

# Hyper Pursuit

## Defense Report 1

A BAAQ2 Project  
Baptiste Arnold, Abigaëlle Panhelleux,  
Quentin Rataud, Angela Saade

January 2022

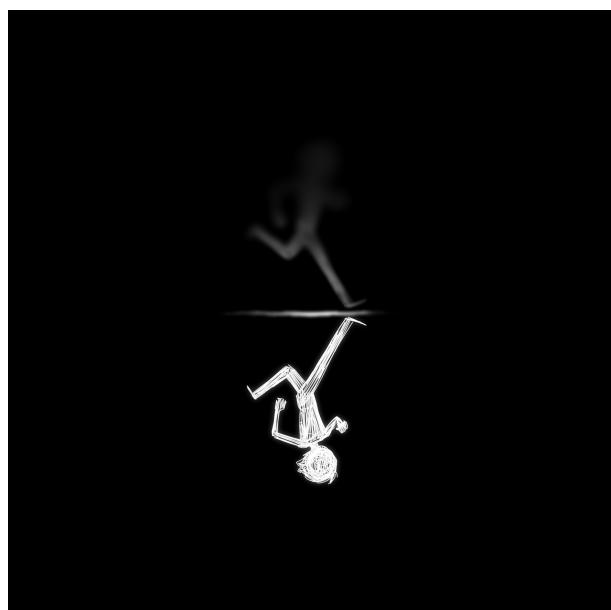


Fig. 1: The game logo

## Contents

<b>1 Specifications follow-up</b>	<b>4</b>
1.1 Introduction . . . . .	4
1.1.1 The team . . . . .	4
Members . . . . .	4
Name . . . . .	4
1.1.2 The game . . . . .	5
Concept . . . . .	5
Origins . . . . .	5
Game Backstory . . . . .	6
Logo . . . . .	7
1.2 Our Goal . . . . .	8
Our Ambition . . . . .	8
Benefit for the players . . . . .	8
Benefit for the group . . . . .	8
1.3 Technical Means . . . . .	9
Git . . . . .	9
Multiplayer . . . . .	9
AI . . . . .	9
1.4 Project Breakdown . . . . .	10
Planning . . . . .	10
Distribution of Roles . . . . .	10
<b>2 Achievements</b>	<b>12</b>
2.1 Abigaëlle . . . . .	12
2.1.1 The Backstory . . . . .	12
Introduction . . . . .	12
The Lore . . . . .	12
Some facts about the twins . . . . .	12
2.1.2 Level design and Minecraft . . . . .	13
Minecraft Server . . . . .	13
Starting cut scene . . . . .	13
Optical illusion . . . . .	14
2.1.3 Character design . . . . .	14
2.2 Baptiste . . . . .	15
Multiplayer-Photon . . . . .	15
Website . . . . .	17
2.3 Quentin . . . . .	18
2.3.1 Portals . . . . .	18
Render . . . . .	18
Teleportation . . . . .	18
2.3.2 Proof of Concepts . . . . .	19
2.4 Angela . . . . .	23
Test Room Concept . . . . .	23
Test of a Pair of Portals . . . . .	23

---

Test Room . . . . .	24
<b>3 Conclusion</b>	<b>25</b>
3.1 What was achieved . . . . .	25
3.2 What has to be done . . . . .	25



Fig. 2: The BAAQ2 team with the mascot Lola

## 1 Specifications follow-up

### 1.1 Introduction

#### 1.1.1 The team

**Members** The team consists of four S2 students, two girls and two boys, from the A3 class. Figure 2 shows the four members of the group. Table 1 shows the names of the four members of the group.

The group consists of Baptiste, Abigaëlle, Quentin and Angela. None of the four has ever completed a Unity project beforehand.

**Name** The name of the team is BAAQ2, a word play between the initials of each team member's name (Baptiste, Abigaëlle, Angela and Quentin) and "Back to". This way, the game can be said to be a "BAAQ2 game", a "Back to game".

Name	Login
Baptiste Arnold	baptiste.arnold
Abigaëlle Panhelleux	abigaelle.panhelleux
Quentin Rataud	quentin.rataud
Angela Saade	angela.saade

Table 1: The members of the BAAQ2 team

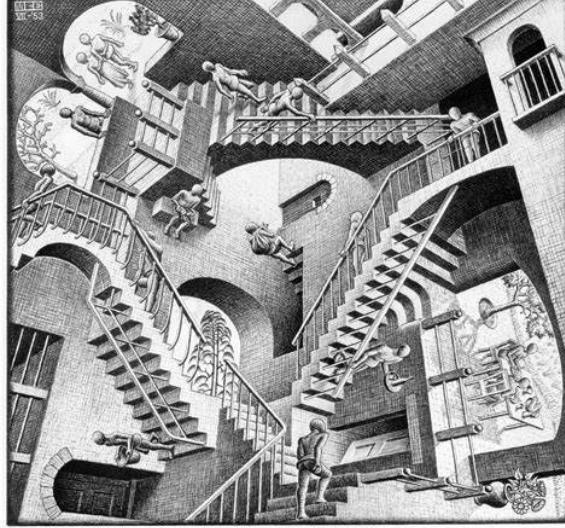


Fig. 3: *House of Stairs*, an example of drawing made by M. C. Escher

### 1.1.2 The game

**Concept** The game was decided to be a cooperative non-euclidean-looking puzzle game. It will be played by two players, who will have to advance in parallel in two similar mazes, cooperate as well as share information in order to progress through the rooms. The non-euclidean part refers to the world looking as if we played inside *Maurits Cornelis Escher's* magical world. Figure 3 illustrates a notable work from M. C. Escher.

**Origins** Many characteristics of the game were inspired from other games.

There are plenty of other games that allow one to play in these kind of impossible worlds. Notable ones are *Antichamber*, *Manifold Garden* and *Superliminal*, which all feature some physically impossible events and room dispositions. Figure 4 shows a sketch made by the developer of *Antichamber*, revealing the different parts of the map where the player gets teleported smoothly without noticing it, making possible the seemingly impossible room disposition. Figure 5 shows a similar concept we have done using a similar method. Blue rectangles and black arrows indicate from where to where the player will be teleported when traversing the portal. The resulting room looks like a square changing disposition as soon as we begin moving through it, but in reality the player is moving through three different rooms.

The game will nearly be monochromatic, allowing us to easily make optical illusions. From a particular point of view, an entire room could look like a 2D scene in black and white, thus telling a bit of the story behind the game. This is an idea that comes from a game called *Contrast*, where each room uses only two colors, making the disposition of the levels difficult to understand. Figure 6 shows an example of a level in the game *Contrast*. Figure 7 shows a similar visual effect we achieved using precise lighting and colors.

The game is to be played by two players simultaneously in a cooperative way, like some other cooperative puzzle games, *We were here* for example.

