

**Development of a 2d video game focused
in puzzle resolution switching between
two complementary characters**

Initial memory

by

María Marrero Gómez

Tutor: Antonio Morales Escrig

INDEX

1. INTRODUCTION	Page 3
1.1 TFG summary	Page 3
1.2 Key words	Page 3
1.3 Work motivation	Page 3
1.4 Related subjects	Page 3
1.5 Objectives	Page 3
1.6 Awaited results	Page 4
1.7 Programs	Page 4
2.GDD	Page 5
2.1 Story of the game	Page 5
2.2 Characters	Page 5
2.2.1 Meredy, the priestess	Page 5
2.2.2 Celnyra, the thief girl	Page 5
2.2.3 Enemies	Page 5
2.3 Narrative design	Page 5
2.4 Game design	Page 5
2.5 Gameplay	Page 6
2.6 Interface	Page 6
2.7 Controls	Page 7
3.PLANNING	Page 8
3.1 Planning table	Page 8
3.2 Resources evaluation	Page 8
4.REQUIREMENTS ANALYSIS	Page 10
4.1 Functional requirements	Page 10
4.2 Non functional requirements	Page 12
5.SYSTEM DESIGN	Page 13
5.1 Class diagrams	Page 13
6. WORK DEVELOPMENT	Page 16
6.1 The title	Page 16

6.2 Art -----	Page 16
6.2.1 Meredy, the priestess, design and animations -----	Page 16
6.2.2 Celnyra, the thief girl design and animations -----	Page 17
6.2.3 Zombies -----	Page 19
6.2.4 Soldiers -----	Page 19
6.2.5 The boss -----	Page 20
6.2.6 Dungeon floors -----	Page 21
6.2.7 Interactables -----	Page 24
6.2.8 Interface -----	Page 25
6.2.9 Controls menu -----	Page 26
6.3 Sound-----	Page 29
6.3.1 Music -----	Page 29
6.3.2 Sound effects -----	Page 29
6.4 Programming-----	Page 29
6.4.1 The two protagonists -----	Page 29
6.4.2 Enemies -----	Page 33
6.4.3 Dungeon -----	Page 34
6.5 Testing -----	Page 35
7.RESULTS -----	Page 37
8.CONCLUSIONS AND FUTURE WORK -----	Page 38
8.1 Conclusions -----	Page 38
8.2 Future work -----	Page 38
9.BIBLIOGRAPHY -----	Page 39

1.INTRODUCTION

1.1 TFG summary

This work consists of creating a 2D art game centered in the mechanic of switching between two complementary characters with different abilities. The game is divided in different dungeons, which are composed by connected rooms. These rooms will contain puzzles, or enemies, or simply require one specific ability of one of the two characters. The integrated puzzles will be solved in a way that both protagonists will be essential propitiating the use of the switching mechanic. The player objective will be to resolve each dungeon puzzle while exploring in search of either an exit or the boss, depending on the dungeon. In the last case the player should defeat the boss

1.2 Key words

2D game, Adventures, Dungeons, Puzzles.

1.3 Work motivation

The main motivation for this work is making a videogame where the two main protagonists act complementary in order to progress further in the levels. The puzzles that the protagonists will come across are designed with the idea of making the player switch between the two characters. The final objective is giving the impression that the two protagonists work together like a team.

1.4 Related subjects

Design 2D, Artificial Intelligence , Game motors, Programming 1 and Programming 2.

1.5 Objectives

The main objectives are:

- Completely develop the two characters with all their animations.
- At least complete one floor of the dungeon with all the puzzles intended.
- Develop two types of enemies : zombies and soldiers
- Make a boss that requires both characters.

The most important part of the game are the two protagonist since the mechanics are focused in switching them depending in the abilities needed. Taking this into account the first objective is developing all their animations. After that, the dungeon and it's puzzles is the second most important feature to develop. Initially , the objective was only one floor, since it was not clear how much time it was going to take to develop. When the first floor was finished the objective changed to doing two more. The two kinds of enemies are necessary. Each enemy can only be defeated when one specific character attacks it, provoking the switching between characters desired.

Lastly, the boss is the final confrontation of the dungeon. It can not be defeated with one blow and only one of the characters can hurt it, but the player will need to switch characters in order to dodge the boss laserbeams. In the last part of the development new objectives have been added to the list:

- Including the menus: Main menu, controls menu and pause menu.
- Including audio and music.
- Including dialogue between characters at specific points.

This objectives are only necessary for adorning the experience, that is why they were not added until after the first objectives were reached.

1.6 Expected results

It is planned to have one dungeon at least with integrated puzzles, this dungeon is intended to act as a tutorial. Moreover it is expected to have the two main protagonists finished in addition of the boss and the two main types of enemies, including all their animations.

1.7 Programs

- Unity, for programming and assembling all the game assets and scripts.
- Photoshop CS6, for drawing all the animation frames and assets of the game.
- Paint Tool Sai, for drawing the introduction images.
- Vector Magic, for transforming bitmap images to vectorial images.

2.GDD

2.1 Story of the game

The story events happen in a fantasy age that combines medieval and actual elements. Prior to the game the two protagonist accompany the legendary hero and his soldiers to fight the chaos lord. But in the fight everyone is defeated. Celnyra, the thief girl, is trapped and locked in a dungeon, the hero disappears and Meredy, the priestess, falls from a cliff.

Fortunately, under the cliff is a river that saves Meredy from dying.

At the beginning of the game Meredy awakes in the same dungeon where Celnyra is trapped. Meredy will have to rescue Celnyra and fight against the lord of chaos once more but this time the legendary hero will not be there to help them.

2.2 Characters

2.2.1 Meredy, the priestess

Very innocent and friendly. She always tries to help others. Her light magic apart from healing can make plants grow and destroy zombies. Between the two protagonists is the only one that can swim. However she can not fight any other enemy besides zombies.

2.2.2 Celnyra, the thief girl

Celnyra is selfish and usually seeks her own good. She only decided to fight the evil because there was a high reward. Opposite to Meredy she is dexterous and agile in combat with the disadvantage that she can not swim. Stealth is her speciality.

2.2.3 Enemies

There are two types of enemies. The zombie enemies can only be defeated by healing magic while the soldiers can only be defeated by weapon attacks.

Dungeon bosses are special enemies that although they can be of either of the two types are not as easy to defeat as basic enemies.

2.3 Narrative design

As explained before the plot of the game is that the two protagonists even though they have the weakest roles of a classic RPG game have to confront the chaos lord due to the disappearance of the hero that was supposed to save the world. The events previous to the game are explained with tree introductory images with text. When the game starts the current character selected is the priestess that after a brief dialog starts searching other survivors. After a while she encounters the thief girl and frees her. From this point onward the interactions of the characters will be show as dialogs and the teamwork will be reflected in the switching between characters mechanic.

2.4 Game design

The world of the game is divided into several dungeons. These dungeons are formed by interconnected rooms which contain integrated puzzles that will be solved using the skills of the two main characters.

Puzzles are not explicit puzzles, but obstacles that stand between the player and his goal. They must make sense inside the dungeons, that's why they are integrated puzzles.

These puzzles can be, for example, to reach places to activate mechanisms, find entry to a secret room, dodge enemies to catch an object, search underwater to get a key, etc.

Some rooms puzzles are combats with enemies. There are different types of enemies, some can only be overcome by the healer and others can only be overcome by the thief.

The dungeon bosses have to be defeated between the two characters. In this way beating these enemies can be considered another type of puzzle.

The challenge that is proposed is that when the player finds a puzzle raises their strategy based on the skills of the protagonists, breaking the puzzle into smaller pieces that can be solved using one of them.

For example, the player must reach a lever but can not get to it because there is a door that blocks the way. Behind the door there are some guards that can not be defeated because they have too much life or armor. The player does not have the key to the door, but next to the character there is a stretch of water where objects can be found.

