Second Report

Falling asleep studios

Johan Emmanuelli
Ilan Mayeux
Luca Sarubbi
Matthieu Porte (*Project leader*)

Contents

1 Project Advancement

1.1 Gameplay

1.1.1 What was done

As Ilan was already very at ease for developing the gameplay, he kept improving the main gameplay.
The player

During the last defense, we showed that the gameplay was advanced with multiple main features: Items, Weapons, Movement, etc. A few more items were added:

- **The Spear** - The spear is the third weapon of the game a most probably the most powerful one. This weapon has a great range of melee attack, a fast attack speed and, the ability to be thrown at enemies and recalled. This weapon will be the best with the fireball to kill dragons.
- **Potions** - Healing wounds is essential to survive, and therefore, potions can now be picked up and used. Different kinds of potions, with more or less a good amount of healing.
- **Chests** - Chests can now be opened, allowing one to obtain currency and potions or even better rewards. Chests can be opened by both players.

As mentioned above, a currency system was created. We do not know yet how much it will impact the game, but it could be a good way to have a score for the game or to buy items to merchants.

Quest System

The map is huge (1.5 per 1.5km), and new players may be easily lost on what to do, where to go, etc. We do not wish to implement a minimap as the map is still relatively small and straight forward. Hence, the quest system was implemented. There are for now three types of quests:

- Quest Area: This quest requires the player to reach the designated area. To be completed, it requires 1 or 2 players on the location where a light beam can be seen anywhere on the map.
- Quest Item: This quest requires the player to pick up the designated item. When the quest is active, the item glows up.
- Quest Monster: This quest requires the player to slay all the monsters. When this quest becomes active, it spawns the monsters in the area.

Each quest then point towards the next quest, allowing to easily build the main story.

Tutorial System

Games often offer tutorials, and this game is not excluded. The quest system structures the story of the game, but new players still ignore the principle of it. That is where the tutorial system plays. There are for now two different ways for it to be activated:

- By a quest: When a quest becomes active, it may or not has a tutorial attached to it. If it does, then the tutorial overlay is also activated, helping the player.
- By the Weapon Help key: When holding an item with the weapon wheel, press F1 will display information to the player about how to use it and everything else.

Now the player can slowly learn the game by the mix of the quests and tutorials. For example, when the game start, the first game is about the movements: Moving with WASD, etc, then how to pick up weapons, then how to use them and slay weak enemies, etc. Until the player reach its final destination.

Water System

Our game has a lot of water in it: Rivers, Lakes, Waterfalls, etc. Yet, until now, nothing was done on it. The player could basically jump and walk in the lakes, which is definitely not a good thing and greatly reduce immersion in the game. Therefore, we improved all that by adding buoyancy when jumping into water, and adding animations depending on the status: if you are in the water, you swim to stay at the surface, then if you start moving, you swim forward. The same way: animations of falling were added, so that the player cannot “run” while falling.

1.1.2 What needs to be done

The gameplay can be considered as done. Diversity remains to be added. What could remain is adding additional quests (story part), and more items (different type of potions, weapons, etc.). Chests also need to be added everywhere on the map (leveldesign). The Ghost jump input should become continuous (holding = flying longer) instead of more jumps.

1.2 AI

1.2.1 What was done

As planned in the book of Specifications, Ilan handled the A.I. part and will explain what was done and what needs to be done.

For this defense, the volumetric pathfinding was made. This was quite difficult as there was only a little information about it.

So, how did I implement the volumetric pathfinding? The first step was to split the space. Indeed, unlike the traditional pathfinding, we want to save the spaces where there are no place for the A.I. to move on. Because for a volumetric pathfinding, we want to allow the A.I. to freely move anywhere, except where there are obstacles. To do, so, we used the tree data structure, and more precisely, an Octree.

What is an octree? An octree is a tree that has at most 8 children.

So, the algorithm is working as follow: We add all the game-objects with colliders we want to take in account. Then, the root of the octree will make a cube that fits all these game-objects, as big as we need it to be. And now we have the space to divide, we divide the space into 8 subspaces as long as there is an object that is smaller than the cube. When this is done, any cube that is smaller than the object is deleted.

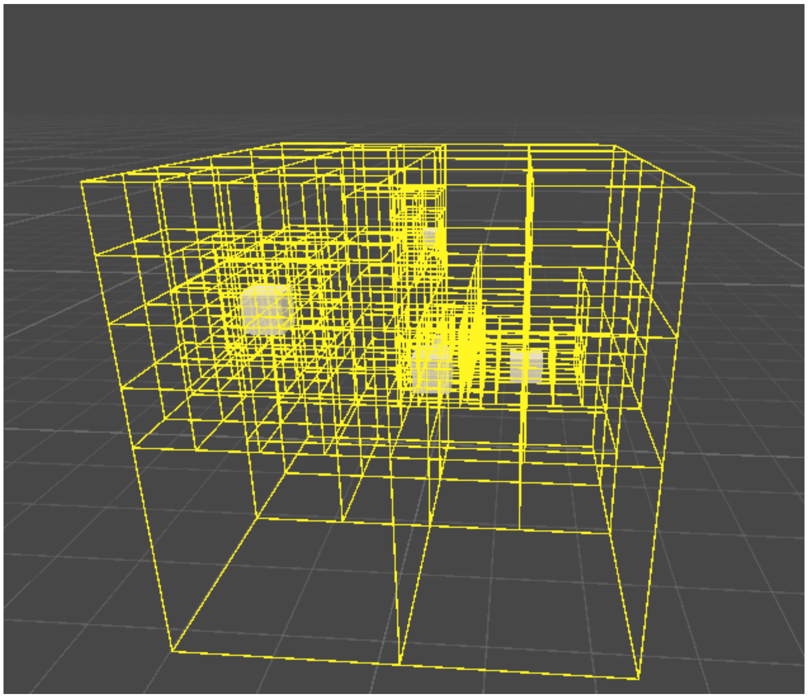


Figure 1: The space divided as an Octree

Now that we obtained our octree and divided the space, we need to build a graph from it. When the graph will be done, making a pathfinding from it is a piece of cake. Yet, building the graph was the hardest path and the slowest as well. The graph needs to be precise enough to have a short path towards the destination but not too precise or with many links otherwise the memory cost will be enormous. I tried many different way, but the one with the less links and still a good amount was linking all the adjacent cube without a game-object in it. By doing so, the links are still good enough to allow one to travel through the octree, and without too much links.

