

# Hyper Pursuit

## Defense Report 2

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

January 2022

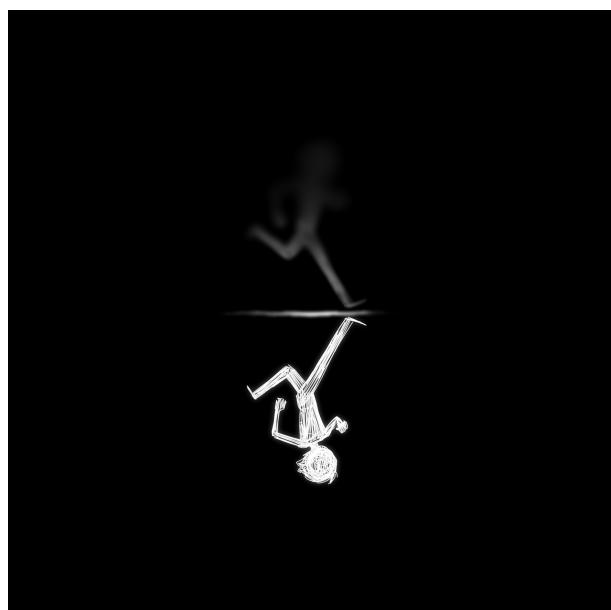


Fig. 1: The game logo

## Contents

<b>1 Specifications follow-up</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.1.1 The team . . . . .	3
Members . . . . .	3
Name . . . . .	3
1.1.2 The game . . . . .	4
Concept . . . . .	4
Origins . . . . .	4
Game Backstory . . . . .	5
Logo . . . . .	6
1.2 Our Goal . . . . .	7
Our Ambition . . . . .	7
Benefit for the players . . . . .	7
Benefit for the group . . . . .	7
1.3 Technical Means . . . . .	8
Git . . . . .	8
Multiplayer . . . . .	8
AI . . . . .	8
1.4 Project Breakdown . . . . .	9
Planning . . . . .	9
Distribution of Roles . . . . .	9
<b>2 Achievements</b>	<b>11</b>
2.1 Abigaëlle . . . . .	11
First cut scene . . . . .	11
Level design . . . . .	12
2.2 Baptiste . . . . .	13
Bugs . . . . .	13
Multiplayer . . . . .	13
Website . . . . .	14
AI . . . . .	14
UI . . . . .	14
2.3 Quentin . . . . .	15
2.3.1 Multiplayer portal . . . . .	16
2.3.2 Collectable . . . . .	17
2.4 Angela . . . . .	18
Denial . . . . .	18
2.4.1 The butterfly . . . . .	18
2.4.2 Itinerary of the players in the maze . . . . .	18
<b>3 Conclusion</b>	<b>20</b>
3.1 What was achieved . . . . .	20
3.2 What has to be achieved . . . . .	20



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

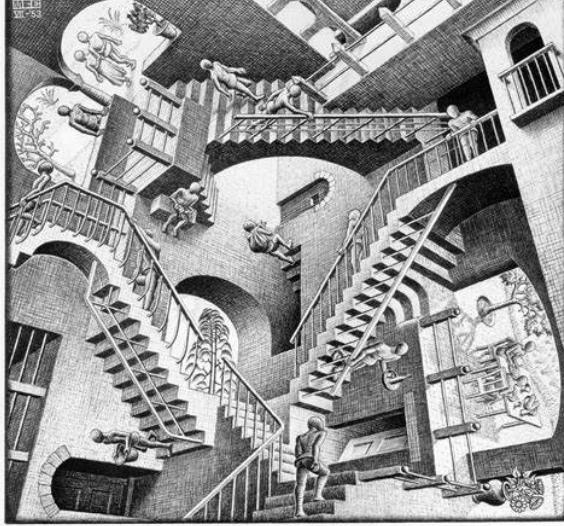


Fig. 3: *House of Stairs*, an example of drawing made by M. C. Escher

### 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.

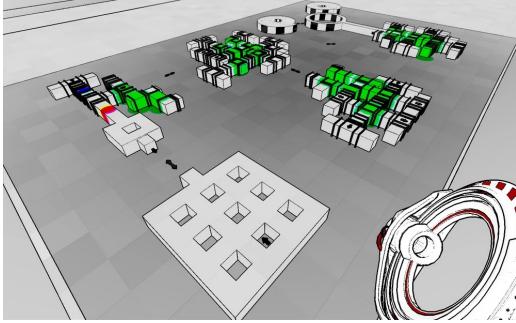


Fig. 4: Impossible connections achieved using discrete teleportation in *Antichamber*