The strategy in this case would be that the healer, who is the only one who knows how to swim and dive, will look for the key in the water and then open the door. Once in the room of the lever the thief who is the only one who can use stealth, evades the unbeatable guards and activates the lever.

2.5 Gameplay

Although the player carries the two characters on his team, only the one assigned to him at that moment is shown on the screen. The current character can be changed by pressing the character change key at any time, except in areas where only one of the two characters can stay for reasons of story.

When advancing through the floors and trying to solve the puzzles, it must be taken into account that there are objects that only a character can interact with and enemies that only a character can defeat. Also in a matter of mobility, the two characters can not do the same.

For example, there are areas of water in which the thief can not be used because, as already mentioned in his character record, she can not swim. In these cases the player will have to change to the priestess. In the stealth zones, the priestess can not pass as a thief in stealth and the player will be forced to change the thief.

In the bosses the same thing happens, the two main characters work as a team to defeat them.

The skills of the healer are: being able to grow plants, swim, heal and have their attacks harm the zombie monsters. When the player switches to the healer the energy bar is mana and each time he cures he spends mana. Once spent, you can not recover until you find a source of magic.

The thief on the other hand can defeat enemies that are not zombies, do a flip to dodge quickly and go through the stealth zones.

When the player switches to the thief, the energy bar changes to stamina, and every time he dodges she spends stamina. When using stealth the stamina bar will be progressively emptied and when leaving the stealth mode it will slowly refill. Stamina recovers over time

2.6 Interface

The main interface of the game, the one the player will be seeing the most of the time, displays only two things: the character information and the number and type of keys the player has.

The character information contains the group life and the energy of the current character along with her image and name. The life is shared between the two protagonists so for the two of them the life bar is the same. The energy depends of the current character, it can be mana or stamina and it is not shared. The number of keys and the type of keys displayed is only so the player can access it without having to open a menu.

The main menu is simple, it has three options the player can access through clicking the respective button: start the game, see the controls and exit the game.

The controls menu is divided in three sub menus, one menu for each character controls and other menu for general information. The decision of dividing it in three is for the player commodity since is easier to find the information wanted this way.

The pause menu grants access to the controls menu or the main menu and has also the option of exiting the game.

2.7 Controls

On the keyboard:

KEY W - Walk up.

KEY S - Walk down.

KEY A - Walk to the left.

KEY D - Walk to the right.

KEY O - Character skill.

KEY I - Character skill.

KEY K - Character skill.

KEY L - Character action when the character is near to an object.

KEY P - Change characters.

The movement of the character is carried out with the W,A,S,D keys.

Keys O,I and K activate the skills of each character and key L serves as an action button.

Key K while in the water serves to dive, once diving it serves to return to the surface.

The action button has different functions depending on the situation. In the case of being near an object that can be caught the action is to take the object and in the case of being near someone the action is to speak with that person.

In case of adjusting the controls to a command the character would move with the left joystick and the two keys would be replaced by two of the control buttons.

3. PLANNING

3.1 Planning table

Art	Main protagonists design, drawing and animation. Boss and enemies design, drawing and animation. Dungeon rooms drawing and creation. Interactables drawing and creation. Menus drawing and creation.	80 hours
Programming	Character movement programming. Enemies programming and AI. Interactables programming. Event programming. Menu programming.	70 hours
Final Memory	Final memory redaction.	35 hours
TFG presentation	TFG presentation creation.	15 hours
Test and depuration	Testing for possible bugs.	30 hours
Design	Puzzles and dungeon progression design. Boss design. Resource distribution design.	50 hours

3.2 Resources evaluation

The video game, for now, is only the first dungeon and the first boss, also the art and the levels are composed by multiple PNG images which, opposite to the 3D models with high quality textures, do not require height requirements to work and do not take much space in memory.

Having this into account any current average computer can play the game. The game supports a broad range of screen resolutions since unity offers this feature and I purposely did not restricted it. Moreover, the screen information is prepared to adjust to screen size.

Like mentioned before the videogame takes very little space in memory which makes it more accessible for any computer.

The game is designed to work with keyboard and the options in the menus are selected with mouse. It can be adapted to a controller in the future but for now is incompatible.

The human team is composed by only one person. The music is from youtube and it does not had copyright. The materials utilized for the game are laptop, with Unity for personal use to do the programming. Photoshop CS6 with a wacom tablet for the art. The sounds are recorded with a mobile phone and are edited with Audacity.

4. REQUIREMENTS ANALYSIS

4.1 Functional requirements

Requirement 1: Player – Main menu

In the main menu , by clicking in the new game button the player should enter the game, the controls button should redirect him or her to the control menu and the exit button should quit the game.

Requirement 2: Player – Move the character

The keyboard keys A,W,S,D should activate the characters respective animations and move them in the desired direction :

Key W should make the character move up.

Key S should make the character move down.

Key D should make the character move right.

Key A should make the character move left.

Requirement 3: Player – Change character

Key P should change the current character and the HUD images and energy bar.

Requirement 4: Meredy – Heal

Key I should replenish the life bar. This should cost all mana of the mana bar.

Requirement 5: Meredy – Grow Plants

Key L near a plant should make the plant grow. This should cost 1 mana unit of the mana bar.

Requirement 6: Meredy – Attack

Key K should activate the attack animation and the projectile .

Requirement 7: Meredy – Swim and dive

When Meredy enters in contact with water she should swim, if the key K is pulsed she should dive and if key K is pulsed again she should stop diving and go back to swimming state.

Requirement 8: Celnyra – Dodge

Key I should activate the dodge animation. This should cost stamina from the stamina bar.

Requirement 9: Celnyra – Attack

Key K should activate the normal attack animation if Celnyra is not in stealth mode and the stealth attack animation if she is.

Requirement 10: Celnyra – Stealth mode

Key O should activate the stealth mode animations. While activated the stamina bar will progressively decrease.

Requirement 11: Mana – Replenishing

Mana only replenishes when entering in contact with a mana fountain.

Requirement 12: Stamina– Replenishing

Stamina replenishes while time passes.

Requirement 13: enemies– Alert state

enemies should follow the current character when they see it.

Requirement 14: Zombie– Damage

The zombies only can take damage when in contact with the priestess projectile.

Requirement 15: Soldier– Damage

The soldiers only can take damage when in contact with the thief girl attack.

Requirement 16: Interactables – Activate

The interactables should activate when the character is near them and key L is pulsed.

Requirement 17: Life bar – Decreasing.

The life bar should decrease 1 unit when the characters take damage.

Requirement 18: Player – Control menu

The control menu should show all the information necessary for the player to understand the game controls.

Requirement 19: Player – Pause menu

The pause menu should let the player go back to main menu scene, see the controls menu or quit the game.

Requirement 20: Game over menu– Game over

The game over menu should let the player go back to main menu scene.

Requeriment 21: Victory menu– victory

The victory menu should let the player go back to main menu scene.

4.2 Non functional requirements

Requeriment 1 – Animations made by frames

The animations should be composed by frames so the animation can be done in unity.

Requeriment 2 – Dialogs

The dialogs should display the two characters on screen and the text should contribute to contextualize the game.

Requeriment 3 – Sound effects

The sound effects should be in .wav format since other formats like .mp3 can not be lopped.

Requeriment 4 – Menus

The menus should be functional with the mouse.

Requeriment 5 – Dungeon floors

Each floor of the dungeon has to had a different color so it does not feel monotonous.

