3815ICT – Software Engineering

Minesweeper

Milestone 1 – Conceptual Design

Student: Zak Barker

Student#: S5085150

Subject: 3815ICT Software Engineering

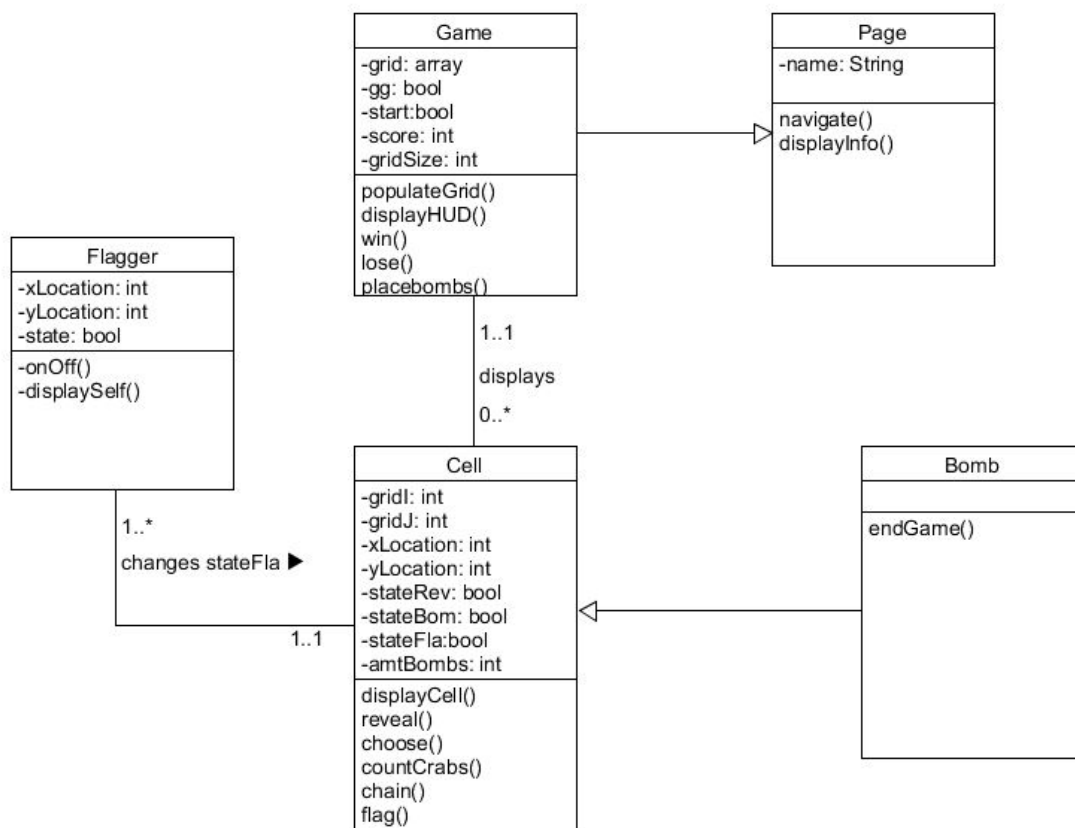Email: Zak.Barker@Griffithuni.edu.au

1. Class Diagram

Figure 1 – Class Diagram for Project

2. Description of Classes

1. Page & Game Class

The page Class stores information about each page. This is a basic class with functionality to show pages and navigate between them. This is the superclass for **Game** which is an essential class.
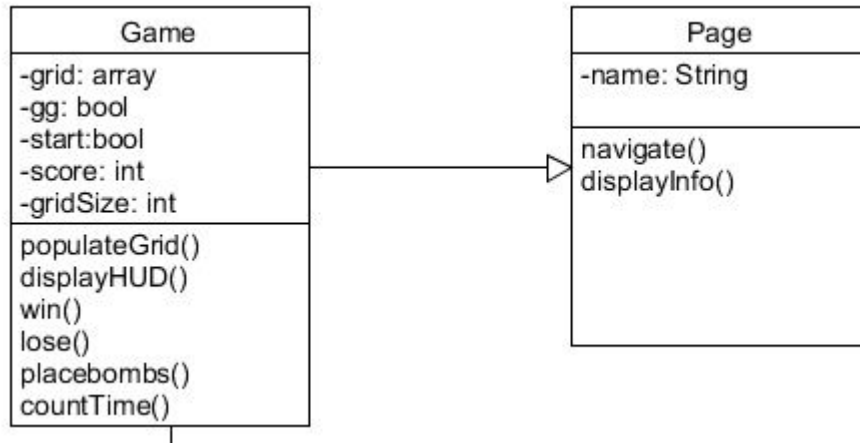
Figure 2 – Class Diagram for Page and Game

