

HEAD4CHE PRODUCTION COMPANY

S2 PROJECT

## 1<sup>st</sup> defense report



A1

Promo EPITA 2029

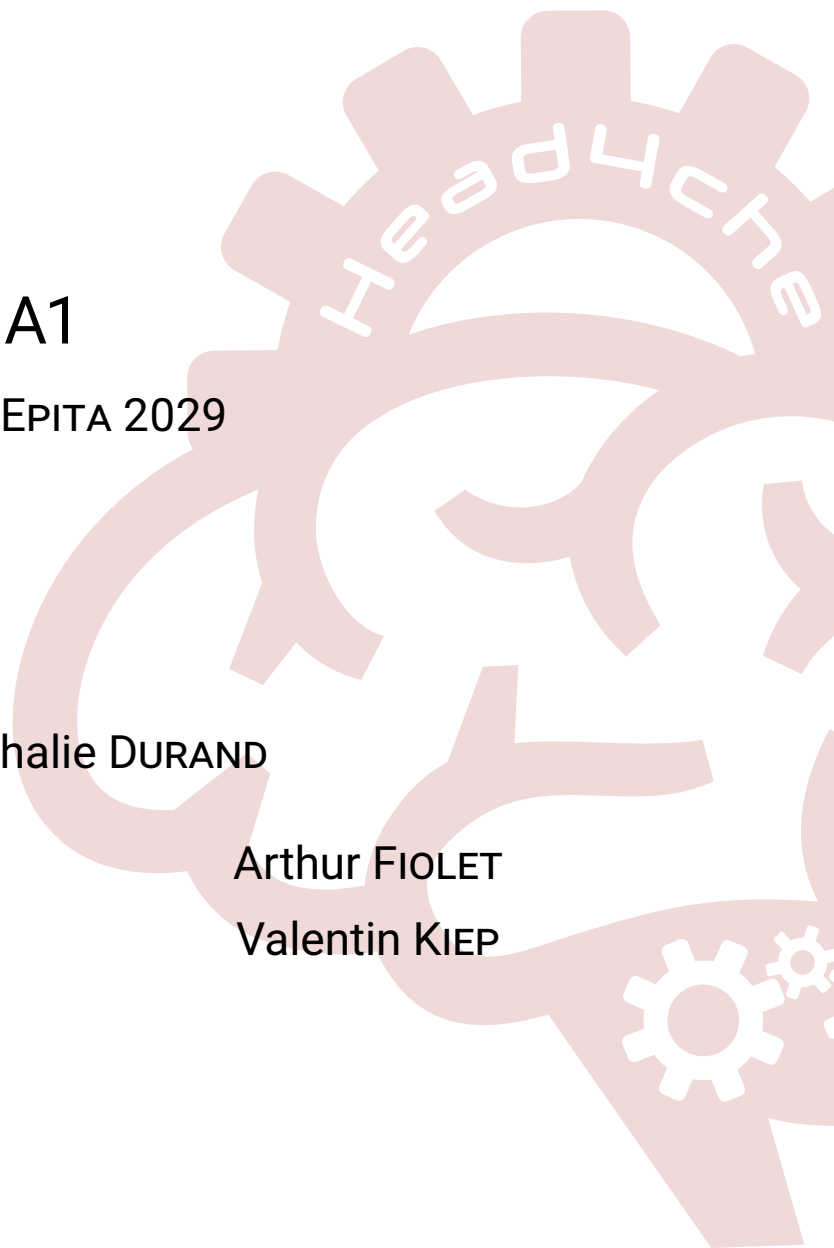
Jeanne Thalie DURAND

Tidiane BATHILY

Adam GRAZIANI

Arthur FIOLET

Valentin KIEP





# Contents

<b>I. Introduction</b>	<b>2</b>
<b>II. Project Advancement</b>	<b>3</b>
1. Task distribution . . . . .	3
2. Tools used . . . . .	4
3. Website . . . . .	5
4. Game engine . . . . .	6
a. Player object . . . . .	6
b. Menus . . . . .	7
c. Crosshair . . . . .	8
d. Highlighting . . . . .	8
e. Game Translation . . . . .	9
5. Multiplayer . . . . .	10
6. Puzzles . . . . .	11
7. 3D Modeling . . . . .	12
a. The Subway . . . . .	12
b. Animations . . . . .	14
c. The Character . . . . .	15
8. Scenario . . . . .	15
<b>III. Previsions and Planning</b>	<b>17</b>
1. Summary . . . . .	17
2. Differences . . . . .	18
a. Level Design and Gameplay . . . . .	18
b. Game Engine . . . . .	18
c. Multiplayer and Networking . . . . .	18
d. 3D Modeling . . . . .	18
e. Visuals . . . . .	19
<b>IV. Conclusion</b>	<b>20</b>



# I. Introduction

Four our project, we established the **Head4che Production Company**, a studio focused on delivering exiting gaming experiences.

Our first game, called **SubWay Out**, is an escape game designed to challenge players' problem-solving skills. In SubWay Out, the player finds themselves trapped alone in an subway train, with no clear explanation of what happened. Their mission is to conduct their investigation, and solve a series of intricate puzzles in order to find a way to leave, uncovering hidden clues along the way.

