THE FINAL SPELL

3D RPG VIDEO GAME DEMO **DESIGN** AND **ART IMPLEMENTATION** SUPPORTED BY **AGILE PRACTICES**

Cristian Cantos Rosello

Final Degree Work
Bachelor's Degree in Video Game Design and Development
Universitat Jaume I

Supervised by: Diego José Díaz García

ACKNOWLEDGEMENTS

I would especially like to thank David Gómez for being a great partner during all this time. Without him and without his support, I would not have achieved everything I have done.

I thank Diego Díaz for being our tutor and helping us whenever we needed it.

Thanks to my family for all the support they have given me over the years.

ABSTRACT

This document describes the whole process of making a 3D video game demo for the Final degree project in Design and development of video games, especially the artistic section of the video game.

The artistic section consists of the design of the game art for a medieval fantasy RPG video game. The different parts of the art of The Final spell are: 3D models, user interface, particle systems, shaders and improving the visual quality of the game.

The game engine used in The Final Spell is the Unity 3D engine. In addition, Unity offers several tools to improve the visual quality that were used throughout the project.

KEYWORDS

Game development, game programming, game art, game design, agile practices.

INDEX

1. INTRODUCTION

The Final Spell is a 3D RPG individual video game with low-poly style in third person. Along the videogame, players will learn to do different spells that require ability.

This project was made by two students, David Gómez focused in the programming part and me focused in the art part.

1.1 Motivation

The motivation of this project came at the moment we realized that in the design and development of video games degree lacks teams of students with functional completed video games, especially 3d video games. For this reason, we decided to show all the potential of making a video game with every component: narrative, art, programming and music. Having already experience together in the video game development,, we decided to make this project together and show all the process of the team working and video game demo creation.

1.2 Goals

The main objective of this project is to show all the design process of the RPG video game demo in Unity 3D engine.
Then, artistic section covers:

- Creation of 3D models of characters, the world and enemies
- Animate characters and enemies
- Creation of 2d art for the user interface
- Make shaders, particles and postprocessing to improve the visual quality.

1.3 Initial state and tools

Before starting the project, the design of the game was already available, done during 2020 using the tool: WikiPad. The art and coe was done from scratch starting the first day of the final project degree subject.

For the development of The Final Spell, Unity 3D has been used with Visual Studio 2019 for the code, 3D models have been created with Blender [1] and Clip Studio Paint [5] for the textures. Discord [6] was the main tool for the communication between team members and Github [7] has given the students a way to share the development of the video game demo.

1.4 Related subjects

- **VJ1204** - Artistic Expression
- **VJ1205** - Programming II
- **VJ1216** - 3D Design
- **VJ1222** - Video Game Conceptual Design
- **VJ1223** - Video Game Art
- **VJ1224** - Software Engineering
- **VJ1226** - Character Design and Animation
- **VJ1227** - Game Engines
- **VJ1231** - Artificial Intelligence

2. GAME DESIGN

The Final Spell is a 3D RPG videogame of medieval fantasy in low-poly for PC. Video games like The Witcher, Skyrim or Valheim have been a great reference for the characteristics of RPG (fight, missions, exploration, enemies, etc.) of the project.

The first design of The Final Spell was made with WikiPad by my partner David. When it came to the development between the two, all this information moved to a Discord server so that the team was organised and able to review all the design of the videogame in an easy way.

2.1 Story

Since the Great Fire happened, La Acacia has been in balance separating virtue from danger, prosperity from ruin, the human race from the dark beings, and the North from the South.

A constant war, started several millennia ago, confronts the most privileged with the dark. Both sides have the same objective: to unify the continent towards their side.

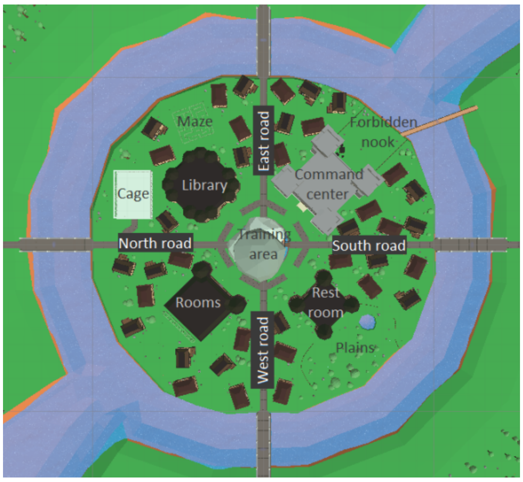
The North is protected by the Summit of the Wizards, chosen for their skills and potential and made the most powerful in the upper half of La Acacia.

The South, however, is invaded by the so-called dark beasts, who have controlled every corner of their side, taming - with dark magic - all living beings in their way.

A few years ago, a spy wizards team managed to penetrate deep into the hot, dangerous and heavily populated lands of the South, discovering something that could defeat the North and end the war.

The player controls a sorceress who has arrived to the Valley of Immortality to learn to control magic at her advantage and thus be able to fight in the war against the South.
There, the player meets a second sorceress which has an enemy relationship since the second sorceress wants to be better than others and will defeat whoever stands in her way.

Throughout the video game, the player will meet the spy wizards team and will discover what happened in the travel to the South. This team tried to find weak points for destroying the South colony, but as they entered a village, they found that the Dark high ranks were creating dark sorcerers and sorceresses to finish with the North and end the war. So then, the wizards pick a dark sorceress baby girl. This girl is the second sorceress which the player confronts at the end of the demo.



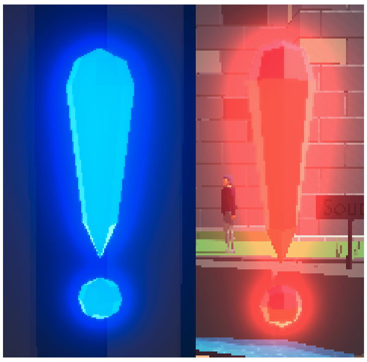
2.2 Exploration and quests

The Valley of Immortality is the only place that the player can explore in the demo of The Final Spell.

The valley is divided into four areas, which are separated by roads. Bridges connect the rest of the world with The Valley of Immortality.

The areas to be emphasized are the Forbidden Nook, the Training area, the Cage and the Plains since it is necessary to complete the quests of the demo.

The quests are divided into mainly and secondary. In the video game can differentiate in main quests have red exclamations and secondary quests have blue exclamations. Finally, the quests in the demo are:



First main quest	Talk to the instructor in the Training area. Kill 1 wild pig in the cage.
Second main quest	Talk to the instructor in the Training area. Kill 5 wild pigs in the cage.
Third main quest	Talk to the instructor in the Training area. Kill 8 butterflies in the plains.
Fourth main quest	Talk to the instructor in the Training area. Fight against the second sorceress in the Training area.
Secondary quest	Talk with the kid in the entry of the Forbidden nook. Defeat all the slimes in the Forbidden nook.

Furthermore, the player can complete achievements that focus on exploring the world. When the player completes the achievements, the shelves in the library are filled with books that contain lore about the achievement completed. In the demo there are two achievements available:

Pet the dog	Pet 10 times to any dog and talk with the old man between two dogs.
Complete the maze	Complete the maze behind the Library and talk with the advanced sorceress.

2.3 Items and inventory

The Final Spell contains some items that the player can get by killing enemies, collecting or buying. These objects have one of these three utilities: resource, consumable, saleable.

Type	Items	How to get
Resources	Geranium	Buying or collecting
Consumables	Health potion	Buying
Saleables	Dark essence	Killing dark enemies

	Butterfly wing	Killing butterflies
	Tusk	Killing wild pigs
	Slime liquid	Killing slimes
	Horn	Killing wild pigs

2.4 Character actions and abilities

The Final Spell has a protagonist that can make various actions like interact with the world, talk with other characters, pet animals and make some spells.

The difference between other RPGs and this videogame is the spells. These spells from the beginning the player don’t know how to do, they need to do trial and error to figure out how to do them. Moreover, the player can communicate with other NPCs to discover clues about how to do the spells.

2.5 Experience

Most RPGs video games use the “experience” to level up the character and get new abilities and develop them. However, in The Final Spell there isn’t experience, the player has to focus to learn the different spells of the game that requires precision, organisation and resources.

2.6 Combat

The combat is in real time where the player needs to do some spells to defeat the enemies. Currently, the demo of The Final Spell has three enemies:

Enemy	Range attack	Health	Damage
Wild pig	Melee	Medium	Low
Butterfly	Medium range	Low	High
Slime	Melee	High	High

In addition, the demo of The Final Spell contains three spells:

Spell	Range	Damage	Resource
Quick water drop	Long	Low	Water source
Floral Flame	Short	Medium	Flora
Lethal orb	Short	High	Geranium

2.7 Interface

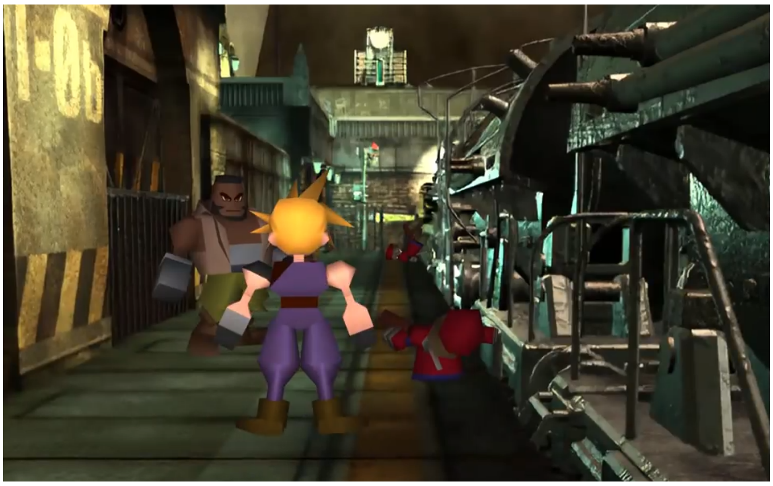
The user interface helps the player to know the situation about the character. Furthermore, it helps to learn how to improve spells. Thus, the player can see a life bar of the character, buttons to open the inventory, the map and the book of spells.

Also, there is a dialogue box that it will open when the player interacts with other NPCs.

Likewise, there is a shop in which the player can buy and sell items through the currency of the game.

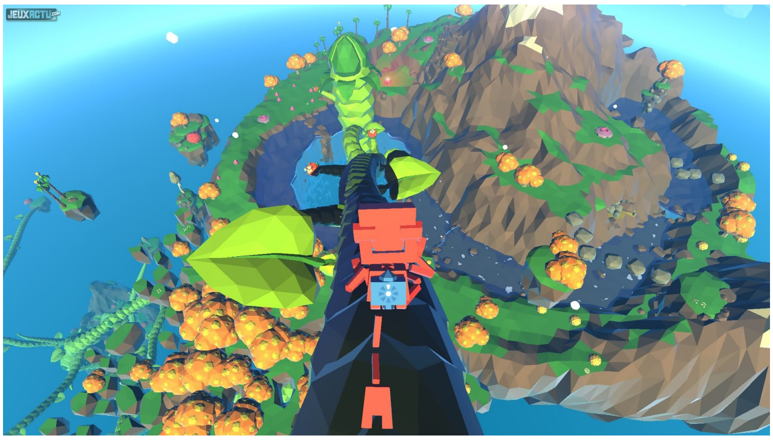
3. ART DESIGN

In The Final Spell there are two art types: 3D for the world and 2D for the interface.



As for the world, every 3D model has a low-poly style [8]. This style used in old video games that tried to create models in low-resolution and this way it was easier to model because of few details and optimize the time of rendering since the technology didn't allow many details like Final fantasy 7 (PS1, 1997). However, currently the low-poly is a technique of 3d minimalist art despite having a much more sophisticated and powerful rendering technology that allows for more realistic results, a lot of video games continue to opt for this style by classifying them like “low-poly video game”. With new technologies the low-poly video games opt to put better illumination and post-processing like Grow Up (PC, 2015). Therefore, in this work there is a section dedicated to illumination and post-processing.

Low-poly videogames with spot colors use a color palette created from the beginning so that all the models use the same colors. Therefore, The Final Spell uses a color palette and added black pixels in case it was necessary to add any color later.



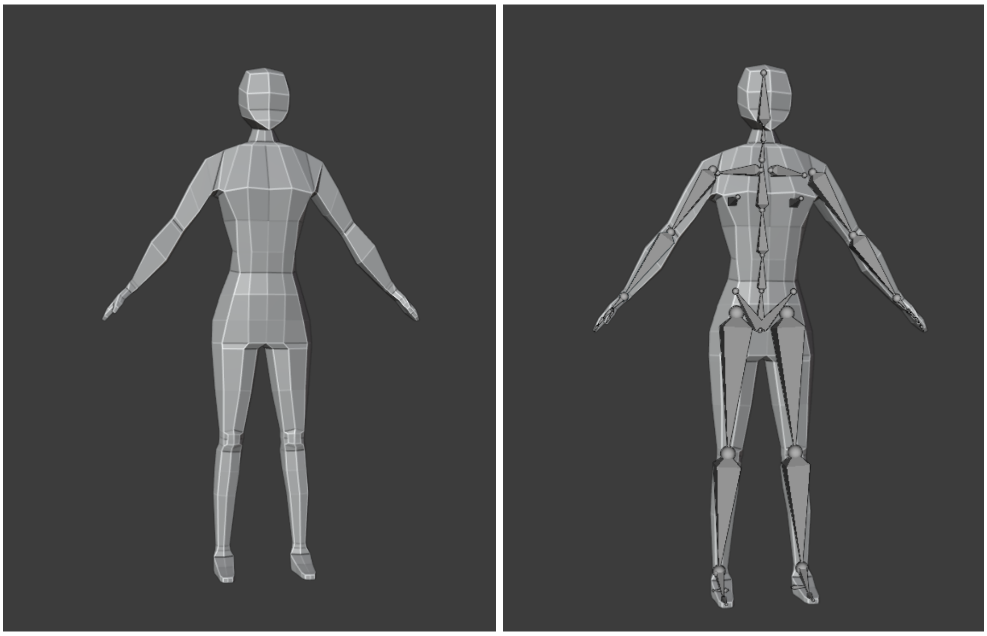
All the models in The Final Spells were made with Blender. The UVs of each model are placed in the desired color of the palette and then import to Unity. In Unity, a material with the color palette has to be applied to the model.

Something to say is the Level of detail. It is a technique that is taken advantage of the human eye since it loses detail of the elements as they move away from the camera. Thus, models with low polygons will be far away from the camera. This technique is to increase the video game efficiency, especially video games with open worlds. The Level of detail isn't used in the demo of The Final Spell as it is a small world and it isn't necessary as it wouldn't affect the efficiency of the game. On the other hand, in the complete video game it would be needed, since it is an open world, then it is necessary to use techniques to improve the efficiency, Level of detail being one of them.

As for the art used for the user interface, it has its own style. It has a colored line art depending on the object, that is to say, it won't be completely black. In addition, the shadows and lights are blurring to give a visual effect that objects have volume.

3.1 Characters

All the human models are created from the same 3D model and skeleton. In this way, it can create different characters in a faster way. This model was done from two images showing the human side and front to get a realistic anatomy.

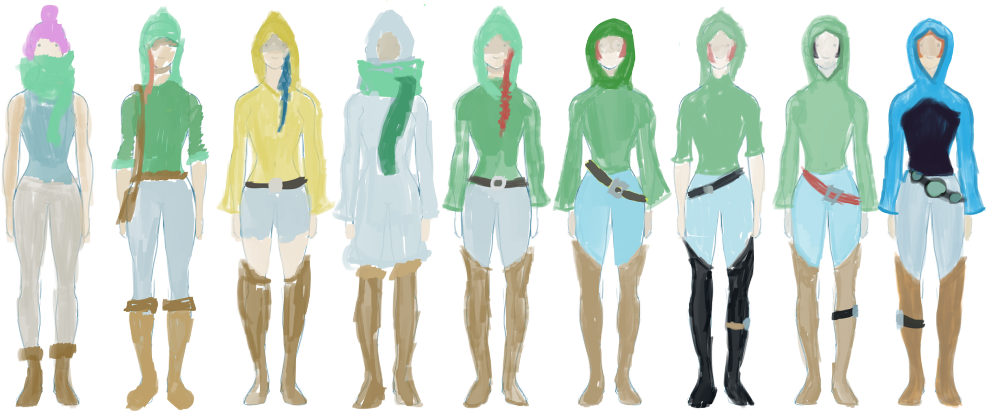


In this model all the human animations will be created, including the main character and civilian animations. As all humans have the same skeleton, the same animations can be used on different characters. Then in Unity we will have a folder with all the animations and save space because we don't have to import the same animations every time we import a character.

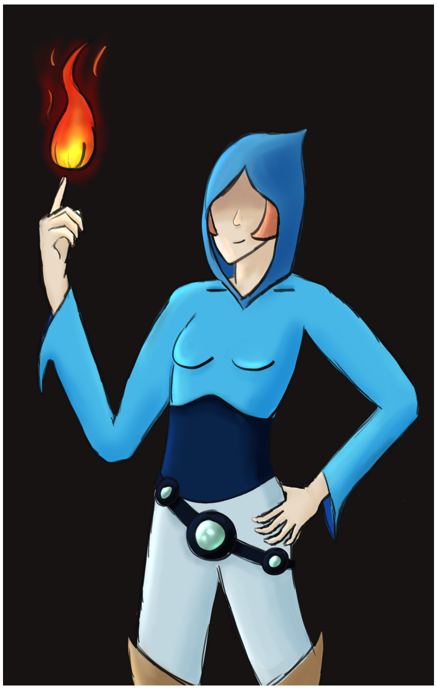
Finally, the clothes of each character are created and linked with the bones for the animations, also small tweaks will have to be made so that the body doesn't go through the clothes when it is making movements.

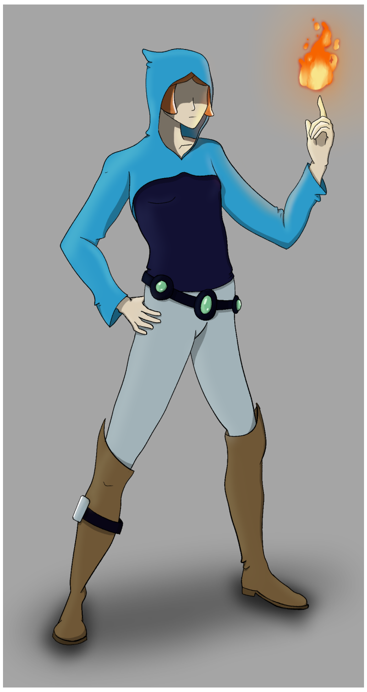
3.1.1 Main character

In order to get our final art decision, our main character went through several designs. We finally decided to choose the one that seemed more silent but still powerful. In all the designs we have kept one part of the character head covered, reinforcing the look of a quiet person. The blue is the most predominant color in the design since it expresses tranquility.



The character portrait went through two phases in order to meet the style and quality that we were looking for. This portrait isn't useful for the videogame, but we needed to show how the 3D model was to be.