5. SYSTEM DESIGN

5.1 Class diagrams

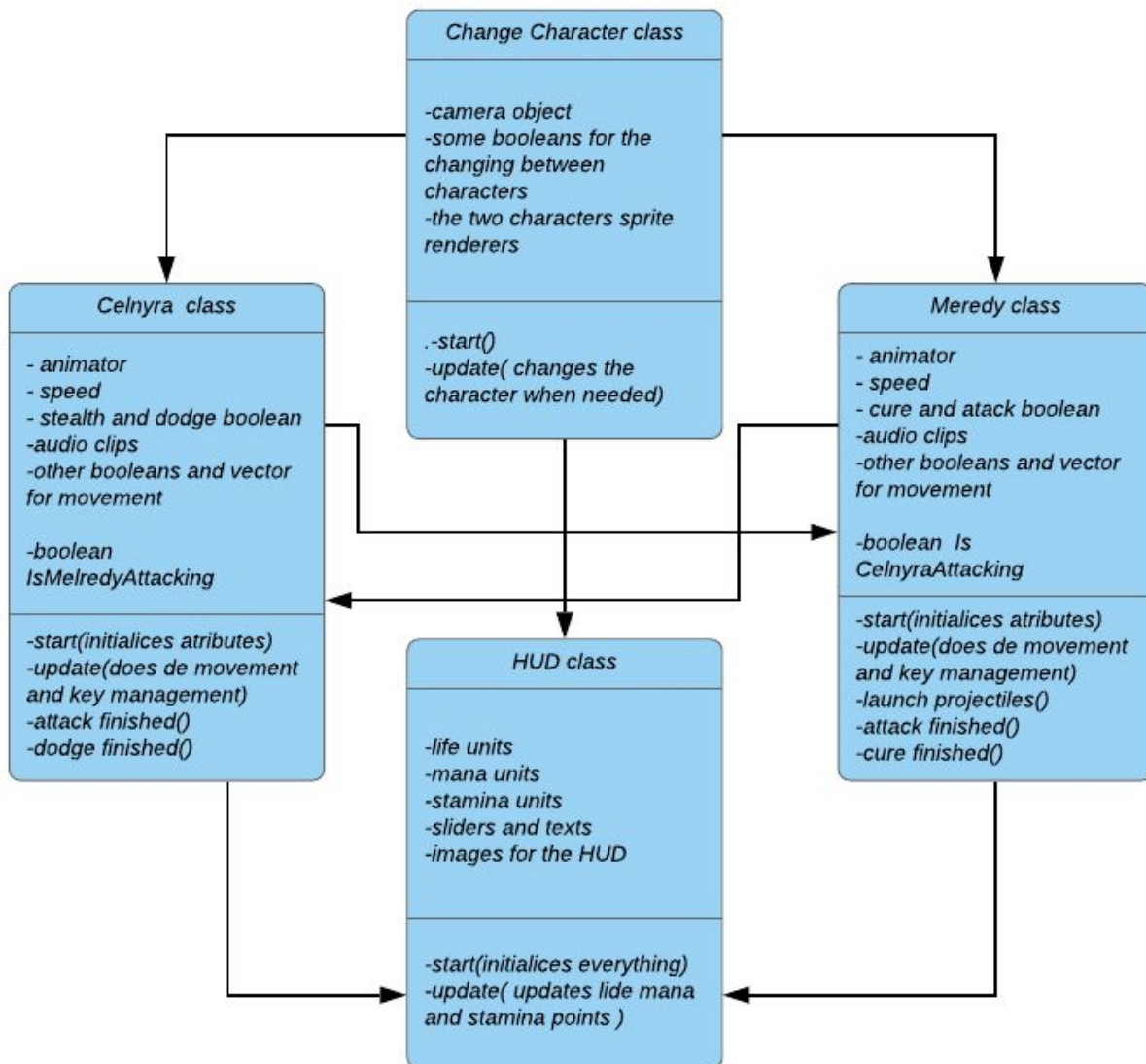


Diagram 1: Character classes, hud class and character change class relation.

Diagram 1 represents the connections between the character classes, the HUD class and the Change Character class. The names of the classes and the attributes are only for understanding better the classes, they are not the real names. The two characters had a boolean which is public and the other character class can change it if needed. This allows the two classes to communicate with the other. The HUD class only has the value of the characters stats and updates the life bars when the value changes. The values are public and all classes can change it. The Change Character class decides which character is visible and attaches the camera to it. The HUD character class can read the current character from the character change class and update the information showed. In the two character classes when one skill costs mana or stamina it is communicated to the HUD class that updates the energy units and changes the value of the slider.

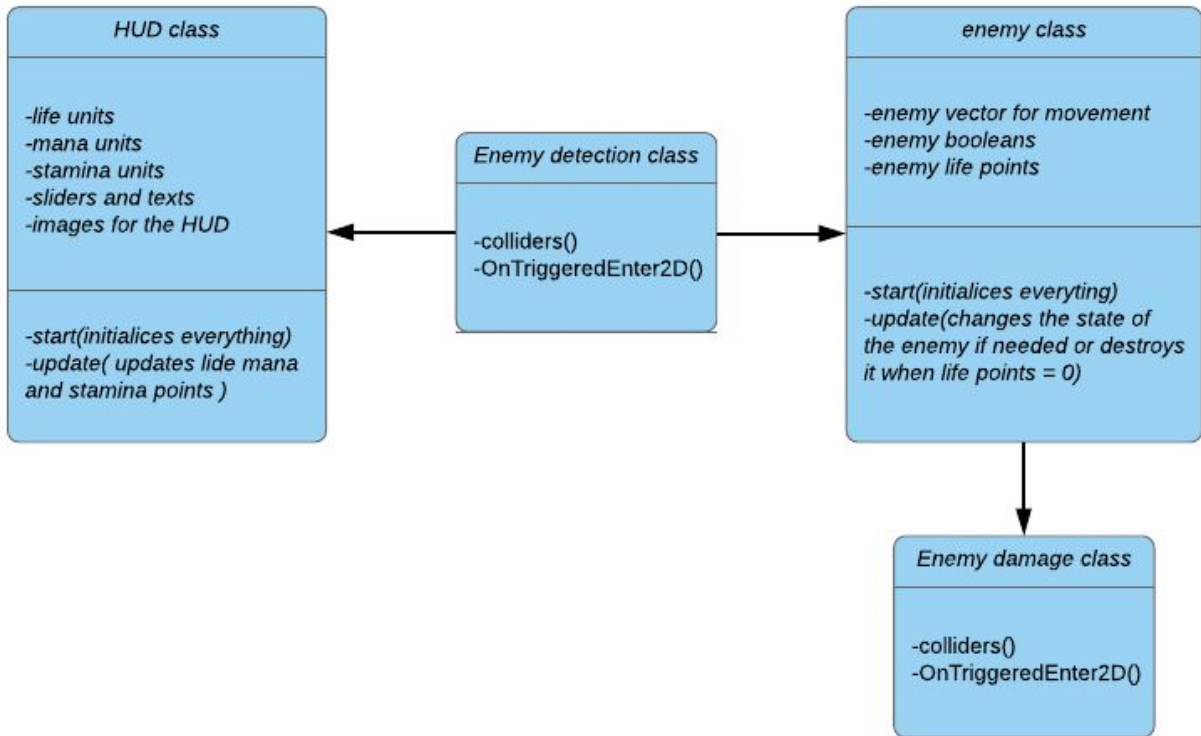


Diagram 2: Enemy classes and its relation with the HUD class.

The enemies core class, enemy class in Diagram 2, depends on two other different classes. One that detects if is touching the character and other that detects if it is in the near range. The first one communicates the HUD class to decrease the life points, and the second one changes the state of the enemy in the enemies core class. The enemy core class manages the state of the enemy and its life points and deletes it if needed.