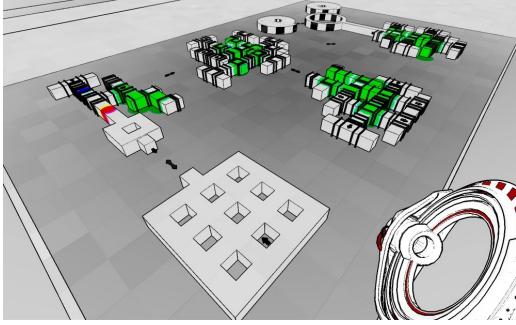


Fig. 4: Impossible connections achieved using discrete teleportation in *Antichamber*

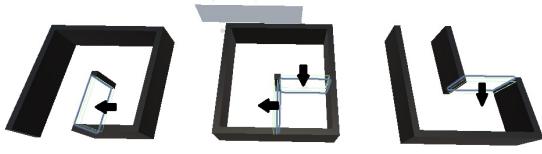


Fig. 5: Similar impossible connections made possible using teleportation in our game

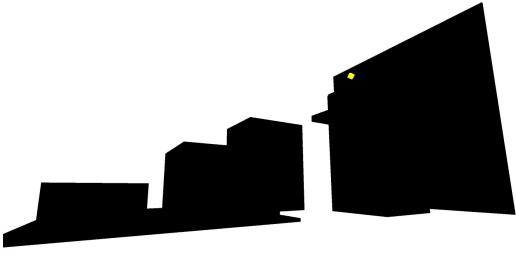


Fig. 6: A *Contrast* level, where it is hard to understand the room due to the presence of only two colors

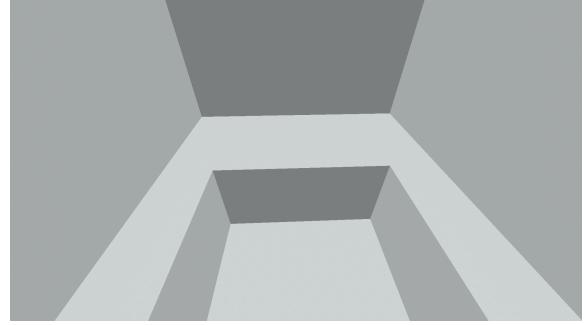


Fig. 7: A similar example of a room in our game where the lack of colors make it difficult to understand what is being seen

In the game, the two players will be chasing something that will guide them through the “maze”. Because of this and of the characteristics of the game, the game has been decided to be called “Hyper Pursuit”, to reflect the “impossible” traits of the game, “Hyper” being a reference to a fourth dimension.

The game backstory features two twins in a coma, having lost their entire memory. This is inspired from the game *Five Nights at Freddy’s 4*, that takes place inside a hospital. This is also inspired by a friend having lost his entire memory recently.

**Game Backstory** The game starts with a cut-scene of the two characters. In the maze that is their own mind, their goal is to recover memories from the past, in order to wake up finely. Their backstory is told in parts through cut-scenes and different events. The main story-telling thing is the disposition of the levels, describing events the two characters lived in the past.

In general, the two worlds will be very similar, except that when one is mainly black, the other will be mainly white. They need to help each other in order to find an object that is flying in both of their realm, which is the main gaming part. If both players successfully catch it, the game ends with both of them exiting this world forever. But if they do not pick up every collectable

representing their memories, they will not wake up when quitting this world. In every level of the game, the two players will be able to pick up collectables. The collectables will be found by placing the character in a very specific place of a level, making the room itself describe a scene, and being the starting image of a cut-scene telling what the protagonist is remembering. If both players pick up every single one of them and both catch the final key to end the game, they do not end up with the beeping machine: they rather both open their eyes, finally.

**Logo** The logo is a striated drawing of a man, who is running upside down, with a blurred reflection of himself. The logo only uses the two main colors of the game, black and white. Figure 1 shows the current logo of the game.

The running character represents the protagonist's pursuit of the world's exit key. The striated drawing represents the coma, where nothing is really well-defined. The reflection of the character is its twin, blurred because the memories involving the other character are initially gone and slowly coming back to their mind. The main character is running upside down to represent the lack of rationality in the bizarre world of the coma.

Figure 8 shows an illustration of the game and how the game is aimed to look like. This illustration can easily be put in parallel with the current logo of the game.

The illustration and the logo were drawn by Nino, using *Autodesk Sketchbook*.



Fig. 8: Illustration of the game

## 1.2 Our Goal

**Our Ambition** This game has for objective to diversify multiplayer puzzle games. Nowadays there are much more solo puzzle games than multiplayer puzzle games even though gamers usually prefer to play with friends rather than playing alone. There has always been more enjoyment in playing in co-op than in solo whether it's for chilling, solving problems or just having fun. What's more, there does not seem to exist any well known non-euclidean looking multiplayer puzzle game. Combining both a co-op environment and a weird world could yield an interesting result that could attract many players. This is the main reason why the game will be a cooperative puzzle game.

**Benefit for the players** Like any puzzle game, playing *Hyper Pursuit* will make the player improve their cognitive function. In particular, for this game, logic skills will be highly useful in order to solve the different puzzles to progress further and further into the game, and pattern recognition skills will be highly used in order to pick up each collectable. They will also have fun and a huge satisfaction when solving the problems. The game will not disorient the players but will frustrate them: this is where the fun comes from!

*Hyper Pursuit* will also allow the player to escape reality for a short time, using its very unique backstory. Moreover, thanks to the multiplayer mode that is in the core of the project, players can have fun with their friends or can meet new people who have the same taste for games.

**Benefit for the group** The project has revealed to be very beneficial for every team member. Because none of us has any experience in making a video game, we all learned a lot. We learned how to use git in order to work on a group project as a team ; we learned how to use Unity to create a game and to use networking ; we learned how to use Blender to model 3D objects ; and we still have a lot to learn before the completion of the project.

### 1.3 Technical Means

**Unity** is the main software used for the project. For the making of the website, HTML and CSS have been used. The character has been modeled using Blender, which will be used later for other elements of the game. Some level design has been done in a *Minecraft* world before implementing them in Unity. For the trailer, *iMovie* will provide the necessary means. Sound will be taken for the voices and music with a *ZOOM H4n* mic.

**Git** We needed something to share our codes, our scenes, our prefabs... modify or fix the errors. Git offers all this possibilities quite simply. That's why we decided to use Git to maximize our workflow. So we set up a repository<sup>1</sup>. We added a gitignore<sup>2</sup> to avoid uploading each time 5Gb of useless files. We also added a git attribute.<sup>3</sup>

**Multiplayer** The game will not simply use multiplayer mode as a feature: multiplayer mode is the core of the project. We decided to create a game that can only be played in a two players mode. There will be no AI that can replace a missing person: without a working multiplayer mode our game is unplayable. When the game executable is launched, the first player can choose to host or to join the party, which will determine the character he will play. The two players could play together by being connected on the same European server. The multi is managed thanks to Photon.

**AI** The main Artificial Intelligence present in the game is the little object the players need to catch in order to finish the game. Its behaviour will depend on the position of the players, in order to never get caught before the final level of the game. It will be changing rooms, playing with both characters, going in and out of the two worlds they live in.