The Game class is utilized for each game mode. It will contain arbitrary information such as gridSize and score needed for calculations. The important members are the grid which holds all of the Cells. The gg and start members signify when the game starts and finishes.
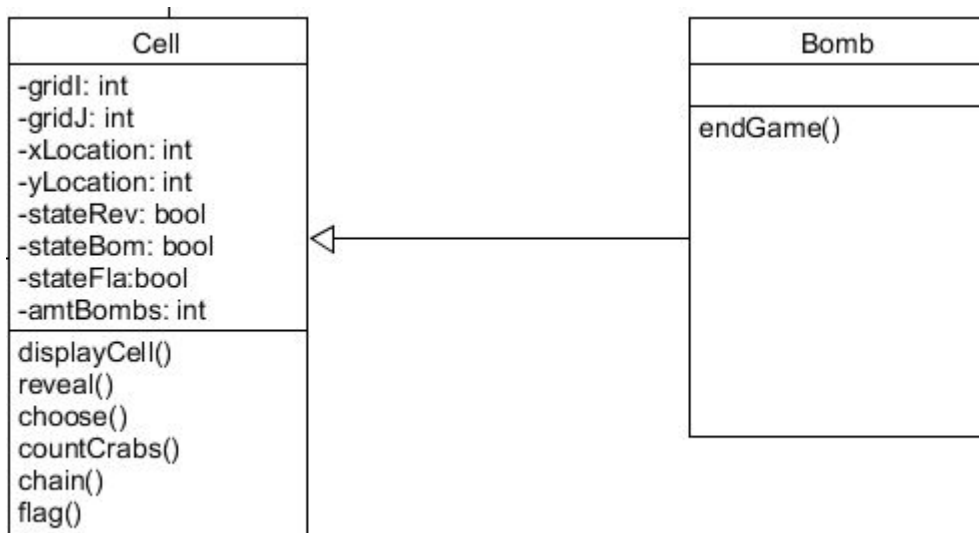
Functions include

- Populate Grid – This function adds all of the cells to the grid
- Display HUD – This function displays the 'heads up display' information such as the score
- Win and Lose – These functions govern the functionality after victory or loss conditions have been met.
- Place Bombs – This function randomly plants bombs throughout the grid.
- Count Time – This function keeps track of the timer and stores it as score.

2. Cell & Bomb Class

Each Cell carries information about its location within the grid. It also stores a reference to its location in the GUI. It contains three Booleans which relate to its state.

- Rev = Whether state is revealed or not
- Bom = Whether state is a bomb or not
- Fla = Whether state is flagged or not

Finally, each cell contains information about the amount of bombs neighbour it.

```
         Cell                              Bomb
─────────────────────           ─────────────────────
-gridI: int                     
-gridJ: int                     ─────────────────────
-xLocation: int                 endGame()
-yLocation: int                 
-stateRev: bool      ◁──────────
-stateBom: bool                 
-stateFla:bool                  
-amtBombs: int                  
─────────────────────           
displayCell()                   
reveal()                        
choose()                        
countCrabs()                    
chain()                         
flag()                          
```
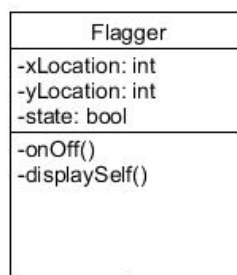
The methods attached to the Cell class are

- Display Cell – Used to display Cell on the GUI

- Reveal – Reveals Cell when clicked

- Choose – Simple function to determine whether Cell has been clicked(chosen)

- Count Bombs – Counts neighbouring cells which have bombs

- Chain – When an empty Cell is revealed, chain will reveal all the neighbouring cells.

- Flag – Function will change the Flagged state of the cell

The bomb class is a simple child class of Cell and its only additional function is End Game. This will trigger a loss condition when bomb is revealed.

3. **Flagger**

```
        Flagger
─────────────────────
-xLocation: int
-yLocation: int
-state: bool
─────────────────────
-onOff()
-displaySelf()

```

Flagger is an object which activates the Cell's flagged state. It stores information about its own location in the GUI along with a state – Whether it is turned on or off. If flagger is turned on, it will have the functionality to flag unrevealed Cells. The methods follow:

- On/Off – This function turns the flagger either on or off
- Display Self – This displays the flagger on the GUI

3. Documentation on Classes

It is important to note that due to working in JavaScript, Doxygen was not a possibility for documentation generation. In fact, the creation of classes is not possible in JavaScript. There is a technique using a prototype which is able to add methods to a function. This function is able to contain members and act as a Class. It is not exactly the same, however, it was necessary in order to work in a p5 environment. It is possible to emulate classes within JavaScript by utilizing this technique.