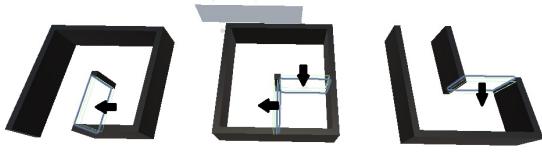


Fig. 5: Similar impossible connections made possible using teleportation in our game

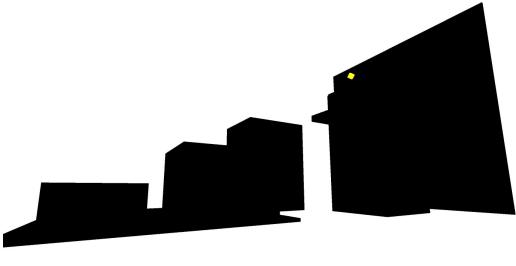


Fig. 6: A *Contrast* level, where it is hard to understand the room due to the presence of only two colors

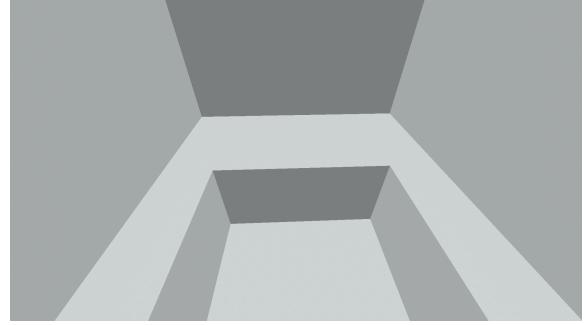


Fig. 7: A similar example of a room in our game where the lack of colors make it difficult to understand what is being seen

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** Git is the main software used to optimize our workflow. It allows us to share our codes, our scenes, our prefabs, and to work in parallel. There is a repository<sup>1</sup> for the project, conveniently configured with a specific `gitignore`<sup>2</sup> and `gitattributes`<sup>3</sup> made for Unity projects, to avoid uploading binaries and non-necessary files.

**Multiplayer** The game do 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 not playable. When the game 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 can play together by being connected on the same European server. The multiplayer 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, a predefined path and its distance to the walls, 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 end of the first defense, for 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 Gantt diagram made for the project at the time of the first defense, with the planning for each main task. This diagram has been updated since. Figure 10 shows the updated Gantt diagram. Level implementation finishing before level conception was a mistake and has been corrected. 3D modeling took longer than expected, and User Interface has been reworked recently.

**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, Abigaëlle will mainly be in charge of the 3D modelling, assets and animations ; Angela will mainly be in charge of the level design ; Baptiste will mainly be in charge of the website the multiplayer and the AI ; Quentin will mainly be in charge of the level implementation.