This document is a summary of all the advancements done during the first two months of work, comparing our previsions from the technical book of specifications to what actually happened.



## II. Project Advancement

### 1. Task distribution

TASK DISTRIBUTION						ADVANCEMENT		
	Jeanne	Tidiane	Arthur	Adam	Valentin	B2	B3	B4
<b>DESIGN</b>								
Scenario	M			A		80%	95%	100%
Level design	A			M		65%	95%	100%
<b>PROGRAMMING</b>								
Gameplay		A		M		5%	60%	100%
Game engine		M	A			50%	85%	100%
Website			M		A	15%	50%	100%
<b>GRAPHICS &amp; SOUNDS</b>								
3D Modeling	M		A			40%	80%	100%
Visuals		A	M			10%	40%	100%
Sound		M			A	0%	30%	100%
<b>COMMUNICATION</b>								
Branding	A				M	80%	90%	100%
Trailer				A	M	0%	10%	100%

Table II..1: Task distribution table

**M** : Manager / **A** : Assistant

**B2** : 13/01/2025

**B3** : 10/03/2025

**B4** : 26/05/2025



For this defense, we chose to focus on all the game essentials.

For instance, most of the work was done on implementing a functional game engine. For this, Tidiane started by implementing the player's base functionalities. Valentin also worked on this, especially in order to handle slopes, as we encountered the problem of our player slowly flying down instead of just walking on the ground in this particular case. Jeanne created the 3D assets of the game from scratch, and implemented them in Unity. Adam implemented the first puzzle, and Arthur created the home menus and the website. Valentin also created the crosshair to enable interaction with items in the game, and Tidiane implemented a system to highlight the objects the player can interact with. Adam worked on a multiplayer system, and even though the networking part still requires work, it is already possible to play with two instances of the game on the same computer.

## 2. Tools used

In order to work as efficiently as possible, we selected the tools to use in this project:

- Unity This is the Game Engine we are working with, the entirety of the project is done with it. Therefore, we are coding the scripts in C#.
- Github Used for version control and collaborating on the codebase, the website and task repartition. We also use some of its other functionalities:
  - GitHub Pages As a way to host our website for free.
  - GitHub Actions To build a static version of our website each time it is updated
- Blender Used for the 3D modelizations of the assets.

### 3. Website

In order to promote our project and to make it accessible to anyone, we wanted to create a website. We first had to choose three things: how to host it, which tasks we should get and how we were going to create it. The first thing we found was the host. As we use GitHub for source control anyway and considering the fact that it offers the possibility to host static websites for free (coded in HTML, CSS and client-side JavaScript), it was obvious for us to choose this service, GitHub Pages, as a web host.

After this, we thought about which language to use to code the website. We had the classic option of directly writing HTML and CSS, but it didn't seem efficient enough. We considered the possibility of using server-side languages such as PHP, but our website isn't going to require any server-side processing and such languages wouldn't be handled by GitHub Pages, so we quickly forgot this idea. Instead, we made the choice to use a JavaScript framework to develop the website, that we would then compile to a static website. Our choice has therefore been to use React JS.

At first, we were compiling and publishing the website manually, but automating this through a workflow quickly became a priority because it was a repetitive task that was considerably slowing down the coding process. In order to do this, we used another function of GitHub called GitHub Actions, that allowed us to automatically build and publish the website when we push it to our repository.

Concerning the domain, we could have used the free one from GitHub Pages, but we wanted our own to have a shorter, better and simpler URL so we bought it and we set everything up so that it mirrors the GitHub website

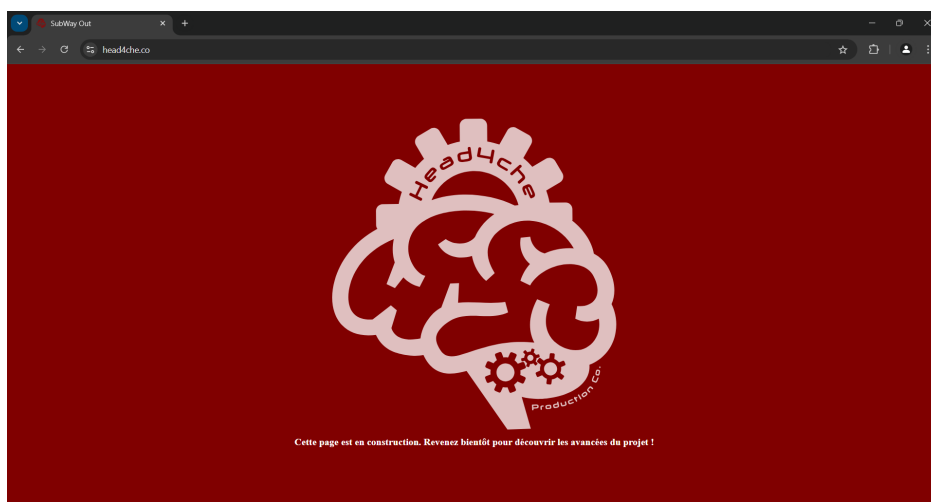


Figure II..1: Developpement homepage of the website.



