Yassine Mankai
Lauriane Bertrand

Report Project Fundamentals of Extended Reality

git repository: https://github.com/damir401/mazeVR

1) Introduction

We approached this project with two main objectives: discovering the process of designing a virtual reality experience and learning the basic tools of Unity. We had at our disposal a Pico G2 4K VR headset with a remote controller and both had 3 degrees of freedom (the 3 axes of rotation). We were asked to start by fixing a problem of our choice and try to solve it by proposing a VR application. For that purpose, we discussed in our group the different advantages a VR app can have compared to traditional solutions and we concluded that a sense of presence is a major game-changer when it comes to conveying an experience. From there, we worked on the meaning of presence as it's a complex concept and englobes different dimensions. Then, we decided to focus on the spatial aspect of it and look for related problems.

In the following document, we will first walk through the different choices we made, then the design process we took, and finally the details of our prototype.

2) Project Goal/Objectives

The goal of our project is to make the user discover and explore new places. Indeed, a typical day for someone is to get up, go to work/school, come back, stay at home, and sleep. One can get pretty tired and want to get out of its daily life. However, people mostly live in small apartments in big cities and with nowhere to change their mind, especially since the covid pandemic. That's why our objective is to make people do new activities and discover new places without moving. These activities should be a vehicle for exploration and succeeding in them should be easy so that they don't overshadow the main goal of this design, which is to give the user a sense of presence in a large world. To avoid constraining this "presence", the activities must not follow any linear logic and the user has to be able to perform them in any order. We want to entertain them and to make them forget about their problems and their real environment. In other words, "How can we make the user **explore** while being **entertained** at home?"

3) Design (method and approach, possible design list)

To find our design we chose to make different scenarios and find solutions to our problem. You can find below a few scenarios we thought about that are compatible with our initial problem. We tried to include some "extreme user scenarios" as well, to force us to think about more original ideas. That's why scenarios 3 and 4 represent only a small number of our target users.

Yassine Mankai
Lauriane Bertrand

Scenario 1:
The user is returning home after a long day of work. He's exhausted and wants to take some time to recover. He doesn't have any series or movies to watch so instead he decides to play a game. The game should be **restful** and **not too complicated** but still **entertaining**. He should be able to take time to look around and be free to move in the virtual environment as he pleases. Still, the game must have a goal, to make it fun to play.

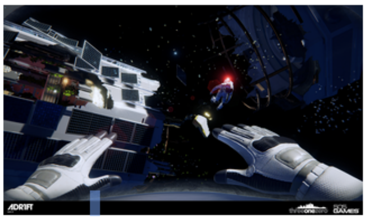
Scenario 2:
The user wants to take a walk in a peaceful place. However, it's winter so it's really cold outside and it's not pleasant to walk in the cold. That's why he chooses to use VR instead. His goal is to **walk around** and see **beautiful landscapes**.

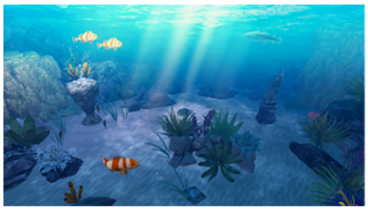
Scenario 3:
The user is a person with a disability and can't walk without assistance. Still, he wants to move and have fun. To solve his problems, he chooses to use VR. The app should be **entertaining** and fun. It should allow the user to **forget about his handicap** (here walking problems). That's why the game needs to involve movement (in the virtual environment).

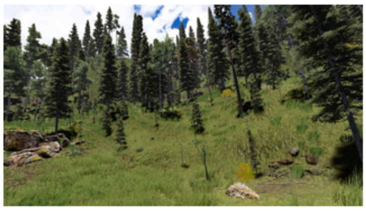
Scenario 4:
The user is old and can't do a lot of activities. People around him want to find activities he could do alone to keep him busy. The app should be **simple enough to understand** and **not demanding physically**.

Our scenarios highlighted the importance of movement in lovely and vast landscapes. That's why we focused on games with big or changing scenes. Moreover, it should be entertaining but not too complicated. We thus thought about having some kind of mini-games in our environment. We finally came up with 3 main ideas and 2 different environments.