Task	Abigaëlle	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

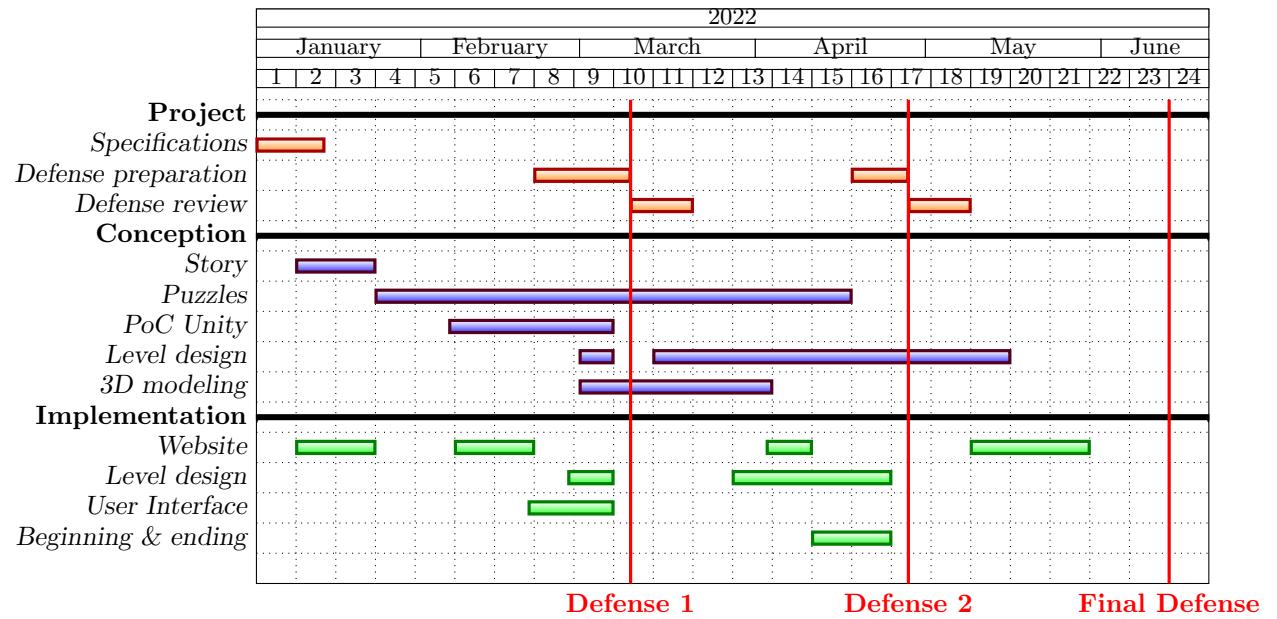


Fig. 9: Gantt diagram for the project, at the time of the first defense

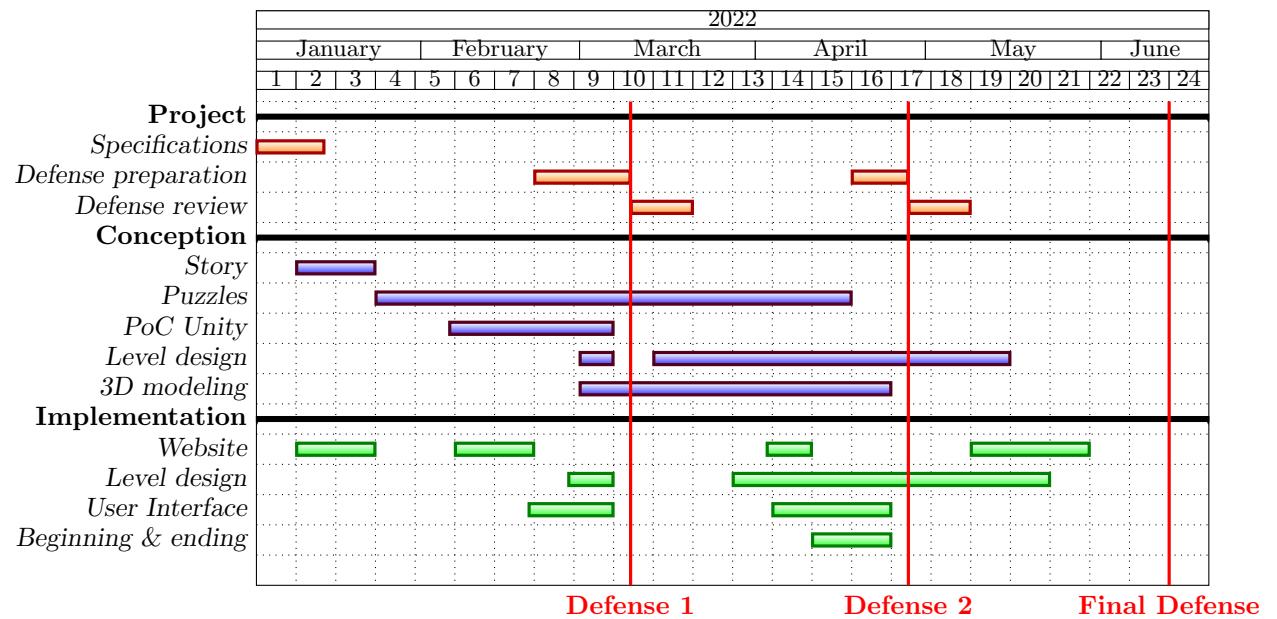


Fig. 10: Updated Gantt diagram of the project

## 2 Achievements

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

### 2.1 Abigaëlle

Between the first and the second defense, my main role was to create the first cinematic and add some details to the first level.

**First cut scene** I had a bit of an idea of what I wanted to create : I wanted a first cut scene that was simple, but that left a lot of questions. I wanted the two characters in the same room. I took a lot of time to think about the cut scene, how I wanted it to be played, and this is what I finished with : the two characters are in the same room, with a wall that separates them. They wake up, get up, and see each other through a mirror. But it doesn't seem like a mirror, because their color isn't the same : in the mirror the character behind has another skin color (black for the white one and white for the black one). He tries to touch the mirror but faints before reaching it. It is a simple cut scene but now the problem was to make it... I already made the character, so changing the skin color wasn't difficult, but animating it will be, because I never animated before. At first, I wanted to do everything in Unity, but I found it too difficult, so I decided to make it fully on Blender, create a video and use that as a cut scene. So I started watching tutorials about animation, and the first thing that I needed to do is to create an armature for the character. I think this was the most time consuming thing that I did. I started creating the room, the armature, and as I was finishing it... My pc crashed. And the automatic saves didn't work, so I lost a lot of work. Work, that I had to redo, and here it was a lot easier because I knew what I was doing. I finished the armature, and started the animation. It was tricky at first, but a lot easier to do then the armature. I got quite the hang out of it, and finished the animation. But another problem that I encountered was the camera. I didn't know how to animate the camera, and the field of view was too narrow. After some research, I managed to find how to fix it and finished the recording. I just had to add some effect of blinking and done!