---

<sup>1</sup><https://github.com/Akilson/Hyper-Pursuit>

<sup>2</sup><https://github.com/github/gitignore/blob/main/Unity.gitignore>

<sup>3</sup><https://github.com/github-for-unity/Unity/blob/master/src/GitHub.Api/Resources/.gitattributes>

Task	Defense 1	Defense 2	Final defense
Website	Exists	Contains all information	Contains animations
Story	Nearly finished	Entirely written	Told through cut-scenes
Puzzles	Nothing	Conceived	Integrated in the levels
Levels	One is designed	Conceived	Implemented
Multiplayer	Nearly finished	Bug fix/finish	local multi
AI	Nothing	Nearly finished	Finished

Table 2: State of the project through the defenses

## 1.4 Project Breakdown

**Planning** Table 2 shows the advancement goal we had at the beginning of the project of every main task throughout different milestones, namely the three defenses. This planning still hold today : nearly every concept has been proven to be possible before the first defense. Some puzzles concept were made, and an example level was designed and implemented. A basic website was created, and the story is nearly finished. What's more, the multiplayer system has been implemented. The game should start to be playable, and an entire plan of the game should exist for the second defense. The game should be fully completed and be working correctly before the final defense.

Figure 9 shows the first Gantt diagram made for the project, with the planning for each main task. This diagram has been updated since. Figure 10 shows the updated Gantt diagram. Not all puzzles were conceived, so extra weeks were added for this task.

**Distribution of Roles** Table 3 shows how roles are distributed. Even though it is written that a task is taken care of by only two people, each task will be completed with the help of every member of the group. For example, the coding process of the website is done by the person in charge of it, but the layout, the design and the texts can come from every member.

Overall, Abigaelle will mainly be in charge of the 3D modelling and assets ; Angela will mainly be in charge of the level design ; Baptiste will mainly be in charge of the website and the multiplayer ; Quentin will mainly be in charge of the puzzle design.

Task	Abigaelle	Angela	Baptiste	Quentin
Website			in charge	substitute
Story	in charge			substitute
Puzzles	substitute			in charge
POC unity		substitute	in charge	
Level design		substitute		in charge
Level implementation		in charge	substitute	
Multiplayer		substitute	in charge	
AI	substitute			in charge
Cut scenes	in charge	substitute		

Table 3: Roles of the members

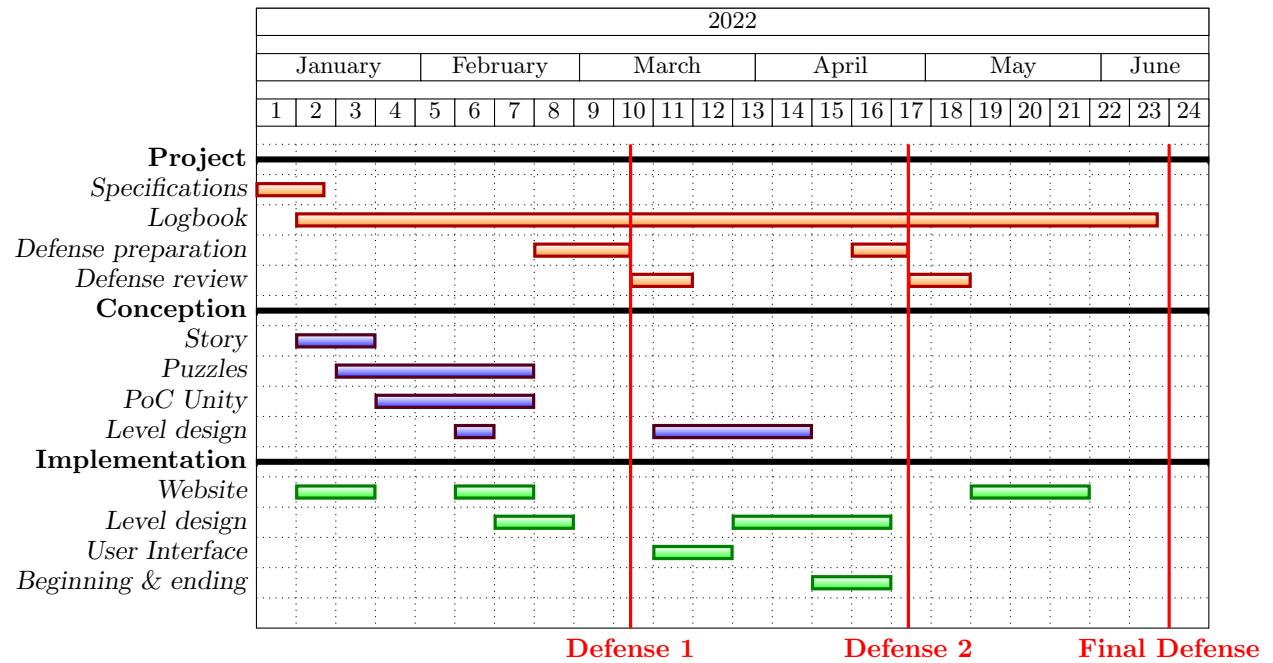


Fig. 9: First Gantt diagram for the project

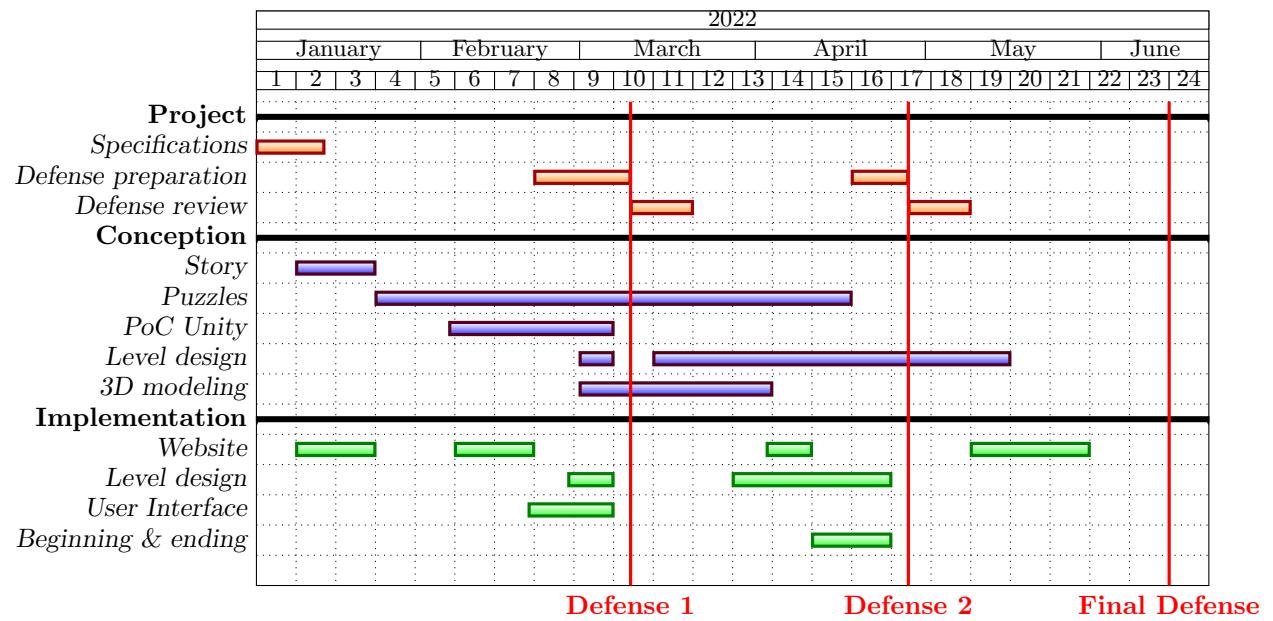


Fig. 10: Updated Gantt diagram of the project

## 2 Achievements

We have distinctly distributed our roles, that's why we will present for each person what they have done for the first defense.

