Project Specification – Minesweeper

Names: Andrew Chang, Michael Lee

Period 5

Overall Description

Minesweeper is a simple video game.

In this project, we created a complete minesweeper game with a 2D array and recursion.

When a player runs the program, the game window will show up and there will be a menu bar and a game board. Using the items in the menu bar, the player will be able to see the best records, change the size of the game board, play a new game, or exit.

To play minesweeper, you will have to left-click or right-click each cell and uncover all the cells in which mine is not buried. When you left-click a cell, one of the three events will happen. If there is no mine buried in the cell and also no mine is buried nearby, the cell will open itself and also all the adjacent cells that don’t have mine buried inside of them. If there is no mine buried in it but there are mines within the closest 8 cells, the cell will only uncover itself and display the number of the mines detected within the closest 8 cells. Lastly, if there is a mine buried inside the cell the player clicks, the game will be over and show the “game over” pop-up window. When you right-click a cell, you can mark the cell with a flag by right-clicking to prevent misclicking and failing the game.

If you succeed in opening all the cells that mine is not buried in or you marked all the mines with flags correctly, a “Congratulations!” pop-up window will appear and show you the current record and the best record.

Class/Interface Overview

Top-Level Class

Minesweeper – This is the main class of the project. It will count time with MyTimer, and create a game window.

Major Classes

Board – This class manages the change in each cell. It will initialize the game board when the game starts. It has a 2D array of Cells where the game will actually run. It processes the inputs and applies them to the game.

Cell – a single cell of the board that contains information of is the cell revealed, if there is a mine or a flag in the cell, and what number should be displayed on the cell.

MyTimer – a timer that starts counting when the game is opened. It will count the time and the record of that game will be shown on a pop-up window at the end of the game. The timer will be reset whenever a new game starts.

GUI Classes

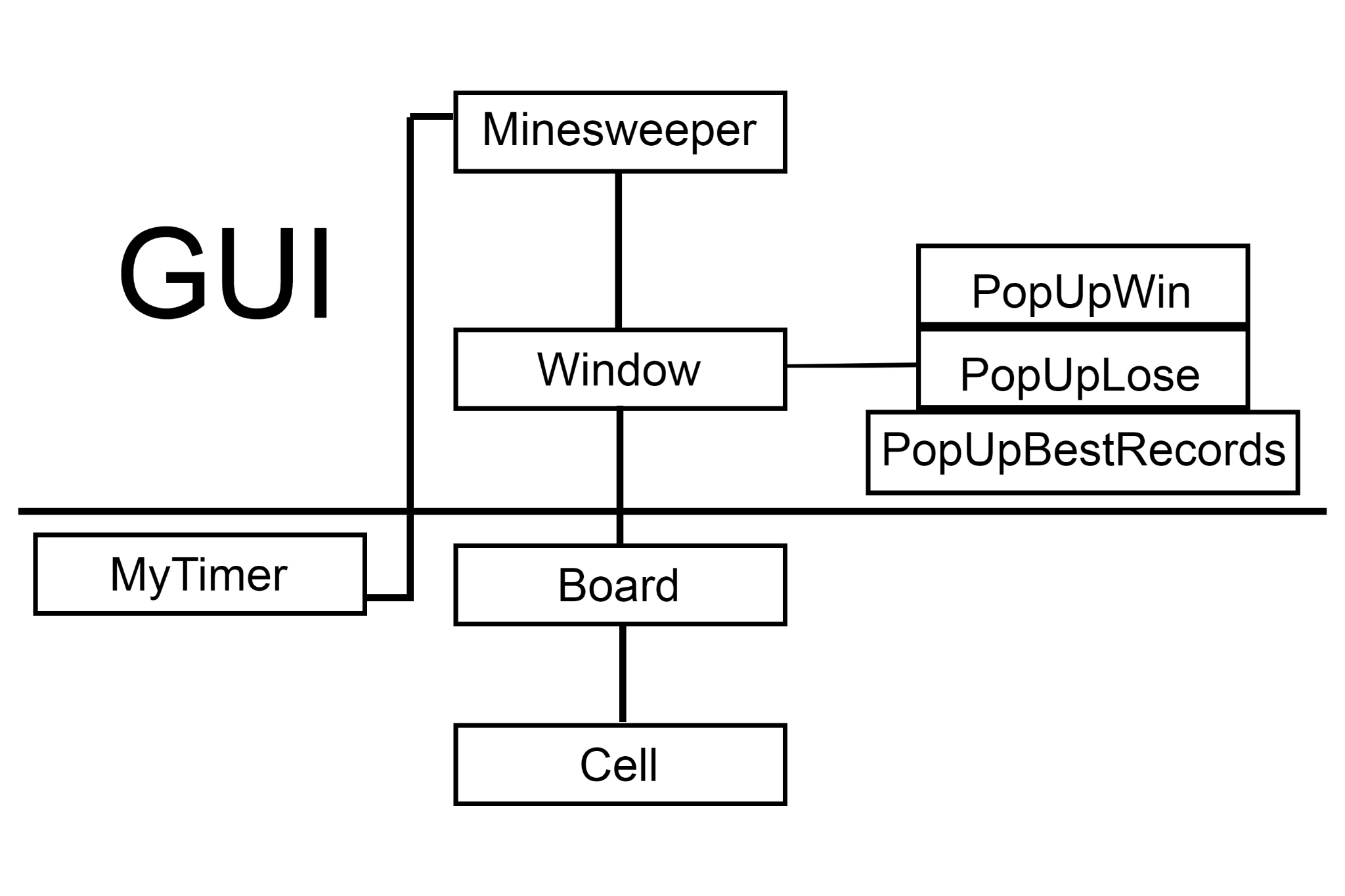
Window – This class will display the game, manage inputs using GUIs, and record the best records.

PopUpWin – a pop-up window that appears when the player satisfies the win conditions and wins a game. Displays the time taken for the game and the best records for current difficulty.

PopUpLose – a pop-up window that appears when the player satisfies the lose conditions and loses a game. Displays the time taken for the game and the best records for current difficulty.

PopUpBestRecords – a pop-up window that appears when a player clicks “BestRecords” in the menu bar. Displays the best records for each difficulty. If nothing is recorded, show “--:--:--” instead of the best time.

Rough Class Diagram



Structural Design

The following data structures will be used.

| **Description** | **Structure** |
| --- | --- |
| The game board | Cell[][] |

HIgh Level Major Class Specifications

Board

* Attributes
  + Cell[][] gameBoard
  + int boardSize
* Methods
  + void arrayChange(int size)
  + void assignMines()
  + void assignNum()
  + void clearBlanks(int row, int col)
  + int countAround(int row, int col)
  + boolean inBounds(int row, int col)
  + void refresh()

Cell

* Attributes
  + boolean isMine
  + boolean isRevealed
  + boolean isFlag
  + int count
* Methods
  + int getCount()
  + boolean getIsFlag()
  + boolean getIsMine()
  + boolean getIsRevealed()
  + void setCount()
  + void setIsFlag()
  + void setIsMine()
  + void setIsRevealed()

MyTimer

* Attributes
  + String curTime
  + int timeCnt
  + boolean stop
* Methods
  + string getCurTime()
  + int getMin()
  + int getSec()
  + int getTenMilSec()
  + void resetTimer()
  + void startTimer()
  + void stopTimer()