

CS111 Project Description, Spring 2013

In this project you will be using the knowledge you gained from lectures and homeworks and applying them to a lengthier programming assignment. The answers to some of your questions do not strictly reside in this document. You are encouraged to look at other sources for inspiration. This semester's project will involve the casino game of Minesweeper. This is an *individual* project, and as such you will be held to the University's academic integrity policies. You are welcome to discuss strategies and ideas with other students (and will be encouraged to in recitation), but you must code the project on your own.

This project will have multiple phases, with regular checkpoints due in the coming weeks. It is your responsibility to remember these dates and to be prepared to present them during recitation. If you do not make a deadline, you can still make up the points with a 20% penalty during the week following the due date. No credit will be given after that point.

Documentation is required. You should document your progress in a text file. Think of this as a progress diary. What worked? What didn't? What did you have trouble with?

This project is what you make of it. Listed below are point guidelines as to how they relate to the number of points given to the project (total 100). However, you should not be limited by what we suggest the minimum work required here is. To get the most out of this project, I urge you all to push yourselves to find your limits. You might surprise yourself! Good luck!

Milestone 1: Design 10pts (Due week of 3/18/2013)

You are to familiarize yourself with the Minesweeper game play. Play the game a few times. Read and understand all the rules associated with the game <http://windows.microsoft.com/en-us/windows7/minesweeper-how-to-play>, and read this document in its entirety. Do the programming process for a game of minesweeper (except for coding step). Sketch out one iteration of game play – e.g. what the player will see on his screen before and after one interaction with the game. For the purposes of this step, imagine a completely text based board, where cells in the minesweeper grid are denoted by single character values, and that a user's "clicks" are accomplished by asking the user for a location on the board to reveal/mark.

Milestone 2: The Game Board 10pts (Due week of 3/25/2013)

Create a representation of the minesweeper as a 2D Array. Initialize it to some initial state. To make it easier, first create a 5x5 board, and place 5 bombs wherever you would like, then update the numbers that indicate neighboring bombs. Create a second representation of the board that represents the revealed state of each cell (e.g. it has a 1-1 mapping to the first board that indicates whether a cell is in a revealed state) Write code to print this boards. Lastly, create a simple interaction with the user, whereby you ask the user for a particular cell, and you display what is in that cell.

Milestone 3: Game Loop + Method-izing your code 20pts (Due week of 4/1/2013)

Write the following methods, keeping in mind you will have to determine the appropriate inputs/outputs (e.g. arguments/return types).

`printBoard()`: prints a Minesweeper board to screen in text form. Keep in mind, that the printed board should reflect which cells have already been revealed.

`revealCell()`: appropriately modifies the state array when a cell is to be revealed

You should test both of these methods before moving on.

Now, write a main game loop that makes use of the two methods you just wrote. Your game loop should interact with the user asking her for any inputs you had identified in Milestone 1, and trivially, should be able to identify when the user has lost (e.g. hit a mine). At this point, no cascading reveals are necessary – e.g. If a user selects to reveal a blank space, reveal only that blank space.

Milestone 4: Cascading reveals 20pts (Due week 4/8/2013)

You are now to modify `revealCell()` from last week to perform the cascading reveals. This time, instead of just updating the state of the revealed cell, you must determine how to reveal the surrounding blank spaces according to the rules of Minesweeper. Recitation last week should have prepared you to attack this problem.

Milestone 5: Text based game completed + Class-ifying the board 10pts (Due week 4/15/2013)

At this point, your text based implementation should work great! Congratulations! Now, let's see if we can get this to look nicer. Create a class representation of the board including all internal fields and methods. Call `MsGUI.win()` and `MsGUI.lose()` when appropriate. Download the GUI driver from sakai. Have fun playing your new GUI based Minesweeper. You deserved it!

Milestone 6: Enhancements 20 pts Documentation 10 pts (Due weeks of 4/22/2013)

Now, you have a choice. Implement one or more of the following enhancements.

1. Random mine placement (with arbitrary board size)
2. Read in board from a file (with arbitrary board size)
3. Support for non square/rectangular boards (with arbitrary board size)
4. Anything else? (Get peer leader approval before starting this one)

Finally, your documentation (which you've been doing all along right?) is due this week. You will be doing a code review with your peer leaders this week as well.