One Page Statement Plan of the "Emoji Maze" CMPT276S21_Group22

As noted from the title, "Emoji Maze" will be an emoji-themed 2D game. The maze's map is based on a dungeon's layout from tabletop role-playing games so it is relatively open with several dead ends. From a game design standpoint, these can allow the player to consider trying to collect treasure against the risk of moving enemies blocking their exit.

The win condition is to collect all of the "regular" rewards and to exit the maze. These rewards along with "bonus" rewards are found throughout the maze to increase the player's points once obtained. Points will be tracked by the game and if the player accumulates a negative score by picking up too many punishments, they will lose the game. As well, encountering a moving enemy will also cause the game to end in failure.

Main Character: 😀, 😃, 😄, ...

The main character of our game will be a randomly chosen face emoji from a defined large list to give diversity between playthroughs. The current position of the main character is stored and updated every time the character moves in any direction (up, down, left or right) using the arrow keys. The updated position is constantly checked and compared with the positions of the barriers (checking whether a cell is open to move into). If the cell is closed, the main character is repositioned to its current cell. The cells with the rewards or enemies are still considered open, thus each open cell is still checked for whether the player encounters these.

Enemies: 👾, 😈, 👹, 👽, …, 🧙, 🔥, 🌪, 💥, ☢, …

Enemies are separated into two types, moving and punishment, which are both implemented as subclasses of the enemy class (which itself is superclassed by the character class). The current position of all moving enemies is stored in a 2D array. While we are not fully set on this idea, we have thought about using a search algorithm to execute how the moving enemies will go towards the player. The emojis representing the movable enemies and punishments in the game will be randomly selected but to minimize confusion, only one emoji for each type per game will be used for that playthrough.

Rewards: 💰, 💎

The regular rewards for this game will be the emoji for a money bag while the bonus rewards that randomly spawn will be diamonds. Once the regular rewards are collected, the cell will change to a normal background space so that it cannot be collected again. This is similar to the bonus rewards which will randomly choose a spot to place it until either a player collects it and the points are added or enough time passes and it disappears.

Board/Map: 📦

The outside wall will encompass a map and will be static in every run of the game. The board will be 25x25 containing 500 cells in total. To organize the cells we will use a 2D array. We have decided to have only one opening of our map that will serve as both the beginning entrance and exit once the player has collected all the required regular rewards. Each of these bricks takes up one cell and sets that cell's status to closed. All entities start on set cells but only the main character and moving enemies are permitted to move into available open cells and the punishments and rewards disappear once collected. To track the location of everything on the map, every cell has at least three boolean variables noting whether it is open and whether there is either a punishment, reward, or nothing on it. Using these booleans, the system will adjust the score accordingly when a player steps into a cell.

Functionality (Interface):

Must-Have:	**May-Have:**	**Restrictions:**
- Score & Time display function. - Game result display page - Tracing characters' locations. - Allowing the user to control the movement of the main character.	- Landing welcome page. - Instruction pages. - Interface to choose the main character's appearance. - Warning for missing rewards.	- The outline and appearance of the map are fixed. - English supported only.