## 4. Game engine

### a. Player object

One of the initial tasks we needed to accomplish was to create a basic player character, one that could at least move within the game environment. Since our game is played from a first-person perspective, there was no immediate need for a detailed 3D model of the player character, as the primary focus was on functionality rather than visual representation. We faced an issue about gravity and the player falling too slowly. Until after a lot of work it has eventually been fixed.

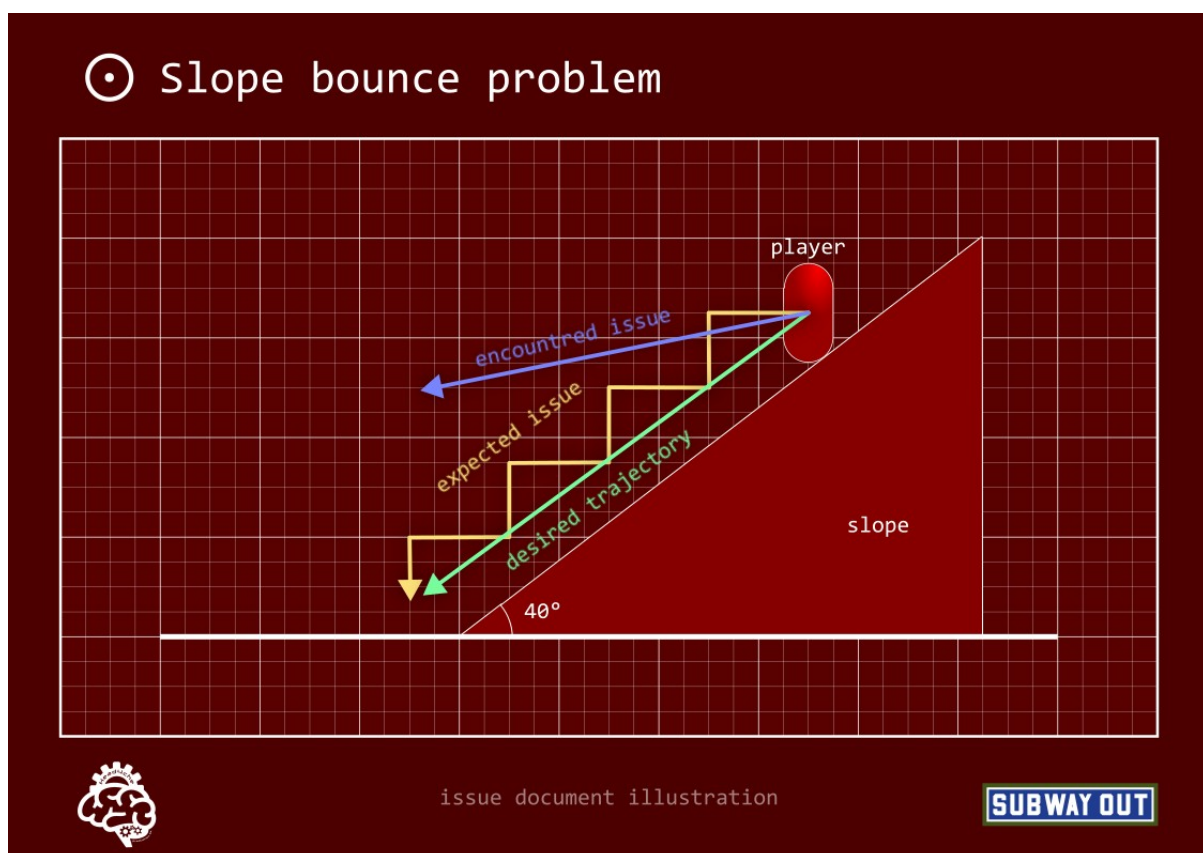


Figure II..2: Shema of the issue encountered



## b. Menus

Creating the home, options, and multiplayer menu screens was an important early step in the development of our game, as these screens serve as the main hubs for player interaction and navigation. The home screen provides the player with a starting point, offering options such as starting the game or accessing other features. The options menu will allow players to adjust important settings, like sound or language, ensuring they can customize their experience.

For now, the priority was to have something that worked, so only the home menu is functional, we still have to implement the logic of the option menu and of the multiplayer one. Also, there is still work to do on the user interface design. We are going to reorganize the buttons, and to put a background to insist on the atmosphere and visual identity of SubWay Out.