<u>The first idea</u> consists of walking in a pretty place (mountain, desert, sea,...), either chosen randomly or by the user. To make it more entertaining and force exploration, we ask the user to clean the place, by picking up trash objects in a limited time. The goal is to clean as much as possible of the area. Below you can find some examples of what we were thinking about for the different scenes

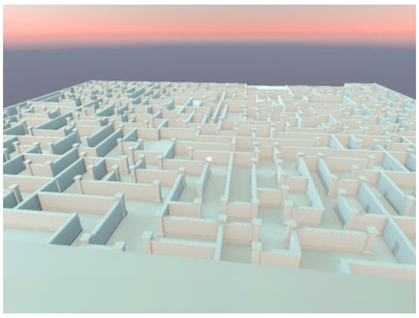






Yassine Mankai
Lauriane Bertrand

__The second idea__ is to create an escape game maze. Like a regular maze, the user must find the exit, but to make things more engaging, we ask the user to win a certain number of mini-games beforehand. At the beginning of the game, we give the user 3 images in a precise order that he must memorize, and then we ask him to find them in the maze. When he finds an image, he needs to complete a specific mini-game to obtain it. Finally, when he collected the 3 images he must find the exit and place the 3 images in the right order to completely escape and win. As for the previous idea, we focus on the exploration process (the user must find the exit as well as images and thus explore more the maze).
An important aspect that we considered was to envision these mini-games as separate worlds that expand the main universe (the maze) and add to it visually and thematically. This mini-game mechanic allows us, the designers, to create cohesive gameplay between different scenes with their unique game logic. We achieve this while still meeting our original goal, as these independent worlds are a good way to evoke a sense of vastness that effectively anchors the exploration part of the design.
A second aspect to consider is to make these mini-games adapted to 3D so that the user doesn't lose immersion in the environment. The use of depth should be essential to the mini-game which will motivate the user to move around the scene and to have a more active role in shaping the experience.



__The third idea__ is similar to the second one, as it takes place in a maze as well. This time the user only needs to find the exit but the maze is full of traps. Monsters, light changes, dead ends, the user will have to change direction regularly before finding the exit. Even the maze itself changes regularly, to increase the exploration process. At the same time, some places are accessible only through teleport portals, encouraging the user to explore different environments.

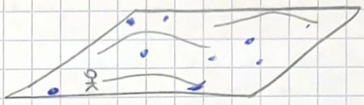
All those designs are summarized in the sheets you can find below.

Yassine Mankai
Lauriane Bertrand

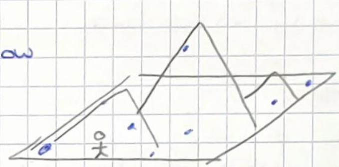
IDEA 1 : CLEAN THE PLACE

- Different scenes (sea, space, forest, ...)
- GOAL : clean the area : objects on the ground, under the ground, flying ..
- CONSTRAINT : limited time.

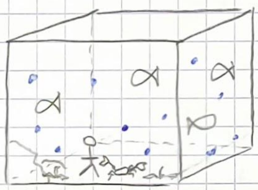
Desert : able to dig in the sand.



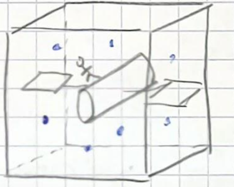
Mountain : able to dig in the snow



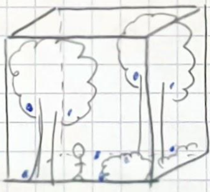
Ocean : objects floating around the user
the user can walk only on the ground of the ocean



