# Design Overview for Maze Game

Name: Tharin Sandipa
Student ID: 103166002

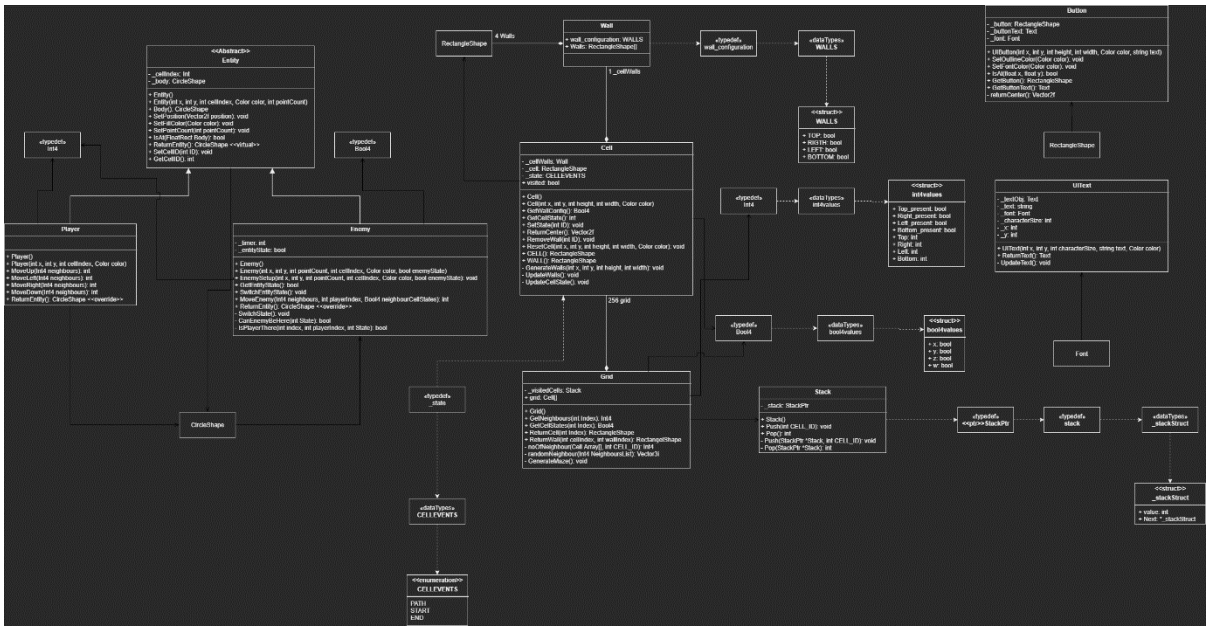
## Summary of Program

When the user first launches the application, a GUI window will open with instructions on how the game works like text giving hints on how the enemies work and how to move the player and the goal of the game after which the user will click a button that will then close the current window then launch a new window.

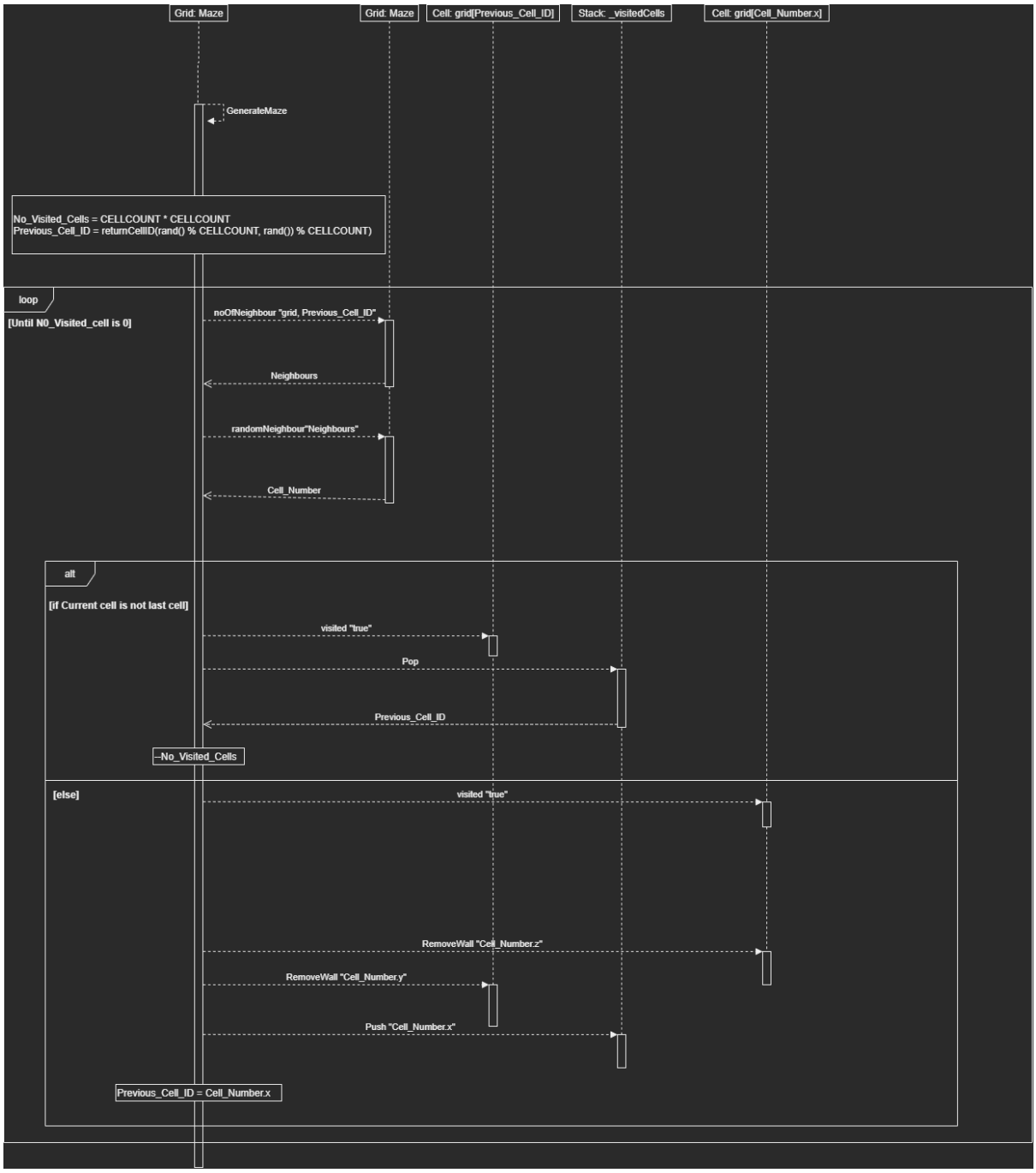
This new window which can be thought of as the game window, will then begin a sequence of actions that will load entities such as a player and enemies after which the program will initialise a maze object which will generate a maze with no load times meaning less than 30 milliseconds, after which the game will begin, and the use can interact with the player. The enemy entities will have randomly generated moves with different states that allow players to pass through them as it's a bunch of corridors it would only be fair. The players goal is to reach to the end of the maze after which another window will launch.