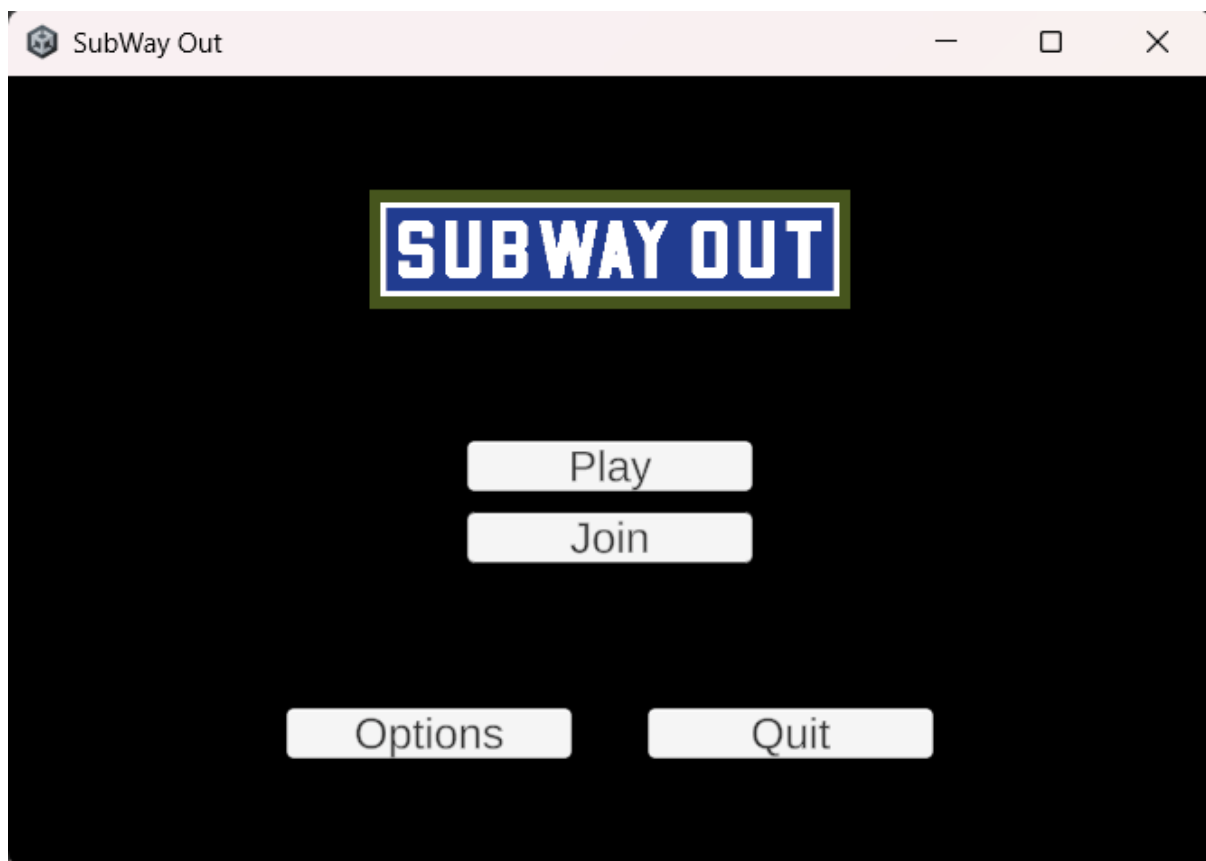


Figure II..3: Temporary home menu

### c. Crosshair

We had to design the way players were going to interact with their environnement and objects around them. In this perspective, we got to the conclusion that because we wanted the game to look as close as possible to a real-life experience, using a first-person view was the best possiblity, and to make interactions easier, we chose to add a crosshair in the middle of the screen.

Doing this, we had to study the matter of how to get the object at the foreground from the player point of view.

We successfully imlemented it, and today we are able to interact with objects.

Another important step related to this game mechanic is the implementation of the ability to pick up objects. During this part of the work, the main issue we faced is the behaviour when having objects in hand next to walls or other decor elements. Sometimes, the object that the player would go through that decor element, because collisions were not handled for the object.

After some research, we found a way to fix this and to have a more realistic user experience.

### d. Highlighting

An issue we really wanted to avoid concernig the user experience is the problem of movable objects, that you might accidentally drop somewhere and end up looking for instead of focusing on the game.

This issue would generate a lot of useless frustration for players, without even giving them the thrill of searching or the happiness of finding what they were searching for.

Because of this, we chose to implement the possibility to highlight already found interactible objects. This functionality will ensure a better experience when playing the game.



