

# Phase IV Report

## 1. Project Details

Project: 2D Maze Game in Java - Survive in the end

Development team: Group16 (Risa Kawagoe, Rongsheng Qian, Anika Sheikh, and Sibei Zhou)

Development Tools:

- intelliJ IDE with JDK
- GitLab
- Maven
- Junit

Project Phase 4 Deadline: April 17<sup>th</sup>, 2022

## 2. 2D Maze Game in Java - “Survive in the end”

### 2.1 Overall description of the game

Our game, Survive in the End, is an arcade style pseudo 3D game. The game is based on a zombie apocalypse horror story where the survivors include good survivors, bad survivors, and the main player. While the good and bad survivors have the same visuals, the main player will lose HP when coming in contact with the bad survivors. On the contrary, the good survivor will increase the main player's HP when in contact. Throughout the maze, there are numerous vaccines scattered for the player to collect. The player's objective is to collect the required number of vaccines and find the good survivor who will open the exit door for the player to exit the maze. The required number of vaccines depends on the difficulty level chosen by the player.

### 2.2 Original plans and final product

In terms of our GUI design, we had initially created a 2D maze during phase 2. The maze looked trivial with simply 2 types of tile, walls and floor. To improve the visuals and add more depth to the maze we converted our 2D maze to a pseudo 3D maze. We added several more tiles that acted like shadows and refined edges.

When we initially started coding we were only focused on adding new classes/files without any specified folder structure. This worked out okay for us in the beginning as we did not have many files. However, as our project started to grow it became confusing for us to find files because of our clustered files. To solve this issue, we made a change to our folder structure to divide up the files into abstract, static, and dynamic classes. This led to increased productivity as files were faster to recognize and edit.

We mostly tried to stick to the UML diagram we created in Phase 1. However, since our game design changed from the initial design we had to add some classes to account for those changes. For example, we replaced the 'Wall' class to 'checkCollision' class. This is because we used a rectangle inside our characters to check for collisions instead of the entire parameter of the character. Initially we also had some significantly clustered classes such as the GameFrame class and inputKey class. This was because we used those classes to integrate our major classes with each other. To improve the readability of the code we introduced some new classes such as Command (and its subclasses (ChangeLevelCommand, EndScreenCommand, TitleScreenCommand)), GameAttributes, GameImage, GameState. These classes helped us distribute the functionalities of the game in a way that increases

cohesion and decreases coupling, and reduces the amount of things one class (mainly the GameFrame class) was trying to do.

### 2.3 Lessons learnt through this project

Throughout the project, we learnt how good quality communication affects the product of our work. Even just asking for help or initiating a casual conversation helped us a lot in progressing with our work because that was what made it easier to communicate other important information. For example, if we added a new parameter to a constructor we would notify everyone in the group so that anyone who would potentially in the future have to use the constructor would be able to appropriately instantiate the object from the class without having to wonder why they get an error.

As we did not have any specific leader to guide the team, having mutual information sharing was key in having smooth development as things became more and more complicated in the project file. We mainly had three ways of sharing information and communicating. For casual communication, we used a discord server (chat and voice call) to substitute a working environment face-to-face conversation since it was not always convenient for all of us to meet up in person. Another form of communication was a shared Google doc which contained information that had to be recorded such as scheduling and documentation. Additionally, if necessary, we would meet up in person to work on the project. Using the right tools in the right situation helped us avoid having unnecessary in-person meetings.

## 3. Tutorial (Video: [paste link here](#))

The title screen will first appear when you run the program. There are three options 1) start the game 2) change the game level 3) exit and close the game application. Use the up/down arrow keys to move the cursor(the red hand) and the enter key to select the option.

### 3.1 How to select your preferred game level

Select the option “CHANGE LEVEL” from the title screen in order to select your preferred game level. There are three levels 1) Easy 2) Intermediate 3) Challenge. The game level will always be set to the default level “Easy” when you first open the application or when you return to the title screen after playing the game.

### 3.2 Playing the game

To play the maze game, select “START GAME”. There will be a short tutorial guide and the background story before you enter the maze.



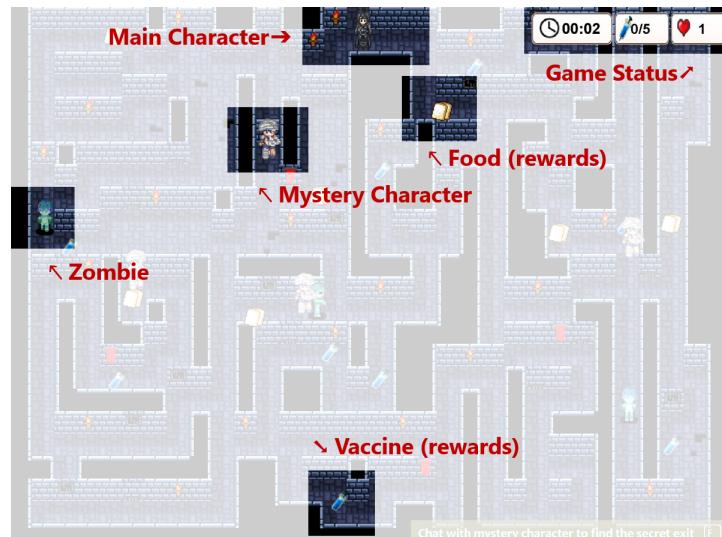
### 3.2.1 Story behind the game

The main character is a student in Simwards Wizarding School whose mission is to escape the school to call for help from the outside world. Although all the paths that connect the school and the outside world are blocked during the semester, there is a secret exit hidden in the dungeons of the campus which connects to the outside world. This maze game features the process of the main character to find the way out of the dungeon.

### 3.2.2 User's goal and how to achieve the goal

When the game starts, the main character will appear at the starting point of the maze. Use the arrow keys (up/down/left/right) to control the main character. The status of the game (the timer, the number of vaccines collected, and the HP level) will be displayed at the top-right corner. The game starts with zero vaccines and HP level of 1. The secret exit is hidden at the beginning. The user's goal is to find the secret exit and escape the campus.

In order to find the hidden exit, talk to a mystery character. To talk to the mystery character, approach the mystery character and press F key. Mystery characters can be either a "Good Survivor" or a "Bad Survivor". The bad survivor reduces your HP and the good survivor will reveal the secret exit for you. However, there are two things to note: 1) the good survival cannot reveal the secret exit unless you have collected the required number of vaccines. 2) You cannot know whether the mystery character is a good/bad survivor.



### 3.2.3 Winning/losing criteria

The main character will instantly die when it encounters a zombie. Zombies can come after you if you are close by. Keep sufficient distance between them to stay safe. Additionally, when the HP hits zero, the user will lose. Collect rewards(Food and Vaccines) that are spread across the maze. Food items will increase your HP and Vaccines are required in order to talk to the mystery character.