Space : objects floating around the user
the user walks only on the vessel

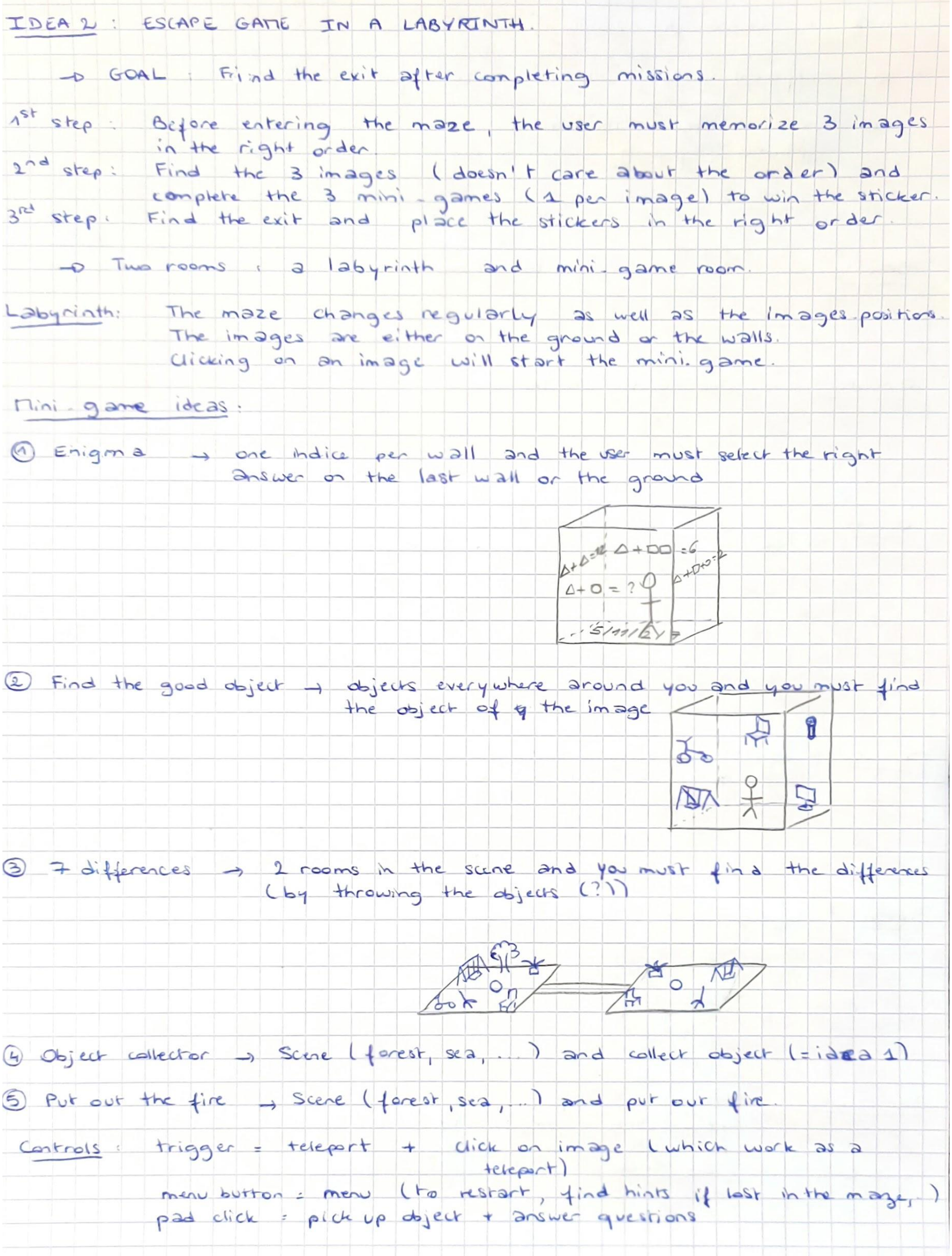


Forest : objects in trees or on the floor



The controls: trigger = teleport
menu button : menu (to select scenes, restart, see timer, ..)
pad click = pick up object
pad touch + round movement = dig .

Yassine Mankai
Lauriane Bertrand



IDEA 2 : ESCAPE GAME IN A LABYRINTH.

→ GOAL : Find the exit after completing missions.

1st step : Before entering the maze, the user must memorize 3 images in the right order.
2nd step : Find the 3 images (doesn't care about the order) and complete the 3 mini-games (1 per image) to win the sticker.
3rd step : Find the exit and place the stickers in the right order.

→ Two rooms : a labyrinth and mini-game room.

Labyrinth: The maze changes regularly as well as the images positions. The images are either on the ground or the walls. Clicking on an image will start the mini-game.