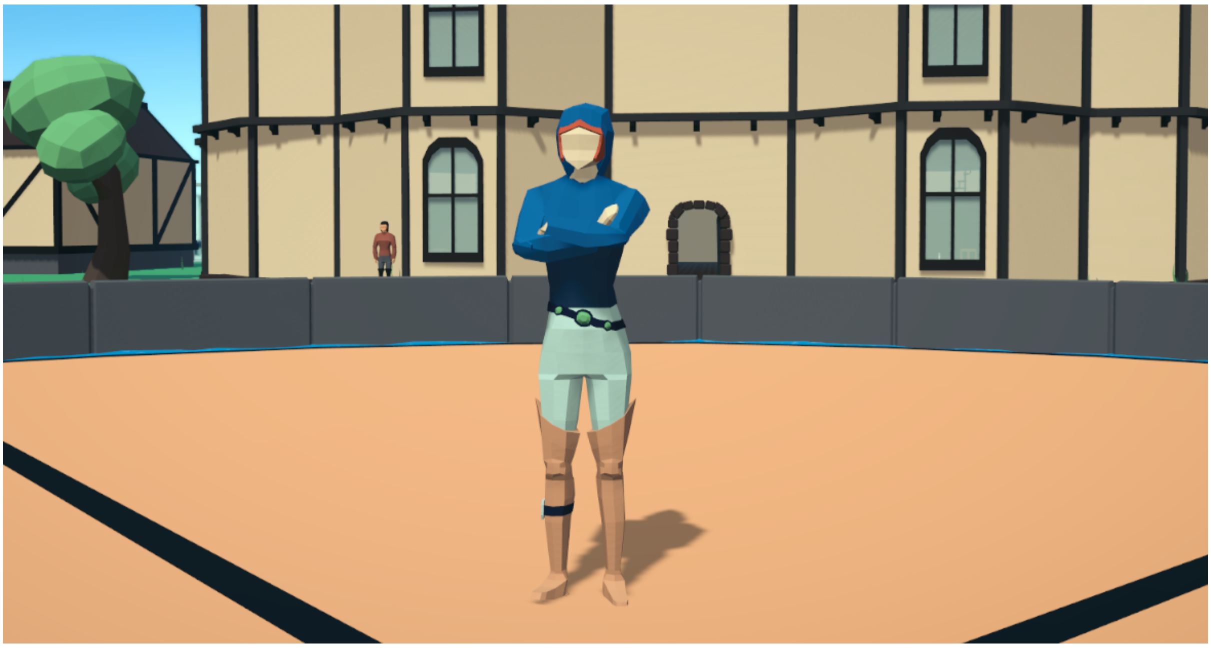


Finally, the 3D model of the main character was created from the base model mentioned earlier.

As for the animations, this character has a lot since she needs to do all the spells. The animations are:

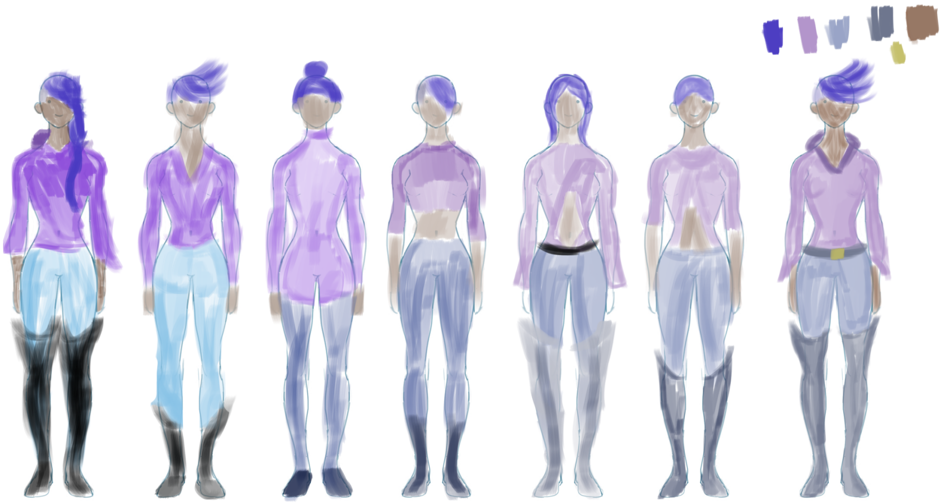
- Two types of idle, one is when she is stopped and another is when she is stopped during 5 seconds.
- Walking and running
- Idle crouch and walking crouch to make less noise

- Die
- Five states for the Floral flame spell
- Two states for the Lethal orb spell
- Two states for the Quick water drop spell



3.1.2 Second main character

This secondary character has one of the most important roles in the game. In the game lore, this character is connected to the dark enemies (the South). To show that connection in a subtle way, it was decided to use the same colors as the dark enemies: blue and purple. To get the desired style of this character, we went through a series of sketches focusing on the clothes and hair shape.



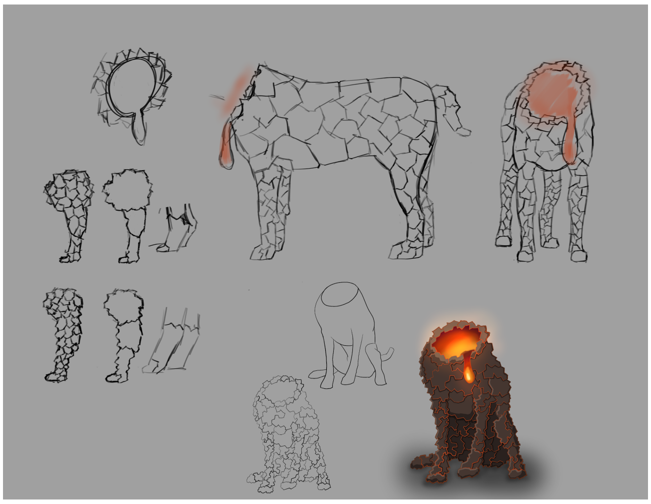
Compared to the main character, this one has a sharper style, creating a dangerous and evil feeling.

The player can fight with this character at the end of the demo. She can only attack with the Quick water drop spell.

3.1.3 Friendly NPCs

There are three types of friendly NPCs in The Final Spell: instructors, advanced sorcerers and sorceresses, apprentices and civilians.

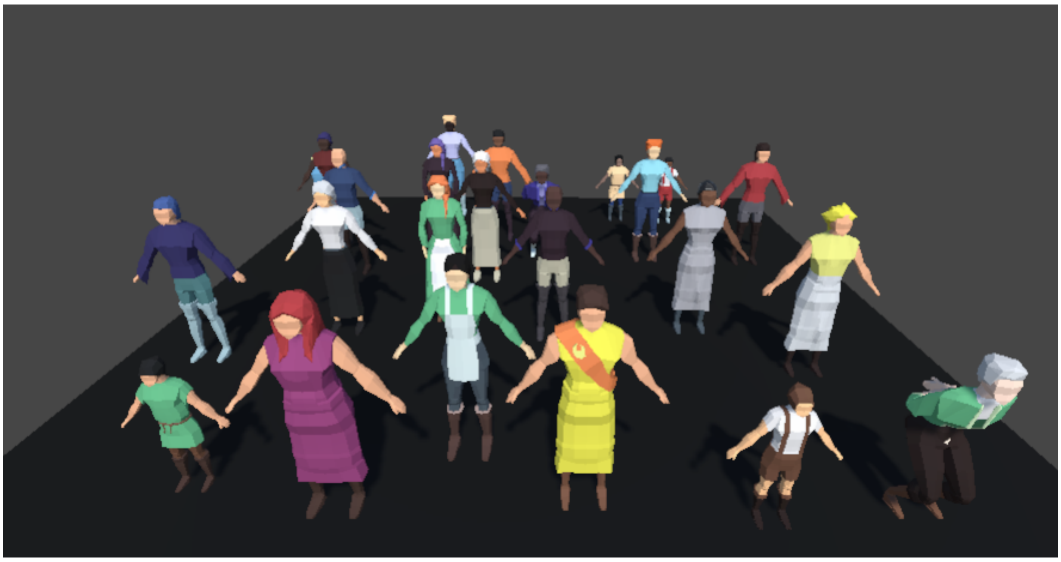
Instructors wear a toga that covers their whole body except their arms, the color of the toga will depend on the instructor's choice. They also wear long boots and will have a golden flame shaped brooch.



Advanced sorcerers and sorceresses wear the same toga as instructors, however they don't wear the golden brooch, showing that they are a lower rank.

Apprentices wear a hood, which they decide whether to wear or not. They wear long boots and long trousers, all the colors are decided by the apprentices.

Finally, civilians wear clothes that don't look like people who can do magic. Among them, there is a fantastic dog who is there to pet and get an achievement in the demo.



3.2 Enemies

In The Final Spell demo there are 3 enemies which all have the same color palette which consists of cold colours to differentiate them from the environment (forests, fields, cities, etc.) and red eyes to differentiate them from the friendly NPCs.

Each enemy will drop different items when killed. However, they all drop dark essence, which in terms of the game's lore, is used to transform animals into evil beings.

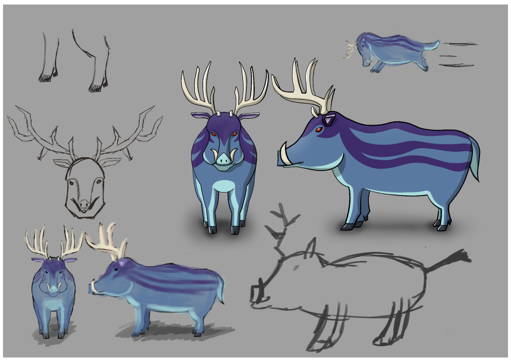
In addition, all enemies have various states. There are different animations and different skeletons for each enemy.

- Idle: one is when it is stopped and another is when it is stopped during 5 seconds.
- Wander: animation which the enemy will be moving slowly.
- Run: animation which is used when the enemy has detected the player.
- Attack: animation when the enemy is close to the player to attack them
- Die: animation for when the enemy has been defeated by the player.

3.2.1 Wild pig

The ears, horns and tail of the wild pig follow the style of the deers, but his body and head are inspired by a real-world wild pig. This animal performs his melee attack like a real-world pig would defend himself, but putting the horns in front like deer would do.

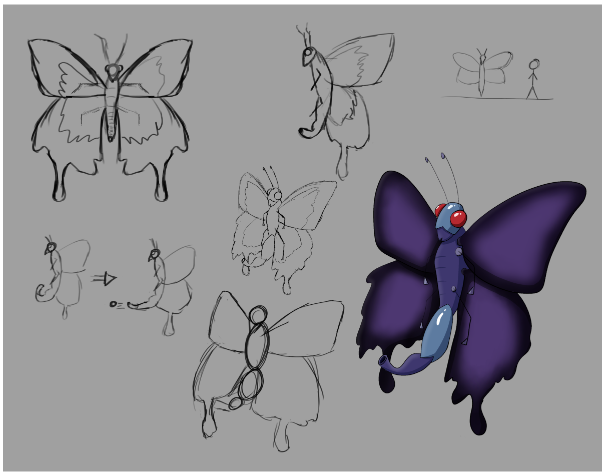
They are easy to defeat but funny to fight. They drop three items: a horn, a tusk and a dark essence.



3.2.2 Butterfly

Inspired by the Blue Ulysses butterfly, this visually harmless animal has the same height as a human character. Furthermore, it has armor (represented with the blue parts) which is made of his own slime. Their attacks are strong, but they have low health.

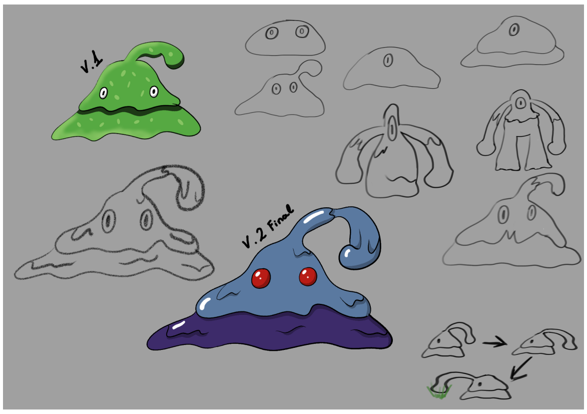
Its long-distance attack is a slime ball shot with his tail, inspired by a wasp defending himself. The balls are projectiles that follow an objective until they reach it. Butterflies drop butterfly wing and dark essence.



3.2.3 Slime

The slime was one of the first enemies that was designed. During the development of the concept art, it was decided that instead of following the industry-standard color of slimes (green), it was better to use the blue and purple colors, keeping the style of the other enemies.

This enemy has a melee attack, using his slime ball on the head to hit. They are big and deal a lot of damage. Slimes drop slime liquid and dark essence.



3.3 Environment

As for the world, in the final game it is intended that the entire La Acacia will be created, but for the demo of The Final Spell we will only focus on the Valley of Immortality, being the city where the apprentice wizards will learn spells from their masters.

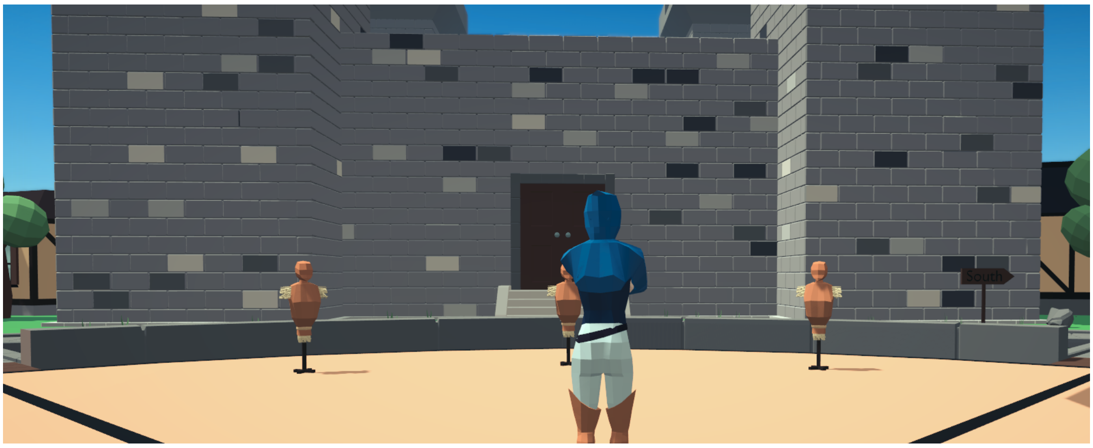
The Valley of Immortality is surrounded by a river and the city is made up of four important buildings: The command centre, the restroom, the rooms and the library. The library is the only one the player can access.

In addition, there are five other parts: a maze, a cage, the plains, the forbidden nook and the training area.

Finally, the scene has various plants, trees, stones, houses, paths, etc.

3.3.1 Buildings

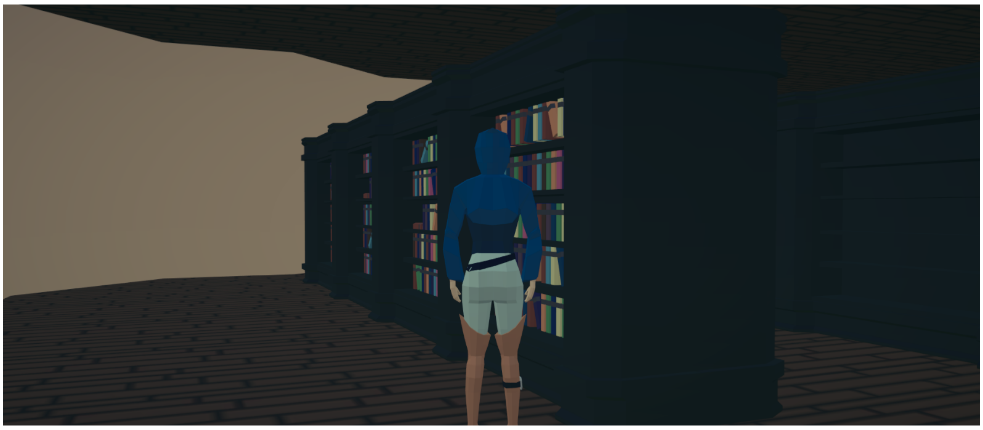
The command centre is the most important place in the Valley of Immortality as it is the place where the sorcerers and sorceress reunite. Because of this importance, the building is made of stone.



The restroom where people can relax and eat, any citizen of the Valley of Immortality can go to the dining hall if necessary.

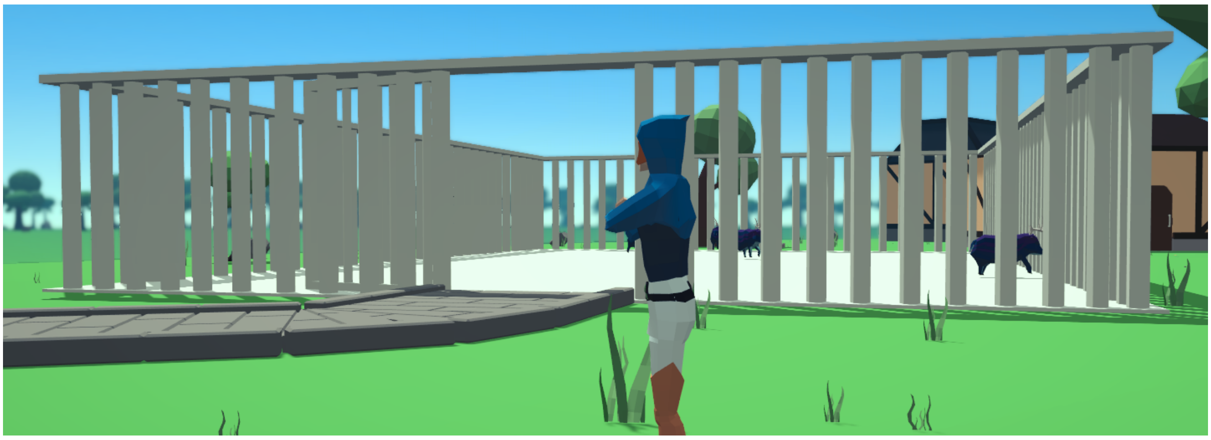
The dormitories so that any apprentice coming from a distant village has a place to sleep.

The library is the only building in the demo that the player can enter as it will show the demo achievements. In the library there are various pieces of furniture such as tables, candles, chairs, stairs and the bookshelves. The player fills the bookshelves with books by completing the achievements. In order to avoid repetition in the bookshelves, several collections of books were made and placed randomly on each shelf.



3.3.2 Other places

The cage is made to hold wild pigs. This place was made only for the demo so that the player will be able to face the wild boars. Enemies will not be able to get out of the cage so the player will be protected. The instructor asks to come here to complete the first and second main quests.



Plains are made for butterflies. This place was made only for the demo so the player can face the butterflies. Enemies will not be able to get out of the cage so the player will be protected if he gets out of the cage. The instructor asks to come here to complete the thirst main quest.

third?

The forbidden nook was made for a secondary quest in the demo. In this place there will be a child that the player will have to save from the slimes in the area. It is behind the command centre and surrounded by fences to keep enemies out, but one fence is broken so that apprentices can enter and practice against monsters. Also, there are several weapons in case the apprentices run out of spells, so they can always defend themselves.

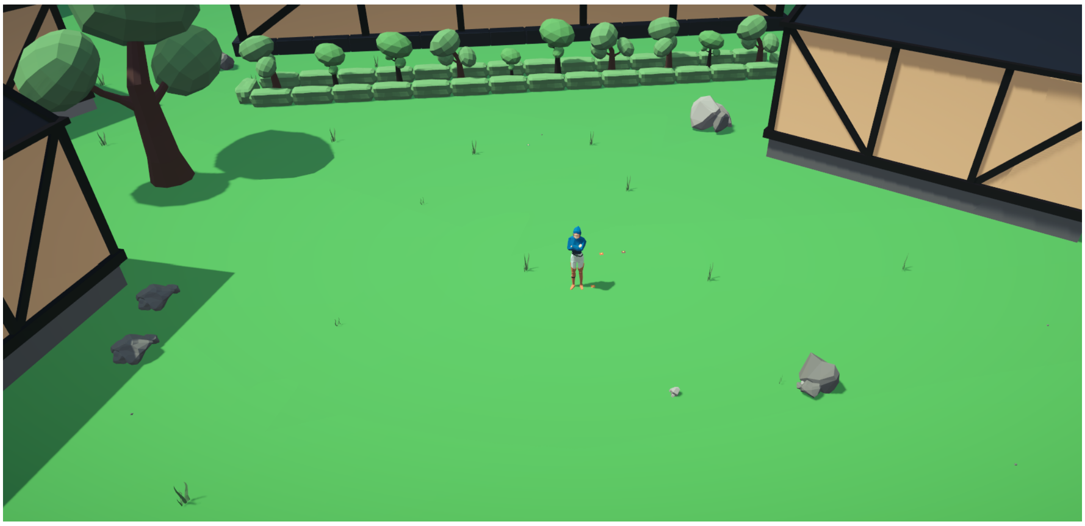
The training area is the place where the apprentices train, where players will meet their instructor and can do a battle against their partner.