Fig. 11: snapshot of the first cut-scene



Fig. 12: The stalactites that were made for the rooms

**Level design** For the level designing, I had to add some object into the scene. With the help of a friend of Quentin, I was left with some stalactites. I started to add them randomly to the first level, but was not able to fully implement them for some corruption problem. For the collectable, we wanted something that was in relation with the denial. Denial, it's having no hope. Or at least beginning to feel the infinite despair. The little cut scene will be the two characters, children at that time, chasing a butterfly, but cannot catch it. The collectable will then be this butterfly, hidden in the walls of the level.

## 2.2 Baptiste

This section discusses the modifications made on the website and the multiplayer, the beginning of our final AI and the creation of a very simple UI.

During the last defense we presented a classic multiplayer with the possibilities to join or create rooms. But it was not adapted to our game. For example, the maximum number of players that could join our room was 16 and not 2. So for this defense we have completely finished our multiplayer by adapting it and fixing some bugs that were present last defense.

**Bugs** During the first defense the display of the rooms wasn't working. Each time that a room was created it was displayed by destroying the last one. As it is shown on figure13 only one room could be joined even if there were 4 rooms on the network. To fix this we used a dictionary that has as key the name of a room and as value the info of the room. So for each room present in the RoomListContent (it's a list that contains all the rooms) we add it to the dictionary. Then for each room in the dictionary we display it on our list of rooms as shown on figure14.

**Multiplayer** Once the mutliplayer is fixed we needed to adapt it. So we wanted to activate the launch game button only when there were at least two people, to limit the maximum number of players per room to two and so block the access of a room while displaying a message of error like "room is full". To activate the start button only when there is two players, we just checked if the player is the host or not. We set the button active to the person that is not the host because it means that there is at least two players. Then to block the access to a room where there are already two players, we just checked the PlayersList available for each room in the roominfo thanks to photon. If the length of this list is less than two then the player can join the room. Otherwise we change the name of the room to "room is full" so that on the list of room it is displayed "room is full". Finally, we also created two different prefabs of players so that a different character controller is instantiated depending on if you are the host or the client. So our multiplayer is now finished, the only thing we could change is the design.

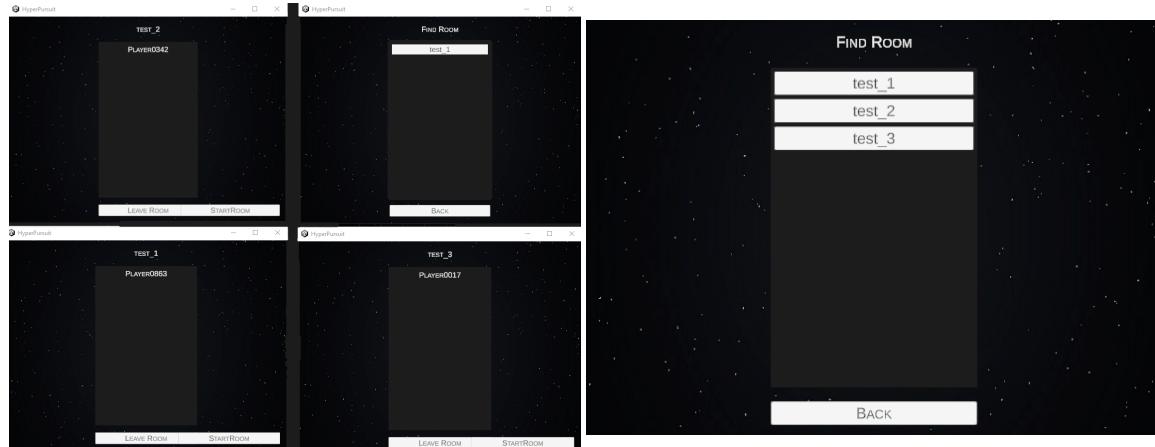


Fig. 13: Find room scene (not yet fix)

Fig. 14: Find room scene (fix)

**Website** The website hasn't changed a lot since the last defense, we just changed a little bit the design. We changed the background, passing it to full black and we added a new font style, as show on figure15. But we are working in parallel on another website. We will chose on the final defense the website we want for our game. On one hand we have a website made from scratch, so we know what it does and how to easily change it, but the design is really simple. On the other hand we have a website inspired from a template, that has a really nice and professional design, but we haven't totally done it.