Mini-game ideas :

① Enigma → one indice per wall and the user must select the right answer on the last wall or the ground

② Find the good object → objects everywhere around you and you must find the object of q the image

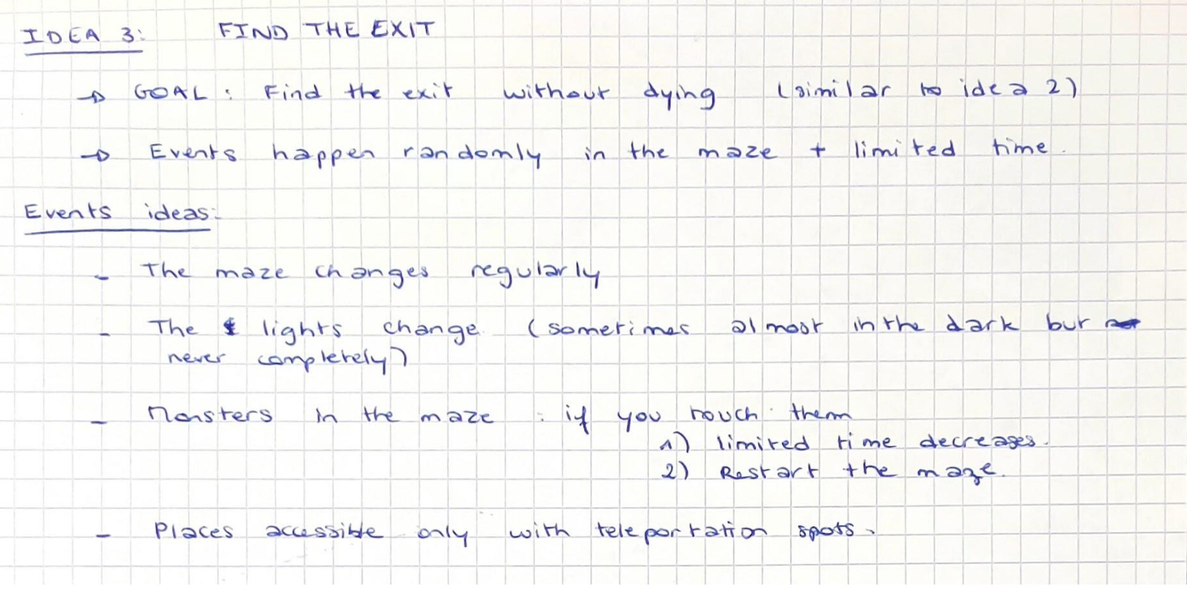
③ 7 differences → 2 rooms in the scene and you must find the differences (by throwing the objects (?))

④ Object collector → Scene (forest, sea, ...) and collect object (=idea 1)

⑤ Put out the fire → Scene (forest, sea, ...) and put our fire.

Controls : trigger = teleport + click on image (which work as a teleport)
menu button = menu (to restart, find hints if lost in the maze,)
pad click = pick up object + answer questions

Yassine Mankai
Lauriane Bertrand



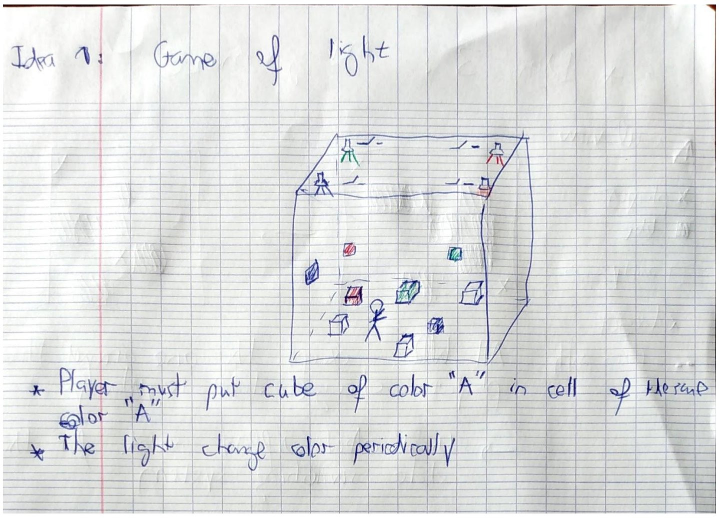
IDEA 3: FIND THE EXIT

→ GOAL: Find the exit without dying (similar to idea 2)

→ Events happen randomly in the maze + limited time.

