Foti Aivaliklis

Pippin Barr

CART263 (Section A)

March 31st, 2024

Final Project – Proposal Document

After learning and playing around with the three unique libraries we have discovered over the course of the semester, for the final CART263 project, I have concluded that I want to make further developments and advancements on my Phaser 3 project, Lost in the Abyss. The following proposal document will cover the following topics in regard to the development of the final project:

1. A brief description of my already existing project and where I would like it to become.
2. The core ideas for its production and some technical challenges that may be encountered along the way.
3. A rough timeline of the project's development on a week-by-week basis until the submission deadline
4. Some visual sketches to depict what I would envision it to look like

Firstly, my Phaser 3 project titled Lost in the Abyss is a horror puzzle game where the player's goal is to escape the maze without encountering any dead ends as that will send you back to the beginning of the maze. In creating this project, I did not plan on creating a horror game, but as I was in the process of producing it, I began to fancy the idea more, so I went along with it. I already have some horror elements in Lost in the Abyss already such as the fog of war effect, the eerie ambiance noise, and the creepy title and ending art. As for how I would like to further develop this game into a fuller project, my intention is to add some context story to situate the player in the game at the beginning and end of the game, as well as in between completing each maze. In addition, I would also like to create 1-2 additional maze maps for the player to progress through with some potential special mechanics, such as special traps or jump-scares, similar to that of the first maze. The brief idea I have for the story is that the player decided to an explore a cave one day but ended up taking a wrong turn and got lost in its abyss. In between completing the mazes, there will be updated diary entries from the player on their current situation within the abyss. As they complete more mazes, their entries become more hopeful of escaping the abyss, leading up to completing the final maze and ending the game with their inevitable escape. Depending on the skills of the player, I envision the gameplay length to last between 5-10 minutes total.

To continue, as stated previously, I will be using the Phaser 3 library to further develop this project. When composing the first version of Lost in the Abyss I had used many different techniques from the Phaser 3 library to bring my ideas to fruition, to which I plan to use as well when continuing its development. For instance, when creating the maze and now the future mazes, I mainly utilized the program Tiled and the concept of tilemaps and tilesets for composing every aspect of them. To begin, Tiled is a free and open-source level editor that

allows the user to create their own levels for their games. How I used it to create my current map maze and plan to use it for the future ones goes as follows: first I import my custom-made block, or tileset, that I would like to use to create the maze itself, making sure it's the right size and shape in relation to the Tiled tilemap dimensions. Next, I would select the blocks I wanted to use to create the maze's walls and add the specific property "collides" to it and check mark it under the properties tab; this is what will make the maze's walls have collision. Once that is completed, you can make the level as normal and then export it to the assets file, so it is ready to be implemented into your code. The next step is to import your tileset, finished tilemap, and the Tiled file into your coding folder and load it in the Boot.js file, so that when you load the map in your Play.js file, it recognizes all of the necessary elements to make it appear. Lastly, to properly import the tilemap, you must load in the tileset using the code:

```
this.add.image(0, 0, "*insert tilemap key code*");

let map = this.make.tilemap({ key: "*insert tilemap key code*" });

let tileset = map.addTilesetImage("*insert tilemap key code", "*insert tileset key code*");

map.createLayer("*layer 1 (background)*", tileset);

let maze = map.createLayer("*layer 2 (maze)*", tileset);

maze.setCollisionByProperty({ collides: true });
```