Now that everything we have prepared for the first defense is fixed, we still have a lot to improve. For example last time we couldn't quit the game without doing alt-f4.

**AI** During the game, the players are trying to catch up their memory. So we wanted to have something that is constantly escaping from the players. We decided to make this with an AI. The AI will follow a path while escaping from the players. For now we have a capsule that follows a path determined with checkpoints, so it is a simple patrolling AI for now.

**UI** Our UI is very basic because we are not doing a puzzle-game that needs a well developed and very nice UI. All the hints are included in the maps and the players don't have a lot of mechanics (the player moves and that's all), so we don't need a sophisticated UI. So we just created a pausemenu that allows you to quit the game or change the settings that are for now composed of a volume slider and a quality drop-down.

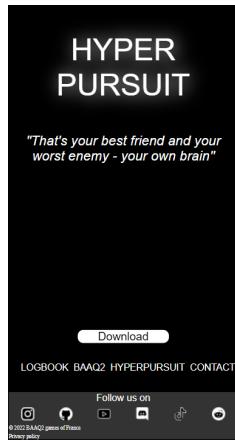


Fig. 15: Main page



Fig. 16: Pause Menu

### 2.3 Quentin

This section focuses on how the portals and different game mechanisms have been adapted to support the multiplayer mode.

Section 2.3.1 discusses how portals are working on multiplayer, issues related to it, and the tricks used to solve these issues. It will also go through new game mechanisms deriving from how these issues were solved.

Section 2.3.2 introduces collectables and what they look like.

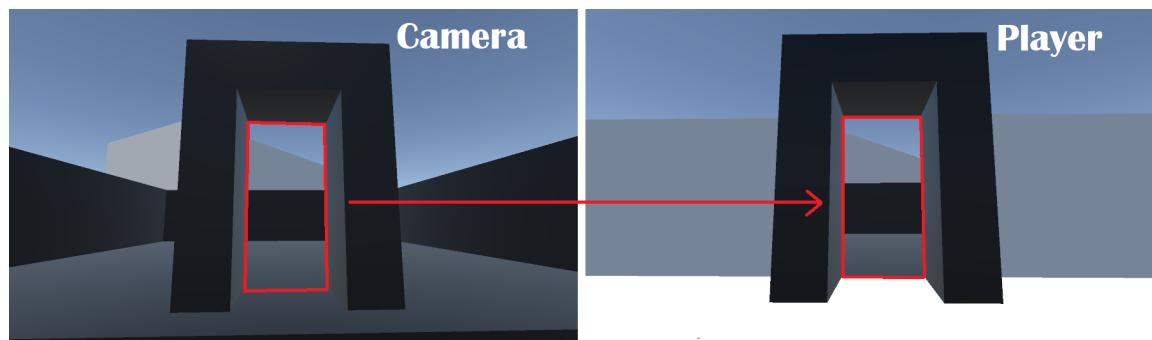


Fig. 17: How the image to render on the portal is computed

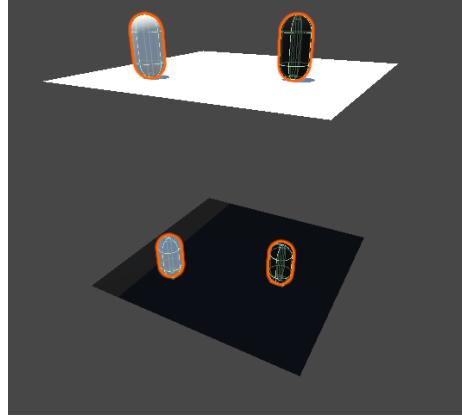


Fig. 18: Each players version of the map

### 2.3.1 Multiplayer portal

The portal magic resides essentially in the render plane, where a well-computed and really specific image is rendered. This image is computed from the relative position of the player, from the portal and the distance between portals, as shown in figure 17. The image is then set as a material for the render plane.

The first issue encountered was the detection of the player. In single player mode, the player was a GameObject present in the scene, as well as the camera of the player, that could be easily dragged as a public attribute of every script without any difficulties. But in multiplayer mode, things are different. Players are not present in the scene, but are instantiated from a prefab at the start of the game. Every script was rewritten in order to detect the player using a search by tag at the moment where they appear in the scene, and players' cameras are automatically detected from the structure of the player's prefab.

And there comes the main issue : What if there are two players ? As both players have a different position, we would need a specific image for each player to be drawn on the portal. But the portal cannot have two different images rendered at the same time.

One solution could have been to choose which image to display depending on whether the game is launched as the host or not. But the trick that was actually used is totally different. The entire map is duplicated, and each player will play on its own copy of the world. Portals will only work for one specific player, but never ever two players will be able to see the same portal at the same time, so the issue is solved !