3.3.3 Decoration

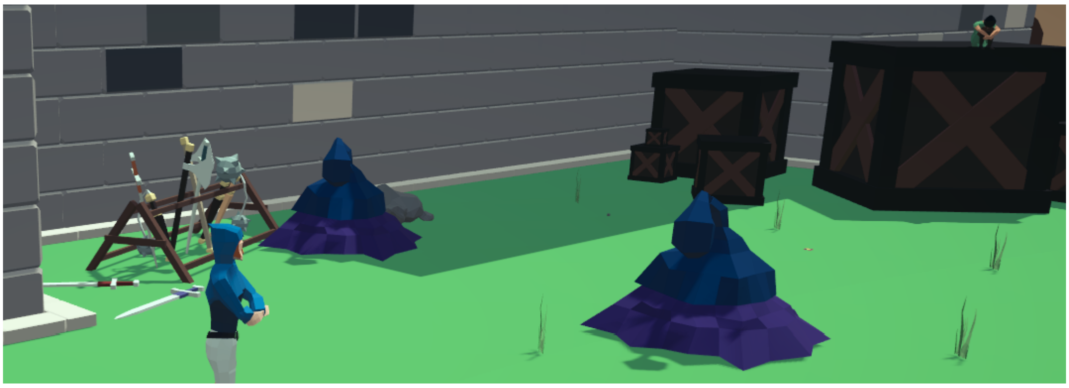
Apart from all the places mentioned above, the world has been decorated with different elements.

First of all, two different types of houses were modelled to justify having NPCs other than wizards. They have been placed in Unity and duplicated to add a larger amount of houses.

Various types of rocks, grasses, flowers and trees have been modelled and placed randomly around the map, and the sizes and rotations have been changed to create different types.



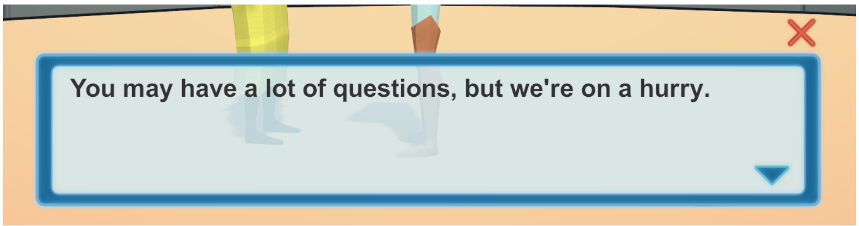
For the forbidden nook, several boxes have been added for the child to hide from the slimes and a weapon stand to justify the apprentices being able to use weapons if they run out of resources when practising against the monsters.



To add paths, several tiles have been created and placed in Unity to make the main paths. Bridges were also created as tiles and are connected to the main paths. Finally, a shader has been created on the bridges to show that there is a magic block so that the player can't leave the Valley of Immortality in the demo.

3.4 User interface

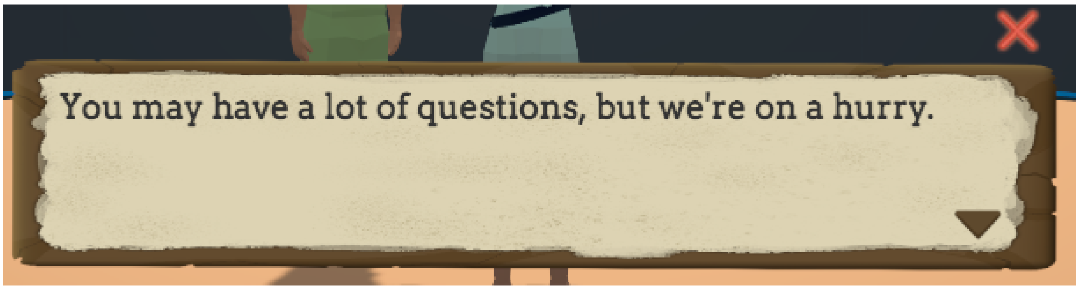
Before, the user interface had a bright bluish style to give it a magical feel. However, it didn't match well with the rest of the game's artwork. For this reason, the UI was redone to be more organic and match with the map shown in the map menu.



You may have a lot of questions, but we're on a hurry.

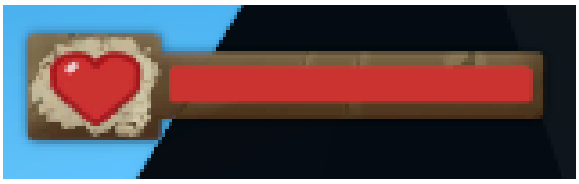
These menus are:

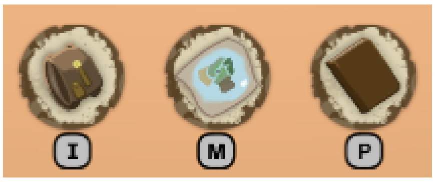
- Dialogue: This menu will show all the conversations the player has with the NPCs.



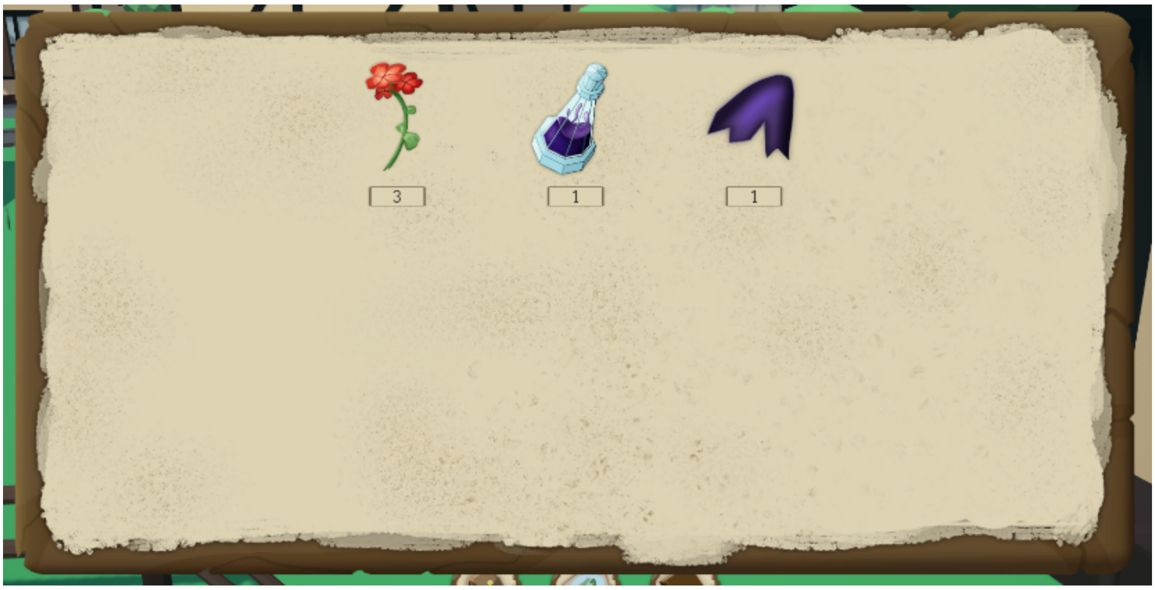
You may have a lot of questions, but we're on a hurry.

- Life bar and menu buttons: The menu buttons are used to open the Inventory, Map and Spellbook menus.

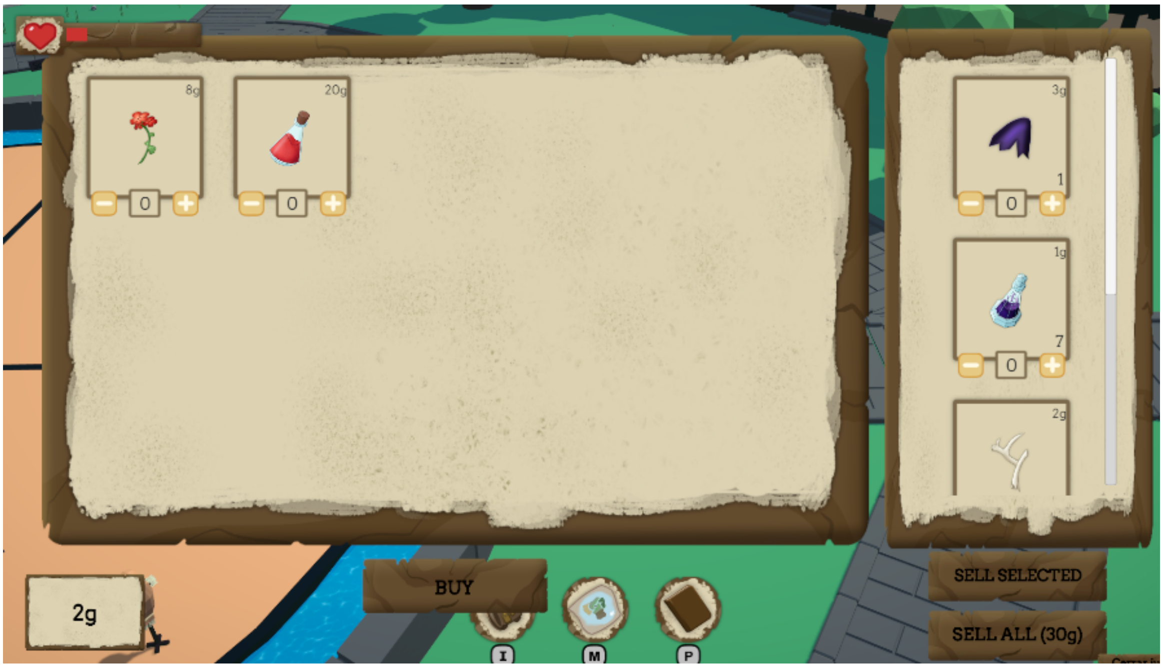




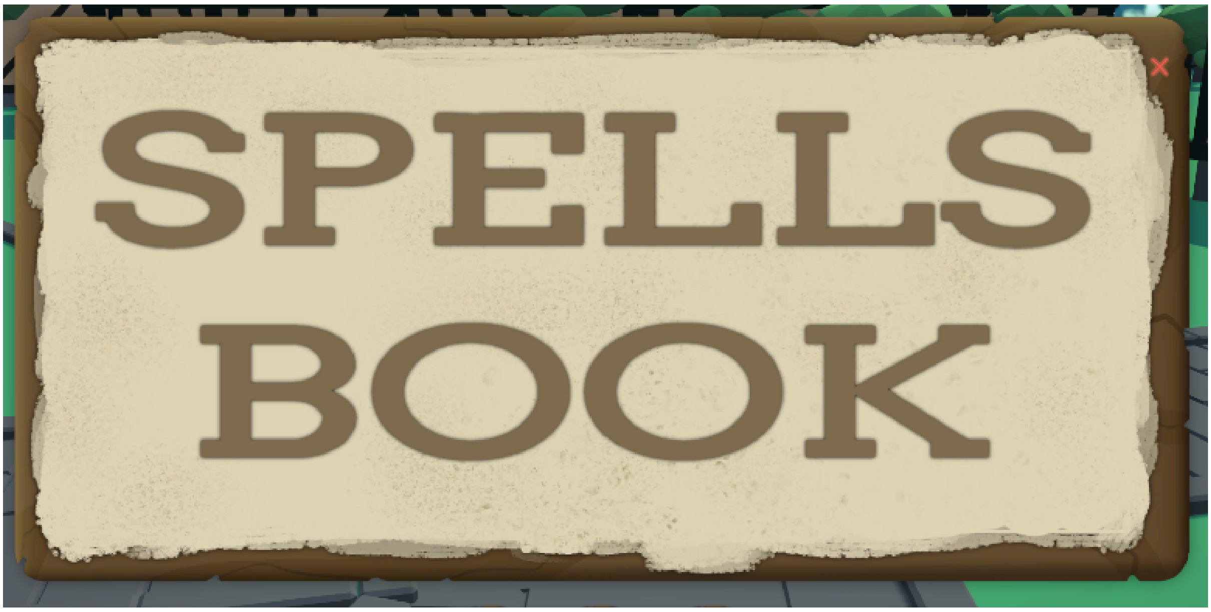
- Player Inventory: In this inventory players can see all the items they have collected during the game.



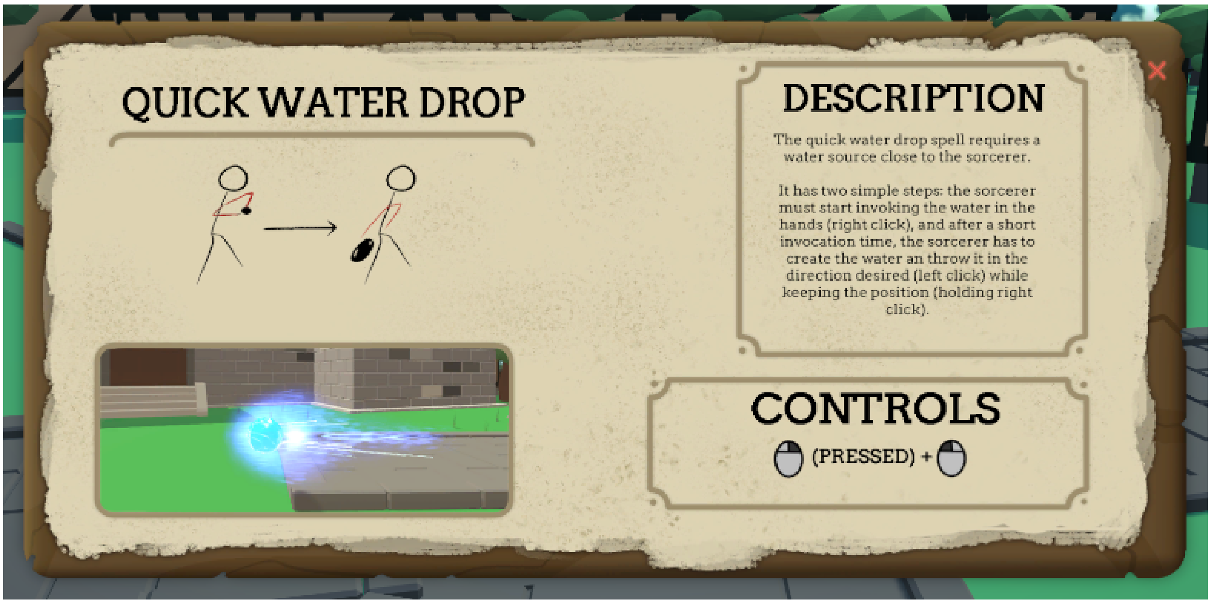
- Merchant's inventory: players can buy items from the merchant via the inventory on the left or sell items to the merchant through the inventory on the right.



- Spell books: This book contains all the spells the player has learned. Currently in the demo, there are three spells that will be unlocked throughout the demo, these are: Floral Flame, Quick Water Drop and Lethal orb. Each spell contains a description, an image from the video game, the movements the character makes to cast the spells and the buttons the player has to use to invoke them.



SPELLS BOOK



QUICK WATER DROP

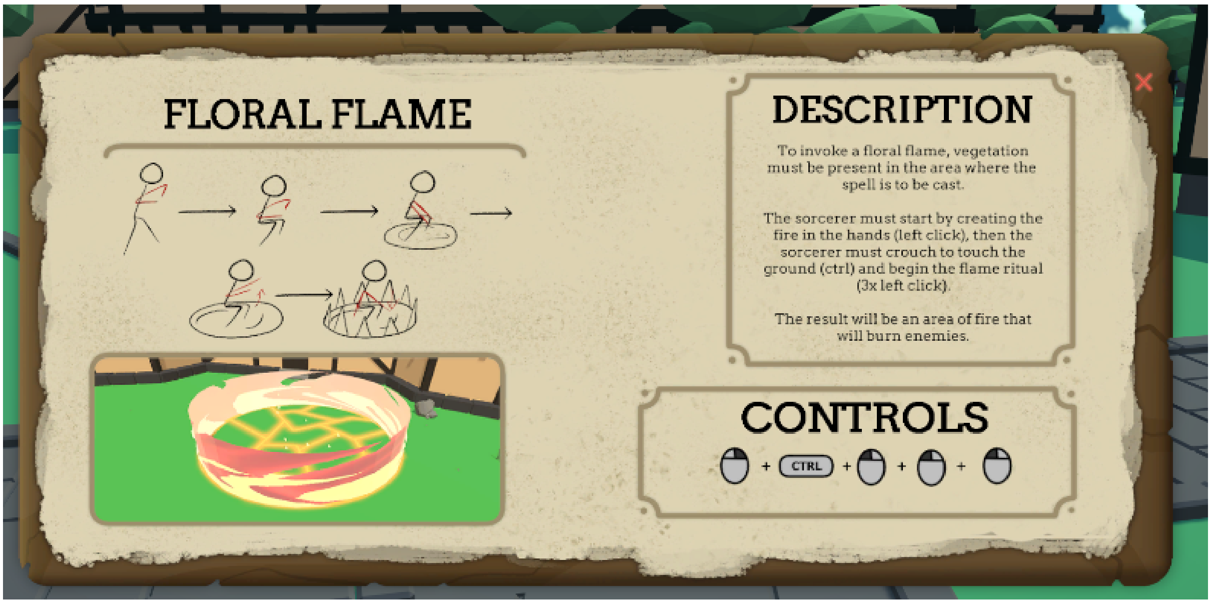
DESCRIPTION

The quick water drop spell requires a water source close to the sorcerer.

It has two simple steps: the sorcerer must start invoking the water in the hands (right click), and after a short invocation time, the sorcerer has to create the water an throw it in the direction desired (left click) while keeping the position (holding right click).

CONTROLS

🖱 (PRESSED) + 🖱



FLORAL FLAME

DESCRIPTION

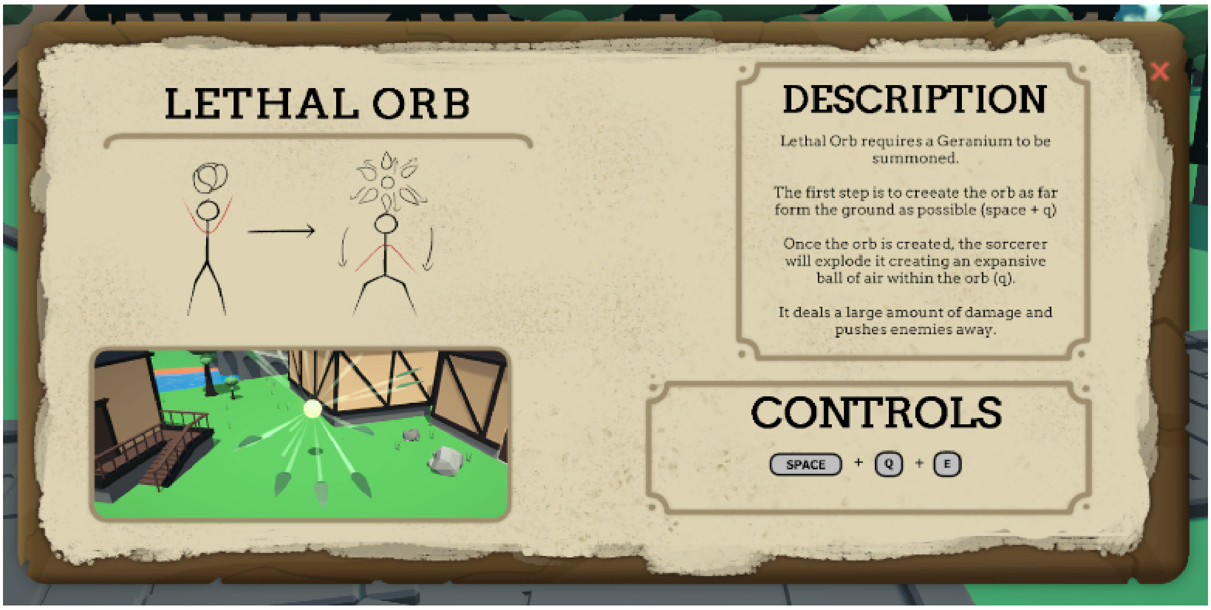
To invoke a floral flame, vegetation must be present in the area where the spell is to be cast.

