

# CS335 Program 1

## “Minesweeper”

### Due Wednesday 10/11

## 1. Introduction

In this assignment you will write an IntelliJ Java project which implements the “minesweeper” game: [https://en.wikipedia.org/wiki/Microsoft\\_Minesweeper](https://en.wikipedia.org/wiki/Microsoft_Minesweeper)

The game starts by placing a number of “bombs” in unknown random locations in a 2-D grid of cells. The player sees a grid of blank cells - the location of the bombs is unknown at the start. Play proceeds by allowing the player to click on any cell in the grid. The goal is to avoid clicking on a cell where a bomb is located. If the player clicks on a bomb, the game ends (the timer stops) and the entire board is revealed. Selecting a cell that is not a bomb should cause the cell face to change color and display a number that represents the number of its neighboring cells that contain bombs. Selecting a cell that does not border on any bombs should cause “clearing”, where all cells that are neighbors with the selected one and also do not border any bombs are automatically revealed.

A cell has 8 neighbors: the north, south, east and west cells, together with the four diagonal ones. Cells on the boundary of the grid have fewer neighbors (only five for a cell on an edge and corner cells only have three). The player scores a victory by clicking to reveal all cells that are not bombs without ever hitting a bomb.

The Java program you write to implement the game must implement the complete layout of the game interface, including menu options (**start** new game, **settings** that control grid size and level of difficulty, **quit** the game, and **help**) and game timer. Minimum grid size is five by five; maximum is ten by ten. The height, width, and number of bombs must be able to be set independently. Height and width must be clamped in the range 5 to 10, and bombs from one to one half the number of spaces on the currently defined grid. In addition, the settings pane should let the user (alternatively) choose from three pre-set skill levels: Beginner (5x5 grid with 5 bombs); Intermediate (8x8 with 15 bombs); and Expert (10x10 with 30 bombs).

## 2. Details

You must implement your solution to the interface using Java. Implement the grid of cells and the methods to manage the cells for this game as Java classes. Use private data to store the state of the board, such as which cells have been selected and which contain bombs. With a good design (constructor, accessor, mutator, helper methods, private instance variables, and inheritance) the game can be controlled by a simple driver, which should allocate an instance of the class to set up the game. The constructor of the class must ultimately allow the driver code to assign a variable grid size, and a variable number of bombs. The class should assign bomb locations for the appropriate number of bombs randomly using the random number generator.

## 3. What to Turn In

Submit your IntelliJ project files as a single zipped archive (or tar/gzip) to Canvas. Be sure to name the driver file (with main()) Bombs.java. The names of the other files you may submit can be anything. If you include custom icons for your bombs, be sure to include them in the archive.

## 4. Example Interface Components