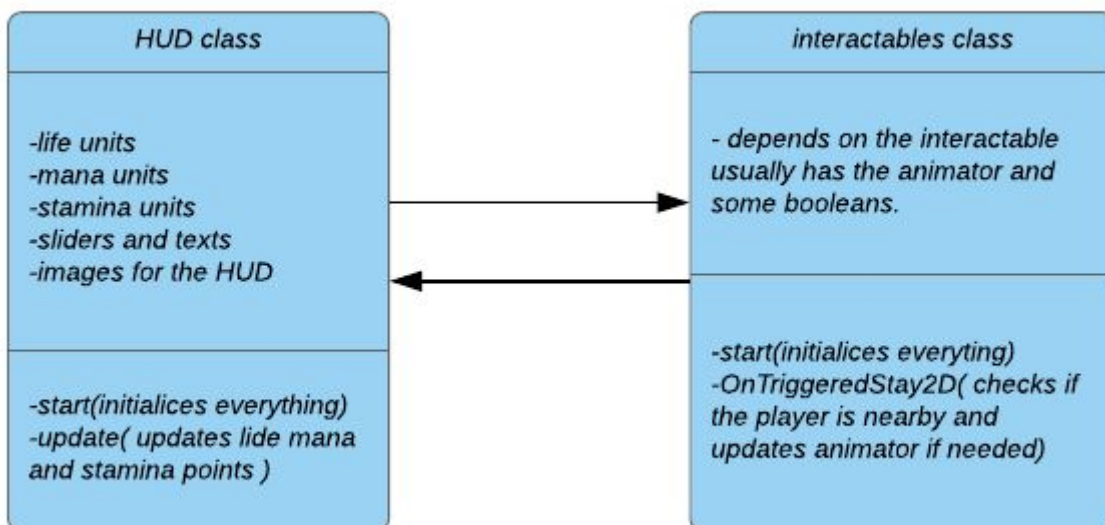


Diagram 3: Relation between the HUD class and the interactables classes

The interactable object class only interacts with the HUD and some concrete animators. When it detects the character nearby and the key L is pressed communicates the HUD class if it needs to actualice a value or not.

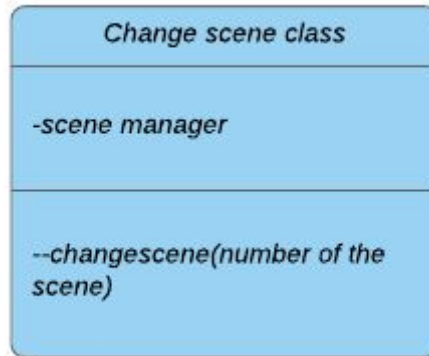


Diagram 4: The change scene class.

Lastly the change scene class is a very small class that is referred when there is a transition between scenes. It only has one function, the ChangeScene function.

6.WORK DEVELOPMENT

6.1 The title

One important thing to take into account when choosing the title is that the words “dungeon” , “quest”, “adventure”, etc, are not suitable since they are overused in the videogame world. Moreover the title should talk about the two protagonists and their adventure. The first option for the game name was “Estamos perdidas” which has two meanings in spanish, “perdidas” as lost because they do not know where they are, and “perdidas” as they do not have a chance of winning since they are the weakest members of a RPG team. However in english one of the two meanings is lost, being that the word “perdidas” traduces to lost only. So the title located in english will take only the second meaning explained before and become “We are doomed”.

6.2 Art

6.2.1 - Meredy, the priestess, design and animations



Meredy design is based in the classic RPG priestess design. The tunic which is a classic piece of clothing for the healer is intentionally white and blue. This is because white and blue are colors often associated with purity. Her hair is yellow and short in order to transmit her positivity and cheerfulness and she has blue large eyes for the purpose of conveying her innocence. The healing staff at her back channels her magic into spells.

1. Meredy final design.

Meredy animations include the skills of healing, growing plants and launching a projectile, as well as walking , swimming and diving in the four directions.



2. One frame of each of the Meredy animations.



3. The two left attack frames with the two projectile frames

6.2.2 - Celnyra, the thief girl design and animations.



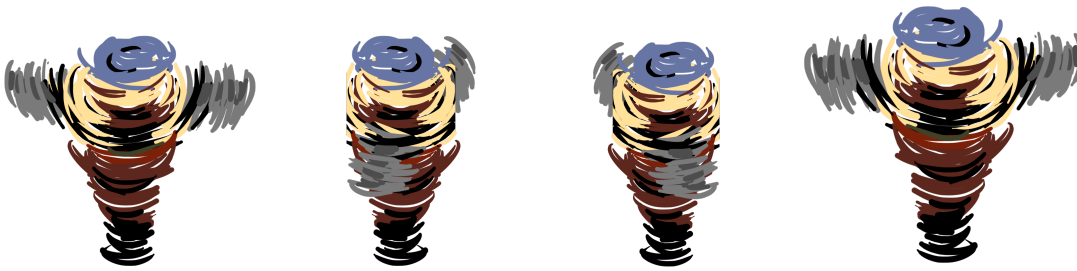
Celnyra has a modern design than Meredy. She wears dark clothes that favor stealth and her jacket and trousers are not based on classic games but on a more current clothing. Unlike her clothes Celnyra hair is light blue-grey. The initial design of Celnyra had completely grey hair, however the gray color make her character look older and colorless which was not what was planned.

1. Celnyra final design.

Celnyra animations include the rotatory attack in normal and stealth mode, as well as animations for dodging , walking and walking in stealth mode.

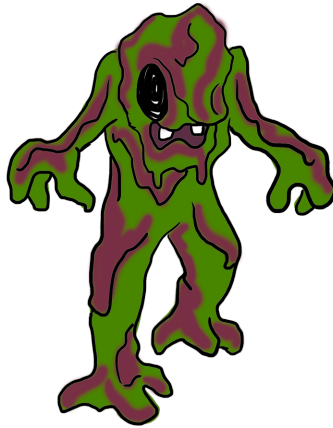


2. One frame of each of the Celnyra animations.



3. Rotatory attack and up and down dodge

6.2.3 - Zombies



Zombies can only be defeated by light magic. They are created from decomposed bodies so their appearance is incredibly distasteful. Green and purple are their principal colors and their texture resembles the worst type of mud. They were revived by the chaos lord and only obey him.



1.Zombie movement frames

6.2.4 - Soldiers



The soldiers wear the chaos lord army red uniform. This enemies are human even if they are in the

evil army, therefore they can be defeated by normal attacks while light magic only heals them.



1.Soldier movement frames

6.2.5 - The boss



The boss of the first dungeon is of the undead type of enemy but in contrast with the lower rank zombies he has a more intimidating aspect and is much more powerful.



1. Boss normal floating animation.



2. Boss almost dead floating animation



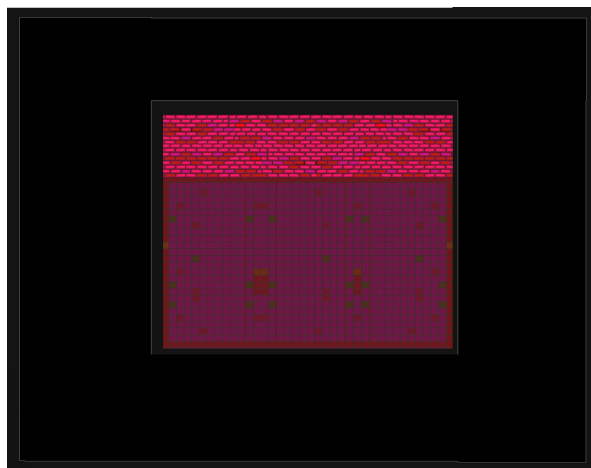
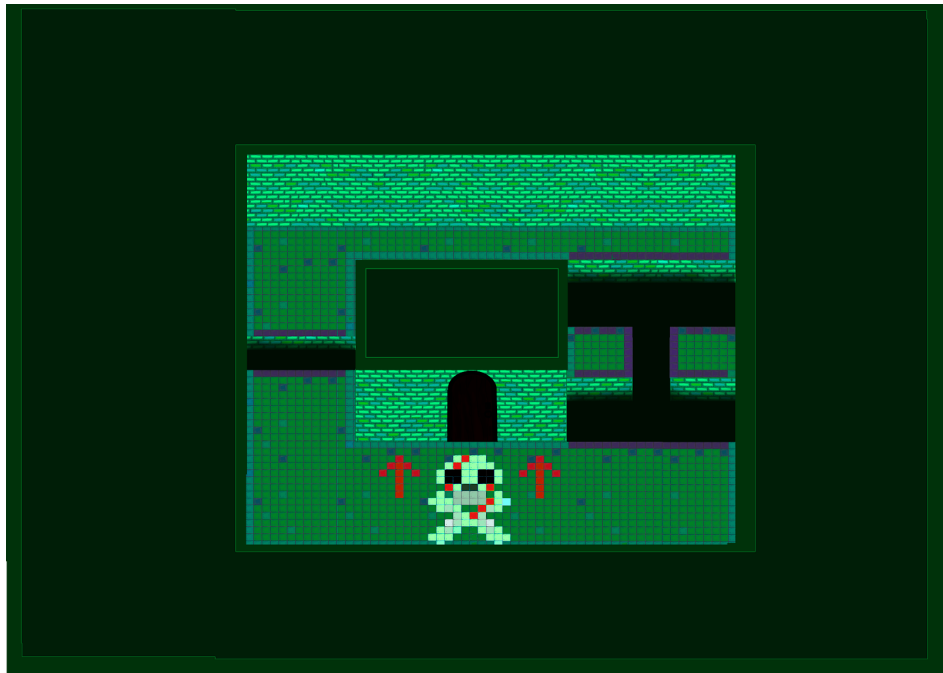
3. Boss laser beams animations

