Degree's Final Project Report
Bachelor's Degree in Video Game Design and Development

# Design and Development of a playable 3D adventure game in an ice world setting. Walpurgis Night: Ice Souls

Author: Irene Pérez Collado
Tutor: Antonio Morales Escrig

# 1. Summary

This document introduces the technical proposal for the final degree project for the Design and Development of Videogames degree. This project consist in the development of a game built on Unity 3D [1] which genre is a fantasy adventure RPG [2] game based on Nordic mythology. In this project, the players will be challenged to overcome ordeals all along the game, for this they need to use and/or combine the master skills of the four Valkyries that they will control during a limited time.

One of the objectives of the project is the application of using Motion Capture [3] techniques to try and give more credibility to the character animations in the game.

## 2. Key Words

- RPG

- Adventure

- Mythology

-Ice and Snow

- Weather

- Night

# Index

# Figure Index

# Table Index

# Nomenclature

3D – Three Dimensional

RPG – Role-Playing Videogame

TFG - Final Degree Project

AI – Artificial Intelligence

2D- Two Dimensional

C++ - General-purpose programming language

GDD – Game Design Document

NPC –non-player character

PNG – Portable Network Graphics

MGS2 – Metal Gear Solid 2

# 3. Technical Proposal

## 3.1 Introduction and Reasons for the Project

The development of this TFG is linked to the desire of the developing and completing a previous work that it was initialized with the collaboration of another two classmates during the third year of the degree. The aforementioned project involved a great effort at narrative level but unfortunately could not be finished.

Walpurgis Night is a game that mix the genres of role and adventure. The aim of the game is to face the enemy, an evil organization that is planning to destroy the world to create it anew thus becoming their new deities. The way to stop their plans is to avoid the Ragnarok [4] by finishing a series of dungeons; in order to do this, the players have to control a group of Valkyries [5], each of them with unique abilities, and be reunited with the Valkyries that have been kidnapped by the enemy before the midnight of the Walpurgis Night [6] and invoke Walpurga [7], a goddess, to provide the Earth with her protection. The mission of Brunilda [8], who is the main character, it to gather all the other Valkyries that have being kidnapped and scattered all around the Yggdrasil's [9] worlds.

Among the nine worlds included in the game concept, Jötunheim [10], the kingdom of ice giants is the one selected to be developed for the present TFG. This world is based in the strategy to overcome challenged related to ice physics.

For the remaining eight worlds, two have been selected by my peers:

- Niflheim [11]: The Mist Kingdom based on infiltration, where deceiving techniques are preferable.
- Svartalfheim [12]: The Dark Elves Kingdom based on stealth and where strategy and interplaying with the scenario are needed.

Given that these projects are a part of the same game where every world is a different level, the share and common readymade elements are:

- The model of the three principals Valkyries and their basic body structure.
- The model of the object and basic weapons.

- The user interface and the game menus.
- The basic behaviours of the enemies (Actions like walk and attack).
  - Some musical pieces and sounds effects.

As it has been said, this project is based on Jötunheim, a world of ice and snow where the player can control three Valkyries (Geminis, Pisces and Aries), who have to discover this new world while looking for their imprisoned companion, being Taurus the one that they have to rescue in this level. This level stands out for the use of the simulation of physics that the ice grant and its ability to develop puzzles and strategies that will challenge the player abilities.

Based on the fact that this is a world with those characteristic, the landscape that is presented is a cold world with different shades of blue and textures and also snow and hail zones. These zones and the game tonalities will be affected by the present time in the game that it will affect the difficulty that the player will confront.

The project genre (RPG) is based in exploration, narrative and the use of character's skills in the game and of the player too in order to surpass a series of obstacles, e.g. Golden Sun [13], where the player can move objects to clear the way by solving smalls puzzles or riddles. Nonetheless many of these abilities are wasted due to shift fighting. In this project, the aforementioned characteristics are going to be used in battle.

In Jötunheim, there are two types of basic enemies, and ice golem and a stone golem. The stone golem is way stronger and has more resistance but the ice golem can freeze the Valkyries.

The initial point of the game are the ruins and from this point, the player will advance to Utgard [14], the giant fortress to discover where Taurus is. The interaction with NPCs will be available and the objective is to visit the dungeon where the Valkyrie is being arrested. In this dungeon, the player has to overcome the obstacles to go to other rooms while has to defend himself against the enemies that harass him. This level will finish after rescuing and saving Taurus, the imprisoned Valkyrie, and finally defeating Thrym [15], king of Jötunheim.

Two types of attacks are available for the player in order to overcome these challenges: physical and magic attack.

- The magic attack is dependent of the manna that the character possesses at that moment and it can be filled using objects that are in the scene and also, with the time. There are also objects available that restore health points and time.

Finally it is useful to emphasize different details that can be found throughout the game, such as the use of the ice physical properties to develop puzzles or the great narrative weight of time, as well as, as the use of motion capture techniques to provide more realistic animations depending on the character as previously mentioned in the abstract.

## 3.2 Related Subjects

**VJ1216 3D Design**

- All the elements in the scene (scenarios, objects, characters) have been modelled thanks to the tools and skills learnt in this subject by using the program 3DS max [16].

**VJ1222 Conceptual Design of Video Games**

- The main target of this subject is to be able to create and analyse games developing some understanding that allows the creator to know the keys for their correct operation and development.

- To design correctly this level, it is necessary to understand these acquired concepts to build a balanced world.

**VJ1223 Video Game Art**

- In a game, the graphic and artistic items are of paramount importance as these are what are exposed to the player´s eyes influencing his predisposition to play or leave it

- The necessary skills to develop Graphic Game Elements (GUI, Textures and Game Concept Arts) are acquired in this subject.

**VJ1226 Character Design and Animation**

- The objective of this subject is to obtain the skills to develop and create a character that can be animated.

This subject is the essential for the TFG due to the use of the Motion Capture System that will help us to obtain a skeleton with the information about animations much more credible and fluid than if they were sketched by hand.

**VJ1227 Game Engines**

- TFG is going to be developed in the Unity 3D engine, program taught in this subject.

**VJ1231 Artificial Intelligence**

- As in every game, the player will face some enemies that will try to prevent him from reaching their goals. These enemies will have some basic AI to patrol and attack the player as soon as the player is detected by them.

In addition to these subjects, there are others such as:

**VJ1201 Physics**

- Due to the intention to obtain realistic Ice physics, this subject is related.

**VJ1204 Artistic Expression**

- In a game where it is intended that the art has a great weight, it is necessary to obtain the ability to sketch the elements that must appear on the scene.

**VJ1205 English**

- To demonstrate the ability to write a technical text in English.

**VJ1224 Software Engineering**

- This subject teaches the ability to plan, conceive and direct projects as well as the development and evaluation of computer systems to evaluate their reliability. Given that this is a big project, the planning and controls are needed.

## 3.3 Objectives

- Development of a scenario with several 3D modelling and animations

- To show the influence of time in the behaviour of the enemies and in the scenarios.

- Present a functional demo

- Learning the use of capturing movements' methods by using cameras and specialized suits.

- The ability to model and adapt a story based on concepts of mythology known in a video game

## 3.4 Task Planning and Timing

It is planned to start once the proposed work is accepted, at the present moment some concept arts are already made.

The following weekly work distribution is scheduled:

- 2 days for the Modelling
- 2 days for programming
- 3 Various activities such as preparation of memory, documenting or testing.

The force idea is having most of modelling tasks finished by the end of May after that time only small details is expected to remain and consequently programming is expected to occupy most of the time.

| TASK | DESCRIPTION | DEPENDENCE | HOURS |
|---|---|---|---|
| **A** | Setting up mechanics, characters, objects… Development of the GDD | -------- | 7 |
| **B** | Sketches and setting up art, development of the concept art | A | 20 |
| **C.1** | Main characters and enemies modelling in 3ds max | A, B | 40 |
| **C.2** | Scenarios and its elements modelling in 3ds max | A, B | 40 |
| **C.3** | Objects modelling | A, B | 20 |
| **D** | Capture and generation of the characters' animation | C.1 | 10 |
| **E.1** | Programming main characters and enemies characteristics | C.1 | 30 |
| **E.2** | Programming AI and mechanics that will appear in the game | E.1, C.2 | 40 |
| **E.3** | Environment programming, setting up the environment final results | C.2, C.3, E.2 | 35 |
| **F** | Retouching and detail implementation (finishing the aesthetic part and retouching little details) | Previous steps | 5 |
| **G** | Prepare the project report | Previous steps | 40 |
| **H** | Prepare the project presentation | Previous steps | 6 |
| **I.1** | Weekly testing (test and debug the mechanics implemented) | C, E | 2 |
| **I.2** | Final testing (test and play the game like a user would do) | F | 5 |

*Table 1: Project's Schedule*

| Week | Task (hours) | Artistic Task (hours) | Programming Task (hours) | Hours in the week |
|---|---|---|---|---|
| 20/02 - 06/03 | A (7) | B (20) | | 27 |
| 06/03 - 20/03 | | C.1 (10)<br>C.2 (10) | E.1 (5)<br>D (5) | 30 |
| 20/03 - 03/04 | | C.1 (10)<br>C.2 (20) | E.1 (10) | 40 |
| 03/04 - 17/04 | | C.3 (10) | E.2 (10)<br>E.3 (10) | 30 |
| 17/04 - 01/05 | | C.1 (10) | E.1 (10)<br>E.2(10) | 30 |
| 01/05 - 15/05 | | C.1 (10)<br>C.3 (10) | E.1 (5)<br>E.2 (5) | 30 |
| 15/05 - 29/05 | | C.2 (10) | D (5)<br>E.2 (15)<br>I.1 (1) | 31 |
| 29/05 - 12/06 | | | E.3 (25) | 25 |
| 12/06 - 26/06 | G (20) | | F (5)<br>I.1 (1) | 26 |
| 26/06 - 09/07 | G (20)/H (6) | | I.2 (5) | 31 |
| TOTAL | | | | 300 |

*Table 2: Project Organization with the partial and total hours for each task*

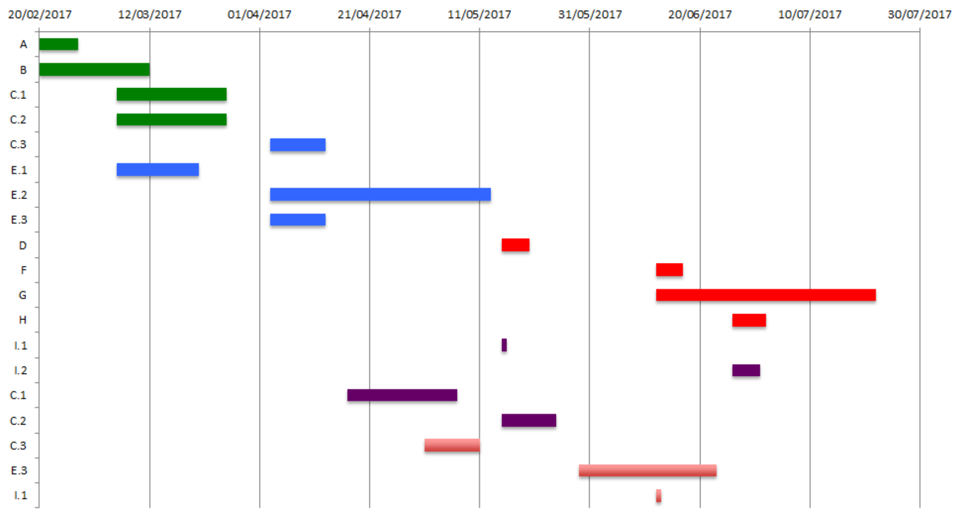


*Figure 1: Project schedule illustrated with a Gantt chart*

## 3.5 Expected Outcome

-The presentation of a functional and attractive game with many possibilities of further development. It is thought to continue and expand the game by creating a whole with other worlds designed by partners on their own projects.

- Working in a project like this will increase self-confidence

    I have had to carry out the project just by myself.

    I will have to defend it before a board of examiners.

    It will show what I have learnt over this last four years.

The goal is to be able to present a playable demo and videos at the TFG project presentation and defence.

## 3.6 Tools

- OptiTrack [17]: Motion Capture Systems []: Movement capture tool to be used in animations
    - *To see more of this tool, see the Appendix*

- Unity 3D: Engine to be used to develop the game.

- Autodesk 3DS Max: Used in characters, object and environment modelling.

- Adobe Photoshop CC [18]: Used in textures and some elements of the user interface

- MotionBuilder [19]: Used to clean the animations and make loops

- Microsoft Word: Used to write the documentation of the project.

# 4. Related Work

## 4.1 Techniques that have been used

As it has been said this Project started as a game developed for three subjects.
VJ1223 Videogame Art (Arte del Videojuego).
VJ1222 Conceptual Videogame Design (Diseño Conceptual del Videojuego).
VJ1224 Software Engineering (Ingeniería del Software).

This previous game, whose name was, Walpurgis Night, was conceived and developed by a three-member team: I and two more partners.
In order to create this game, it was necessary to coordinate us. Each one of us prepared what needed to every subject e.g.

- **Videogame Art** it was necessary to prepare some character sketches and establish a defined artistic style.
- **Conceptual Videogame Design** (Ten pages, GDD…) was organized so it explained these matter topics, such as the design of the story, and the level to be presented.
- **Software Engineering** a subject whose aim is to teach how to plan in order to meet project deadlines, in addition to teach new working tools.

The initial stage was developed in Visual Studios [20] using Cocos2d-x [21]. Cocos is an open source cross-platform unified package of game development tools which allows developers to create 2D games using C++ programming language, with this program it is possible to access to a library made up of physics, collisions, animations, audio rendering and more tools needed in game development.

However, loading time of rendering, the null existence of interface and possible problems caused at installation are troubles that are also found in this platform.

This type of setbacks is one of the reasons why this project is developed in Unity 3D because this tool is faster to develop scenarios and collisions and much more intuitive, besides it offers the possibility to create bigger projects and scaling up from 2D pixel art to 3D.

The original design was conceived and planned for 2D therefore the initial approaches were based in simpler objectives and mechanics, due to the constraints inherent to 2D.

The way animations are obtained is one of the big differences found, while in Cocos2D different postures were made in Photoshop as well as attack actions and casting magic because the artistic style used was pixel art a sprite sheet was prepared with animations. In Unity movement capture was made with OptiTrack to pull out the skeleton used to animate 3D characters.

Aelune Chronicles is another related work, it was made on the second year of the degree for the subject Design and Development of Web Games with another two classmates, although it is true this game was a turn-based combat game with box-style battle scenarios as can be found in Fire Emblem [22] or Final Fantasy Tactics [23], these projects share a big narrative weight in the story, the use of dialogues among characters or a stats system for every character.

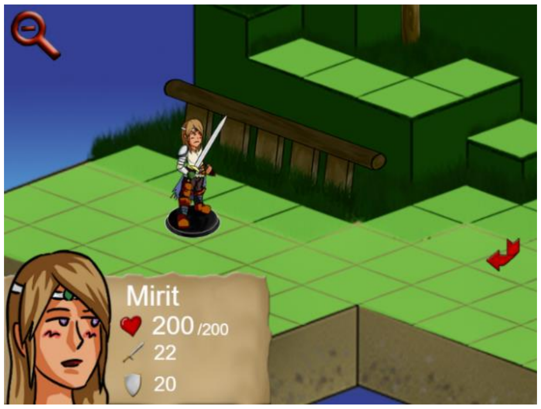


*Figure 2: Screenshot from the game Aelune Chronicles*

## 4.2 Games on which the Project is based

### 4.2.1 Mechanics

Golden Sun was one of the main sources of inspiration for the original design, in this game player can control four interchangeable characters, each one of them had a power and specific abilities needed to go on and overcome puzzles and difficulties that could appear in their way.

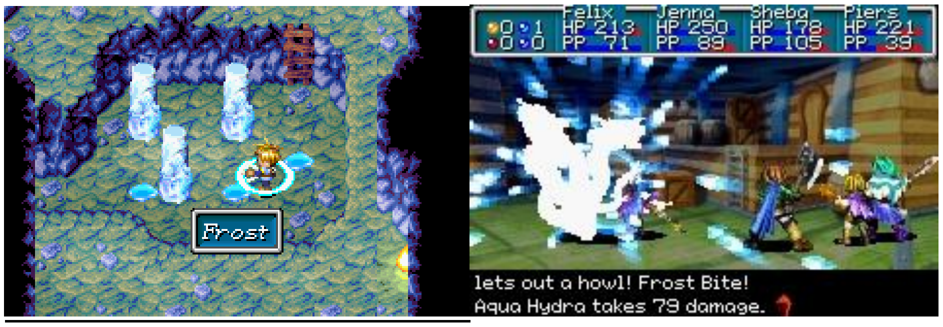


*Figure 3: Screenshot from Golden Sun using their abilities in and off battle. Images from Internet*

The legend of The Legend of Zelda: Majora´s Mask [24] is the main source for the role of time in controlling and influencing the progress of the game. In this game, the aim is to save the world by avoiding the moon crash on the earth in 72 hours. Player must repeat this three-day cycle to stop the catastrophe; in the end, he will acquire a certain power over the passage of time in order to achieve the goals.

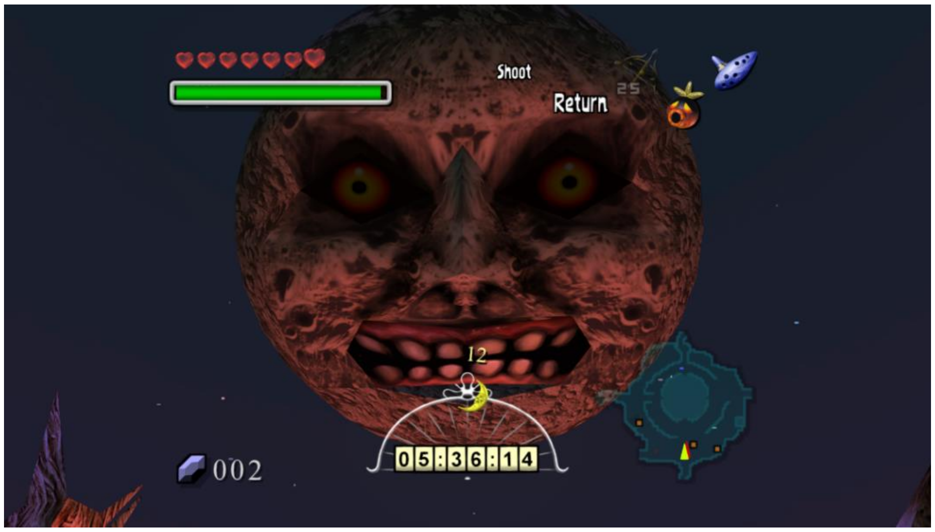


*Figure 4: Screenshot from Zelda: Majora's Mask, the HUD shows the time left. Image from Internet*

In Walpurgis Night, time also has a great influence on the game and can mean the end of the game if midnight comes and the Valkyries have not been saved by player. As well as in the Majora's Mask the player gets a certain power to control passage of time (play a song in the ocarina of the character), in Walpurgis there is a determined object to be found by player. This object can be saved in his inventory for further use. When used clock will turn back one hour.

The Legend of Zelda Wind Waker [25] and The Legend of Zelda Ocarina of Time [26] have been the inspirational idea for Jötunheim as world to be designed, in those games there are a dungeon or an ice level where character's physics is affected in some way depending on the surface the character is on.

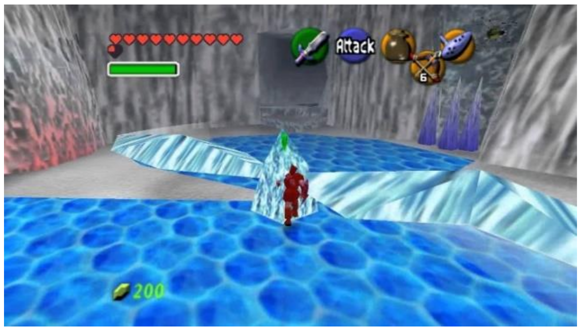


*Figure 5: Screenshot from Zelda: Ocarina of Time inside a frozen dungeon. Image from Internet*

Unity 3D has the possibility of creating 'physics materials' where dynamic friction, static and the combination between them influences and it is possible to apply these physics to objects or characters.

The choice of this level has been greatly influenced by the desire to know the way this physics worked.

### 4.2.2 Artistic Style

A very important aspect in every game is the visual appearance, as it what player can see at first sight, this may influence the player decision to play the game or not.

In addition, the artistic style is another narrative medium that can say a lot about the type of story that is why finding a comfortable and visually pleasing artistic style is so important because it must not interfere with the gameplay.

Jötunheim is defined as a world of stone and ice giants, it is a frozen world where mainly blue and bright tones predominate, with high saturation and marked contrasts, simulating a crystalline appearance. Blue color is friendly to the eyes, but not to cause fatigue in the player, other colors will be incorporated as the color white, mainly because of the location of this world.

Before starting with the artistic section, a previous study of other similar games or some games whose graphic section was very attractive was made.

It took into account the Zelda Wind Waker for its use of shades Toon Shader that contribute a polished and striking appearance with a naked eye, being this one of the aspects that were wanted to reach with the visual section. As is also what is intended to be achieved with the textures presented by this game, so they offer a good look under the use of the Toon Shader [27].

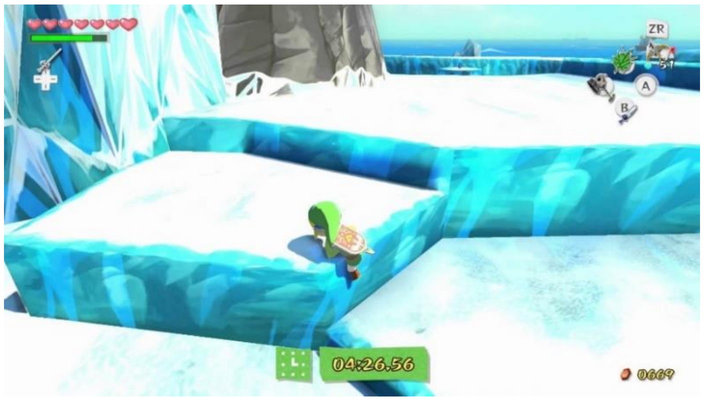


*Figure 6: Screenshot from Zelda: Wind Waker in an ice scenario. Image from Internet*

However, unlike this game, a low poly modeling style was decided to use in order to bring more realism to the scene, moving away from the style of cartoon modeling offered in this Zelda.

The decision to use a low poly style comes from the need to simplify the load of the modelling of scenarios and characters, as well as the charm that this technique offers and the great amount of effects that can be obtained with the light, becoming sometimes a narrative element by itself.

Rime[28], Spanish game launched by Tequila Works in 2017, has also been influential in the use of the contrast between the white present in some elements of the stage and the incident blue light due to the time on which the game is based (between the eight p.m. and midnight). In Rime appears a skybox full of stars with a great full moon that matched with the expectations for Walpurgis' final score. However, it is necessary to consider, as it has previously been with the Zelda Wind Waker, the difference in the artistic style and the type of modelling.

Trailers and images available for Praey for the Gods [29], a game still in process of development, based on a frozen world where the protagonist fights against giants has been influential in the way to model Jötunheim world, especially the way the snow has to fall in this world and the idea of the use of fog and its effects in Walpurgis and its functioning.

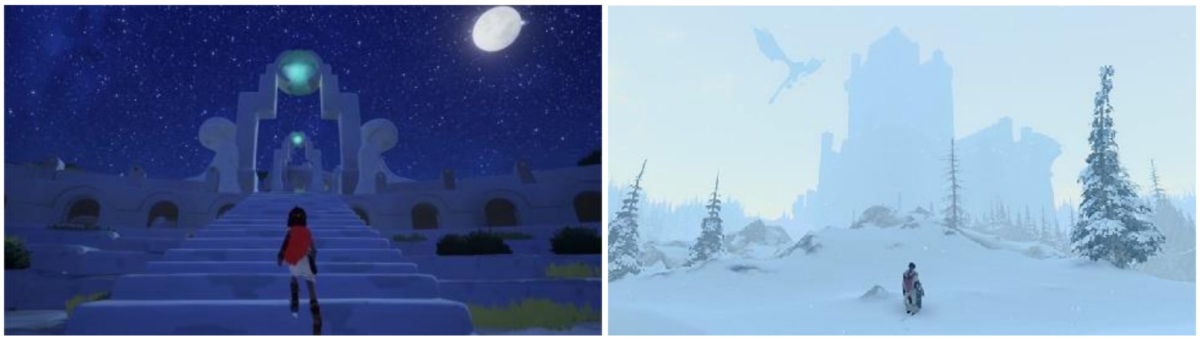


*Figure 7: Screenshot from Rime and Praey for the Gods. Images from Internet*

While the world that can be seen in Pray for the Gods seems to be a world hostile to the protagonist, Jötunheim does not present the sensation of imminent and persistent danger lurking the player.

## 4.2.3 Narrative

In the narrative aspect, Walpurgis presents the classic structure of an RPG adventure game. The player must visit different scenarios where he must perform a certain mission (find his kidnapped companion Valkyrie before midnight) and return with her to the meeting point.

This level is the representation of a basic level or beginner in any RPG, the level that serves the player to gain confidence and ease with the game.

It has been talked previously about the section of mechanics in Golden Sun and its influence on the game, its influence in the section of design and narrative needs to be mentioned too.

In this game, the characters possess and can find along the map some beings called Djinns, elemental spirits who help the characters in their mission. There are four types of Djinns that represent the four elements: water, earth, fire and air, in addition each of them possess a special ability.

Therefore this is a clear and direct influence on Valkyries' powers as zodiacal signs are dived as per different elements, e.g. Taurus, the Valkyrie who needs to be rescued, is an earth sign and as its alchemical symbol[30] 'The modification through freezing' has the special ability to freeze.

## 4.3 Nordic Mythology Relationship's with the Game

The whole game is based on the Nordic mythology [31]; due to this an exhaustive search for information has been made in order to gain reliability.

Ragnarok is defined as the battle of the end of the world, a battle mainly between gods that will end up affecting the whole world and shake the roots of Yggdrasil, the tree of life. Its representation in this game is to be the connection between the different worlds that make up the game. These have a hierarchical organization, meaning that the worlds that are located in their deepest and hidden roots are those with a lower category, Jötunheim is in that category.

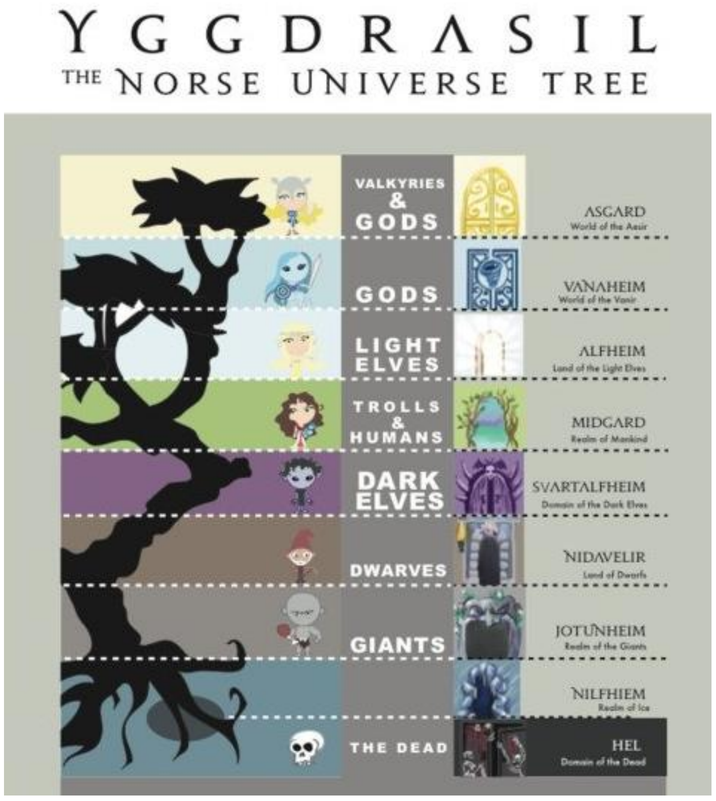


*Figure 8: The representation of Yggdrasil worlds division. Image from Internet*

The consequence of this great battle with which the world will ends will bring the rebirth of a new one, a world where evil and misery will not exist, thus achieving an era of peace and prosperity.

This concept is related to the game where the Ragnarok is summoned by the heroes as a reason to eliminate the gods in order to create a new world under its guidelines. In this new world, there will be no god, so no one will ever find himself subjected to divine whims or yokes.

Walburg of Heidenheim was a religious woman who truly existed and was venerated by her miracles, reaching the title of Saint. Walpurga is considered a pagan goddess.

The name of the game 'Walpurgis Night' comes from the night when this saint is revered. This night between April 30th and May 1st is said to be one in which witches used to celebrate their Sabbaths but the saints eliminate them at dawn.

Walpurgis Night and the game are related due to the parallelism of the meaning of the witch burning and, therefore, the elimination of evil, with the night in which the Valkyries must stop the heroes and the second Ragnarok, representing the eternal struggle between good and evil.

*Figure 9: Representation of Brynhildr. Image from Internet*

The Valkyrie figures are defined as female deities who served the god Odin to choose heroic fallen warriors who would fight alongside him in the Ragnarok. The Valkyries of the game work under the orders of the Walpurga goddess, who represents Odin this time as they are their warriors in charge of avoiding the second uprising of the heroes and stop their plans.

The names of the Valkyries appear in Norse mythology, as for example Brunilda (Brynhildr), this is the name of the most famous Valkyrie and the name of the Valkyrie protagonist of the game.

What's more, all Valkyries are related to a zodiacal sign which in turn represent each of the twelve basic alchemical processes representing the magic each Valkyrie can use. Due to their ability with these powers, the Valkyries are considered as witches by many, thus they became known as the Witches of Walpurga.

# 5. Design

*This chapter summarizes the game design document of the elements that compose the level assigned for this project. For a longest and developed explanation of the level concept and its elements and a deeper look to the whole concept of the project, see 'Anexo II'.*

When the design of a game starts some doubts arise:
What genre will it belong to?
What will make it different from other games?
Is this an original concept or has it been seen previously?

There are many ways to tell a story in a video game and the way in which it is told will affect its development and its gameplay. How can a story be told at the same time that the player enjoy both the game challenges and mechanics? How would it be possible to balance both aspects?

When a person face a mythology based game (a mythology partially known by the gamer), the main objective is to build an entertaining story that will make the player interested in it. At the same time as the mechanics and the gameplay are pleasant and interesting in order to achieve a balanced game that appeals to the player.

This project is based on one of the concepts of Nordic mythology, the world of Jötunheim, its history, its beliefs and the adaptation that have been made of these to develop a playable level with its own identity while being part of a common universe.