The sorcerer must start by creating the fire in the hands (left click), then the sorcerer must crouch to touch the ground (ctrl) and begin the flame ritual (3x left click).

The result will be an area of fire that will burn enemies.

CONTROLS

🖱 + CTRL + 🖱 + 🖱 + 🖱



LETHAL ORB

DESCRIPTION

Lethal Orb requires a Geranium to be summoned.

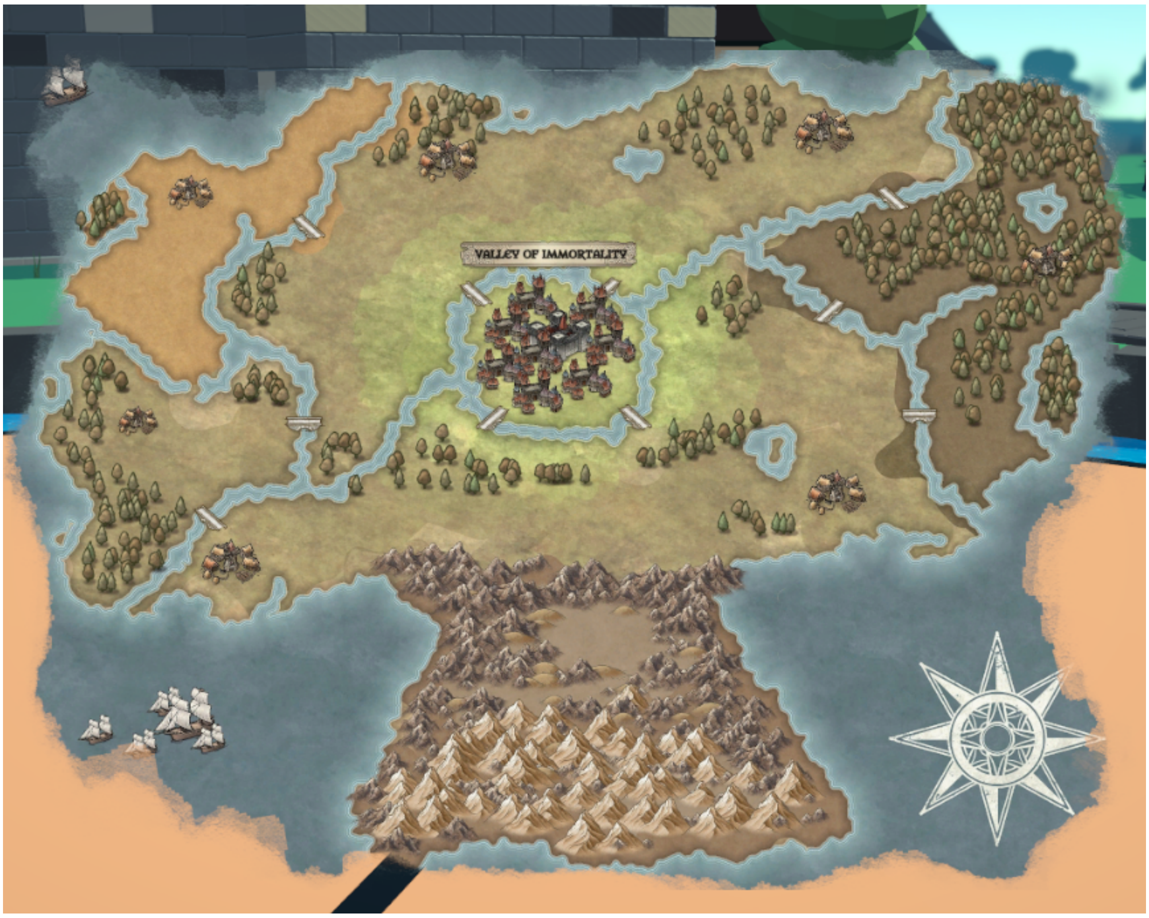
The first step is to creeate the orb as far form the ground as possible (space + q)

Once the orb is created, the sorcerer will explode it creating an expansive ball of air within the orb (q).

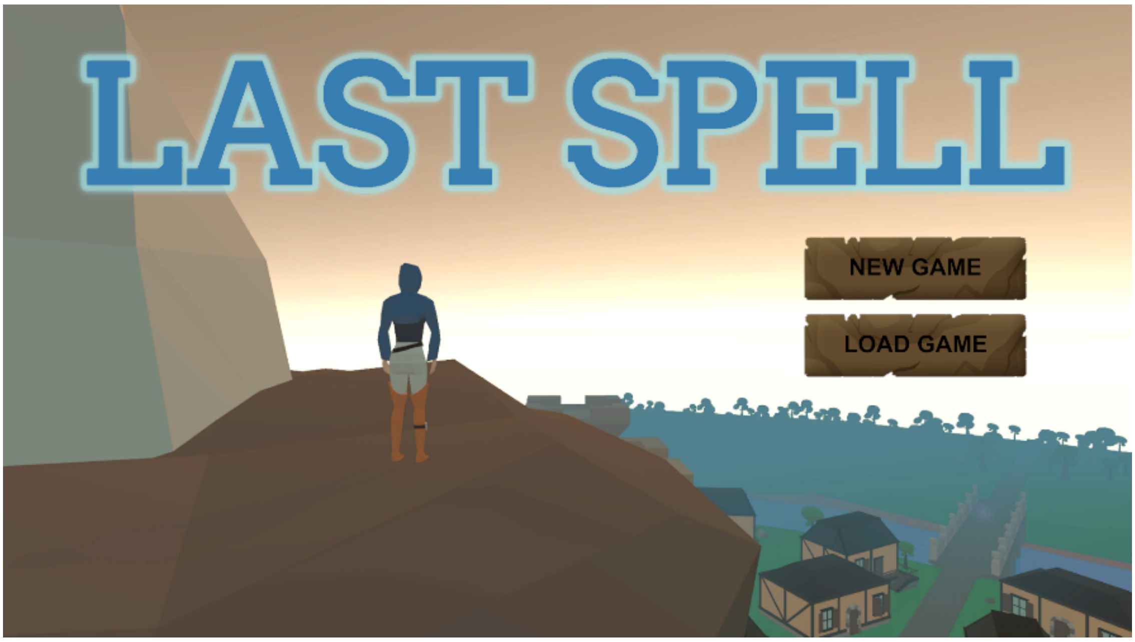
It deals a large amount of damage and pushes enemies away.

CONTROLS

Another menu of the game that does not appear with the blue lineart is the map that has been created using the tool https://inkarnate.com/. This map shows the entire continent (La Acacia) but for the demo only the centre city is available: The Valley of Immortality.

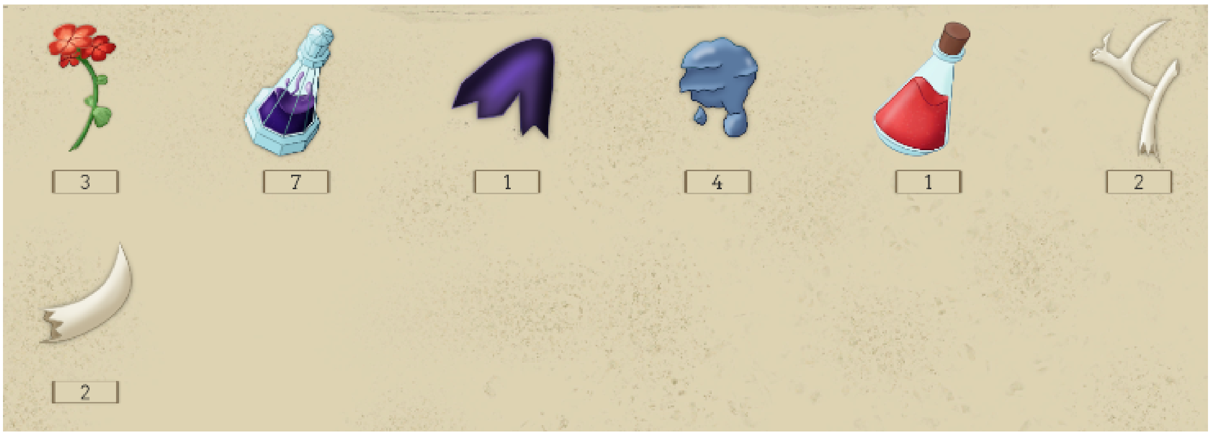


Furthermore, the start menu contains the game title that has been implemented in the same style used for the Spellbook and an image has been added, built in Unity, where the character is looking at the background above the statue La Acacia in a sunset.



In terms of items, there are several items that can be obtained throughout the demo. All items can be sold in the shop for money. These are:

- Butterfly wing: you can get it by killing butterflies.
- Dark essence: can be obtained by killing any enemy
- Geranium: can be obtained by picking them up in the plains or buying them from the merchant.
- Heal potion: this item can only be obtained by buying them from the merchant. It is used to heal the character.
- Horn: can be obtained by killing wild boars.
- Slime liquid: can be obtained by killing slimes.
- Tusk: can be obtained by killing boars.



3.5 Unity tools

3.5.1 Shaders

Shaders are used to make different special effects, these are scripts that have some mathematical calculations and algorithms that allow the processing of the colour and lighting of each pixel on screen.

These shaders are related to the materials, which are definitions of how the surface should be rendered, all the information needed by the material will be provided by the shader.

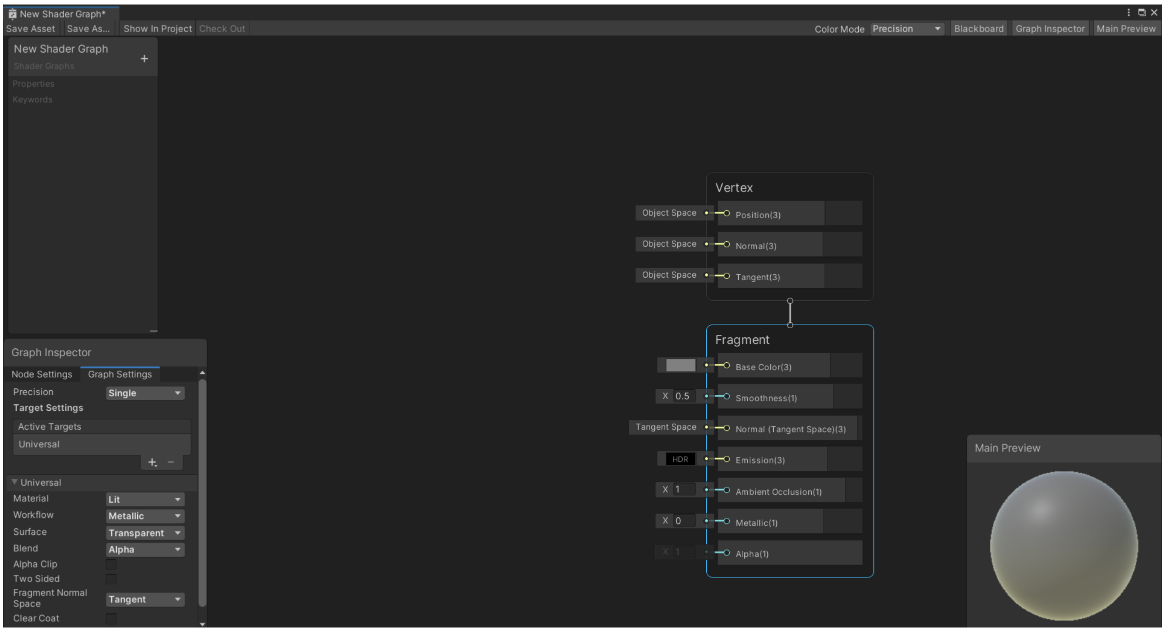
Shaders work through the render pipeline in Unity. In a new project by default the "Built-in" render pipeline is used. However, in our project we have added the Universal Render Pipeline which allows us to use more complex shaders using less GPU resources.

In the Video Game Design and Development degree, we were taught in the computer graphics subject how to program shaders using scripts with the CSGL. However, with the Universal Render Pipeline, the Shader Graph [2] has been created, which is a tool that allows us to create shaders in a simpler and faster way using nodes.

So, for our project we have used the Shader Graph to create different effects such as water, fire and other effects for magic spells.

As for the interface of the Shader Graph.

- At the top left are all the public variables that can be changed in the materials.
- Bottom left are the settings of the public variables and the material settings where we have the type of material, the surface, transparency, etc.
- On the bottom right we have the preview of the shader on a mesh.
- In the centre we have the Vertex shader and the Fragment shader.



Precision: Single

Target Settings

Active Targets: Universal

Universal

Material: Lit

Workflow: Metallic

Surface: Transparent

Blend: Alpha

Alpha Clip: □

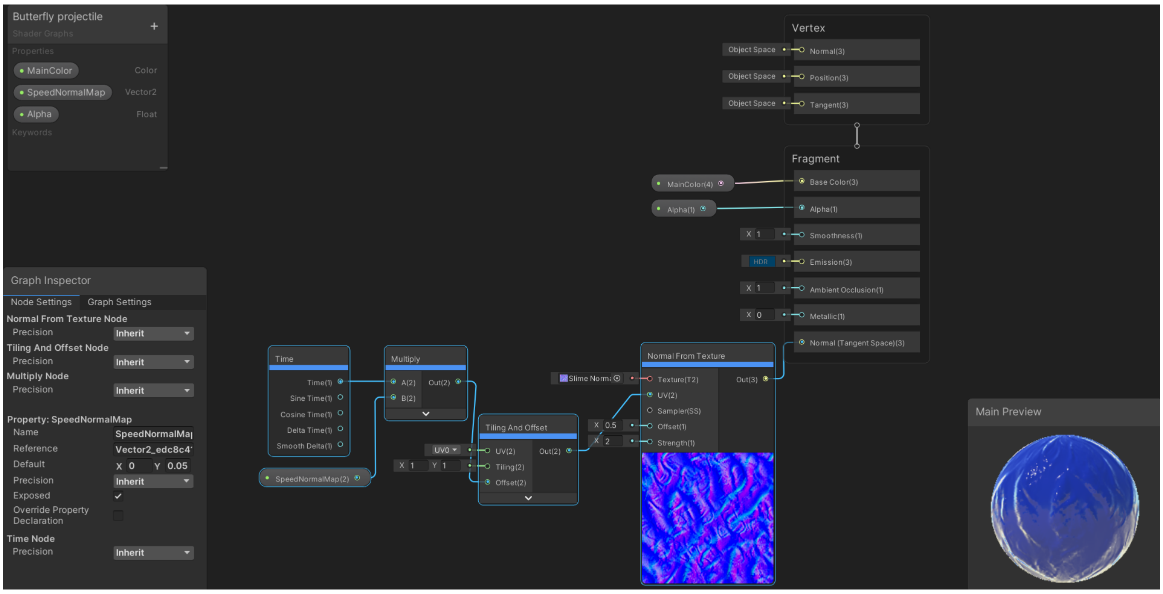
Two Sided: □

Fragment Normal Space: Tangent

Clear Coat: □

From here we have been modifying the parameters to get the shaders we want. One comment to make before continuing, all the shaders have movement thanks to the "Time" node, but it won't be able to notice as it only shows images in the document.

The first shader we have is the projectile of the enemy "Butterfly", this shader is very simple as the only thing that is complex is that a map of normals, which appears to be slime, moves. So then, with the blue colour of the butterfly and the normal map it gives the appearance of slime.



Graph Inspector

Node Settings | Graph Settings

Normal From Texture Node
Precision: Inherit

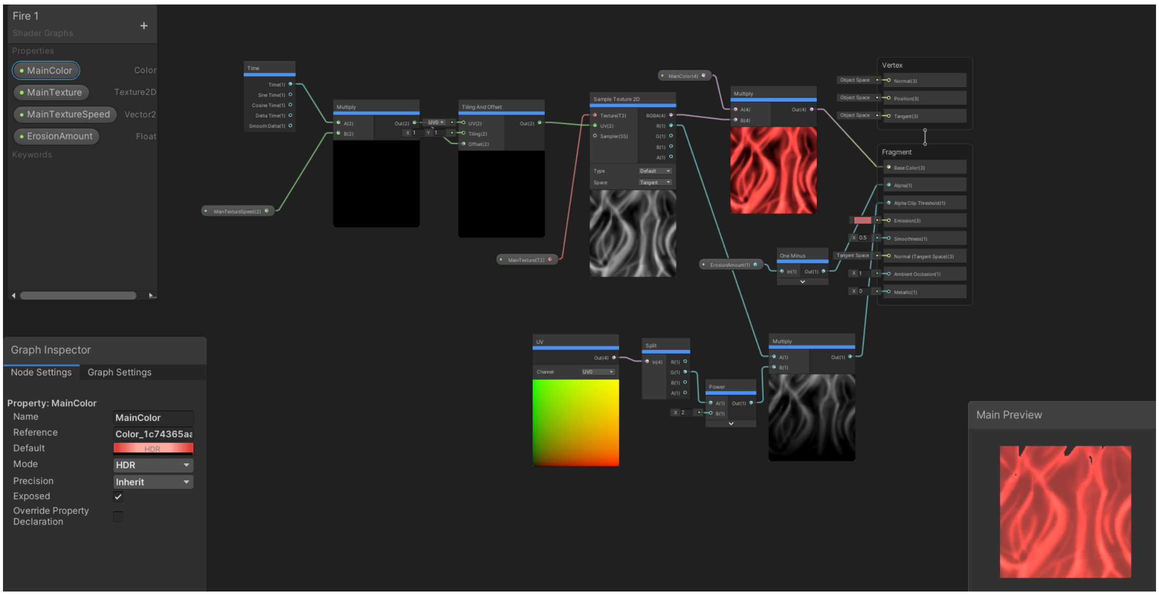
Tiling And Offset Node
Precision: Inherit

Multiply Node
Precision: Inherit

Property: SpeedNormalMap
Name: SpeedNormalMa
Reference: Vector2_edc8c4
Default: X 0 Y 0.05
Precision: Inherit
Exposed: ☑
Override Property Declaration: □

Time Node
Precision: Inherit

The Floral Flame spell uses two different shaders. The first one is a fire that by using the Time node and the Offset node moves the texture made in Clip Studio Paint. Finally with the UV you can get some erosion to make it look more like fire. This shader is used for two different materials, the first one is red and with less erosion and the second one is yellow and with more erosion, this way the yellow is placed in front of the red to give the appearance of fire.

