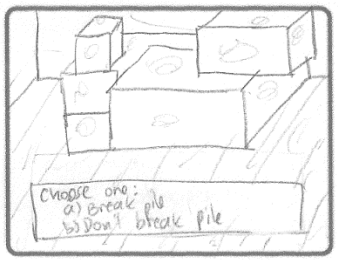


SQ: 2.1 SH: 2
DB with MC's thoughts appears

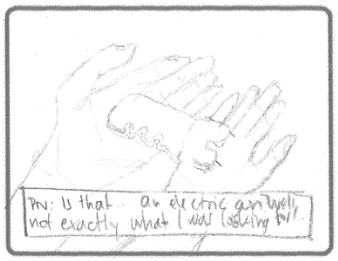
1) Pile of boxes event



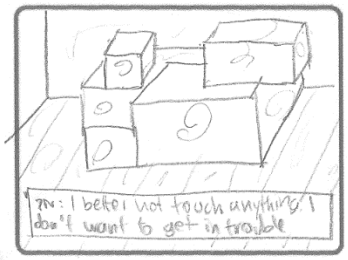
SQ: 2.2 SH: 1
Once MC gets to a certain distance to the pile of boxes, a DB with MC's thoughts will appear



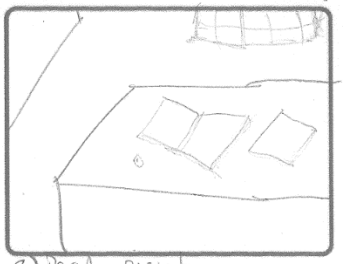
SQ: 2.2 SH: 2
DB with the two options



SQ: 2.2 SH: 3.a
MC breaks the pile of boxes and finds an electric gun

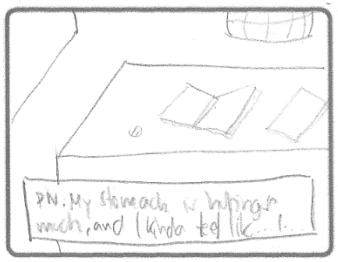


SQ: 2.2 SH: 3.b
MC decides not to break the pile. DB is updated



2) Real event

SQ: 2.3 SH: 1
IF MC has eaten the hot in the previous level he will die when this event activates



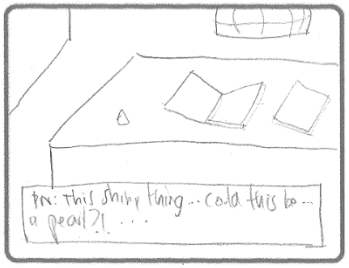
SQ: 2.3 SH: X.1
DB appears, image starts to get blurry



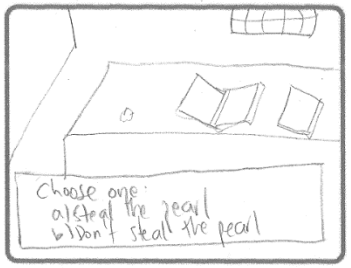
SQ: 2.3 SH: X.2
Image fades out. The screen is blacked out and a screen with "GAME OVER" appears.

SLUMBER storyboard
Yasmine González Bibed

Page: 7 of 17



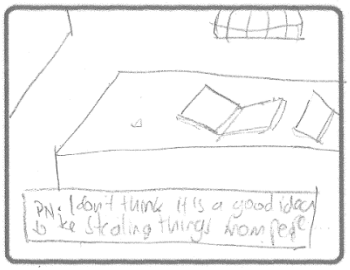
SQ: 2.3 SH: 2
DB box with MC's thoughts appear



SQ: 2.3 SH: 3
DB 'updates' showing both choices

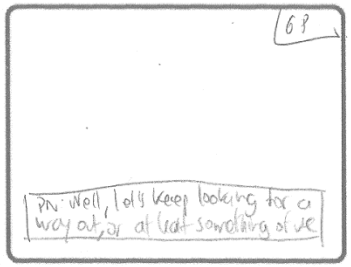


SQ: 2.3 SH: 4.a
MC decides to steal the pearl
DB updates

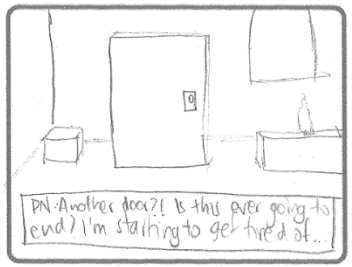


SQ: 2.3 SH: 4.b
MC decides not to steal the pearl DB updates

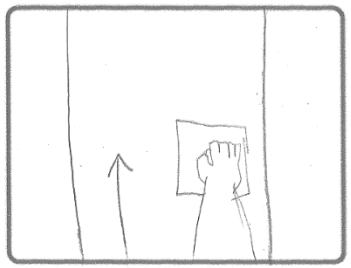
LEVEL 3: DEEP DARKNESS



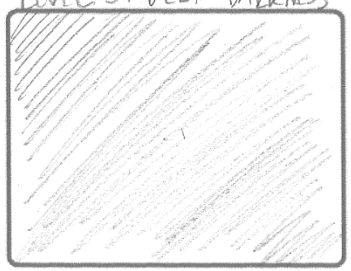
SQ: 2.4 SH: 1
DB appears showing MC's thoughts after both level events are cleared



SQ: 2.5 SH: 1
MC finds the newly appeared door. DB showing MC's thoughts appears



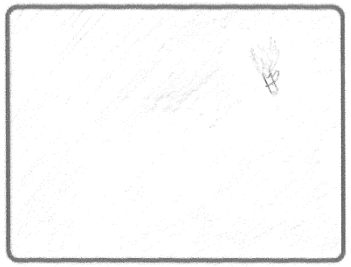
SQ: 2.5 SH: 2
MC goes through the door



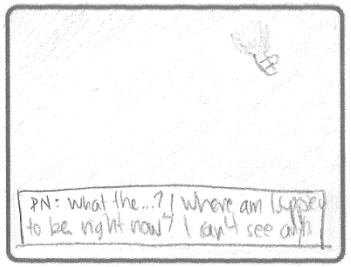
SQ: 3.1 SH: 0
Transition screen

SLUMBER storyboard
Yasmine González Bibed

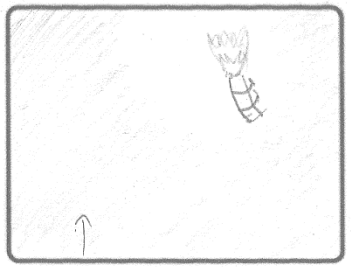
Page: 8 of 17



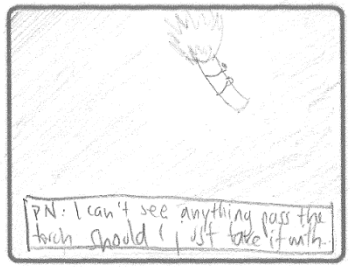
SQ: 3.1 SH: 1
MC finds himself in the darkness He can only see a torch ahead of himself