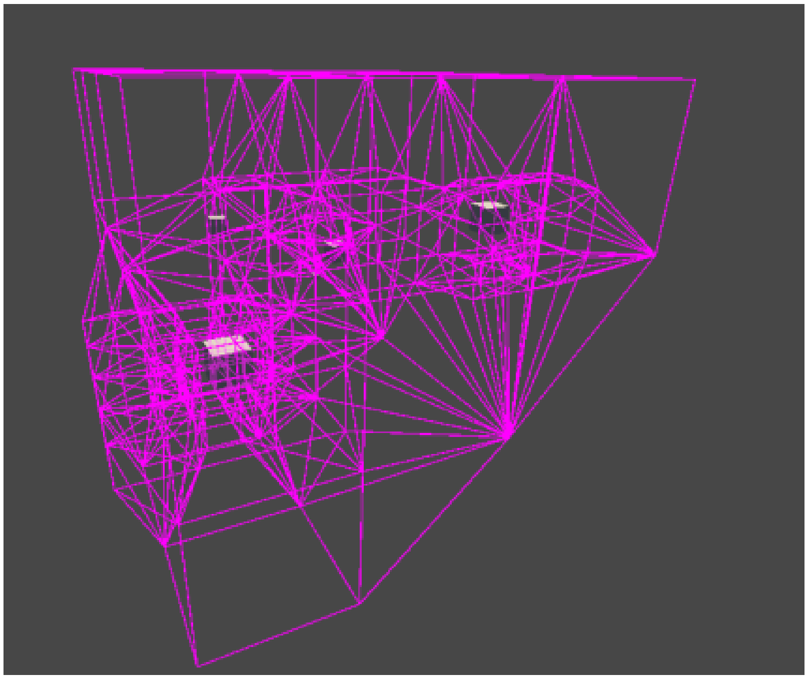


Figure 2: The graph made from the octree

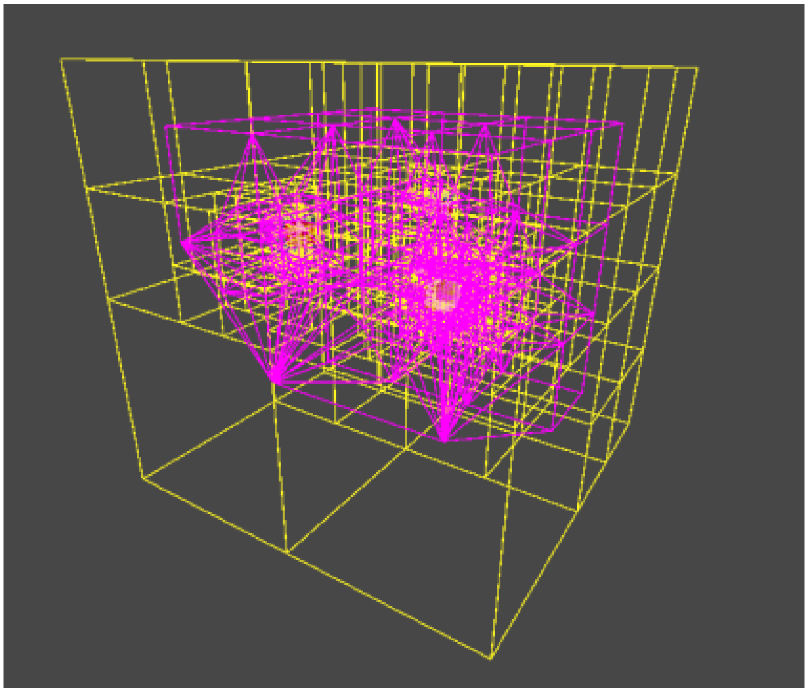


Figure 3: The graph and the octree together

Now that we obtained our graph, we just need to implement the pathfinding from it. As usual, the A* Algorithm was used. This is not the first time, nor the last (See GOAP) So the Monster class was inherited by the FlyingMonster, adding additional methods to control the monsters When the monster wants to move to a location, it first checks whether or not there is an obstacle between it and the goal. If not, it goes straight forward to it, otherwise, it uses the flying pathfinding to reach it.

While implementing this solution, lot of issues were encountered. For example, when I first tried the Graph, it cost me 26 Go of RAM. Luckily I still had some left to close Unity, but it was not very user-friendly. This was because too much edges were made between the nodes of the graphs. Then, another test took 10 min to generate the graph. This solution was light in term of memory, but consuming in term of time. Still a no, loading the map for 10 minutes is a no. The current solution (adjacent cubes) is the fastest and the lightest solution found until now.

Hence the Volumetric pathfinding was done.

This is it for the A.I. part. Some bugs were fixed (slime), ported to multiplayer, and some code optimised.

1.2.2 What needs to be done

The slime also needs a noise detection and a behaviour when it occurs, detecting the footsteps of the player.

Adding additional entities such as animals could add some variety to the game. Since it is happening in the wild, some dears, goats or even bunnies could emphasis the atmosphere.

Serializing the graph in a file could be a great one to avoid generating the octree in loading time. We tried making it work, but the main issue was that there were circular references (A graph is not one way) and so, there was no solution for now. One way could be to travel and serialize the graph by hand, but we still need to think whether or not there exists a better and more efficient idea.

1.3 Story

1.3.1 What was done

With the cinematic the story is really coming to life, creating quests and a tutorial at the beggining helps the player to feel integrated in the universe.

1.3.2 Side quest stories

Also, side quest stories have been conceived. The "the lost friend" quest has a plenty of dialogues with two NPCs and the journal of a dead person. In this quest, the two players will discover a little story by talking and interact with NPCs and object. A story in which a couple, Jack and Amanda, and their friend Mike decided to go move at the bottom of the mountain. As the quest goes, the two players discover the obsession of Mike for the mountain, that caused his death. The story stimulates the player to discover his environement.

1.3.3 What needs to be done

This is really the task we've waited to do, as most of the features are done now, we will be able to truly focus on it, and create even more characters and challenges and, the most important part: bring to life the story we imagined. Two friends, overcoming difficulties. United after death. Where their goals match: reaching the light at the end. The same light that will celebrate their passion, their friendship, their story. We want to tell the following message: friendship is the most powerful thing. *Enough to bring back to life your best friend.*

1.4 3d modeling

As we are not 3d designers, we had to look on the internet for some assets we could use. Yet, we had a quite precise idea of what we wanted in terms of environment. Matthieu did all the 3d modeling that is original to the project.

1.4.1 What was done

Firstly I learnt how to use the blender, then I focused on doing element of the environment. I created wood and stone bridges, fences, but most importantly I modeled modular houses so that we could get exactly what we need.