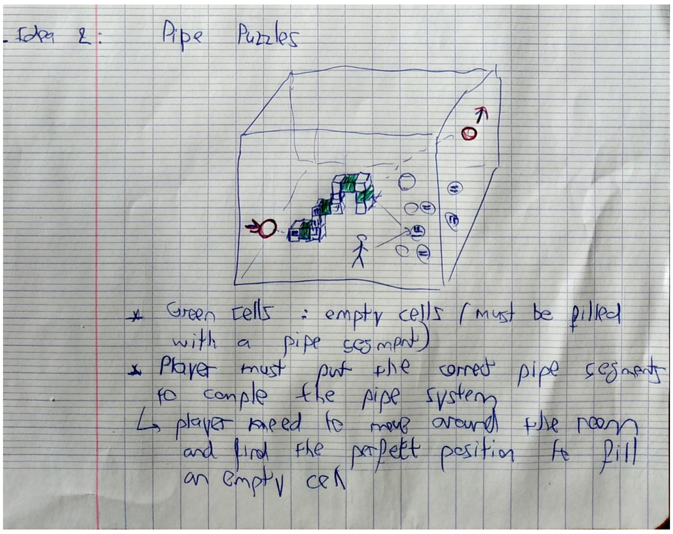
Events ideas:

- The maze changes regularly
- The lights change (sometimes almost in the dark but never completely)
- Monsters in the maze: if you touch them
  1) limited time decreases.
  2) Restart the maze.
- Places accessible only with teleportation spots.



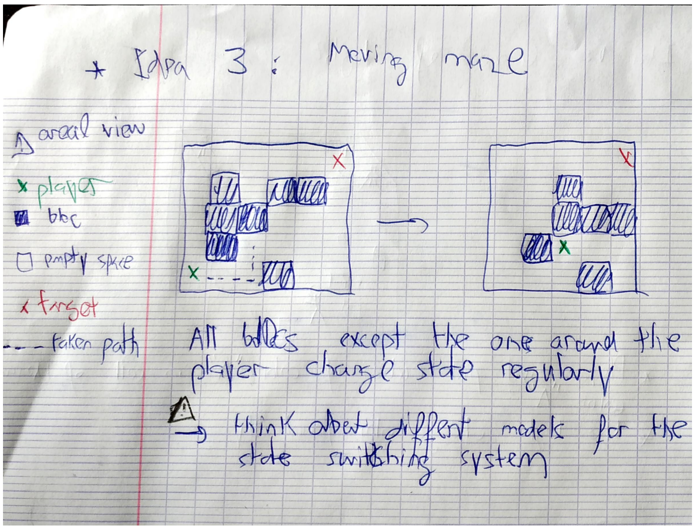
Idea 1: Game of light

* Player must put cube of color "A" in cell of the same color "A"
* The light change color periodically



Idea 2: Pipe Puzzles

* Green cells : empty cells (must be filled with a pipe segment)
* Player must put the correct pipe segments to comple the pipe system
  ↳ player need to move around the room and find the perfect position to fill an empty cell