### 2.1 Abigaëlle

#### 2.1.1 The Backstory

**Introduction** It was a lot more difficult than I anticipated. I needed to create a world that wasn't too complicated to understand, but also a world that could be told easily by little cut scenes. I wanted to still have a non obvious story, so I stuck with the "BAAQ2 team is in the lore" idea, and decided to focus on the characters we play.

**The Lore** One game, 6 characters, two point of views, one story.

This game is the story of 6 characters, all included in a tragic incident. This incident occurred not so long ago : the newspaper says that two police mans were severely hurt during a rescue mission. The two rescuers are twins : Kate and Aiden. They were investigating into a kidnapping case. 4 teenagers, who were studying in a programming university, were kidnapped by a man never caught by the police, James Dow. After an investigation, the two characters discover the location of the missing teenagers. Sadly, as the police mans arrived at the place, the kidnapper made the car crash, wanting to kill them. The twins got severely injured, and were put into a hospital quickly, while James Dow escaped. The only good news in this, is that the teenagers were abandoned by the kidnapper, so were set free and found by the rest of the police. The twins were put into a coma, and died a few weeks later. The game is actually made by the teenagers that were kidnapped. The game's purpose is to thank the twins that lost their lives for them, but also to tell everyone the identity of their kidnapper, find him and bring him to justice. The different rooms represent the five steps of grief. Collectables can be found in the game to give clues about the story. It is hidden, and the players have to be investigators, just like the twins, to gather all the clues and discover the truth. In the game, the game is structured into 5 chapters, each of them representing a room and an emotion of grief : denial, anger, bargaining, depression, and acceptance.

**Some facts about the twins** To be a bit more precise, I decided to create some facts about the twins, that will be used to find the collectables in the game.

1. **Denial** For the denial room, because the denial is like "he's mistaken", one moment of the lives of the twins could be a bit like that feeling. The first year of their studies in police, they had to try a major in investigation (detective) and play a game : in groups of two, they had to investigate a crime scene and discover the killer. Killer, that is one person in one duo of the class. The killer is supposed to do everything to make someone pay and not him. It was actually Kate the killer, and played into tricking Aiden. The memory could be when Aiden finds out that Kate was the killer (spoiler : she won, by killing his brother).
2. **Anger** To accentuate the fact that they are twins, a scene simple and a "Mom! I want the same thing as Kate!" can be done.



Fig. 11: Screenshot of the entrance of the starting cut scene

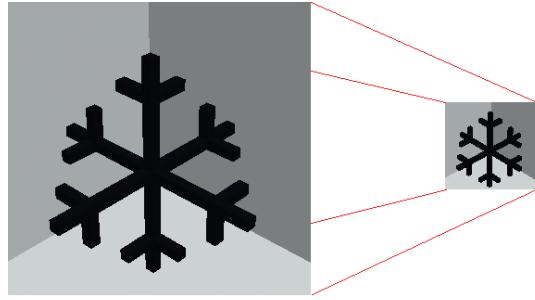


Fig. 12: Screenshot of an optical illusion made



Fig. 13: Screenshot of the starting cut scene

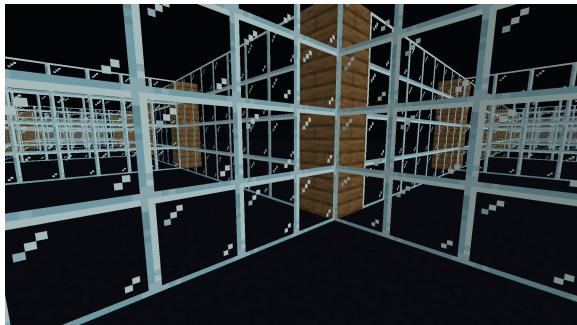


Fig. 14: Screenshot of the starting cut scene

3. **Bargaining, depression and acceptation** For these chapters, no cut scenes is needed, but there will be some discussions between the two characters. For the last one, at the very end, there will be a final cut scene, representing one of the end unlocked.

### 2.1.2 Level design and Minecraft

**Minecraft Server** Before doing the level design, we started, with Quentin, the minecraft server so that we could work together at the same time. I was the one that created the server, using Aternos, and we began using it. Because creating big rooms is complicated, we used a mod called WorldEdit and some command blocks.

**Starting cut scene** We had a bit of a hard time figuring out what to do, but I managed to conceive the first cut scene in Minecraft. Because it is too time consuming to do a playable cut scene, I decided to create rooms, one below the other, to represent the different steps of the cut scene. To navigate through the rooms, doors and ladders are posted. Next to the entrance, a lot of signs are attached to the walls near the door, to explain how the cut scene will play out.

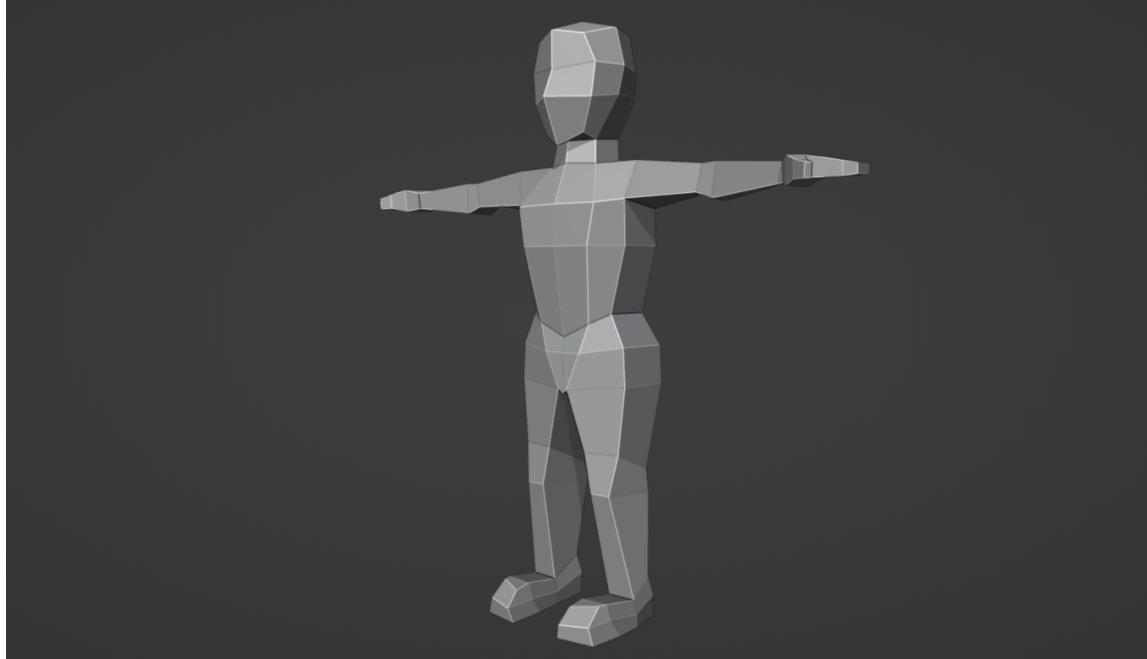


Fig. 15: Character design on Blender

**Optical illusion** We also tried to perform some optical illusion, and did a snowflake, words on walls and obstacles, and finally we started doing some teleportations. It was at first pretty simple : two corridors, and two teleportation points. When the character touches the teleportation point, it is teleported to the second teleportation point, and vice versa. To the eye, the character does not move, and it is the corridor that, with magic, goes back to the same room.

