Assignment #1 – TicTacToe and Minesweeper in Java

Date assigned: Tuesday, January 7, 2014
Date due: Part I TicTacToe Thursday, January 9, 2014 (by 1PM)
Part II Minesweeper, Friday, January 10, 2014 (by Midnight)
Points: 75

Create a workspace called your PUNetID. You will drop your entire workspace into the CS260-01Drop folder on Turing when an assignment is due. You can only drop your code one time. I have set permissions this way!!!!

Part I TicTacToe (Should take 2-4 hours)

I have written most of the game of TicTacToe in a project called TicTacToeJava that can be checked out from the course repository svn+ssh://zeus.cs.pacificu.edu/home/CS260Public/2014/SVNROOT_CS260_2014. You are to check out the project (**make sure to Disconnect from Subversion deleting the meta data**) and implement the game of Tic Tac Toe, as shown below, in the file PlayTicTacToe.java. Write methods as necessary; however, all methods in PlayTicTacToe.java will need to be static as the main method is static.

```
****************
*  TIC TAC TOE  *
****************

P)lay or Q)uit: P

  0   1   2

  -|  -|  -   0
-----------
  -|  -|  -   1
-----------
  -|  -|  -   2

X Player select: 0 0

  0   1   2

  X|  -|  -   0
-----------
  -|  -|  -   1
-----------
  -|  -|  -   2

O Player select: 1 1
```

```
  0   1   2

  X|  -|  -   0
-----------
  -|  O|  -   1
-----------
  -|  -|  -   2

X Player select: 0 1

  0   1   2

  X|  -|  -   0
-----------
  X|  O|  -   1
-----------
  -|  -|  -   2

O Player select: 2 2

  0   1   2

  X|  -|  -   0
-----------
  X|  O|  -   1
-----------
  -|  -|  O   2

X Player select: 0 1
X Player select: 0 2

  0   1   2

  X|  -|  -   0
-----------
  X|  O|  -   1
-----------
  X|  -|  O   2

X WINS!!!

*****************
*  TIC TAC TOE  *
*****************

P)lay or Q)uit: Q
```

Notes:

1. The user can enter uppercase or lowercase P or Q.
2. The only error checking is making sure the user enters an uppercase or lowercase P or Q and that the user enters an empty location on the Tic Tac Toe board.
3. If the game is a tie, output "Game is a Tie"

4. If invalid input is entered, simply display a new prompt on the next line.
5. You should finish this portion of the assignment no later than noon on Wednesday and preferrably by Tuesday evening.

<u>Part II</u> Minesweeper (Should take 10-15 hours)

The game minesweeper is a single-player game with the object of clearing a minefield without detonating a mine. The purpose of this assignment is to write a complete Java version of minesweeper in a project called MinesweeperJava. The logic (and code) gained from this assignment will then be used to implement an Android version of minesweeper later in the course.

Details of the game can be found at:
http://en.wikipedia.org/wiki/Minesweeper_(video_game).

Details of our Java game:

1. Set a constant to determine the dimension of our grid. Set the dimension to 9 initially giving us a grid of 9x9 or 81 squares.
2. Ask the user to input a difficulty level (EASY is 0, MEDIUM is 1, and HARD is 2)
3. Set the number of mines based on the difficulty level:
    a. EASY (specified by 0) is the dimension of our grid or 9.
    b. MEDIUM (specified by 1) is the dimension of our grid plus 1 times 3 which is 12.
    c. HARD (specified by 2) is the dimension of our grid plus 2 times 3 which is 15.
4. Some details of the game:
    a. Bombs are placed in random positions on the grid and identified with a value of @.
    b. During a turn, the user selects a cell to reveal the cell's contents
        i. A cell that has adjacent bombs is the only cell revealed and a number indicating the number of adjacent bombs is displayed in the cell the next time the grid is output.
        ii. A cell that has no adjacent bombs is marked with a SPACE. If a cell has no adjacent bombs, then all adjacent cells (including the diagonals) are looked at. For each of the adjacent cells, then either i. or ii. applies. Yes, this is a recursive definition although you do not need recursion to solve the problem.
        iii. A cell selected by the user that has a bomb terminates the game

c. If all non-bomb cells are identified with a value other than the initial value (a . in the following example) before a cell with a bomb is selected, then the user wins; otherwise, a bomb was hit and the user loses.