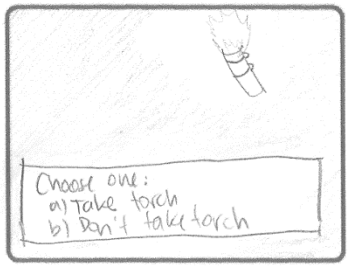
SQ: 3.1 SH: 2
DB showing MC's thoughts appears



SQ: 3.1 SH: 3
MC gets closer to the torch



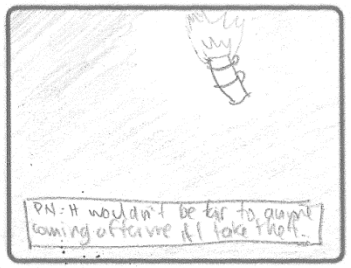
SQ: 3.2 SH: 1
When MC gets to a certain distance from the torch a DB appears



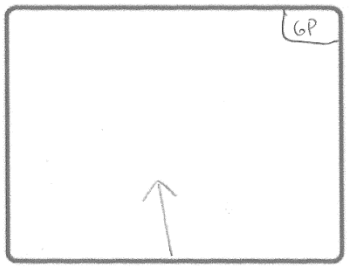
SQ: 3.2 SH: 2
DB updates with player's choices



SQ: 3.2 SH: 3.a
MC takes torch and is able to see what's ahead of him



SQ: 3.2 SH: 3.b
MC doesn't take torch



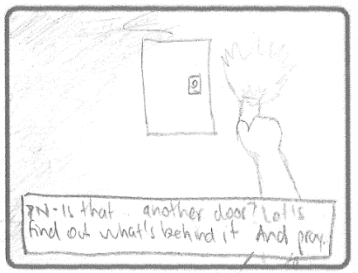
SQ: 3.3 SH: 1
Gameplay sequence where player may or may not fall into the hole that's ahead

SLUMBER storyboard
Yasmine González Bibed

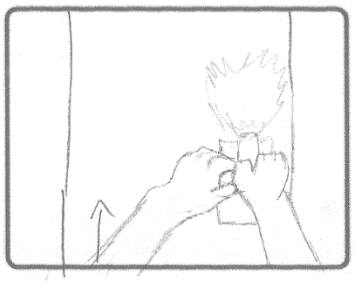
Page: 9 of 17



SQ: 3.3 SH: X
MC falls into the hole
Game over screen



SQ: 3.4 SH: 1
MC avoids the hole and finds another door. DB appears

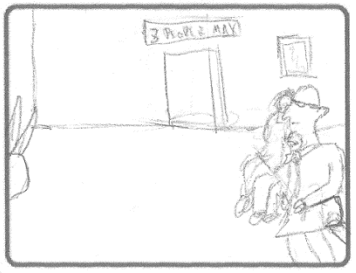


SQ: 3.4 SH: 2
MC opens the door and goes through it

LEVEL 4: WHITE ROOM



SQ: 4.1 SH: 0
Transition screen



SQ: 4.1 SH: 1
MC gets into a white room where he finds 3 people who stare back at him.



SQ: 4.1 SH: 2
DB appears with player thoughts



SQ: 4.1 SH: 3
The man stands up and DB shows what he's saying



SQ: 4.1 SH: 4

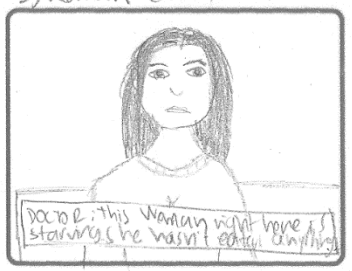
SLUMBER storyboard
Yasmine González Bibed

Page: 10 of 17

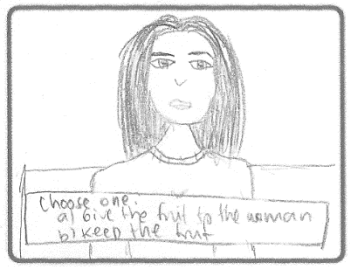
1) Woman event



SQ: 4.1 SH: 5
DB updates. Doctor points to the people behind him



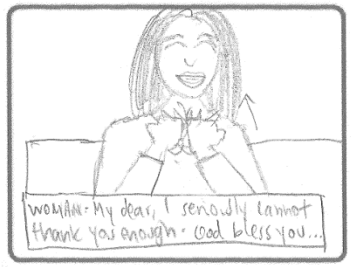
SQ: 4.2 SH: 1
DB updates. Doctor takes MC to the woman



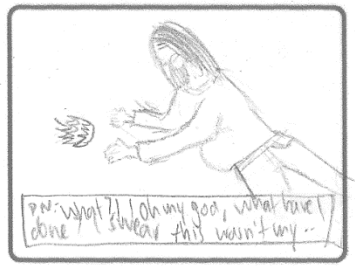
SQ: 4.2 SH: 2.a
MC has fruit, he has to choose between giving it to the woman or not



SQ: 4.2 SH: 2.a.a-1
MC gives woman the fruit
The woman takes the fruit



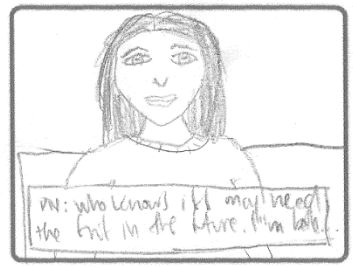
SQ: 4.2 SH: 2.a.a.2
The woman thanks the MC and starts eating the fruit



SQ: 4.2 SH: 2.a.a.3
Woman falls down to the floor and dies. DB updates



SQ: 4.2 SH: 2.a.a.4
Doctor examines the woman and then stands up and faces MC. DB updates



