CS9053
Section I2
Prof. Dean Christakos
Final Project
Released: April 21st, 2021
Due: Friday, May 14th, 2021

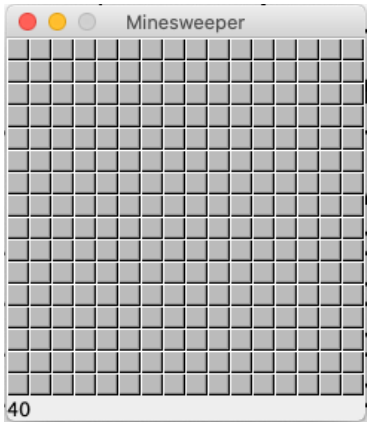
You are going to implement the game Minesweeper, with a few key variants.

Introduction to Minesweeper

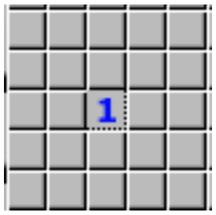
You may be familiar with the game Minesweeper. If not, I've enclosed a demonstration you can run by executing:

```
java -jar minesweeper.jar edu.nyu.tandon.Minesweeper
```

As you can see, this creates a 16x16 grid of tiles, and there are 40 mines hidden throughout the grid.

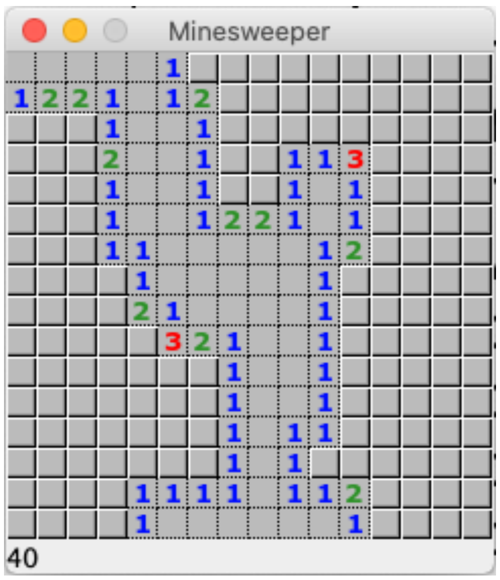


If you click on a tile, hopefully there won't be a mine there, but you will see a number indicating how many mines there are in adjacent tiles. For example, here, the number is "1", which means in one of the 8 surrounding tiles, there is a mine:

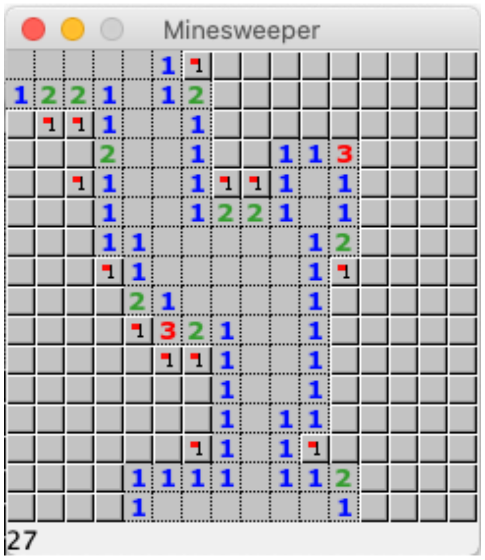


Of course, we don't know which one, and we can only guess and hope that when we click an adjacent tile, there will be only a 1 out of 8 chance of hitting a mine.

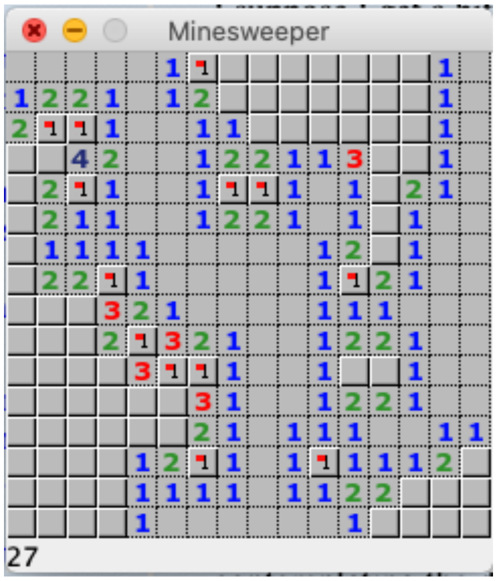
In the event that there are no adjacent mines (ie, that the number would be "0"), the game will automatically reveal tiles until it comes across tiles with an adjacent mine (and, thus, a number):



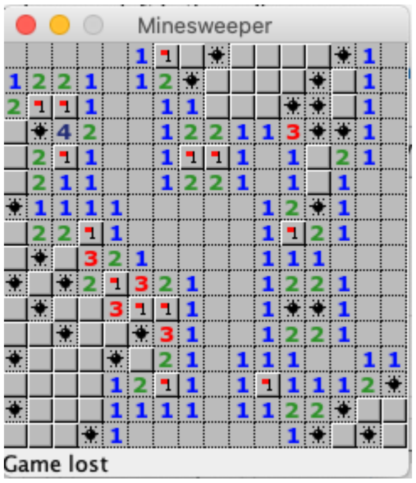
This makes things easier because now you can determine where some of the mines definitely are. You can right-click on the tiles where we can certainly be sure contains mines.