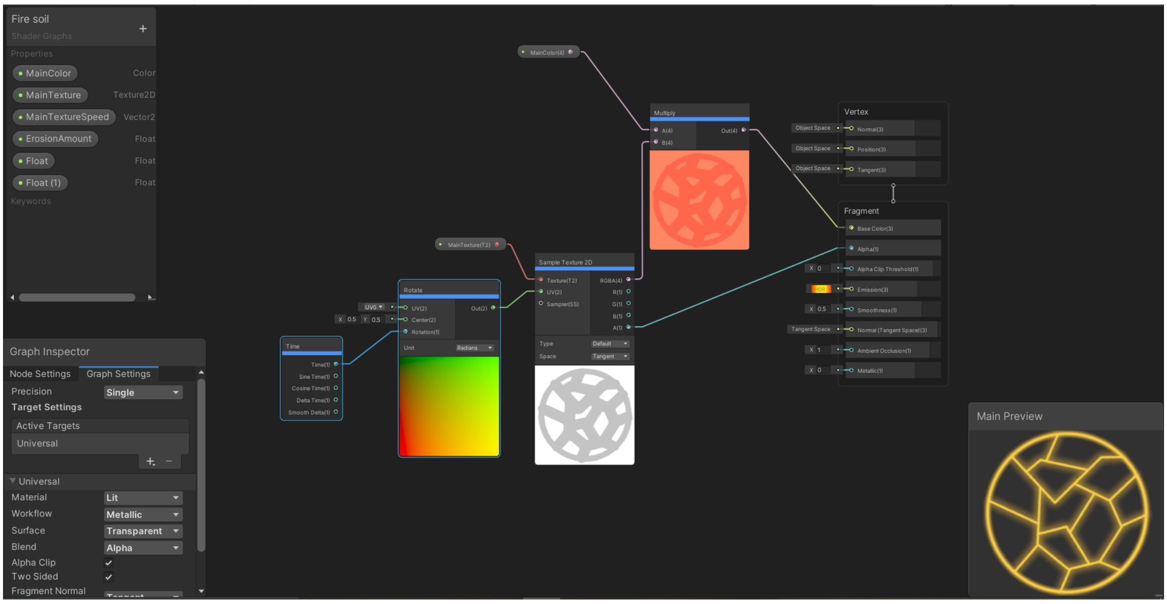


Graph Inspector

Node Settings | Graph Settings

Property: MainColor
Name: MainColor
Reference: Color_1c74365aa
Default: HDR
Mode: HDR
Precision: Inherit
Exposed: ☑
Override Property Declaration: □

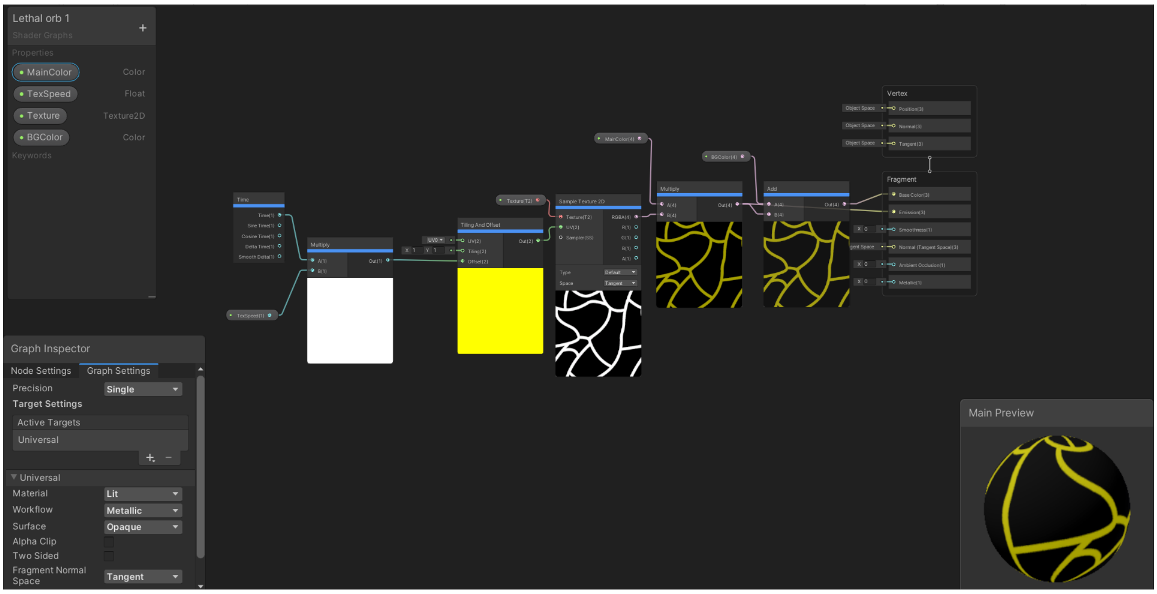
The second shader uses a texture made using Clip Studio and using the Time and Rotate node so that the texture can be rotated without the need to use an animation made in Blender.



MainColor Color
MainTexture Texture2D
MainTextureSpeed Vector2
ErosionAmount Float
Float Float
Float (1) Float

Precision: Single
Target Settings
Active Targets: Universal
Universal
Material: Lit
Workflow: Metallic
Surface: Transparent
Blend: Alpha
Alpha Clip: ✓
Two Sided: ✓
Fragment Normal

The next shader is for the Lethal Orb. Using a texture made in Clip Studio it has been given colour and with the Time node it has been given movement.

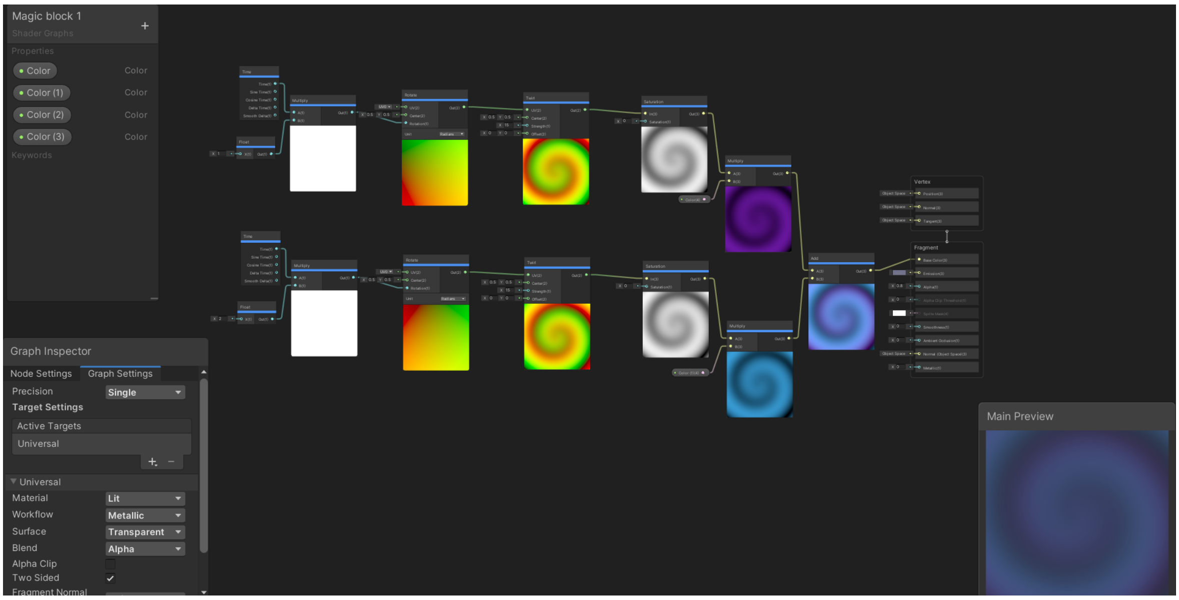


MainColor Color
TexSpeed Float
Texture Texture2D
BGColor Color

Precision: Single
Target Settings
Active Targets: Universal
Universal
Material: Lit
Workflow: Metallic
Surface: Opaque
Alpha Clip: □
Two Sided: □
Fragment Normal Space: Tangent

The next shader is for the magic blocks so that the player can't leave the city in the demo. The graph shader provides a "Twirl" node that makes the twirling effect. So then with a Rotate node from time you can make that magic effect. In order to give the desired colour, the Twirl needed to be left in black and white to multiply it with the requested colour, the way to achieve this effect is to desaturate the Twirl.

Finally, in this shader two similar Twirl effects have been added, the difference is that the top effect rotates slower than the bottom effect (double the difference).



Magic block 1
Shader Graphs

Properties

- Color — Color
- Color (1) — Color
- Color (2) — Color
- Color (3) — Color

Keywords

Graph Inspector

Node Settings | Graph Settings

Precision: Single

Target Settings

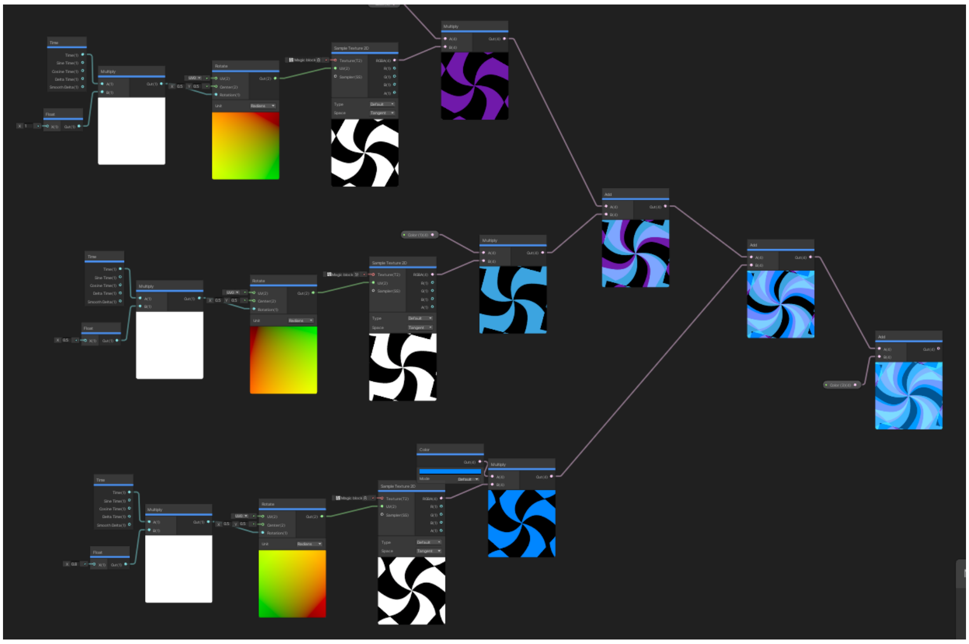
Active Targets
Universal

Universal

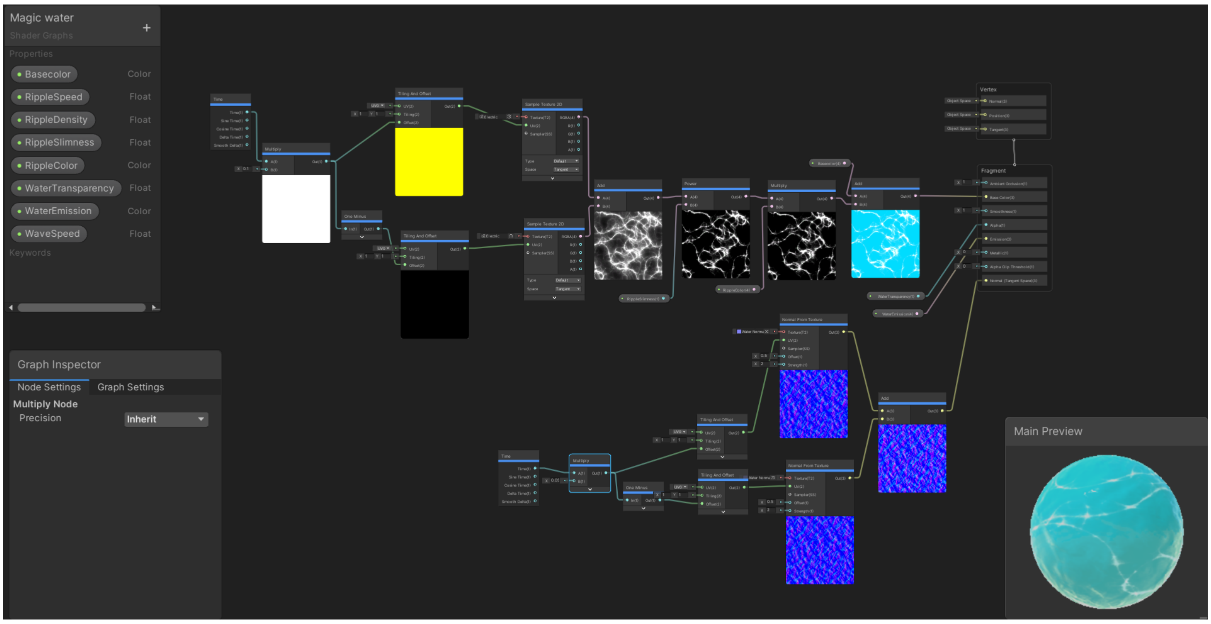
Material: Lit
Workflow: Metallic
Surface: Transparent
Blend: Alpha
Alpha Clip: □
Two Sided: ✓
Fragment Normal

Main Preview

However, this was not originally the case. Before we knew that the Twirl node existed, a texture was made to achieve it but it didn't come out as desired.

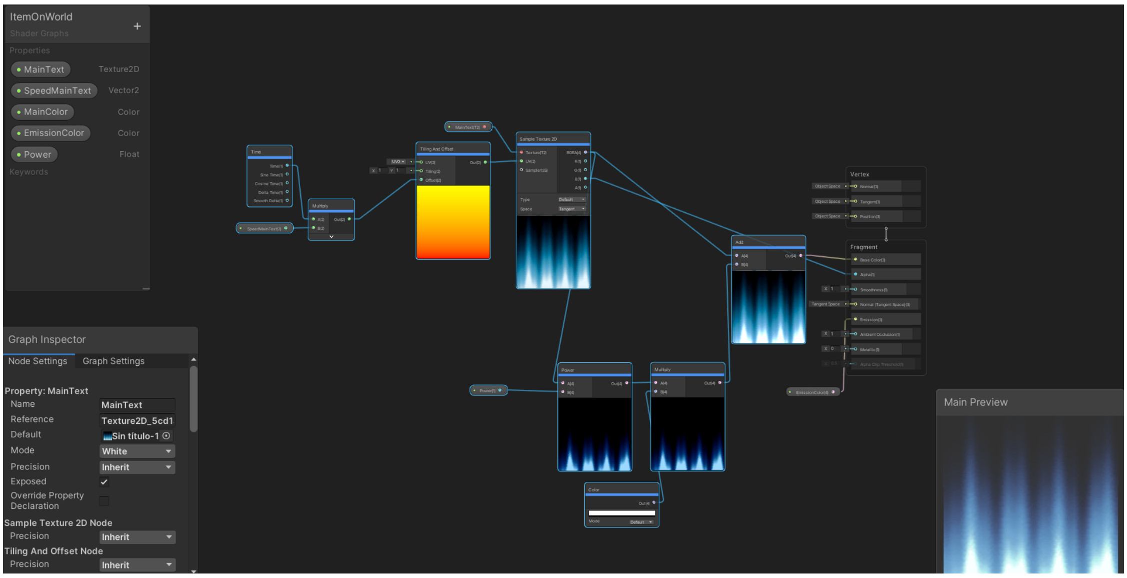


The shader for Magic water is made only for Quick water drop spell. It uses an electric texture found on Google for a magic effect and a normal map also found on google to give it a water effect.


Magic water
Shader Graphs
Properties
Basecolor Color
RippleSpeed Float
RippleDensity Float
RippleSlimness Float
RippleColor Color
WaterTransparency Float
WaterEmission Color
WaveSpeed Float
Keywords

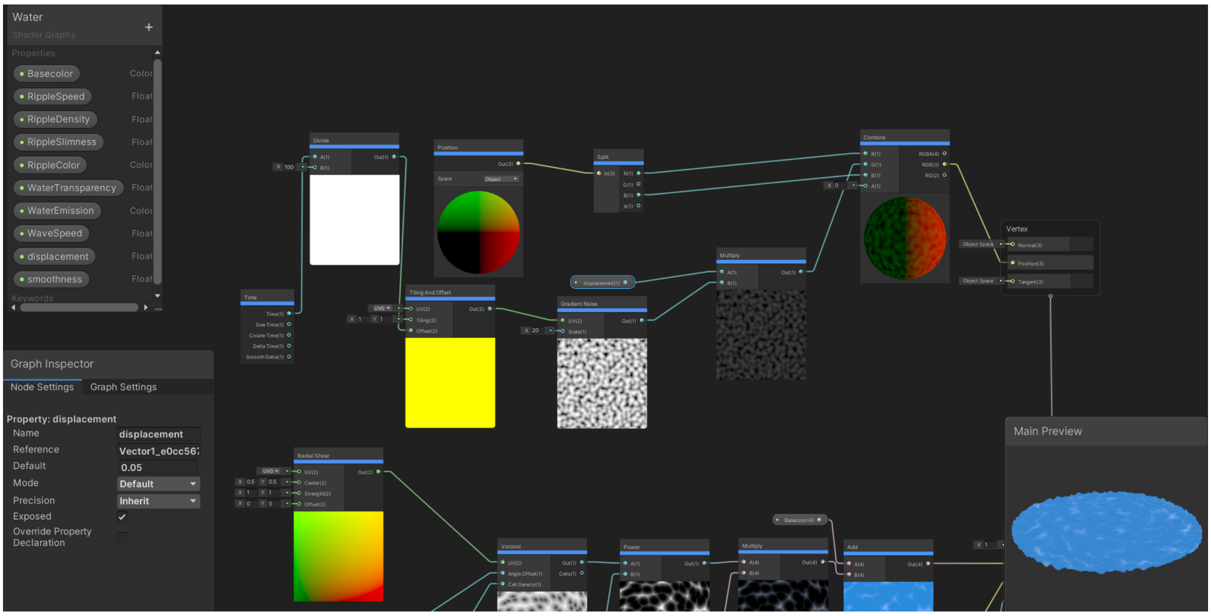
Graph Inspector
Node Settings Graph Settings
Multiply Node
Precision Inherit

This shader is used to mark the objects you can pick up on the world. Basically it uses the offset node to move the texture over time and uses a texture made in photoshop which has been given a gradient and a wave effect.


ItemOnWorld
Shader Graphs
Properties
MainText Texture2D
SpeedMainText Vector2
MainColor Color
EmissionColor Color
Power Float
Keywords

Graph Inspector
Node Settings Graph Settings
Property: MainText
Name MainText
Reference Texture2D_5cd1
Default Sin título-1
Mode White
Precision Inherit
Exposed ☑
Override Property Declaration □
Sample Texture 2D Node
Precision Inherit
Tiling And Offset Node
Precision Inherit

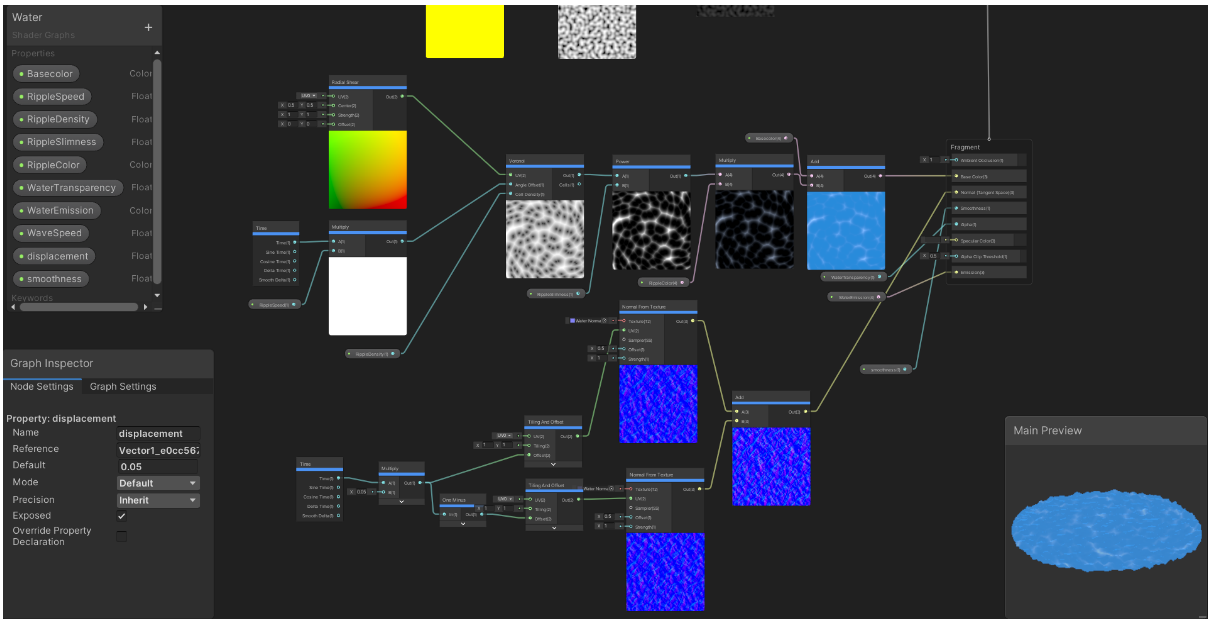
The last shader is the most complex as it makes a more realistic water effect.
The first part shows how a plane has been made to have a displacement effect for a moving water effect. The graph shader provides the gradient noise effect and by multiplying it with the displacement we can get the height to depend on the gradient noise making the darkest parts the lowest and the lightest parts the highest. So then it is combined with the coordinates of the vertices (which have to be split to be able to use them with the Combine node) and finally it is added to the position of the vertices, in this way this node will affect the vertices.