Having these concepts clear, the design began to be developed. The GDD is a document of design that synthesizes the aspects that will appear in the game and that encompasses both mechanics, history, the genre of the game ... as well as all the aspects that are necessary to explain and be taken into account.

The GDD of this project must be differentiated into two aspects:
- The common GDD, shared with the other team members who are developing the other two levels related to this universe.
-The GDD own level.
These differences are already indicated in the document itself, providing a specific colour at each level to differentiate who owns each.

While the common GDD explains the more global concepts of the whole project, the individual GDD is the one that has been followed to develop the TFG.

One of the main goals to fulfill was to perform the three scenes within the game and make them playable, although some initial ideas and nuances have been changed as the project was developed, seeing them at the end unnecessary and seeing the delivery date each time closer and closer.

## 5.1 Summary of the GDD

Jötunheim is the world developed for this project, this is a world where giants and golems coexist. These characters have a bad relationship with gods and humans but helped the 'heroes' when the first Ragnarok happened.

Now that the 'heroes' are planning the second Ragnarok, the giants have helped them and have kidnapped one of the Valkyries, Taurus.

It is the goal of the Valkyries to rescue their partner in order to stop the second Ragnarok that would end, once for all, with all the gods and would let the world in the hand of the 'heroes'.

In the game three Valkyries can be used at the beginning and once Tauro has been rescued she is also playable like the others. These Valkyries have each one a special ability that make them different.

| **Valkyrie Name** | **Physical Attack** | **Magical Attack** |
|---|---|---|
| Geminis | Dagger | (she does not have magic) |
| Aries | Axe | Fire / Area fire |
| Pisces | Lance | Ray / Gungnir |
| Taurus | Fists | Earth / Freezing powers |

*Table 3: Character's information table*

The enemies that appear in the game are the next ones:

| **Enemy type** | **Attack Type** |
|---|---|
| Giant Guard | Low physical Attack |
| Stone Golem | Physical Attack |
| Ice Golem | Low physical attack with freezing effect |
| Giant's king | Physical attack with freezing effect |

*Table 4: Enemies' information table*

Along the game, the player will find different objects in the game with a specific behavior each one that it will be helpful in the adventure. It also will deal with ice physics and different enemies that will make this journey a bit more difficult while discover a new word and a complex story.

# 6. Development and Results

*The elements explained in this chapter are the one developed for the project, but it is needed to remember that is also a project with other classmates so they make use of this elements as much as I use the ones they have developed. For more information, visit 'Anexo I'.*

As it was said before, Unity is the engine that has been chosen to build this project. This engine allows the developers to develop many components and details in the game without programming too much, something that it would be difficult to do in others engines.

## 6.1 Time Controller

The time is a really important concept for the game, that's the reason it needs to be implemented and controlled while it's interactive for the player in order for them to use objects that will delay the clock one hour back.

The Game Over is also related to the hour, if the clock mark midnight before Taurus has been rescued (the twelve Valkyries have to be rescued in the whole concept of the game before midnight) the game will end.

The code has two parts, in one part, the hour and minutes are controlled and printed in the HUD, for that a text is used in the canvas.

```
gametime = Time.time;
contadorSec = 60 * 20;
```

Gametime controls the time in seconds since the start of the game and contadorSec save the time that has to pass in order to be 20:00. Then, in the Update function:

```
  if (Time.time - gametime > 7.5)
        {
            gametime = Time.time;
            contadorSec++;
            cont++:
```

```
            magic.TakeMagic(1);
        }

        if (!takeTime)
        {
            hora = (int)(contadorSec / 60f);
            print(hora);
            min = (int)(contadorSec % 60f);
            time.text = hora.ToString("00") + ":" + min.ToString("00");
            Moon()

        }
```

The first 'if' controls if it has pass technically one second (in order to make the game more playable and allow the player to have more time to play), with this it controls if 7.5 seconds has passed in order to upload one second in the clock. Every second the selected Valkyrie will recover one amount of magic.

takeTime is a bool that controls if the object 'Hourglass' has been activated in the inventory, in this case, the clock will go back one hour.

```
  if (takeTime)
        {
            contadorSec -= 60;
            if(contadorSec <= 60 * 20) { contadorSec = 60 * 20; }
            hora = 0;
            time.text = hora.ToString("00") + ":" + min.ToString("00");
            takeTime = false;
            Moon()
        }
```

This script also controls the position of the moon in the sky through the time.

```
   public void Moon()
    {
        moon.transform.rotation = Quaternion.Euler(new Vector3(moonVel * cont/4, 0, 0));

    }
```

## 6.2 Character Selected

Three Valkyries are allowed to be controlled at the beginning of the level with the inclusion of a fourth one when this Valkyrie has been rescued so the option to change the Valkyrie when the player needs it has to be enabled.

A list with the Valkyries the three controllable Valkyries is made, this list is composed by the prefabs of Gemini, Aries and Pisces. GameObject refers to the Valkyrie selected at that moment and with the Index it will be possible scroll through the list. The other two list save the icon of the Valkyrie selected showing its zodiac symbol and the second shows all the Valkyries possible to use in the level.

```
    //Lista de las valquirias
    public List<GameObject> Valquirias;
    GameObject currentValquiria;
    public int charactersIndex;
    int sizeOfList;

    //Lista del personaje seleccionado en ese momento
    public List<Image> CharacterActive;
    Image currentActive;

    //Lista de las valquirais POSIBLES SELECCIONABLES en ese nivel
    public List<Image> ValquiriaEnJuego;
```

With a bool it will be possible to check if a new character has to be added in the list.

```
public bool newcharacter = false;
```

With this function everything is set in order to make a good scroll in the list.

```
 void Start()
    {
        charactersIndex = 0;
        currentValquiria = Valquirias[charactersIndex];
        sizeOfList = Valquirias.Count;
        currentActive = CharacterActive[charactersIndex];

    }
```

In the Update function, it will check if there is a new character to add in the list or if it just need to change the character. The function to change the character is this:

```
    public void changeCharacter()
    {

        if (Input.GetButtonDown("Fire1"))
        {
            currentValquiria.SetActive(false);
            currentActive.gameObject.SetActive(false);
            SavePosition();

            charactersIndex++;

            if (charactersIndex == sizeOfList)
            {
                charactersIndex = 0;
            }

            currentValquiria = Valquirias[charactersIndex];
            currentActive = CharacterActive[charactersIndex];

            currentValquiria.transform.position = new Vector3(PlayerPrefs.GetFloat("Px"),
            PlayerPrefs.GetFloat("Py"), PlayerPrefs.GetFloat("Pz"));
            currentValquiria.transform.eulerAngles = new
            Vector3(PlayerPrefs.GetFloat("Ox"), PlayerPrefs.GetFloat("Oy"),
            PlayerPrefs.GetFloat("Oz"));

            currentValquiria.SetActive(true);
            currentActive.gameObject.SetActive(true);
        }
    }

    void SavePosition()
    {
        PlayerPrefs.SetFloat("Px", currentValquiria.transform.position.x);
        PlayerPrefs.SetFloat("Py", currentValquiria.transform.position.y);
        PlayerPrefs.SetFloat("Pz", currentValquiria.transform.position.z);

        PlayerPrefs.SetFloat("Ox", currentValquiria.transform.eulerAngles.x);
        PlayerPrefs.SetFloat("Oy", currentValquiria.transform.eulerAngles.y);
        PlayerPrefs.SetFloat("Oz", currentValquiria.transform.eulerAngles.z);
  }
```

It is a simple process, if the input that is defined to do this function is selected, the actual Valkyrie will stop being Active and its position will be saved. This functions with PlayerPrefs [32] that will store the position and the rotation of the Valkyrie in the moment the change was requested. This is needed because the next character needs to have the same position and rotation that the last one.

One these is saved, the index will increase in one position, if the new position is equal to the size of the list, and this index will be equal to the initial position. The new Valkyrie that has been requested will be the one in the position of the index and its zodiac symbol will be activated. The position and rotation will be loaded now thanks to the Save Position function used before and it will be now activated to it appears on the screen.

The way to add a new character is this, in the same code this function will appear:

```
    public void AddCharacter(GameObject newvalquiria)
    {
        Valquirias.Add(newvalquiria);
        newcharacter = true;
    }
```

So now that a new character is true, the size of the list will be updated to the new size. However, this is not the only thing that is needed. In the game there has to be a Game Object that represents the new Valkyrie with a Collider on Trigger on this function:

```
    public void OnTriggerStay(Collider other)
    {
        if(other.tag == "Player")
        {

            if (theValquiria != null && !añadir.Valquirias.Contains(theValquiria)) {
                añadir.Valquirias.Add(theValquiria);
                añadir.CharacterActive.Add(ImagenCorrespondiente);

                añadir.ValquiriaActivada.Add(ValquiriaenJuego);
                ValquiriaenJuego.gameObject.SetActive(true);
                añadir.newcharacter = true;
                Destroy(gameObject);

            }
        }
    }
```

So if the controllable Valkyrie collides with this Game Object, and the Game Object 'theValquiria' (a public game object that contains the prefab of the new Valkyrie to be added) is not null and hasn't been added before, this new Valkyrie and its zodiac symbol will be added to the previous list and will destroy the this false Game Object that just works as an imitator.

## 6.3 Interaction

Interactions are needed so the character can differentiate between NPC and object and act in consequence.

In order to achieve this interaction, it exist a distance that allow the character to 'see' the element with has to interact.

For this, a RaycastHit [] is used and a float with the 'distanceToSee' the elements.

```
    public float distanceToSee;
    RaycastHit whatHit;

    void Update()
    {
       if(Physics.Raycast(transform.position, this.transform.forward, out whatHit,
       distanceToSee))
        {
            Debug.Log("TOCO ALGO");
            if(whatHit.collider.gameObject.tag == "NPC")
            {
                Debug.Log("Soy un NPC");
                if (Input.GetKey(KeyCode.L))
                {
                    //Do things
                }

            }
            if (whatHit.collider.gameObject.tag == "item")
            {
                //Do things

            }
        }
    }
```

What this code does it cast a Raycast from the character and check the tag of the gameObject that is colliding, if the tag is NPC then the character will interact with it talking (this function has been developed by other member of the group), if the tag is item, it will interact with the object and add it to the inventory.

This code has experienced changes over the development of the common project due to the changes in the final approach that the interaction with the items were going to have. At the end, there is a Vector 3 Distance that calculates the distance between the Valkyrie and the object to interact.

## 6.4 Inventory

An inventory is totally necessary when the characters has to stock objects that will be useful in the future, the common objects are a health and magic potion, the hourglass, and the shield and reviver orb.

The development of the inventory has changed from the first implementation until the final version. At first, it was the pickup script the one that controlled the tag from the object that collide with the Valkyrie.

```
void OnCollisionEnter(Collision collision)
    {
        GameObject i;

        if(contador < 6)
        {
            if (collision.gameObject.tag == "1")
            {
                Debug.Log("1");
                i = Instantiate(icons[0]);
                i.transform.SetParent(inventoryPanel.transform);
                contador++;
            }
```

In order to improve the performance of the interactions, this had to be changed when the interaction was controlled with a vector 3 distance.

Now each object has a Destroy script that controls its tag and a functions returns a number depending of what tag is that (except if the tag is the one that belongs to the resurrection orb).

This function is called from the changed Pickup script that each Valkyrie has because the function devolve an int that it refers to the position of the prefab that it is on an array and it is an array of GameObjects with a button and an icon that refers to each object and their action.

This function controls the tag from each object and returns a number:

```
int SendInventary(){

    int iconSend = 0;
          tagItem = GetComponentInChildren<BoxCollider>().gameObject;
          print ("Tag: " + tagItem.tag);

          if(tagItem.tag.Equals("1"))
    {
        iconSend = 0;
    }
          else{
                 if (tagItem.tag.Equals("2")) {
                        iconSend = 1;
                 } else {

                        if (tagItem.tag.Equals("3")) {
                               iconSend = 2;
                        } else {
                               if (tagItem.tag.Equals("5")) {
                                      iconSend = 4;
                                      } else {

                                             if (tagItem.tag.Equals("6"))
                                             {
                                                    iconSend = 5;
                                             }
                                      }
                               }
                        }

                 }
                 print (iconSend);
                 return iconSend;
          }
```

Then, in Update, this number is sent to another function:

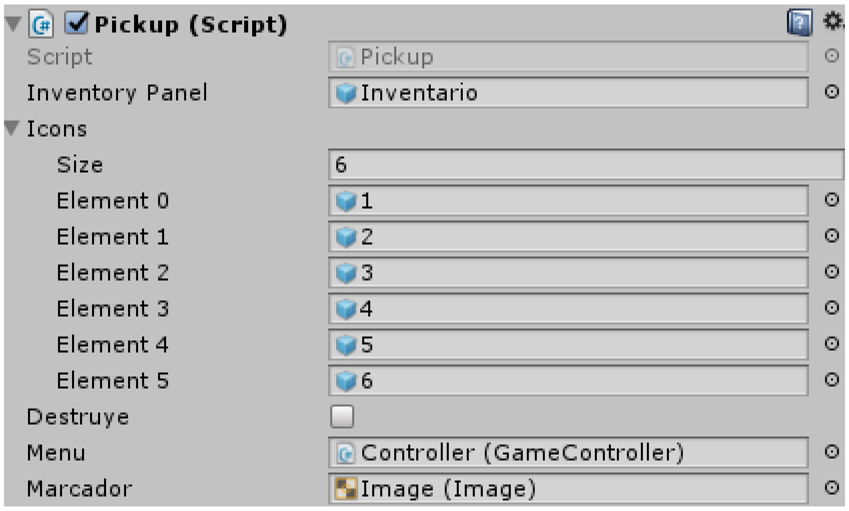
```
            if (gameObject.tag != "4")
            {
                int send = SendInventary();
                pick.RecieveItem(send);
            }
```

This function receives that number and it will be the position in the array of each object.

```
public void RecieveItem(int icon)
{
    GameObject i;
```

```
        if(inventory.contador < 6)
        {
            i = Instantiate(icons[icon]);
            i.transform.SetParent(inventoryPanel.transform);
            inventory.objects.Add (i);

            inventory.contador++;
        }
    }
```



Pickup (Script)

| | |
|---|---|
| Script | Pickup |
| Inventory Panel | Inventario |
| Icons | |
| Size | 6 |
| Element 0 | 1 |
| Element 1 | 2 |
| Element 2 | 3 |
| Element 3 | 4 |
| Element 4 | 5 |
| Element 5 | 6 |
| Destruye | ☐ |
| Menu | Controller (GameController) |
| Marcador | Image (Image) |

*Figure 10: Some of the elements that compose the pickup script*

The inventory is set in a way that more objects can be added to the game, the only that has to be made it to increase the array to make place to those new objects and in on the Destroy script that belongs to every object, control their tag.

### 6.4.1 Available Objects

There are different objects available in the game with different effects each one. As it was explained before, the inventory can be expanded depending of what is needed for each level, e.g., in Jötunheim the basic objects can be found along the key for the enemy's dungeon.

The actions performed by the basic objects are:

- **Health potion**: when the player uses this potion, the TakeHealth function is called, this function is in the HealthMagic script from the Valkyries and add an amount of health in the current health.

  Code in the HealthMagic script;

```
public void TakeHealth(float amount)
```

```
{
    if(CurrentHealth <= initialLife)
    {
        CurrentHealth += amount;
        SetHealthUI();
        if(CurrentHealth > initialLife)
        {
            CurrentHealth = initialLife;
        }
    }
 }
```

Code in the ItemController to perform the action:

```
public void itemHealt()
{
    increase.TakeHealth(15);
    inventory.contador--;
    Destroy(gameObject);
}
```

- **Magic potion**: this potion works in a similar way that Health potion, but this time calls the TakeMagic function instead.

- **Hourglass**: this object interact with the Time Controller script, it has to turn back one hour in the clock to give the player more time.

  In the Item Controller function, this object performs this action:

```
public void itemTime()
{
 time.TakeTime();
 inventory.contador--;
 Destroy(gameObject);

 }
```

  The itemTime function calls this one that it was already explained before. This bool changed to true causes changes on the Update function.

```
public void TakeTime()
{
 takeTime = true;

 }
```

- **Shield orb**: With this object the character has access to a Shield that provides protection to the Valkyrie with a limit use time. This is a hemispheric geosphere prefab made on

3DS max with a transparent material that contains the 'myshield' script, this function is also called like this:

```
public void itemShield()
{
  myshield.shieldActivated();
   inventory.contador--;
   Destroy(gameObject);
 }
```

```
The shieldActivated function works this way:
```

```
private void Start()
{
    box = gameObject.GetComponent<BoxCollider>();
    box.isTrigger = true;
    mesh = gameObject.GetComponent<MeshRenderer>();
    mesh.enabled = false;
}
```

When it starts, the box collider is trigger so it can cross through objects and the mesh renderer does not render the object, so it does not appear on the scene.

```
public void shieldActivated()
{
    isActive();
    StartCoroutine(noActive());
}

public void isActive()
{

    mesh.enabled = true;
    box.isTrigger = false;

}

public IEnumerator noActive()
{
    yield return new WaitForSeconds(6f);
    box.isTrigger = true;
    mesh.enabled = false;

}
```

The shieldActivated functions calls the isActivate function. In this function the mesh is rendered and the collider is no longer trigger while is still activated. The IEnumerator wait until the coroutine has finished with the given time and the collider and the mesh returns to their initial state. The shield provides the character an extra protection in case of need.

The images below belongs to an early phase of the development, but it allows to see the implementation of the shield.

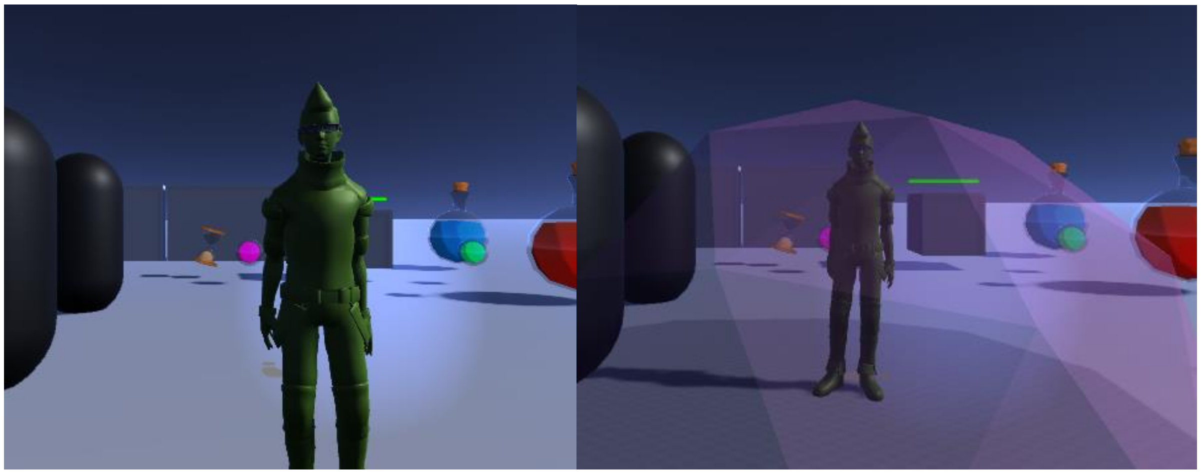


*Figure 11: Representation of the shield element (in an early implementation) once it's activated*

- **Magic orb**: when this orb is activated, the player can use magic during a limit time without spending manna and this allow the Valkyries to use their more powerful spells without expenses.

  The Item Controller calls the HealthMagic script with this:

```
public void itemFreeMagic()
{
    increase.ZeroWaste();
   inventory.contador--;
    Destroy(gameObject);
 }
```

  ZeroWaste activates a bool in the function UseMagic that if it is false, the magic will descend when it is used, but if the bool is true, the player will use this magic without cost until the IEnumerator that controls the time passed ends and change the bool to false.

```
IEnumerator WasteMagic()
{
    yield return new WaitForSeconds(10f);
    isMagic = false;
}
```

- **Resurrection orb**: The original concept for this object was that the object is used automatically if the Valkyrie dies and it is in the inventory, if the player tries to use it while is still alive it will be deleted. Due to some problems with the implementation when it is used because it has to be automatically and be deleted but after this was made the inventory did not update correctly, the concept was changed.

The new implementation controls if the player has interacted with this object, in this case, increases a counter set on the HealthMagic script that controls how many resurrection orbs has the player. When the player has one or more, an image is set active in the canvas with a text that shows how many orbs of this type can be used. The action of the object is still the same, if the Valkyrie dies but there is one or more orbs, then the life is full again.

- **Frozen key**: this key appears after the player fight and win against the dungeon guard. The action of the key is to open the doors that are blocking the entrance of that dungeon.

  Its action is to call the animations that allow the portals to rotate 45 grades in the Y position.

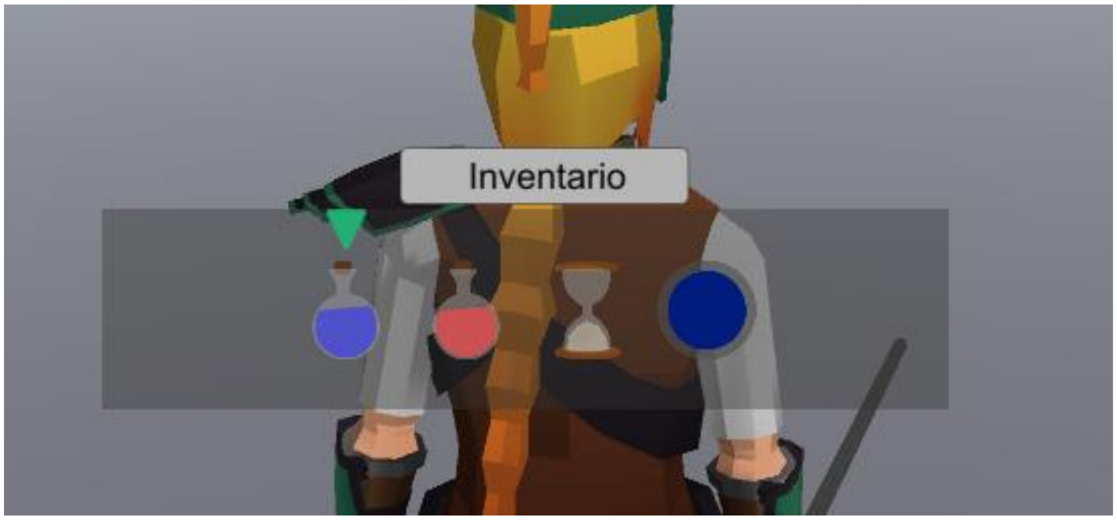
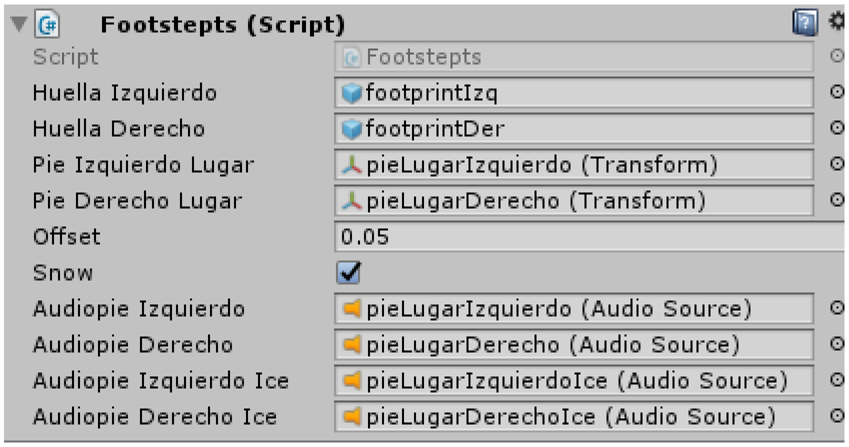


*Figure 12: Screenshot from the game and a visual sample of the Inventory*

## 6.5 Footsteps

Some of the surface on the game is snow and when someone walks over the snow leaves traces. So in order to make a realistic situation, this was implemented, but before that, there was the need to remember that snow was not going to be the only type of surface on the game, there's also ice, and on ice things change. This script is also used for infiltration motive because the enemies react to the sound.



| | |
|---|---|
| Footstepts (Script) | |
| Script | Footstepts |
| Huella Izquierdo | footprintIzq |
| Huella Derecho | footprintDer |
| Pie Izquierdo Lugar | pieLugarIzquierdo (Transform) |
| Pie Derecho Lugar | pieLugarDerecho (Transform) |
| Offset | 0.05 |
| Snow | ✓ |
| Audiopie Izquierdo | pieLugarIzquierdo (Audio Source) |
| Audiopie Derecho | pieLugarDerecho (Audio Source) |
| Audiopie Izquierdo Ice | pieLugarIzquierdoIce (Audio Source) |
| Audiopie Derecho Ice | pieLugarDerechoIce (Audio Source) |

*Figure 13: Some of the elements that compose the Footstep's code*

Two functions for each foot exist, that's because it's controlled when one foot hits with the ground, and this is made with a Physics Raycast.

This fragment of code shows what it has to be made with the right foot.

```
public void PieDerecho()
{
    print("pie derecho");

    if (snow)
    {
        RaycastHit hit;

        if (Physics.Raycast(pieDerechoLugar.position, pieDerechoLugar.forward, out
        hit))
        {
            AudiopieDerecho.Play();
            GameObject print = Instantiate(huellaDerecho, hit.point + hit.normal *
          offset, Quaternion.LookRotation(hit.normal, pieDerechoLugar.up));
            Destroy(print, 2f);
        }
    }
}
```

The moment the Raycast is casted and hit with the ground, the sound is played and the printed image is rendered on the surface. This image will disappear after five seconds.

A Trigger Box Collider controls when the surface is snow or ice:

```
private void OnTriggerEnter(Collider other)
{
    if(other.tag == "Ice")
    {
        steps.snow = false;

    }
    if (other.tag == "floor")
```

```
        {
            steps.snow = true;
        }
    }
```

The health for the Valkyries and the Enemies have been developed thanks to the knowledge gained with the tutorials that were done during the practices in the subject VJ1227.

## 6.6 Health and Magic from the Valkyries

The health and the magic are two means that have to be in the HUD in order for the player to control those attributes.

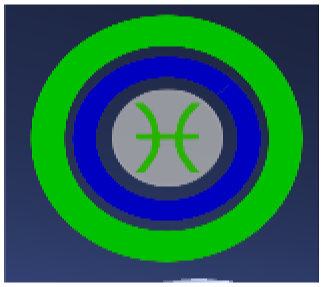


*Figure 14: Health and magic representation in the canvas*

The green circle represents the health and the blue one the magic, both are a slider with a circular image that is filled with colour on the code. There is a difference between both components, while the health is common for all the Valkyries because if one die there is no way they could stop the Ragnarok, each one have a different slider and a different amount of magic with.

The character lose health every time an enemy attacks or if hits with some elements in the scene, this function the same way for magic, but in this case it lose magic every time the Valkyrie cast a spell and depending of what type a spell is (basic or strong magic) the amount of magic lost is different.

```
    public void TakeDamage(float amount)
    {
            CurrentHealth -= amount;
            SetHealthUI();

            if (CurrentHealth <= 0f)
            {
                    Dead = true;
                    OnDeath();
            }
    }

    public void UseMagic(float amount)
    {
            if (!isMagic) {
                    if(CurrentMagic >= 0 || CurrentMagic - amount >= 0)
                    {
                            CurrentMagic -= amount;
                            valquiria.currentMana = CurrentMagic;
                            SetMagicUI();
                    }
            }
            else
            {
                    StartCoroutine(WasteMagic());
            }
    }
```

If the Valkyrie has not health left the Death function will be called and the game object will be deactivated, causing the game to end. In the case of the magic, it will not let to cast another spell until the Valkyrie has enough manna for that.

```
    private void OnDeath()
    {
            if (contaResucitar != 0 && Dead) {

                    CurrentHealth = initialLife;
                    SetHealthUI ();
                    contaResucitar--;
                    text.text = "x " + contaResucitar;
                    Dead = false;

          if (contaResucitar == 0)
          {
              image.gameObject.SetActive(false);
          }
          } else {
                    GameOver = true;
                    usado = false;
     }
 }
```

This function is also called to recover health and magic, the Valkyrie will recover health when the specific object stock in the inventory is used and the same goes for the magic, except that the magic also increase each second one amount.

This script is shared by all the Valkyries in a parent object because those are actions that all of them are going to realize and the health has to be common for all of them.

## 6.7 Enemy's Health

The same way that the Valkyries have a health component, the enemies needs to have one too, for that, each enemy has a slider over them that represents their life. Thanks to this, the player can control when they are going to defeat them.

This code works like the Valkyries' health one, except that the enemies do not have a way to recover life and ones have been defeated, it is eliminated.

The only one that does not fulfill this is the ice golem and the king, because if it is attacked with ice magic it will recover life, instead of lose it.

## 6.8 Valkyries' physical attack

### 6.8.1 Aries

Aries has an axe for the physical attack so the axe object has to be a children of the arm to follow the movement of the animation.

This object also needs a collider to perceive when it hit the enemy and call the TakeDamage function from the Enemy health.

### 6.8.2 Taurus

Taurus is a physical attacker so she attacks without weapons, only with her fists. Because of this, a specific animation was filmed for this character.

For develop this attack, the character needs to have two trigger colliders in each fist, so this way the attack will count when the fist touch the collider of the enemy.

When the input that start the physical attack is called, a function calls the Tauro Attack script and starts the animation. As each fist has a collider, it has to be controller when the enemy enter that trigger collider and if the attack has started, if both cases are true, the enemy will lose life and the animation will finish.

```
void OnTriggerEnter(Collider col)
{
    if (col.tag == "Enemy" && attack == true)
    {
        col.GetComponent<EnemyHealth>().TakeDamage(amount);
        attack = false;

    }
}
```

## 6.9 Valkyries' magic attack

While the simple magic is the same for all the Valkyries, just changing the particle system based on what element in the correct for them, each one has a different powerful magic so in order to accomplish this and for the program to know what character is the one requesting the attack and for that their character index has to be checked.

```
StrongMagic(controller.charactersIndex);

void StrongMagic(int index)
{
       if (index == 0)
```

```
        {
               print("Piscis");
               Piscis();
        }
        if (index == 1)
        {
               print("Aries");
               Aries();
        }

        if (index == 3)
        {
               print("Virgo");
               Tauro();
        }
}
```

Request the index of the current Valkyrie (the one requesting the order) and depending on its index it will be one Valkyrie or other so it will activate one type of magic.

From the six Valkyries available in the game, the ones that has been implemented by me are Aries and Taurus

### 6.9.1 Aries

Aries element of magic is fire and has two types of magic attack, the simple and the strong magic.

The simple magic instance a particle system on a spawn point located on her left hand every time the input R1 is pulsed.