SQ: 4.2 SH: 2.a.b.1
MC keeps fruit to himself. DB updates and woman thanks the MC

SLUMBER storyboard
Yasmine González Bibed

Page: 11 of 17

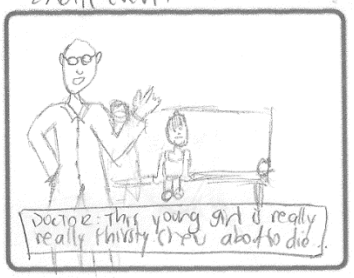


SQ: 4.2 SH: 2.a.b.2
Woman thanks MC. DB updates

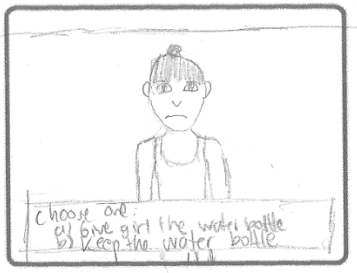


SQ: 4.2 SH: 2.b
MC doesn't have a fruit. He tells the woman and she thanks him anyway.

2) bird event



SQ: 4.3 SH: 1
Doctor points to the girl and talks to MC



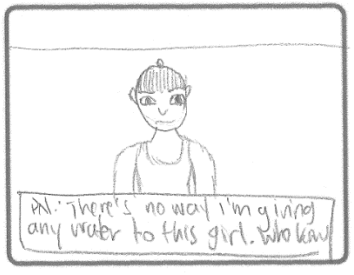
SQ: 4.3 SH: 2.a
MC has a water bottle. He has to choose between giving it to the girl or not.



SQ: 4.3 SH: 2.a.a.1
MC gives girl the water bottle she thanks him and drinks water



SQ: 4.3 SH: 2.a.a.2
Doctor thanks MC



SQ: 4.3 SH: 2.a.b.1
MC decides not to give the girl the water bottle



SQ: 4.3 SH: 2.a.b.2
Girl dies

SLUMBER storyboard
Yasmine González Bibed

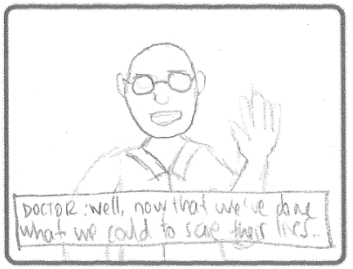
Page: 12 of 17



SQ: 4.3 SH: 2.b.1
MC doesn't have a water bottle

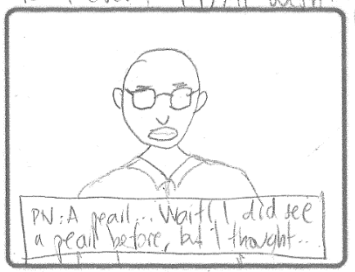


SQ: 4.3 SH: 2.b.2
Girl dies



SQ: 4.4 SH: 1
MC and doctor talk.
DB updates
1.2) MC has pearl

Pearl event 1.1) MC doesn't have pearl



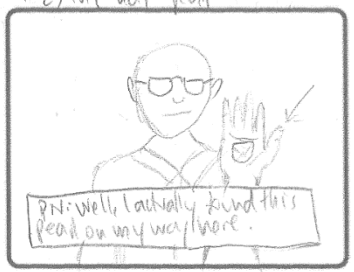
SQ: 4.5 SH: 1.X.1
Doctor talks to MC, and tells him about the pearl
1.2.1) Sacrifice



SQ: 4.5 SH: 1.X.2
Doctor does spray over MC. MC dies



SQ: 4.5 SH: 1.X.3
Game over screen



SQ: 4.5 SH: 1.1
MC hands pearl to doctor and he takes it. DB with 3 choices appears



SQ: 4.5 SH: 1.1.1
Doctor warns MC about the elevator. MC thinks about the option he has.

SLUMBER storyboard
Yasmine González Bibed

Page: 13 of 17

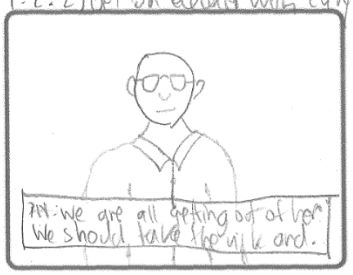


SQ: 4.5 SH: 1.1.1.1
MC tells doctor that he will sacrifice himself

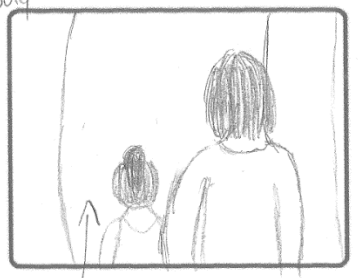


SQ: 1.5 SH: 1.1.1.2
Game over screen

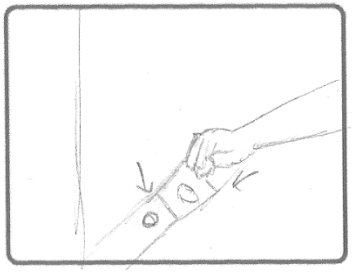
1.2.2) Get on elevator with everybody



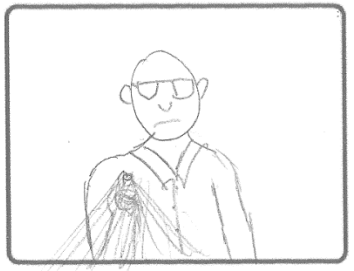
SQ: 4.5 SH: 1.1.2.1
MC tells doctor that they are all getting on the elevator



SQ: 4.5 SH: 1.1.2.2
MC, girl, woman and doctor get on the elevator



SQ: 4.5 SH: 1.1.2.3
Doctor places the pearl in the elevator's device and pushes the button



SQ: 4.5 SH: 1.1.2.4
MC looks at doctor and he pens spray over him and the other two