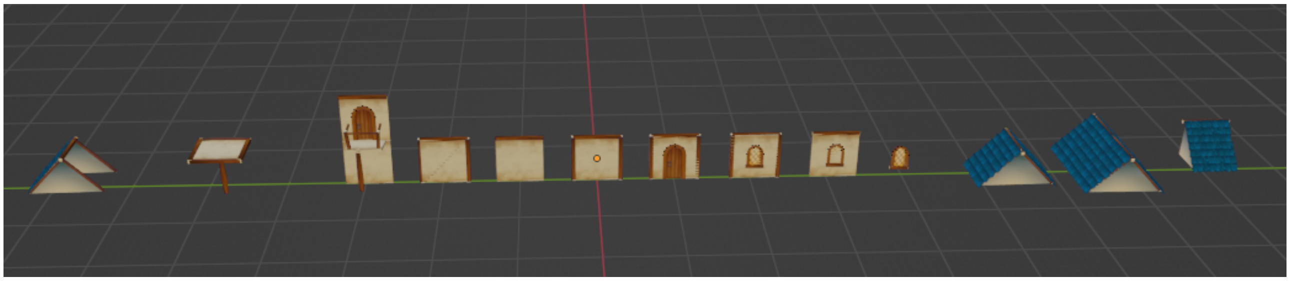


Figure 4: The pieces of the modular environment

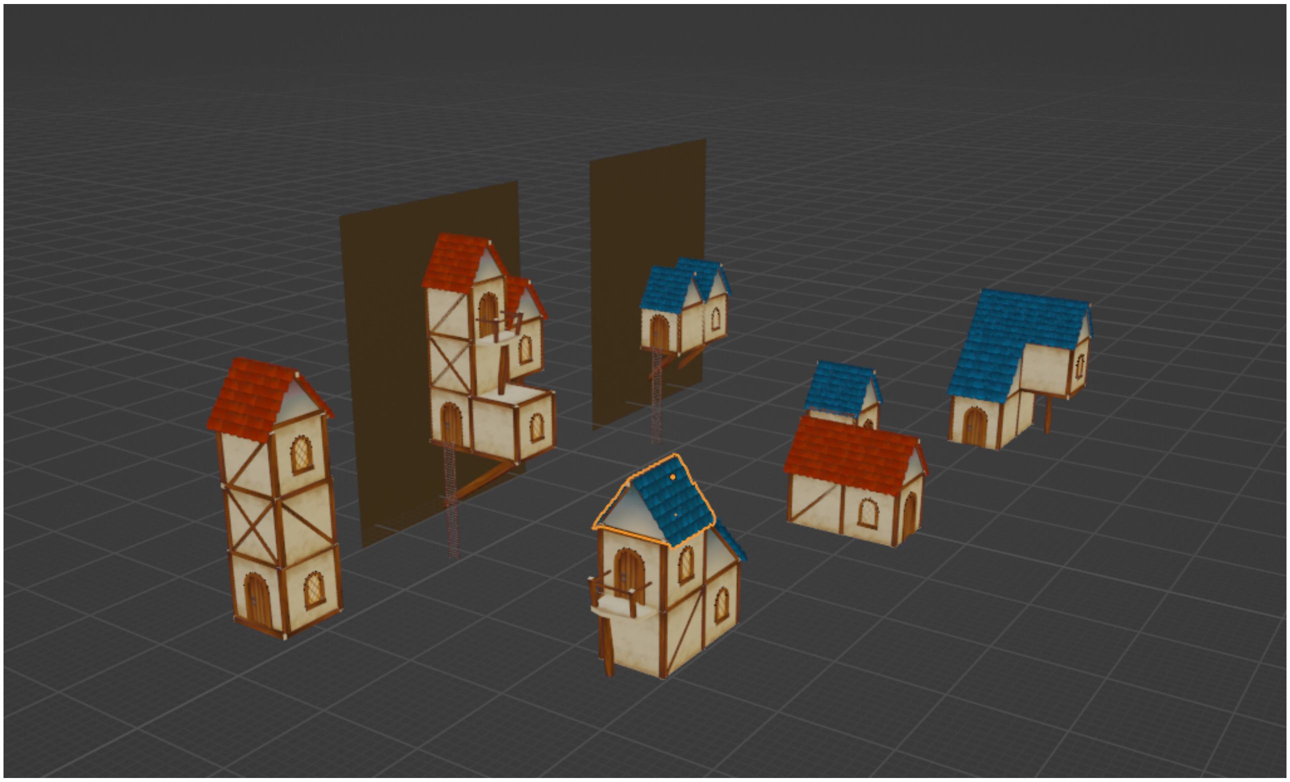


Figure 5: The pieces of the modular environment

Not everything can be shown here, neither everything will be used in the game. But through experimenting I have learnt more in depth how 3d models work.

1.5 Level Design

The map was already well advanced, yet, it was quite empty, Ilan worked to improved the general level handler while everyone added different scenes to populate the map. Johan made a quest called "The lost friend" And Matthieu took care of the dungeon.

1.5.1 What was done

Map changes

THE CHURCH IS NO LONGER AT THE VERY TOP OF THE MOUNTAIN This is probably the most important change as it was the contrary of what we said during the first defense. For the sake of the game, we had to make the church not at the very top.

Then, the whole structure of the main mountain was revisited, and it now definitely allows the player to climb to the very top without pressing the space bar.

We also added more trees and new varieties of them and grass.

Ephemeral Scenes (Ilan)

A huge map can easily grow heavy. Heavy means performance issues. Performance issues means non-user-friendly. And this is not acceptable as Falling Asleep Studio is the best friends of users.

Therefore, we created the ephemeral scenes, which could save a lot of resources. What is an ephemeral scene? As the name suggests, it is a scene that can easily be loaded or unloaded. When the player enters the area of the ephemeral scene, it loads additively the scene. When he exits, it unloads the scene.

As it can sound very simple, the performance gained are non negligible. And since the game is split in different scenes, it allows us to work on different scenes without impacting each others.

Therefore, the map was split into different scenes, while a Main Scene remains, the one handling everything and the spawning of these ephemeral scene. The MainScene is the one that has all the data that we still want to display: for example the silhouette of the villages, the rivers, etc.

Waterfalls (Ilan)

Rivers and lakes were set, yet there was no waterfalls as I was unable to do something clean. After many takes, and creating my own texture, I finally managed to make something that looks like what humans call a waterfall. As these waterfalls are made of particle system, they were put in the ephemeral scenes of their regions.

The lost friend(Johan)

This quest has been created with a village at the bottom of the moutain. This village is composed of two little houses and two NPCs. Also, we added a gate with a locked door and a padlock to unlock. To succeed in this quest, the ghost player has to find the hidden journal of a dead person to be able to find the code for the padlock. If the human tries to read the journal, he there will be a message saying "you try to read, but you can't.". This quest is not completely finished yet, and it will be solvable in two ways. The first one is to find the code in the old journal, and the second one is to solve a famous integral.