The strong magic attack on an area with a particle system. When the strong magic attack is selected, set as active the game object Aries Magic and calls for the turn on object function.

```
void Aries()
{
       AriesMagic.gameObject.SetActive(true);
       AriesMagic.GetComponent<AriesMagic> ().TurnOnObject ();

}

public void TurnOnObject(){

       enemy =
       GameObject.FindGameObjectWithTag("Enemy").GetComponent<EnemyHealth>();
       StartCoroutine(TurnOffObject());
       fixedEnemy = GameObject.FindGameObjectWithTag
       ("Fijar").GetComponent<colliderPJ> ();

       time = 1f;
       damage = 15;
       one = false;

       if (fixedEnemy.enemigoFijado != null) {
```

```
                        gameObject.transform.position =
                        fixedEnemy.enemigoFijado.transform.position;
                        GameObject attack = Instantiate (Ariesmagic,
                        fixedEnemy.enemigoFijado.transform.position,
                        fixedEnemy.enemigoFijado.transform.rotation);


                } else {
                        gameObject.transform.position =
                        gameObject.GetComponentInParent<Rigidbody> ().transform.position;
                        GameObject magic = Instantiate (Ariesmagic,
                        gameObject.transform.position, gameObject.transform.rotation);
                }
                StartCoroutine (Fire ());

        }
```

This function find the health script from the enemy and starts a coroutine that will set off the game object Aries Magic. What it does this function is distinguish between two options, if there is an enemy fixed (this function has be developed by other classmate from the group), the strong magic will be instantiate in that position, if not, it will be in the Valkyrie position and will have effect around her zone. Both options call another Coroutine that will destroy the object instantiate before after a few seconds

The way to make damage with this strong magic is that when the game object is activated, the box collider that belongs to this object also is, so if the enemy is inside this trigger collider a function will be called that belongs to the health of the enemy and it will make it damage.

### 6.9.2 Taurus

The element of Taurus is earth, but the strong magic of this Valkyrie is the power of freeze. What this power does is to stop the enemies for a few seconds and make them damage.

This Valkyrie as Aries can freeze the enemy and attack directly or attack by area, the way this work is like Aries script but while Aries just instantiate a particle system, Taurus does this:

```
        IEnumerator Fire()
        {
                yield return new WaitForSeconds(4.5f);
                AI.enabled = true;
                GameObject particles =
GameObject.FindGameObjectWithTag("AriesStrongMagic");
                Destroy(particles);
        }

        private void OnTriggerEnter(Collider other)
        {
                if (other.tag == "Enemy" && other.gameObject != null)
                {
                        AI =  other.GetComponent<PandaBehaviour>();
                        AI.enabled = false;
                        StartCoroutine(MakeDamage());
                }
}
```

As it was explained before, Tauro strong magic freeze the enemies but what really does is to put as disable the AI from the enemy, forcing them to stop until this effect has passed.

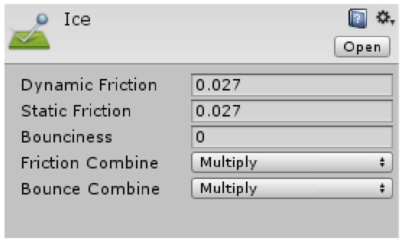
## 6.10 Enemy's Attack

Given a behaviour tree developed by one of the members of the team, the enemy attack is created by modifying that script and adding the animations of the enemy, their characteristics and the colliders to perceive when the Valkyrie has been successfully attacked.

One of the objective was to increase the difficulty of the game with the time, this is achieved by increasing the enemies' attack with the time, so when at eight an attack could make twenty of damage, and at nine the same attack can do forty.

## 6.11 Ice physics

Unity 3D max has a component called Physic Material that simulate the reaction that the bodies have over different surfaces changing the friction and the bouncing.



| | |
|---|---|
| Dynamic Friction | 0.027 |
| Static Friction | 0.027 |
| Bounciness | 0 |
| Friction Combine | Multiply |
| Bounce Combine | Multiply |

*Figure 15: The Physic Material inspector from Unity 3D*

As the objective is to simulate the interaction with ice, both dynamic and static friction should be close to zero to obtain the desired effect. However during the implementation of the physic material a problem was found. While this worked well for the basic 3D objects that Unity allows to create, it didn't work that well for humanoid figures.

Furthermore, while the plane was horizontal (without any inclination in the axes) and the player was moving, there wasn't any resemblance to the natural reaction on the ice.

After this setback, a solution was needed to be found.

Because the objective of this implementation was to find a realistic character behaviour on an ice platform, real physics were implemented. In addition, the problem was separated into two possible situations:

- Horizontal plane
- Plane with rotation

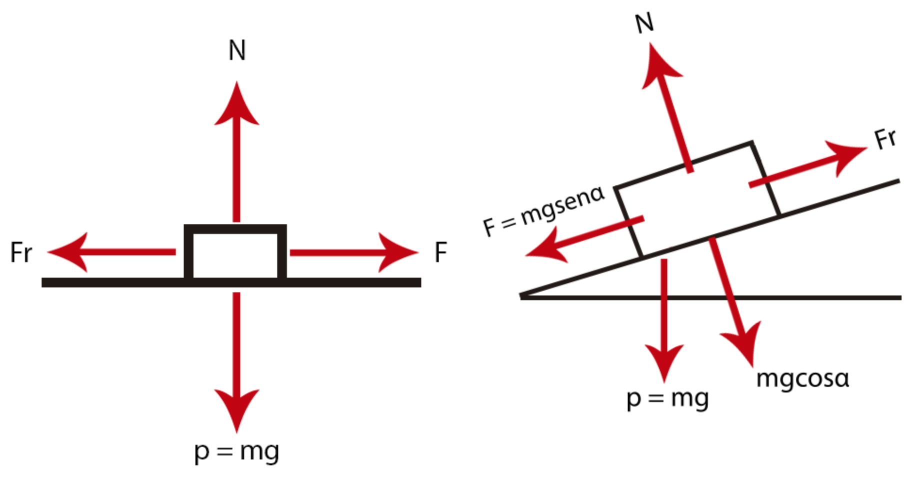


*Figure 16: Forces in a plane*

When a body is in a horizontal surface, in order to move that body, a parallel force to the surface needs to be applied. Given that the surface the character is going to walk is ice, this force has to simulate a small glide.

The normal and the weight are neutralized and the only forces left are the friction force and the force that make the object to move. The friction force goes slowing the movement of the character (so when the character is on an Ice surface the velocity of the Valkyrie will descend) and the other force will be applied to the character axes position so this force will simulate a small glide.

```
private void OnTriggerEnter(Collider other)
{
    if(other.tag == "Ice")
    {
        Forces(other.gameObject);
        velocity = GetComponent<ThirdPersonCharacter>().m_MovingTurnSpeed = 100;
        velocity = GetComponent<ThirdPersonCharacter>().m_StationaryTurnSpeed = 100;
        slide = true;
    }
}

private void OnTriggerExit(Collider other)
{
    if(other.tag == "Ice")
    {
        velocity = GetComponent<ThirdPersonCharacter>().m_MovingTurnSpeed = 360;
        velocity = GetComponent<ThirdPersonCharacter>().m_StationaryTurnSpeed = 180;
        slide = false;
    }
}

void Forces(GameObject plane){

   […]

    Vector3 forward = rb.transform.forward;
    forward.y = 0;

    if(plane.transform.rotation.x == 0 & plane.transform.rotation.z == 0)
    {
          rb.AddForceAtPosition(new Vector3(forward.x  * 20, 0, forward.z * 20),
          rb.transform.position);
    }

 }
```

If the Valkyrie is in an inclined plane, the applied forces works different.

Before start explaining this part, there was a setback, the formulas are designed for two dimensions so here they have had to be adapted to include the Z dimension, therefore, the results obtained may not be true, but adapted to what was intended to be achieved for the game.

In an inclined plane, the function take into account the rotation in X and Z and depending the direction of that rotation (bigger or lower than 0) the forces are applied in a direction or another. Here an example if the rotation in X is bigger than zero and smaller in Z.

```
        if (plane.transform.rotation.x > 0 && plane.transform.rotation.z < 0)
        {
            rb.AddForce(totalForce, 0, totalForce, ForceMode.Force);
            rb.AddRelativeForce(az, 0, ax, ForceMode.Acceleration);
        }
```

The variables totalForce, az and ax are the result of applying the equations of the fall of a body in an inclined plane.

```
        normalForce = rb.mass * 10;
        peso = rb.mass * 10;

        sinx = (float) Math.Sin(plane.transform.rotation.x);
        cos = (float)Math.Cos(plane.transform.rotation.y);
        sinz = (float)Math.Sin(plane.transform.rotation.z);

        Px = peso * sinx;
        Py = peso * cos;
        Pz = peso * sinz;
        Fr = plane.GetComponent<Collider>().material.dynamicFriction * normalForce;
//Fuerza de fricción

        forcex = Px - Fr;
        forcez = Pz - Fr;

        if (forcex < 0)
        {
            forcex = forcex * (-1) * 5;
        }
        if(forcez < 0)
        {

            forcez = forcez * (-1) * 5;
        }

        totalForce = forcex + forcez;

        ax = forcex / rb.mass;
        az = forcez / rb.mass;
```

This are the calculated forces. First the weight of the body is equal to the mass multiplied for the gravity. The sin and cos are obtained from the rotation of the plane that the character is at that moment and here comes the problem, it is in the formula the calculation of the sine in X and the one of the cosine in Y (this is obtained with trigonometry), so, because Z occupies the position of X in its side with respect to the Y, the sine in Z will also be applied. The friction force is obtained by the friction from the material and the normal force.

Once these is all calculated, the forces are obtained by the Newton's theorem (the sum of all forces is mass multiplied by acceleration), that is the reason forcex is obtained by Px less Fr and the same way with the force in z, because P-F = m*a.

If the force obtained is lower than zero, then that force will be multiplied by -1 to make it positive and by 5 to make it bigger and be more noticeable on the game. Now that the force has been obtained, the acceleration can be calculated and applied to the body.

Finally, the two forces have been added because they may not have the same magnitudes, therefore, once applied, the higher magnitude will predominate over the other and will only perform the movement on direction.

## 6.12 Particle System

### 6.12.1 Snowflakes

In a world based on snow and ice, there has to snow and Unity 3D has the possibility to create Particle Systems [34] that helps to create this simulation.

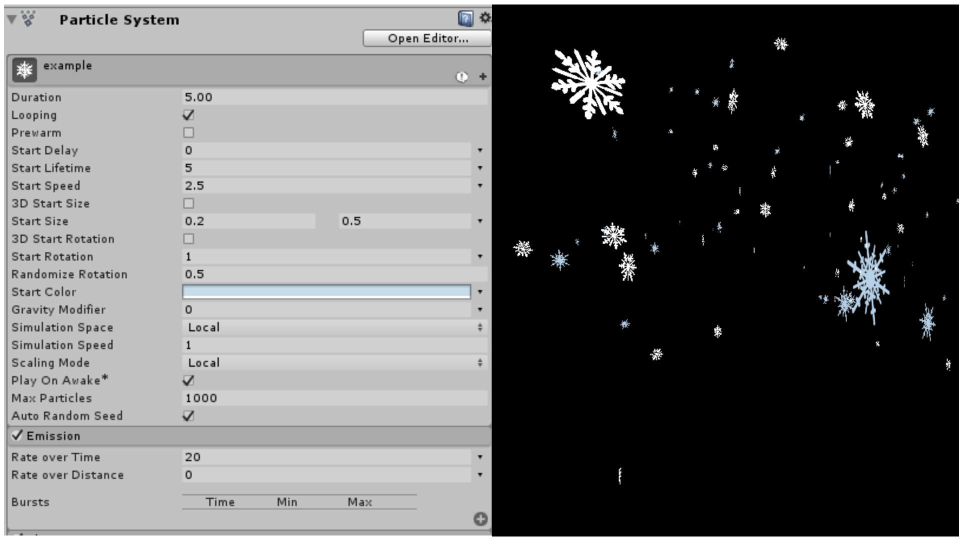
First, an image of a snowflake was created in Photoshop, it was saved as PNG and imported to Unity. Then, a new material was created and with the option Particle/Additive, the image now goes to the texture square that is available.

Figure 17: Example of a Snowflake made with Photoshop

An empty Game Object is created in the Hierarchy and the Particle System is added as a component. This component has a lot of options and possibilities for the particles, and for simulate the effect of snowing the rotation on X was changed so it pointed to the ground.

Elements as the shape was changed to a box to cover more area and make it more realistic, it has also collisions so when it hit the ground it will disappear in a short time.

The texture has different sizes when it starts and there are two types of snowflakes texture so it doesn't look monotonous.



| | | |
|---|---|---|
| Duration | 5.00 | |
| Looping | ✓ | |
| Prewarm | ☐ | |
| Start Delay | 0 | |
| Start Lifetime | 5 | |
| Start Speed | 2.5 | |
| 3D Start Size | ☐ | |
| Start Size | 0.2 | 0.5 |
| 3D Start Rotation | ☐ | |
| Start Rotation | 1 | |
| Randomize Rotation | 0.5 | |
| Start Color | | |
| Gravity Modifier | 0 | |
| Simulation Space | Local | |
| Simulation Speed | 1 | |
| Scaling Mode | Local | |
| Play On Awake* | ✓ | |
| Max Particles | 1000 | |
| Auto Random Seed | ✓ | |

✓ Emission

| | | | |
|---|---|---|---|
| Rate over Time | 20 | | |
| Rate over Distance | 0 | | |
| Bursts | Time | Min | Max |

Figure 18: Example of the Particle System and the result

### 6.12.2 Mist/Fog

One of the effects that the cold creates is the appearance of mist, for this reason, it has been implemented another type of particle system to simulate this effect.

The material used for this is the particle smoke mobile material from the Standard Assets from Unity, and it has been imported from the Import Package.

To get this effect, the rotation has changed rotating 90 on Y and -90 on Z and the shape can be changed into box or sphere to get more types of mist.

*Figure 19: Example of the Mist with box and sphere shape results*

## 6.13 Save and Load Position

### 6.13.1 Save the position

In the game, there is a zone where the Valkyrie can fall, this part is a cliff, if the player falls there is no possibility to go back at least the game has been saved before and the player reload the game.

For this reason, a safe zone has been implemented. This work with two box colliders, one covers the zone that it is near the cliff and there the code save the position of the player.

The collider that save the position does this:

```
GameObject[] player;
Interface index;
```

```
    public GameObject position;

       void Start () {

        player = GameObject.FindGameObjectsWithTag("Player");
        index = GameObject.FindGameObjectWithTag("Controller").GetComponent<Interface>();
       }

    private void OnTriggerEnter(Collider other)
    {
        if(other.tag == "Player")
        {
            Save();
        }
    }

    public void Save()
    {
        PlayerPrefs.SetFloat("x",
index.Valquirias[index.charactersIndex].transform.position.x);
        PlayerPrefs.SetFloat("y",
index.Valquirias[index.charactersIndex].transform.position.y);
        PlayerPrefs.SetFloat("z",
index.Valquirias[index.charactersIndex].transform.position.z);
    }

    public void Load()
    {
        index.Valquirias[index.charactersIndex].transform.position = new
Vector3(PlayerPrefs.GetFloat("x"), PlayerPrefs.GetFloat("y"), PlayerPrefs.GetFloat("z"));

    }
```

The problem that it was here is that there are three characters that can be used in the game, so it has to be present the need to save the position of the current character on the scene, in order to do this, the code has to have access to the current character index in the list that stores the Valkyries that has been implemented in 'Character Selected'.

### 6.13.2 Load the position

Once the position has been saved, when the Valkyrie falls over the cliff and the box collider that cover this zone detect it the position saved it is loaded. After this, the Valkyrie returns to the zone where the position was saved.

```
    private void OnTriggerEnter(Collider other)
    {
        if (other.tag == "Player")
        {
            zone.Load();
        }
    }
```

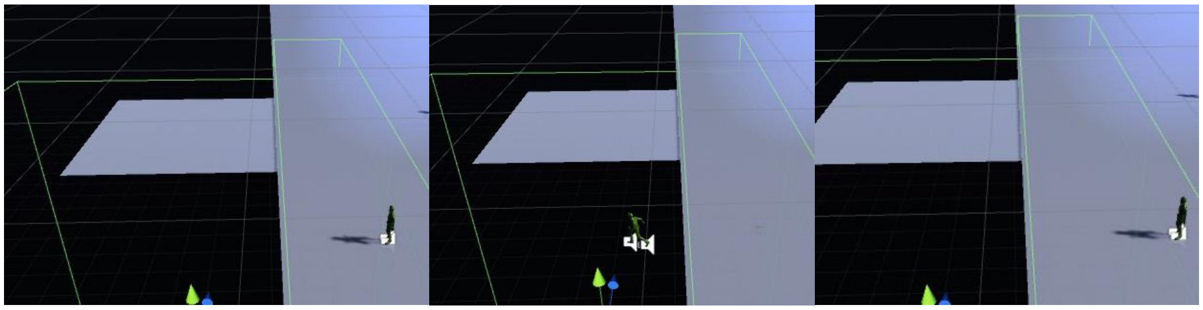


*Figure 20: Example of the implementation, the character falling and returning to the save point.*

## 6.14 Post Processing

Since the Unity 5.5 upgrade, it is possible to download the complete version of this asset [35]. This asset is a combination of multiple image effects into a single post-processing pipeline.

With this asset is possible to obtain good results with the effects of Fog and Color Grading which change the final result of the game into a one more polished.

With fog is possible to enhance the perception by shading distant objects, in fact, Unity allows to set the colour of the fog and the distance where the fog will start and will end. This he fog was one of the effects sought for the project since it is an effect that occurs in environments with snow and ice and brings details to the scene.

With Color Grading the colour of the image can be changed, creating with that a colder aspect with some fantastic tints.

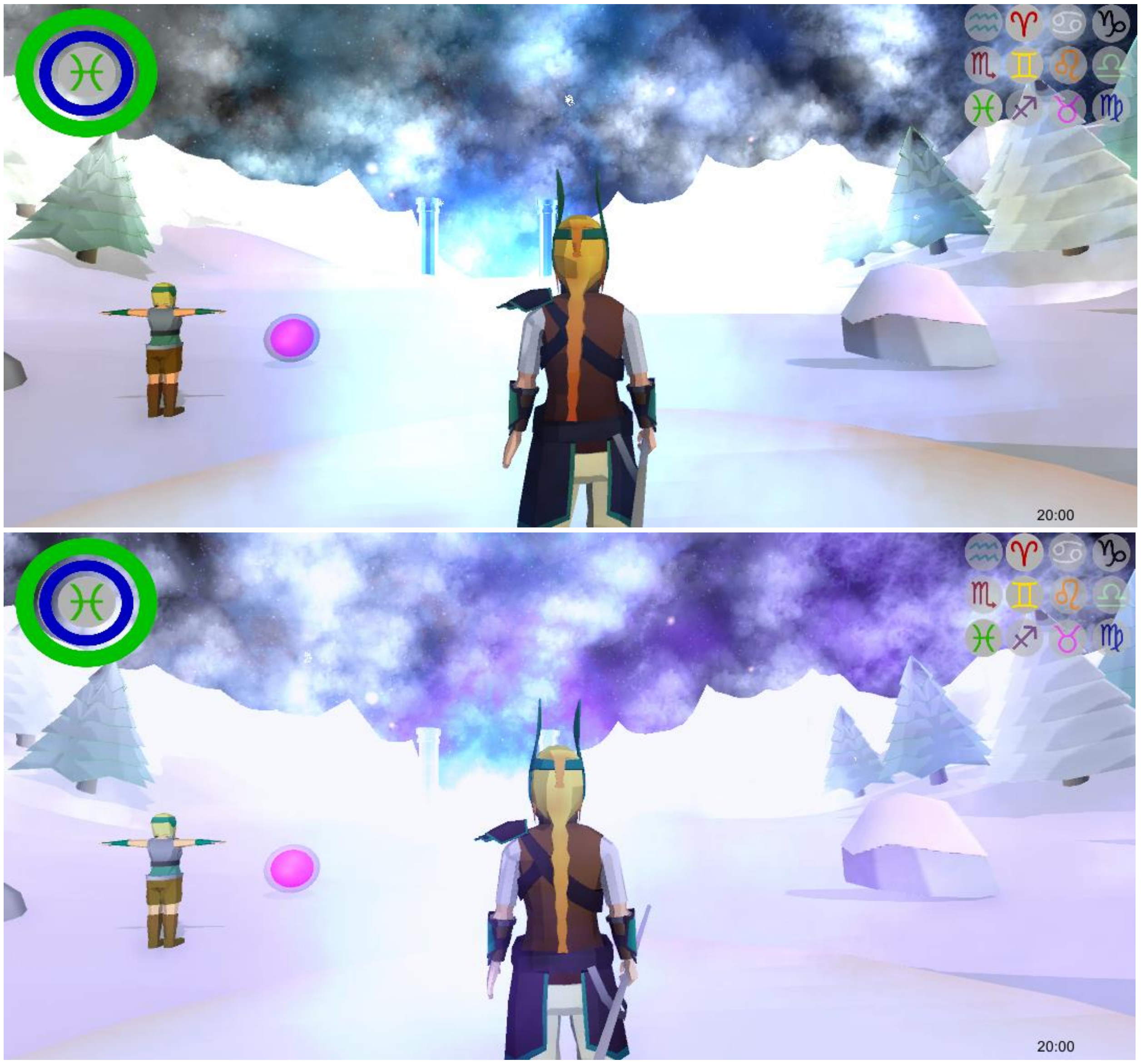


*Figure 21: The first image is the game without the post processing effects, the second is with the effects.*

How can be observed in the images, the difference is subtle but changes a lot the complete final result of the game. This way all the elements in the game seems to be part of the same world, before, some elements stood out over others.

# 7. Motion Capture

One of the objective of the project was to capture animations with the Motion Capture system and create more realistic animations for the game.

The subject that is related to this point is VJ1226 Character Design and Animation, during this subject a practice was realized with this method. Because of that, the idea of using them for the animations of the 3D characters was thought in order to learn more about the techniques used in real game studios.

The way to start with this method is to set the eight cameras in a space that allow the actor (the one who is going to performance the movements that are going to be captured) to have free movement. Then, start the program that calculate the available space and set the ground. After this, the actor has to wear the costume with the detectors in order for the program to create the skeleton, once the skeleton has been created, the actor can performance the movements and the program can start recording them. Finally, this information has to be saved as binary in order for other programs to be able to use it.

*Disclaimer: The following content is included on the three projects due to its common development*.

*Notification: Some things have changed in this common document about Motion Capture in order to add the references about the named games*.

Motion Capture technology, also known as Mocap, is a set of tools and techniques whose aim is recording accurate animations. Famous companies use this technology, achieving realistic results and a huge variety of different poses and movements. Companies such as Naughty Dog [36], developers of the recognized The Last of Us [37] [38] or the Uncharted [39], series, Quantic Dream [40], who developed Beyond Two Souls [41] [42] with facial Mocap, Crystal Dynamics [43] who worked with motion capture for the Tomb Raider (2013) [44] and Rise of the Tomb Raider [45], or Ubisoft [46] company with Watch Dogs [47] [48] and the Assassin's Creed [49] [50] series by using real parkour athletes. Even in 2001, one of the most known video game series, Metal Gear Solid [51] introduced Mocap techniques in MGS2 [52] but it was not until 2004 Metal Gear Solid 3: Snake Eater [53] when the use of Mocap was fairly relevant.

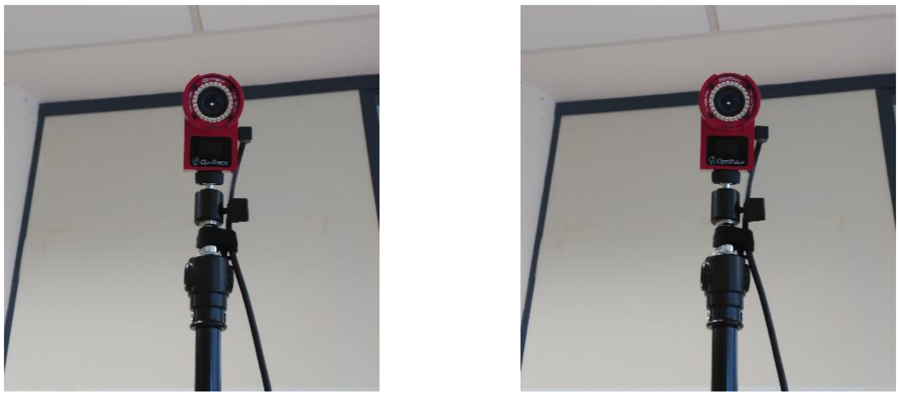
*Figure 22: Mocap recording for The Last of Us.*

For Mocap recording, there are some necessary elements and some specific situations that must be prepared. It is needed a large enough void room, several cameras surrounding the room, and a special suit with several markers, calibration tools and a software that controls the cameras, record the poses and be able to export them. The room can be added some different elements to help the actors such as chairs, stairs or boxes, as long as there are no light-reflecting elements that would hinder the camera detection. There must be at least 6 infrared cameras for a proper animation recording. These cameras have from 100fps to 300fps frame rate and their resolution is from 0.3MP to 4.1MP. The Mocap suit has got Velcro to attach the markers. The more markers the suit has attached, the more accuracy the animation will get.

## Configuration

OptiTrack Mocap is a passive motion capture system, this means that markers do not cast light but reflect it. The Mocap tools that have been used to record animations for these projects are eight OptiTrack cameras, two camera connectors, a calibration T-bar, a calibration L-bar, the OptiTrack suit, 37 markers (some of them are included in a plate for each hand) and a PC with the Motive application. The configuration process is the following:

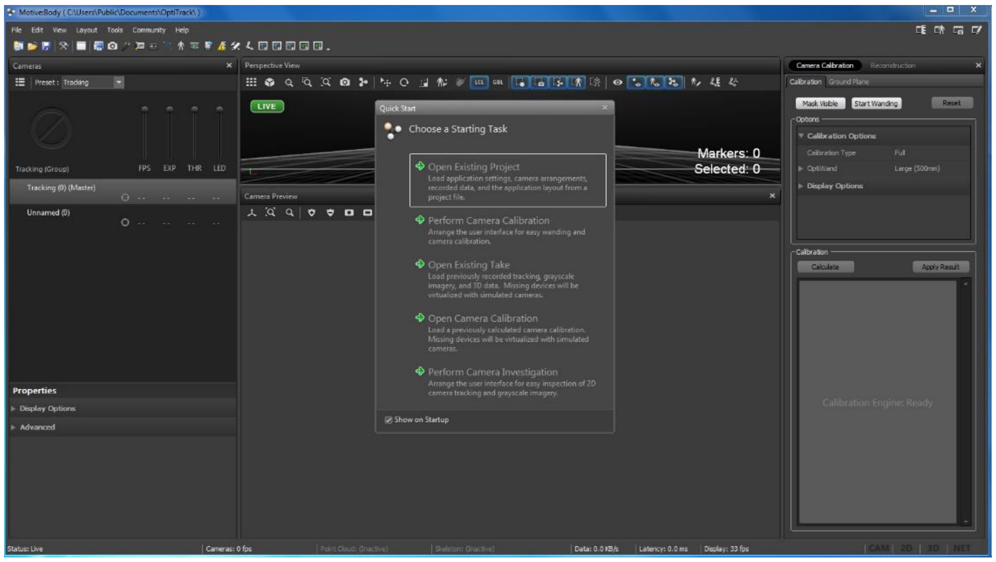
First of all, the cameras must be located around the capture area, this area could be square or circle. Is important that they are approximately aiming to the center of it. In this case, the cameras where situated in a round area in the room TC 1215 AA of the University, with enough space to capture the animations. The cameras have to be connected in two balanced groups with two camera connectors, and those connectors connected to the computer.



(a) *Non-detected camera.* (b) *Detected camera.*

*Figure 23: OptiTrack cameras*

Secondly, the computer and the Motive application must be opened so that the cameras are detected by the software. To verify whether the camera is connected it is just simple as seeing if the camera displays a number. Its number identifies the camera for the program. Cameras may detect some reflections that will be represented with white spots in the cameras' views. Then, it is necessary to mask the reflections, in this way the cameras will not consider those reflections as the actor's markers. After masking, those spots will turn red.



Quick Start

Choose a Starting Task

Open Existing Project
Load application settings, camera arrangements, recorded data, and the application layout from a project file.

Perform Camera Calibration
Arrange the user interface for easy wanding and camera calibration.

Open Existing Take
Load previously recorded tracking, grayscale imagery, and 3D data. Missing devices will be virtualized with simulated cameras.

Open Camera Calibration
Load a previously calculated camera calibration. Missing devices will be virtualized with simulated cameras.

Perform Camera Investigation
Arrange the user interface for easy inspection of 2D camera tracking and grayscale imagery.

[✓] Show on Startup

Markers: 0
Selected: 0

Calibration Type: Full
OptiWand: Large (500mm)

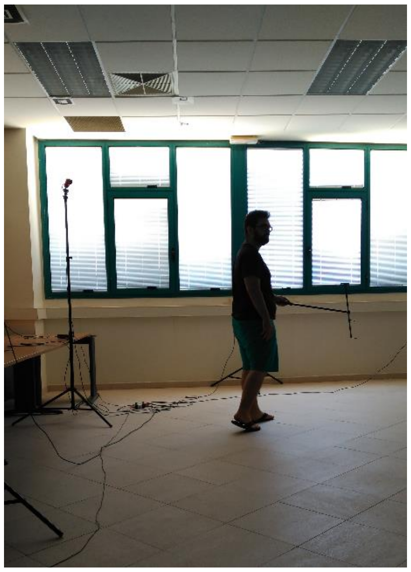
Calibration Engine: Ready

Status: Live | Cameras: 0 fps | Point Cloud: (Inactive) | Skeleton: (Inactive) | Data: 0.0 KB/s | Latency: 0.0 ms | Display: 33 fps

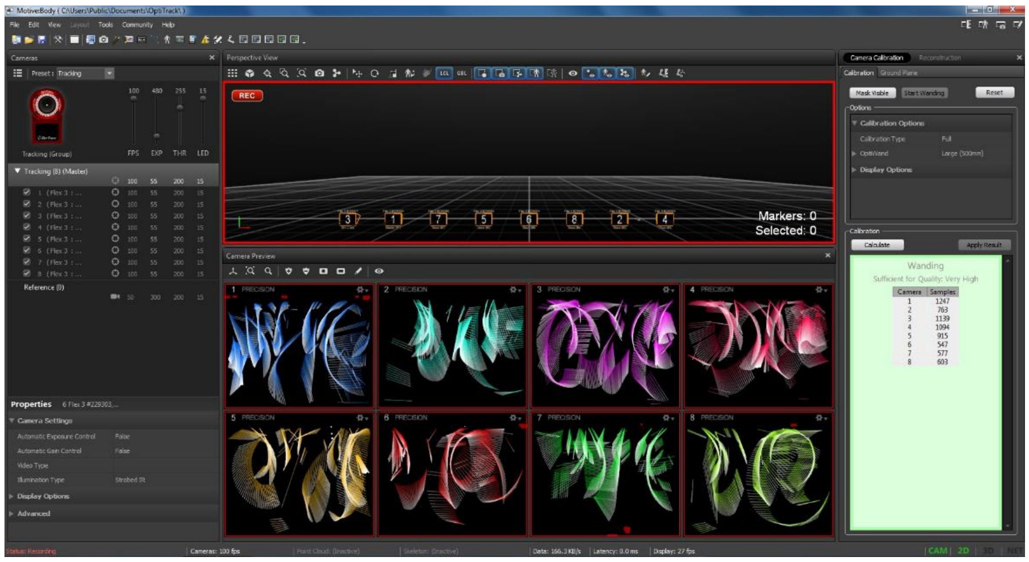
*Figure 24: Motive interface*

Then, cameras location must be learned by the software, it is necessary to calibrate the space by using the calibration T-bar, a T-form bar with three markers that is slowly waved through the area. Those markers are separated with a specific distance that the application knows so it can be used to a proper calibration. From the application, it must be indicated that calibration process is starting. Then, a person will wave the T-bar through the capture area, by drawing "eights". This process must continue until there are a proper amount of samples. After calibrating the location of the cameras, the floor detection must be done. The L-bar must be approximately left in the center of the detection space, so that the floor will be detected. The ground calibration must be saved in the computer, otherwise the animation recording will not be possible. It is noticeable that the whole calibration step must be done as soon as a camera is moved or displaced.