SQ: 4.5 SH: 1.1.2.5
The screen fades to black a DB appears

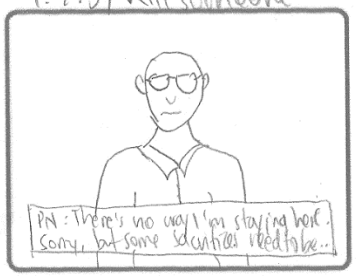


SQ: 4.5 SH: 1.1.2.6
Game over screen

SLUMBER storyboard
Yasmine González Bibed

Page: 14 of 17

1.7.3) Kill someone



SQ: 4.5 SH: 1.1.3.1
DB shows MC's thought. MC chooses who to kill

1.2.3.1) Kill girl



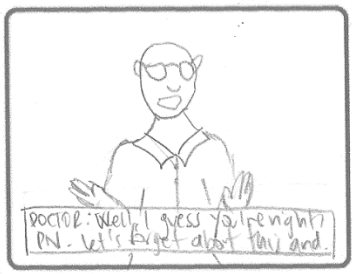
SQ: 4.5 SH: 1.1.3.1.1
MC goes to the girl and kills her with the electric gun



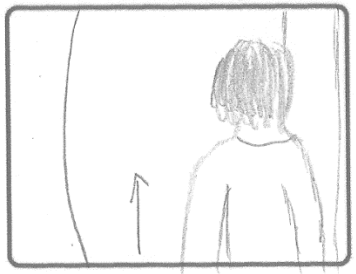
SQ: 4.5 SH: 1.1.3.1.2
Girl falls to the floor. MC apologizes



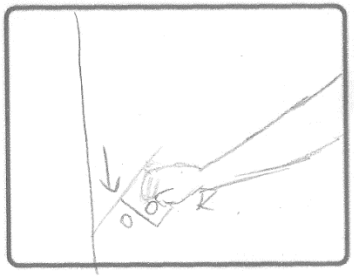
SQ: 4.5 SH: 1.1.3.1.3
MC gets closer to the girl, she talks to him and dies



SQ: 4.5 SH: 1.1.3.1.4
MC stands up and date talks to him



SQ: 4.5 SH: 1.1.3.1.5
MC, doctor and woman get in the elevator



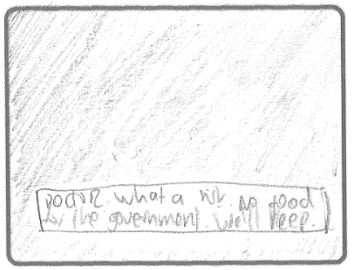
SQ: 4.5 SH: 1.1.3.1.6
Doctor places pearl, pushes the button



SQ: 4.5 SH: 1.1.3.1.7
The screen fades to black. A DB appears

SLUMBER storyboard
Yasmine González Bibed

Page: 15 of 17

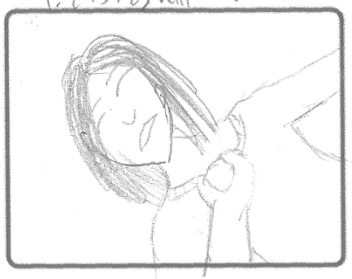


SQ: 4.5 SH: 1.1.3.1.8
the screen fades to black. A DR appears

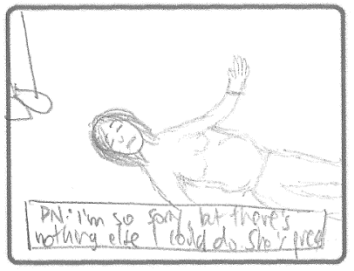


SQ: 4.5 SH: 1.1.3.1.8
game over screen

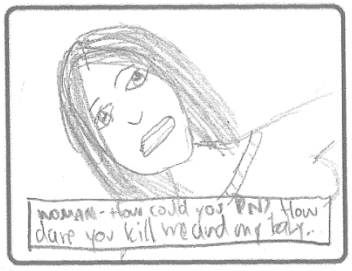
1.2.3.2) Kill woman



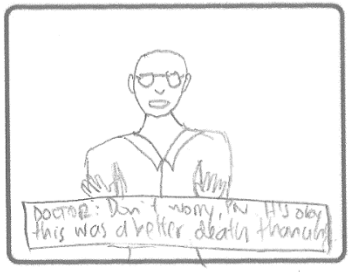
SQ: 4.5 SH: 1.1.3.2.1
MC goes to kill the woman with the electric gun



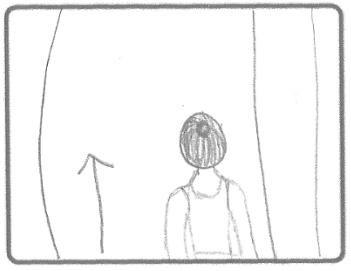
SQ: 4.5 SH: 1.1.3.2.2
Woman falls to the floor. MC apologizes



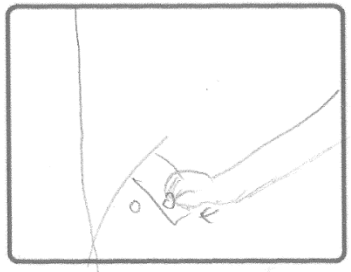
SQ: 4.5 SH: 1.1.3.2.3
MC gets closer to the woman, she talks to him and dies



SQ: 4.5 SH: 1.1.3.2.4
MC stands up and doctor talks to him



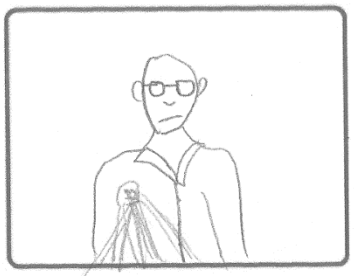
SQ: 4.5 SH: 1.1.3.2.5
MC, doctor and girl get in the elevator



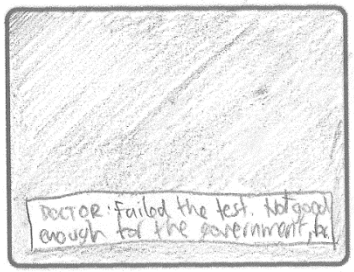
SQ: 4.5 SH: 1.1.3.2.6
Doctor places pearl, pulses the button