The next window is the win screen where they will be presented with the option to play again if they so choose. Conversely if the enemy actually catches the player in the previous screen, then the next window will be the fail screen which will give the user a tip and a button to retry.

### UML Class Diagram:



## UML Sequence Diagram of process for generating a maze:



## Required Roles

Classes –

| Responsibility | Type Details | Notes |
|---|---|---|
| Entity | | This is a template class that will be used as a base class to make new entities via inheritance |
| Player | x, y, index, color | This is the player class that will oversee making and allowing the player to be interactable. |
| Enemy | x, y, pointCount, index, color | This is the enemy class and will oversee making, moving the enemy as well as maintaining the enemy state which is essential to making the game beatable |
| Stack | | This will be needed as the backtracking algorithm which is used in the maze generation heavily relies on being able to move backwards and this will allow such functionality |
| Wall | | This class will be used to maintain the wall configuration of each cell in the 2d grid |
| Cell | x, y, height, width, color | This object should be able maintain a body and return it when required to, to render a cell, as well as provide information about the cell when requested to. |
| Grid | | This is the class that will handle the generation of the maze after making a 2d grid of cell objects and should be able to send out the cell bodies as well as their walls when required to. |
| Button | x, y, height, width, color, text | This class should be able to display a button on the required area and have |

| | | |
|---|---|---|
| | | other visual features such as highlight on hover and such. |
| UIText | x, y, characterSize, text, color | This should be able to allow the user to display text as titles, paragraphs or labels wherever required. |

Stack Struct –

| Variable | Type | Notes |
|---|---|---|
| value | int | An integer variable that holds the cell index |
| Next | *Stack | A pointer variable that points to the next node |

WALLS Struct –

| Variable | Type | Note |
|---|---|---|
| Top | Bool | Holds whether the north wall is present or not |
| Right | Bool | Holds whether the east wall is present or not |
| Bottom | Bool | Holds whether the south wall is present or not |
| Left | Bool | Holds whether the west wall is present or not |

Int4values Struct –

| Variable | Type | Notes |
|---|---|---|
| Top_present | Bool | Holds whether the north wall is present or not |
| Right_present | Bool | Holds whether the east wall is present or not |
| Left_present | Bool | Holds whether the west wall is present or not |
| Bottom_present | Bool | Holds whether the south wall is present or not |
| Top | Int | Holds the cell index of the north cell |
| Right | Int | Holds the cell index of the east cell |
| Left | Int | Holds the cell index of the west cell |
| Bottom | Int | Holds the cell index of the south cell |

Bool4values Struct –

| Variable | Type | Notes |
|---|---|---|
| X | Bool | Holds a Boolean value |
| Y | Bool | Holds a Boolean value |
| Z | Bool | Holds a Boolean value |
| w | Bool | Holds a Boolean value |

CELLEVENTS Enumeration -

| Value | Notes |
|---|---|
| PATH | Denotes that this cell is a path |
| START | Denotes that this cell is the start of the maze |
| END | Denotes that this cell is the end of the maze |