6.2.6 - Dungeon floors

There are four floors in the dungeon. Each floor has different colors to prevent them from being monotonous. The current art for the floors is a rework of the original since the previous art had visual issues and was not attractive to the eye. The art of the levels is divided in different layers: walls, ground and ceiling.



1. First and second floor art without de interactables and the water zones.



2.Third floor and boss chamber art.

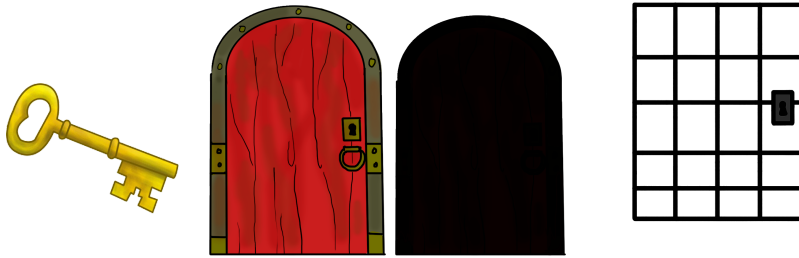
The water zones are in different layers than the rest of the level since is easier to adjust to the room.



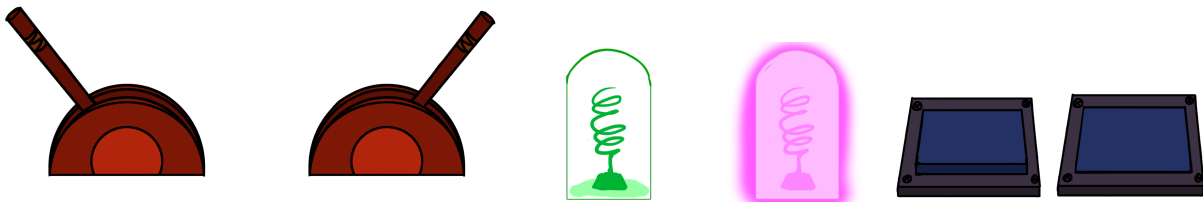
1.Examples of the water zone layers

6.2.7 - Interactables

The interactables are fundamental for the game puzzles. There are various types of interactables depending on the function they have.



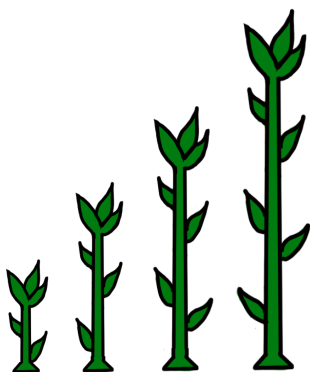
1. Keys and doors interactables.



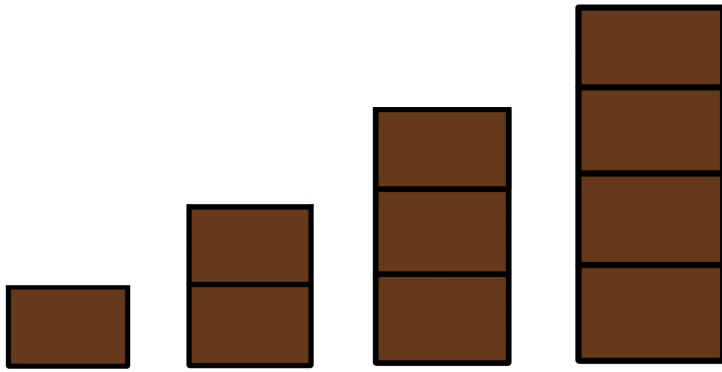
2. Activators interactables.



3. Laser beams interactables.



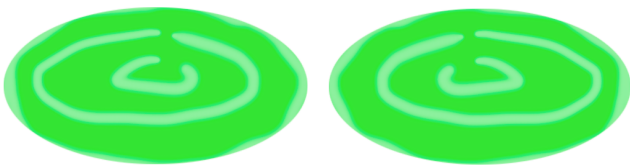
4. Plants interactables.



5. Bridges interactables.



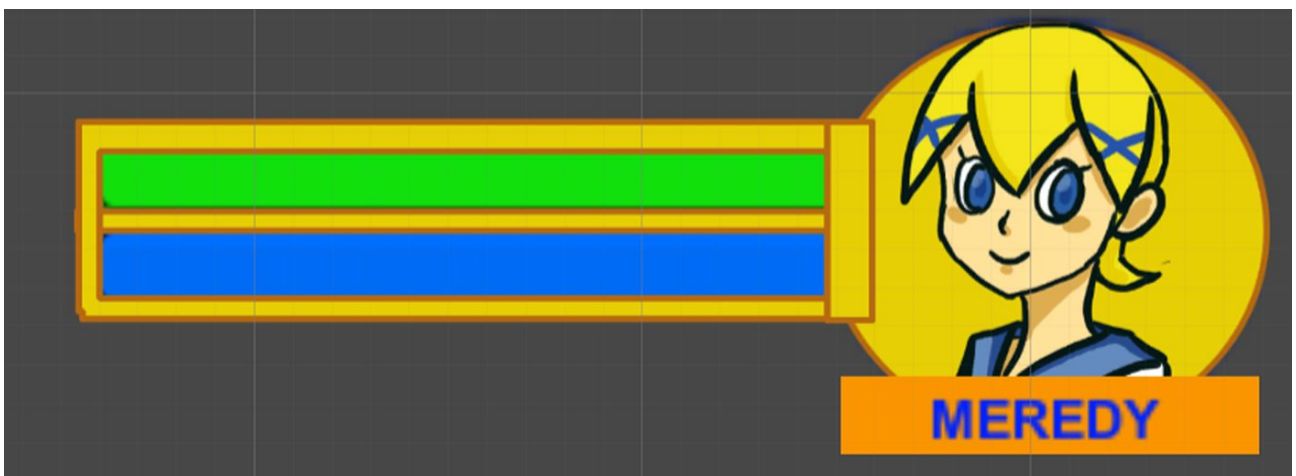
6. Hidden hints interactables.