### 2.1.3 Character design

I started having this idea after doing a tutorial made by Unity. In their tutorial, we had to create a mansion game, and there was a character with a filename that I did not know. With a bit of research, I found out that we could, with some 3D modeling app, import 3D character. With that in mind, I downloaded Blender. I knew that we wanted a character design that could be both for a woman or a man. Because I didn't know anything of Blender, I decided to watch the tutorial videos Blender gives to the users on YouTube. Then, it gave a tutorial to make a owl, and I decided to put what I learned into motion.

After finishing the owl, I started to work on the character. With the team, we decided to have a low poly character, that is a character with a low number of polygons, with no face. Having no face and minimized hands and feet is a characteristic that dreams have. To do that, I watched a tutorial on YouTube, and changed some details to fit what we wanted to have. For the shading, it is not yet done, but will be quite simple : only a white skin for one of them and the other a black skin.

## 2.2 Baptiste

**Multiplayer-Photon** The game and his puzzles involves two players, so we must have a working multiplayer. We had multiple choice and we finally decided to use Photon. We have chosen Photon because it is beginner friendly, the only disadvantage that we can find to Photon it's that each room are limited to 16 players on the free version. But since our game is only played by two persons, we only needs 2 players per room, hence the small limit per room don't bother us.

For the moment our multiplayer allow us to create and join rooms, find room and launch the players on a game map. The multi works in a simple way, first we have a menu (lobby) manager then we have the game and the players:

1. Menu Manager
  - Figure 16 shows that when the game is launched, we try to connect to the photon server, once it is done, the main menu scene is displayed.
  - Figure 17 shows that on the main menu we have 3 choices : either we find a room, either we create a room, or we can just quit the game.
  - Figure 18 shows that the find room menu is composed of a list of all the rooms that exists on the server (the server is limited to Europe).
  - Figure 19 shows that the create room is composed of an input field, that will be the name of our room displayed in the find room menu. The one that creates the room is automatically the host thanks to Photon.
  - So once we join or create and join a room the room menu is displayed with a list of all the players and a start game button to launch the game only accessible to the host as shown on Figure 20.
2. Player Instances
  - Once all the players join a room we will launch the game with an instance of a players for each person in the room.
  - For this we have a room manager that is unique and exist until there is nobody in the game.
  - The room manager will instantiate a unique player manager (thanks to Photon view) for each person in the room.
  - The player manager instantiate a player controller that is the capsule as shown on Figure 21 that represent our player with the associated code that allow us to move on the map.
  - Thanks to the unique player manager even though the code of the player controller is the same for everyone each person control his character.
  - And so when a player quit the game it is only the player manager and controller that will be destroy.

During the implementation of the multi we encountered many problems. We first started without a lobby manager. So we could only join a room when we know the name of the room. Then our instances were a complete mess. The cameras was not synchronizing, there was only one camera for all the players, the only thing that was working was the movement that were synchronized. So we



Fig. 16: Loading scene



Fig. 17: Menu scene

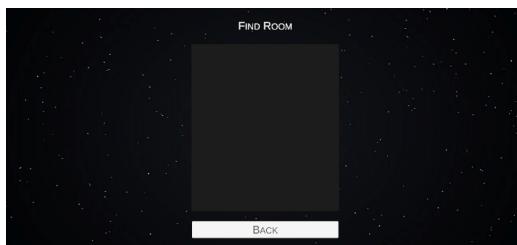


Fig. 18: Find Room Menu scene

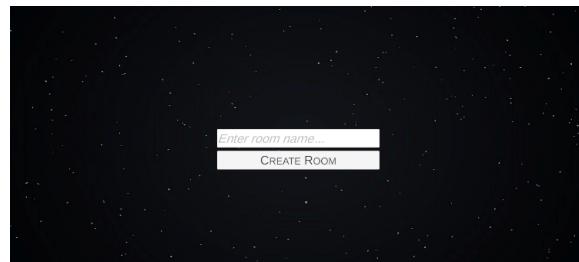


Fig. 19: Create Room Menu scene

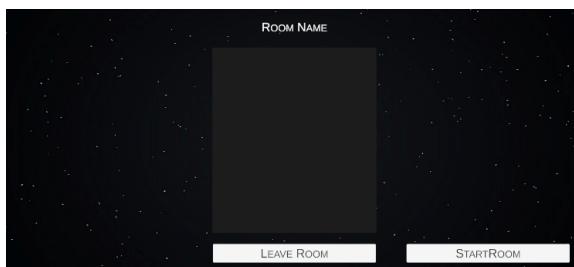


Fig. 20: Room Menu scene

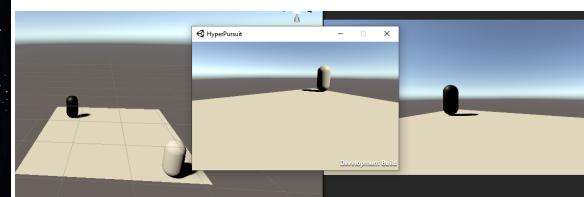


Fig. 21: Players Instances

decided to follow a more complete and longer tutorial that explains step by step from scratch how to create a multiplayer (fps that we adapted for our puzzle game) on photon. So, after following this tutorial we succeed to have a working multiplayer. But at that time, we were not working on a common repository. We don't know why but when we import the multiplayer to the common project there were many errors (unity was not detecting the photon view of the player, unity could not find the prefabs...), we don't really know why and where the errors were coming from so we just re-coded everything that was not working from scratch on the common project. Finally, we had a multiplayer that is working. There is just one problem on the find room menu (each time we create a new room the room are not stacking on the find room list of rooms, it is just overwriting the precedent room), we will try to fix it for the next defense.

**Website** <sup>4</sup> The person in charge of the website had no knowledge about HTML and CSS, that's why we decided to make the website from scratch even though we could just copy and modify a template. So, the first website is like a playground were the Baptiste trains his web skills. The website will change for each defense. For the next defense we will use a template and complete all the information and for the last defense we will try to hide some hints on the website so that the website is in the lore of the game.

For now, our Website is classic, there is a footer with all out social networks links, a header with links that refers to the different pages of our website and the title of the page and a body that contains all the necessary information.