One last thing remains, is that if the two players are in two completely separated worlds, they will not be able to see each other. That is why each player will have on his version of the world, a "spectrum" of the other player, which is, a visual representation of each player on the other player's part of the map. This will allow each player to have a good idea of where the other player is. Figure 18 illustrates this principle. The white player is located on the top plane, and the black player is located on the bottom plane. Each player can see a "spectrum" of the other player on its plane, even if the player is not really there.

What's more, this allows us to easily make slight modifications between the two players' maps ! For example, we will revert the colors for one player, so black on white becomes white on black, to

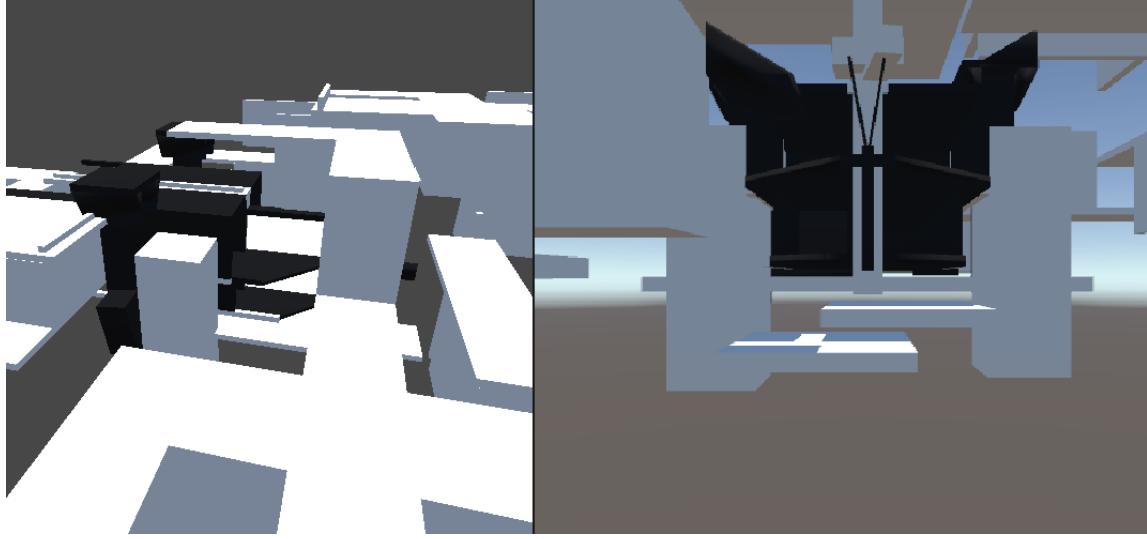


Fig. 19: The butterfly collectable

match the character's primary color. Then, we can give some clues to one specific player by adding the clue on only one of the two maps. We can also base puzzles on this principle. For instance, we can have a path that is only open to one player whereas the other player has a wall, and this might be an indication to show that this path is particular, because only one of the two players will be able to see it. Thus, if both players communicate well, they will spot the difference and know that there is something special to do with the path.

### 2.3.2 Collectable

Throughout the game, many collectables will be hidden in each room (it is expected to have five of them), if the player places himself in a specific position. It takes the form of a perfect alignment between colors and objects of the room, which will result in a particular image appearing. If both players "collect" and find every single one of these specific positions, this will result in a "Happy Ending" at the very end of the game. Figure 19 shows the first concept of the collectables implemented, namely a "butterfly", appearing when the room is seen from a specific position.

## 2.4 Angela

**Denial** The first room represents the first chapter as well as the first emotion of grief: denial.

It is mainly composed of two things: a complex maze with tons of choices that will frustrate the players and the collectable: the “butterfly”. One wrong decision and they will go back to the beginning.

First of all, the two players do not have the same maze, but they have to take the same route to reach the exit. So for example, the white player at some point will have more choices than the other player or no choices at all that lead to the beginning and vice versa, but the correct route will be all the common choices between the two mazes.

### 2.4.1 The butterfly

This part of the room is composed of two floors and 8 portals. Each floor contains a right and left side. On the second floor, at the end of the tunnel of the right and left side there is respectively a right and left turn.

Mainly, the player starts from a point in the butterfly and the main goal is to reach the tunnels and finally the exit. The entry of the tunnels is on the right part of the second floor. But while the player is trying to reach this entry he will realise that he is not where he thinks he is.

The player starts off on the second floor of the butterfly where the entry of the tunnels is. In addition, everytime one of the players misses the correct choice he will return to the first floor of his own butterfly and he will retry to reach the entry of the tunnels on the second floor to finally exit them.