The dungeon(Matthieu)

I started making the dungeon, that will be the transition between the game on the ground and the summit of the mountain. There are a few rooms with different challenges, to bring diversity. There is still the laser puzzles but this time I've made bigger room alowing a better use of this feature. There's also a labyrinth which I've writen a bit about in the puzzles sections. Some rooms are made of parkour where the characters have to jump on platforms to reach a goal (It will be obviously easier for the ghost since he is able to fly a bit so he will eventually be able to help the other player). There is also fight rooms where monsters will have to be dealt with.

1.5.2 What needs to be done

The map was done. 1.5km is very large so it needs to be filled. Adding dungeons where the player can resolve the enigms and fight monsters, adding village to exchange ressources and communicate information. Adding some treasure chests.

The design of the church may also change. It is a bit too big and impressive.

When the church rings its bells, some events could occurs. There could be also a negative counter part where the world is plundge in the darkness with a dangerous threat ahead.

Some additional effects could be added to add more ambiance. For instance, we need to create our own waterfalls effects.

1.6 Puzzles

Matthieu took the responsability to plan and setup the basis we will need to create all the puzzles on the map.

Our game in its duality of worlds has two facets: the first is the combat system which forces players to fight certain monsters and to leave the others to their partner. The second facet is the reflection around puzzles with a particularity in each dimension. The two players do not usually have access to the same information, nor do they have the same freedom of action. When we design a puzzle, we follow a method: each problem must be new in some way for the player, to some extent. The player must have almost all the knowledge to solve it and each puzzle has a new aspect, which we will be able to use in future puzzles. Each level is therefore in its own way a small tutorial in disguise. The puzzles themselves are composed as follows: The player should not feel lost, he should even have an intuition of the solution. By following this intuition the player will naturally realise the real difficulty of the puzzle: the "catch". It can take the form of an inconsistency, an action still unknown to the player or a detail that the player may have missed (a missing object for example). A problem that is too small The moment the player starts to find the game simple, he realizes that that he had underestimated it. This is how we hope to keep the player interested in our project.
Here is the puzzles concepts that are working now :

1.6.1 Connect 4

In order to offer puzzles that are not as linear as they sometimes are, we decided to implement a variant of Power 4 in one of our levels. This is a cooperative version of the board game (both players can play together, which encourages communication and listening). The two players play against a computer that predicts all moves to a certain depth using the minimax algorithm. This is an algorithm that applies to game theory for two-player zero-sum (and full-information) games consisting of minimising the maximum loss. It causes the computer to review all the possibilities for a limited number of moves and to assign a value to them that takes into account the benefits to the player and to his opponent. The best choice is then the one that minimises the player's losses while assuming that the opponent seeks to maximise them. This algorithm is very memory and computationally intensive when one wants a consequent search depth and a very high level of play. This is why we have reduced this depth to a medium level, so as not to block the player or consume too many resources.

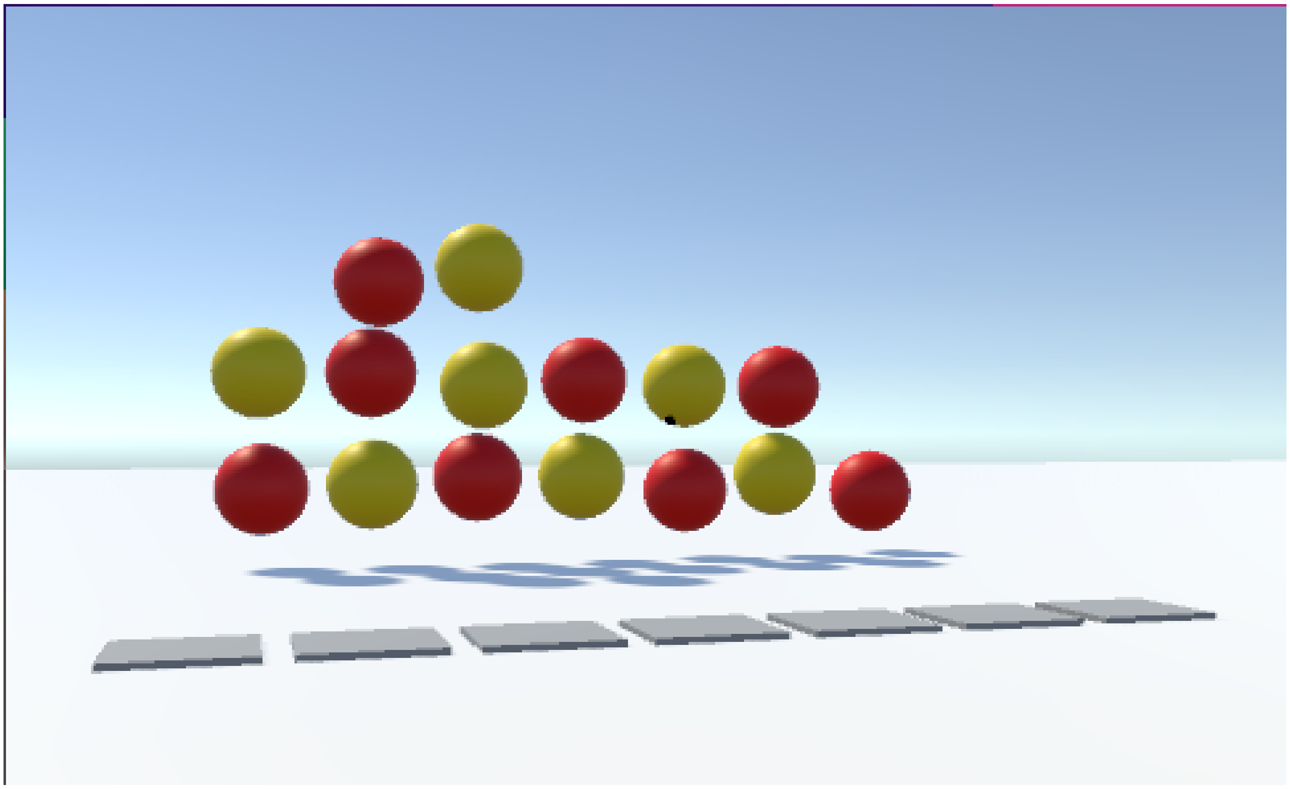


Figure 6: A game of connect 4

To come back to the idea of "catch" mentioned above, the players are naturally led to think that they need to win a game to keep going in the adventure, but in fact, it doesn't matter if you win a game or not. To keep going you need to jump on the stone of the game to reach a platform. We know that it can potentially be tough to understand for the player, so there will eventually be an npc here to give advice to the players. The players will have the possibility to come back and try to play again, this time with the possibility to earn some kind of reward.