SLUMBER storyboard
Yasmine González Bibed

Page: 16 of 17



SQ: 4.5 SH: 1.1.3.2.7
Doctor puts spray over MC and girl. Screen fades to black



SQ: 4.5 SH: 1.1.3.2.8
DB appears



SQ: 4.5 SH: 1.1.3.1.8
Game over screen

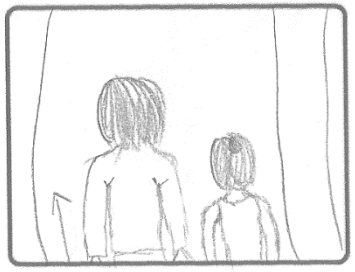
1.2.3.3) kill doctor



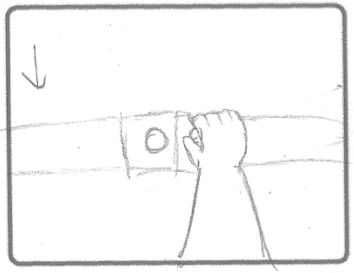
SQ: 4.5 SH: 1.1.3.3.1
MC kills doctor. Doctor falls to the ground



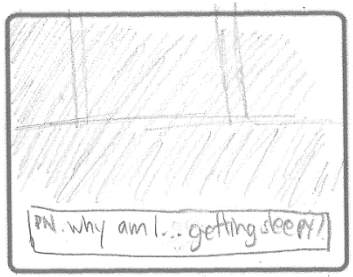
SQ: 4.5 SH: 1.1.3.3.2
MC gets closer to doctor. Doctor talks to MC



SQ: 4.5 SH: 1.1.3.3.3
MC, girl and woman get in the elevator



SQ: 4.5 SH: 1.1.3.3.4
MC places pearl and pushes the button. Elevator starts going down



SQ: 4.5 SH: 1.1.3.3.5
Elevator keeps going down, everything starts getting dark and MC falls asleep. (everything gets dark)

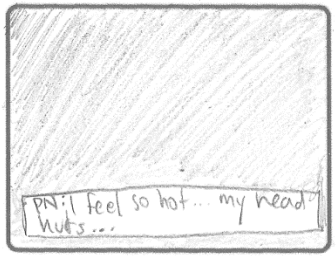
SLUMBER storyboard
Yasmine González Bibed

Page: 17 of 17

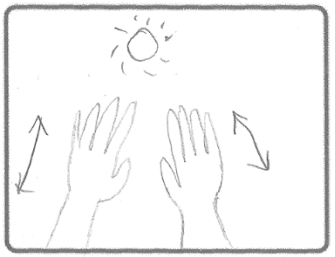
LEVEL X: DESERT



SQ: 1.S SH: 1.1.3.3.6
Transition screen



SQ: S.1 SH: 1
DB with MC's thoughts appears



SQ: S.1 SH: 2
MC opens his eyes and finds himself in the desert again



SQ: S.1 SH: 3
MC puts his hands down. DB appears



SQ: S.1 SH: 4
MC closes his eyes. finding screen

SQ: SH:

SQ: SH:

SQ: SH:

10. Annex 2. Video game script

"SLUMBER"

By

Yasmine González Bibed

Multimedia Engineering
Universitat de València

THE PLAYER WILL HAVE TO INTRODUCE THEIR NAME BEFOREHAND. PLAYER WILL BE REPLACED IN-GAME WITH THE ACTUAL NAME OF THE PLAYER.

LEVEL 0: DESERT

The scene starts in absolute darkness.

PLAYER

I feel so hot… It seems like I'm about to vanish

PLAYER OPENS HIS EYES AND SEES THE SUN IN THE SAME DIRECTION HE IS LOOKING AT.

(CAMERA PAN TO THE FRONT)

PLAYER FINDS HIMSELF LAYING ON THE SAND IN A DESOLATE DESERT WITH THE SUN SHINING BRIGHTLY OVER HIM. THERE'S NOTHING ELSE APART FROM THE SUN AND THE TONS OF SAND.

PLAYER STANDS UP ABRUPTLY.

PLAYER

Where am I? My head hurts so much I feel like it's about to explode.

Well, I might as well find a way to get out of here.

PLAYER STARTS WALKING STRAIGHT FORWARD UNTIL HE SEES SOMETHING.

PLAYER

Wait, what is that?

PLAYER KEEPS ON WALKING

PLAYER

Could that be… a door?

PLAYER IS ASTONISHED. HE FINDS A WHITE DOOR IN FRONT OF HIM, IN THE MIDDLE OF THE DESERT.

PLAYER OPENS THE DOOR AND GOES THROUGH IT

LEVEL 1: TROPICAL FOREST

The PLAYER finds a tropical forest when he opens the door. There are tall trees and a huge amount of vegetation, which barely allows the sun to shine brightly. The place also feels really humid.

PLAYER

What is this place? Why am I in a tropical forest all of a sudden? I only went through the door and…

PLAYER looks back at the door, but it is nowhere to be found

PLAYER

What??!! Where did the door go? Now there's no way I can go back.

Whatever, it's not like I'd want to go back to that desert.

Well, let's take a look around this place and see if I can find something that lets me get out of this place.

It seems like there are two ways I can go from here.

PLAYER starts exploring the forest.

There are two events in this level. They can take place in either way. After both events happen, another door that looks just like the one that was at the desert will appear.

Lake event

The player is thirsty, and wonders if drinking the water from the lake or not. Whether he does drink it or not, it won't affect the journey. He also has the option to store it in a bottle.

PLAYER

Oh god, look at this lake. It is definitely a vast change from the desert.

PLAYER gets close to the water of the lake.

Now that I think about it, I feel really thirsty. I don't know for how long I've been in that desert, but my throat feels as dry as an apricot.

I wonder if it's okay if I drink some water directly from the lake. I mean, it doesn't look like it's not drinkable but this is an unknown forest whatsoever.

I could also store it in this bottle I happen to have on me, just in case I need it in the future.

The PLAYER has 3 choices:

1) Drink water from the lake.

PLAYER

(Drinks water) Damn, that felt amazing. Let's keep going.

2) Do not drink water from the lake.

PLAYER

I better not drink this water. Who knows if it's infected with bacteria.

3) Store water in the bottle.

PLAYER

(Stores water in a bottle) There we go. Better to have some water than nothing at all!

After the event:

PLAYER

Let’s keep looking for something that gets me out of here.

Fruit event

The player is hungry, and wonders if it is okay if he eats the fruit he just found. If he eats it, he will be poisoned and will die in the next level. He can also keep the fruit and save it for later.

As he gets close to the clearing of the forest:

PLAYER

That's a strange rock over there... PLAYER gets closer to the rock. I should get closer to it.