HTML is really simple, but we had some trouble to understand how to display correctly the image and the block text with CSS. For example, we have used flexbox a little randomly, just by (brute forcing) trying until it works. We mixed sometimes "display flexbox" and "display block". In fact, HTML and CSS are not hard to understand but it was really hard to imagine the website. We started randomly so we have lost a lot of time to rewrite the HTML to create block according to how we wanted to display something.

---

<sup>4</sup><https://akilson.github.io/website/>

## 2.3 Quentin

### 2.3.1 Portals

**Render** The main method we used to achieve impossible room disposition was the use of portals. Each portal is linked to another portal. This way, when the character traverse through a portal, he get teleported to the corresponding portal smoothly.

In order to achieve the smoothness of the teleportation, a specific image is computed and displayed on the portal at each frame. This is done with the use of *RenderTextures*. A well-positioned camera outputs its view on a RenderTexture, associated to the Material of the portal. The position of the camera is computed in a C# script. The camera is placed in order to have the same view of the second portal as the player see the first portal. A shader is used on the material in order to “cut” the portal from the view of the camera and paste it on the Render plane.

Figure 23 shows a simple example of a room which use two portals. When the character – represented as the default grey cylinder – traverse the black door, it gets teleported to the other room as if the other room where right behind the wall, in spite of the room being slightly rotated. Figure 24 shows the view of the camera and the view of the player. The corresponding portal viewed from the camera is copied to the material of the portal viewed by the player. From the point of view of the player, it looks like the other room is located right behind the door.

It is to note that, in order for the camera to see the room behind the portal, the black wall between the portal and the camera has been made transparent to the camera. This has been done using *culling masks*.

**Teleportation** Because we do not want the player to hit the portal and slow down during the teleportation nor to see what is behind the portal, the player need to be teleported a bit before reaching the render plane. This is why we use a *Collider plane* which will execute the script to teleport the player once he collide with it.

Once again, we do not want the player to get teleported to the Collider plane of the corresponding portal, because then the player would either get teleported back (since he would be colliding with the other collider plane), either be able to reach the render plane if we do not teleport it. To fix this issue, we teleport the player to another *Receiver plane*, located further away from the render

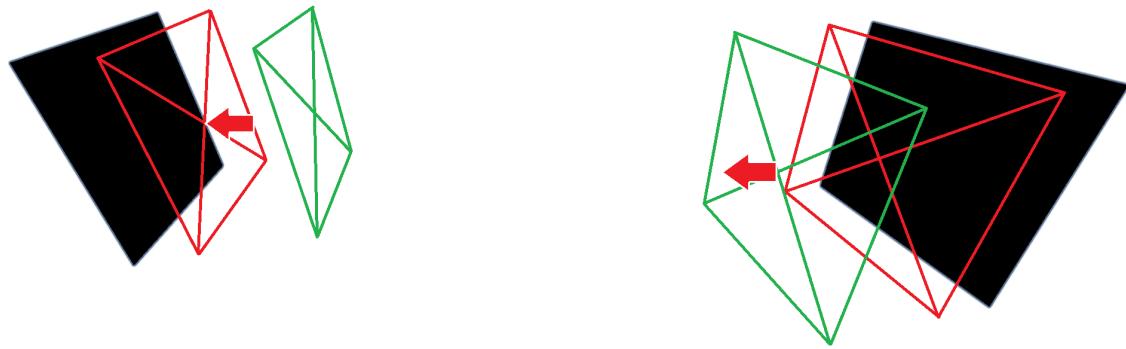


Fig. 22: How the teleportation works through portals

plane.

Figure 22 shows an example of placement of a pair of portals and their respective planes. The render planes, displaying a cutted part of a camera, are represented by a black rectangle. The collider planes are denoted by red rectangles, and the receiver planes are denoted by green rectangles. When the player collide with a collider plane, he is teleported to the corresponding receiver plane.

The offset between the planes needs to not be too small – otherwise the player could collide with the other collider plane when teleported – and not be too large – otherwise the player would get teleported too soon.

The placement of every plane needs to be extremely precise, as any offset would result in a visual glitch that would betray the entire effect.

### 2.3.2 Proof of Concepts

In order to use the portals as efficiently as possible, their use needed to be as simple as possible. That is why a prefab of two linked portals has been created.

To explore the field of possibilities of what can be made using portals, some little proof of concepts has been done.

The example Figure 23 is one of them. No matter where the two rooms are located, from the point of view of the player, the two rooms seems to be next to each other.

The second concept done was an impossible tunnel, looking smaller from the inside than from the outside, using only a pair of portals. Figure 25 shows how the tunnel looks like from the inside. Figure 26 shows how the tunnel looks like from the outside. The result is incredibly disturbing. The player can go through it instantly, but has a long way to go to contour it.

This effects is the result of a pair of portal located near the extremities of the tunnel. Figure 27 shows how the portals are placed in order to get this incredible effect.

A third concept was a small room, where it seems like we need to do eight 90-degrees right angle to explore it entirely, even though there only seems to be one pillar in the middle of it. Figure 28 shows the transition as seen by the player when entering the room. It seems like the room is closing right behind the pillar, and by doing two full turns around it the exit appears from behind it.

In reality, this effect is the result of two pairs of portals. Figure 29 shows where the two portals are located. It is noticeable that the effect works the other way around.