Here is an example of your game play.

```
***********
Minesweeper
***********

Enter difficulty level
(0 = EASY, 1 = MEDIUM, 2 = HARD): 0

  0   1   2   3   4   5   6   7   8

  .|  .|  .|  .|  .|  .|  .|  .|  .   0
-----------------------------------
  .|  .|  .|  .|  .|  .|  .|  .|  @   1
-----------------------------------
  .|  .|  .|  .|  .|  .|  .|  .|  .   2
-----------------------------------
  .|  .|  .|  .|  .|  .|  .|  .|  .   3
-----------------------------------
  .|  .|  .|  .|  .|  .|  .|  .|  .   4
-----------------------------------
  @|  .|  .|  .|  .|  .|  .|  .|  .   5
-----------------------------------
  .|  .|  .|  @|  @|  @|  .|  .|  .   6
-----------------------------------
  .|  @|  .|  .|  .|  .|  .|  .|  .   7
-----------------------------------
  @|  @|  .|  .|  .|  @|  .|  .|  .   8


Enter X and Y Coordinate: 3 8

  0   1   2   3   4   5   6   7   8

  .|  .|  .|  .|  .|  .|  .|  .|  .   0
-----------------------------------
  .|  .|  .|  .|  .|  .|  .|  .|  @   1
-----------------------------------
  .|  .|  .|  .|  .|  .|  .|  .|  .   2
-----------------------------------
  .|  .|  .|  .|  .|  .|  .|  .|  .   3
-----------------------------------
  .|  .|  .|  .|  .|  .|  .|  .|  .   4
-----------------------------------
  @|  .|  .|  .|  .|  .|  .|  .|  .   5
-----------------------------------
  .|  .|  .|  @|  @|  @|  .|  .|  .   6
-----------------------------------
  .|  @|  3|  2|  4|  .|  .|  .|  .   7
-----------------------------------
  @|  @|  2|   |  1|  @|  .|  .|  .   8
```

```
Enter X and Y Coordinate: 0 0

  0   1   2   3   4   5   6   7   8

   |   |   |   |   |   |   |  1|  .   0
-----------------------------------
   |   |   |   |   |   |   |  1|  @   1
-----------------------------------
   |   |   |   |   |   |   |  1|  1   2
-----------------------------------
   |   |   |   |   |   |   |   |      3
-----------------------------------
  1|  1|   |   |   |   |   |   |      4
-----------------------------------
  @|  1|  1|  2|  3|  2|  1|   |      5
-----------------------------------
  .|  .|  .|  @|  @|  @|  1|   |      6
-----------------------------------
  .|  @|  3|  2|  4|  .|  2|   |      7
-----------------------------------
  @|  @|  2|   |  1|  @|  1|   |      8

Enter X and Y Coordinate: 5 8
Boooom!!! You lose.
```

If the player wins, then display the message "Congratulations. You win."

Notes:

1. Make sure the user enters 0, 1, or 2 for the difficulty level. If an improper selection is made, display the input message again and continue until a proper response is entered.
2. Make sure the user enters a valid position on the Minesweeper board. If an invalid position is entered, display the input message again and continue until a valid position is entered.

Goals for Assignment #1:

1. Write a Java application using multiple classes
2. Use packages to better organize all classes
3. Use good OOP techniques in designing your solution
4. Use the Java API which has a rich library of routines (e.g. Vector, Stack)
5. Use JUnit framework for testing classes

Specifics:

1. When an assignment (or portion of an assignment) is due, drop your entire workspace PUNetID with the completed project into the course drop folder on Turing.

2. Your code is to be written using the development tools specified in the syllabus.

3. If you come to me with a question regarding your solution, I will have you load your project onto a machine in the CS lab. I will not look at your code on your computer or on paper as it just takes me too long to get at the problem. Further, I want you to bring in your lecture notes in case I want you to look up something. Remember, I'm not just a tell you the answer guy. Make sure you understand how to use the developer tools and that you can run your program in Eclipse.

4. If you want help with a compiler error, you must be able to tell me exactly what statement you put in your code that caused the error and be able to isolate the error. If you have typed in a bunch of code and have not tested your code as you've gone along, I'm not going to help you sort out the mess. You've been warned!!

5. For Part I, you are to print out only PlayTicTacToe.java completely documented. For Part II, you are to print out the class with main followed by each class you create in the MinesweeperJava project in that order. You are to print out each .java file fully documented using the Java Coding Standards V1.1. Your printout is due to me by the time specified. If the time specified is after our last meeting of the day, then your printout is due at the beginning of the next class meeting or 6PM the next day (to Thomas or myself) whichever comes first.