When reaching the first floor, the player has no other choice than find a way to go up to the second floor. So if he is on the right side of the first floor, the player has to pass to the left side because on the right side there is a “bad portal”. By passing from the right side to the left side of the first floor, the player will be teleported to the left side of the upper floor, but he will realise that he only reached the upper floor. Similarly, by passing from the lower left to the lower right, the player will be teleported to the upper right and not the lower right.

Moving on to the second floor. By taking the turn of the right side, the player will not stay on the right side but will be teleported to the left turn of the upper left side of the butterfly which leads to the entry of the tunnels. Similarly, if he takes the left turn of the left side he will be teleported to the right turn of the upper right side where he will reach a nice balcony to mess with him.

### 2.4.2 Itinerary of the players in the maze

The maze is composed of 3 tunnels as well as a “good” portal and “bad” ones. Each “good” choice will lead the player to a correct place and each “bad” one will teleport him to the “bad portal” in other words to the beginning of the level.

Both players start off from their beginning points: on the left turn of the upper part of the butterfly.–

The white player starts off by going directly straight, he then reaches a point where he is forced to go to the right unlike the other player who can either go to the right, to the left or straight forward. The correct choice is to take the common way which is to go to the right and any other choice will take him to the bad portal which leads to the beginning.

After that, the black player has only one choice, the correct one, to go straight, while the other has all the choices. When going straight, they will get teleported to another tunnel where the white

player has a dead-end and the other has all the three choices.

So by communicating the choices they have, they will realize that their only choice is to do a half-turn which will teleport them to a third tunnel, where the black player has the choice to go forward and the white player can go either right or left but neither of those three choices is correct.

The final and good thing to do is doing another half-turn to get teleported to the final destination: the good portal also known as the exit.

### 3 Conslusion

#### 3.1 What was achieved

Briefly, many achievements were made. Firstly, the multi was finalized. The rooms were limited to 2 players, additionally, the white and black colors were respectively assigned to the host and the client. So basically the multi will not be changed anymore. Secondly, a settings menu was added: to be able to quit the game in game for instance. In addition, the portals now work in multiplayer. The white and black players have respectively their own portals, each one can only pass through the portals assigned to his color. Thirdly, a first level representing the first emotion of grief was totally implemented. In addition, animations were done and a cut-scene was created on blender.

#### 3.2 What has to be achieved

For the final defense, a final version of the website is expected to be done. Moreover, finalizing the puzzles as well as the level designs and the creation of the cut-scenes.

