

Minesweeper

Brian Herman & Diana Carrillo

March 7, 2014

Software Design Document

TABLE OF CONTENTS

Purpose.....	3
Scope.....	3
System Overview.....	3
Architectural Design Diagrams.....	3-4
Decomposition Description.....	4-5
Data Dictionary.....	5
Overview of User Interface.....	5-6
Sample User Interface.....	6-8
Requirements Matrix.....	9

Purpose

This software design document describes the architecture and system design of a Minesweeper game.

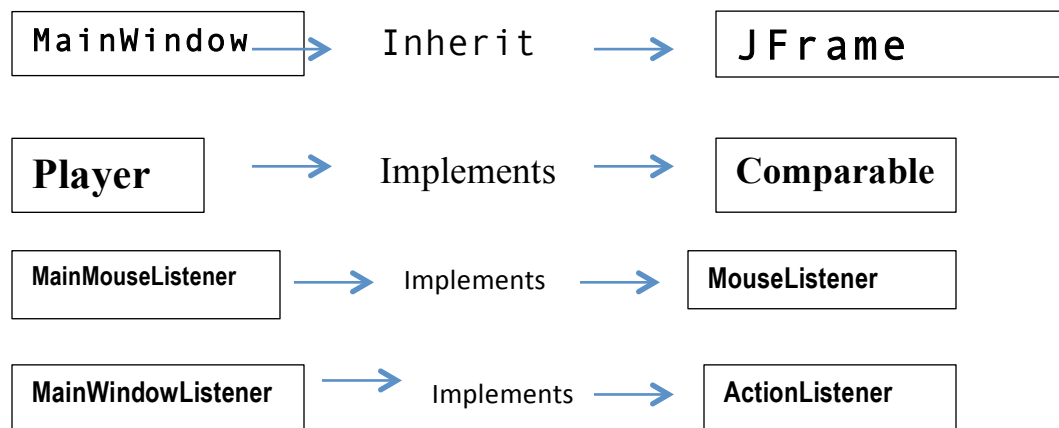
Scope

This project was set out to create the game of minesweeper. Our objective was to have a single user click buttons, and attempt to disable 90 buttons, without clicking on any of the ten bombs. Other functions include: the ability to right-click and the user can either place an "M", meaning "there is definitely a mine here, or a "?" on the button, meaning there may or may not be a mine under this button; a number displaying the number to mines "left"—according to how many "M"s are on the board; a timer that doubles as a score; a game menu with the following functionality: reset game (start over), top ten scores, reset top ten scores, and exit; there is also a help menu that gives directions to play the game and an about section. This program is mainly for entertainment purposes, and has the full functionality of the classic Minesweeper game.

System Overview

Our entire program consists of three classes, a main window class, a class that contains the minesweeper board, and lastly the player class.

Architectural Design Diagrams



The **MainWindow** class inherits from **JFrame**, allowing us to utilize all the GUI stuff that **JFrame** has to offer. **MainWindow** contains **main**, the code for the actual window, menus and the action listener for the main window, the mouse listener, a method for resetting the game, getting the top ten scores, and the flood fill algorithm.

The MinesweeperBoard class creates the board, sets up mines, checks for revealed/visible cells, gets the length and width of the board, and checks if the user has won.

The Player class implements “Comparable<Player>”, which allows us to be able to compare the scores for the top ten list.

Super Class: JFrame	
Class MainWindow	
Methods:	
Main	Sets menu on top bar, creates new window
MainWindow	Creates menus, displays, action listeners, basically makes the window functional
MainMouseListener	Allows for the right-clicking functionality(flags, etc.)
MainWindowListener	Controls the menus and buttons

Decomposition Description

MainWindow Class

1. Main
 2. MainWindow
 3. MainMouseListener
 4. MainWindowListener
- 1-Creates a new board (MinesweeperBoard Class)
 - 2-Puts the title of the game on the top of the window
 - 2-Creates listeners to interact with user
 - 2-Creates the board data
 - Randomly places bombs (with help from MinesweeperBoard Class)
 - Creates a grid of buttons (Thanks JFrame), menu bar with all the options & adding action listeners for them
 - Sets up the layout of how the user will see the game
 - Displays the number of flags and the timer & adds them to JPanel
 - 3-Allows the user to right-click: once to disable the button and display an “M”, twice for a “?”, and the third time it resets back to an enabled button.
 - 4-Checks if the user won (disabled 90 buttons without hitting a mine, with help again from the MinesweeperBoard Class) & add their name to the empty “Top Ten” list, or if they beat a score already on the list
 - 4-Option to restart game
 - 4-Implements the flood fill algorithm

MinesweeperBoard Class

1. MinesweeperBoard
2. get
3. revealed
4. length
5. width
6. getMines
7. isWon
8. setVisible

- 1-Randomly places bombs
- 1-Sets the numbers on the board which indicate where the bombs are
- 2-gets a cell from the board to access it from other classes
- 3-checks to see if the cell is revealed for the revealing algorithm
- 4-Gets the length of the board
- 5-Gets the width of the board
- 6-Gets the number of mines on the board
- 7-Checks to see if the user has disabled 90 non-mine buttons
- 8-Makes a cell visible so

Player Class

1. compareTo
2. toString

- This class is essentially used to sort the players in the ArrayList that is the top ten scores.