Property: displacement

Name	displacement
Reference	Vector1_e0cc56
Default	0.05
Mode	Default
Precision	Inherit
Exposed	✓
Override Property Declaration	□

The second part is about fragment. It uses a normal map found on Google that gives a water effect. Finally, use a radial shear to give a wave effect, with a voronoi and the power is given to the white parts of the texture.



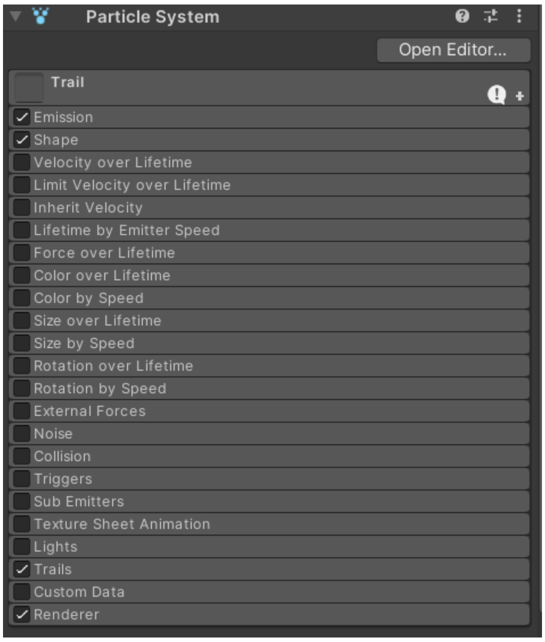
Property: displacement

Name	displacement
Reference	Vector1_e0cc56
Default	0.05
Mode	Default
Precision	Inherit
Exposed	✓
Override Property Declaration	□

Finally, the materials are created from these shaders. It must be taken into account that they depend on the UV of each object, so they must be done correctly so that the shaders can be appreciated.

3.5.2 Particles

Particle systems are meshes or images shown in large numbers, this effect is especially used in video games for magic effects, fire, water, smoke, etc.


Particle System

Open Editor...

Trail

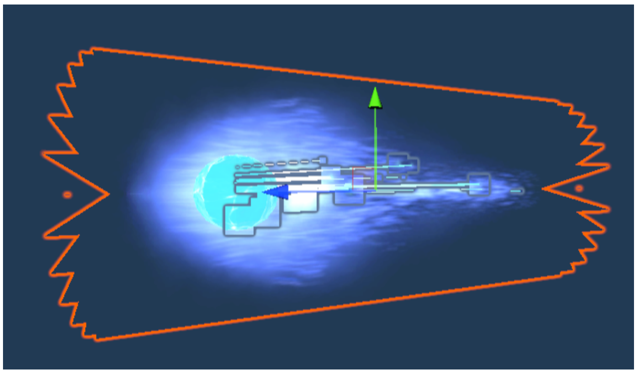
☑ Emission
☑ Shape
☐ Velocity over Lifetime
☐ Limit Velocity over Lifetime
☐ Inherit Velocity
☐ Lifetime by Emitter Speed
☐ Force over Lifetime
☐ Color over Lifetime
☐ Color by Speed
☐ Size over Lifetime
☐ Size by Speed
☐ Rotation over Lifetime
☐ Rotation by Speed
☐ External Forces
☐ Noise
☐ Collision
☐ Triggers
☐ Sub Emitters
☐ Texture Sheet Animation
☐ Lights
☑ Trails
☐ Custom Data
☑ Renderer

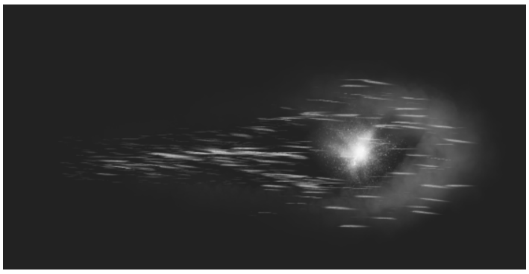
The system emits particles to random positions within a region. Each particle will change (size, colour, rotation, speed, etc.) over its lifetime.

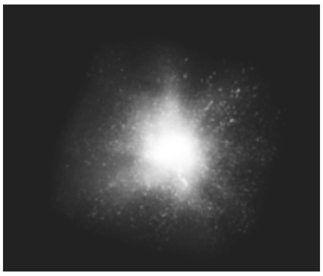
The Universal Render Pipeline offers a Particle graph to create particles using nodes like the Shader graph. However, we didn't want complex particles for this demo, so we used the basic particle system offered by Unity [3].

The first particles of Last Spell are the ones used for the Quick water drop spell. This spell is composed of two different particle systems:

- The first Particle System is the one around the water ball. These particles rotate on a single axis throughout its lifetime, thus creating a moving effect. The texture that this particle uses is made using Clip Studio Paint.
- The second particle system uses a texture which moves along one axis away from the ball to give the effect of speed. In addition, a trail has been added to give the effect of water droplets moving away from the ball.

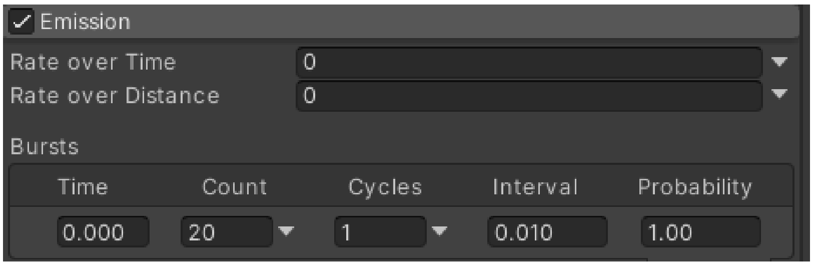






The following Particle System is used to give an effect of hitting enemies. It is composed of two:

- The first System is some balls that are emitted in a sphere applying burst emission since this won't emit the particles from time, but at each interval it will emit x number of particles, x being the count factor, at a time to give burst effect.
- The second particle system is just trails that are also emitted in a sphere randomly with a burst.



☑ Emission

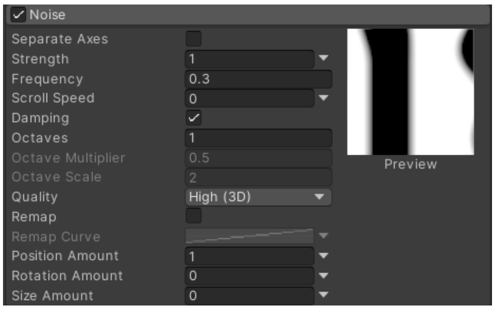
Rate over Time 0

Rate over Distance 0

Bursts

Time	Count	Cycles	Interval	Probability
0.000	20	1	0.010	1.00

The next particle system is a flame to combine with the Floral Flame spell. These particles use a texture made in Clip Studio and use the Noise attribute of the Unity Particle System, so the particles have an irregular movement to look like flames. They also decrease in size depending on their life time.



☑ Noise

Separate Axes ☐

Strength 1

Frequency 0.3

Scroll Speed 0

Damping ☑

Octaves 1

Octave Multiplier 0.5

Octave Scale 2

Quality High (3D)

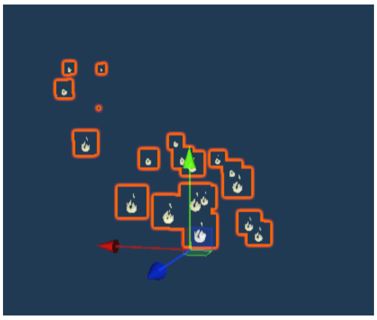
Remap ☐

Remap Curve

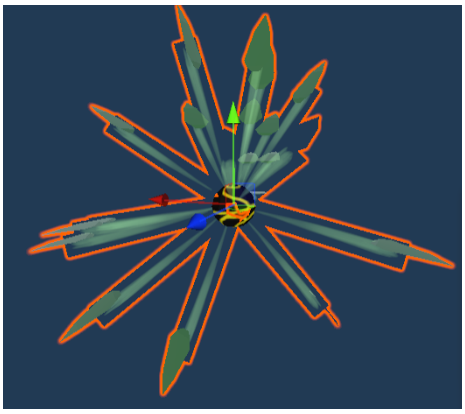
Position Amount 1

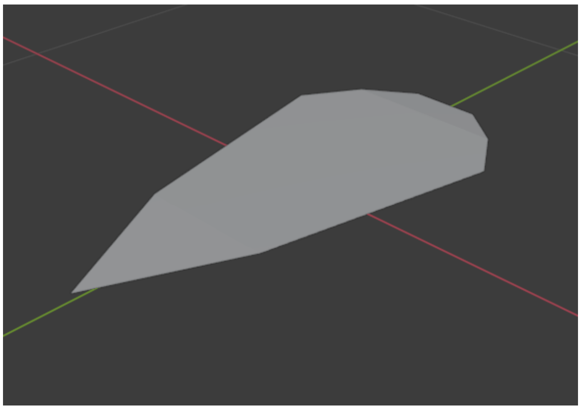
Rotation Amount 0

Size Amount 0



Lastly, the next particle system is the one used for the Lethal orb spell. These particles use a mesh made in Blender which is shaped like a leaf, it also uses the Burst emission and a speed effect is made using a Trail.





3.5.3 Visual quality

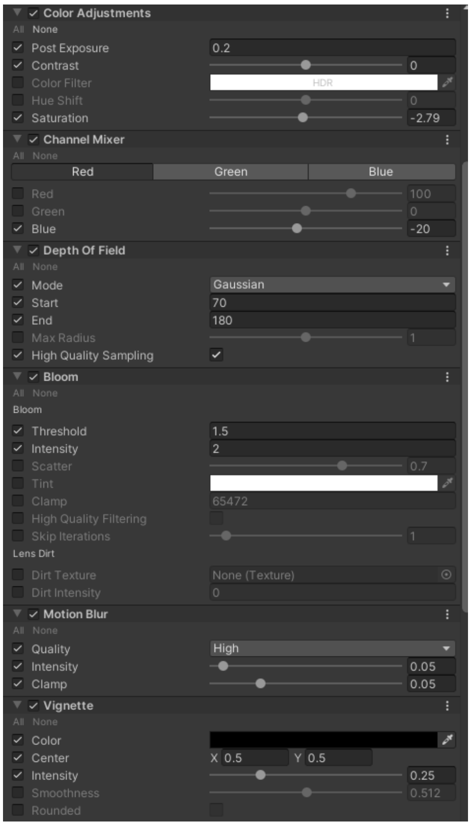
Post-processing [4] is a tool offered by Unity to add effects and improve the appearance of video games.
In order to add it to your project, you just need to place a GameObject in the scene, add a script component called "Volume" and activate post-processing in the game camera.

The "Volume" has two modes:

- Global: it affects the whole scene in the same way, being this mode used in the project.
- Local: a collider is added and only within the collider the post-processing can be seen.

Finally, the attributes that have been added to the Post-processing of the project are:

- Colour adjustments: to adjust the brightness and contrast to make the image softer.
- Channel Mixer: adjusts the balance of each colour where blue is the one that has been increased in the project as it is the most important.
- Depth of Field: applies a blurred effect on the backgrounds and focuses more on the foregrounds.
- Bloom: an effect to increase the brightness of objects that emit light.
- Motion Blur: effect to blur the image when the camera moves.
- Vignette: darkens the edges of the image.
- Shadows, Midtones and Highlights: to adjust the colour and brightness of shadows, midtones and highlights.
- Split Toning: apply two different tones to the image.



☑ Color Adjustments
All None
☑ Post Exposure 0.2
☑ Contrast 0
☐ Color Filter HDR
☐ Hue Shift 0
☑ Saturation -2.79

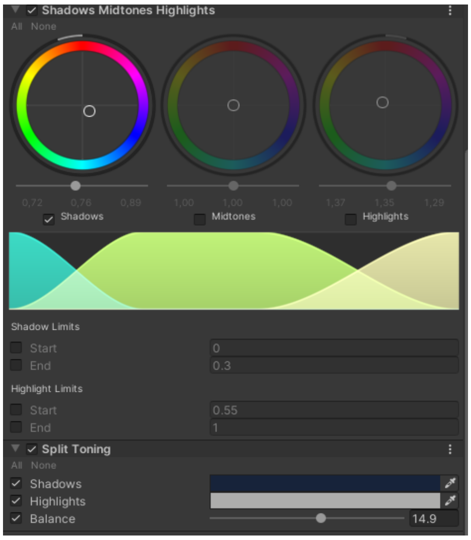
☑ Channel Mixer
All None
Red Green Blue
☐ Red 100
☐ Green 0
☑ Blue -20

☑ Depth Of Field
All None
☑ Mode Gaussian
☑ Start 70
☑ End 180
☐ Max Radius 1
☑ High Quality Sampling ☑

☑ Bloom
All None
Bloom
☑ Threshold 1.5
☑ Intensity 2
☐ Scatter 0.7
☐ Tint
☐ Clamp 65472
☐ High Quality Filtering ☐
☐ Skip Iterations 1
Lens Dirt
☐ Dirt Texture None (Texture)
☐ Dirt Intensity 0

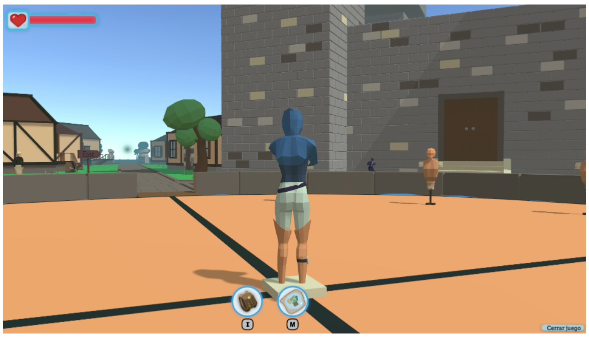
☑ Motion Blur
All None
☑ Quality High
☑ Intensity 0.05
☑ Clamp 0.05

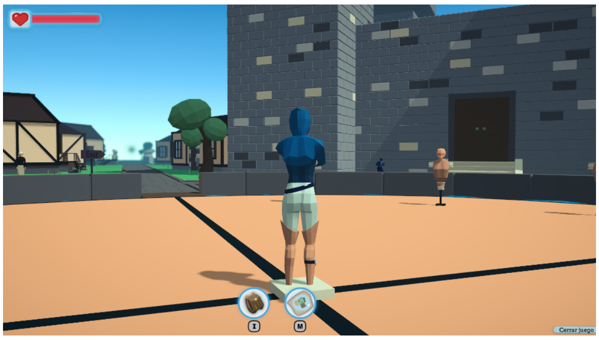
☑ Vignette
All None
☑ Color
☑ Center X 0.5 Y 0.5
☑ Intensity 0.25
☐ Smoothness 0.512
☐ Rounded ☐



☑ Shadows Midtones Highlights
All None
0,72 0,76 0,89 ☑ Shadows
1,00 1,00 1,00 ☐ Midtones
1,37 1,35 1,29 ☐ Highlights
Shadow Limits
☐ Start 0
☐ End 0.3
Highlight Limits
☐ Start 0.55
☐ End 1

☑ Split Toning
All None
☑ Shadows
☑ Highlights
☑ Balance 14.9

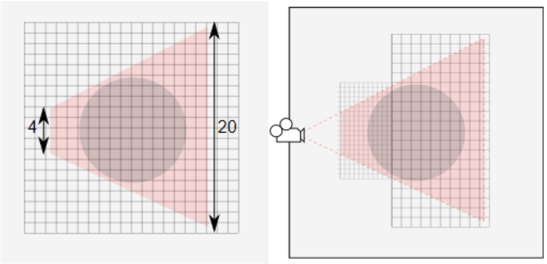




As for the graphics quality of the game, it is set to the maximum because as it is a demo, it can afford a better quality. For this reason things like Antialiasing, shadow resolution and even Shadow Cascades have been added.

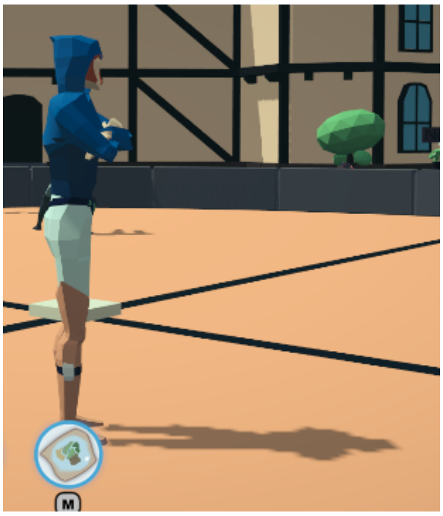
Shadow cascades help to solve the problem called "perspective aliasing" which makes shadows look pixelated when they are close to the camera and have directional lights close to the camera.

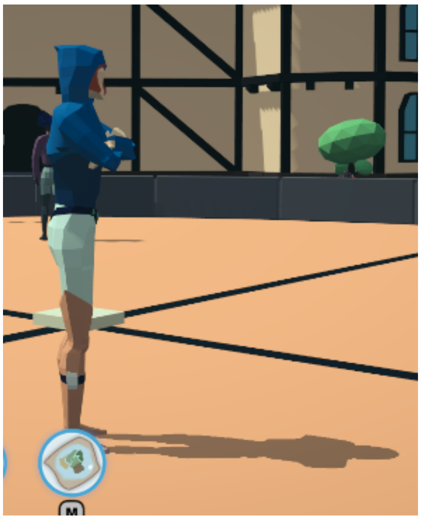
Perspective aliasing happens when a single directional light illuminates the entire scene, making the shadow map too large. Then, the shadow map is disproportionately scaled by having a camera in perspective mode. As shown in the image below, the end of the shadow map covers 20 pixels and the closest point only 4 pixels, but both sizes appear as the same size on screen.

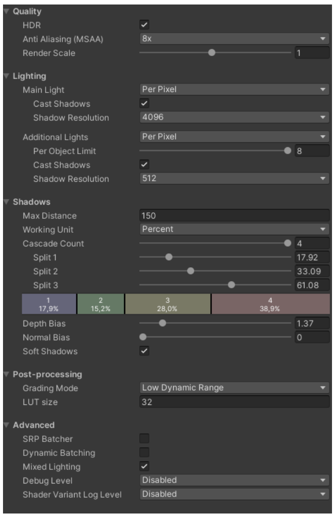


The Shadow cascades [9] divide this area into several parts to reduce the areas closest to the camera.

In the following images you can see how the one on the left uses 1 cascade and the one on the right uses 4 cascades. As previously mentioned, this demo can afford higher quality, so 4 shadow cascades have been added, which is the maximum that the Universal Render Pipeline allows.