1.6.2 Lasers reflexion

Some puzzles are meant to be more about reflexion. One idea we had it to introduce laser that can bounce on mirrors but not on anything else. The ray is comming from a turret and needs to get to an eye. If it does, the puzzle is solved. The turret throw a raycast, and the angle of reflexion is computed by unity every time it hits a gameObject tagged "mirror". A pickup script is also implemented to be able to move the mirror around and solve the puzzle. The ray also does damages so it can be used as a kind of weapon, but the player need to be careful because they will get damage from them eventually. This creates many opportunites of puzzles, with different difficulty throughout the game.

1.6.3 Labyrinth

An idea we've had was to let one player see a whole labyrinth while the other was trying to find the end of it. The dead player is on a platform that only him can see (and can interact with). From this plateform he can help the other player.

1.6.4 What needs to be done

With the level design this task remains uncomplete. We took a lot of time porting everything we did to multiplayer but now the code is working online so nothing stop us from building complex puzzles with all the functionalities we have so far.

1.7 HUD and UI

1.7.1 HUD

Syncing the HUD to the Game (Ilan)

Last time, the Menu done by Johan was on the multiplayer part of the game, and was not linked to the game. So after merging the two branches, I took the role to sync everything with the gameplay. So that, when the player adjust the sensibility, the inversion of the mouse, the sound, it modifies the settings.

For the sound, an Audio Handler was made. By default, the volume are at 100%, having the Audio Handler giving the Audio Source 100% of its original sound. When reducing it, it reduces the volume settings and all the audio handler adjust the sound to a certain percentage of the initial volume. Therefore, it allows having a variety of volume and not everything to 1 or 0.

Of course, these settings are saved and the player won't have to set them everytime the game is started.

1.7.2 UI

Improve menu and create the role selection menu (Johan)

Last time, the main menu was created with no prefabs, only game objects. So the slightest change on all buttons has to be done independently on all of them. Now, there exists a prefab for each UI element, with greatly simplify changes. Also, there was no skybox, so we added a night sky to create a better ambiance. To adapt with this ambiance, all the buttons had to be changed with a more discrete and calm background image. For the room menu, we changed the background of the canvas to a black image, again to adapt the new ambiance.

We also created a role selection menu that is a scene containing a canvas that is loaded when the player presses "Create room". The canvas displays the room name on top of it. It also displays the two players' pseudos, set as default by "Yaya". The menu allows the player that created the room to choose his role and his mate's role, whether Human or Ghost. It has be simply done with two buttons. The background image shows a moutain with a warmer ambiance than the main menu.

1.8 Website

The website is now mostly done, it is available at this adress :
https://matthieuporte.github.io/Luminosite-Eternelle-web/src/
The downloading links are not working yet as the game is not done but most of the content is available and the features are there.

1.8.1 Our vision

Our website is the main gateway to our project. It gives a concrete taste of the game. It must pique the user's curiosity, reflect the general atmosphere of the game, and make him want to download the project. For these reasons, we wanted to create an immersive website with a playful touch. The website must be accessible, we want to inform the user about our project. All the essential information will therefore be on the homepage, or it will be accessible with a click. For the immersive aspect we will take the artistic direction of the game, inspired by its large map, its duality between the world of the dead and the living, and its varied colours. We planned to use the library three.js, but we finally didn't take that option. Instead we recorded a cinematic that we are showing on the home page of the website. To satisfy the most curious internet users, we want to propose a page dedicated to the lore of our universe (its history, its characters), as the history and the characters are a very important element of our project. Making the website hasn't been a struggle since I (Matthieu) knew what I was doing since it is only a static page, with no database. The fact that I was the only one working on it made the process of building it quicker.

1.8.2 What was done

Matthieu made the website entierely, using tailwindcss and flowbite. We did not want to use any template to better reflect our ideas. As said before the intro video of the game is visible form the home page of the site, to get the user involved as soon as possible. The video is fixed so that the user can scroll over it, and some components are transparents as well so that you can see the video again as if you were seeing it through crakcs in the mountain. I used svg between parts to go with this idea.

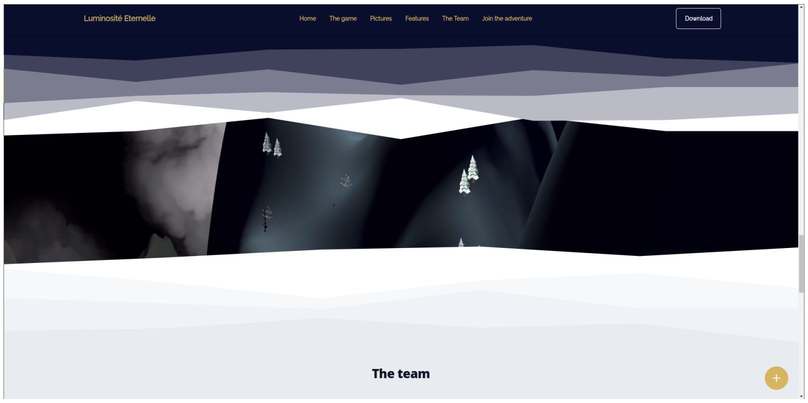


Figure 7: Cracks in the website using svg

1.9 Animations

The gameplay adaptation was handled by Ilan.

1.9.1 What was done

More animations were added and successfully synced on the multiplayer. This include adding swimming, falling, and other animations.

1.9.2 What needs to be done

We are soon enough reaching the quota of animations. Of course, there are still improvements to do. But animations by themselves are doing enough.

1.10 Multiplayer

1.10.1 What was done

This version of the project is mainly a great improvement of the multiplayer. Last defense, the project was split in 2: the main game, and the game with parts being synced to multiplayer and the menu. This was the main issue. This time, we worked hard to make everything in one project. The quest system was synced, mobs, weapons, etc. Moreover, when creating a lobby, you start either as a Human or a Ghost.

We also fixed the slime in multiplayer, where many bugs occured with it.

1.10.2 What needs to be done

Working on the multiplayer is a constant task, everytime we add some new functionality bugs might occur and need to be fixed. Luca is taking care of most of it. This is what we've been struggling the most with, since all the versions weren´t at the same stage at the same time.

1.11 Music and sound effects

1.11.1 What was done

Last defense, there was no sound at all. This time is different. We finally added S.O.U.N.D.S!

Music System (Ilan)

After selecting musics that fit the theme of each area, I made the music system. The principle is very similar to the Ephemeral Scene system. Different Areas with their musics. When entering the area, it takes the control of the music. Hence, the previous music fade out while the new one fade in. The way it is done is smooth and the user experience is not damaged from a brutal change

Added sounds (Ilan)

Many ambiance sounds were added: water, items, walking and so on. There are improvements to do, but in the general aspects, the result is way better, and the immersion improved.

1.11.2 What needs to be done

1.12 Cutscene

1.12.1 What was done

Part handled by Ilan, as the book of specification mentionned.