*(a) T-bar*


*(b) waving the bar*

*(c) L-bar*



Calibration Options

Calibration Type: Full

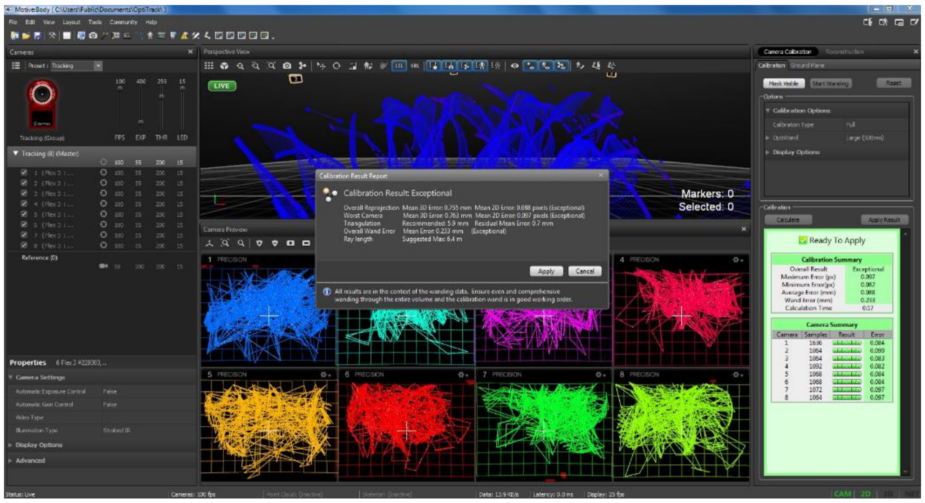
OptiWand: Large (500mm)

Wanding

Sufficient for Quality: Very High

| Camera | Samples |
|---|---|
| 1 | 1247 |
| 2 | 763 |
| 3 | 1139 |
| 4 | 1094 |
| 5 | 915 |
| 6 | 547 |
| 7 | 577 |
| 8 | 603 |

*(d) Taking samples from camera.*


Calibration Result Report

Calibration Result: Exceptional

```
Overall Reprojection  Mean 3D Error: 0.755 mm  Mean 2D Error: 0.088 pixels (Exceptional)
Worst Camera          Mean 3D Error: 0.763 mm  Mean 2D Error: 0.097 pixels (Exceptional)
Triangulation         Recommended: 5.9 mm     Residual Mean Error: 0.7 mm
Overall Wand Error    Mean Error: 0.233 mm    (Exceptional)
Ray length            Suggested Max: 6.4 m
```

All results are in the context of the wanding data. Ensure even and comprehensive wanding through the entire volume and the calibration wand is in good working order.

Calibration Type: Full
OptiWand: Large (500mm)

Ready To Apply

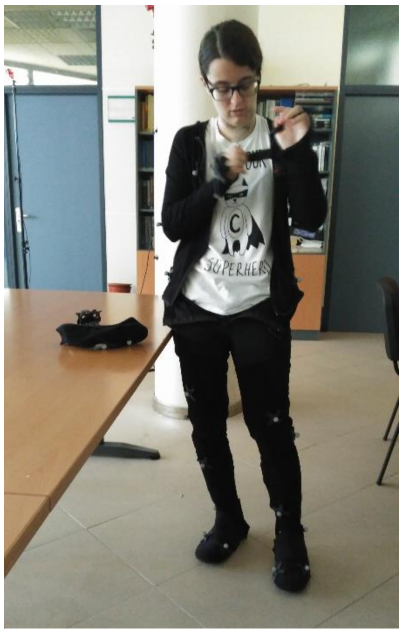
| Calibration Summary | |
|---|---|
| Overall Result | Exceptional |
| Maximum Error (px) | 0.097 |
| Minimum Error(px) | 0.082 |
| Average Error (mm) | 0.088 |
| Wand Error (mm) | 0.233 |
| Calculation Time | 0:17 |

| Camera | Samples | Result | Error |
|---|---|---|---|
| 1 | 1636 | 🟩 | 0.084 |
| 2 | 1064 | 🟩 | 0.090 |
| 3 | 1064 | 🟩 | 0.083 |
| 4 | 1092 | 🟩 | 0.082 |
| 5 | 1068 | 🟩 | 0.084 |
| 6 | 1068 | 🟩 | 0.084 |
| 7 | 1072 | 🟩 | 0.097 |
| 8 | 1064 | 🟩 | 0.097 |

*(e) Successful sampling.*

*Figure 25: Camera calibration.*

Before stating the recording, the actor has to dress up with the special suit, composed by a jacket, pants, shoe and a hat. Then, another person will put the markers on the suit, according to the places where the application suggests to get a correct animation. In the application, the skeleton has to be created by using a 37-marker skeleton model (called Baseline). After that, the skeleton is created by pressing "Create" (a preview of the skeleton can be created by selecting ‘On’ in the "Skeleton Preview" section).


*(a) Actor dressing up.*

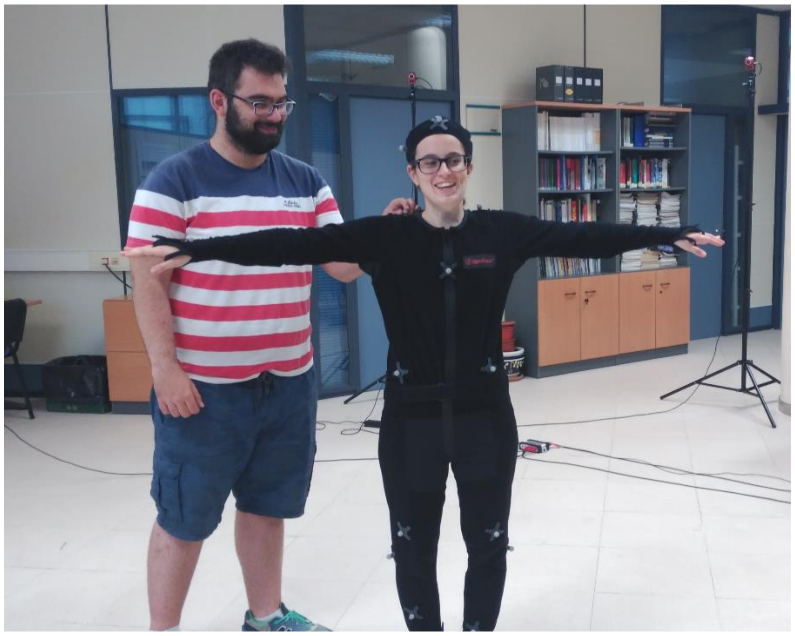
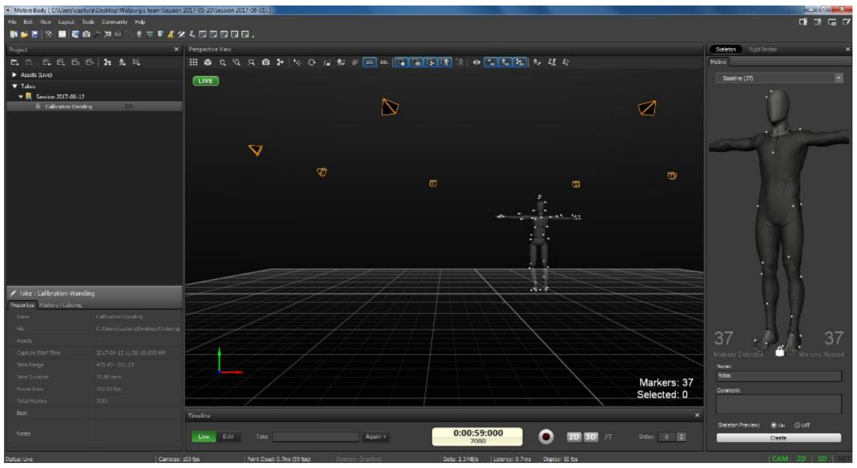
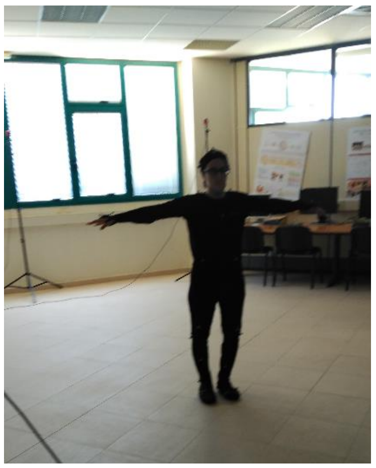

*(b) Putting markers on the actor.*

*Figure 26: Dressing the actor.*



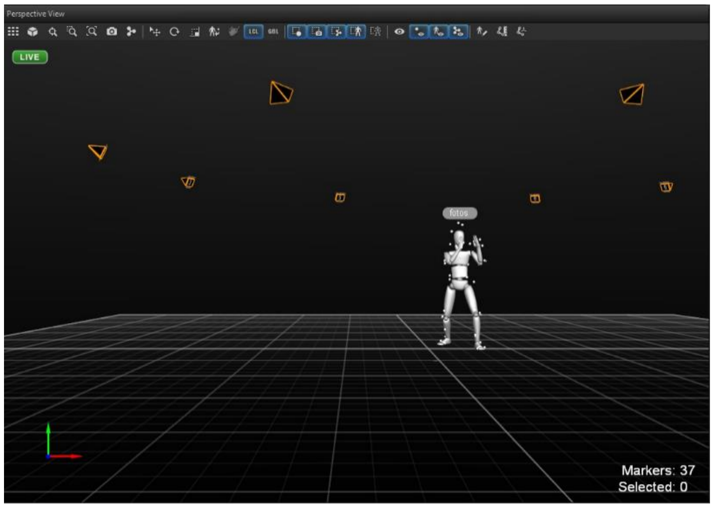


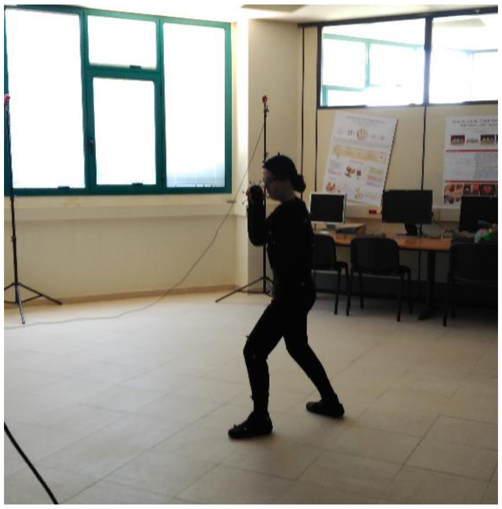
*(a) How to create a skeleton* *(b)Actor making a T-pose.*

*Figure 27: Skeleton creation while t-posing.*

Finally, the actor will be able to be tracked by pressing the Rec button in the program. The cameras will track whatever the actor does while the Rec button is pressed. Once the recording is finished, a take will be saved into the application. In this case, several takes have been recorded with different items like fake swords, axes or spears that would help the actor to make the pose. Sometimes, tracking may present swapping problems, that means that the software swaps the markers tags and distorts the skeleton.

At the time to export, there are many options but the one used for these projects es .fbx binary, since Autodesk software presents problems at reading and importing these skeletons.

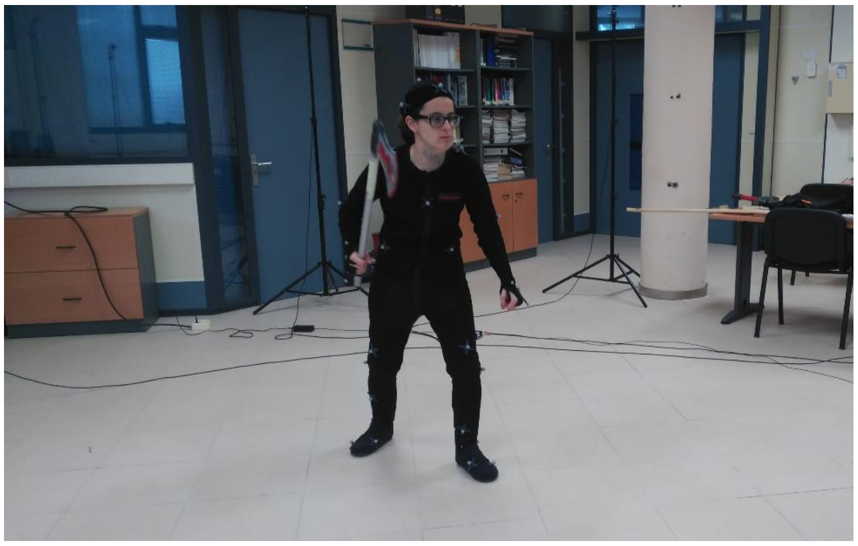




*(a) Posing in Motive.* *(b) Real posing.*

*Figure 28: A real pose and how it looks in Motive.*

*(a) Using a fake dagger.*


*(b) Using a fake axe.*

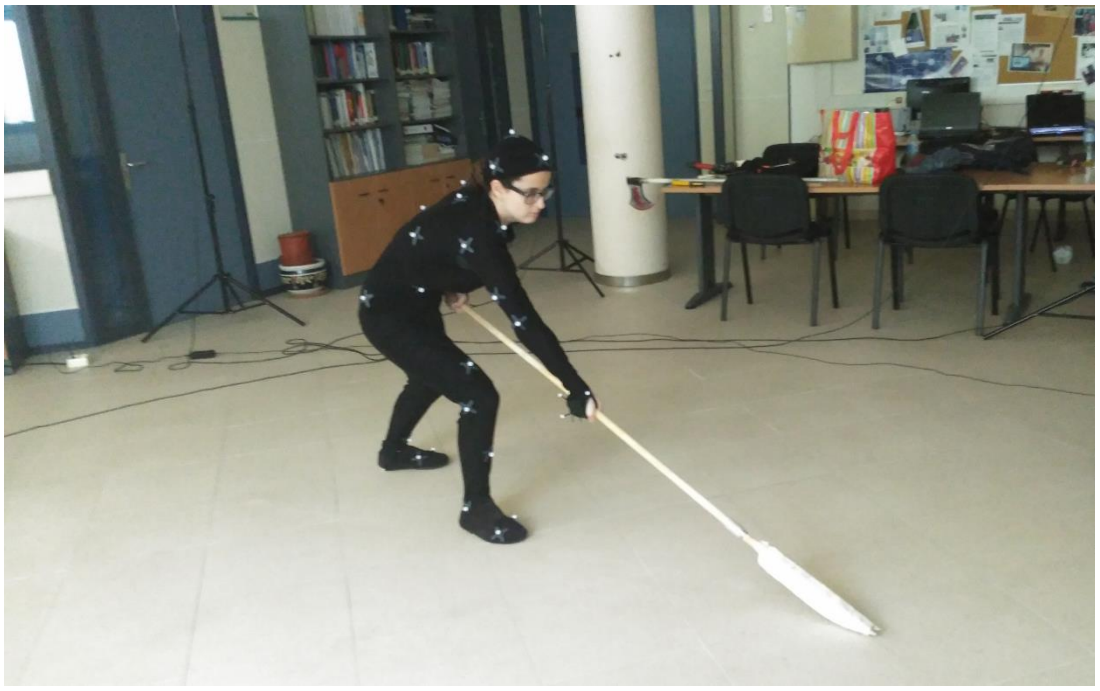

*(c) Using a fake spear*

*Figure 29: A real pose and how it looks in Motive.*

# 8. Animations

Motion Capture is the technique used to obtain the animations as it was explained in the last chapter.

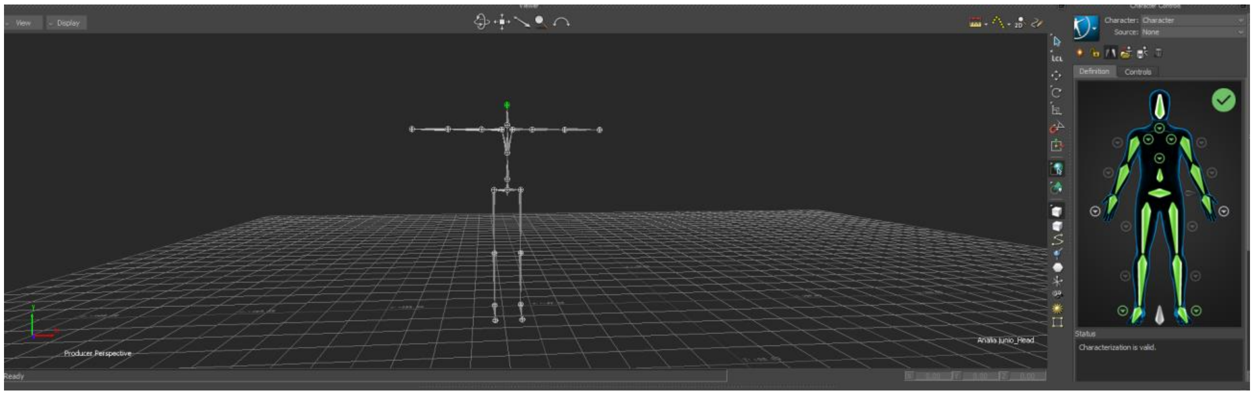
The result of the capturing was around forty clips of animation between the Valkyrie and the enemies' actions. But a problem happened with this captures, some of them were not useful to use due to some problems with the rotations of the axes, making them make strange movement. Because of that it was used a program to clean the animations, this program is MotionBuilder.

## 8.1 MotionBuilder

MotionBuilder is useful to manipulate and refine the data from the animations created with the Motion Capture technique.

The animation is opened in the program and the obtained skeleton is selected. Once it has been selected, the skeleton must be defined and the relationship between the skeleton obtained by the data and the program must be created.

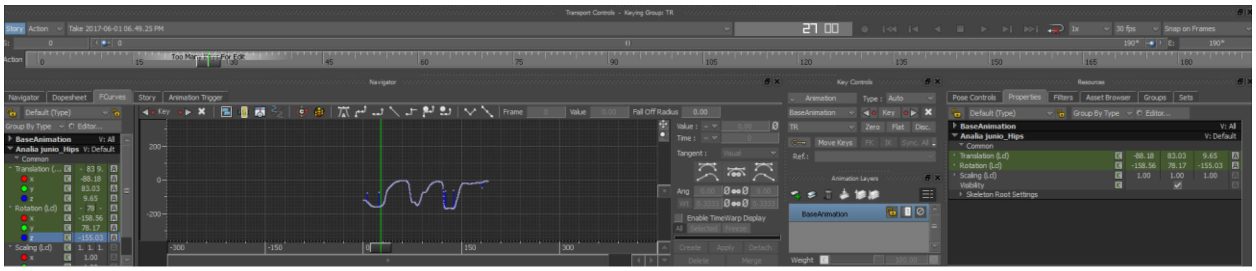
But before perform this connexion, the skeleton must be in a T-post in order to have all the axes in the correct direction, if it is not, it can create some problems while starting the connexion.



Status

Characterization is valid.

a) *A screenshot of a connexion correctly created*

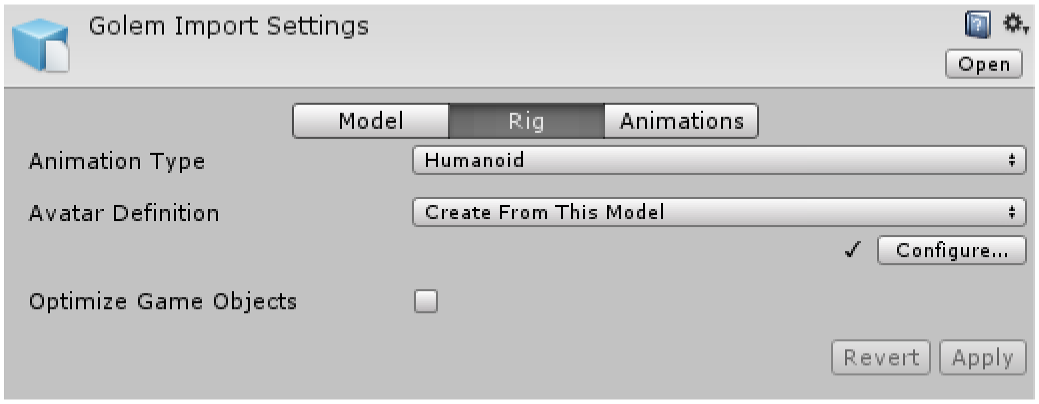


b) *The frame-by-frame controller and the Fcurves editor.*

*Figure 30: A screenshot with the elements in MotionBuilder*

After this, the skeleton has to be locked and defined as a biped. The next step is to control the frame-by-frame animation and pay attention to the frames that perform some unwanted movement. The reason for this control is because the Fcurves of the animation corresponding to each frame has to be open and when this undesired movement is found, go over that frame and eliminate that moment from the control curve.

Finally, when the animation has been cleaned or at least the parts that are interesting to use, this animation is imported to Unity 3D and in importing settings, they have to be set as 'Humanoid' in the animation type and the same with the 3D model that is going to be the one affected by the animation.



Golem Import Settings

Animation Type: Humanoid

Avatar Definition: Create From This Model

Optimize Game Objects: ☐

*Figure 31: The Import Setting that transform the model to a Humanoid type.*

In the animation mode, the 3D model can be dragged to see, before implementing, how the final result is going to be and edit it to change the start and the end in the frames of the clip. In this mode, there is also possible to put events on the animation, selecting the clips when this has to happen, this has been used for the footsteps, in order to control better when the foot hit the ground and when it has to play the footsteps audio.

However, a 3D model will not work properly with animations if the model has not an exported skeleton.

Another thing to take into account is that MotionBuilder also allows to create animation loops selecting the starting frame and the ending frame. This has been used to create the loop for the enemies walk movement.

The way to create the skeleton from the character has been two, for the characters with a form that does not match very well with the basic human structure it has been used the program Mixamo [54] because it was difficult to reach a good skeleton with the envelopes and the biped figure that 3D max allow to create when the type of figure that it is been searched it is a non-human but with human skeleton behaviour and even if the auto rigging algorithm from Mixamo does not respect the different parts from a mesh (the different parts of a mesh are combined as one), the skeleton obtained match well for the animations.

The second way is creating a biped that match the model, 3DS max allows to create bipeds that has to be adapted to the position of the mesh. In figure mode, the biped can change the position and rotation of its parts making this possible to place the skeleton in their corresponding positions based on the human anatomy.

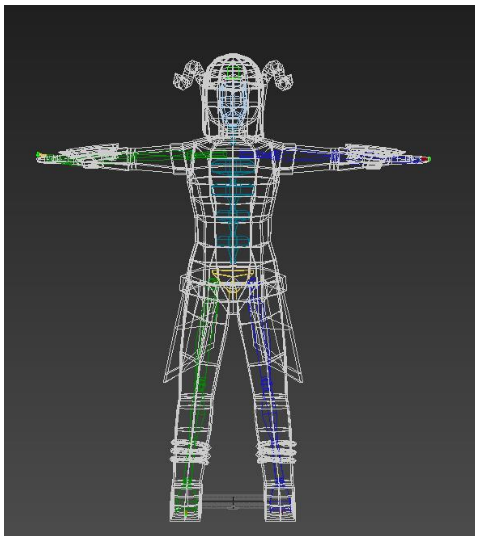


*Figure 32: Image of the biped placed in the mesh*

Once the biped is placed, the mesh needs to has a physics component, in this component there is 'Attach to Node' that when is selected it has to select the mass centre in the biped and it will create the relationship between the biped and the mesh.

The final part is to put the envelope in the bones. With envelopes it is possible to control the skin deformation and it limits the range of influence.

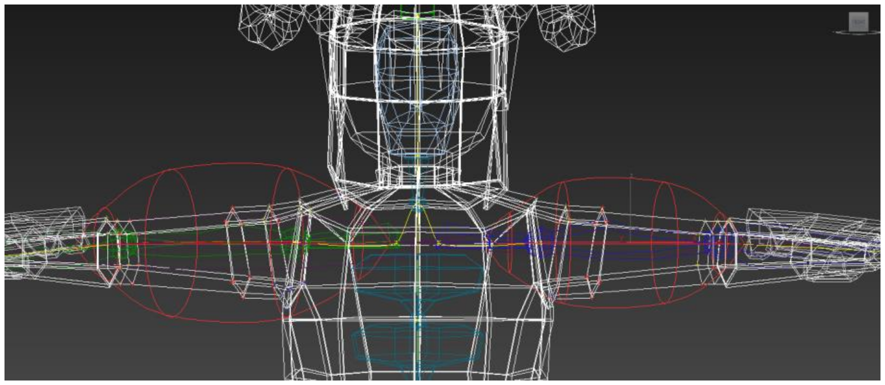


*Figure 33: Image of the envelopes explained before*

After all this is in place, some checks have to be done by moving the bones and testing different postures to check that the envelopes are well placed and the mesh does not deform too much with the movement.

# 9. Art

The art in the project is one of the more important aspect because it is the first impression that the player will have about the game.

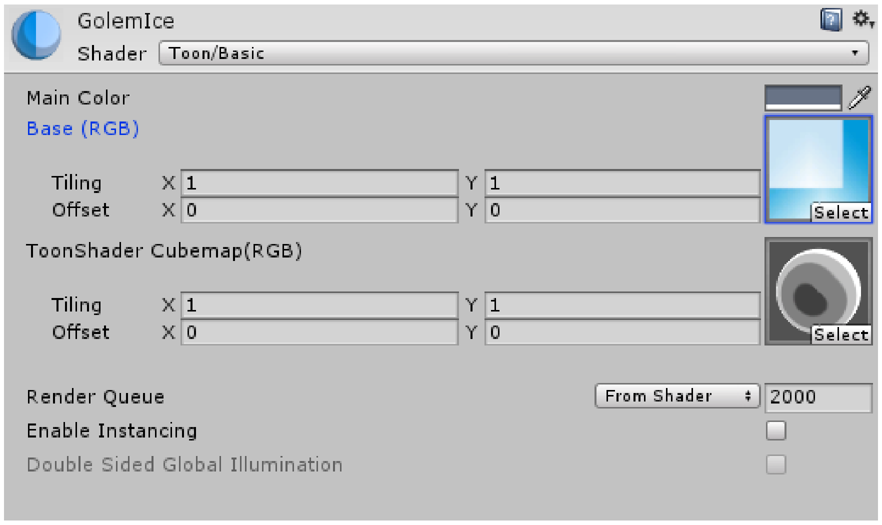
As it was explained before, the level is based in Nordic mythology and in that mythology there is this word named Jötunheim. This world is a frozen world where giants live, and because of that the colour palette had to be made with this concepts.

In the first approach with this section, the images that were found about this world were taken into account.

*Figure 34: Example of demonstrations of Jötunheim. Images from Internet*

As it was explained in the Related Work, the type material that is wanted to be used for the textures is Toon Shader. This effect can be downloaded from the Standard Assets that Unity 3D has and it is very simple to use. Also, to simulate a starry sky to support the idea that the game is happening at night, a skybox from the Asset Store has been used [55].

When a new material is created the only that has to change is the type of shade, once this effect has been download, the shade can be changed into 'Toon/Basic' that is the type of shader used in the project, but Toon Shader also offers three more types of shaders: Basic Outline, Lit and Lit Outline. Given that the effect sought was to simulate the shader in Zelda Majora's Mask, the Basic was good enough.



GolemIce
Shader Toon/Basic

Main Color
Base (RGB)

Tiling X 1 Y 1
Offset X 0 Y 0

ToonShader Cubemap(RGB)

Tiling X 1 Y 1
Offset X 0 Y 0

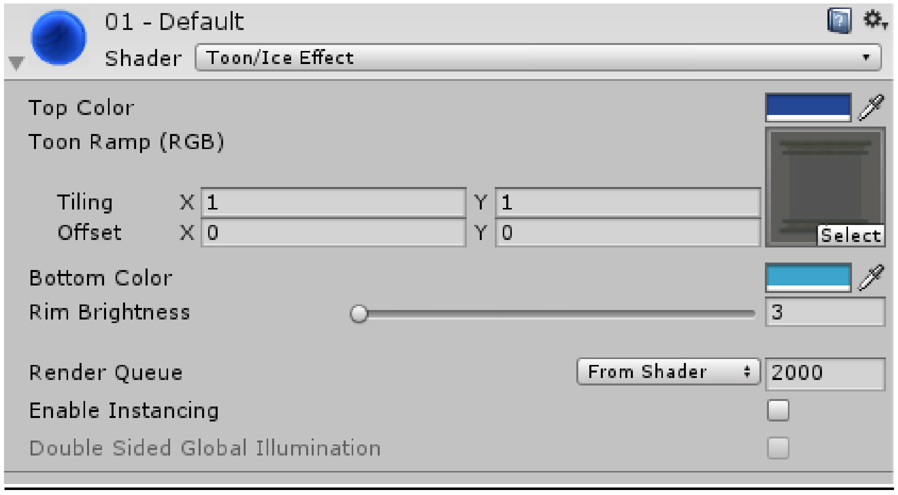
Render Queue From Shader 2000
Enable Instancing
Double Sided Global Illumination

*Figure 35: The Toon Shader Basic material in Unity Inspector.*

In the image can be seen an example of how this shader works. In Base goes the texture that has to be implicated, in order to look good, the UV mapping of the model that is going to wear this material needs to be prepared, ToonShader Cubemap determines the shadows and the lighting in the object and 'Main Color' allows to change the colour of the entire model in order to improve the final result.

