**MineSweeper – The Game – Part 2**

**Lab Goal :** This lab was designed to teach you more about recursion, matrices, abstract classes, interfaces and inheritance.

**Lab Description :** Create a rudimentary game of Minesweeper. When a cell is clicked and it is in bounds and it is not a mine and it is not adjacent to any mines (count == 0) your program should reveal the cell as empty and continue checking adjacent cells (including diagonals). If the count is not zero it should reveal the cell as empty an reveal the number of mines adjacent to it and stop recusing.

In order to count the number of adjacent cells you will need to check each adjacent cell for a mine (including corners). There is no need to do this recursively since each cell has a set number of adjacent cells (8). You can do this elegantly using an array (int[] = {1,0,-1}) or less elegantly using a ton of ifs.

***algorithm help***

REVEALING CELLS

if ( r and c are in bounds and [r][c] is not a mine and [r][c] = zero )

mark spot as visited

8 recursive calls up down left right and corners

COUNTING MINES

for each cell

if an adjacent cell has a mine

increase count

set the count for that cell to the total

**Files Needed ::**

**Locatable.java**

**Cell.java**

**EmptyCell.java**

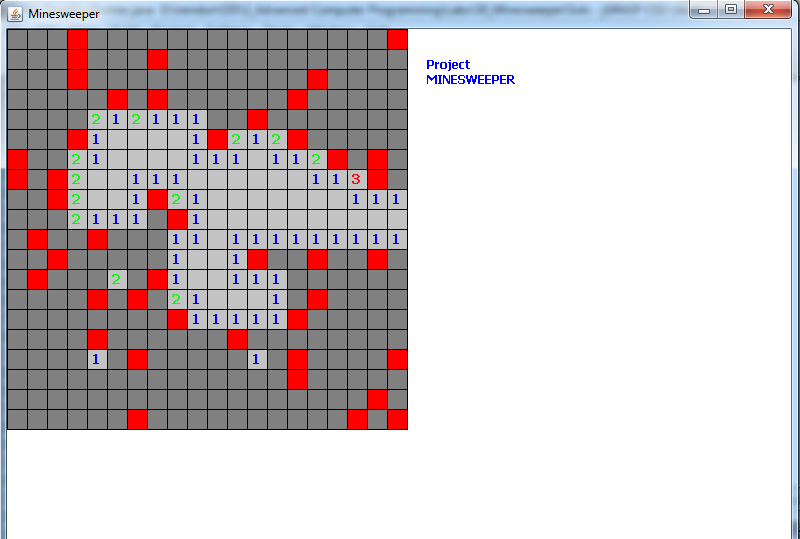
**MineCell.java**

**Grid.java**

**Minesweeper.java**

**GraphicsRunner.java**

**Sample Output :**



**This cell is adjacent to one mine cell**