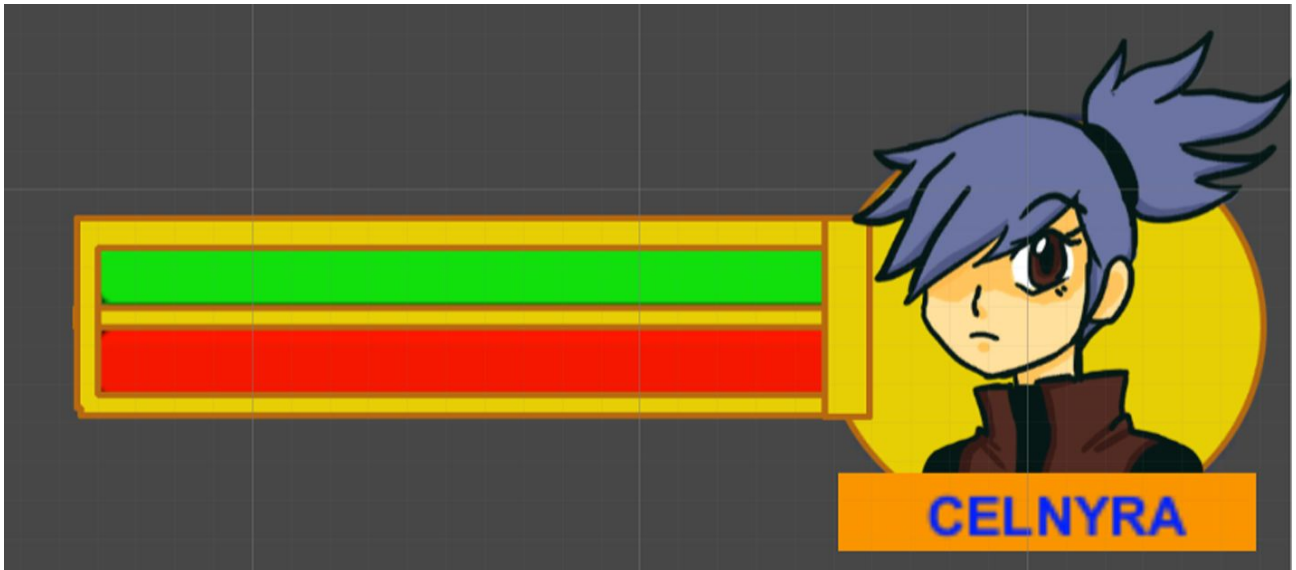


7. Magic fountains interactables.

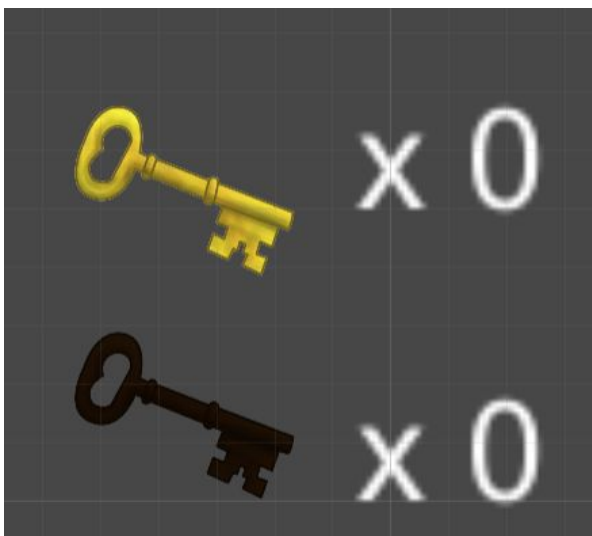
6.2.8 - Interface

The interface contains the life and energy bar as well as the number and type of keys owned.





1.HUD bars and images depending on the current character.



2.Display of the type and number of keys owned

6.2.9 - Controls menu

Due to the game having many mechanics the controls menu has all the information that the player needs in case of not remembering how the game works. There are five pages of controls for each character and three more pages for other information of interest.

CELNYRA CONTROLS



- Key W - Up
- Key A - Left
- Key S - Down
- Key D - Right
- Key L - Take objects / activate objects
- Key I - Dodge
- Key K - Attack
- Key O - Stealth mode On/ Off
- * CELNYRA CAN'T SWIM!

STEALTH MODE



- You enter stealth mode when pressing the O key.
- While in stealth mode enemies can't see you.
- The stealth mode decreases the stamina progressively.
- When you run out of stamina you will exit the stealth mode.
- You can attack while in stealth mode.

DODGING

While dodging you are invincible.



- Dodging takes a little portion of the stamina bar.
- You can't dodge without stamina.

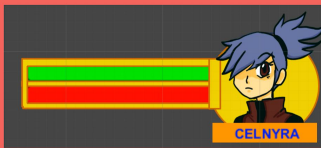
ATTACK



- You can't damage zombies with Celnyra's attacks.
- But soldiers will die if you strike them.

LIFE AND STAMINA

The green bar is the life that the two protagonists share.



- The red bar is only for Celnyra and represents her stamina.
- Stamina refill with the pass of time

1.The five pages of Celnyra controls.

MEREDY CONTROLS



- Key W - Up
- Key A - Left
- Key S - Down
- Key D - Right
- Key L - Take objects /Activate objects
- Key I - Heal
- Key K - Launch projectile, make plants grow or dive while you are in the water.

SWIMMING AND DIVING




- Touch the water to Swim automatically.
- Key K while in the water- Dive or emerge
- Sometimes while diving you can encounter hidden hints.

PLANTS, HEALING AND MANA



Make plants grow takes little mana.

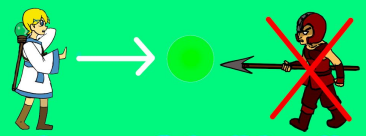


Healing takes all of your mana.

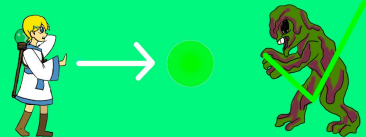
The magic fountains refill your mana.

LIGHT PROJECTILE

The light projectile doesn't damages life beings



However they are very effective with the zombie enemies.



LIFE AND MANA

The green bar is the life that the two protagonists share.

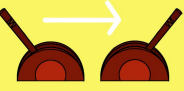


The blue bar is only for Meredy and represents her mana.

You can only refill the mana bar in the magic fountains.


2.The five pages of Meredy controls.

INTERACT WITH OBJECTS



Press key L near an object to activate it.

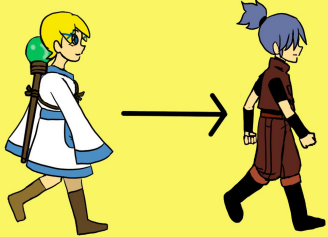
Sometimes you will not need to press key L.



Press L near a key to collect it.


CHANGE CHARACTER

Press key P to switch between characters.

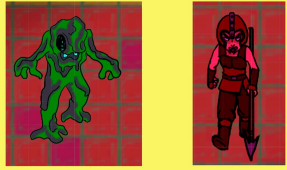


OTHERS

You had the number of key you have of each type in the left superior corner.



The biggest enemies with diferent color from the others are unbeatable you have to avoid them.



3.The three pages of other information.

6.3 Sound

6.3.1 - Music

The two musics chosen for the game are music without copyright from Youtube.

The first one is "Epic Motivational and Cinematic Inspirational Background Music" by AShamaluevMusic.

And the music for the boss fight is "Dragon War" by Makai Symphony.

6.3.2 - Sound effects

The sound effects included in the game are made with the mobile recorder and the audacity free program. The process for the sound effects is simple. Once recorded with the help of the audacity program the audio tracks are edited. Some of the effects are made by duplicating audio tracks and others are made by changing the audio track pitch.

6.4 Programming

6.4.1 - The two protagonists

Each character class manages the animator and the key input for that character and is connected to the other protagonist class with public boolean variables so they both can know the state of the other one. Therefore the character class can communicate with the other character class at the same time it can be independent. In order to have the two protagonists moving at the same time without colliding with each other the collisions in the player layer are disabled.

The key management is a very simple code. It only takes into account the booleans for the skills of both characters and changes the animations and the movement vector when the conditions are met.

The switching characters system in the one in charge of switching the two characters it is situated in a different object from the characters since it needs to be always enabled. It basically changes the current character public string so any class can access it.

```
if Celnyra has been rescued from the dungeon
{
    if Key P is pressed
    {
        if the current character is Celnyra
        {
            -Disable Cel sprite renderer and other
            attributes
            -Call CamaraToMeredy() function
            -Change current character to Meredy
        }
    }
}
```

```

else ( if current character is Meredy)
{
    if Meredy is not swimming or diving
    {
        -Disable Cel sprite renderer and other
        attributes
        - CamaraToCelnyra();
        -Change current character to Meredy
    }
}
}
}

```

1.Pseudocode for the switching function.