Cutscenes were an additional thing mentionned in the book of specifications. They were considered as not mandatory, depending whether or not we succeded reaching our goals. As we are almost done with our goals, cutscenes are getting added. For example, the end of game cutscene is almost done, roughly 2-3 minutes long and of course, better than any AAA game. The trailer cutscene is getting improved day after day.

A cutscene handler was made. When the player enters the cutscene, it deactivates the HUD, display some ”cutscene bar“ and change the input map. If the player skip the scene (with N for now) or that the cutscene ends, it goes back to the previous state.

1.12.2 What needs to be done

Short cutscenes to ”discover“ an Area, similar to those of the Zelda Games. Basically, when a region is first loaded, a cutscene that shows the region and its environment, and with it, puzzles, enemies, etc.

We also need to add the beginning cutscene to explain how one of the two friend dies and why their goal is the shining top. This is important to build the lore and the story of the game. It will basically show them together, as fellow alive humans, until one, accidentaly trip in the death mountains. However, the temple will ”save” him. And bring forward the message to go to the temple in the mountain, to follow the light. This is also explaining why they follow the beam of light (for tutorial reasons) and why their adventure starts here.

2 Personal experiences

2.1 Matthieu

I already had some small experience with unity/blender because I participated to game jams but this project has a size never seen before for me. It’s very interesting and challenging since for the first time I have to cooperate on a project. It makes it sometimes slower to progress but I am learning a lot about my friends, myself, and github.

2.2 Ilan

This project is almost over. Only one defense remains and we have managed to accomplish so many things together, as a team! Not only we created various features, but we reached our goals so far! The vision of this project, what was planned 5 months ago, we are almost there! And this is why, we will do our best to bring the best! The experience we accumlated from it will be reused efficiently, so that we can reach new horizons.

2.3 Johan

For a person that had no experience at all in making a video game, I am proud of what I’ve done. I have learned a lot on GitHub and how a collaboration project works. I also have gained knowledge and understanding of various concepts on Unity. I am really happy to do it with people that can help me to understand certain notions and I really feel pulled up. I am really excited to finish what I have to do the best way I can do it, and I am glad to see the amount of experience that it brings to us.

2.4 Luca

I like to discover new things. This project is an opportunity for me to broaden my knowledge of computers by working in a group and on an application (unity) that I am using for the first time. It is also nice to do the network part of the project because it helps me to understand how a video game works. We have high ambitions for our project, so I’m looking forward to continue implementing new features.

3 Task distribution throughout time

	1st	2nd	3rd	Current
Multiplayer	60%	90%	100%	100%
Gameplay	60%	80%	100%	80%
A.I.	40%	75%	100%	75%
3D (characters)	50%	80%	100%	100%
3D (environment)	50%	80%	100%	100%
Level design	50%	80%	100%	80%
Puzzle design	45%	65%	100%	70%
HUD and UI	50%	100%	100%	90%
Website	30%	70%	100%	75%
Animations	30%	60%	100%	100%
Music composition	50%	75%	100%	75%
Cinematics	0%	50%	100%	50%

We are on time on most of our tasks. The one where we are late are the less important or the one not mandatory for the first presentation. Some were interrupted because we made what we wanted to do by ourselves and will fill with external assets. Lots of work remains but we are in a great position since the harded was done beforehand. Sounds, Cinematics and others are here to complete the project we want to handeout.

4 Conclusion

Luminosité Éternelle is a game about friendship, about relying on our teammate to achieve our goal. The context and art direction is fantastical, so that anyone would be willing to play the game even though we think itś going to be hard for children.

The game mostly count on the gameplay mecanic such that one player is alive while the other is dead. Thus each will have access to different conversations, different clues and different interactions with the environement in general. Their goal is to find their way up the mountain where they hope to find help for resurrecting the dead player.

We make it a point to ensure that the experience given while playing the game is moving, therefore we will pay a particular attention to the details like the color palette of the game, the music and the cinematics.

This project has already reached a development state we did not expect. Lot of works remains, small bugs as well and many upgrades await. But overall, we are really pround of what we did as a team. We are ready to bring this project to life.

Sincerely,
Falling Asleep Studio's team.