PLAYER

Hold on, is that a fruit? It actually looks delicious.

Also, all of this wandering around has made me so hungry. I can’t even remember the last time I ate something. My stomach is growling so hard.

The PLAYER has 3 choices:

1) Eat the fruit.

PLAYER

(Eats fruit) Oh god, this fruit does taste delicious. It’s sweet and juicy. I’m so glad I found it.

2) Do not eat fruit.

PLAYER

I think I’ll pass. Lord know if this thing is edible at all.

3) Keep it in pocket.

PLAYER

(Puts fruit in pocket) I’d rather save the fruit for later. I think I can hold my hunger for a little longer.

After the two events have passed, another door will appear. A cutscene with the placement of the door will be played and PLAYER will be wandering the place until he finds the door.

PLAYER

Is that another door just like the one back in the desert?

I should get where it is and take a look

And when the PLAYER finally finds the door he gets through it and is taken to the next level.

LEVEL 2: GLOOMY ROOM

The PLAYER finds a gloomy and dark room when he opens the door. The walls of the room are made of old wood and there are stalls of boxes, bottles and dust everywhere.

PLAYER

This dark room… Where exactly am I? This looks like a pirate room straight out of a movie. Everything looks really creepy.

I might as well take a look around here. There might be something that could be useful to get the hell out of here.

PLAYER starts exploring the room.

There are two events in this level. They can take place in either way. After both events happen, another door that looks just like the ones that appeared before will appear.

Pile of boxes event

PLAYER

This pile of boxes looks crazy. I wonder what's stored in them. Maybe I can find something useful for my sake.

However, I don't really know whose room is this or if I am even allowed to be here. Maybe is not a good idea to touch anything.

The PLAYER has 2 choices:

1) Break pile.

PLAYER

(Breaks pile) Is that… an electric gun? Well, not exactly what I was looking for but I might as well keep it on me.

2) Do not break pile.

PLAYER

I better not touch anything. I don't want to get in trouble.

Pearl event

If PLAYER has been poisoned, he will die as soon as this event activates.

PLAYER

My stomach is hurting so much, and I kinda feel like… I… have… a… fever…

I can't- I can't- keep up- with this. (PLAYER DIES)

If not, his journey will go on.

PLAYER

This shiny thing… could this be… a pearl?!

Wait, does that mean this place actually belongs to a pirate? I'm so confused…

So, should I steal the pearl or leave it where it is? Who knows if I get in trouble if I even touch it.

The PLAYER has 2 choices:

1) Steal the pearl.

PLAYER

(Puts pearl in pocket) It doesn't look like the owner's here. Maybe he is not even alive. Whatever.

2) Do not steal pearl.

PLAYER

I don't think it's a good idea to be stealing things from people. I better keep away from it.

After the two events have passed, another door will appear. The PLAYER will be wandering the place until he finds the door.

PLAYER

Another door?! Is this ever going to end? I'm starting to get tired of this thing.

Whatever, let’s just find out what’s behind the door this time.

PLAYER finds the door and goes through it.

LEVEL 3: DEEP DARKNESS

The PLAYER gets into complete darkness after going through the door. There's only a torch attached to a medium height metal stick some meters away from the player. A few meters away from the torch there's a hole which will cause the PLAYER's death if he steps on it. After the hole there's another door which leads into the next level.

PLAYER

What the…? Where am I supposed to be right now? I can't see anything except from that torch. I should get closer to it.

PLAYER walks towards the torch.

There's one event in this level. Once the PLAYER gets close to the torch it will start.

Torch event

PLAYER

I can't see anything pass the torch. Should I just take it with me? This room is pure darkness and who knows what's ahead of me.

However, if I do take it, what if someone else comes into this room after me and the torch's gone? There's no way someone can do something without any type of light in here.

The PLAYER has 2 choices:

1) Take torch.

PLAYER

(Takes torch) Now I can see clearly what I am walking into.

2) Leave torch where it is.

PLAYER

It wouldn't be fair to anyone coming after me if I take the torch with me. I'll just walk into the darkness and hope that I can make it through.

Either way, the PLAYER should surpass the hole, in order to survive and not fall down the hole. If the PLAYER takes the torch he will be able to spot the hole and avoid it, otherwise, the PLAYER is more likely to fall down the hole and die.

If PLAYER avoids the hole:

PLAYER

Is that… another door? Let's find out what's behind it. And pray for this nightmare to end already.

(PLAYER opens the door and goes through it.)

LEVEL 4: WHITE ROOM

The PLAYER gets into a white room, where there's two people sitting on a bench on the right side of the room: a WOMAN and a GIRL, and there is a man in a suit (DOCTOR) standing beside them. The man is looking at the two people sitting on the bench, who look like they are in pain. The walls are entirely white, there's a small plant in the far left corner of the room, few decorations and a door in the far center of the room.

PLAYER

Where am I right- Wait, is that actually people??!!

Those two women look in pain, so I better go talk to the man standing over there

PLAYER looks to the three people in the room. And gets close to the man in the suit in order to talk to him.

PLAYER

Hello! Who are you? What are you doing here?

The DOCTOR turns to the PLAYER.

DOCTOR

Hello. I suppose you must be confused right now. First of all, let me ask you: are you alright?

PLAYER

Yes. Thank you.

DOCTOR

Nice to meet you, I am simply a doctor that is stuck in here, just like you and these two people beside me. I will briefly explain to you what is going on here. But first, I need you to listen to me for a second.

This woman right here is starving. She hasn't eaten anything in many days, and she could die in any moment due to the fact that she is pregnant.

She won't last long if she doesn't anything soon. Do you happen to have any food on you to save her life by any chance?

You better go talk to her yourself and see if there's something you can do to help her. Also, come see me after you're done with her. There's something else I need from you.

Woman event

1) If PLAYER has a fruit, he has 2 options:

PLAYER

(Thinking to himself) So, what should I do? Should I give her the fruit or keep it for myself? Who knows how long I am going to be wandering around.

1) Give woman the fruit. WOMAN will die because the fruit is poisoned.

PLAYER

Oh, well I happen to have a fruit I picked up before. Here, take it (PLAYER gives fruit to the woman).

WOMAN

My dear, I seriously cannot thank you enough. God bless you. You've just saved mine and my baby's life (WOMAN eats fruit).