* Idea 3 : Moving maze

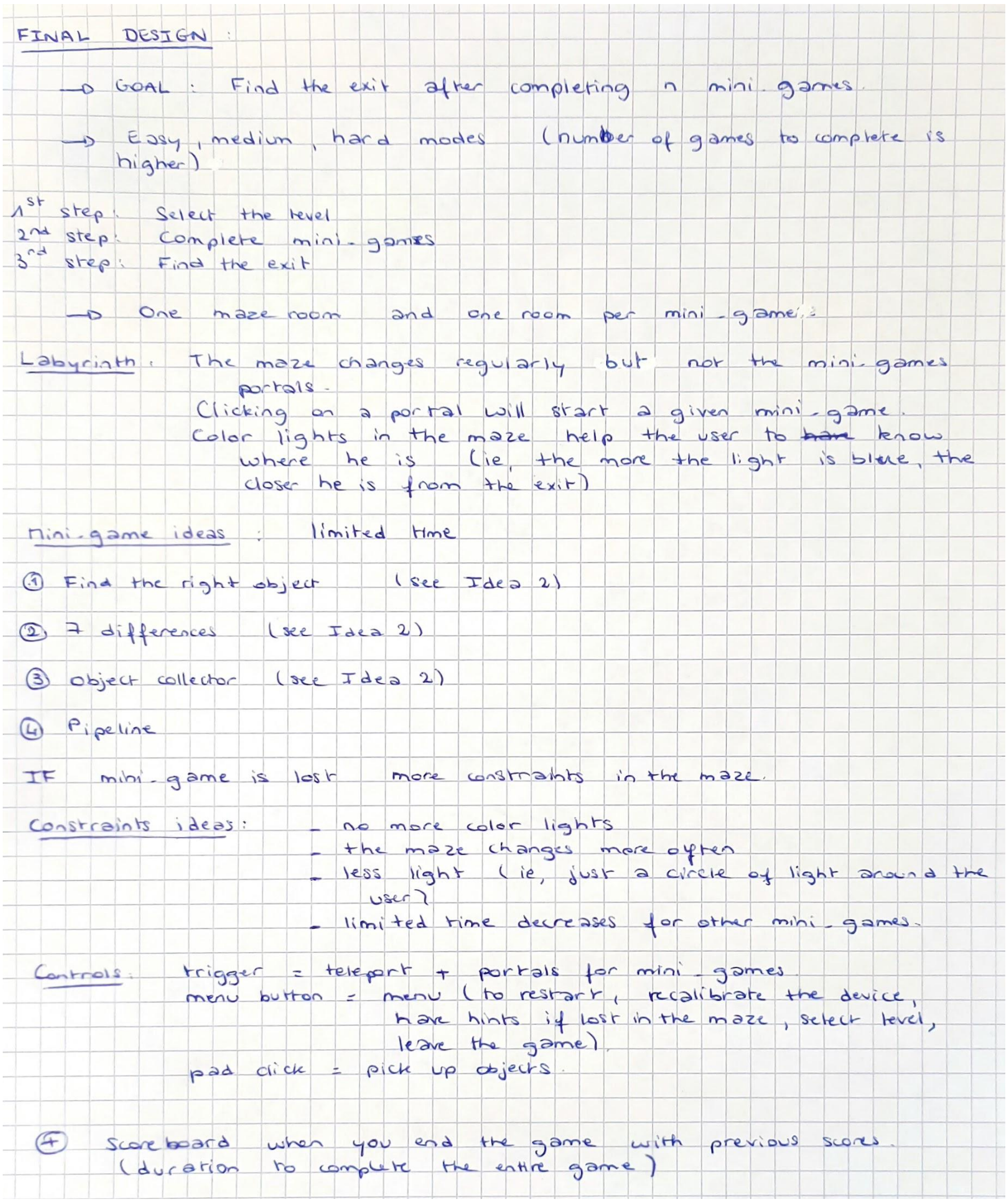
All blocks except the one around the player change state regularly

⚠ → think about different models for the state switching system

Yassine Mankai
Lauriane Bertrand

4) Prototyping

To create our final prototype, we considered all our design ideas and selected and merged the parts we found interesting while adding new ideas. We kept the maze idea, as we thought the first idea would quickly be annoying. Still, we kept it as a mini-game idea “Clean the area” which can take place in such environments. We also removed the images and replaced them with portals. The reason is that we want users to take the game easily and let them explore the maze freely. The portal mechanism keeps a coherent system of teleportation between scenes as all portals behave the same, which enforces the continuity between activities. That way, they still need to complete a precise number of mini-games but they can choose the ones they want to complete and they don’t have to remember anything. In case they lose the mini-game, they have multiple constraints added for the rest of the game such as light changes, no more color indication in the maze...



FINAL DESIGN :

→ GOAL : Find the exit after completing n mini-games.

→ Easy, medium, hard modes (number of games to complete is higher)

1st step : Select the level
2nd step : Complete mini-games
3rd step : Find the exit

→ One maze room and one room per mini-game.

