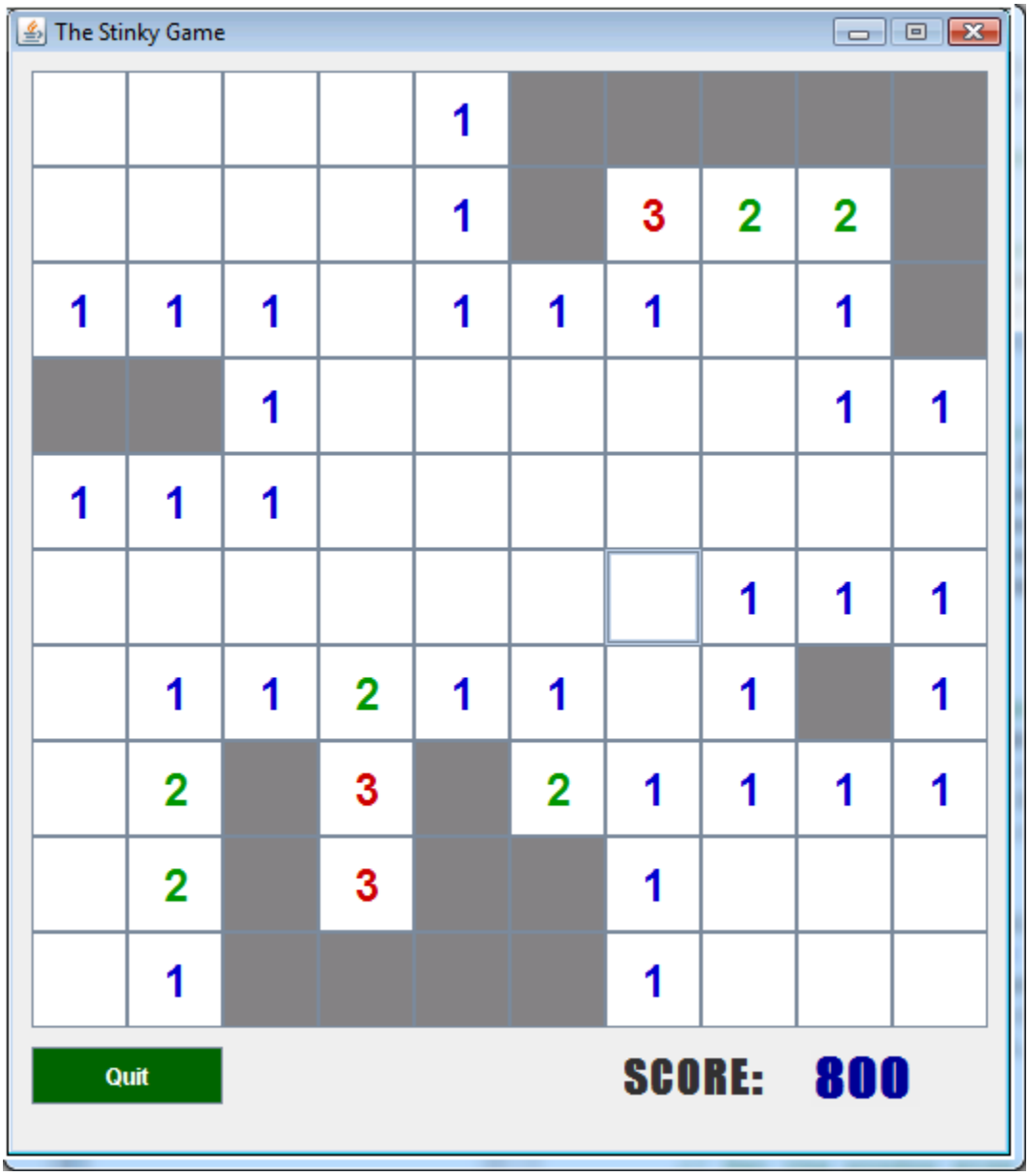
# COMP1406 - Assignment #2

## (Due: **Wednesday, Feb 9th @ 9:00pm)**

In this assignment, you will create a graphical user interface for the **StinkyGame** that you created in assignment #1. Working solutions to assignment #1 will be available online if you want to use that code.

## (1) The **StinkyGameApp**

Create a class called **StinkyGameApp** that allows the user to play the game using a graphical-based user interface (i.e., GUI). This class represents a replacement for the **StinkyGameTextApp**. That is, both applications use the same **StinkyGame** object as their model. You may write the code in any way that you want provided that it produces a window that is VERY similar to the following image and that it follows the interaction rules mentioned below:



You should make a subclass of **JPanel** to represent a panel of buttons and use that panel in your main window. You can adjust the size of the panels as you add them to the main window. You will probably find it useful to create methods in your JPanel class such as **disableButtons(), enableButtons()** and **updateButtons().** The **updateButtons()** could be called to set the background color of the buttons according to a particular **GameBoard** arrangement. It may be a good idea as well to pass in the **GameBoard** for the panel in the constructor of your JPanel class and then store that as the model for the panel.

