# Minesweeper Clone

HCI 2018-2019 Programming Assignment

Prof. Andrew D. Bagdanov

November 12, 2018

## 1 Overview

Minesweeper is a single-player puzzle video game. The objective of the game is to clear a rectangular board containing hidden "mines" or bombs without detonating any of them, with help from clues about the number of neighboring mines in each field. The game originates from the 1960s, and has been written for many computing platforms in use today.

The player is initially presented with a grid of undifferentiated squares. Some randomly selected squares, unknown to the player, are designated to contain mines. Typically, the size of the grid and the number of mines are set in advance by the user, either by entering the numbers or selecting from defined skill levels, depending on the implementations. (In the Microsoft variant, this is limited to 30 times 24 with 667 mines.)

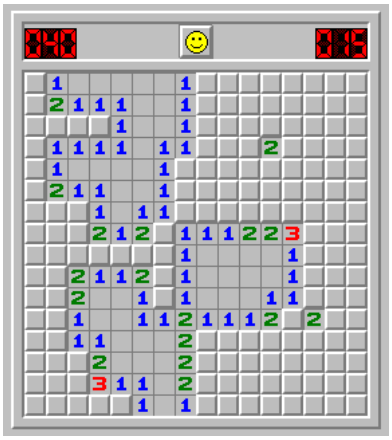
The game is played by revealing squares of the grid by clicking or otherwise indicating each square. If a square containing a mine is revealed, the player loses the game. If no mine is revealed, a digit is instead displayed in the square, indicating how many adjacent squares contain mines; if no mines are adjacent, the square becomes blank, and all adjacent squares will be recursively revealed. The player uses this information to deduce the contents of other squares, and may either safely reveal each square or mark the square as containing a mine. The game is won when all mine-free squares are revealed, because all mines have been located. In this programming assignment you will implement the Minesweeper game as a GUI application.

## 2 Rules of the Game

The of Minesweeper are fairly simple:

- You are presented with a board of squares. Some squares contain mines (bombs), others don't.
- If you click on a square containing a bomb, you lose. If you manage to click all the squares (without clicking on any bombs) you win.
- Clicking a square which doesn't have a bomb reveals the number of neighbouring squares containing bombs. Use this information plus some guess work to avoid the bombs.
- To open a square, point at the square and left click on it.
- To mark a square you think is a bomb, point and right-click.
- If you mark a bomb incorrectly, you will have to correct the mistake before you can win. Incorrect bomb marking doesn't kill you, but it can lead to mistakes which do.
- You don't have to mark all the bombs to win; you just need to open all non-bomb squares.

As with most things, a picture (or in this case a **website**) is worth a thousand words.



Here is an online implementation that you can use as a basis for your implementation: http://minesweeperonline.com/

## 3 Assignment

For this assignment you must implement a the Minesweeper Game. You may implement this in **any programming language** and using **any GUI framework** you desire. Almost anything goes, however these features your GUI must implement to receive full credit:

- **A working version of Minesweeper**: your GUI must implement the Minesweeper game in its entirety. The game must be **playable**, and must correctly detect the win and lose conditions of the match.
- **Variable Grid Size**: your implementation **must** let the user select the size of the grid from a menu.
- **Status Indication**: your implementation **must** indicate the number of mines unmarked and the time since the start of the game.

Your Minesweeper implementation can also support the following features (which will be considered as **extra credit** in the final evaluation:

- **Saving and loading of game state**: your Minesweeper Clone might support the loading of a serialized version of the board state. This can be any file format you choose. If the user quits a game in progress, the game should save the board state. When the game is run again, it should continue from the previously saved game.
- **Leaderboard**: your implementation might track the high scores for each board score. High scores are measured as the shortest time to win the game.

## 4 Evaluation

You **must** submit the source code for your complete implementation by the **deadline for registering for the exam**. Late submissions will **NOT** be accepted.

A note on **cheating**. I realize that there are **many**, **MANY** implementations of Minesweeper out there. That is great, and you should look at them and learn from them. However, the code you submit **MUST BE YOUR OWN WORK**. Learn from others, but **write and submit your own implementation**.

Your implementation will be evaluated based on how you apply the programming models and constructs learned during the course. More specifically:

- **Functionality** (25%): your code must **work**. In your submission, include a minimal README that explains how to run your program (including any dependencies).
- **Code hygiene** (25%): keep things **clear**. This means writing concise, clear, and **reasonably documented** code. You should carefully identify **model**, **view**, and (maybe) **controller** components of your implementation.
- **Completeness** (25%): did you implement all of the **required** features? See the list above, but your GUI must implement all of the required features to earn full credit.
- **Correctness** (25%): does your GUI **correctly** implement all of the required features? Above all, your GUI should not crash and should operate consistently with the assignment requirements.
- **Extras** (max 10% bonus points): above and beyond. If you implement **extra features**, make sure you highlight them in the documentation for your submitted project.

## 5 Hints

Here I will collect some useful hints and advice for anyone choosing this programming assignment (this list might be updated, check back):

- The focus of this assignment is on **complete** and **correct** implementation using **best practices** (i.e. MVC). Your interface does not have to be **pretty**. Make it all **work**, make it work **correctly**, and **then** make it pretty if you have time.
- That being said, points will be deducted if there are any strange or **buggy** behaviors of your game. Your Minesweeper clone should be complete, correct, and consistent.

## 6 Useful Links

There are a **TON** of resources on Minesweeper. It's almost pointless to list them here, just Google it. But anyway:

- The Microsoft Windows version of Minesweeper – this is likely the version we all know.
- A page with some interesting mathematical facts about Minesweeper.