Another of the shaders used is a modification of the Lit Toon Shader. This shader has been used for the surfaces that needed to look more like ice due to the effect that this shader provides, a toon ice effect.

The shader code was obtained from an artist and developer who make tutorials about 3DS max and Unity 3D [56].



01 - Default
Shader Toon/Ice Effect

Top Color
Toon Ramp (RGB)

Tiling X 1 Y 1
Offset X 0 Y 0

Bottom Color
Rim Brightness 3

Render Queue From Shader 2000
Enable Instancing
Double Sided Global Illumination

*Figure 36: The Toon Ice Effect Shader material in Unity Inspector.*

The following examples are the models created for the project:

## 9.1 Scenarios

This level is divided in three parts: the initial scene, Utgard (the city of giants) and the frozen dungeon. Each scenario is a different challenge for the player and the colour palette reflect that conflict because it goes from faded blue colours with glimpses of green to more electric and energetic blue colours.

### 9.1.1 Initial Scene

The image that is shown below shows the renderer images obtained in 3D Max from the modelling of the first scenario.

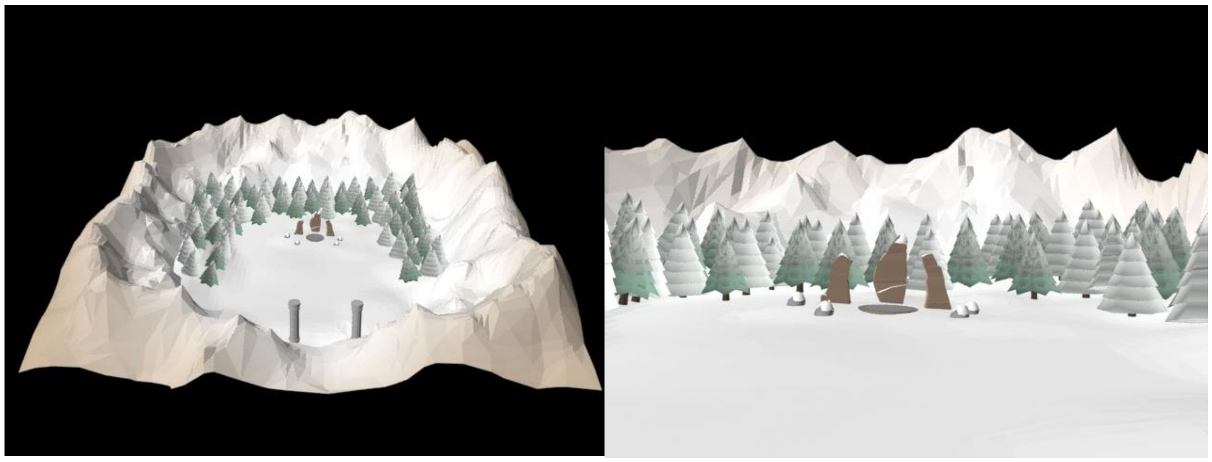


*Figure 37: First scene example renderer in 3ds max.*

A small forest can be found in this scenario that surrounds the stone ruins where the Valkyries will do their first appearance. These ruins are the initial point in the scenario.

All the scene has being modelled in 3D Max trying to achieve a good result without using a big number of polygons.

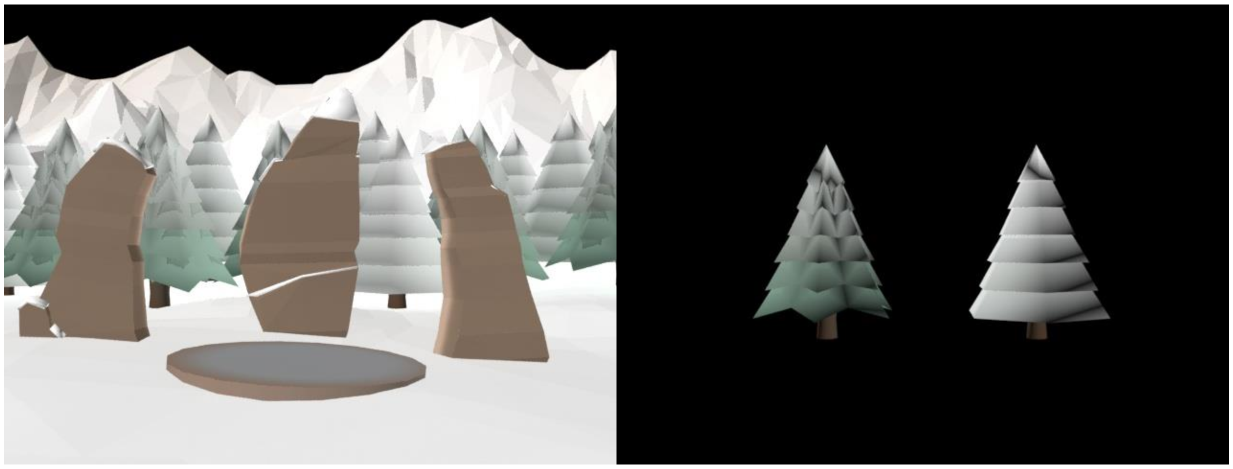


*Figure 38: A better look to the elements in the scene*

In order to place the textures in a way that creates the desired effect, it is necessary to make a Unwrap UVW [57] of the modelling to obtain its texture map, then the texture map has to be saved and put it in Photoshop to place in the desired position the colours that we want to appear as the finale result.

After this, the image created has to be placed (without the edge lines) as a texture for the object. If the effect obtained is not the one desired the texture map can be edited in the UV editor and the vertices can be modified until you get the desired result.

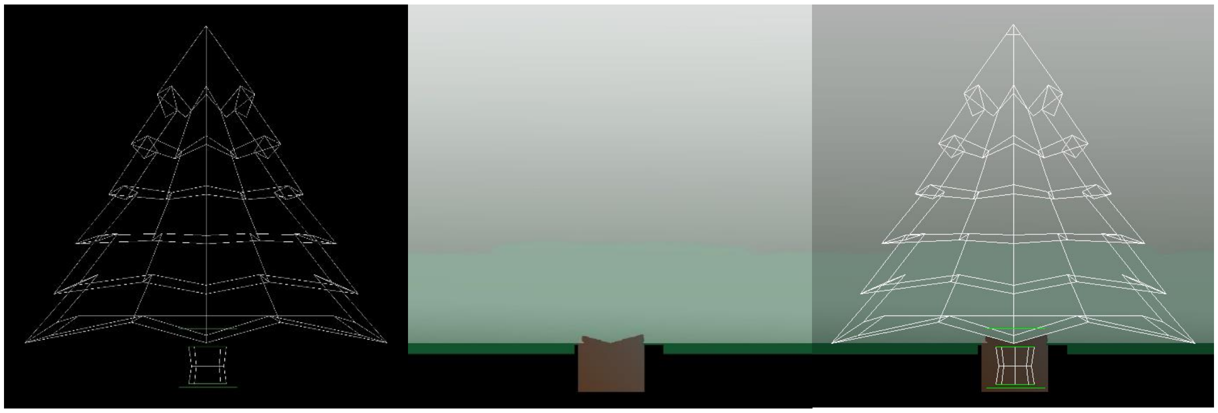


*Figure 39: The process of the Unwrap UVW*

### 9.1.2 Utgard

The city of the giants is the second available scenario. Due to the giant's size and the scarcity of a large amount of resources, the giants live inside caves to guard against the great storms that sometimes ravage Jötunheim. The entrance of the frozen dungeon is in this place but it is being guarded by the guardian that patrol the entrance.

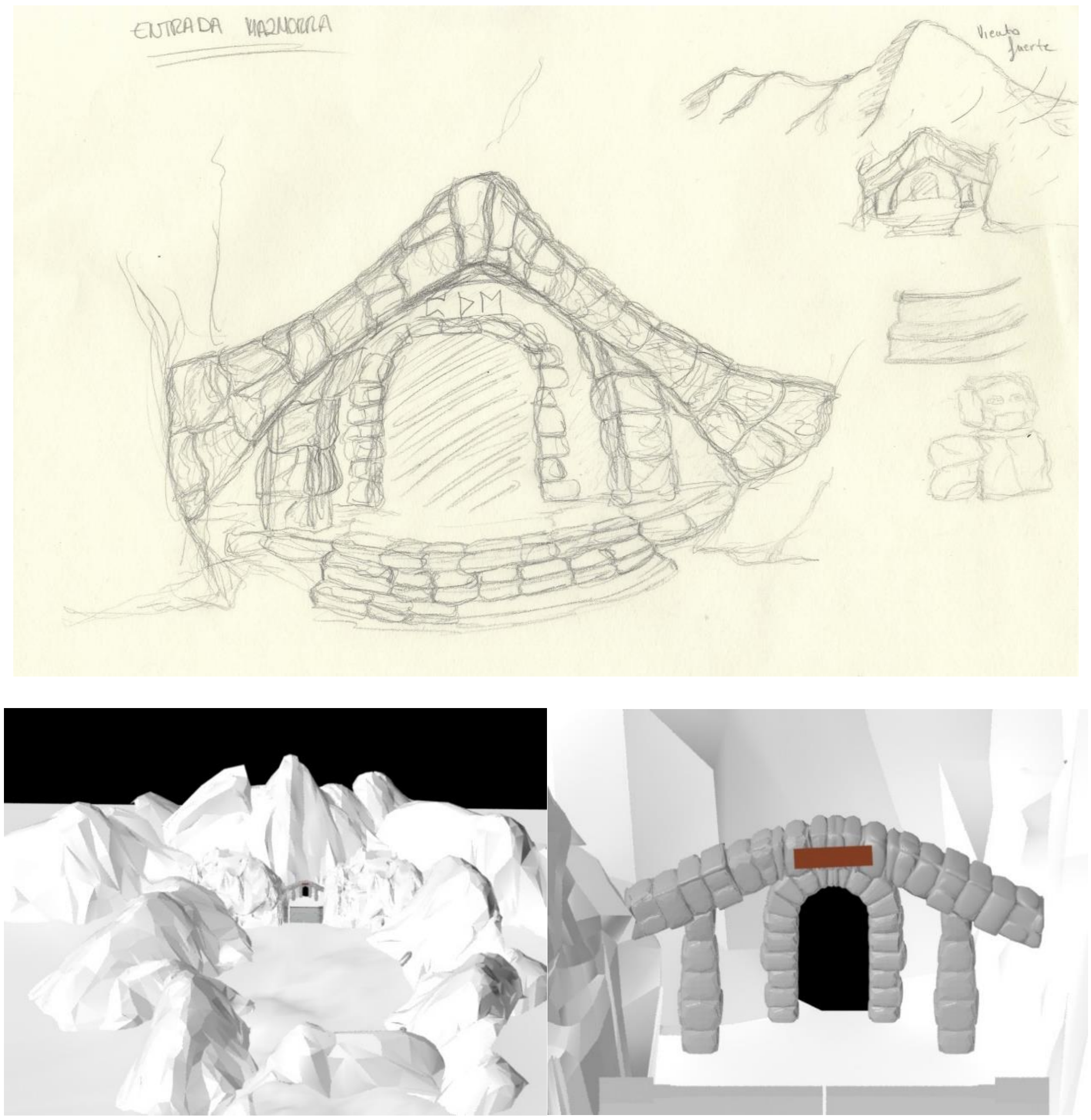


*Figure 40: Image from the concept art of the entrance of the dungeon and renders from the Utgard scene without textures.*

## 9.1.3 Frozen Dungeon

The dungeon is the last scenario of the level and it is formed by four rooms, the first one is the entrance where there are some enemies, the second is where the frozen bridge is, the third is like an ice slide and the fourth one is where the Valkyrie Taurus is being trapped and where the last battle takes place.

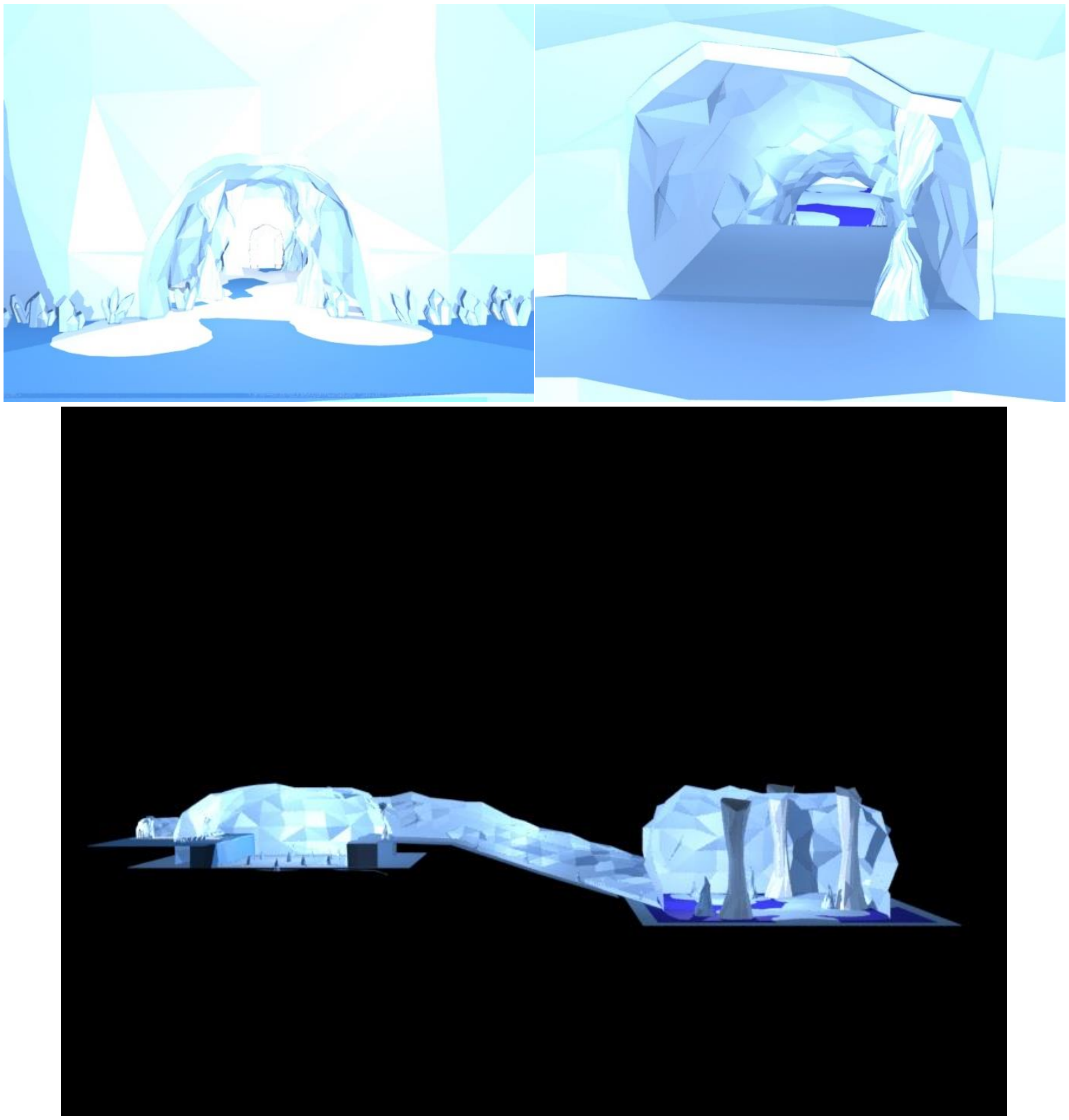


*Figure 41: Screenshots from the ice dungeon*

## 9.2 Enemies

There are three types of enemies in the game:

### 9.2.1 Giants

The giants are based on a humanoid figure so their constitution remembers to a human but with the superior part of the body wider.

There are two models of giants, the normal NPCs and the guard giant with whom the player will have to face.

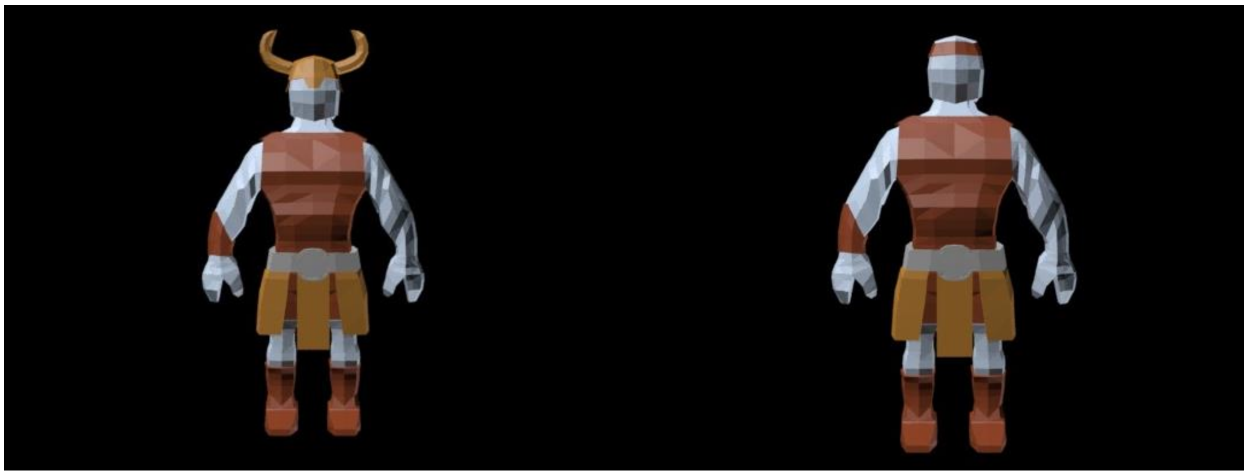


*Figure 42: Screenshots from the giants' models.*

### 9.2.2 Golems

The golems are based on a set of rock masses or ice to simulate the sensation of a heavy body that can be difficult to defeat.

- Stone Golem in unity with Toon Shader

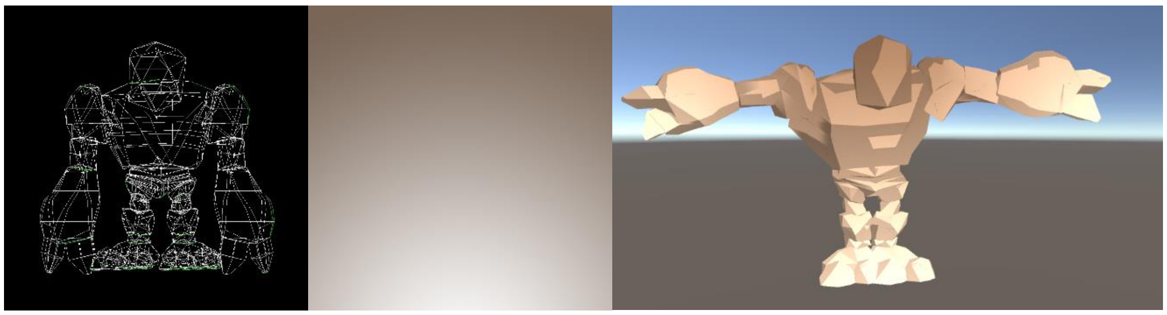


*Figure 43: The process for the stone golem (UVW map, texture, final result with the shader in Unity).*

- Ice Golem in unity with Toon Shader.

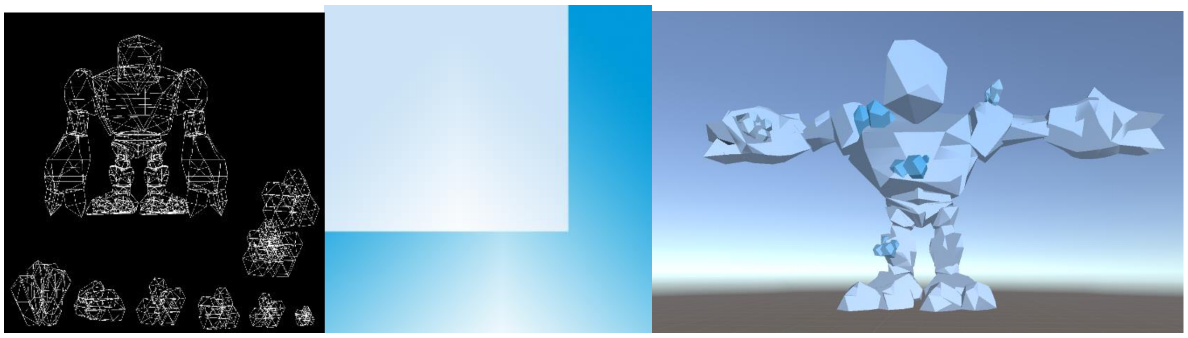


Figure 44: The same process for the ice golem

### 9.2.3 Giant's King

The king is the one who rules Jötunheim, because of that it has to be different to the other giants to stand out that position of power. It is based in the same humanoid form that the other giants but the clothes and the details are different.

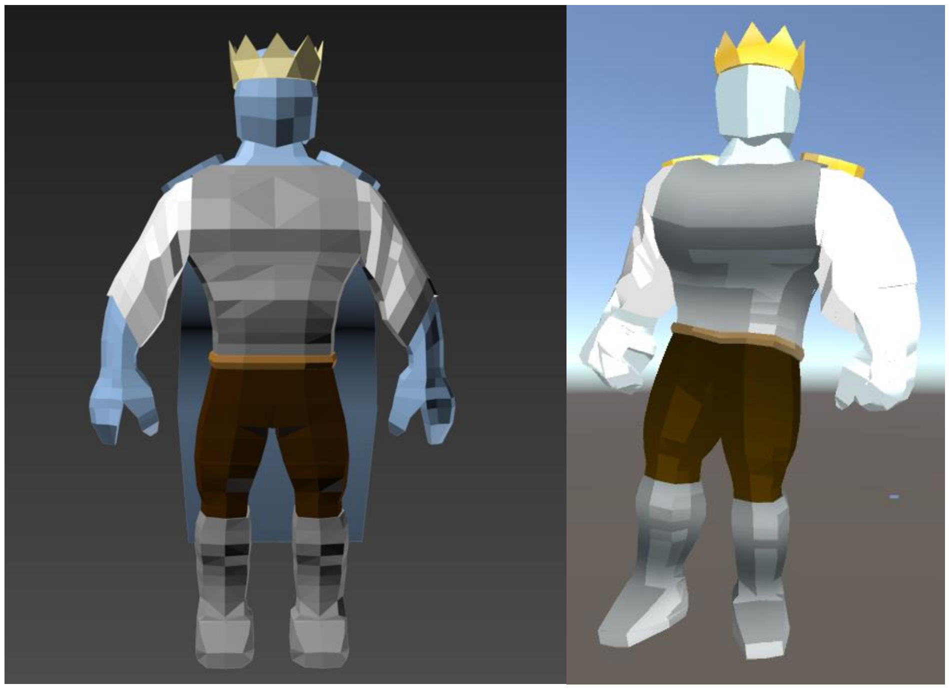


Figure 45: The model of the king of the giants.

## 9.3 Objects

In order to obtain the objects they have to be represented on the scene and also, they need icons to be presents on the inventory.

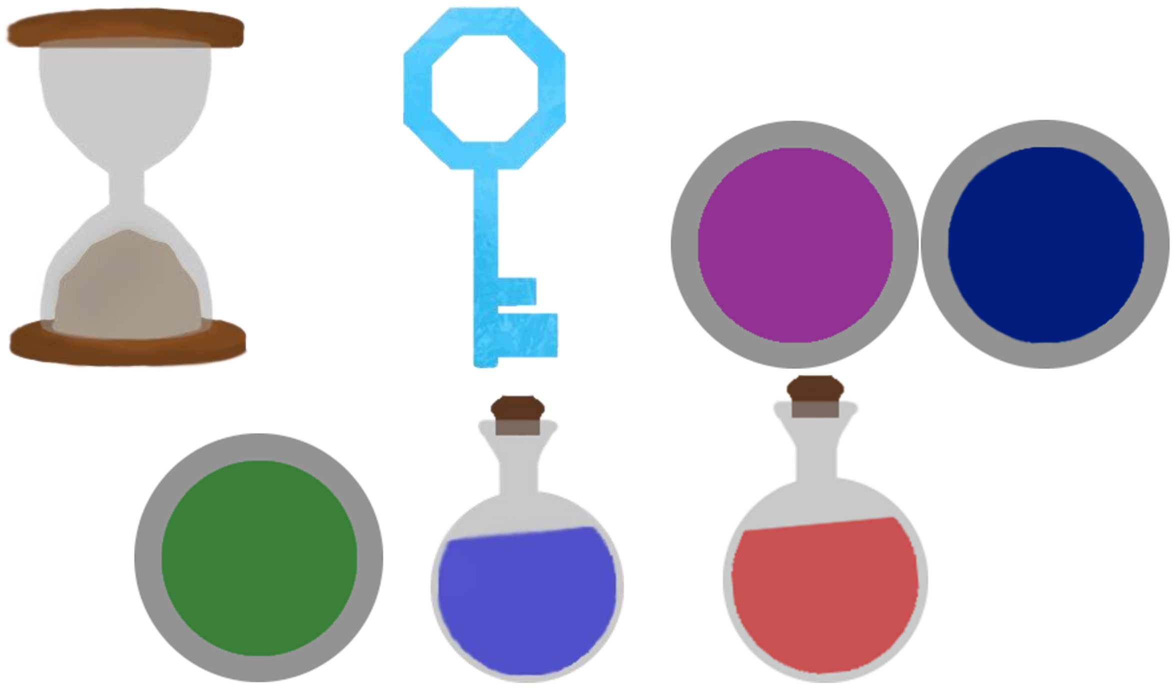


Figure 46: Icons for the inventory made with Photoshop

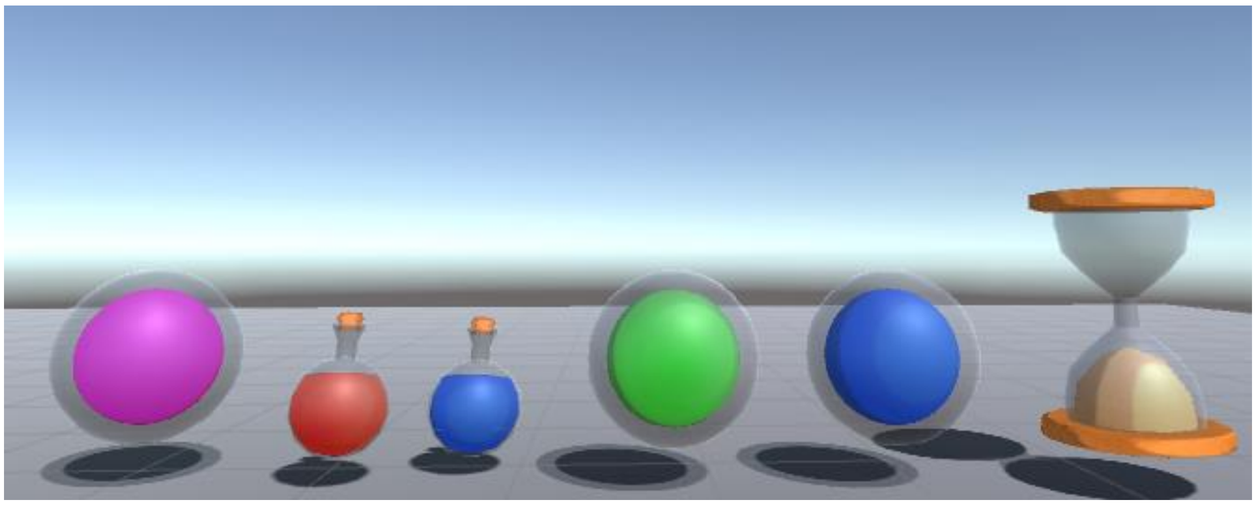


*Figure 47: Objects in scene with Toon Shader texture in Unity 3D*

## 9.4 Valkyries

Aries is a Valkyrie with fire powers, for that reason, the selected colours are red, black and grey to simulate the violence that this character can create. This model may be subject to change.

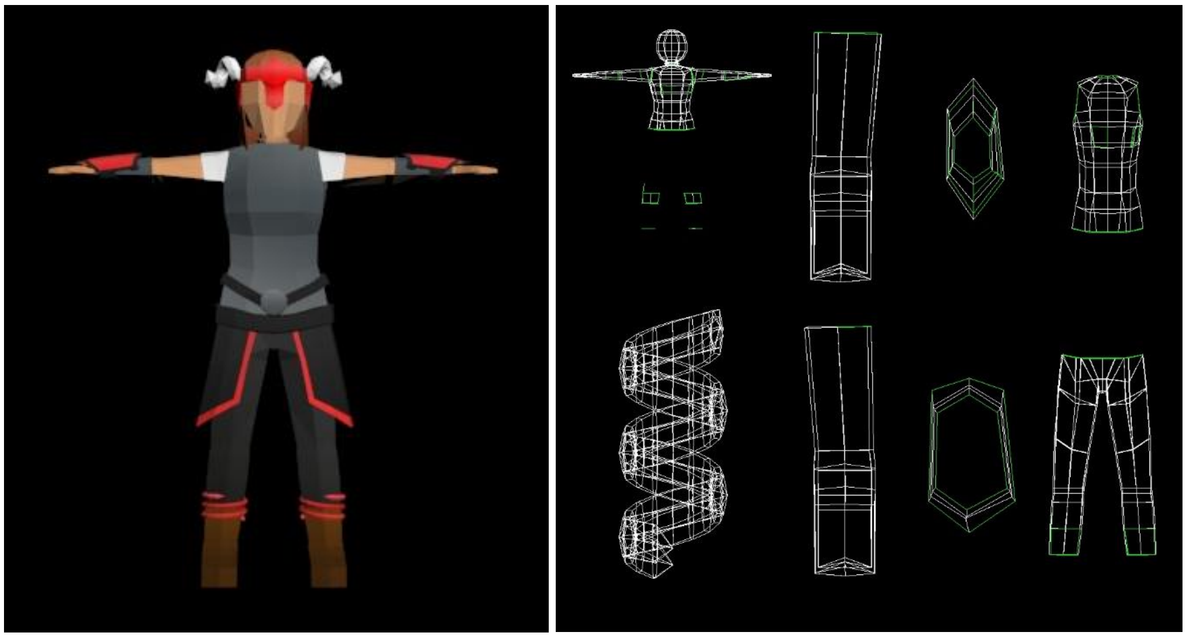


*Figure 48: The 3D Model of Aries and its unwrap.*

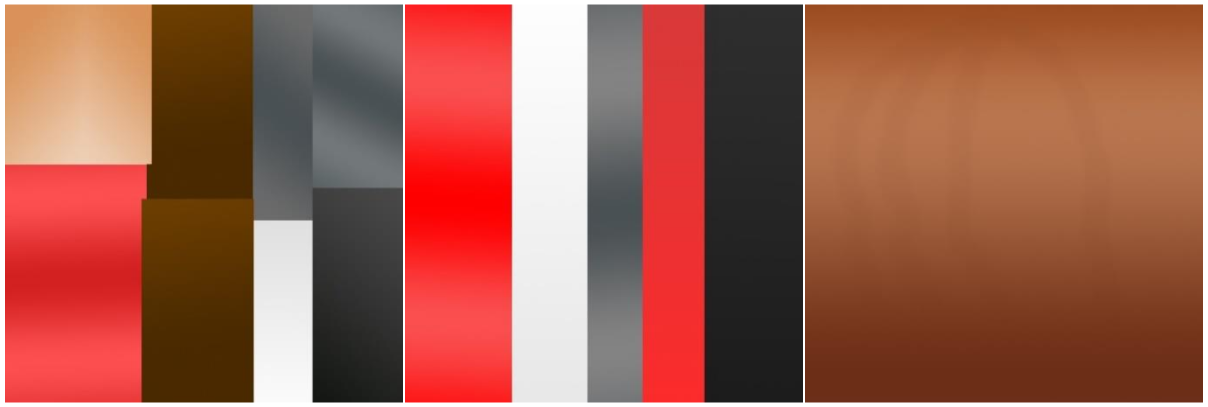


*Figure 49: Textures of Aries*

Tauro is the Valkyrie that the player has to find in Jötunheim. This Valkyrie is more physical than the others in a way that she uses physical attacks with her fists and her magic power is based on freezing the adversary.