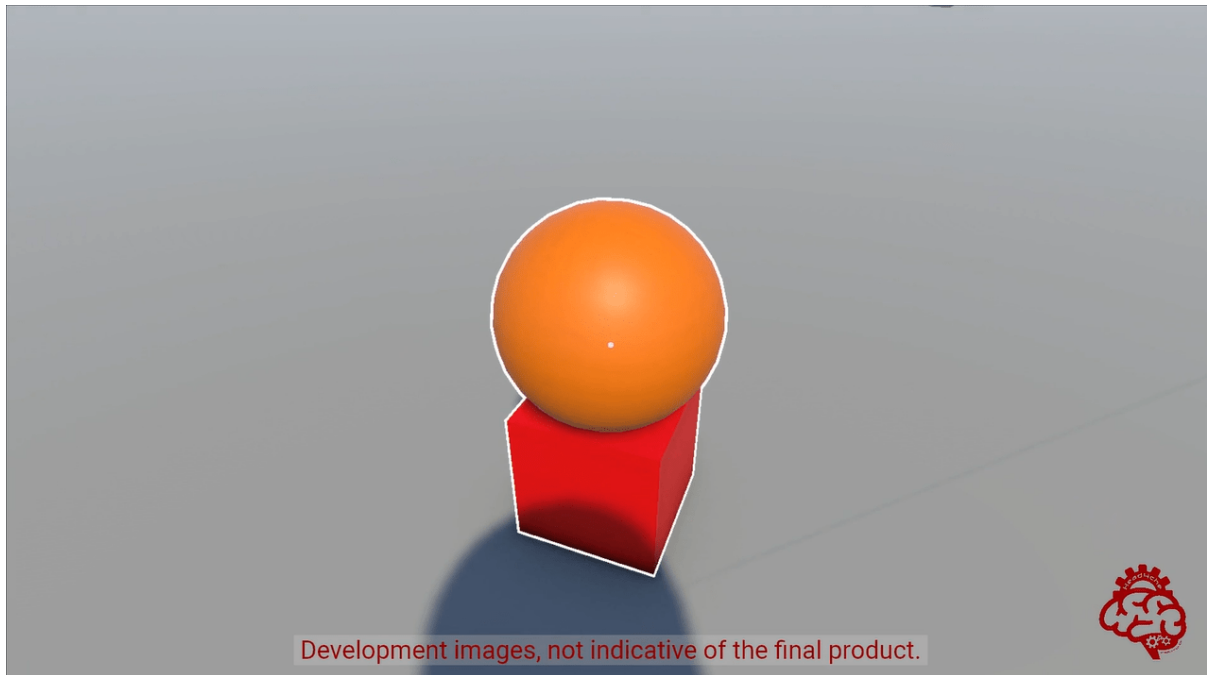


Figure II.4: Shema of the issue encountered

### e. Game Translation

It is important for us to make our game accessible to a public as wide as possible. That is why we originally picked English as the game language.

However, after some reflexion, a better option appeared to us: instead of limiting the game access to a unique language, why wouldn't we implement a translation system in order for everyone to be able to understand and play to SubWay Out?

As there was no reason not to do it, we made this idea a reality, and now we have an easy way to add supported languages to the game. Moreover, the software automatically detects where the player is from in order to select the more appropriate language.

## 5. Multiplayer

Concerning the multiplayer aspect of the game, the first choice we had to do was to choose a method to follow. Our choice fell on Netcode Unity. As of today, players are synchronized, and interactible objects will be synchronized in the same way.

The only issue left is the connection between two clients. Indeed, the multiplayer can only work if two instances of the game can communicate. Currently, this is the main factor preventing us from having a fully functional multiplayer. Indeed, we are already able to play a multiplayer game on two instances of the software in a single computer, but it doesn't work once we try connecting to another one.

The issue here is getting the IP address of the computer you want to connect. If we successfully get it, we will be able to play in multiplayer mode.

To solve this issue, we are working on a custom connection system. A player (called the client) will get the IP of another client (called the host) through a "station code" given in the multiplayer screen. This code is basically a representation of the IP address of the host, so the client will be able to send a request to enter a multiplayer game.