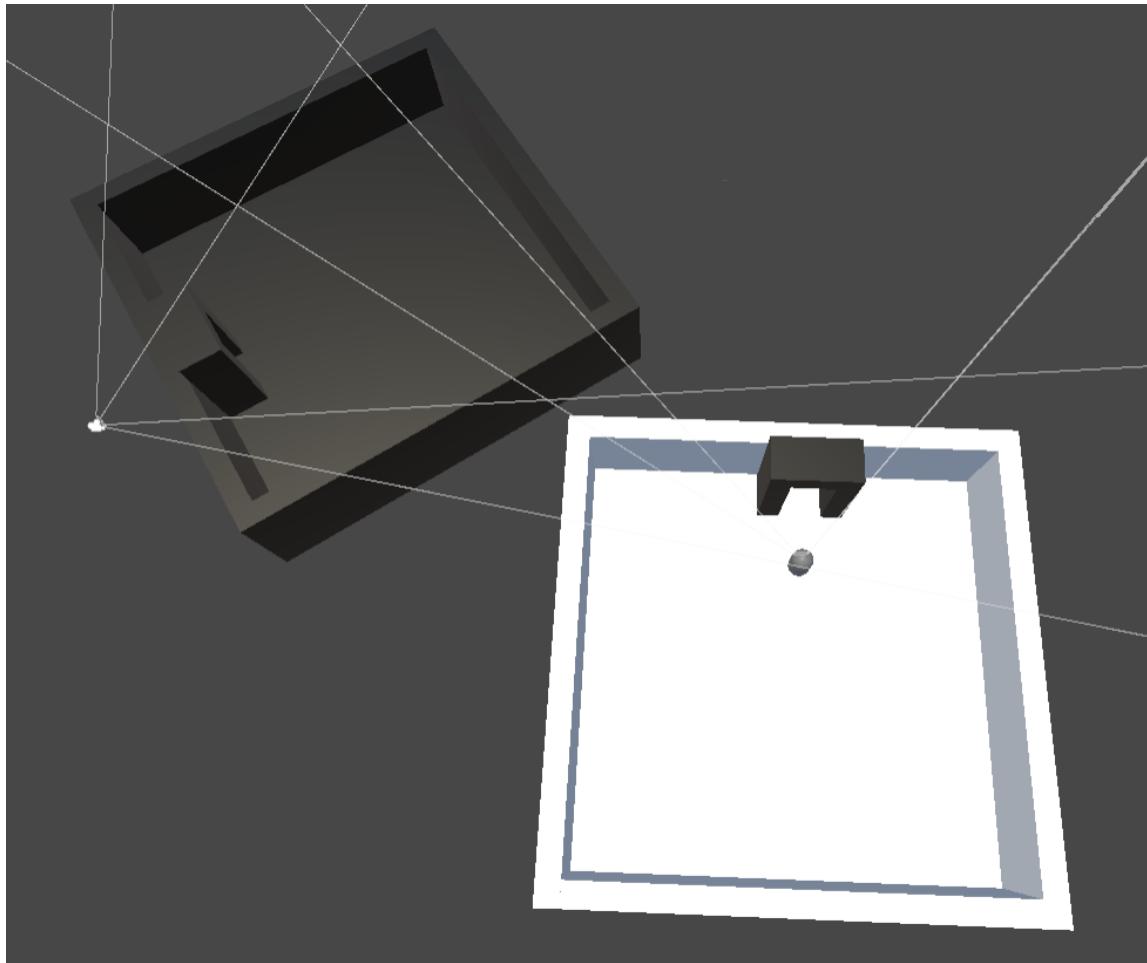


Fig. 23: Example of a simple use of a pair of portals

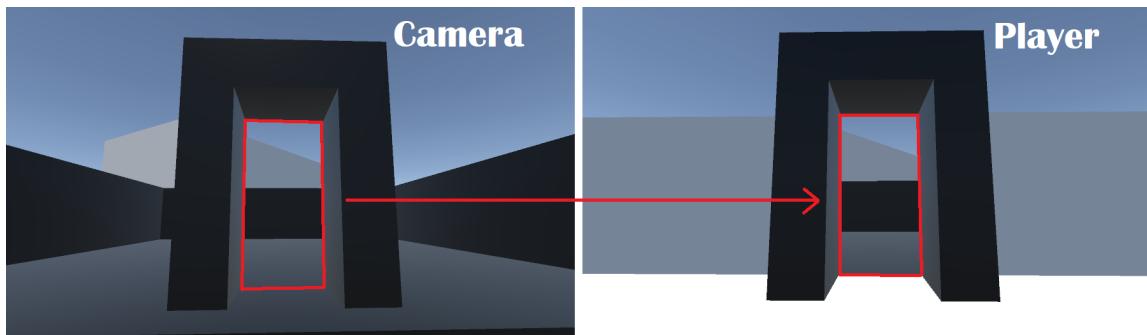


Fig. 24: Functioning of the render plane of the portal

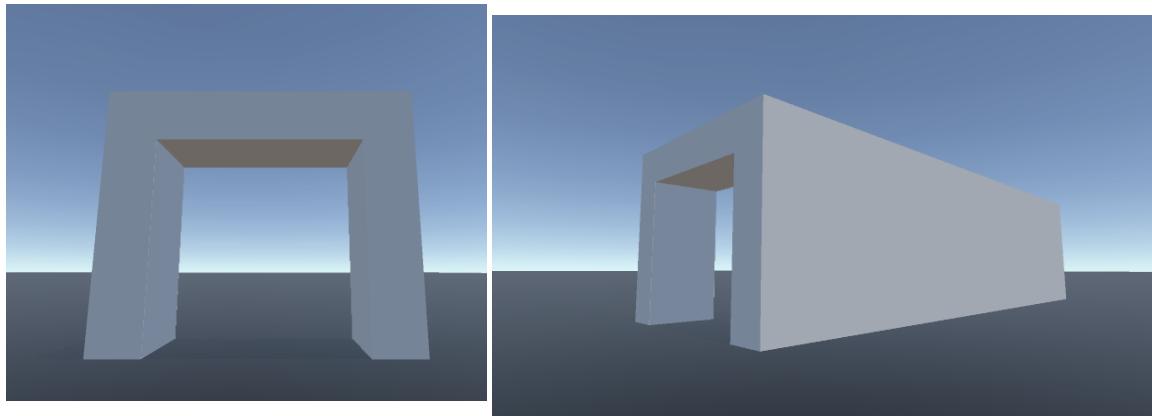


Fig. 25: How the tunnel looks like from the inside

Fig. 26: How the tunnel looks like from the outside

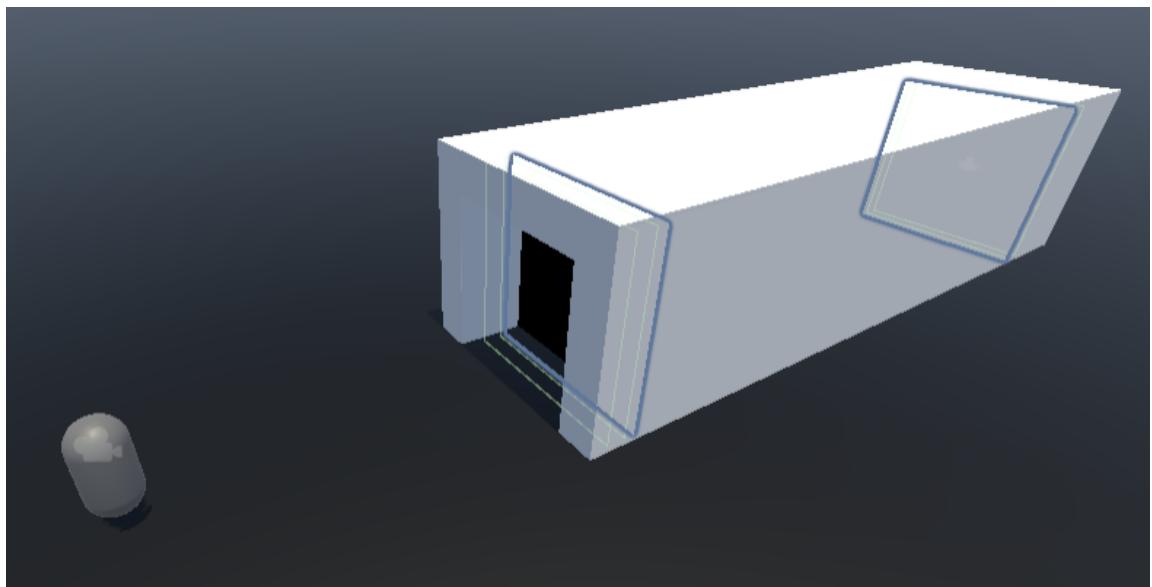


Fig. 27: How the portals are placed

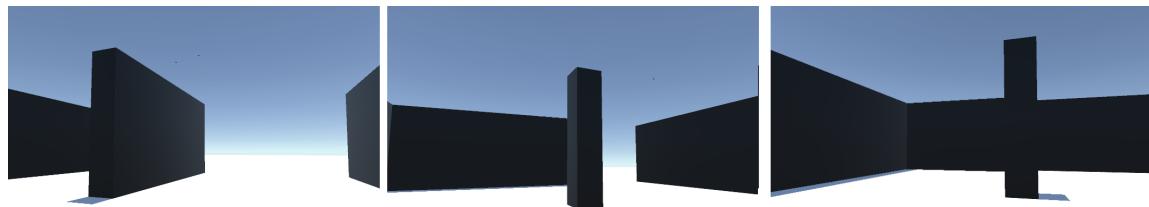


Fig. 28: The transition the player experiences when entering the room

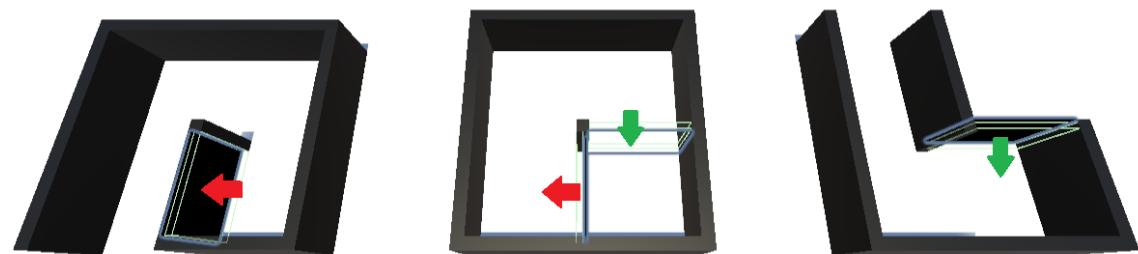


Fig. 29: How the three rooms are combined into one using portals

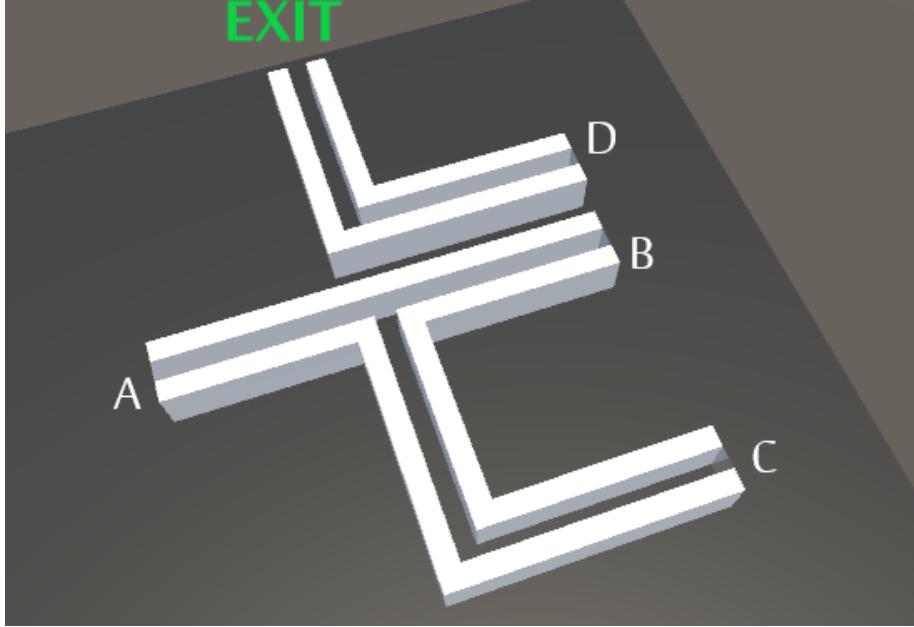


Fig. 30: Example of a room that represents upcoming levels

## 2.4 Angela

**Test Room Concept** We tried to implement a room which is a small representation of the obstacles that will be in each level.

This room contains a hall which leads to a point where the player needs to choose whether to go right or left to continue his level and find the exit. If he chooses to go to the right, he will return to where he started as well as if he chooses to go to the left. So he will be in an endless pattern unless he figures out the suitable way to escape it. The correct way to reach the exit room is to do a half-turn and go through that portal which will transport him to the final room, the exit.

We used four portals: A, B, C and D as seen in the Figure 30. The two portals A and B are connected to the portal C and the C is connected to the D. This means that if the player goes through either the door A or B he will pop out in front of the door C. When he does a half-turn and goes through the C he will pop out in front of the door D and will reach the final hall leading to the exit.

**Test of a Pair of Portals** To get the hang of using portals, we tested at first a teleportation with a pair of portals and tested the teleportation. Which did not work at first because of the placement of the trigger plane. In other words, the player could see through the portal the correct image and he can pass through it, but the teleportation was not possible; he just passes through the door but he is not teleported to the destination place. That was caused by the adjustment of the position of the portal, the trigger plane was placed far from the portal. In other words, everything was moved except for the collider that was left behind. The problem was quickly fixed by adjusting the coordinates of the collider and the portals worked perfectly afterwards.

Then, the work for our example room started.