Here is how the interface should work:

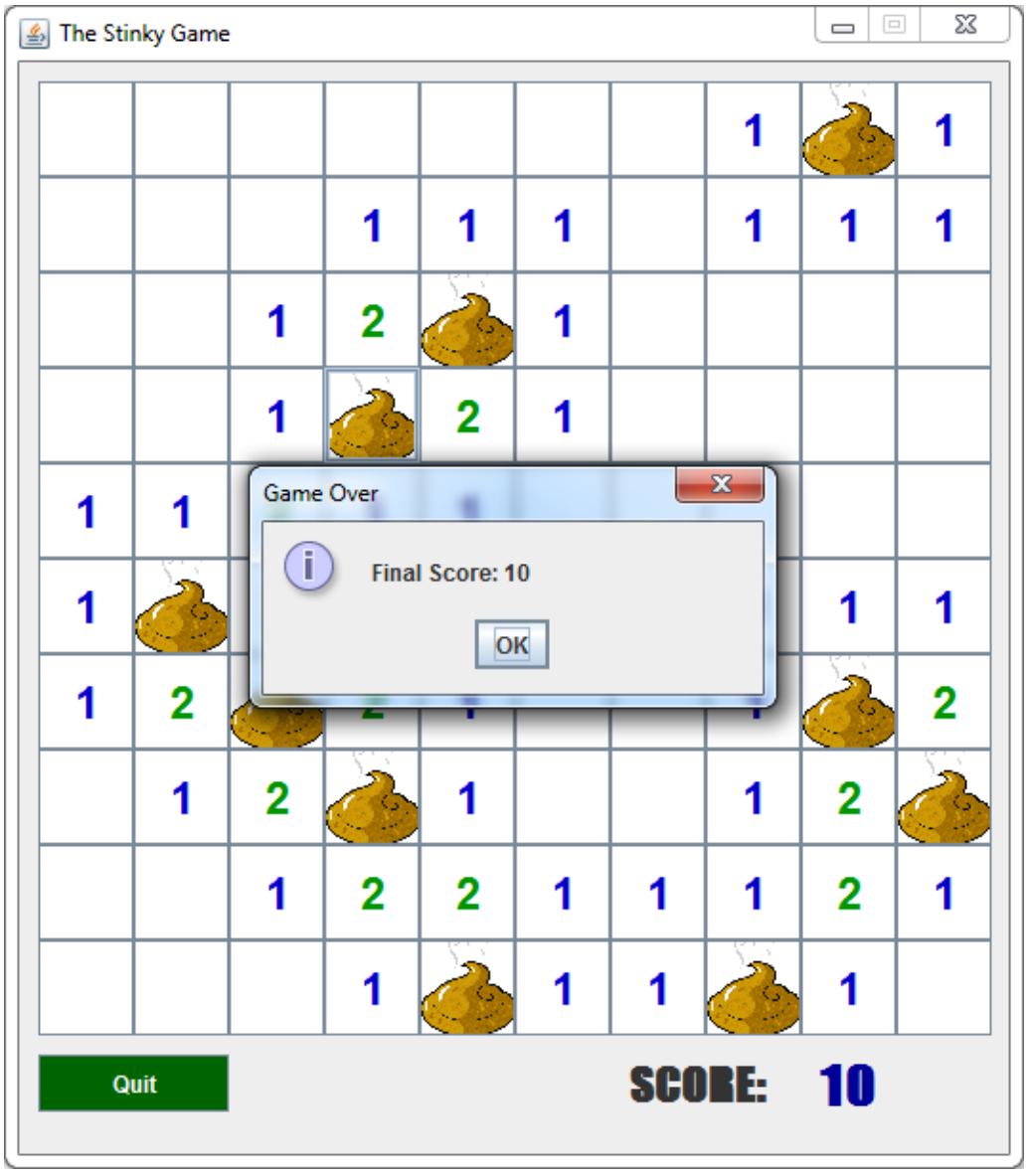
a) The window need not be resizable. You can arrange your components by specifying their size and location individually. You may find this easier to start with. The Score label's font that I used was `label.setFont(new Font("Impact", Font.BOLD, 24));` and the value of the score was the same font, but set to 28pt instead of 24pt … but you can vary these fonts if you want to.

b) The user should be able to click on the buttons on the game board. When a button is clicked, that location in the game board should become uncovered, showing the smell-strength value (i.e., an integer). If the smell-strength is 0, no number should be displayed but the locations should become white. 1's should be shown blue, 2's green, 3's red, and 4&over should be black. The score should also be updated during each uncovering (recall 10 points per uncovering).