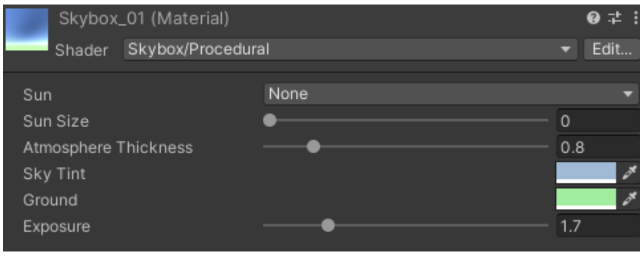






Quality	
HDR	☑
Anti Aliasing (MSAA)	8x
Render Scale	1
Lighting	
Main Light	Per Pixel
Cast Shadows	☑
Shadow Resolution	4096
Additional Lights	Per Pixel
Per Object Limit	8
Cast Shadows	☑
Shadow Resolution	512
Shadows	
Max Distance	150
Working Unit	Percent
Cascade Count	4
Split 1	17.92
Split 2	33.09
Split 3	61.08
1 17,9% · 2 15,2% · 3 28,0% · 4 38,9%	
Depth Bias	1.37
Normal Bias	0
Soft Shadows	☑
Post-processing	
Grading Mode	Low Dynamic Range
LUT size	32
Advanced	
SRP Batcher	□
Dynamic Batching	□
Mixed Lighting	☑
Debug Level	Disabled
Shader Variant Log Level	Disabled

As for the environment, a Unity Skybox has been selected (although the attributes have been changed) to involve the whole scene so that players can see further.

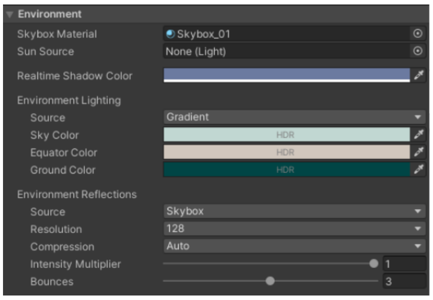


Skybox_01 (Material)
Shader Skybox/Procedural
Edit...

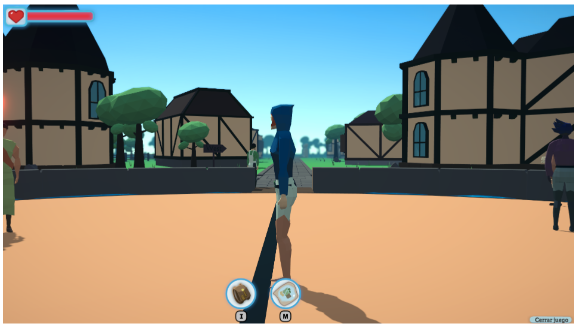
Sun: None
Sun Size: 0
Atmosphere Thickness: 0.8
Sky Tint
Ground
Exposure: 1.7

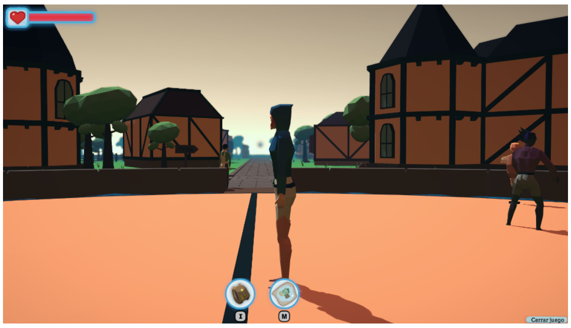
Secondly, the environment lighting helps to give different tones to the environment, so we can get a sunrise or sunset by changing its attributes. The environment lighting has three modes:

- Skybox: the environment lighting depends on the colours of the selected Skybox.
- Gradient: the environment lighting has three different colours: one for the sky, one for the horizon and one for the ground. This mode is the one used in the project.
- Colour: the ambient light is only one colour that will affect the whole scene equally.



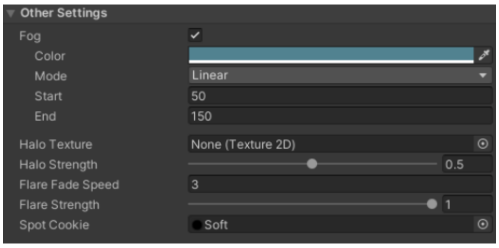
Environment
Skybox Material: Skybox_01
Sun Source: None (Light)
Realtime Shadow Color
Environment Lighting
Source: Gradient
Sky Color: HDR
Equator Color: HDR
Ground Color: HDR
Environment Reflections
Source: Skybox
Resolution: 128
Compression: Auto
Intensity Multiplier: 1
Bounces: 3





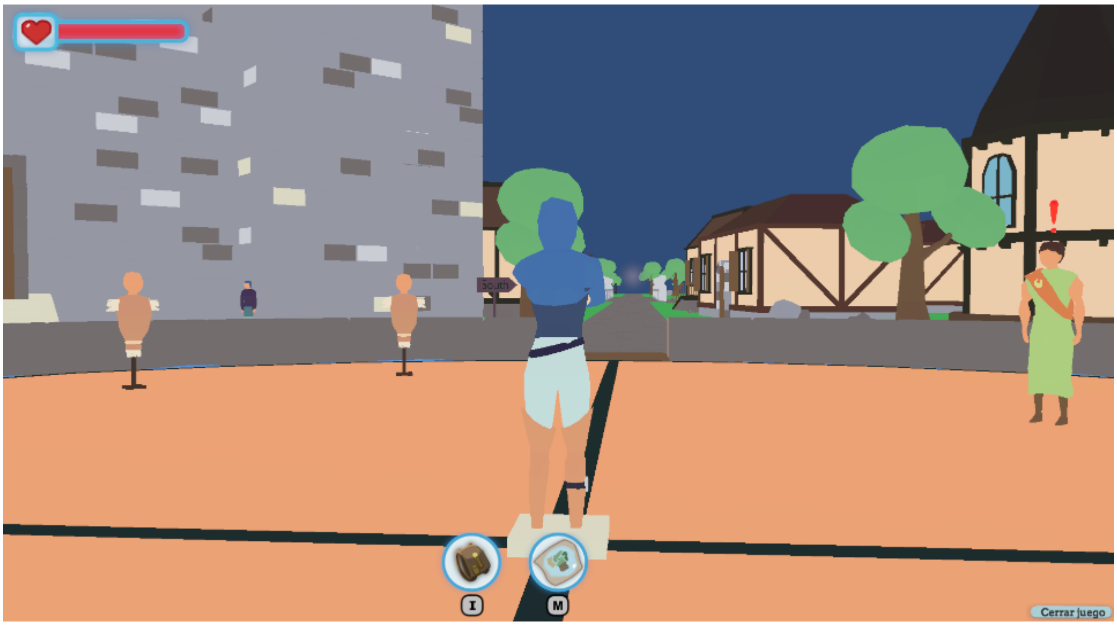
In addition, Unity allows a fog to be added to the background of the scene to emphasise depth. This fog has three modes:

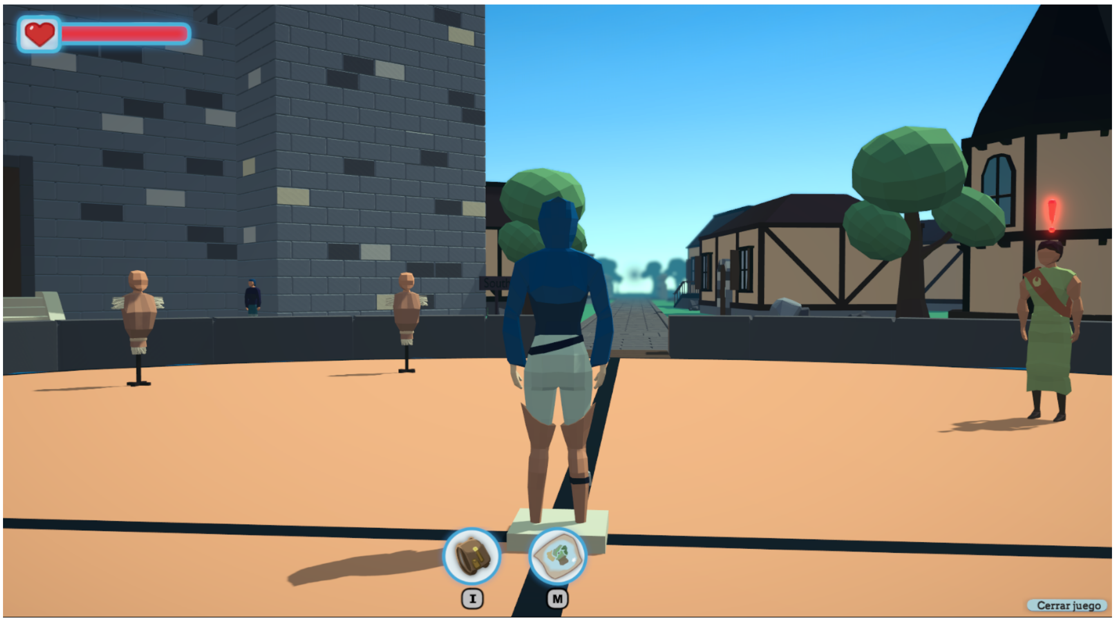
- Linear: the density of the fog increases linearly with distance. You only have to point the distance from where the fog starts to the camera and where the GameObjects in the scene get darker. This mode is the one used in the project as it allowed more control over the fog.
- Exponential: the fog density increases exponentially with distance.
- Exponential squared: fog density increases rapidly with distance.



Other Settings
Fog ☑
Color
Mode: Linear
Start: 50
End: 150
Halo Texture: None (Texture 2D)
Halo Strength: 0.5
Flare Fade Speed: 3
Flare Strength: 1
Spot Cookie: Soft

So, with all these components it has been possible to improve the visual quality of the demo. Below is the difference between not having any of these components activated and having them activated.



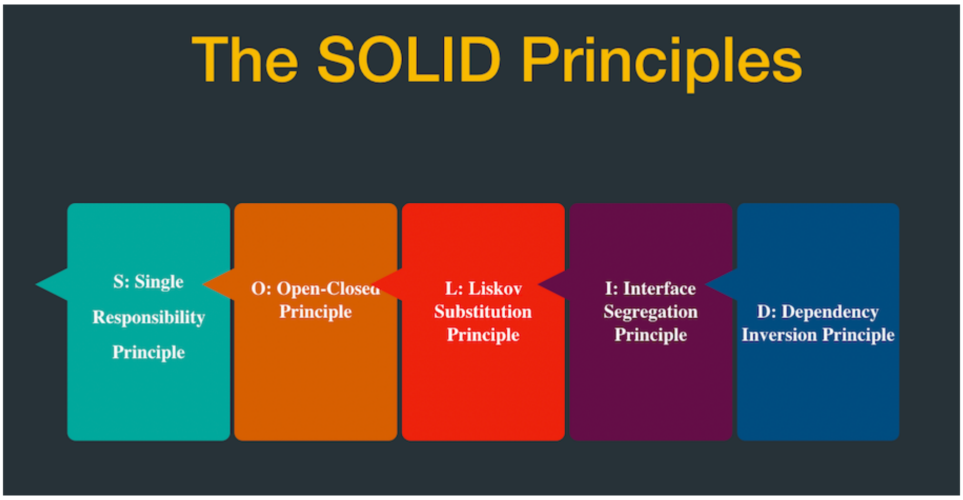


4. CODE

The Final Spell has been programmed following the SOLID principles [10], in this way it has been programmed in an efficient and understandable way. The language used is C# using Visual Studio 2019 IDE, and several diagrams have been designed to explain the functionality of each system by David Gómez.

The SOLID principles that The Final Spell follows are:

- Single responsibility principle: every class should have only one use
- Open-closed principles: entities should be open for extension but closed for modification
- Liskov substitution principle: objects in a program should be replaceable with instances of their subtypes without altering the quality of the program
- Interface segregation principle: many client-specific interfaces are better than one general-purpose interface
- Dependency inversion principle: depend upon abstractions, not concretions



The SOLID Principles

S: Single Responsibility Principle

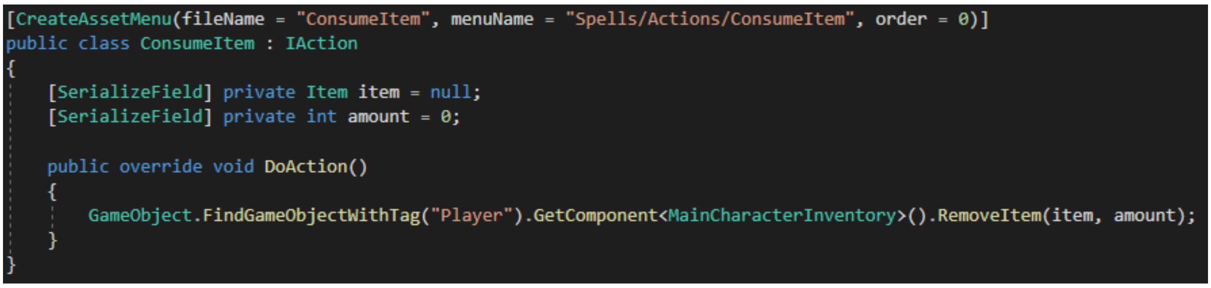
O: Open-Closed Principle

L: Liskov Substitution Principle

I: Interface Segregation Principle

D: Dependency Inversion Principle

In addition, David made scriptable objects that help to store information, so other GameObjects can access that information without having to instantiate the object within a scene. This is useful for making lists of objects, which we only have the code base and need to change the properties of the base to add different objects, and we can easily modify objects.



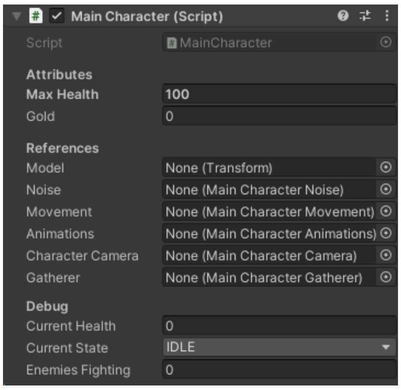
```
[CreateAssetMenu(fileName = "ConsumeItem", menuName = "Spells/Actions/ConsumeItem", order = 0)]
public class ConsumeItem : IAction
{
    [SerializeField] private Item item = null;
    [SerializeField] private int amount = 0;

    public override void DoAction()
    {
        GameObject.FindGameObjectWithTag("Player").GetComponent<MainCharacterInventory>().RemoveItem(item, amount);
    }
}
```

Finally, all GameObject follows the same structure to facilitate debugging without programming:

- Attributes header where customizable variables are displayed
- References header where dependencies are stored (loaded on game start)
- Debug header where not editable variables are displayed

4.1 Main character

The main character has the most complex system as it is the one that can interact with the world, fight, run, shop, etc. All components (movement, animations, camera, spells, noise and gatherer) of the main character are connected to a parent component called MainCharacter, except the spells.

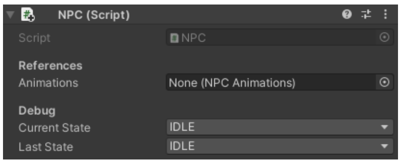


Main Character (Script)

Script	MainCharacter
Attributes	
Max Health	100
Gold	0
References	
Model	None (Transform)
Noise	None (Main Character Noise)
Movement	None (Main Character Movement)
Animations	None (Main Character Animations)
Character Camera	None (Main Character Camera)
Gatherer	None (Main Character Gatherer)
Debug	
Current Health	0
Current State	IDLE
Enemies Fighting	0

4.2 NPCs

The Final Spell is full of NPCs, but each has its own purpose: some have quests to give the player, some help the player, and some are just there to fill out the world. There are three main types of NPCs: quest-givers, merchants and sorcerers and sorceresses; and all three can have routes and dialogues.



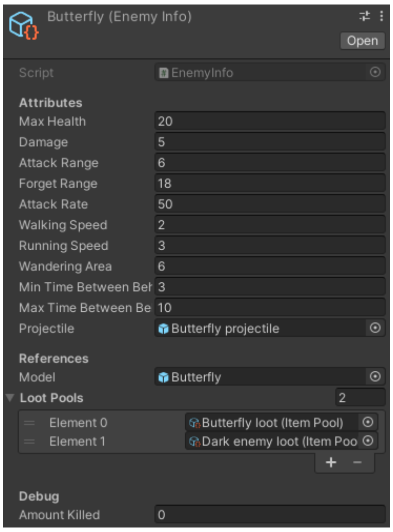
NPC (Script)

Script	NPC
References	
Animations	None (NPC Animations)
Debug	
Current State	IDLE
Last State	IDLE

4.3 Enemies

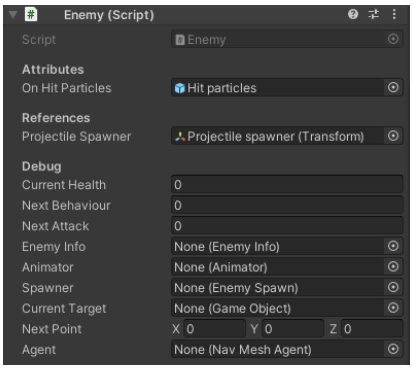
Following SOLID principles, the information of each enemy is stored in Scriptable objects. This information is used for the Enemy component and the EnemySpawn component. The Enemy component controls the behaviour of the enemies, it focuses on running all the information in the Scriptable object.

In addition, all enemies behave through the use of Unity's built-in pathfinding system. The navigation mesh is implemented in a way that enemies cannot leave the areas they are supposed to be in, like butterflies in the plains.



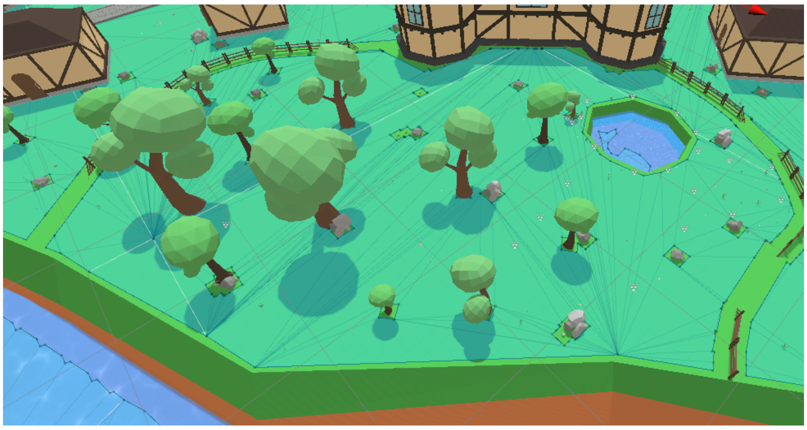
Butterfly (Enemy Info)

Script	EnemyInfo
Attributes	
Max Health	20
Damage	5
Attack Range	6
Forget Range	18
Attack Rate	50
Walking Speed	2
Running Speed	3
Wandering Area	6
Min Time Between Bel	3
Max Time Between Be	10
Projectile	Butterfly projectile
References	
Model	Butterfly
Loot Pools	2
Element 0	Butterfly loot (Item Pool)
Element 1	Dark enemy loot (Item Poo
Debug	
Amount Killed	0



Enemy (Script)

Script	Enemy
Attributes	
On Hit Particles	Hit particles
References	
Projectile Spawner	Projectile spawner (Transform)
Debug	
Current Health	0
Next Behaviour	0
Next Attack	0
Enemy Info	None (Enemy Info)
Animator	None (Animator)
Spawner	None (Enemy Spawn)
Current Target	None (Game Object)
Next Point	X 0 Y 0 Z 0
Agent	None (Nav Mesh Agent)



5. MUSIC

The music of The Final Spell is created with the help of Manel Espinosa. Two songs have been made by him: a default theme song for the whole video game, which is relaxing, and the other one when the player starts a battle with an enemy, with an epic tone.