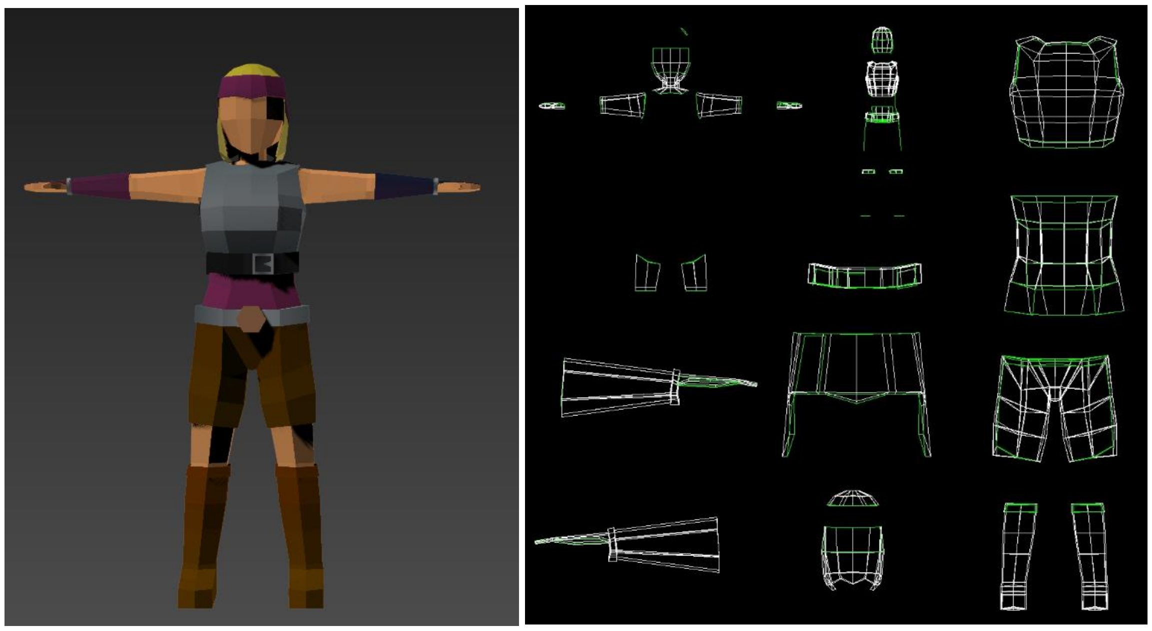


Figure 50: The 3D Model of Taurus and its unwrap.

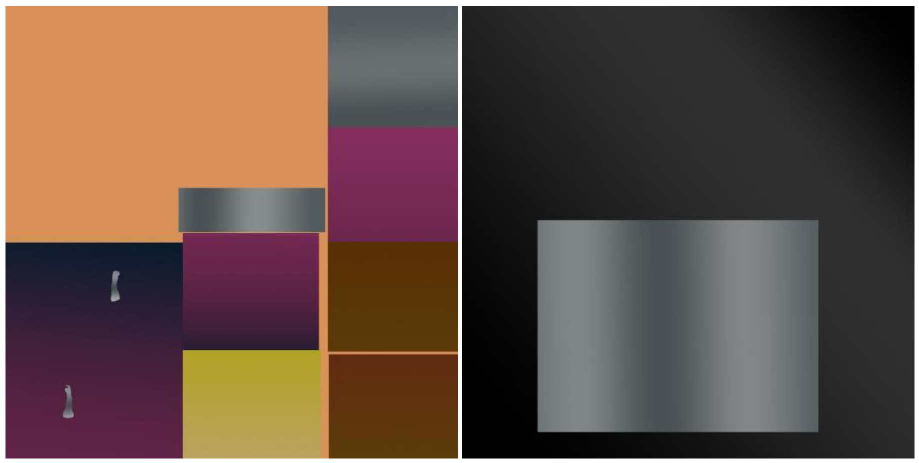


Figure 51: Textures of Taurus

The Valkyries are built based of a common body structure to create the sensation that all them are a similar group and to achieve a concordance with the rest of the elements in the game and also, for futures plans and developments.

In addition, the colour palette of the Valkyries are based on their zodiac symbol and the color that the interface have of each one and they abilities in the game, that why Aries has red colours due to her control over fire magic or why Taurus has purple in her clothes, because the symbol of Taurus in the game has a similar colour.

# 10. Testing & Evaluation

In this chapter it is necessary to emphasize the difference between the two types of testing:

- Individual

-Collective

This difference is needed because the project is both, individual but with common means, so when a task was finished it was needed to make a group control in order to see if the result was the one find it for everyone and to control if it worked okay with the tasks created by the others.

## 10.1 Individual

For the individual testing and evaluation, the plan to follow was to set up the goals to achieve with each task assigned and once it was thought that goals were achieved, a control of the implemented started.

If the implemented worked at the first time, it did not take it as the final one but it started a control of all the possible forms that implementation could be used. When it did not work, the procedure was to find the point when the problem started and make prints to see were the problem was or to control if some object was being received as null.

For example, with the change character implementation, there were problems once the new character was added that was because the character that it was trying to add was the one on the scene but that character needed to disappear ounces she was playable.

So after this problem, a not active character was set in the hierarchy and the character on the scene would add that not active character, so the problem was resolved.

Another problem was with the footsteps prints, the prints did not appear in the same position as the player even when the Raycast was implemented, this problem was resolved when in the direction of that object, it appear that the X axes looked up so there was no way the ray cast would intercept the ground and the solution was to make the X axes looked down, to the ground.

The biggest setback was with the Ice Physics, when the Physic Material did not work as expected and the result that was sought was not adequate.

Because of this, different approaches were studied and after a few tests, it was decided that the way to resolve this problem would be with the forces applied in an inclined plane. This plan also give some problems like the one that the obtained forces were so small that the result was not appreciable, for this reason, the forces where multiplied with different numbers until the result obtained was acceptable.

The final challenge was to find the direction a body has to move when those forces were applied. In a surface without rotation to a body glide in the direction of the movement but when there is rotation different forces are applied and depending of that the magnitude of the rotation, the body will glide in a direction or another (because it is not the same a surface with rotation positive on X than negative on Z).

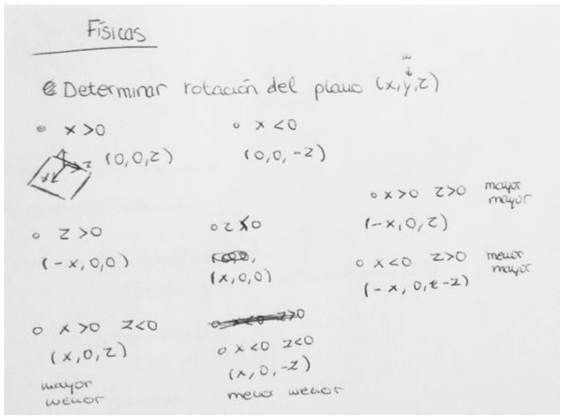


*Figure 52: An early study of the rotations to control the character movement*

## 10.2 Collective

*The complete information about this is in "Anexo I".*

After a task was finished and had the approval of the group, the next step was to adapt it to the hand control and resolve all the problems that could appear with that.

The group had a semanal reunion to see the advance in each task that was being controlled by a Trello [58] and a final delivery date, this way both the individual work and the collective work was controlled.

When the project was more advanced, the group started to have more reunions and nights non-stopping work.

# 11. Deviations

*The planning can be found in the chapter of the Technical Proposal.*

The initial planning has changed during the project. When the schedule was established it did not take into account the two holiday's weeks (Magdalena and Easter, when during Easter was the final degree travel with the classmates).

Furthermore, some task that wasn't in the initial planning had been added and other removed depending of what the project needed and some of the task that stayed were longer or shorter that first expected. At the beginning, the game was going to have four scenarios where the second was going to be a place for the player to learn how to use the controls against some enemies, and also, the amount of 3D modelling has been lowered as time went on in order to achieve other more important aspects.

This type of setbacks and the problem of lack of time have reduced some sections of the project.

| Week | Task (hours) | Artistic Task (hours) | Programming Task (hours) | Hours in the week |
|---|---|---|---|---|
| 20/02 - 06/03 | A (7) | B (20)<br>C.2 (5) | E.1(2)<br><br>D(2) | 36 |
| 06/03 - 20/03 | | C.1 (10)<br>C.2 (7) | E.1 (3)<br>E.2(10)<br>D (2) | 32 |
| 20/03 - 03/04 | | C.1 (3)<br>C.2 (5) | E.1 (15)<br>D(3) | 26 |
| 03/04 - 17/04 | | C.1 (10)<br>C.2(4) | E.1 (5)<br>E.3 (15) | 34 |
| 17/04 - 01/05 | | C.1 (2)<br>C.3 (5) | E.2 (3) | 10 |
| 01/05 - 15/05 | | C.1 (10)<br>C.3 (15) | E.1 (5)<br>E.3 (5)<br>D(10) | 45 |

| | | | | |
|---|---|---|---|---|
| 15/05 - 29/05 | | C.2 (16) | E.2 (15)<br>E.3 (5) | 36 |
| 29/05 - 12/06 | G (5) | C.2 (3) | E.2 (12)<br>E.3 (10)<br>I.1 (3) | 33 |
| 12/06 - 26/06 | G (30) | C.1 (10) | F (3)<br>I.1 (2) | 45 |
| 26/06 - 09/07 | G (5)<br>H (6) | | F(3)<br>I.2 (5) | 19 |
| TOTAL | | | | 316 |

*Table 5: Final Project Organization with the deviations marked in red*

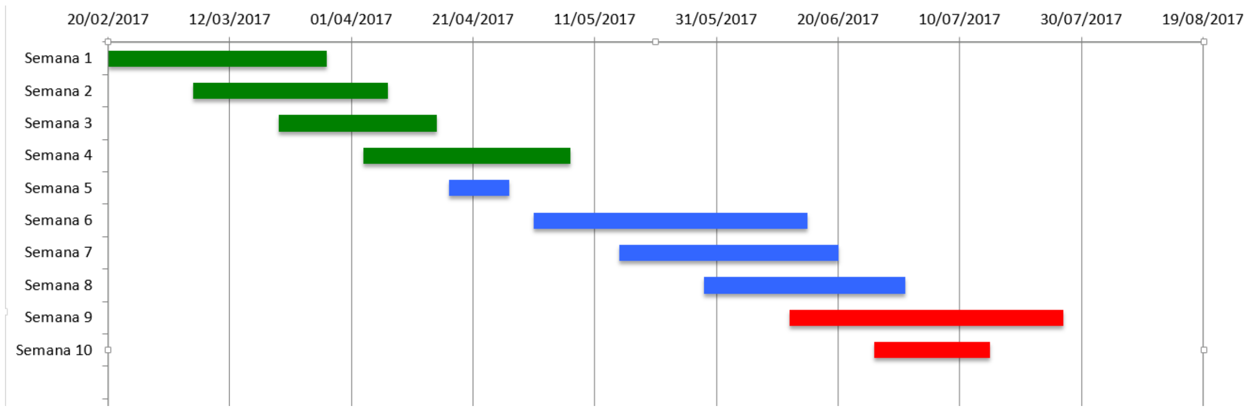


Figure 53: Final project schedule Gantt chart

At the end, around 316 hours has been used for the project, and this is because in the first table there was not taken into account the holidays or the problems that could happen along the project. Because of this, the table needed to be reorganized to match the new setbacks.

About the the problems that happened, for example, one of the unexpected problems were the animations and the problems that were faced with the importation. It past some time until the idea of using another program to be a bridge between two programs in order to import the animations and be able to use them.

# 12. Conclusions

Throughout the project different techniques have been used and learned.

For starters, the technique of motion capture. Although the basic was learned in a practice, the skeleton wasn't exported, and the animations weren't cleaned to make them be ready to be used in Unity 3D, all this has been learned. In addition to that, it has also been learned how to do loops and this is useful for the animations with actions such as walking. This technique is quite important and useful since it is the way in which many studies record their animations, so learning something that is used in the studies is a good advantage.

Another of the useful things learned is the fulfilment of the deadlines. When during the other years a game was created in groups, most of the times the deadlines were not fully accomplished and at the end many parts of the game were leave to do at the last minute. Although this is not a good way to work in a project, for a student group could work due to how many people can be in that group, but when the project is individual, the deadlines need to be fulfilled or the project will be dangerously delayed.

In this project, the deadlines have been tried to comply once the structure of the schedule was remake to include the days that were going to be impossible to work in the project. The ability to work under these deadlines and a schedule is something really important for a labour future because companies cannot allow to have many delays in a project.

Furthermore, one of the strong aspects of the game is the story behind it, being able to adapt a dense mythology to something playable and give a meaningful and attractive story is something that makes a game much more striking.

Another of the points learned is planning, during the project too many things have been tried to develop and sometimes the project has not advanced as much as it was needed because of the saturation of trying to achieve so many things. Therefore, some initial approaches in the game have not been successful, for example, in the game there was going to be a river with a texture that would move depending of the time, in Utgard, the giants would not live in caves, but in constructions made of ice and snow and before entering the Dungeon there would have to fulfill a mission to be able to enter, not to overcome the guard. Additionally, there was going to be a zone with a hail storm, in the final project that was removed, due to a bad results.

## 12.1 Objectives Achieved

The objectives to be achieved were:

**- Development of a scenario with several 3D modelling and animations**

The game has three scenarios with different elements in each one, seven useful objects and the enemies and Valkyries have the animations of the capture motion applied.

**- To show the influence of time in the behaviour of the enemies and in the scenarios.**

The game is influenced by the time so in order to increase the challenge, the enemies' attack will increase with the hour and for the scenarios, there is a moon with a directional light that move influenced by the minutes, so that light will change position each minute and in consequence, it will have effect in the scene.

While the original idea was to create a day/night cycle that idea was separated because the game starts at eight pm, when the moon and the starts are already on the sky so this development would not be as necessary as it was thought in the first place.

However, this does not mean that it cannot be used for futures implementations due to the current implementation of the moon, this actual implementation can be adapted in the future to add a complete day/night cycle in the other worlds that compose the conceptual design of the complete game.

**- Learning the use of capturing movements' methods by using cameras and specialized suits.**

This has been achieved and it also promoted with the need to learn more programs in order to clean the animation and make them useful for unity.

**- The ability to model and adapt a story based on concepts of mythology known in a video game**

All the story created for Jötunheim is based in Nordic mythology and it has been connected to the Valkyrie's game story in order to create a new universe.

## 12.2 Things to do or change in Jötunheim

Unfortunately, not everything has been achieved, mainly due to the great initial workload that was proposed.

- The game was going to have more puzzles in the dungeon to play more with the Ice Physics. The idea is to continue adding challenges for the player.

- Before the player was able to enter the Dungeon, there were going to be a test to get the key, this has been changed in the project but the goal is to change it again once the project goes on.

- More climate effects will be added to improve the simulation of a world full of dangers.

- The problem with low poly is that when the animations are made there are not enough vertex to create a good animation and those vertex overlap. In order to solve this, more vertices will be added to the modelling.

- Improve the models and add them more details because some the result of the giants models are not the ones that were wanted for the game.

## 12.3 Plans for the Future

Since the project began, the plan was to continue development once it was delivered. As it was commented previously, the complete game is in development from the third year in the degree and it was planned to move it to 3D and continue to develop it as the TFG.

However, the idea is to continue working on the game, fixing and adding more details to the level created and developing the remaining worlds. From the first moment there has been so much enthusiasm and desire in this project and it is the intention to finish it in a way that is a good presentable end product.

As can be seen in 'Anexo II', there are nine worlds that make up the complete game and each one has different characteristics and details that can bring many new versions in future approaches. In addition, it is a game with a great weight narrative, with an attractive story that is intended to extend it to allow more options and more mechanics.

Along this project there have been good times and bad times, it has served to grow and know how to face problems and learn useful techniques that can be applicable to video games. But in spite of everything, this will be a project that certainly will not stay end here, but will continue to develop and evolve.

# 13. Bibliography

[1] “Unity 3D” https://unity3d.com/es, Feb 2017
[2] “RPG” https://en.wikipedia.org/wiki/Role-playing_video_game, Feb 2017
[3] “Motion Capture” https://en.wikipedia.org/wiki/Motion_capture, Jun 2017
[4] “Ragnarok” https://en.wikipedia.org/wiki/Ragnar%C3%B6k, Feb 2017
[5] “Valkyrie” https://en.wikipedia.org/wiki/Valkyrie, Jun 2017
[6] “Walpurgis Night” https://en.wikipedia.org/wiki/Walpurgis_Night, Feb 2017
[7] “Walburga” https://en.wikipedia.org/wiki/Saint_Walpurga, Feb 2017
[9] “Yggdrasil” https://en.wikipedia.org/wiki/Yggdrasil , Feb 2017
[10] “Jötunheim” https://en.wikipedia.org/wiki/J%C3%B6tunheimr, Feb 2017
[11] “Niflheim” https://en.wikipedia.org/wiki/Niflheim, Feb 2017
[12] “Svartalfheim” https://en.wikipedia.org/wiki/Svart%C3%A1lfar, Feb 2017
[13] “Golden Sun” https://en.wikipedia.org/wiki/Golden_Sun, Feb 2017
[14] “Utgard” https://en.wikipedia.org/wiki/%C3%9Atgar%C3%B0ar, Feb 2017
[15] “Thrym” https://en.wikipedia.org/wiki/%C3%9Erymr, Feb 2017
[16] “3D Max” http://www.autodesk.es/, Feb 2017
[17] “OptiTrack” http://optitrack.com/, 2017
[18] “Photoshop” http://www.adobe.com/es/products/photoshop.html, Feb 2017
[19] “MotionBuilder” https://www.autodesk.com/products/motionbuilder/overview, Jun 2017
[20] “Visual Studio” https://www.visualstudio.com/, Feb 2017
[21] “Cocos2d-x” http://www.cocos2d-x.org/, Jun 2017
[22] “Fire Emblem” https://en.wikipedia.org/wiki/Fire_Emblem, Jun 2017
[23] “Final Fantasy Tactics” https://en.wikipedia.org/wiki/Final_Fantasy_Tactics, Jun 2017
[24] “The Legend of Zelda: Majora’s Mask”
https://en.wikipedia.org/wiki/The_Legend_of_Zelda:_Majora%27s_Mask, Jun 2017
[25] “The Legend of Zelda: Wind Waker”
https://en.wikipedia.org/wiki/The_Legend_of_Zelda:_The_Wind_Waker, Jun 2017
[26] “The legend of Zelda: Ocarina of Time”
https://en.wikipedia.org/wiki/The_Legend_of_Zelda:_Ocarina_of_Time, Jun 2017

[27] “Toon Shader” https://en.wikipedia.org/wiki/Cel_shading, Jun 2017

[28] “Rime” https://es.wikipedia.org/wiki/Rime_(videojuego), Jun 2017

[29] “Praey for the Gods” http://www.praeyforthegods.com/, Jun 2017

[30] “Alchemical Symbol” https://en.wikipedia.org/wiki/Alchemical_symbol, Jun 2017

[31] "Nordic Mythology" https://en.wikipedia.org/wiki/Norse_mythology, Jun 2017

http://norse-mythology.org/gods-and-creatures/, Jun 2017

[32] "PlayerPrefs" https://docs.unity3d.com/ScriptReference/PlayerPrefs.html, Jun 2017

[33] "Raycasthit" https://docs.unity3d.com/ScriptReference/RaycastHit.html, Jun 2017

[34] "Particle Systems" https://docs.unity3d.com/ScriptReference/ParticleSystem.html, Jun 2017

[35] "Post Processing Stack" https://www.assetstore.unity3d.com/en/#!/content/83912, Jun 2017

[36] "Naughty Dog" https://es.wikipedia.org/wiki/Naughty_Dog, Jun 2017

[37] "The Last of Us" https://es.wikipedia.org/wiki/The_Last_of_Us, Jun 2017

[38] "Mocap The Last of Us" https://www.youtube.com/watch?v=ZdEPZ1VaP6k, Jun 2017

[39] "Uncharted" https://es.wikipedia.org/wiki/Uncharted_(serie), Jun 2017

[40] "Quantic Dream" https://es.wikipedia.org/wiki/Quantic_Dream, Jun 2017

[41] "Beyond Two Souls" https://es.wikipedia.org/wiki/Beyond:_Dos_Almas, Jun 2017

[42] "Mocap Beyong Two Souls" https://www.youtube.com/watch?v=5DwHjNenAmw&has_verified=1, Jun 2017

[43] "Crystal Dynamics" https://es.wikipedia.org/wiki/Crystal_Dynamics, Jun 2017

[44] "Tomb Raider 2013" https://es.wikipedia.org/wiki/Tomb_Raider_(videojuego_de_2013), Jun 2017

[45] "Rise of the Tomb Raider" https://es.wikipedia.org/wiki/Rise_of_the_Tomb_Raider, Jun 2017

[46] "Ubisoft" https://es.wikipedia.org/wiki/Ubisoft, Jun 2017

[47] "Watch Dogs" https://es.wikipedia.org/wiki/Watch_Dogs, Jun 2017

[48] "Mocap Watch Dogs" https://www.youtube.com/watch?v=cVnuFxQNEJk&has_verified=1, Jun 2017

[49] "Assassin's Creed" https://es.wikipedia.org/wiki/Assassin's_Creed, Jun 2017

[50] “Mocap Assasin’s Creed” https://www.youtube.com/watch?v=IkRV7qNVugE, Jun 2017

[51] “Metal Gear Solid” https://es.wikipedia.org/wiki/Metal_Gear_Solid, Jun 2017

[52] “Mocap MGS2” https://www.youtube.com/watch?v=_OXW3l-l8ic, Jun 2017

[53] “Mocap Metal Gear Solid 3” https://www.youtube.com/watch?v=vcZ6l-JSa5g, Jun 2017

[54] “Mixamo” https://www.mixamo.com/, Jun 2017

[55] “Skybox” https://www.assetstore.unity3d.com/en/#!/content/25582, Jun 2017

[56] “Ice Effect Shader” https://www.patreon.com/minionsart, Jun 2017

[57] “Unwrap UVW” https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2017/ENU/3DSMax/files/GUID-EA10E59F-DE7F-497E-B399-6CF213A02C8D-htm.html, Jun 2017

[58] “Trello” https://trello.com/, Jun 2017

# Anexo I: Repartición de tareas

Debido a que estos tres proyectos pertenecen al desarrollo del mismo juego, se han tenido que aplicar metodologías de organización del trabajo en conceptos grupales. El conocimiento de dichas metodologías se ha adquirido a lo largo del grado, muy específicamente la asignatura de Ingeniería del Software (código VJ1224). En dicha asignatura se aprendieron metodologías de desarrollo del software, la existencia de ciertos patrones de diseño y se aplicaron técnicas de organización de grupos de trabajo, con la intención de poder trabajar de forma paralela y uniendo el trabajo de cada uno de los proyectos. Posteriormente, se explicarán las diferentes características que se han utilizado para desarrollar el videojuego.

- **Herramientas utilizadas:** Aquí se especifican todos los dispositivos y herramientas que han sido utilizados para el desarrollo del juego:
    - **Google Drive:** herramienta online para poner en común el proyecto, herramientas o archivos relacionados con el proyecto.
    - **Trello:** herramienta online utilizada para realizar el seguimiento y asignación de las tareas del proyecto
    - **Pen drives:** hardware útil para poner en común los archivos relacionados con el proyecto.
    - **Mandos PS4:** dispositivo I/O para probar la interactividad entre el jugador y el juego.
    - **Sistema de captura de movimiento:** método utilizado para la realización de las animaciones, consta de 8 cámaras, un traje con marcadores para el actor, una torre de Mac y una pantalla de ordenador, junto al programa Motive.

- **Reuniones:** Para coordinar el trabajo, se han hecho reuniones semanalmente en el edificio TI de la Universidad, aprovechando las conversaciones con Víctor Ávila, compositor del proyecto en persona. Además, a partir de mayo se han empezado a desarrollar grupos intensivos de trabajo para unificar y corregir cada parte desarrollada de forma individual en un proyecto base común, desde el que cada uno es capaz de desarrollar su nivel. Finalmente, se han hecho llamadas por Skype con la intención de justificar el trabajo hecho hasta ese momento y definir las nuevas tareas.

- **Métodos empleados:** El método empleado es el sistema de Scrum, basado en una o dos reuniones semanales para explicar cómo va el desarrollo de las tareas y la definición de tareas completadas o en curso. Para indicar las tareas que cada uno tenía que hacer, se ha usado la herramienta Trello, que sirve como pizarra de tareas y que cualquiera de los tres puede editar con la intención de definir el estado de cada tarea. Finalmente, se han hecho cada tres o cuatro semanas reuniones a modo de Sprint, para justificar el estado de las tareas previas y la definición de nuevas tareas para el próximo ciclo.

- **Reparticiones de tareas:** La asignación y seguimiento de las tareas se ha realizado mediante la aplicación Trello. Las tareas se asignaron siguiendo este criterio:

    - Analía Boix se encarga del desarrollo del jugador.
    - Irene Pérez se encarga del desarrollo del Interfaz de Usuario.
    - Ángel Torres se encarga del desarrollo de la Inteligencia Artificial.

- **Comunicación:** Se han utilizado varias vías de comunicación, tanto presenciales como no presenciales. Se han hecho varias reuniones todas las semanas de varios tipos. A modo de Scrum, se han hecho reuniones los miércoles y los jueves por la tarde, tanto presenciales como por Skype. Los sábados, por la mañana, ha habido reuniones presenciales con Víctor Ávila, compositor del juego. Además, a cada cierre de Sprint, ha habido reuniones cada tres o cuatro semanas de tipo Sprint en alguna de los horarios indicados anteriormente. En caso de urgencia, se ha hablado por Whatsapp para intentar solucionar problemas muy concretos.

Distribución de partes del código:

| **Nombre** | **Tarea** | **Módulo** |
|---|---|---|
| Analía Boix Scabbiolo | Cámara | Programación |
| | Sistema de fijado de enemigos | Programación |
| | Movimiento del personaje | Programación |
| | Dialogo | Programación + Arte |
| | Ataque físico Base | Programación |
| | Detección input mando | Programación |
| | Ataque físico Piscis y Virgo | Programación |
| | Alquimia de Piscis y Virgo | Programación |
| | Introducción y menú inicial | Programación + Arte |
| | Modelo base de cuerpo | Arte |
| | Modelado Piscis y Virgo | Arte |
| | Investigación adaptación captura de movimiento al modelado<br>+<br>Limpieza y depuración de animaciones | Arte |

| | | |
|---|---|---|
| | Unión de partes del proyecto y adaptación de los inputs | Programación |
| | Apéndices | Diseño |
| Irene Pérez Collado | Interfaz | Programación + Diseño |
| | Cambio Valquiria | Programación |
| | Salud y Magia de las Valquirias | Programación + Diseño |
| | Salud de los Enemigos | Programación + Diseño |
| | Base Interacción | Programación |
| | Menú de Pausa | Programación |
| | Time Controller | Programación |
| | Base del Ataque Mágico | Programación + Adaptación de Partículas |
| | Alquimia Fuerte de Aries y Tauro | Programación |
| | Ataque Físico de Aries y Tauro | Programación |
| | Inventario | Programación |
| | Objetos del Inventario | Programación + Arte + Modelado |
| | Pisadas | Programación |
| | Limpieza de Animaciones y Depuración | Arte |
| | Base Game Controller | Programación |
| | Apéndices | Diseño |

|  |  |  |
|---|---|---|
| Ángel Torres Parada | Guardado | Programación |
|  | Comportamiento básico del enemigo | Programación |
|  | Gestión de datos entre escenas | Programación |
|  | Ataque físico de Géminis y Libra | Programación |
|  | Alquimia de Géminis y Libra | Programación |
|  | Modelado de Aries y Libra | Arte |
|  | Apéndices | Diseño |
|  | Limpieza de Animaciones y Depuración (individuales) | Arte |

*Tabla 6: La organización de las tareas*

Debugging

| **Tarea** | **Personas implicadas** | **Tipo de debug** |
|---|---|---|
| Fijado enemigos | Analía Boix<br>Ángel Torres | Arreglo de bugs |
| Alquimias de personajes | Analía Boix<br>Irene Pérez<br>Ángel Torres | Ampliación |
| Modelados | Analía Boix<br>Irene Pérez<br>Ángel Torres | Decisión de diseño |
| Interfaz | Analía Boix<br>Irene Pérez<br>Ángel Torres | Integración de otros scripts<br>Arreglo de bugs |
| Magia | Analía Boix<br>Ángel Torres | Adaptación a las stats de las Valquirias |
| Interacción Objetos | Analia Boix<br>Irene Pérez | Mejora de precisión |

| | | |
|---|---|---|
| | Ángel Torres | |
| Interacción dentro del Inventario | Analía Boix<br>Irene Pérez<br>Ángel Torres | Arreglo de errores |
| Interacción con NPC | Analia Boix<br>Irene Pérez | Mejora de precisión |
| Menú de pausa | Ángel Torres | Adición de guardado |
| Time Controller | Irene Pérez<br>Ángel Torres | Reconstrucción de la idea |
| Game Controller | Ángel Torres | Adición del flujo de datos |
| | Analía Boix<br>Ángel Torres | Adición escena Game Over |
| Guardado | Analía Boix<br>Ángel Torres | Adaptación |
| Comportamiento básico del enemigo | Analía Boix<br>Ángel Torres | Adaptación |
| | Analía Boix<br>Irene Pérez<br>Ángel Torres | Bloqueo del movimiento en pausa |
| Bloqueo de control en pausa e interacción | Analía Boix<br>Irene Pérez<br>Ángel Torres | Mejora |

*Tabla 7: El control del debugging*

# Anexo II: Diseño Conceptual del Juego

# Walpurgis Night

Diseñado por Irene Pérez, Analia Boix y Ángel Torres

Para: PC
Edades: A partir de 10 años
Fecha de lanzamiento: Julio 2017 (beta)

Walpurgis Night se trata de un proyecto principalmente desarrollado en grupo durante el tercer curso para las asignaturas: Ingeniería del Software, Diseño Conceptual y Arte del videojuego. Se trata de un proyecto conceptualmente muy desarrollado al que se cogió cariño y se vió una oportunidad perfecta para reinventarlo durante el TFG.