Data Dictionary—See the JavaDoc

Overview of User Interface

When running our Minesweeper, the user will see two menus on the top-left corner. Under the “Game” menu, the user has the following options:

- Reset-The user can abandon the current and begin a new game
- Top Ten Scores-A list of the top 10 scores (the lowest amount of time to click 90 non-mine buttons)
- Reset Top Ten Scores-Clears the list of the highest scores
- Exit- Exit the game

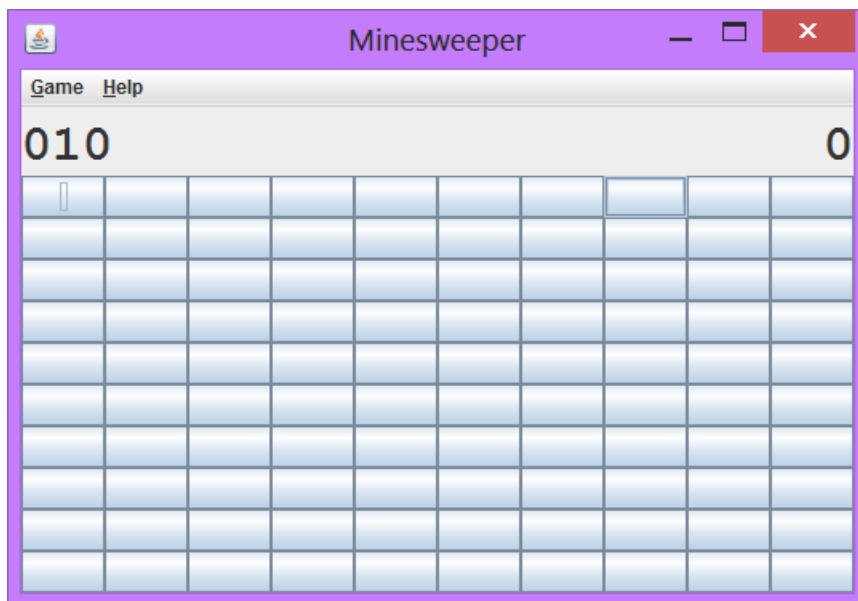
Under the "Help menu", the user will see the following:

- Minesweeper Help- Gives the user a simple description of how to play the game
- About- Lists the creators of the program

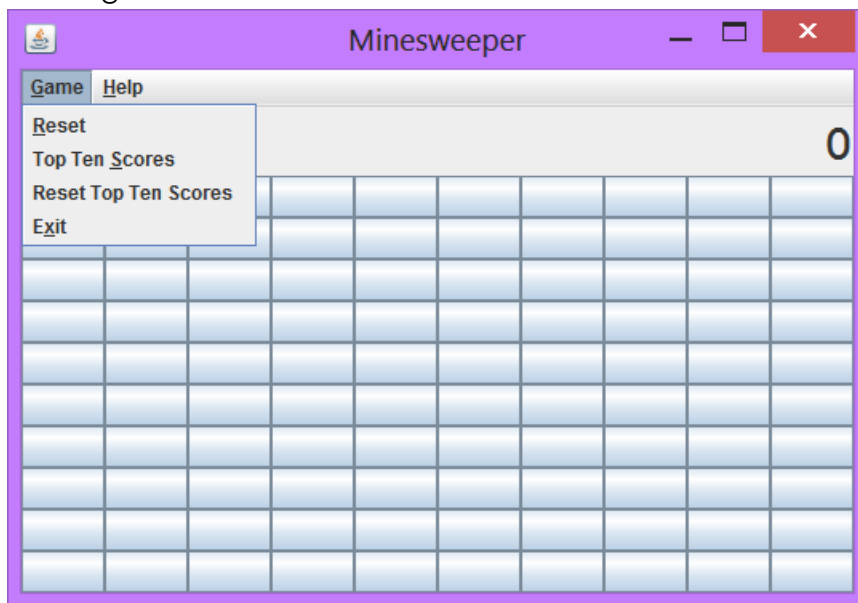
The rest of the functionality lands on the 100 buttons. Under 10 of these buttons, there are mines that, when clicked will cause the player to lose the game. If the user right-clicks a button 1 time, the button will be disabled and will display an "M"- for "there IS a mine"; if the user right-clicks a button 2 times, the button will be disabled and will display a "?"-for "there may or may not be a mine".