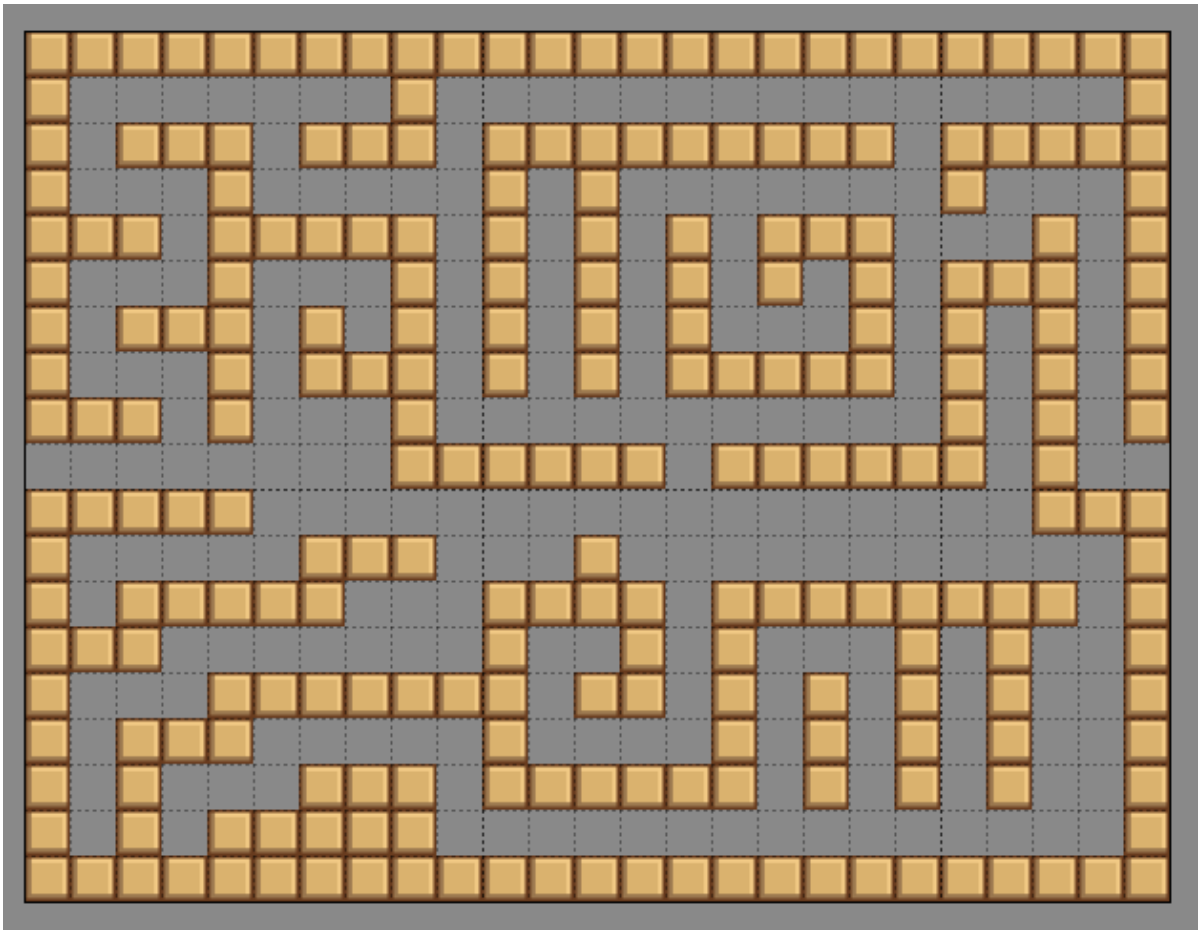
In doing so, this allows for the user to load in the tileset and tilemap effectively, while also recognizing the unique layers created in Tiled in order to differentiate what is on top versus what is on the bottom and what to add collisions to. When doing this at first, this was quite challenging to get to work, so since I am already familiar with how to create a maze the others' creations should go smoothly. Aside from this, the main core ideas that will be utilized will be the use of if else statements for the special traps and jump-scares for the upcoming mazes in regard to the player's position, using the text and bitmapText gameObjects for creating a typewriter typed effect for the story portions, and the use of multiple scenes and files working intertwined with one another. The if and else statements will be used in conjunction with the avatar's position to add special traps and effects to the maze. For example, if you stand against a certain wall, you may be able to go through it as a secret passage, or if you pass a certain threshold the controls are inverted until you reach the exit, or if you encounter the "fake exit" you get a creepy jump-scare and are sent back to the beginning of the current maze, or if you reach a dead end you have to battle an NPC in rock, paper, scissors to continue moving forward. The text and bitmapText will be primarily used to create this typed effect where the text shows up one letter at a time to add a suspense factor to what information the player knows. Lastly, working with the different files and scenes is crucial to making everything come together, as one mistake with a key code or the way a JavaScript file is set up will result in the program crashing. Overall, the key to this project's success is patience and precision. From this, the major challenge that I may require assistance with are the implementations of the special traps and effects within the maze. Each trap and effect will require its own effort in figuring out how, and if it is possible, to make it a reality, learning the code for its functionality, and understanding how it works and how to implement it within my already existing code.

Before concluding with some visuals of what I would like Lost in the Abyss to look like, I will detail a rough timeline for how I will be approaching this project.