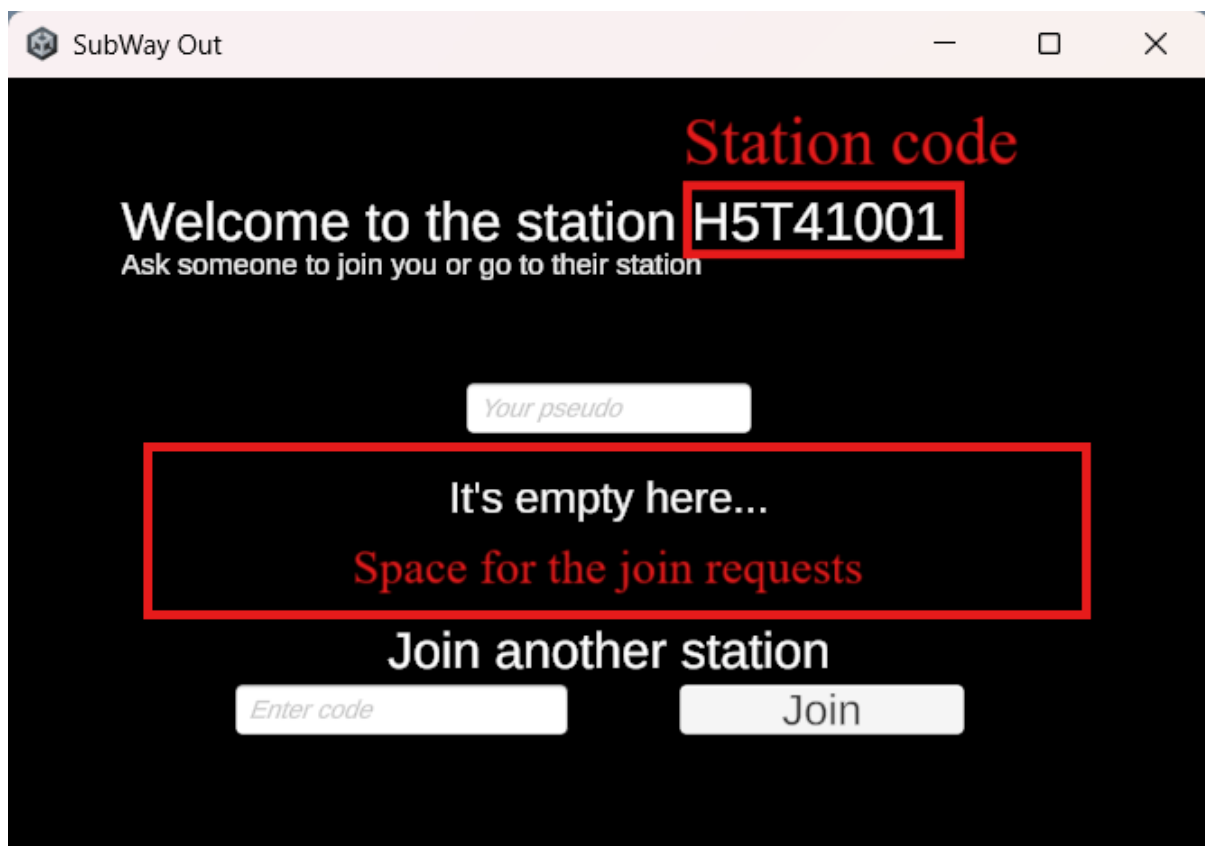


Figure II..5: Multiplayer screen

## 6. Puzzles

As a proof of concept, we implemented a puzzle: a version of the "Hanoi towers" game. The rules of this game are basically the same as the ones of the mathematical game.

There are three stacks and three spheres, the goal is to move the spheres from the first stack to the last one following specific rules.

- Only one sphere can be moved at a time: The player can pick up and place only one sphere during each turn.
- A sphere can only be moved to the top of another stack or an empty stack: When moving a sphere, it must be placed at the topmost position of the stack it is being moved to.
- A larger sphere cannot be placed on top of a lighter sphere: To maintain the challenge of the puzzle, the stacking order must always respect this rule.

When this puzzle is solved, the doors open and the player is finally free, as it will be the final challenge for the players, concluding their adventure in the subway.

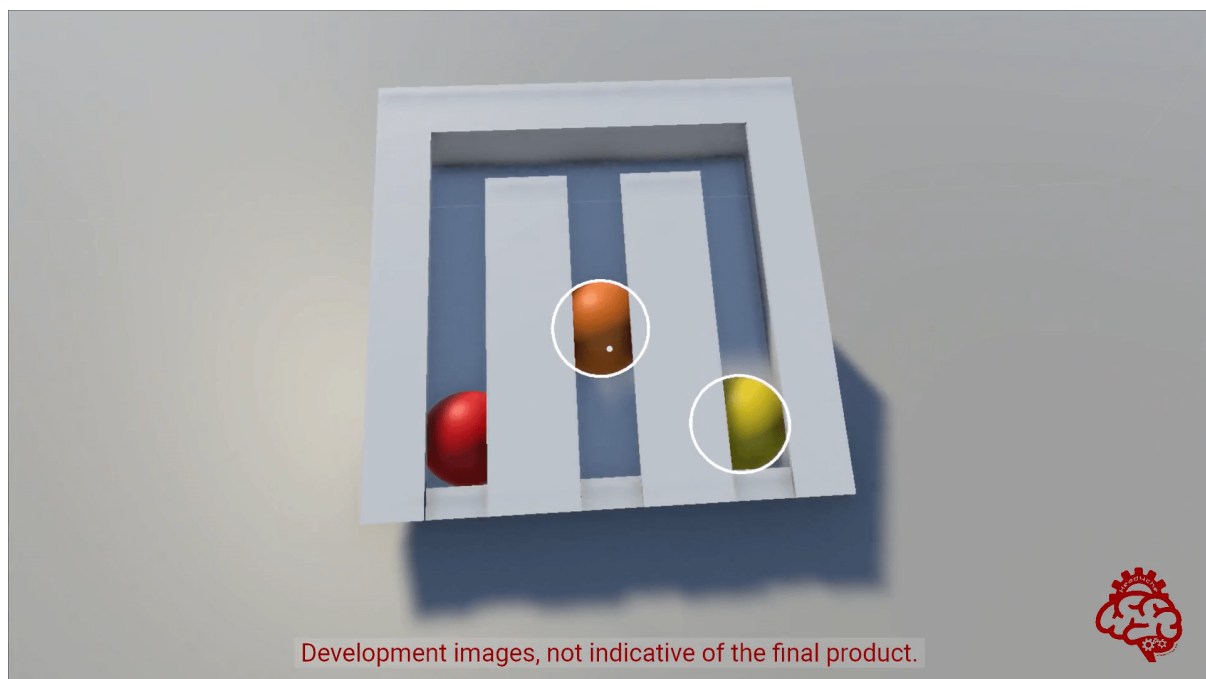


Figure II..6: Tower of Hanoi puzzle

## 7. 3D Modeling

Concerning the realization of the 3D models, we have decided to use Blender, an open-source 3D creation software. We all agreed that creating our own 3D models from scratch would give our project a more personal and distinctive look. Therefore, we took the decision to spend more time designing our three-dimensional objects. At this stage, the main element of the game, which is of course the subway, is completely finished and we are currently working on the second most important part of the modeling, which is the character. We also wanted to be recognisable to our community. In other words, we chose to have a dualistic visual identity, using a contrasting realism for the different elements of the game. We kept the metro very similar to real-life, so that anyone who has ever used the subway would be able to recognise the scenery, but we opted for a humorous design for the different characters.