**Test Room** Since our test portals were functioning, we used them for our example room. And duplicated the portal A and B to create the portals C and D respectively with the suitable modifications. But a problem quickly appeared; the player could not be teleported through the portal A because a teleportation between two portals of the same type is not possible. We had to modify the coordinates of A so they can become similar to those of B and become compatible with the teleportation. By doing that the problem was fixed and the teleportation between the portals A and C worked.

The transition from the current door to the destination door has to be extra smooth so the player does not suspect a thing and so we can tease him a bit. At first, the objects' scales and positions were not created with precise numbers. This caused a slight problem with the teleportation from a hall to another. The player could clearly see the difference between the current place and the upcoming one. A hall was rebuild with precise numbers and this hall was duplicated and modified. In addition, it was used to create the rest of the room so all the halls will have the same dimensions and width, so that the transition would be perfect.

Moreover, by moving from a place to another the lighting differs. So we had to change the brightness to have the exact luminosity in both places so it can be coherent.

### 3 Conclusion

#### 3.1 What was achieved

Briefly, many achievements were made. Firstly, our backstory is completed. Secondly, a character was designed using Blender, a version of our website was created and assets of our portals were created and implemented. In addition, we implemented the multi which is mainly functioning, many proof of concepts were implemented and every concept was proved to be possible. As well as a test room containing an obstacle of future levels was created.

#### 3.2 What has to be done

For the second defense, a second well updated version of the website is expected to be done. Moreover, finding new ideas for the puzzles, designing levels and creating cut-scenes. An implementation of as many puzzles and levels as possible is awaited to be executed. For the lore point of view, it needs to be cut into non chronological parts, and be “implemented” into each of the future rooms.

We will fix the multiplayer, try to limit the rooms to 2 players, assign to our player a role depending on their attribute – the host will be white, and the client will be black for instance –, create spawn points to instantiate our player on different locations depending on their role. Then we should have a complete multiplayer for the next defense that we may not change anymore. We will also work on the UI : add a settings menu to be able to quit the game in game for instance.

The portals will be made even more portable in order to simplify their use as much as possible. Then they will be tested in multiplayer, as we are only sure they work in a single player mode for now. Then, they will be used to create some little puzzles that will be interesting to play with two players at first, in order to create many complete levels for the next defense.