Labyrinth : The maze changes regularly but not the mini-games portals.
Clicking on a portal will start a given mini-game.
Color lights in the maze help the user to ~~ham~~ know where he is (ie, the more the light is blue, the closer he is from the exit)

mini-game ideas : limited time

① Find the right object (see Idea 2)

② 7 differences (see Idea 2)

③ object collector (see Idea 2)

④ Pipeline

IF mini-game is lost more constraints in the maze.

Constraints ideas :
- no more color lights
- the maze changes more often
- less light (ie, just a circle of light around the user)
- limited time decreases for other mini-games.

Controls : trigger = teleport + portals for mini-games.
menu button = menu (to restart, recalibrate the device, have hints if lost in the maze, select level, leave the game).
pad click = pick up objects.

⊕ Scoreboard when you end the game with previous scores. (duration to complete the entire game)

For the prototype, we divided the project into several sub-projects and we tried to go as far as possible. Below you can find the different steps:

1. Create the labyrinth and move around in it
2. Create mini-games (at least 1 or 2)
3. Add areas to access the mini-games
4. Create a menu to restart/change settings/leave the game
5. Improve visuals of the different scenes and add sound effects
6. Add victory constraints and penalties for failing mini-games (time limit, traps, change of light,...)
7. Add scores and a multiplayer mode

In parallel to steps 3 to 7, we wanted to continue creating new mini-games. With the time we had, we managed to go until the 3rd step. We knew we wouldn't be able to finish every part of our prototype because it's a really big game but our goal was to at least finish the first three steps and then go as far as possible. Hence, we met our objective. It would have been great to be able to do the 4th step as it can be a problem not to be able to leave the game. That's why we added a button at the beginning of the maze, even though it shouldn't be part of the final prototype.

In terms of work-sharing and distribution, we used GitHub to have access to others' work and exchanged the device every week so that everybody could test the project. We distributed the work by scene, meaning one person was taking care of one entire scene.

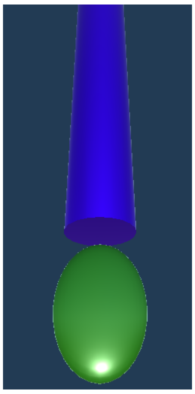
We encountered many problems with GitHub. We had some struggles at the beginning of the project but finally, we managed to make the sharing work completely.

5) Final version (app description and user guide)

As a final version of this project, we implemented a VR game. It's divided into one main scene/universe (the maze) and two mini-games/worlds (PipePuzzle and Differences). A portal system links the main scene to the others and no special order is needed to solve the mini-games. Text hints exist in each scene to indicate what should be performed. We present here the different elements used in the game:

Portal

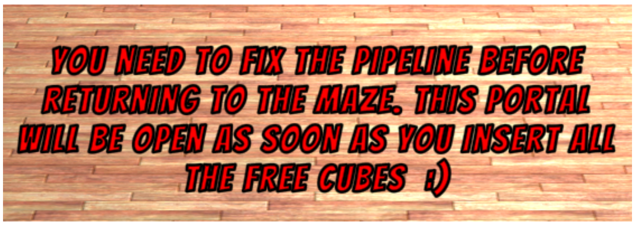
Blocked Portal



Active Portal

Yassine Mankai
Lauriane Bertrand

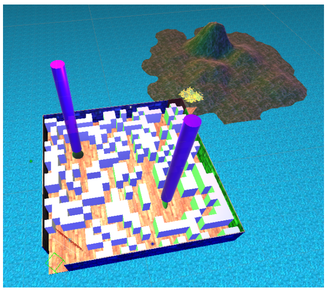
Text Hint



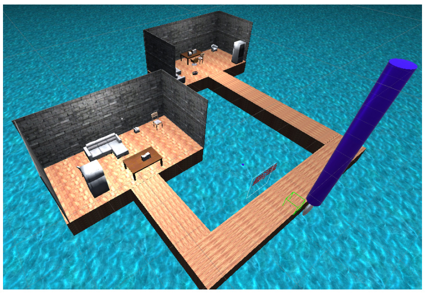
YOU NEED TO FIX THE PIPELINE BEFORE RETURNING TO THE MAZE. THIS PORTAL WILL BE OPEN AS SOON AS YOU INSERT ALL THE FREE CUBES :)