### a. The Subway

To begin with, the model of the subway was based on the Parisian style, and more specifically on the metro line number 7. The current model in use for this line is the MF77. Since we all use this mode of transport, it was natural for us to include this line as a reference for our game.

It facilitated and helped us to see in life size what we needed to model in Blender. To get a complete representation, we designed all the elements separately. For example, the seats were created individually from the tube to make them easier to handle for the animations. The same goes for the windows and the doors. This model is essential and crucial for our game, as it will be the place where the player spends the most time in.

During the construction of the metro, we encountered some issues. Firstly, we never had the opportunity to create 3D models before this project, so knowing how to use Blender was a difficulty for us at first. Hopefully, thanks to a lot of videos and online tutorials, we were able to understand how the modeling toolset works.

Then, we struggled to find the adequate shape for the subway. The need to build an area of considerable size to allow the players to easily navigate in the wagon was in conflict with the necessity to make a realistic subway with proper dimensions.

In the end, after modeling the subway, the 3D model was ready to be implemented into Unity, the platform we use for the creation of our game. Nevertheless, although the implementation of the wagon's skeleton was a success, the textures were a real challenge for us. In fact, they were not applied well, resulting in a total disaster in terms of visuals and colours. Even



when we used a FBX file in Blender to save the shape and the textures of the subway and the elements it contained, the textures were missing when the file was transferred to Unity. After checking the different options for the transfer, everything was selected correctly, but we could not find out what the issue was. Finally, we decided to recreate the textures separately from the FBX file and after implementing them, the result was what we expected.

In conclusion, the implementation of the subway was not so easy, but by regularly practising the manipulation of 3D models on the Blender software and by searching for answers to our problems online, we successfully built our main element.

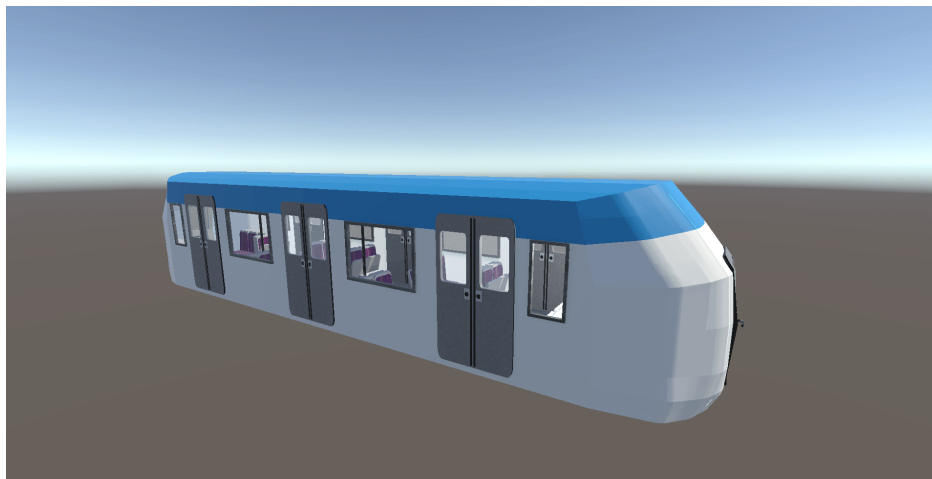


Figure II..7: Outside view of the subway



Figure II..8: Inside view of the subway





Figure II.9: Inside view of the subway



Figure II.10: Inside view of the subway

## b. Animations

Thanks to the platform Unity, we have started to make animations concerning the elements of the subway. Our main concern was the opening of the middle door at the end of the game, that is to say when the player manages to finish the final puzzle. Indeed, to announce that the game is finished and that the player can escape the wagon, our studio has decided that the middle door should automatically open itself. By dividing the middle door into two parts, we were able to modify the position of the door one frame at a time thus allowing us to obtain the natural movement of a subway door, with a seamless transition.



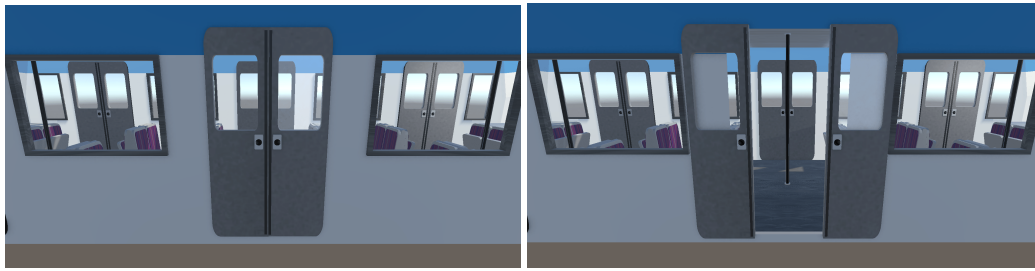


Figure II..11: Doors opened and closed

### c. The Character

In terms of the character, we decided to use a humorous style, similar to that used by Nintendo for the Miis.