Walpurgis Night es un juego RPG con toques clásicos y de Aventuras, basado en la mitología nórdica, gráficos 3D y en tercera persona. Será un proyecto orientado a PC y consolas de sobremesa. El jugador se verá transportado a diferentes mundos, encarnando el papel de las valquirias, las protagonistas de esta historia. Su misión, salvar a sus compañeras capturadas y estar así un paso más cerca de evitar el fin del universo tal y como se le conoce.

El jugador se sumergirá en un mundo lleno de secretos, misterios y traiciones mientras intenta cumplir su objetivo a contra-reloj. Con grandes influencias de juegos como el Zelda Majora's Mask con el contrarreloj, el Zelda Wind Waker y el uso del toon shading y elementos más simples.

Debido a que es un TFG grupal se desarrollarán tres mundos distintos de este universo. en la tabla siguiente se hará una división de dichos mundos y elementos desarrollados por cada alumno con una asignación de colores que identificarán a cada uno de los componentes del grupo y sus partes en el documento. Todo aquello indicado de color negro corresponde a trabajo común.

| | |
|---|---|
| Irene Pérez Collado | **Jötunheim: mundo de hielo** |
| Analia Boix Scabbiolo | **Svartalfheim: mundo de los elfos oscuros** |
| Ángel Torres Parada | **Niflheim: mundo de la niebla** |

*Tabla A: División de trabajos dependiendo de su color*

A lo largo del documento se mantendrá una explicación de la parte del juego completo para mantener el contexto de la história y el juego en sí, pero, se hará hincapié en las diferentes partes de cada uno.

## Historia y Sumario del juego

Son las ocho de la tarde de 30 de abril de un año desconocido. La cultura popular conoce esa noche como Noche de Walpurgis, o Noche de las Brujas. Se considera que ellas se reunirán esa noche para alargar un año más la agonía de la humanidad. Es por eso que la organización Lifthrasir, encargada de mantener el orden mundial, está interesada en capturar a las doce brujas. La realidad es completamente distinta.

Yggdrasil, el árbol de la vida, y en el que entre sus ramas y raíces se encuentran los nueve mundos, era gobernado por un conjunto de dioses y sus normas. Pero un grupo de héroes de Midgard, el mundo de los hombres, decidieron, cansados de vivir bajo la sombra de los caprichosos dioses, crear una organización para cambiar el orden establecido y poder dictar ellos mismos el destino de la humanidad.

El primer Ragnarök, así se llamó al suceso que dejó a los mundos tal y como se conocían sin dioses, fue activado artificialmente por dicha organización. Sin embargo, no todos los dioses perecieron: la Diosa Walburga, diosa de la magia y alquimia, sobrevivió debido a que era conocedora de los planes de los héroes e intentó evitar, poniendo en combate a doce valquirias conocedoras de las artes alquímicas más básicas, que se completasen, pero no lo consiguió.
Éstas valquirias, que debido a la influencia de los héroes son ahora consideradas las brujas mencionadas anteriormente, y se reúnen cada año para mantener hasta el siguiente la esperanza de la humanidad. Si a medianoche no consiguen reunirse, la organización se saldrá con la suya. En esas cuatro horas, Brunilda, valquiria con la alquimia Géminis infiltrada en la organización, debe liberar a sus compañeras. Para conseguirlo, debe recorrer los nueve reinos en busca de las otras, en una carrera contrarreloj.

Al considerar la organización a Walburga un obstáculo para el nuevo mundo, se disponen a llevar a cabo un segundo Ragnarok, eliminando así a la diosa. Para evitar la intromisión de la diosa, los héroes deciden evitar la reunión de las seguidoras de Walburga, las valquirias, aprisionándolas en los nueve mundos del Yggdrasil, para así evitar que su reunión invoque la protección de Walburga sobre la Tierra.

El jugador va a tener que convertirse en una valiente guerrera experta en armas, una poderosa maga que controle las artes alquímicas y una brillante estratega que sepa mover a sus compañeras a la victoria.

## Historia de Jöttunheim y su nivel relacionado *(desarrollado por Irene Pérez)*

Jöttunheim es el mundo de los gigantes, en él conviven los gigantes y los golems de hielo y piedra, estos últimos, creaciones del Rey de este mundo, Thrym.

En la mitología nórdica, los gigantes vivían constantemente amenazando a los humanos de Midgard y a los dioses de Asgard, de los cuales estaban separados por el río Iving. Esta constante amenaza es debido a que los poderes de los gigantes rivalizaba con el de los dioses y querían derrotarles, además, los dioses y los humanos, por aquel entonces, eran aliados. Por ello, los gigantes eran considerados los 'devoradores' que trataban de crear el caos sobre el orden creado por los dioses.

Los héroes, conocedores de esta rivalidad, usaron esto como ventaja para negociar con ellos en el primer Ragnarok. Los gigantes no estaban interesados en gobernar el mundo, solo demostrar que su poder era superior al de los dioses, por lo tanto aceptaron la propuesta y colaboraron con la destrucción de los dioses.

Ahora que los héroes planean un segundo Ragnarok para acabar con la diosa que logró huir del primero, Walpurga, los gigantes han colaborado en el secuestro de las valquírias, en este caso, secuestrando a Tauro y manteniéndola como prisionera en sus mazmorras.

Aquí es donde empieza el nivel, el jugador deberá controlar a las tres valquirias disponibles, Géminis, Aries y Piscis, para rescatar a Tauro y seguir con el plan de detener el segundo Ragnarok. Cada una de estas valquirias posee unos poderes diferentes que facilitará al jugador el avance en el nivel.

El primer escenario es un bosque montañoso tranquilo, donde el jugador tomará el primer contacto con algún golem para aprender a manejar los controles y familiarizarse con el entorno. Además, encontrará alguna poción y algún reloj por si este enfrentamiento ha sido más largo de lo que era necesario.

En el segundo escenario (Utgard, la fortaleza de los gigantes) el jugador podrá interaccionar con los NPCs que allí se encuentran para aprender algo más de la historia del juego y de ese mundo, además, deberá enfrentarse al guardián de la puerta para poder entrar a la mazmorra y encontrar a su compañera Tauro. En esta escena estará granizando, lo que dañará al personaje, pero no al enemigo al que hay que enfrentarse, suponiendo esto una desventaja para el jugador.

En el tercer escenario, nos encontramos ya dentro de la mazmorra. Esta mazmorra es una combinación de hielo y nieve por lo que el jugador se verá afectado por el deslizamiento y las fuerzas que el hielo aplicará sobre las valquirias. Deberá superar un puente de hielo, al que le afectará el movimiento que la valquiria realice, una inclinación de hielo y la habitación donde está Tauro atrapada. Todo esto mientras se enfrenta a los golems que le impedirán seguir hacia adelante y al Rey Thrym, que hará todo lo posible para evitar que Tauro sea rescatada.

## Historia de Niflheim *(desarrollado por Ángel Torres)*

Niflheim es el mundo de la niebla, un mundo inhóspito en el que todo aquel que entra muere, o al menos es lo que se cuenta…

El mundo de la niebla se encuentra en las raíces del Yggdrasil y fue de los primeros territorios que fueron creados alrededor del Yggdrasil. Dichas raíces están siendo continuamente devoradas por Nidhogg, el dragón que regenta Niflheim.

Previamente, Lifthrasir estaba preocupado por el descontrol que podría causar Nidhogg en el Yggdrasil, por lo que decidieron destruir al dragón antes de comenzar el Primer Ragnarök. No obstante, Nidhogg llegó a enterarse de los planes de la organización y decide buscar un pacto con ellos. Nidhogg sobreviviría, pero su poder se vería reducido al salir de la niebla, controlando así el crecimiento de las raíces del árbol. Además, para ganarse la clemencia de la organización, Nidhogg pacta trabajar para Lifthrasir si es necesario. Después del primer Ragnarök, Nidhogg controla Niflheim para Lifthrasir y evitaría la posible intromisión de posibles enemigos en Niflheim.

Mist, la valquiria con la alquimia de Libra, controla los niveles de niebla que circula en Niflheim, nieblas que circulan descontroladas por el mundo. El control de la niebla perjudica a Nidhogg, ya que al reducir la niebla, su poder disminuye; además el pacto con Lifthrasir obligaba a tomar cartas en el asunto sobre Mist. Y consigue secuestrarla con el poder de la niebla en el mundo de la niebla.

Es este el momento en el que comienza el nivel, el jugador controla a Brunilda, Gudr y Prudr para rescatar a Mist y evitar así la ejecución del Segundo Ragnarök. Con sus diferentes poderes, podrán atravesar los mundos, salvando así al resto de valquirias.

Sin embargo, cuando llegan a Niflheim, se dan cuenta de que la niebla es extremadamente densa y que es imposible atravesar Niflheim sin poder ver a causa de la espesa niebla. Esto se debe a que Nidhogg hace uso de la alquimia de Libra para poder sublimar la tierra de Niflheim y crear una densa niebla. Afortunadamente, la voz de Brunilda dicta que Lifthrasir usa Lámparas de Vacío, un mecanismo mágico que les permite viajar a través del mundo de la niebla, disipándola. Resulta que una de esas lámparas debe encontrarse en las instalaciones de la organización.

Brunilda entonces intenta buscarla, aunque sin ningún permiso para acceder a las salas especiales de las instalaciones. Tendrá que luchar contra soldados que no le dejarán continuar su búsqueda o distraerlos, así que debe tener cuidado y evitar recibir daño. La alquimia de Géminis le presta cierta ayuda, permitiéndole crear clones de aquello en lo que esté pensando, incluso Sigfried, uno de los héroes que pertenecen a la organización y con quien tiene una relación más profunda con ella. La fijación de Géminis le permitirá crear copias de Sigfried que le permitirán interactuar con otros soldados o incluso atacar, usando una gran cantidad de maná. Mientras Brunilda sea cuidadosa, encontrará fácilmente la lámpara.

Una vez se encuentra la lámpara, la fijación de géminis permitirá enviar una réplica de la lámpara, funcional como la original. La niebla casi desaparecerá y Niflheim será explorable una vez más. Cuando las valquirias llegan al puente de Niflheim, encontrarán a Mist desmayada. Cuando se levanta, una magia desconocida bloquea el puente. Se asume que esto se debe a la magia que regenta Niflheim, aquella que controla Nidhogg. Si quieren salir del mundo de la niebla, deben derrotar a Nidhogg, consiguiendo así eliminar la magia.

## Historia de Svartálfaheim ***(Desarrollado por Analía Boix)***

Svartálfaheim, antiguamente conocido como Nidavelir el gran reino de los enanos. Estos últimos eran conocidos por sus habilidades como herreros y por vivir bajo tierra. Vivían bajo el reinado del gran Hreidmar, un rey justo y muy querido por la gente del mundo. Hreidmar trabajaba como uno más de sus súbditos, preocupado por el estado del reino y de sus habitantes. Todos los enanos trabajaban como hermanos en las forjas y minas, manteniendo viva la leyenda de sus grandes capacidades de trabajar con metales y crear verdaderas maravillas... Hasta que sucedió el primer Ragnarok, donde una gran cantidad de energía mágica fue liberada, y una parte afectó a Hreidmar. Su alma, al no poder contener ni procesar tanta magia acabó corrompiéndose, pero convirtiéndose en una versión mejorada de un enano, y siendo así el primer elfo oscuro de la historia. Bajo el nombre de Ivaldi, se volvió más poderoso y al sentirse superior decidió que los enanos merecedores de su confianza y los más fuertes, serían merecedores de poder, y así creó su ejército de elfos oscuros. Los enanos que no fueron aceptados, se vieron esclavizados y obligados a trabajar sin pausa y los que decidieron rebelarse fueron congelados en el tiempo, y de esta manera logró instaurar un reinado de miedo.
Por decisión de Ivaldi el nombre del mundo dejó de ser Nidavelir para convertirse en Svartálfaheim, el rey ya no se preocupa por su gente, ni por el estado de su reino, simplemente en crear armas y armaduras cada vez más poderosas con la ayuda de la magia. El mundo antes brillante y conocido por sus grandes pasillos de oro, luces reflejadas en piedras preciosas y construcciones impresionantes dejó de ser lo que era, la oscuridad predomina en este momento, y todo ha sido destruido y/o ha ido deteriorándose con el paso del tiempo. Brunilda, Prudr y Gudr tendrán que buscar a su compañera Skuld que se encuentra capturada en el mundo, en el interior de la gran forja, intentando no perder una gran parte del valioso tiempo que les queda, y liberar también al mundo de la oscuridad que lo consume.

El jugador tendrá que moverse por las diferentes zonas del mundo haciendo uso del sigilo y interaccionando con los NPCs, buscando aquellos objetos que le permitan continuar su camino hasta encontrar a Virgo. El jugador tendrá que hacer uso de la luz a su favor y de las habilidades mágicas de cada valquiria, por ejemplo en una parte del escenario tendrá que localizar 3 cristales para poder abrirse camino.

## Personajes y controles

**PERSONAJES:**

- **Brunilda - Géminis:** Valquiria capaz de crear cualquier cosa según le venga en mente. Es así como crea un réplica de ella misma mientras se encuentra dentro de la organización Lifthrasir. La falsa Brunilda no tiene dicha habilidad ni ataques mágicos, su única opción de defensa es una daga que porta. Tendrá que rescatar al resto de valquirias sin levantar sospechas. Para utilizar la magia y así poder eliminar a los enemigos que se presenten recibirá la ayuda de Gudr y Prudr.

- **Gudr - Aries:** valquiria con el poder de calcinación. Conoce los planes de Brunilda y se unirá a ella para rescatar a las 9 valquirias restantes. Le encanta la guerra, lo que le hace extremadamente peligrosa. Su arma de elección es un hacha.

- **Prudr - Piscis:** Valquiria con el poder de la proyección, lo cual mejora las cualidades de cualquier sustancia. Gracias a ello, y por haber sido sierva directa de Odín en el pasado, conoce cómo convertir una simple rama en la legendaria lanza Gungnir, por ello su arma es una lanza. Muy espiritual, puede ver o sentir cosas que las demás no.

- Skuld - Virgo ***(Desarrollado por Analía Boix)*****:** Valquiria con el poder de la destilación, asociado a la habilidad de sustracción de magia/alma de diferentes objetos y/o entidades. Es una de las valquirias con más poder mágico, y para una correcta canalización de este porta un cetro. Está encerrada en las profundidades de la gran forja de Svartálfaheim.

- Herja - Tauro ***(Desarrollado por Irene Pérez)*****:** Valquiria con el poder de la congelación, asociado a la modificación de la materia. Es un personaje que destaca más en el ataque físico y en su velocidad, pero a causa de eso, su defensa y su poder mágico es bastante inferior al de sus compañeras. Se encuentra atrapada en Jöttunheim.

- Mist - Libra ***(diseñada por Ángel Torres)*****:** Valquiria con la alquimia basada en la sublimación de sustancias, asociado a la habilidad de convertir gas en sólido o sólido en gas. Suele viajar a Niflheim para controlar la cantidad de niebla del mundo y así entrenar sus poderes. Suele defenderse gracias a sus escudos convirtiéndola en alguien difícil de vencer aunque de carácter poco ofensivo. Tras uno de sus viajes a Niflheim, es raptada por un poder desconocido.

## Mundos

Existen nueve mazmorras, coincidiendo con los reinos que existen en la mitología nórdica entorno al Yggdrasil:

- **Midgard:** El reino de los humanos, se caracteriza por ser un lugar apacible en el que los hombres viven con indiferencia los caprichos de los dioses.
- ***Jöttunheim* *(Desarrollado por Irene Pérez)*:** El reino de los gigantes de hielo, un lugar gélido en el que sólo los seres más resistentes pueden sobrevivir.
- **Helheim:** El reino de los muertos, un lugar tabú para cualquier ser vivo. Puede ser comparable con el infierno que todos conocen.
- ***Niflheim* *(Desarrollada por Ángel Torres)*:** El reino de la niebla, un reino vacío, indómito. Allí no ha llegado a pisar ningún ser vivo, salvo Mist, quien controla los niveles de niebla del mundo.
- ***Svartálfaheim* *(Desarrollado por Analía Boix)*:** El reino de los elfos oscuros, una raza de elfos bastante cerrada, pero que cuidan de los hombres, aunque ahora se sienten corrompidos.
- **Muspelheim:** El reino del fuego, gobernado por los gigantes de fuego. Al igual que Jöttunheim, resulta imposible para un ser humano sobrevivir allí.
- **Vanaheim:** El reino de la naturaleza y hogar de los Vanir, deidades relacionadas con la fertilidad y la sabiduría; desde el Ragnarok, corrupto.
- **Alfheim:** El reino de los elfos de luz, un reino pulcro lleno de seres orgullosos y poderosos. Es lo más parecido al paraíso.
- **Asgard:** El reino de los Dioses, una área aurea llena de lujo en la que los dioses disfrutaban de sus vidas eternas… hasta que sucedió aquel trágico evento.

## Gameplay

El jugador irá reclutando valquirias con poderes especiales que deberán ser utilizados en las mazmorras para poder avanzar. Se comenzará manejando a Brunilda, valquiria con el poder Géminis (crear cualquier cosa que haya memorizado), en Midgard, la tierra de los humanos y la primera mazmorra. Su objetivo es ir recorriendo los nueve reinos para rescatar a sus compañeras. Deberá ir luchando con los enemigos e ir ordenando al resto de valquirias para eliminar al jefe que corrompe cada reino.

Tras liberar a las nueve brujas, el jugador debe volver a Midgard para reencontrarse con el resto de valquirias. Esto debe suceder justo antes de medianoche ya que si el jugador tarda más de la cuenta, todas las brujas no se podrán reunir y la organización llevará a cabo el Ragnarok.

En cuanto al tiempo, es un elemento muy importante del juego. Brunilda tiene cuatro horas para rescatar a sus compañeras. Esto llevará al jugador a elegir sabiamente qué hacer para no perder demasiado tiempo en las mazmorras. El tiempo pasará a velocidades diferentes en los mundos, los más cercanos a Midgard, tienen un tiempo más similar al tiempo real, cuanto más alejado se esté más rápido pasará. Siendo un recurso importante a la hora de la creación de estrategias sobre cómo se afrontará el reto del juego.

Todas las valquirias podrán interactuar con elementos específicos del escenario y con los personajes no controlables, entrar en combate con los enemigos y utilizar/almacenar objetos. El jugador será capaz de intercambiar entre las valquirias para poder utilizar convenientemente sus habilidades, que variarán dependiendo de la valquiria que sea.
Las habilidades se activan con R2 y/o triángulo, dependiendo del momento del uso y utilidad y son:

- **Calcinación:** habilidad ofensiva en su totalidad, proyecta gran cantidad de energía a una temperatura extremadamente alta. Hace daño a los enemigos y con determinados objetos los hace arder.

- **Fijación:** alquimia dedicada a crear objetos a partir del pensamiento. Se pueden invocar objetos y seres que han quedado recordados por Brunilda como pueden ser héroes que fueron afines a ella en el pasado, como por ejemplo Sigfried, con la intención de negociar con otros enemigos.

- **Proyección:** habilidad encargada de mejorar las substancias. Un ejemplo conocido es mejorar el metal en oro. Por tanto puede mejorar objetos. La utilidad más característica en este caso es mejorar la lanza que lleva como arma básica y convertirla en la famosa Gungnir, lanza del dios Odín.

- ***Destilación* *(Desarrollado por Analía Boix):*** esta habilidad, tienen una función especial con objetos que han sido imbuidos mágicamente, siendo capaz de sacar la magia de su interior y dejar el objeto en su estado básico hasta que se vuelva a recargar o se recargue solo. Frente a enemigos es capaz de sacar su alma y dejarlos indefensos durante unos segundos, durante este tiempo recibirán un bonus de daño si se les ataca.

- ***Sublimación*** *(desarrollado por Ángel Torres):* capacidad de convertir en sólido un gas y viceversa. Puede ser muy útil para convertir la niebla en hielo o el hielo en vapor. El estado de sublimación de un material es temporal y a vuelve a su estado original.

- ***Congelación*** ***(Desarrollado por Irene Pérez)*****:** como su nombre indica, congela, se puede gastar tanto para atacar como para superar obstáculos. Si este ataque es realizado sobre algún enemigo, éste se congela durante unos segundos, impidiéndole así realizar cualquier acción hasta que se pase este efecto.

Estas habilidades como se ha comentado anteriormente son exclusivas de cada valquiria y tendrán que utilizarse tanto para la resolución de puzles como combate. Además de las armas.

## Modos y mecánicas

**ARMAS:**

- Brunilda - Géminis: daga. Cadencia baja, rango corto y daño muy alto.
- Gudr - Aries: hacha. Cadencia media-alta, rango medio, daño medio/alto.
- Prudr - Piscis: lanza. Cadencia alta, rango alto, daño medio.
- Skuld - Virgo: cetro. Cadencia baja, rango largo alcance, daño medio/bajo.
- Mist - Libra: dos escudos. Cadencia alta, rango bajo, daño bajo/medio.
- Herja - Tauro: puños. Cadencia alta, rango muy bajo, daño alto.

Hay dos tipos de habilidades mágicas y/o alquímicas que las valquirias pueden utilizar, estos dos tipos son la magia básica o débil y la magia especial o fuerte.

**MAGIA BÁSICA:**

Esta magia está asociada al elemento del signo del zodíaco de cada valquiria (fuego, tierra, aire y agua), menos en el caso de Piscis que al ser la hija de Thor su magia cambia para favorecer al daño elemental de rayo.

- Brunilda Falsa - Géminis: tiene maná pero no habilidades mágicas.
- Brunilda Verdadera - Géminis: ataque con bola de aire.
- Gudr - Aries: ataque con llama de fuego.
- Prudr - Piscis: ataque con rayo.
- Skuld - Virgo: ataque con proyectil de tierra.
- Mist - Libra: ataque con bola de aire.
- Herja - Tauro: ataque con proyectil de tierra..

**MAGIA ESPECIAL:**

Esta magia depende del proceso alquímico asociado a cada signo del zodíaco,

- Brunilda Falsa - Géminis: tiene maná pero no habilidades mágicas.
- Brunilda Verdadera - Géminis: Fijación. Invocar a Sigfried.
- Gudr - Aries: Calcinación. De largo alcance, mucho más potente que el básico. Gasta 20 puntos de maná.

- Prudr - Piscis: Proyección. Invocar a Gugnir. Se necesita el 51% del maná para activarla. Daño de área, golpea fuertemente el suelo delante de ella y afecta a todos los enemigos cercanos. Cadencia baja, rango muy alto, daño muy alto.
- Skuld - Virgo: Destilación. Doble de daño sobre el enemigo afectado durante 10 segundos. Utiliza 45 puntos de maná.
- Mist - Libra: Sublimación. Convierte la niebla en hielo y el hielo en vapor. Ataca con granizo generado por la sublimación del propio aire. Gasta 50 puntos de maná.
- Herja - Tauro: Congelación. Congelar al enemigo durante 15 segundos. Gasta 20 de mana

**ORBES DE ALQUIMIA:** Son cristales con poderes especiales que pueden ser utilizados para mejorar la defensa, crear escudos, no morir en caso de que se pierda toda la vida o incluso recuperarla cuando sea necesaria.

- **Orbe Gules:** Aumenta la fortaleza de la bruja, reduciendo a la mitad el daño que pueda recibir.
- **Orbe Azur:** Recupera vida, +25 puntos de vida por cada orbe.
- **Orbe Sinople:** Crea un campo de protección alrededor de la valquiria. Sin recibir golpes puede durar hasta 2 minutos.
- **Orbe Sable:** Revive a la valquiria si pierde toda la vida.

**RELOJ:** cuando se utiliza retrasa una hora el tiempo del reloj del juego. Dando más tiempo al jugador. Uno por nivel.

**POCIÓN DE MANÁ:** Recupera 10 de maná una vez se consume.

**POCIÓN DE SALUD:** Recupera 15 puntos de salud una vez se consume.

FAROLILLO MÁGICO ***(Desarrollado por Analía Boix)*****:** Función simple, iluminar un área de la escena. Función mágica, el uso variará dependiendo de la valquiria:
- Verde mar claro: congela elfos oscuros, puede hacer aparecer pistas en puntos concretos del escenario
- Rojo: convierte en piedra a los enanos enemigos, da pistas falsas en el laberinto
- Naranja: Devuelve al tiempo actual a los enanos congelados en el tiempo
- Azul: quita la protección mágica

LÁMPARA DE VACÍO ***(desarrollado por Ángel Torres)*****:** Elimina el maleficio de la Niebla Eterna. La niebla de Niflheim deja de tener efecto de desorientación que hace volver a la plataforma de inicio.

CAMPANA ***(desarrollado por Ángel Torres):*** Al lanzarse, causa mucho ruido que puede despistar a los enemigos.

MALEFICIOS ***(desarrollado por Ángel Torres):*** El jugador puede sufrir diferentes tipos de maleficios si no resuelve correctamente la mazmorra. Estos pueden variar desde ver la pantalla en blanco hasta tener los controles cambiados hasta que no se resuelva el conflicto infligido (en un futuro se explicarán cómo resolver cada uno de ellos.

- **Niebla Eterna:** Una espesa niebla inunda el terreno. Brunilda necesita encontrar una Lámpara de Vacío.

## Normas

- No se puede usar más de una valquiria a la vez.

- No se puede usar más de un objeto a la vez.

- No se puede saltar.

- Solo puedes usar la magia mientras se tenga suficiente maná para ejecutar un movimiento mágico en la barra de magia.

- Solo puedes interactuar con el entorno con magias o con objetos específicamente preparados para la interactividad.

- Se puede usar magia con cualquier valquiria excepto con la réplica de Brunilda.

- Hay dos formas de conseguir maná, esperando que se reponga a lo largo de la partida o usando Orbes de Maná.

- El jugador morirá cuando caiga al agua o al vacío.

- El jugador no perderá los objetos obtenidos hasta que se usen o muera.

- Los enemigos pierden vida al ser atacados por el jugador salvo que estén defendiéndose.

- Los enemigos morirán cuando su vida llegue a 0.

- El jugador perderá vida al ser atacado por un enemigo salvo que esté defendiéndose.

- El jugador morirá cuando su vida llegue a 0 excepto si tiene un Orbe de Resurrección en su inventario.

- El jugador solo puede atacar con armas o con magia.

- El jugador debe completar los puzles presentes en el nivel para poder avanzar.

- Los enemigos te persiguen si estás en su rango de visión.

- Los enemigos mirarán alrededor si escuchan algún ruido.

- Los enemigos dejaran de perseguirte si te alejas lo suficiente de ellos.

- Puedes cambiar de valquiria cuando quieras.

- Solo se puede seleccionar una valquiria de las que el juego dispone en principio o se ha conseguido reclutar.

- El arma y habilidad correspondiente a cada valquiria tienen estadísticas de ataque y rangos diferentes.

- Solo se puede interactuar con ciertos elementos del entorno si se usa el tipo de magia adecuada.

- Reclutar a la nueva valquiria cuando la encuentras es obligatorio para poder continuar el nivel.

- Si el reloj llega a las 00:00 se acaba el juego.

- El jugador puede alterar el paso del tiempo mediante objetos.

- El jugador no podrá volver a una hora anterior a las 20:00.

- ***(Desarrollado por Ángel Torres)*** En Niflheim, el jugador no podrá continuar el nivel hasta que se consiga la Lámpara de Vacío.

- ***(Desarrollado por Ángel Torres)*** El jugador no podrá volver al terreno inicial de Niflheim si no es reiniciando desde el último punto de guardado o venciendo a Nidhogg, una vez se rescate a Mist.

- ***(Desarrollado por Ángel Torres)*** Sin llaves, el jugador no podrá entrar en las áreas cerradas con llave.

- ***(Desarrollado por Analía Boix)*** Sí a la valquiria no le queda maná la habilidad especial del farolillo no podrá activarse.

- ***(Desarrollado por Analía Boix)*** El jugador no podrá atacar ni defenderse mientras esté activo el farolillo.

- ***(Desarrollado por Analía Boix)*** Los enemigos se verán atraídos por la luz

- ***(Desarrollado por Irene Pérez)*** Si el personaje ha sido congelado, no podrá moverse hasta que se pase este efecto.

- ***(Desarrollado por Irene Pérez)*** Si el personaje está sobre hielo, su movimiento se verá un poco restringido.

## Enemigos y bosses

**Enemigos:** Enemigos débiles y no muy complicados de eliminar que aparecerán a lo largo de la aventura repetidamente. Son soldados de Lifthrasir, con la misión de evitar que el jugador cumpla su misión de reunir a todas las valquirias.

**Bosses:** al final de cada mundo/mazmorra, las valquirias deberán enfrentarse a un jefe de sala para poder salvar a sus compañeras.

- **Jörmundgander:** conocida también como la serpiente de Midgard, será el primero boss del juego. Puede rodear la tierra con su escamosos cuerpo debido a lo gigantesca que puede llegar a ser.

- **Hel:** encargada de custodiar Helheim, ya que es la reina. Es un caso especial del primer ragnarok. Su apariencia representa el cómo ve la gente a la muerte.

- Ivaldi ***(Desarrollado por Analía Boix)*:** El primer elfo oscuro, era anteriormente el rey enano Hreidmar hasta que fue corrompido por la energía liberada por el primer Ragnarok. Cruel y ávido de poder reina con mano dura lo que ahora considera que es su mundo. La luz verde marl le congela temporalmente y la azul permite que sea atacado tanto física como mágicamente. Es el boss de la mazmorra de Svartalfaheim.