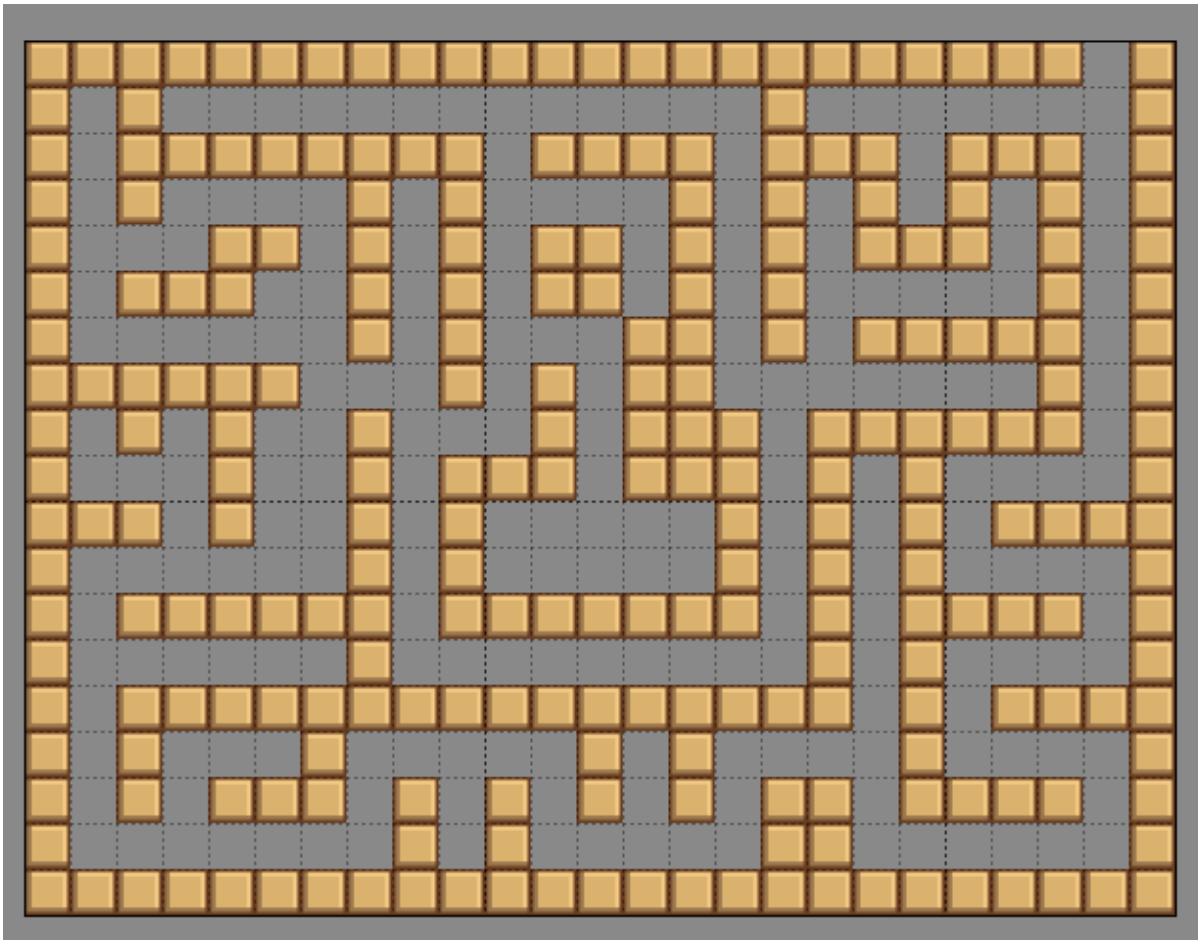
1) Week of March 25th, 2024: complete the project proposal + complete the final project prototype by implementing a new map and a bit of the story at the beginning of the game and in between completing maze 1 and starting maze 2.
2) Week of April 1st, 2024: receive feedback on the project proposal, begin working on adding the special effects to maze 2 and meet with Pippin to discuss any issues I may be encountering with the implementation of the special traps for the mazes.
3) Week of April 8th, 2024: ideally finish composing maze 2 with its traps and begin the composition of maze 3 along with implementing the new portion of the story in between maze 2's completion and the beginning of maze 3.
4) Week of April 15th, 2024: complete the composition of maze 3 with its special traps and effects, finish off the story after exiting maze 3, and playtest to ensure there are no bugs and that it is fully functional.
5) April 21st, 2024: deadline for submission

Finally, below are some sketches of what I would like the visual novel to resemble upon completion.