The game will have 5 playable characters: each representing a member of our studio. Consequently, this will reinforce the closeness that we want to establish with our community. At the moment, the base of the model is ready, but we still need to implement the textures to make each character unique.

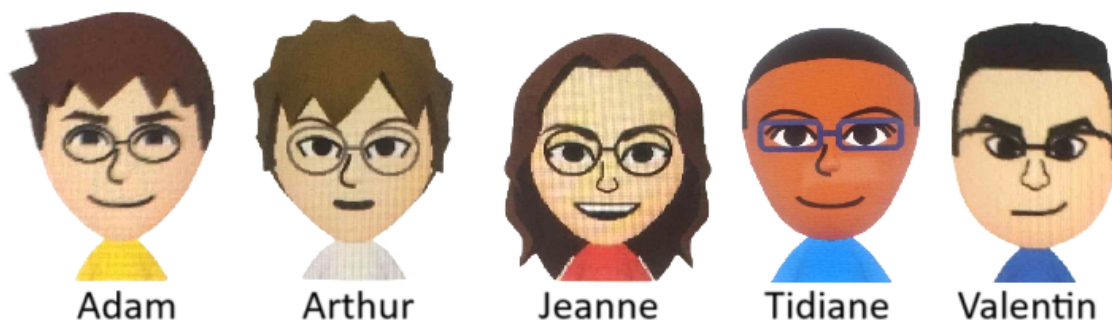


Figure II..12: Inspiration for the characters

## 8. Scenario

The player is a passenger travelling on the Parisian subway. From their noise-canceling headphones, they hear a soft and gentle music that helps them to relax after a hard and exhausting working day. Tired of this soporific and ordinary ride home, lost in the flow of passengers in the wagon and lulled by this pleasant sound, the player inadvertently falls asleep. When they wake up, there is a sense of disbelief: the player is alone in the wagon, which is still moving.

The other passengers are gone, although their luggage is still there, lying on the floor of the subway, as if they had been left behind in precipitation. The lights are flickering, a sense of anxiety is palpable, the atmosphere becomes oppressive: something has happened while the protagonist was asleep. It is up to them to find out what has happened in the subway, which has now been deserted, and they need to understand the reason why all the passengers left this insignificant place. However, their main objective remains precise: to escape from the subway.

## III. Previsions and Planning

### 1. Summary

Here is a table summarizing the previsions of the advancement in the different tasks we have to work on, comparing it to where we actually are:

	Planned advancement	Current advancement
Scenario	80%	80%
Level design	65%	50%
Gameplay	5%	5%
Game engine	50%	75%
AI	0%	0%
Multiplayer	60%	80%
Networking	50%	20%
Website	15%	15%
3D Modeling	40%	60%
Visuals	10%	0%
Sound	0%	0%
Branding	80%	90%
Trailer	0%	0%

Table III..1: Advancement comparaison

**Green** : On time or better / **Red** : late

## 2. Differences

There are obviously some differences between what we planned and what actually occurred, due to various factors and choices.

### a. Level Design and Gameplay

We chose to focus first on implementing a fully functional game engine, in order to be a lot more efficient later on level design and creation. Therefore, we worked less on this part.

### b. Game Engine

For the same reason, we prioritized this part of the project, and therefore worked more than originally planned on the game engine, as it is the foundation for the entirety of the game, and that every single aspect of the user experience relies on it.

It was therefore obvious for us to make of this part the most important thing in these first two months.

### c. Multiplayer and Networking

As explained in section II.5., the multiplayer part is functional, and even though there are some details left to fix we are more advanced than expected on this part. The reason why we wanted to work this much on multiplayer from the start was to keep up with everything and to avoid the situation where we have to change everything to adapt the game to a multiplayer mode. However, as we worked on it, we realized that this concerned only the multiplayer part, and that the networking one would not be a problem. Therefore, we lowered the priority level of the networking, in order to leave time for more important tasks.

### d. 3D Modeling

In order to assure a smooth development process in the future, we worked a lot on the assets of the game, especially on the 3D models. Thanks to this, we have gone ahead of our expectations for this subject.



## **e. Visuals**

We decided to prioritize implementing most functionalities over the design. In consequence, visuals and UI are still in rough shape.

## IV. Conclusion

In summary, we are pleased with the progress we have seen since the beginning of our project. We have made choices concerning the priority of the major features of **SubWay Out**, such as the game engine or the multiplayer functionality. We are aware that these choices have forced us to spend more time on some tasks and thus slow down the progression of others, such as level design, visuals or networking. However, we are confident that we will be able to maintain a good pace of development until the end of our project and thus compensate the small delay we have accumulated for some functionalities.