Moreover the camera follows the current character using the parent-child object. The camera is attached to the current character and switch the attachment to the other when needed.

```

Function CamaraToCelnyra()
{
    change camera parent to Celnyra.
}

```

```

Function void CamaraToMeredy()
{
    change camera parent to Meredy
}

```

2.Pseudocode for the CamaraToMeredy and CamaraToCelnyra functions.

Meredy projectiles has a little complex code in order to detect in which direction it has been launched and move towards that desired direction. Moreover it disappears after some distance and while the projectile is still on screen there can not be another projectile from the priestess. These measures are necessary to avoid having more than one projectile on screen and a possible bug of the attack animation.

```

if there is no projectile currently
{
    if Key k is pressed
    {
        if Meredy is not in the water and is not using the
        heal ability
        {
            Change the Meredy public attack boolean to
            true so the celnyra movement class knows
            Meredy is attacking
            Instantiate a new projectile at the projectile
            origin coordinates
        }
    }
}

```

```

        disparado = true;

        if (Meredy animator is in the up animation)
        {
            direction = "up";
        }

        else if (Meredy animator is in the left
animation)
        {
            direction = "left";
        }

        else if (Meredy animator is in the right
animation)
        {
            direction = "right";
        }

        else
        {
            direction = "down";
        }

    }

}
in the moment key k is not pressed
{
    disparado = false
}

```

3.Pseudocode of the first part of the launching projectiles function that decides when it is possible to launch the projectile.

```

if (disparado == true)
{
    if (direction == "up")
    {
        Sets de direction vector to Up
        Moves Towards the desired direction
    }

else if (direction == "left")
{
    Sets de direction vector to left
    Moves Towards the desired direction
}

```

```

else if (direction == "right")
    {
        Sets de direction vector to right
        Moves Towards the desired direction
    }
}

else
    {
        Sets de direction vector to down
        Moves Towards the desired direction
    }
}

```

4.Pseudocode of the second part of the launching projectiles function that moves the projectile to the direction desired.

As for the energy bar the cost of the skills is calculated in the character class but is updated in the HUD class.

When the priestess makes the plants grow the mana decreases. If the priestess heals the Opposite to the priestess, the thief girl stamina system regulates itself without help of any intractable like the magic fountains. The stealth mode and the dodge cost stamina but the stamina replenishes with the pass of time.

```

void Update()
{
    if Stealth mode is activated
    {
        decrease stamina with the pass of time
    }
    else {
        refill stamina with the pass of time
    }
    set life bar value to the life points value;

    if current character is Meredy
    {
        Change HUD images to Meredy images
        change energy bar color to blue.
        set energy bar value to the mana points value
    }
    else
    {
        Change HUD images to Celnyra images
        change energy bar color to red.
        set energy bar value to the stamina points value
    }
    update number of keys values
}

```

5.Pseudocode for the HUD update function.

6.4.2- Enemies

Zombies are only affected with the power of the priestess, which is a projectile of healing power. The projectiles have the "healingprojectil" tag so the zombie collider detects when is being attacked.

The soldier witch on the contrary is only affected with the thief girl attack which has the "normalAttack" tag.

The soldier and the zombie share a similar state machine with two states. The alert state in which the enemy is not moving and the pursuing state that is triggered when a character enters the range of detection. In the pursuing state the enemy follows the character until the character has exit the detection range or has died. In the case that the current character is the thief and she enters stealth mode the enemy stops pursuing her since she can not be seen.

```
Enemy movement update
{
  if pursuing state has been triggered
  {
    if Celnyra stealth mode is activated
    {
      change state to not pursuing state
    }
    else
    {
      update lastX
      update lastY
      Move Towards the player
    }
  }
  Update animator depending on lastX variable
  Update animator depending on lasty variable

  //This is for deciding if the animator will play the
  left or the right enemie animation
}
```

1.Pseudocode for the enemy state machine.

The boss is always in pursuing state and the contact with him decreases the life bar also it has four laserbeams near him that rotate and only can be dodge with the thief girl. The laserbeams change direction depending on the boss life left.

```
Boss movement update

Boss life bar = life points left;

if (lifePoints >= 7)
{
  LaserBeamsRotateRight();
}
```

```

if (lifePoints >= 5 && lifePoints < 7)
{
    LaserBeamsRotateLeft();
}
if (lifePoints > 0 && lifePoints < 5)
{
    change to the low life points animation
    LaserBeamsRotateLeftandRight();
}

MoveTowards the player

if (lifePoints == 0)
{
    activate the exit ladder
    destroy the boss object from te scene
}
}

```

2.Pseudocode for boss and the boss laser beams movement update

6.4.3 - Dungeon

The floors of the dungeon are first designed room by room and after that in unity the collisions are programed. The room images are organized connecting them with specific pieces.

The water parts that have to interact with the priestess have a special collider attached that triggers the swim animation. Also there are specific colliders at the water zones borders that prevents the thief girl from entering the water.

The interactable objects as keys, doors and levers. are programmed so that they do the function they were designed in the puzzles. In general the keys are programed to add one key in the hud information when they are collected. The doors only open when the player has an specific type of key. The levers and the buttons activate the bridges or other obstacles animators and disables their collisions. The laser beams can not be destroyed but they can be dodged.

```

void OnTriggerStay2D(Collider2D other)
{
    if (other.tag == "Player")
    {
        if the mana bar is not already refilled
        {
            update the mana points in the HUD class to + 1;
        }
    }
}

```

1.Example of the pseudocode for the interactable magic fountain

```

void OnTriggerStay2D(Collider2D other)
{
    if (other.tag == "Player")
    {
        if key L is pressed
        {
            if the key is not special key type
            {
                make the key sprite disappear
            }
            update the number of normal keys in the HUD
        }
        if the key is special key type
        {
            make the key sprite disappear
        }
        update the number of special keys in the HUD
    }
}

```

2.Example of the pseudocode for the keys interactables

```

void OnTriggerStay2D(Collider2D other)
{
    if (other.tag == "Player")
    {
        if (Input.GetKey(KeyCode.L))
        {
            Activate the animation for the activator
            Activate the animation for the activated interactable
            Disable the colliders for the activated interactable
        }
    }
}

```

3.Very simplified example of the pseudocode for the activator type interactables.

The examples above are a pseudocode version of the code for some interactables. The complete code for the activators is too extensive since it takes into account the type of activator and what it is that activates, however the core of the code can be seen in the simplified version .

6.5 Testing

The testing has been held regularly. Each time a new mechanic is included all the animations and mechanics included before are also tested in order to ensure the new scripts and assets don't interfere with each other.

Above everything else the parts of the project that need more testing are the complex animators of the two protagonists and the colliders of the different floors.