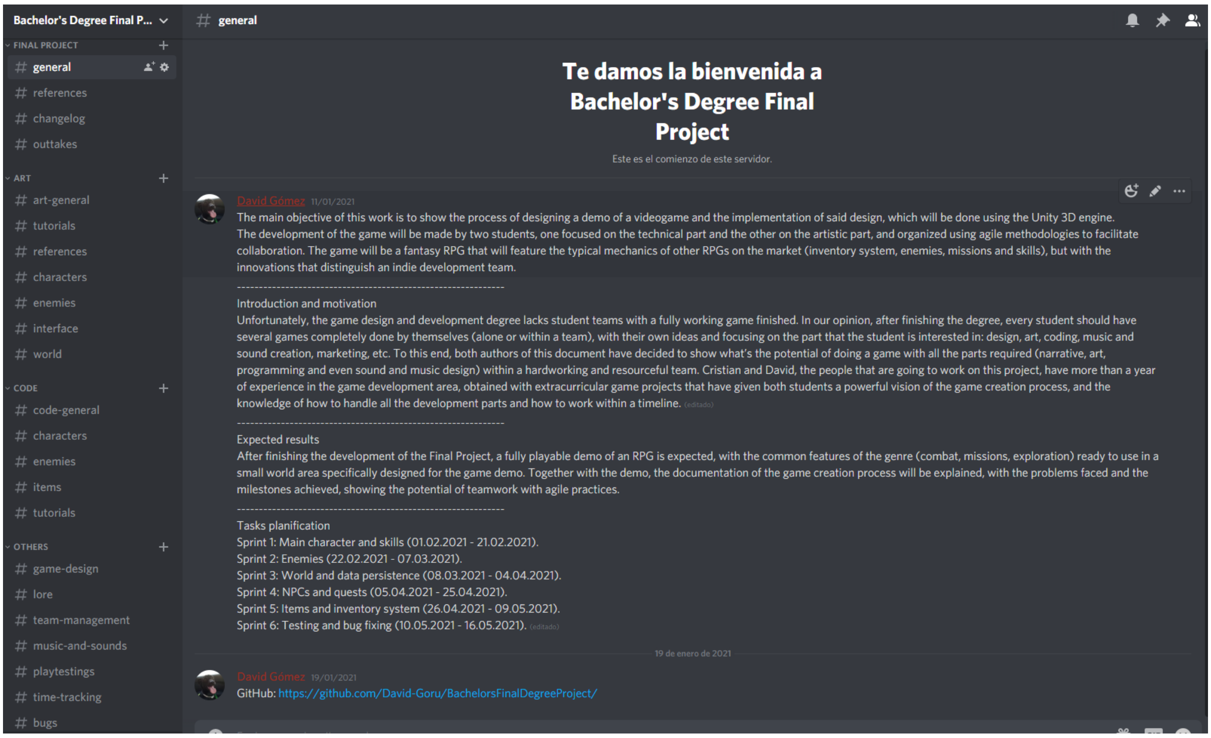
Both songs have a medieval style as this is the era in which the world of The Final Spell is set. The battle song is called "The Great Fire", in reference to the lore of the video game.

These songs are based on other RPGs such as Valheim, The Witcher and World of Warcraft. The ambient music of Azeroth from World of Warcraft has been a big influence on the music of The Final Spell.

The music can be downloaded from the GitHub repository (section 10).

6. TEAM ORGANIZATION

The Final Spell was developed in collaboration with David Gómez who focused on the programming part. The most used tool for communication was a Discord server where everyone published what they were doing during the four months of development.



Te damos la bienvenida a Bachelor's Degree Final Project

Este es el comienzo de este servidor.

David Gómez 11/01/2021

The main objective of this work is to show the process of designing a demo of a videogame and the implementation of said design, which will be done using the Unity 3D engine.
The development of the game will be made by two students, one focused on the technical part and the other on the artistic part, and organized using agile methodologies to facilitate collaboration. The game will be a fantasy RPG that will feature the typical mechanics of other RPGs on the market (inventory system, enemies, missions and skills), but with the innovations that distinguish an indie development team.

---

Introduction and motivation
Unfortunately, the game design and development degree lacks student teams with a fully working game finished. In our opinion, after finishing the degree, every student should have several games completely done by themselves (alone or within a team), with their own ideas and focusing on the part that the student is interested in: design, art, coding, music and sound creation, marketing, etc. To this end, both authors of this document have decided to show what's the potential of doing a game with all the parts required (narrative, art, programming and even sound and music design) within a hardworking and resourceful team. Cristian and David, the people that are going to work on this project, have more than a year of experience in the game development area, obtained with extracurricular game projects that have given both students a powerful vision of the game creation process, and the knowledge of how to handle all the development parts and how to work within a timeline. (editado)

---

Expected results
After finishing the development of the Final Project, a fully playable demo of an RPG is expected, with the common features of the genre (combat, missions, exploration) ready to use in a small world area specifically designed for the game demo. Together with the demo, the documentation of the game creation process will be explained, with the problems faced and the milestones achieved, showing the potential of teamwork with agile practices.

---

Tasks planification
Sprint 1: Main character and skills (01.02.2021 - 21.02.2021).
Sprint 2: Enemies (22.02.2021 - 07.03.2021).
Sprint 3: World and data persistence (08.03.2021 - 04.04.2021).
Sprint 4: NPCs and quests (05.04.2021 - 25.04.2021).
Sprint 5: Items and inventory system (26.04.2021 - 09.05.2021).
Sprint 6: Testing and bug fixing (10.05.2021 - 16.05.2021). (editado)

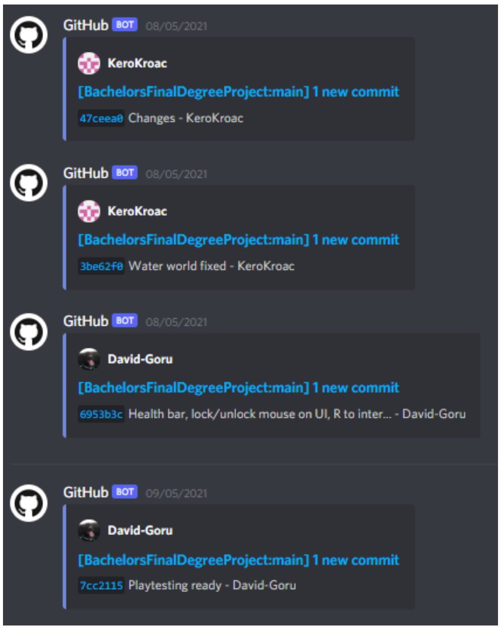
19 de enero de 2021

David Gómez 19/01/2021

GitHub: https://github.com/David-Goru/BachelorsFinalDegreeProject/

The Discord channel was divided into four sections:

- Final project: in this section were general things like game references, changelog and outtakes.
- Art: this section are all the parts that have to do with art: the art-general (explained later), tutorials (shaders, particles, blender, post-processing, etc.), references, characters, enemies, interface and stuff from the world of The Final Spell.
- Code: this section contains everything related to the programming of the project such as characters, enemies, items and tutorials such as how to use the camera.
- Others: this section focuses on the organisation of the game like design, lore, team management, music, playtester's comments, everyone's timer on the project and bugs that could be found in The Final Spell.



GitHub 08/05/2021
KeroKroac
[BachelorsFinalDegreeProject:main] 1 new commit
47ceea0 Changes - KeroKroac

GitHub 08/05/2021
KeroKroac
[BachelorsFinalDegreeProject:main] 1 new commit
3be62f0 Water world fixed - KeroKroac

GitHub 08/05/2021
David-Goru
[BachelorsFinalDegreeProject:main] 1 new commit
6953b3c Health bar, lock/unlock mouse on UI, R to inter... - David-Goru

GitHub 09/05/2021
David-Goru
[BachelorsFinalDegreeProject:main] 1 new commit
7cc2115 Playtesting ready - David-Goru

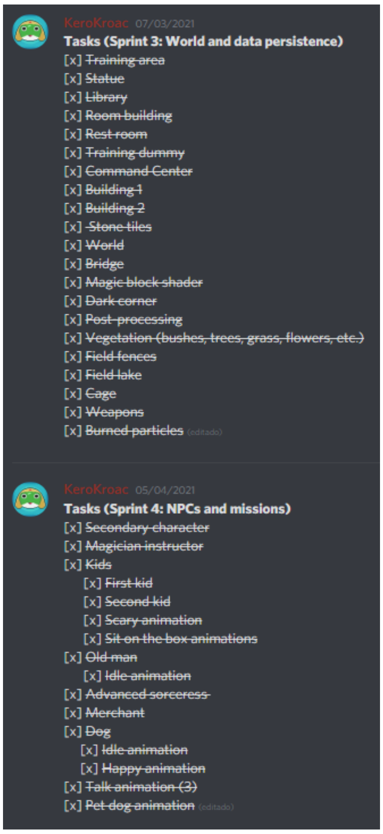
The Changelog channel is very useful for the team to see everyone's changes easily. Github is the tool used for the team to share the project. So then, in this channel there was a GitHub bot that sent a message every time one of the two made a commit of the changes made. This message specified the name of the committer and the title given in the commit.

6.1 Agile methodology

The project has been developed through sprints. In each sprint there was a meeting between the two developers to share their ideas, goals and ideas. At the end, the Discord art-general and code-general channels were updated, which were used to share the tasks to each other in a simple Kanban way.

In each sprint a part of the project was completed in a time set by the students depending on the work they can carry out in each sprint, these are:

- Sprint 1: Main character and skills (01.02.2021 - 21.02.2021).
- Sprint 2: Enemies (22.02.2021 - 07.03.2021).
- Sprint 3: World and data persistence (08.03.2021 - 04.04.2021).
- Sprint 4: NPCs and quests (05.04.2021 - 25.04.2021).
- Sprint 5: Items and inventory system (26.04.2021 - 09.05.2021).
- Sprint 6: Testing and bug fixing (10.05.2021 - 16.05.2021).



KeroKroac 07/03/2021
Tasks (Sprint 3: World and data persistence)
[x] ~~Training area~~
[x] ~~Statue~~
[x] ~~Library~~
[x] ~~Room building~~
[x] ~~Rest room~~
[x] ~~Training dummy~~
[x] ~~Command Center~~
[x] ~~Building 1~~
[x] ~~Building 2~~
[x] ~~Stone tiles~~
[x] ~~World~~
[x] ~~Bridge~~
[x] ~~Magic block shader~~
[x] ~~Dark corner~~
[x] ~~Post-processing~~
[x] ~~Vegetation (bushes, trees, grass, flowers, etc.)~~
[x] ~~Field fences~~
[x] ~~Field lake~~
[x] ~~Cage~~
[x] ~~Weapons~~
[x] ~~Burned particles~~ (editado)

KeroKroac 05/04/2021
Tasks (Sprint 4: NPCs and missions)
[x] ~~Secondary character~~
[x] ~~Magician instructor~~
[x] ~~Kids~~
    [x] ~~First kid~~
    [x] ~~Second kid~~
    [x] ~~Scary animation~~
    [x] ~~Sit on the box animations~~
[x] ~~Old man~~
    [x] ~~Idle animation~~
[x] ~~Advanced sorceress~~
[x] ~~Merchant~~
[x] ~~Dog~~
    [x] ~~Idle animation~~
    [x] ~~Happy animation~~
[x] ~~Talk animation (3)~~
[x] ~~Pet dog animation~~ (editado)

6.2 Playtestings

The Final Spell has been constantly being playtested by the two developers, but also other people have helped with two playtestings.

The first playtesting was done early in development, at the end of the second sprint so that the player can attack enemies and test the spells. The second playtesting was done at the end of the fifth sprint so that players could test the whole world.

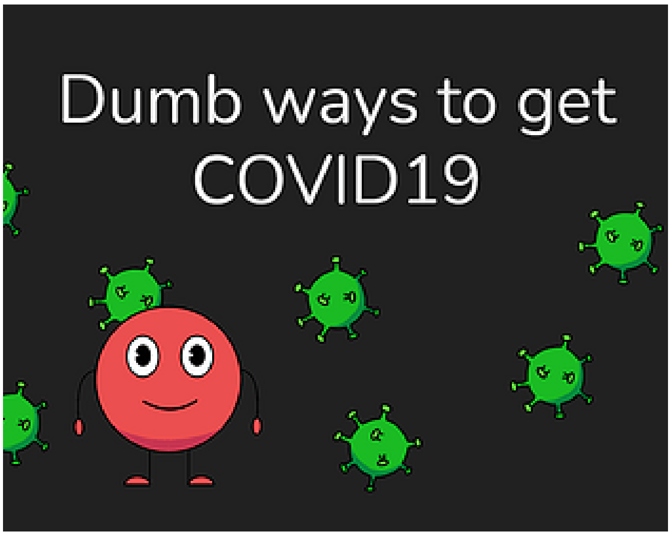
At the end of each playtesting, people could fill out a survey to send their ideas or problems with the game.

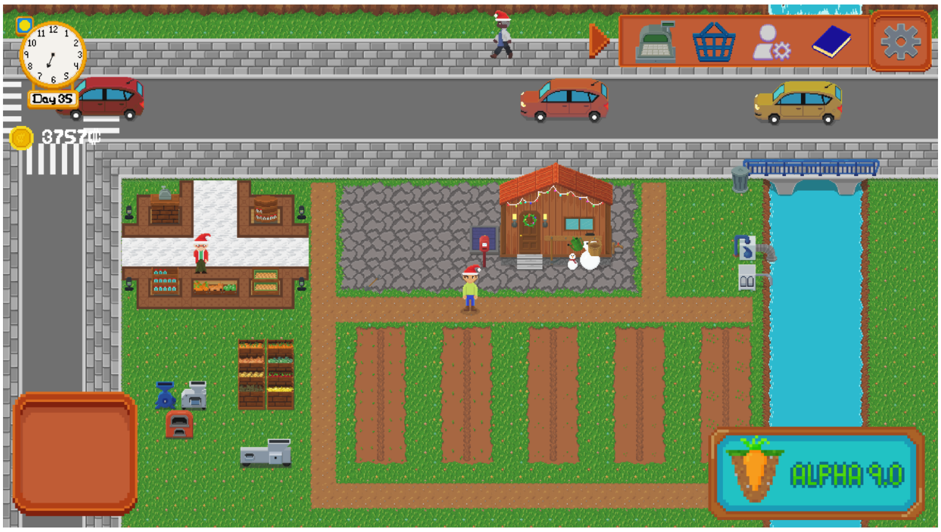
	FIRST PLAYTESTING	SECOND PLAYTESTING
PROBLEMS	➢ Quick water drop gets stuck in second animation / Spells stop working after a while ➢ Camera is hard to control / Camera's sensibility is too high ➢ Floral flame weird final state when finishing the spell	➢ When open the inventory then press right click, the character is bugged in attack position ➢ Colliders don't work well ➢ The character runs without animations when press fast the shift button ➢ Can't click "close game" button in

		the final screen ➢ Camera controls are chaotic ➢ Character go through the castle stairs
THINGS TO CHANGE	➢ Controls having buttons icons ➢ Feedback when dropping/picking up loot ➢ More floral flame range ➢ Avoid using the shift for spells ➢ Butterfly shots going straight forward instead of having a target ➢ Spells book going to next page when clicking half right area/previous page when clicking half left area ➢ Change the font used on the spells book to avoid 'cl' becoming 'd'. ➢ Jump action ➢ More controls for controlling the camera	➢ Capsule colliders don't make movement so difficult ➢ Camera don't go through objects or the character can be seen ➢ Signs should be more brighter ➢ The player can decrease or raise the volume of the music ➢ Know the spells from the beginning - Add a mini map ➢ Add a dodge animation ➢ Pass dialogues with enter or space button
DON'T LIKE	➢ Quick water drop needs feedback displaying where's going to fall ➢ Close-range spells don't have much power, so cost/efficacy relation is unbalanced ➢ Camera rotation is confusing ➢ Models have wrong shades due to the lights	➢ Controls are confusing ➢ Can't redirect the water attack ➢ More places where the character can crouch ➢ The change of the music ➢ The basic fight system can be more flexible ➢ Don't know how many pages are in the spells book ➢ Characters don't repeat what to do in the quests
LIKE	➢ Visual aspect of the game ➢ Models ➢ Spells' VFX ➢ Animations ➢ Fluent movement + camera ➢ Every spell has a unique way of use ➢ Picking up items ➢ Spells book use ➢ Mix between existing animals (purple pigs with deer horns)	➢ Visual aspect ➢ Can change the camera angle ➢ NPCs animations ➢ Missions ➢ Music ➢ Merchant interface ➢ Pet the dog ➢ Hidden achievements ➢ Water shader ➢ Change of music ➢ Main character crossed arms animation

6.3 About us

We have also been working on several projects for a couple of years. There have been gamejam projects, videogames for the degree like Dumb Ways to Get COVID19 [11] (The subject of Fundamentals of didactic videogame design) or our most developed videogame Vivaio [12], which we have been working on for two years and publishing several alphas so that they can be tested.





7. RESULTS

After four months we have been able to complete everything we originally set out to do. The Final Spell demo is complete with its world, characters, enemies, objects, dialogues, etc. This demo can be downloaded in the attachment at the end of the document.

All the tasks of the sprints have been completed on time with the agile methodology originally proposed. In addition, it has been possible to complete more tasks than originally planned, like using post-processing to improve the visual quality.

Thanks to the surveys we have been able to get some feedback in order to improve the demo.

The Final Spell demo contains: one character to interact with the world, four main quests, one secondary quest, two hidden achievements, three types of enemies, several NPCs (merchant, secondary main character, trainer, dogs, etc.), several buildings (the library is the only building modelled inside), several world decorations (trees, houses, plants, rocks, etc.), several exploration places like plains, a maze, a cage, the forbidden nook and the training area.

8. CONCLUSIONS

When I started this project I had not done many things in 3D, right now I feel that I have learned many things that I will be able to continue improving over time and practice.

Also, I have realised that 3D models are not everything in the art of a 3D video game. I have learned how to make a different lighting and visual quality in order to improve the visual quality of a videogame, which I will use for future projects.

In addition, thanks to agile methodologies, we have had no problems in the design of the videogame. I'm sure I'll use these techniques in future projects so I won't have problems in team work.

We have had several meetings with Diego Díaz, our supervisor, who has helped us with the whole process of The Final Spell, being the closest thing to a project manager.

Finally, it can be said that we have a complete demo of an RPG video game with quite interesting features thanks to the support of agile methodologies.

9. BIBLIOGRAPHY

1. **Blender**. https://www.blender.org/
2. **Shader Graph**. https://docs.unity3d.com/Packages/com.unity.shadergraph@12.0/manual/Getting-Started.html
3. **Unity particles**. https://docs.unity3d.com/Manual/ParticleSystems.html
4. **Post-processing**. https://docs.unity3d.com/Manual/PostProcessingOverview.html
5. **Clip Studio**. https://www.clipstudio.net/es/
6. **Discord**. https://discord.com/brand-new
7. **Github**. https://github.com/
8. **Low-poly**. https://www.invergestudios.com/blog/el-arte-low-poly/
9. **Shadow cascades**. https://docs.unity3d.com/Manual/shadow-cascades.html
10. **SOLID Principles**. https://www.digitalocean.com/community/conceptual_articles/s-o-l-i-d-the-first-five-principles-of-object-oriented-design
11. **Dumb Ways to Get COVID19**. https://wedogames-studio.itch.io/dumb-ways-to-get-covid-19
12. **Vivaio**. https://vivaio.itch.io/game

10. GITHUB REPOSITORY

A playable demo, a video gameplay, the music and the source code of the project can be found at https://github.com/David-Goru/BachelorsFinalDegreeProject.