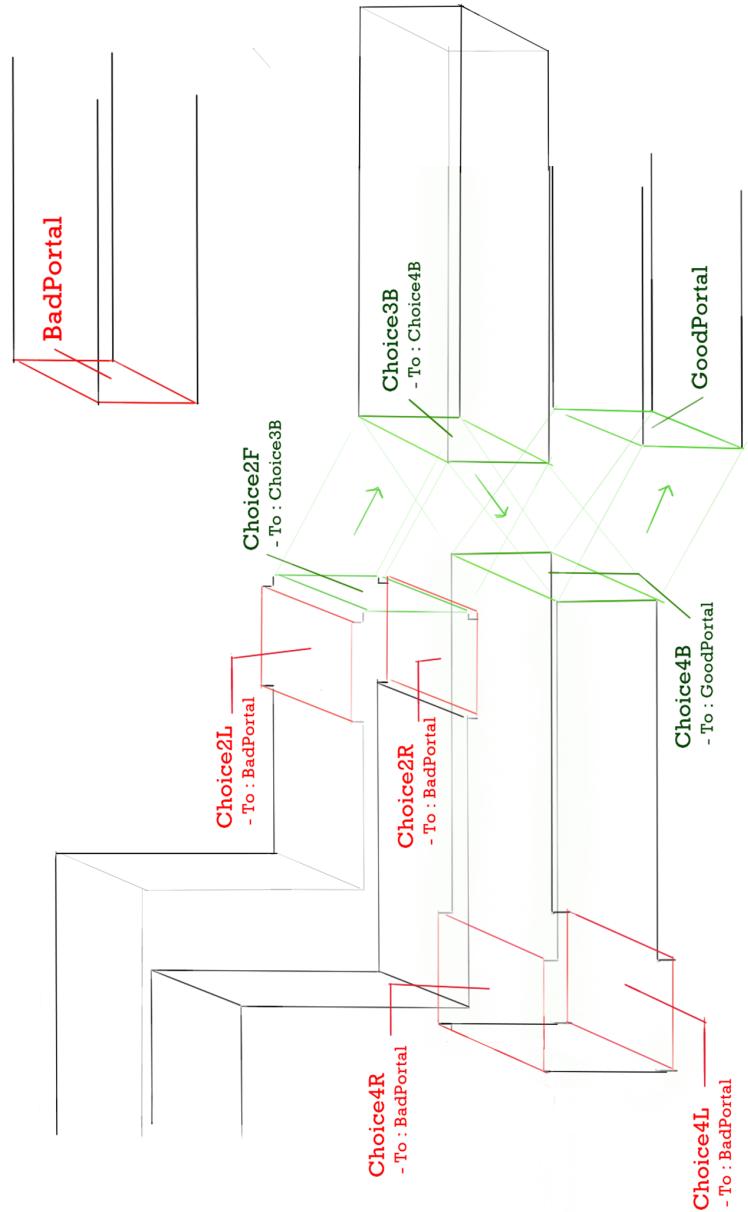


Fig. 20: White player's point of view of the room

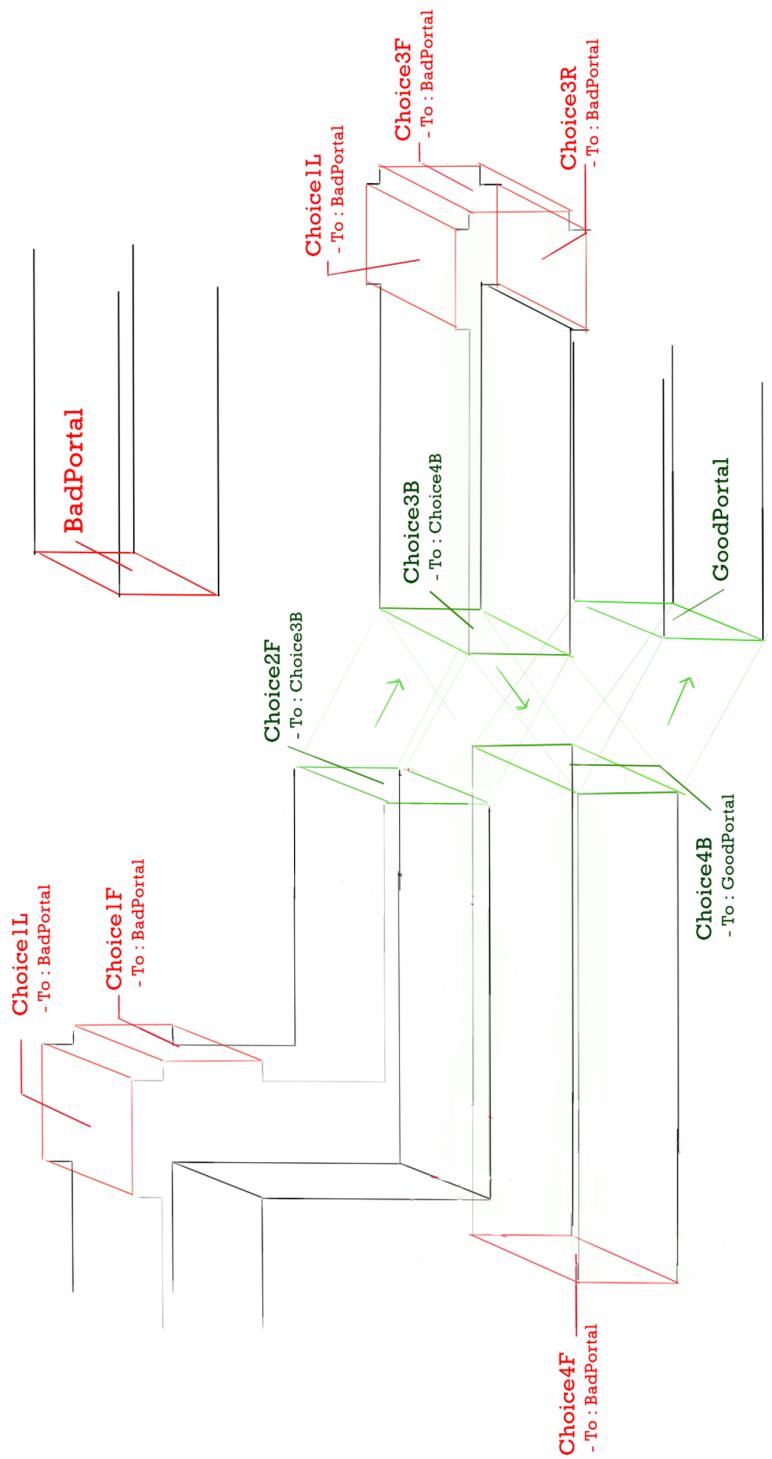


Fig. 21: Black player's point of view of the room