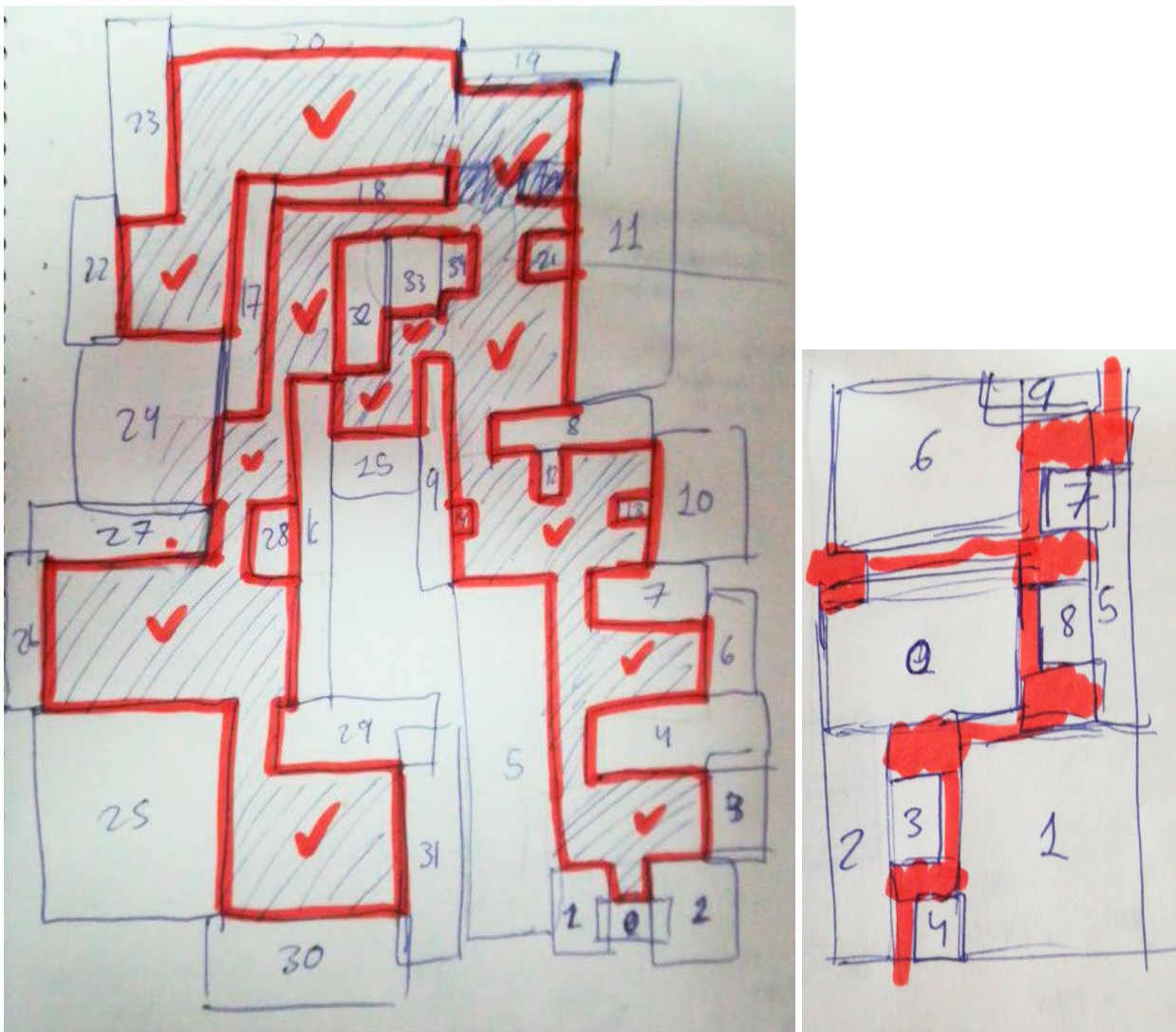
For the testing of the characters animations the best way of testing it is playing the level while

changing really often between animations. Forcing de animator controller this way is easy to find animation bugs or wrong transitions between animation states.

The colliders are tested one by one pushing the character against them. Is necessary that the colliders not only do not allow to pass but also that they do not generate visual issues, like for example when the character walks by a wall.

Since there are a great amount of colliders for scene when testing is necessary to make a scheme of the floor and the colliders and confirm which ones are corrected.

There are puzzle zones that have more intrincated colliders. This zones require a scheme apart from the general colliders one for better testing.



1.Example of the colliders general scheme of the second floor (First image) and the concrete puzzle collider scheme (second image).

The squares with numbers inside at the above images show all the hitboxes of the level numbered, while the red zones show which colliders have been tested and corrected.

7.RESULTS

As shown in the planning table at 2.1 section, at first it was only planned to reach the first objectives. However, the programming was finished earlier than expected which left enough time to include new features.

The new features are the introductory images that explain the story briefly, and the dialog between the two protagonists. Moreover, the music and the audios for the game have been introduced and the different menus are also finished.

There was enough time to develop this new assets due to the fact that the main classes and mechanics were clear since the beginning and the interaction structure, even though connects many classes at once, is based in a simple core code.

There were some complications in the process of reaching the first objectives that needed more time than expected. One of the challenges with the levels was organizing the different layers. In unity the priority of the layer determinantes what sprite will be seen on top of the others. The first design of the levels had many visual issues because the layer priority order.

Programming the projectiles was a very hard task since it had to detect in which of the four directions it had been launched and move towards that desired direction. Also it had to disappear after some distance and there can only be one projectile in the scene. In the end it was decided that the projectile decided its orientation depending on the priestess animation at the moment it is being launched.

The switching characters system was very conflictive since the two characters have to move at the same time and do not collide with each other. The unity feature of disabling the collisions between objects of the same layer was the best solution since trying to solve it by code didn't worked.

The boss required new art and a mechanic that forced the player to switch characters. It was decided that it will only receive damage from the priestess attacks but it will have four laserbeams near him that will rotate and only can be dodge with the thief girl.

In the art side the animations take more time than expected. This complications did not delay the project as it was started early to specifically avoid running out of time.

The new objectives added after finishing the first ones. The reworked assets were the main focus. Celnyra first animation attack and the dungeon floors first art have many visual incorrections consequently they had to be reworked.

Reworking the levels generated another problem, the colliders should be adjusted again to the new art. Therefore all the colliders were tested and adjusted several times.

In the end all the proposed objectives were met. The game is a complete dungeon with three floors and a boss. It also has plot exposition witch was not initially planned. The complete rework of the four levels looks attractive and all visual issues had been corrected. The enemies that were supposed to appear are all done and the fight with the boss is also finished.

The game main mechanic which is switching between characters works quite well. The different abilities of each characters are needed to different puzzles and there is a lot of variety of this puzzles. The dialog also helps to know better and sympathise with the two protagonist as they explore the dungeon.

8. CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

This work has helped me to put together all the things I learned until now in all the aspects that a videogame requires, like the art , the programming, and the level design. I could solve most of the problems I have had while programming tanks to the previous experiences I have had using unity. Also unity has a lot of tutorials and manuals that can be accessed at any moment which is incredibly helpful in this cases. I also discovered I have still to learn a lot about animation and putting sound in an unity project.

Doing a video game requires a lot of time and effort and doing all this effort in a team of only one person is really hard. Still , having the three floors of the dungeon and the boss completed along with the dialogs and menus it is a lot more of what I expected to achieve.

Even though there are things that have been left out I am satisfied with the ending result and I would like to include them in the future.

One of the things I never thought I will put much effort was the title, but in the end I was thinking on it until the end. I really liked both final titles, the english one and the spanish one.

8.2 Future work

Continuing this work in the future will consist in doing more dungeons while advancing the plot. The relationship between the characters should continue to grow and it should be shown in the dialogs. The number of enemies should increase as well as the number of bosses. New puzzles and bosses will be more difficult, and the protagonist can get power ups or new abilities to equilibrate this difficulty increase. I would also like to include the option of saving your game so the player can continue from the last saved game. It would be good to include new music and audio sounds too. There are some visual effects that should be added to make the game a little more consistent and I would like to spend more time with the characters animations. I will not include new playable characters since I want the adventure to focus in the two girl protagonists, and I think more characters would diverse the attention, but dialogs or plot events will probably present some secondary characters that will support , or not , the protagonists.

9.BIBLIOGRAPHY

Music:

- "Epic Motivational and Cinematic Inspirational Background Music" by AShamaluevMusic.
- "Dragon War" by Makai Symphony.

Programs:

- PhotoShop CS6- <https://www.adobe.com/products/photoshop.html>
- Vector Magic https- <https://es.vectormagic.com/>
- Paint Tool Sai https- [https://es.wikipedia.org/wiki/SAI_\(software\)](https://es.wikipedia.org/wiki/SAI_(software))
- Unity- <https://unity.com/es>

Documentation:

- Unity documentation : <https://docs.unity3d.com/Manual/index.html>

Game:

- Link to mega:
<https://mega.nz/#!vRZjFC4B!3ASvAnGV0I3djMDVVWv8qsLH3S91GrvSvI1nA78Mwsc>