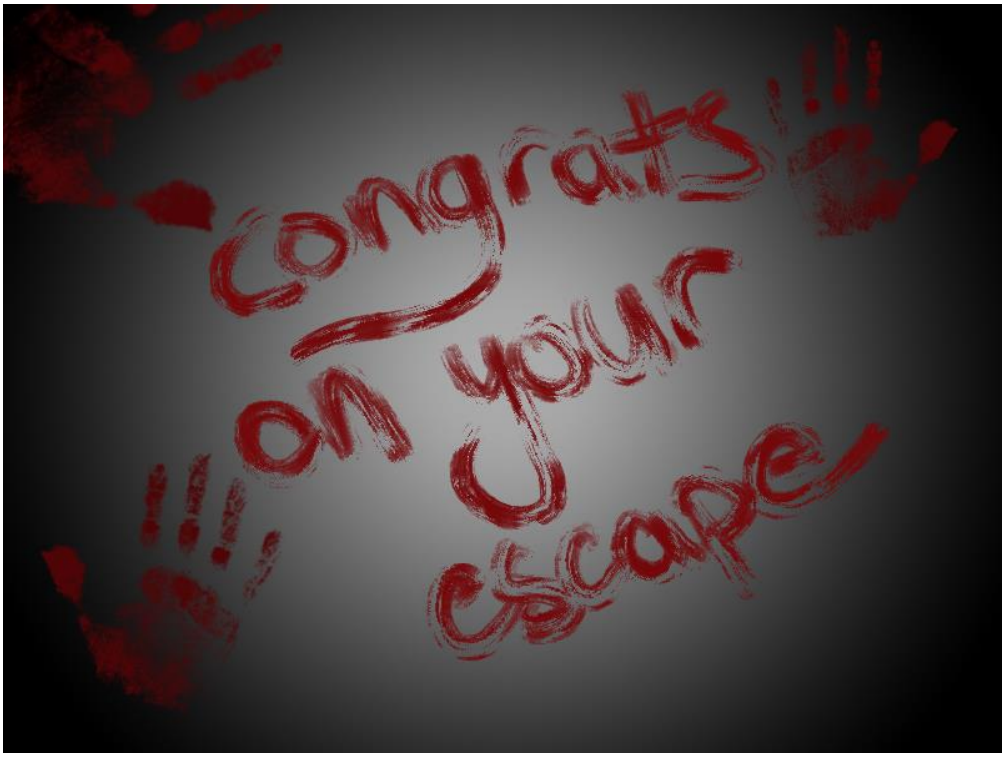
- Maze 2



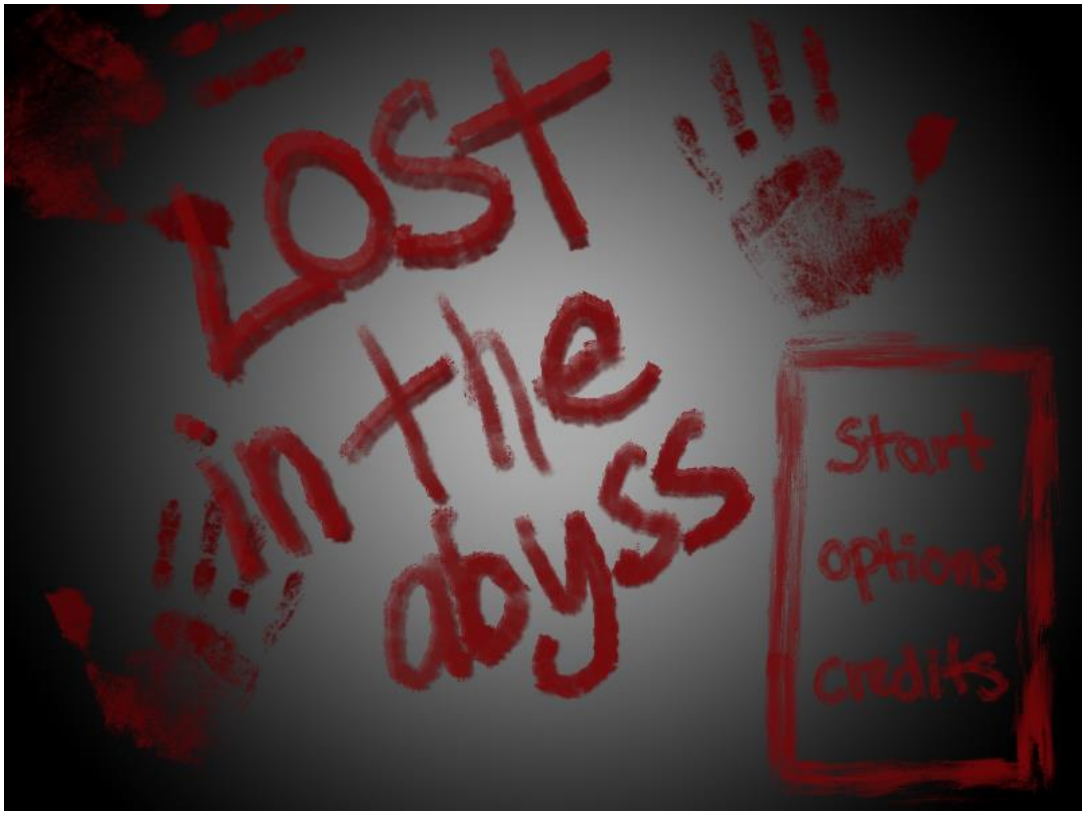
- Maze 3



- Ending screen



- Title screen



- Story text in between mazes