This makes things easier because you have an indication for where the mines are (assuming you clicked correctly), and you can see that the number of mines has counted down from 40 to 27. Then you can click on tiles where you are sure there are no mines. For example, if a tile has a number of "1", and you've set a flag on an adjacent tile, you can be sure that none of the other tiles have a mine, so you can click those:



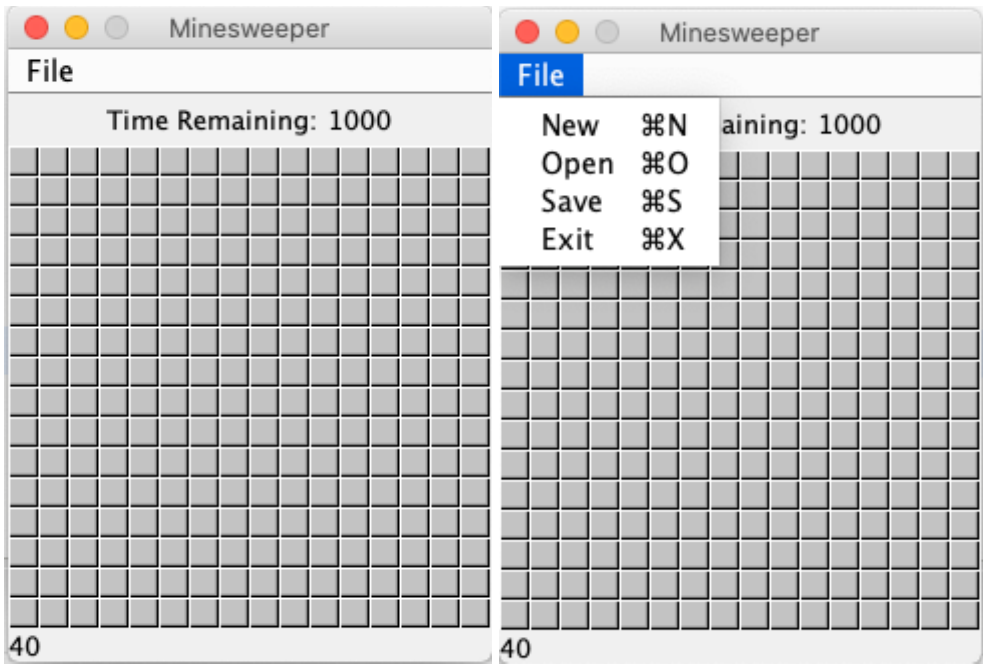
Now you can see a few new opportunities where there are definitely mines. You can continue this until you finish all the tiles or hit a mine. In the event you left-click on a tile with a mine on it, the mines are revealed, and the game ends as a loss:



The Assignment

You are going to implement minesweeper, with a few distinct additional features:

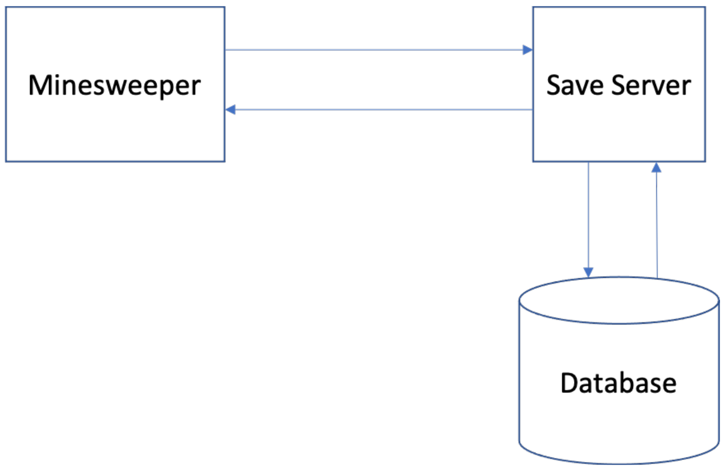
- You will be able to load and save previous games
- The game will have a 1000 second timer that will count down to zero, and the remaining time on the counter will be your score
- When the counter hits zero, you lose, just as if you had hit a mine



You do not actually have to implement the keyboard shortcuts on the menus, but those are worth doing if you're interested.

Saving and loading games

Saving and loading games will be performed on a server. The server will have a database where the saved games will be stored. In general, the architecture will look like this:



The server will be a network server that accepts connections from the Minesweeper application and be designed to handle multiple connections from different Minesweeper applications. It will send over a copy of the game state (including the amount of time left) to the save server, which will store that data in the database.

Extra Credit:

For 5 points extra credit, you can make a module that stores the top 5 high scores in the database and allows the player to view them by selecting a menu item in the Minesweeper app.

Guide, Hints, and Structure:

Obviously, you will want to generate a 16x16 grid indicating where the mines are and where the tiles are. Each element of the grid has either a blank space, a mine, or a number from 1-8 indicating how many mines are adjacent to that space.

In the beginning, each space will be covered by a cover tile. Left clicking on a cover tile removes it. If the underlying tile has a mine, then the game is over, and the score goes to zero. If the underlying tile is empty, then all the adjacent tiles are removed as well. Otherwise, it reveals the underlying tile which will be a 1-8 numbered tile. Right clicking on a cover tile "flags" it as a mine and reduces the count of mines at the bottom.

The state of the game records how much time is left and the state of the 16x16 grid. Now, storing a 16x16 grid in a database table might be difficult. However, storing the game state as an object in the DB may make that easier.

You will be provided with all of the icons, so that should make things straightforward.

Note:

You can discuss the project with other students, but you must write your own code.

Furthermore, there are many examples of Minesweeper source code. Rest assured that we are aware of those examples as well, and you have to write your own code and will compare yours against those other examples!

You have a lot of time, so build these pieces one by one and try things out. Build the 16x16 grid with all the cover tiles. Figure out how to click on a tile and reveal what's below or right-click on a tile and mark with a flag. Then figure out how to randomly place the mines and then place the numbered and blank underlying tiles. Then cover them with the cover tiles and figure out how to automatically reveal tiles with a no adjacent mines in response to a click, and so on.