- **Thrym *(Desarrollado por Irene Pérez)*:** rey de Jöttunheim, se le concedió este título al ayudar en el primer ragnarök. Siempre odió a los dioses y es un gigante muy ambicioso. Controla a los golems de piedra y hielo ya que son creaciones suyas, es un enemigo con un gran ataque físico que puede realizar un gran daño, sin embargo es poco resistente al fuego debido a su naturaleza. Su odio hacia los dioses y hacia aquellos que trabajan para ellos viene desde que le arrebató a Thor su martillo, extorsionando con ello a los dioses para que le entregasen a la diosa Freyja como esposa, pero Thor le engañó y le derrotó.

- Nidhogg *(Diseñado por Ángel Torres)*: dragón que vive en Niflheim, después del primer ragnarök se ha dedicado a atormentar a las almas humanas presentes en Niflheim y a la vez mantener segura la mazmorra. Pactó con Lifthrasir antes del primer Ragnarök sumisión a la organización a cambio de permitirle sobrevivir. Tras el primer Ragnarök, empezó a controlar la niebla de Niflheim, aunque siempre encontró a alguien que contrarrestaba su poder gracias a la alquimia de sublimación, la valquiria Mist.

- **Hafgufa:** monstruo marino de gran tamaño, que recuerda a una isla por las rocas que cubren su espalda.

- **Njördr:** en su día fue un dios vanir, habitante de Vanaheim, pero durante una guerra del pasado entre dioses fue tratado injustamente y se le quitó su poder. Después del primer ragnarök se le devolvieron estos a cambio de trabajar para Lifthrasir y ahora es el protector de la mazmorra que se encuentra en Vanaheim.

- **Dullahan:** después del primer ragnarök corrompió el mundo de los elfos de la luz conocido como Alfheim, se dedica a recorrer el territorio en la noche eterna con su cabeza como linterna en la mano.

- **Fenrir:** lobo de grandes dimensiones y brutal fuerza. Es el protector de Asgard desde el primer ragnarok. Será el boss final en esta localización, siendo uno de los enemigos más duros del juego.

- **Surtr:** gigante habitante de Muspelheim, es un jötunn portador de una inmensa espada de fuego y es el encargado de defender la mazmorra de dicho lugar. Con un papel importante también durante el primer Ragnarok, ya que no teme a los dioses y sus poderes.

- **Völundr:** maestro herrero y artesano, que a causa de sus malas acciones se ve corrompido por esta razón pierde las piernas, se construye sus propias alas. Es uno de los capitanes de la organización. Y será el enemigo al llegar a las 24:00.

## Retos en el Juego

Nuestro objetivo a la hora de desarrollar el gameplay del juego era conseguir que el jugador tuviera que emplear tanto sus habilidades físicas como mentales para ir superando los retos a los que se enfrentaría durante el juego; por ejemplo resolviendo los puzzles de las distintas mazmorras que se iría encontrando a lo largo del juego.

En cuanto a la jerarquía de los retos: el juego está organizado en un objetivo principal, el cual se subdividirá en 10 niveles o mazmorras para liberar a las 9 valquirias restantes y así evitar el Ragnarok. Cada nivel tendrá un jefe de mazmorra. Para llegar a él será necesario eliminar a ciertos enemigos y superar una serie de puzzles que llevarán al jugador a la sala final donde este se encuentra. El objetivo principal de cada nivel es rescatar a la valquiria capturada, el enfrentamiento final no es obligatorio si el jugador descubre cómo evitarlo.

Como misión adicional en cada mazmorra dependiendo del mundo podrá haber NPCs que ayuden al jugador a comprender los secretos e historias de estos.

Uno de los puntos que se quiere tocar es dar al jugador la opción de superar el final del juego de diferentes maneras, ya que tendrá que plantearse una estrategia eligiendo tres valquirias entre las que tenga disponibles. A pesar de que podrá cambiarlas en determinados puntos llamados tótems rúnicos.

El jugador, además, podrá elegir completar algunos retos o no siendo estos tanto implícitos como explícitos. Por ejemplo, conocer todos los secretos de los mundos, superar los maleficios…

## Recursos y Entidades en el Juego

Los <u>recursos</u> que aparecen en el juego son los siguientes:

- **Tiempo:** Recurso intangible que puede ir en beneficio o detrimento del jugador. Este va avanzando continuamente durante la partida. Al principio del juego, serán las 20:00 horas. Si el juego alcanza medianoche, el jugador perderá el juego. El jugador puede encontrar elementos que lo alteren. Dependiendo de la mazmorra, el tiempo avanzará con una velocidad u otra.

- **Maná:** Recurso intangible que las valquirias utilizan para poder usar la magia limitadamente. Los ataques mágicos necesitan maná para poder ser ejecutados. La cantidad será visible para el jugador en cualquier momento.

- **Vida:** Recurso intangible. Las valquirias poseen un porcentaje de vida que podrá ir disminuyendo conforme se enfrente a enemigos o aumentando si se curan los daños.

Las <u>entidades</u> son las siguientes:

- **Valquirias:** Son entidades del juego que el jugador puede controlar. A medida que avanza el juego se irán desbloqueando más valquirias, cada una tiene unas características propias.

- **Ataques Mágicos:** Entidades de combate. Cada valquiria tiene un poder mágico diferente al resto dependiendo de su símbolo del zodíaco y su habilidad alquímica. Se pueden usar para hacer daño a un enemigo, o para activar una mecánica sobre un objeto.

- **Orbes de Alquimia:** Entidades potenciadoras. Son cristales con poderes especiales que pueden ser utilizados para mejorar la defensa, crear escudos, no morir en caso de que se pierda toda la vida o incluso recuperarla cuando sea necesaria.

- **Pociones:** Entidades regeneradoras, recuperan un porcentaje de maná o vida al ser utilizadas.

- **Relojes:** Son entidades encargadas de beneficiar al jugador, una vez activado se restará tiempo del reloj del juego.

- **Lámparas de vacío:** Entidad única, se utiliza para disolver el maleficio de la niebla eterna.

- **Farolillo mágico:** Entidad única, emite luz a la cual se le pueden aplicar poderes mágicos que le dan diferentes efectos.

- **Maleficios:** Entidad que utilizan los diferentes jefes de mazmorra como castigo tras llegar a una conclusión errónea tras la conversación crítica al vencerlo. Puede dificultar gravemente el avance del jugador.

- **Enemigos:** Entidad, son los enemigos dentro del juego, tanto jefes finales como enemigos normales. Hay de varios tipos y dependiendo del tipo de enemigo, tendrán patrones de combate distintos y fuerza de ataque y defensa que varían.

- **NPCs:** Entidad, personajes no jugables dentro del juego. Aportan información sobre el nivel.

De estas entidades aquí descritas, hay cinco entidades de las que sus elementos son únicos. Estos son las valquirias, los jefes (que son enemigos), lámpara, farolillo y los maleficios.

Como recursos y entidades tangibles, es decir, que se pueden tocar y que son visibles en pantalla son las valquirias, que son las únicas que pueden ser controladas, todos los objetos, los enemigos que se pueden vencer y los NPCs.

Los recursos y entidades intangibles, que no se pueden tocar. Estos son el tiempo, el maná y la vida, que tienen límites, los ataques mágicos que son productos de la mecánica de ataque y los maleficios, que son productos de la mecánica de castigo.

De las entidades que hemos definido, estas tienen atributos que las describen. Las valquirias se ven definidas por diversos atributos como la vida, el maná, el ataque, y la defensa (todas ellas cuantificables) la dirección que sigue y la magia que se ve definida por el signo que representa y es característica de cada una. Los enemigos se ven definidos por la vida, el daño que efectúa y la defensa, que son atributos cuantificables y el estado en el que se encuentra el enemigo.

## Acciones de las mecánicas

Las mecánicas que rigen las relaciones entre las entidades son:

**- Atacar con armas:** es la mecánica para eliminar a los enemigos cuerpo a cuerpo desde el punto de vista del jugador o al jugador desde el punto de vista de los enemigos. Con cada ataque, la víctima del ataque, en caso que no esté defendiéndose pierde puntos de vida. Para decidir cuánta vida pierde la víctima y si el atacante puede hacer daño, se tienen en cuenta la cadencia del arma, el rango de alcance y el valor de ataque de la unidad.

**Usar magia:** con esta mecánica se puede interactuar con el escenario y con los objetos que hay en él o atacar a los enemigos. Para usar la magia es necesario tener una cantidad mínima de maná, que variará según la magia que utiliza cada valquiria o si la magia es básica o especial. El uso de la magia implica la pérdida de maná. La réplica de Brunilda es la única valquiria incapaz de usar magia.

**Defenderse:** es una mecánica disponible tanto para ciertos enemigos como para el jugador. Gracias a la defensa, se pueden bloquear los ataques y evitar recibir daño.

**Moverse:** esta mecánica implica el desplazamiento de la valquiria a través del escenario. Es una parte fundamental del desarrollo del juego, ya que sin movimiento el jugador no puede completar los puzles ni salvar a las otras valquirias ni vencer a los jefes.

**Recoger objetos:** esta mecánica añade al inventario objetos consumibles que se encontrarán dispersos por el nivel. No es necesaria para completarlo, pretenden favorecer y ayudar al jugador en el camino.

**Usar objetos:** mecánica asociada a la anterior, permite al jugador utilizar los objetos recogidos y recibir sus efectos. Pueden ser instantáneos o actuar durante un rango de tiempo. Algunos de estos tienen un uso automático, no dependen de que el jugador los active.

Activar farolillo ***(Desarrollado por Analía Boix)*:** mecánica que permite al jugador  en su uso básico iluminar una pequeña parte de la escena, una vez se le aplica energía mágica su funcionamiento y características cambia dependiendo de quien la utilice.

**Cambiar de valquiria:** la mecánica de cambio de valquiria permite cambiar entre las diferentes valquirias disponibles, es imprescindible para completar el juego. Esto es porque los muchos puzles del juego han de resolverse mediante el uso en concreto de alguna de las habilidades de estas, por tanto cambiar entre ellas es de gran importancia para poder acceder a estas habilidades rápidamente.

**Hablar con los personajes no controlables:** los personajes no enemigos pueden ser consultados por el jugador al encontrarse cerca de ellos. Gracias a esto, el jugador puede recibir información relevante sobre la historia o sobre la continuidad en el juego.

**Interacción con elementos del escenario:** una mecánica básica y de gran importancia para completar los diferentes niveles. Permite al jugador aplicar las habilidades mágicas asociadas a las valquirias para poder completar puzles. También puede ser una interacción básica por ejemplo activar palancas.

**Modo sigilo:** mecánica que permite al jugador poder moverse por el escenario llamando mínimamente la atención de los enemigos ya que hace muy poco ruido. Le permitirá si quiere poder superar diferentes niveles sin necesidad de entrar en combate.

**Mover cámara:** esta mecánica permite al jugador observar el escenario a su alrededor, para poder así planear las mejores estrategias para afrontar los niveles de la mejor manera posible.

**Fijar enemigo:** gracias a esta mecánica, el jugador puede mantener como blanco fijo a un enemigo para poder lanzar ataques físicos y mágicos automáticamente en la dirección del enemigo.

Lanzar objeto (diseñado por Ángel Torres): algunos objetos como el cencerro se pueden lanzar, causando un gran ruido al golpear el suelo.

| ACCIONES DE USUARIO | |
|---|---|
| **Mecánica** | **Características** |
| Moverse | Dirección = ($v_x$, $v_y$, $v_z$)<br>$v_y$ constreñido al plano del suelo. |
| Atacar con arma | Al contacto con el enemigo.<br>Causa $x$ daño.<br>Dependiente de la cadencia del arma.<br>Dependiente del rango de cada arma. |
| Usar magia | Sobre elemento interactivo del escenario:<br>Activa efecto del elemento.<br>Sobre enemigo:<br>Causa $y$ daño.<br>En ambos casos elimina $z$ puntos de maná.<br>$y$ y $z$ son dependientes de cada valquiria. |
| Defenderse | Bloquea daño del enemigo (daño enemigo = 0). |
| Recoger objeto | El objeto desaparece del escenario.<br>El objeto se añade al inventario. |
| Usar objeto | En pociones, orbes (salvo Orbe Sable) y relojes:<br>El objeto desaparece del inventario.<br>Realiza el efecto que se espera del objeto.<br>En Orbes Sable:<br>Se usa automáticamente cuando vida = 0.<br>El objeto desaparece del inventario.<br>Recupera toda la vida de la valquiria.<br>En objetos equipables (lámparas):<br>El objeto aparece como equipado.<br>Realiza el efecto siempre que esté equipado. |
| Cambiar de valquiria | Cambia todas las estadísticas de la valquiria y su arma. |
| Hablar con NPCs | Ejecuta flujo de conversación.<br>Recibe información del NPC. |
| Interacción con el escenario | Activa objetos del escenario. |
| Modo sigilo | Volumen de sonido muy reducido.<br>Velocidad de valquiria muy reducida. |
| Mover cámara | Desplazar orientación de la vista. |
| Fijar cámara | Cámara centrada en un enemigo concreto.<br>Dirección de ataque (físico o mágico) = dirección enemigo. |

*Tabla B: Explicación de las distintas acciones que el usuario puede realizar.*

De las entidades y recursos que hemos descrito anteriormente, prácticamente ninguna aparecerá mediante una fuente. Sólo podremos considerar la vida y el maná como unos recursos que se irán recuperando a medida que pase el tiempo.

Además, Los enemigos finales y los orbes serán los únicos elementos que pueden ser drenados del sistema del juego. Los enemigos pueden ser drenados cuando el jugador lo elimina venciendolo y los orbes se drenan al utilizarlo sobre una valquiria.

En el juego, habrá mecánicas que plantearán retos, como los maleficios. Sin embargo, como los maleficios son mecánicas en sí mismas, se considerará superado el reto que estos plantean el reto será superado.

Como hemos dicho anteriormente, los enemigos y los NPCs tienen mecánicas de interacción, sin embargo son elementos no jugables. Para solucionar esto se implementarán algoritmos que regulen sus movimientos y que activará una secuencia de acciones que irán haciendo avanzar las conversaciones cuando interactúe con las valquirias.

En el apartado de exploración, el personaje avanzará por el escenario, encontrándose enemigos o NPCs con los que interactuar, se encontrará orbes y pociones que se almacenarán en el escenario al ser recogidos por el jugador, será afectado por maleficios o se encontrará runas que le aportarán información para el juego.

En cuanto al combate, será un 1 contra 1, donde el jugador deberá golpear un número de veces X al enemigo para derrotarlo. Para combatir podrá usar tanto armas como magia, permitiéndole de esta forma más combinaciones de ataque.

Las mecánicas en su conjunto y en un entorno compartido funcionan de esta manera:

| ACCIONES DE LOS RECURSOS GLOBALES | | |
|---|---|---|
| **Mecánica** | **Características** | **Acciones** |
| Poción de maná | Aumenta maná al usarse.<br>Se puede encontrar en todos los niveles. | Ocupar espacio en inventario.<br>Desaparecer de escenario.<br>Fortalece magia. |
| Poción de vida | Aumenta vida al usarse.<br>Se puede encontrar en todos los niveles. | Ocupar espacio en inventario.<br>Desaparecer de escenario.<br>Recupera vida. |
| Orbe Sable | Recupera toda la vida si vida = 0.<br>Se puede encontrar en todos los niveles. | Desaparecer de escenario.<br>Recupera toda la vida. |
| Reloj | Retrasa el tiempo una hora.<br>Se puede encontrar en todos los niveles. | Ocupar espacio en inventario.<br>Desaparecer de escenario.<br>Retrasa el tiempo una hora. |
| Orbe Escudo | Crea un escudo alrededor del personaje por un tiempo limitado.<br>Se puede encontrar en todos los niveles. | Ocupar espacio en inventario.<br>Desaparecer de escenario.<br>Crear un escudo protector. |

| | | |
|---|---|---|
| Orbe Magia | Permite usar magia sin gastar maná por un tiempo limitado.<br>Se puede encontrar en todos los niveles. | Ocupar espacio en inventario.<br>Desaparecer de escenario.<br>Permite no gastar maná |

Tabla C: Explicación de las acciones de los recursos del juego.

| ACCIONES DE LOS RECURSOS DE NIFLHEIM (*desarrollado por Ángel Torres)* | | |
|---|---|---|
| **Mecánica** | **Características** | **Acciones** |
| Soldados | Patrullar.<br>Ataques débiles.<br>Se pueden encontrar en Base Enemiga. | Moverse<br>Conversar (si es con Sigfried)<br>Ataque con arma<br>Perder vida<br>Matar<br>Morir |
| Comandantes | Ataques cargados.<br>Se pueden encontrar en Base Enemiga. | Moverse<br>Conversar (si es con Sigfried)<br>Ataque con arma<br>Defenderse<br>Perder vida<br>Matar<br>Morir |
| Sigfried | Conversar.<br>Se puede encontrar en Base Enemiga. | Conversar<br>Moverse |
| Nidhogg | Ataques muy fuertes como dragón..<br>Se puede encontrar en Coliseo de Niflheim. | Moverse (en tres dimensiones)<br>Atacar con las garras<br>Perder vida<br>Matar<br>Morir |
| Mist | Conversar.<br>Se encuentra en puente de Niflheim. | Conversar<br>Añadirse al equipo. |
| Poción de maná | Aumenta maná al usarse.<br>Se pueden encontrar en Niflheim y Base Enemiga. | Ocupar espacio en inventario.<br>Desaparecer de escenario.<br>Fortalece magia. |
| Poción de vida | Aumenta vida al usarse.<br>Se pueden encontrar en Niflheim y Base Enemiga. | Ocupar espacio en inventario.<br>Desaparecer de escenario.<br>Recupera vida. |
| Orbe Sable | Recupera toda la vida si vida = 0.<br>Se puede encontrar en Niflheim. | Desaparecer de escenario.<br>Recupera toda la vida. |
| Reloj | Retrasa el tiempo una hora.<br>Se pueden encontrar en Base Enemiga. | Ocupar espacio en inventario.<br>Desaparecer de escenario.<br>Retrasa el tiempo una hora. |
| Lámpara de vacío | Deshace automáticamente la Niebla Eterna.<br>Se pueden encontrar en Base Enemiga. | Desaparecer de escenario.<br>Deshace la Niebla Eterna (maleficio). |
| Cencerro | Emite sonido al colisionar con algo. | Ocupar espacio en inventario. |

| | | |
|---|---|---|
| | Se pueden encontrar en Base Enemiga. | Desaparecer de escenario.<br>Emitir sonido al ser lanzado. |
| Campana extractora | Absorbe olor de la comida.<br>Se encuentra en Base Enemiga. | Activar/desactivar.<br>Absorber olor. |
| Cocina | Produce olor.<br>Se encuentra en Base Enemiga. | Activar/desactivar.<br>Crear olor. |
| Termostato | Aumenta o disminuye.<br>Se encuentran en Bases Enemigas. | Aumentar/disminuir temperatura. |

*Tabla D: Explicación de las acciones de los recursos de Niflheim.*

| ACCIONES DE LOS RECURSOS DE JÖTTUNHEIM ***(desarrollado por Irene Pérez)*** | | |
|---|---|---|
| **Mecánica** | **Características** | **Acciones** |
| Golem de Piedra | Patrullar.<br>Ataques débiles.<br>Resistencia alta<br>Se pueden encontrar en la Mazmorra Congelada | Moverse<br>Ataque<br>Perder vida<br>Matar<br>Morir |
| Golem de Hielo | Patrullar<br>Ataques que congelan.<br>Resistencia baja.<br>Se pueden encontrar en la Mazmorra Congelada. | Moverse<br>Ataque<br>Defenderse<br>Perder vida<br>Matar<br>Morir |
| Udgard | Gran poder de ataque<br>Gran resistencia, excepto a ataques de fuego<br>Se puede encontrar en la Mazmorra Congelada | Moverse<br>Atacar<br>Perder vida<br>Matar<br>Morir |
| Gigantes | Son NPC básicos que viven en Jöttunheim<br>Conversar.<br>Se encuentra en el poblado antes de la Mazmorra Congelada | Conversar |
| Herja / Tauro | Valquiria atrapada en este mundo.<br>Conservar.<br>Se encuentra en la Mazmorra Congelada. | Conservar<br>Añadir al grupo<br>Una vez en el grupo, usar sus habilidades |
| Poción de maná | Aumenta maná al usarse.<br>Se pueden encontrar en Jöttunheim. Y en la Mazmorra Congelada. | Ocupar espacio en inventario.<br>Desaparecer de escenario.<br>Fortalece magia. |
| Poción de vida | Aumenta vida al usarse.<br>Se pueden encontrar en Jöttunheim. Y en la Mazmorra Congelada. | Ocupar espacio en inventario.<br>Desaparecer de escenario.<br>Recupera vida. |
| Orbe Sable | Recupera toda la vida si vida = 0. | Desaparecer de escenario.<br>Recupera toda la vida. |

| | | |
|---|---|---|
| | Se puede encontrar en Jöttunheim. Y en la Mazmorra Congelada. | |
| Reloj | Retrasa el tiempo una hora.<br>Se pueden encontrar en Jöttunheim. Y en la Mazmorra Congelada. | Ocupar espacio en inventario.<br>Desaparecer de escenario.<br>Retrasa el tiempo una hora. |
| Orbe Escudo | Crea un escudo alrededor del personaje por un tiempo limitado.<br>Se puede encontrar en todos los niveles. | Ocupar espacio en inventario.<br>Desaparecer de escenario.<br>Crear un escudo protector. |
| Orbe Magia | Permite usar magia sin gastar maná por un tiempo limitado.<br>Se puede encontrar en todos los niveles. | Ocupar espacio en inventario.<br>Desaparecer de escenario.<br>Permite no gastar maná |
| Llave de Hielo | Abre la puerta de la Mazmorra Congelada<br>Se consigue derrotando al guardián de la puerta | Ocupar espacio en el inventario.<br>Desaparecer del escenario.<br>Permitir el paso a la mazmorra. |

*Tabla E: Explicación de las acciones de los recursos de Jötunheim.*

| **ACCIONES DE LOS RECURSOS DE SVARTALFHEIM *(Desarrollado por Analía Boix)*** | | |
|---|---|---|
| **Mecánica** | **Características** | **Acciones** |
| Enanos Esclavos | Patrullar<br>Ataques lentos pero fuertes<br>No ven en la oscuridad<br>Llevan un farol<br>Les atrae la luz que no sea roja<br>Les afecta la luz roja<br>Se pueden encontrar en el mundo | Moverse<br>Ataque<br>Perder vida<br>Matar<br>Morir |
| Elfos Oscuros | Patrullar<br>Ataques rápidos y medio fuertes<br>Ven en la oscuridad<br>Les atrae la luz que no sea azul<br>Les afecta la luz azul<br>Se pueden encontrar en el mundo | Moverse<br>Ataque<br>Defenderse<br>Perder vida<br>Matar<br>Morir |
| Enanos | NPC básicos<br>Conversar<br>Se encuentran en determinadas zonas del escenario de la Gran Caverna | Conversar |
| Enanos Congelados en el tiempo | NPC básicos<br>Vuelven al tiempo actual mediante luz naranja<br>Se encuentran en determinadas zonas del escenario de la Gran Caverna | Conversar |
| Farolillo | Efecto normal: ilumina dentro de un rango<br>Efecto mágico: depende de la valquiria | Iluminar<br>Usar habilidades<br>Gastar maná |

| | | |
|---|---|---|
| | • Verde mar claro: congela elfos oscuros, puede hacer aparecer pistas en puntos concretos del escenario<br>• Rojo: convierte en piedra a los enanos enemigos, da pistas falsas en el laberinto<br>• Naranja: Devuelve al tiempo actual a los enanos congelados en el tiempo<br>• Azul: quita la protección mágica | Conseguir pistas<br>Conseguir información |
| Skuld | Valquiria encerrada en el laberinto de la forja del mundo | Conservar<br>Añadir al grupo<br>Una vez en el grupo, usar sus habilidades |
| Poción de maná | Aumenta maná al usarse.<br>Se pueden encontrar en todos los niveles | Ocupar espacio en inventario.<br>Desaparecer de escenario.<br>Fortalece magia. |
| Poción de vida | Aumenta vida al usarse.<br>Se pueden encontrar en todos los niveles | Ocupar espacio en inventario.<br>Desaparecer de escenario.<br>Recupera vida. |
| Orbe Sable | Recupera toda la vida si vida = 0.<br>Se pueden encontrar en todos los niveles | Desaparecer de escenario.<br>Recupera toda la vida. |
| Reloj | Retrasa el tiempo una hora.<br>Se pueden encontrar en todos los niveles | Ocupar espacio en inventario.<br>Desaparecer de escenario.<br>Retrasa el tiempo una hora. |
| Cristales mágicos | 3 Cristales localizados en la Gran Caverna<br>1 lo lleva un elfo oscuro<br>1 lo encuentras sobre una caja<br>1 lo tiene un NPC | Desaparecer del escenario.<br>Permitir el paso a el laberinto |
| Llave del laberinto | Poder acceder a la sala central donde se encuentra Skuld<br>Se obtiene en una sala del laberinto derrotando a los enemigos | Desaparecer del escenario<br>Permitir entrada en la sala central |

*Tabla F: Explicación de las acciones de los recursos de Svartalfalheim.*

Las mecánicas para el resto de niveles serán similares, ya que se mantendrán constantes durante todo el juego y aunque la historia o los objetivos varíen, la forma de realizarlos será la misma.

## Controles

Serán iguales para todas las valquirias, sin contar la diferencia de habilidades. Los controles están pensados para que el jugador tenga un fácil acceso a todas las características del juego. Se jugará con un mando de consola, concretamente un mando de PlayStation 4:

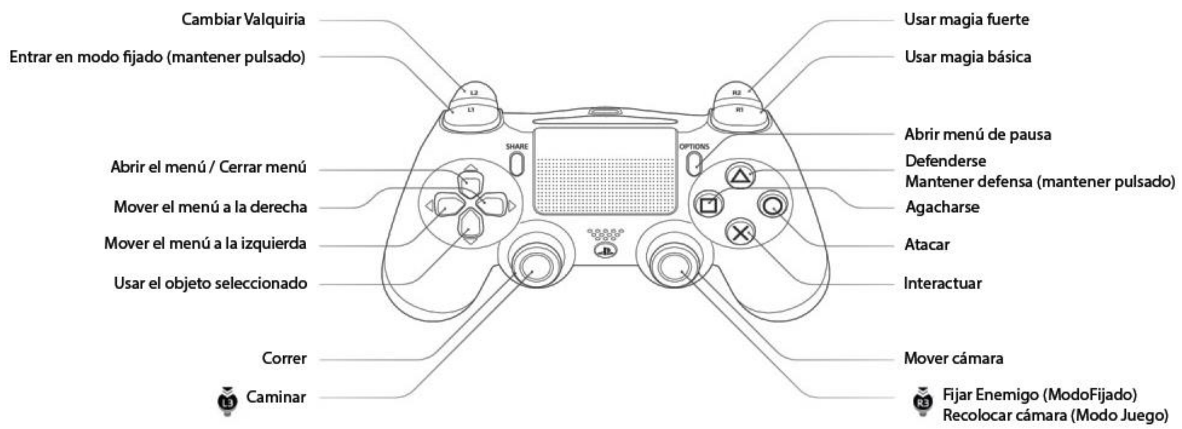

Cambiar Valquiria
Entrar en modo fijado (mantener pulsado)
Abrir el menú / Cerrar menú
Mover el menú a la derecha
Mover el menú a la izquierda
Usar el objeto seleccionado
Correr
Caminar
Usar magia fuerte
Usar magia básica
Abrir menú de pausa
Defenderse
Mantener defensa (mantener pulsado)
Agacharse
Atacar
Interactuar
Mover cámara
Fijar Enemigo (ModoFijado)
Recolocar cámara (Modo Juego)
SHARE
OPTIONS
L2
L1
R2
R1
L3
R3

*Figura 1: Explicación visual de los controles del juego.*

- **Joystick izquierdo**: mover al personaje controlable por el escenario, caminando. Moverse por los diferentes menús.
- **Joystick derecho:** controlar el movimiento de la cámara.
- **R1**: activación de la habilidad mágica básica de la valquiria, asociada al signo del zodiaco correspondiente. Ataque mágico débil.
- **R2:** activación de la habilidad mágica fuerte de la valquiria, única de cada una, asociada al poder alquímico.
- **L2:** sirve para cambiar de valquiria entre las disponibles.
- **Círculo:** ataque físico, dependerá del arma de elección de cada valquiria.
- **Cuadrado:** agacharse, modo sigilo.
- **X:** interacción con NPCs, objetos y/o elementos del escenario.
- **Triángulo:** defenderse.
- **Cruceta:** cruceta superior abre el inventario, las crucetas izquierda y derecha te permite moverte por él y la cruceta inferior te permite usar el objeto seleccionado.
- **Cruceta inferior:** activar desactivar farolillo
- **L1:** con farolillo activado activar la luz mágica
- **Options:** menú de pausa.
- **L3:** correr.
- **R3:** recolocar la cámara detrás del personaje.
- **L1**: entrar en el modo fijar enemigos.
- **R3 dentro de modo fijar:** fijar a un enemigo para dirigir los ataques a este.

## Sonido

El trabajo de sonido se está llevando a cabo conjuntamente con un compositor externo a la carrera estudiante del conservatorio. A continuación Víctor Ávila añadirá su punto de vista sobre el proceso de la creación y componentes del sonido:

La base musical y sonora de Walpurgis Night, ha sido diseñada para reflejar, el mundo en el que transcurre el juego, utilizando como base musical el elemento de mitología y magia y sobre todo la presencia de la protagonista, también en la banda sonora.

Por una parte, quería darle importancia al hecho de utilizar mitología nórdica como motor de la historia y que, auditivamente sea claramente definible. Es por ello que se han utilizado referencias armónicas y rítmicas de temas y danzas populares noruegas.

Se quería que la banda sonora, igual que la experiencia sonora, fuera dentro de las posibilidades, un elemento más de la historia y funcionará como un potenciador de las sensaciones que el juego transmitiese. Es por ese motivo que se utilizan elementos técnicos como la espacialización, o diferentes tipos de reverberación, dependiendo del escenario en el que transcurre la acción. Consiguiendo así mayor inmersión del jugador.

Se decidió trabajar como elemento de unión los diferentes temas la voz femenina, representando el canto de la protagonista, y hacer esta extensible en la mayor parte de los temas, dando una altura y timbre de voz específico a cada una de las valkirias.

En todo momento se da un ambiente de misterio o tensión, inclusive en las pantallas de guardado, consiguiendo que el jugador esté a la expectativa en todo momento y siempre quiera avanzar.

Se niega cualquier temporalidad o elemento que pueda mantener un pulso, haciendo una música incidental de lugar más que de acción.

En resumen, se ha compuesto una banda sonora, que cuente la historia a la par que el juego o las imágenes y que complemente a estas, ofreciendo una experiencia al jugador lo más inmersiva posible.