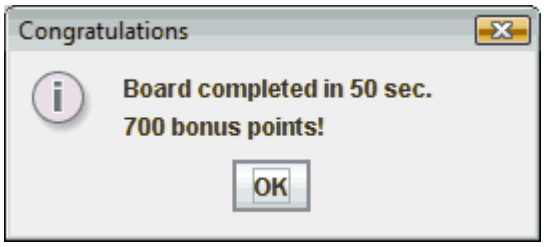
c) Upon startup, the **Start/Quit** button should actually have text that says **Start.** After first pressed, the text on the button should say **Quit** for the remainder of the game.

d) If the **Quit** button is pressed, the window should close.

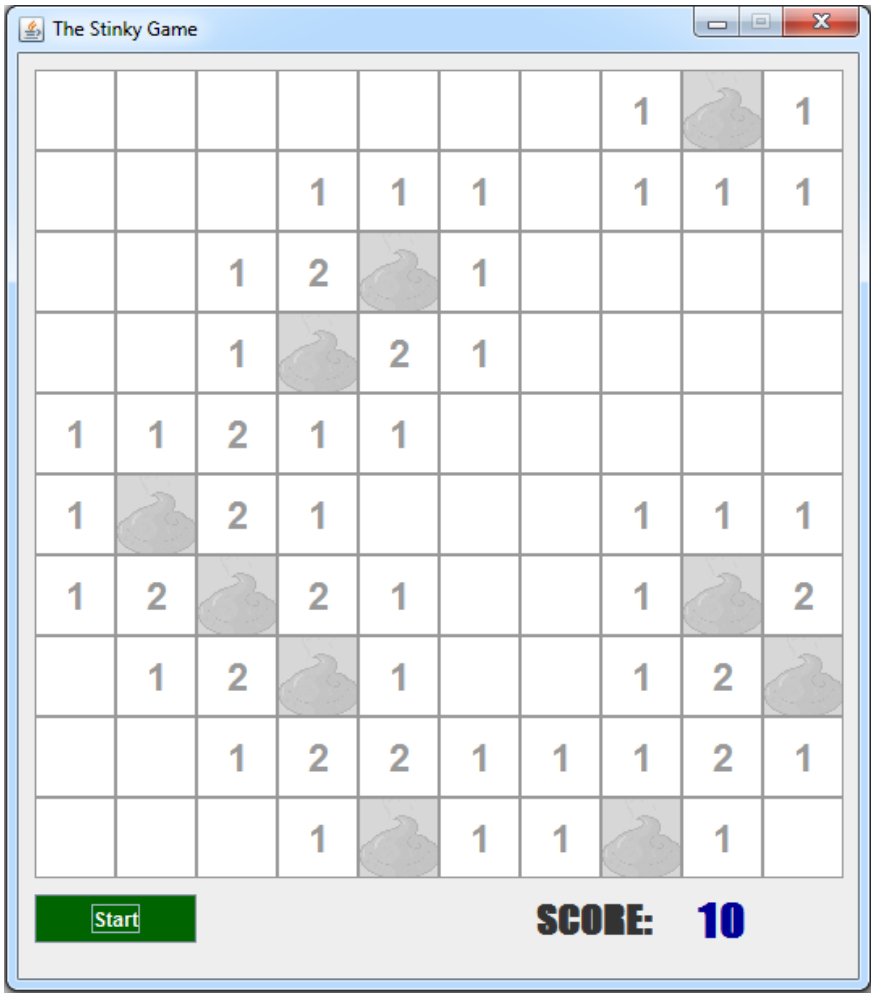
e) If at any time the user accidentally uncovers a "stinky", then the game should end showing the locations of all stinkies (by displaying some kind of ImageIcon on the buttons representing "stinky" locations as shown here) as well as bringing up the following JOptionPane with the final score →



f) If at any time, the user has uncovered all “non-stinky” locations, then the game should not end but the locations of all stinkies should be shown and the following **JOptionPane** showing the time taken to complete the board as well as the final score should be shown:




g) After either of these **JOptionPanes** are closed, ALL game board buttons should be **disabled** and the **Quit** button should say **Start** again as shown here:




h) If the **Start** button is pressed again, the game board should be reset and a new board should appear. If the board was completed successfully, the score should not be reset, as it is the same game which is continuing. Otherwise, if a stinky was uncovered, then a new game should start with a score of 0.

---

**NOTE:** For each coding question, submit your .java and your compiled .class files as well as any necessary .txt files or images. Submit your assignment using **WebCT**. Note that if your internet connection at home is down or does not work, we will not accept this as a reason for handing in an assignment late ... so make sure to submit the assignment WELL BEFORE it is due !