Sample of User Interface

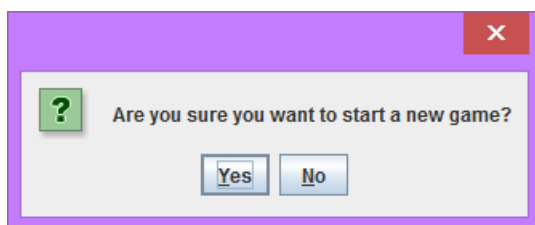
A sample of what the board will look like with a new game:



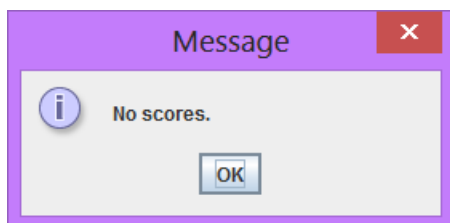
Viewing the "Game" menu:



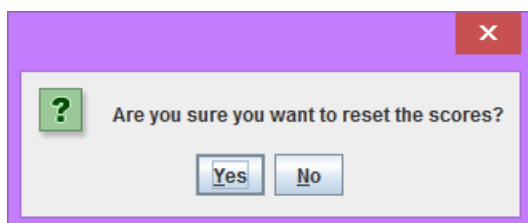
Dialog box that pops up when user clicks "Reset":



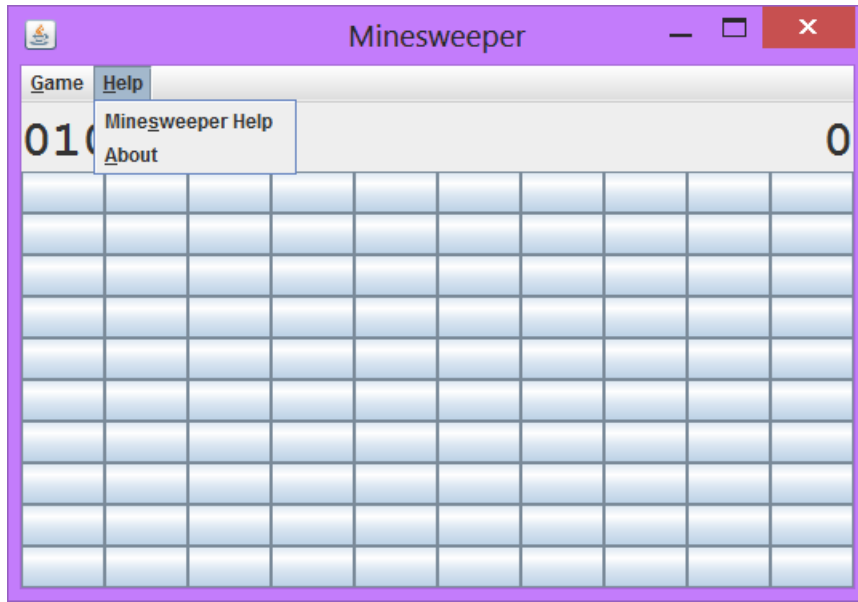
Dialog box that pops up when "Top Ten Scores" is clicked (with no scores):



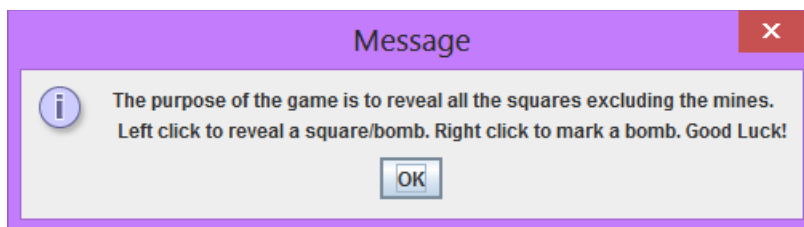
Dialog box that pops up when "Reset Top Ten Scores" is clicked:



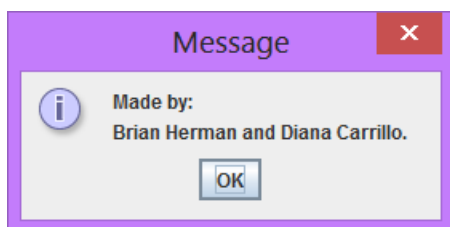
Clicking the "Help" menu:



Dialog box that pops up when "Minesweeper Help" is clicked:



Dialog box that pops up when "About" is clicked:



REQUIREMENTS MATRIX

Requirements:	Done:	Not Done:
10 mines, 10x10 grid	X	
Left/Right Click Functionality	X	
No text on 100 buttons	X	
Games ends on mine click	X	
Games ends on 90 buttons disabled	X	
On Left click, check if button is disabled	X	
All mines are shown when one is clicked		X
Shows number adjacent mines, if clicked & no mine	X	
# of mines left to be found	X	
Reset option	X	
Timer starts @ click	X	
Game Menu	X	
Top Ten	X	
Exit	X	
Help	X	
About	X	
Mnemonic option	X	
Clear Top Ten	X	
Use image for mine	X	