Hold on for a second… this fruit… IS POISONED (WOMAN dies).

PLAYER

What?!! Oh my god, what have I done? I swear this wasn't my intention.

What should I do now...? I can't let her die. Not like this. I think I'd better go talk to the doctor.

2) Do not give the woman the fruit.

PLAYER

(Thinking to himself) Who knows if I may need the fruit in the future. I'm better off keeping it for myself. Besides, I don't even know this woman…

(To the woman) Hmm… sorry, I don’t have any fruit on me.

WOMAN

Don't worry dear, thank you anyways.

2)If PLAYER doesn’t have a fruit:

PLAYER

(Thinking to himself) Damn, I feel so bad... I should've taken the fruit I saw back in the forest...

My apologies. I did see a fruit on my way here, but I thought it’d be best not to take it.

WOMAN

Don’t worry, dear. I shall thank you anyways.

PLAYER goes to see DOCTOR after the WOMAN event. After the WOMAN event if player didn’t give the fruit to WOMAN:

PLAYER

It looks like there's nothing I can do to help her.

DOCTOR

Well... no worries. She will survive if she's strong enough. If she isn't then she's simply not worth living. Anyways, you should talk to that girl. You may be able to help her.

As far as I know that young girl is really really thirsty. She is about to die of dehydration. If you happened to have some water on you, you'd definitely save her life for sure.

Otherwise, if PLAYER gave the fruit to WOMAN:

PLAYER

Doctor... what has happened??!! That fruit... that wasn't my intention... We have to saver her.. We can't let her die... not like that...

DOCTOR

It's no use. She's dead. But don’t put the blame on yourself. She was too weak already, it was about time she would die. So forget about it.

Now then, as far as I know that young girl is really really thirsty. She is about to die of dehydration. If you happened to have some water on you, you'd definitely save her life for sure. Now go talk to that girl if you don't want to let another life go to waste

Girl event

PLAYER goes to GIRL and talk to her.

Don’t you happen to have any water on you?

1) If PLAYER has a water bottle, he has 2 options:

PLAYER

(To himself) Do I give her the water I stored in the bottle? Or should I just keep it for myself? I may not be able to make it much longer without drinking any water.

1) Give the girl the water bottle. The girl will survive.

PLAYER

Here! I filled this bottle with water from a lake I found on the way here. I hope it makes you feel better (PLAYER gives the water bottle to the girl).

GIRL

Thank you so much! (GIRL drinks water) Water never ever tasted this good before. I truly can’t thank you enough.

PLAYER

No problem. I’m glad I filled the water bottle.

(To himself) I’m so happy I was able to help this girl out. Now I should go back and talk to the doctor to see if there’s something else I can do to help them.

2) Do not give the water bottle to the girl. The girl will die because of dehydration.

PLAYER

(Thinking to himself) There’s no way I’m giving any water to this girl. Who knows what the future holds for me. I should be prepared for whatever may happen.

(To the girl) I don’t, unfortunately.

GIRL

No… probl- (GIRL dies).

2)If PLAYER doesn’t have a water bottle:

PLAYER

(To himself) I really should have filled the water bottle... Well let's just talk to her...

Hello! How are you? I don’t have any water on me, sorry. I could’ve filled a bottle with the water from the lake, but I chose not to.

GIRL

It’s.. okay… (GIRL dies).

PLAYER

Oh my god! I can't believe this... I feel so bad right now... I have to save her, let's talk to the doctor. There has to be something I can do to save her

Pearl event

DOCTOR

Well, now that we've done what we could to save their lives, I shall give you a glimpse of what's going on here. Stop me at any point if you're lost.

PLAYER

Alright, sure I will.

DOCTOR

I suppose you've gotten through many doors to get here. Well, turns out that all of this is an induced dream, this is not real at all.

PLAYER

An… induced dream you say?

DOCTOR

Exactly. I know it is hard to believe, but all of us in this room are in an induced dream. This is all set up by the government, which is trying to create a utopian society. In order to do so, they are taking every single individual in the society: kids, babies, adults, elders…, and putting them through this induced dream.

This way they are able to get rid of those who don't act according to the government's ideals. And thus, only the ones who make the decisions the government wants will survive.

PLAYER

So... a plan of the government to create an utopian society?

But wait… How come I don't remember anything I've done prior to this supposed induced dream apart from my name?

DOCTOR

That's easy. All of our memories have been erased in the process of getting into this dream. Call it a brain-wash if you'd like. Of course they need you to know your own name, so it's easier for them to classify us.

PLAYER

It's really hard to digest all of this, to be quite honest. So, how are we supposed to get out of this dream?

DOCTOR

There’s only one way out of this dream. And that is by going through that door.

However, in order to make it work we need a special pearl which will open it. No one that has made it so far has had it with them.

And there's another issue. Only three people are able to go through the door, and then it will shut instantly

If PLAYER has the pearl he will survive, otherwise the DOCTOR will kill him.

1) If PLAYER doesn’t have a pearl.

PLAYER

A pearl… Wait! I did see a pearl before, but I thought it’d be best not to take it.

DOCTOR

Well, you’ve made the wrong decision. (DOCTOR kills PLAYER).

PLAYER dies.

2) If PLAYER has a pearl.

If PLAYER has electric gun, it depends on the amount of people that survived in the room. If PLAYER does not have the electric gun:

DOCTOR

Hold on... it looks like you don't have anything on you that allows you to get rid of someone.

Then you’re not worthy of living in our new society.

Die.

DOCTOR kills PLAYER.

If PLAYER has a electric gun

PLAYER

Well, I actually found this pearl on my way here.

DOCTOR

Wonderful! I can’t believe this! You’re our savior. Thank you, now we will be able to make get out of here.

1) if there is three or less people left in the room

DOCTOR

Please lead the way to the door.

PLAYER will go to the door and will be killed by the doctor from behind.

DOCTOR

Failed the test. You simply do not follow the government’s ideals.

Now die.

DOCTOR kills PLAYER

2) If there are more than 3 people left in the room

DOCTOR

However, there’s a problem. Only three people can go through the door and then it will shut instantly. So only three of us can go through it. One of us has to stay here or we could all three try to go through it, but I'm scared it's not going to work. It is up to you, after all, you’re the one who brought the pearl.

Also, before you make a decision, let me tell you something important. Not only I’m a prestigious doctor, but I only carry with me the documents that will tear down the government and stop this madness.