Exit Button

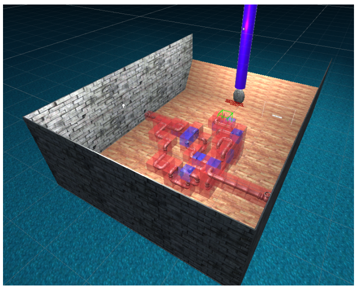
Scenes



MainScene (maze)



Differences



PipePuzzle

The game starts by placing the player in the entrance (corner) of a randomly generated maze where two portals are randomly created in two separate areas. The maze layout changes every 30 seconds. And from each side of the maze there is a light source with a specific color (red, blue, green, gray). These colors are a sort of indication to understand direction inside the maze. When the player finds a portal, (s)he can activate it to teleport to the mini-game world. There, the player finds a blocked portal that can only be activated when the puzzle is solved, only then, (s)he returns to the spot on the maze where they exited. After completing both mini-games, the maze disappears and on the opposite corner to the entrance, the player can find a door leading to a gold island (reward).

Both implemented mini-games are 3D immersive versions of the classical game puzzles. PipePuzzle extends the pipeline game genre. The player must complete the pipeline by inserting the appropriate pipe boxes. Differences is an implementation of the “find the differences” genre. The player must throw away the outliers existing in the second room.

We decided to keep a coherent control system between scenes to avoid confusing the user. Due to implementation issues, we worked on this prototype with two casted rays (curved for teleportation and portal activation and straight for object interaction).
In this game, the player can perform the following actions :

- teleport using the trigger button.
- pick/drop an object (pipe box or piece of furniture) using the primary 2D button.
- activate a portal using the trigger button.
- exit the game by clicking on the “exit game” button with the trigger

6) Conclusion & perspectives (self-criticism and planned improvements)

Issues related to the design process:

Although the mini-game mechanic was a powerful tool to expand the game universe and easily bestow a sense of vastness, it's hard to keep track of that initial design problem when creating these separate worlds and we can easily divert off the track especially in the case of this final prototype since it only has 2 mini-games, so, they can thematically eclipse the general tone of the game. This would be less of an issue if we had a large number of small and fast mini-games.

General implementation issues:

Some improvements can be done to our prototype. First, as previously mentioned, we didn't manage to implement the complete prototype (as our project was really big). Thus, the main improvement would be to finish it. Also, we focused more on the global app to have an idea of what the game would look like in terms of gameplay. That's why we didn't focus on "prettiness". It could be interesting to have some pretty textures on the maze's walls as well as more realistic portals, to increase the presence of the user. Moreover, our app faces a major problem. When changing scenes, the headset and controller don't recalibrate, meaning that if you change scenes looking up, the next scene will appear above you. The same goes for the controller, where you need to find the "origin" of the controller, place yourself in this direction and recalibrate using the home button each time you change scenes. That's why one improvement would be to correct this issue. Finally, we didn't manage to implement the menu so we used simple buttons and texts instead. We should remove them and replace them with the right design.

Issues related to the MainScene:

For this implementation, we used a simple maze generation algorithm where we flip randomly a cell (while taking into consideration its neighborhood to calculate the flip probability). This algorithm is fast but does not ensure that a path exists to each portal. The fact that the maze changes make it very likely that a path will appear in the following minute. This moving maze feature is actually an original design idea to make the exploration less about memorizing cells and more like a fun experience. That being said, we could avoid this issue by using a classic maze generation algorithm.

Issues related to the PipePuzzle mini-game:

Our final version does not take into consideration the rotation with which the box is inserted. As long as it's the correct pipe type, the insertion is performed. An improvement was envisioned but was not implemented before the deadline. The main idea is to define a metric (the norm of a difference between two quaternions) that asserts whether two rotations are close enough or not and to check this before handling the collision.Also, inserting boxes is still a tedious task. Although this drives more movement in the scene, it's a cognitively demanding task and we need to avoid any mental load in our game. This can be avoided by enabling the user to change the distance between the hand and the selected object.

Yassine Mankai
Lauriane Bertrand

Issues related to the Differences mini-game:

For the moment, we only take into consideration outliers as differences but we could expand our implementation to color or size variations, position, … Also, we only created a static scene but we could generate the scene randomly to have various arrangements.