5 Appendix

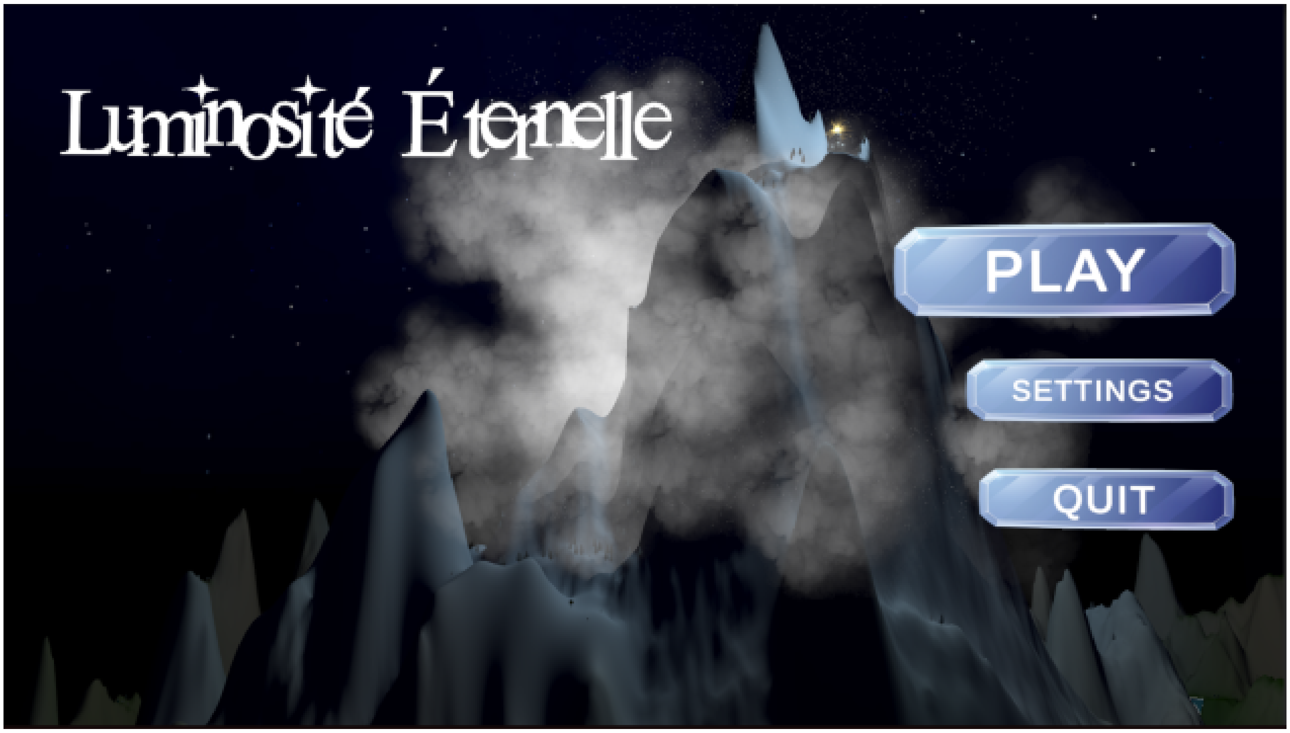


Figure 8: The game menu

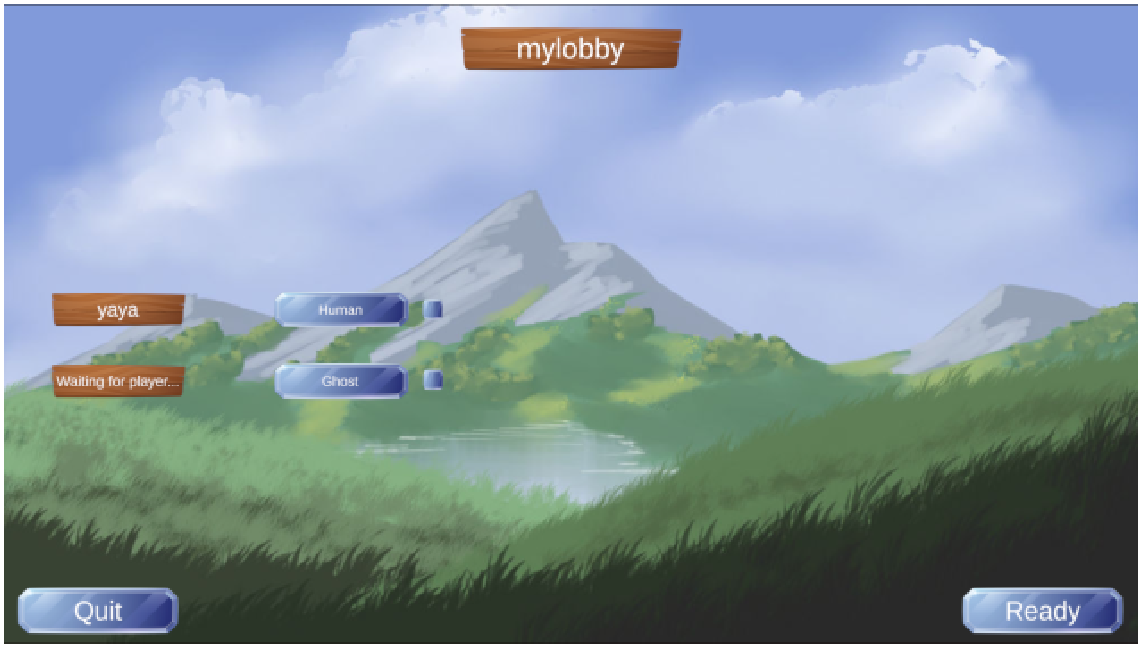
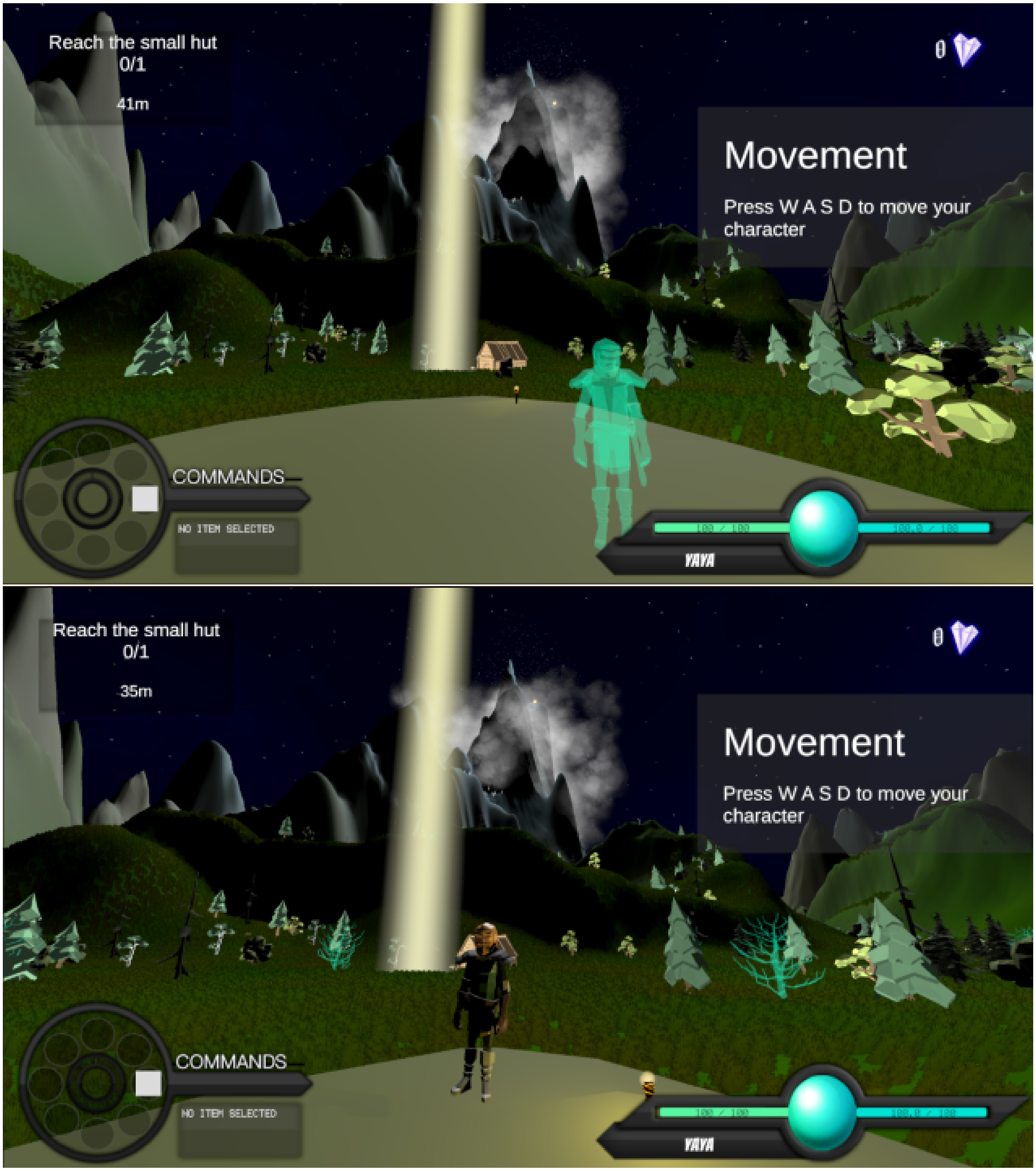