Now it's all down to what you think it's best.

PLAYER

(To himself) The doctor’s right. We could take the risk and try to go through it being more than three people. Also, maybe the girl doesn’t count as a person. Otherwise someone has to be left behind. But who if so?

(To himself) The doctor has the documents to stop the government. And there’s no way I’m either leaving a pregnant woman or a girl behind, right? I guess I could also sacrifice myself, it’s for the best, after all.

PLAYER has three options:

1) Sacrifice himself

PLAYER

There’s no way I’m leaving any of you behind, and it’s dangerous if we all get in the elevator. I guess I’ve played my part. It all ends here for me.

Thank you doctor, I hope you can tear down the government. I’m counting on you.

DOCTOR

Sure then, someone who thinks this way is not worth of our new society. Now die, I’ll do the honors.

DOCTOR kills PLAYER

2) Try it being more than 3 people

PLAYER

We are all getting out of here. We should all take the risk and go for it.

DOCTOR

Sure then, someone who thinks this way is not worth of our new society.

Now die, I’ll do the honors.

DOCTOR kills PLAYER

3) Sacrifice someone

PLAYER

(To himself) There's no way I'm staying in here. Sorry, but some sacrifices need to be made.

PLAYER can kill one out of the three:

- Kill GIRL:

PLAYER

(To himself) Anyways, she's only a girl. She's of no use to save the rest of the society. I should go to her and do it.

PLAYER goes to GIRL AND kills her.

PLAYER

Sorry! I had no other option. Anyways, she's only a girl. She's of no use to save the rest of the society. I should forget about it, and lead the way to the door.

PLAYER goes to the door

DOCTOR

Failed the test. You simply do not follow the government's ideals.

Now die.

DOCTOR kills PLAYER

- Kill WOMAN:

PLAYER

(To himself) I'm so sorry, but there's nothing else I could do. She's pregnant and she's not going to make it anyways. Besides, we don't even know what's ahead of us. I should go to her and do it.

PLAYER goes to WOMAN and kills her.

PLAYER

Sorry! But it had to be you... I'll head to the door and see what I can do about the door and the pearl.

PLAYER goes to the door.

DOCTOR

Failed the test. You simply do not follow the government's ideals.

Now die.

DOCTOR kills PLAYER

- Kill DOCTOR: PLAYER will survive.

PLAYER kills DOCTOR

PLAYER

Sorry! But it had to be you... I'll head to the door and see what I can do about the door and the pearl.

PLAYER goes to the door and crosses it using the pearl.

LEVEL X: DESERT

PLAYER goes to the desert from Level 0. He finds himself laying in the sand, just like the first time.

PLAYER

I feel so hot… my head hurts…

Wait… why am I…

here again?

END

11. References

- “2018 Video Game Industry Statistics, Trends & Data - The Ultimate List.” *WePC.com*, 6 June 2019, www.wepc.com/news/video-game-statistics/.
- “2019 Essential Facts About the Computer and Video Game Industry.” *Entertainment Software Association*, 29 May 2019, www.theesa.com/esa-research/2019-essential-facts-about-the-computer-and-video-game-industry/.
- Chikhani, Riad. “The History Of Gaming: An Evolving Community.” *TechCrunch*, TechCrunch, 31 Oct. 2015, techcrunch.com/2015/10/31/the-history-of-gaming-an-evolving-community/.
- Dobrilova, Teodora. “How Much Is the Gaming Industry Worth in 2019? - TechJury.” *Tech Jury*, 4 Apr. 2019, techjury.net/stats-about/gaming-industry-worth/.
- Dudkin, Ilya. “Gaming and Technology: How Video Games Have Changed Over Time.” *Skywell Software*, Skywell Software, 7 Dec. 2018, skywell.software/blog/gaming-and-technology-how-video-games-have-changed-over-time/.
- “Game Engines - How Do They Work?” *Unity*, unity3d.com/what-is-a-game-engine.
- History.com Editors. “Video Game History.” *History.com*, A&E Television Networks, 1 Sept. 2017, www.history.com/topics/inventions/history-of-video-games.

- "The History of Educational Video Gaming." *Immersed Games*, www.immersedgames.com/the-history-of-educational-video-gaming/.
- Johnson, Neilie. "Gemini Rue Review." *IGN*, IGN, 8 May 2012, www.ign.com/articles/2011/04/01/gemini-rue-review.
- "How Much Does It Cost to Make a Video Game?" *Vironit. Custom Software Development.*, 13 Sept. 2018, vironit.com/how-much-does-it-cost-to-make-a-video-game/.
- Gómez, Isabel. Graduada en Derecho por la universidad CEU San Pablo.Doble Master de Acceso a la Abogacía y Asesoría Jurídica de Empresas en el Instituto de Empresa.Programa Executive Fashion & Law. "CASI PERO NO: CASOS DENEGATORIOS DEL COLOR COMO MARCA." Enrique Ortega Burgos, enriqueortegaburgos.com/casi-pero-no-casos-denegatorios-del-color-como-marca/.
- "Blind Forest." Ori, www.orithegame.com/blind-forest/.
- Fate/Grand Order Official USA Website, fate-go.us/.
- "Out Now for Windows, Nintendo Switch, PlayStation 4, Mac, and Linux." Firewatch, www.firewatchgame.com/.
- studio, Ler digital. Beat Saber - VR Rhythm Game, beatsaber.com/.
- Brown, Anna. "Who Plays Video Games? Younger Men, but Many Others Too." Pew Research Center, Pew Research Center, 7 Aug. 2020, www.pewresearch.org/fact-tank/2017/09/11/younger-men-play-video-games-but-so-do-a-diverse-group-of-other-americans/.

- “Male and Female Gamers: How Their Similarities and Differences Shape the Games Market.” Newzoo, 3 Jan. 2020, newzoo.com/insights/articles/male-and-female-gamers-how-their-similarities-and-differences-shape-the-games-market/.
- Richter, Felix. “Infographic: The Most Important Gaming Platforms in 2019.” Statista Infographics, 18 Mar. 2019, www.statista.com/chart/4527/game-developers-platform-preferences/.
- “2018 Video Game Industry Statistics, Trends & Data - The Ultimate List.” WePC, 14 Aug. 2020, www.wepc.com/news/video-game-statistics/.