Figure 9: The role selection menu / lobby



Reach the small hut
0/1
41m

Movement

Press W A S D to move your character

Reach the small hut
0/1
35m

Movement

Press W A S D to move your character

Figure 10: The human view vs the ghost view, with behind the beam of light representing the next quest area

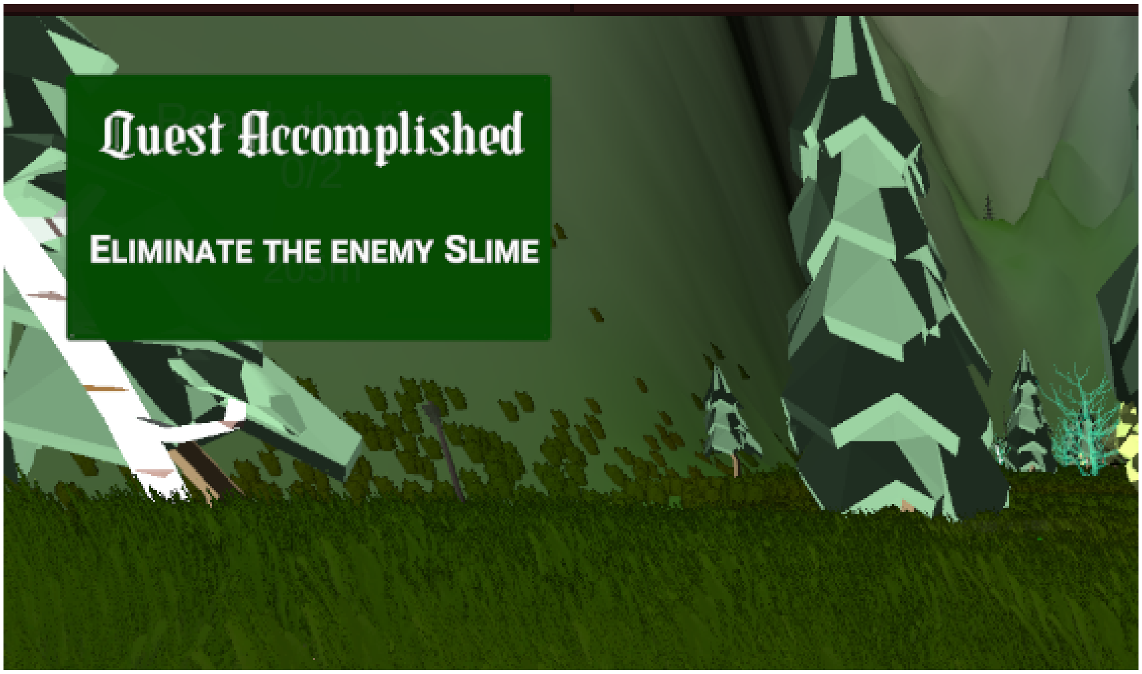


Figure 11: Quest accomplished!

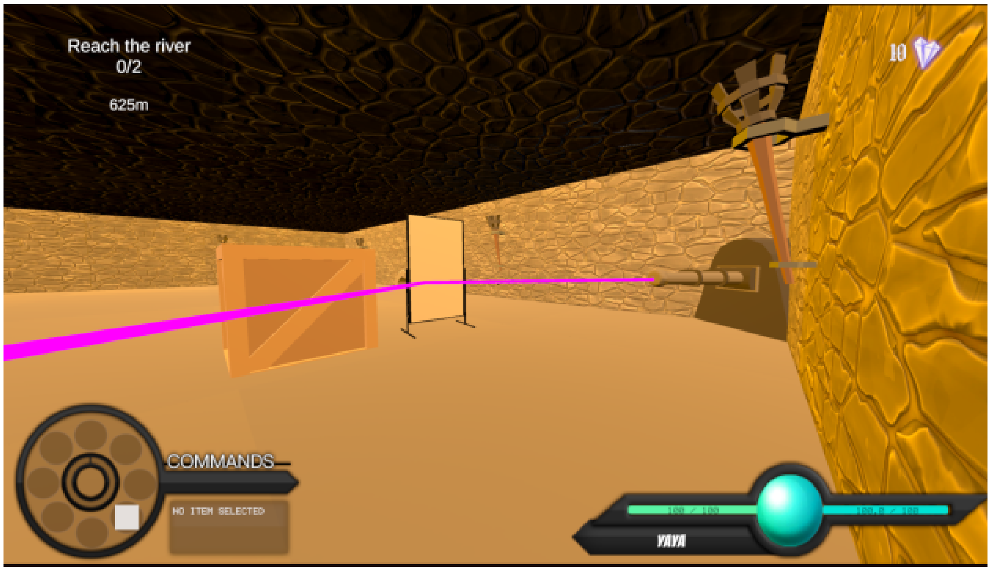


Figure 12: A piece of the dungeon with the laser puzzle

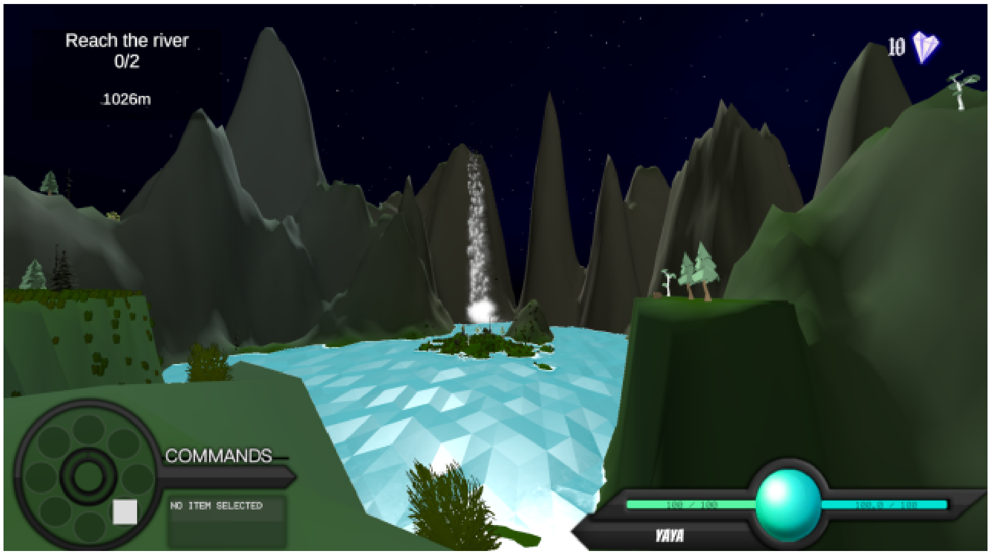


Figure 13: The big water fall of the upper lake area