NAMES: John Karuntzos / Dakota Krogmeier / Kristopher Hoce

Worksheet 24: Work with either ONE or TWO other students to complete this worksheet. You must work with at least one other student on this assignment, and should submit one paper. Be sure to describe your key idea at a high level.

1. Consider the game Minesweeper. If a cell that does not have any bombs in cells adjacent to it is selected, the contents of largest region surrounding that cell are revealed. The border of this region consists of cells **adjacent to bombs** (as an example, look at the start-of-class worksheets for this week). Develop a **recursive** algorithm to reveal the contents and border of the largest region surrounding a selected cell. Consider the grid representing the game to be a global parameter, and not part of the recursive call. Identify a cell by its coordinates, x and y, and suppose that each cell has a property reflecting either that the cell contains a bomb (-1) or the number of bombs that border the cell (0 – 8: for a border cell, this value must be positive), as well as a Boolean to indicate if the contents are revealed. See image. (6 points)

   KEY IDEA:

   If the cell is unopened then check If the selected cell is a mine or a number 1-8, then only open the individual cell. Otherwise open all the surrounding cells and repeat this process

   Algorithm RevealContents(*cell(x,y)*):

   Input: a cell in a Minesweeper grid (the grid is global), identified by its x- and y-coordinates

   Output: The grid is modified so that the contents of the largest bomb-free region surrounding the cell is revealed; reveal a particular cell using "reveal cell contents" as the instruction

   PROCESS:

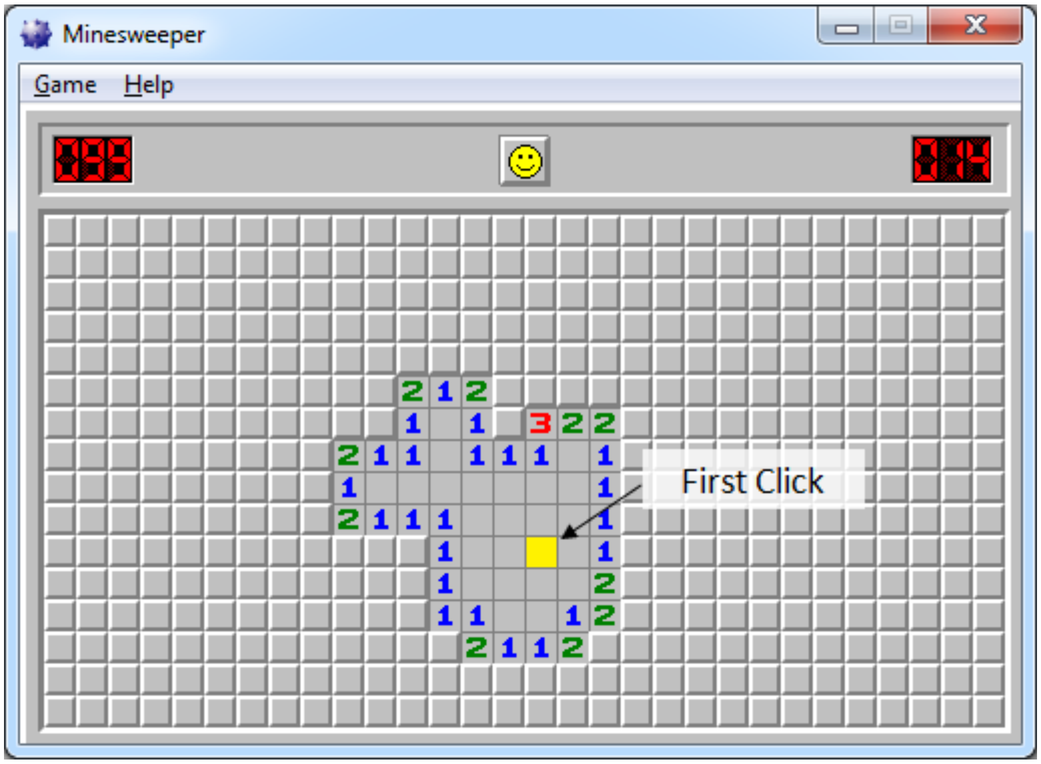
```
x <— cell.x
y <— cell.y

if grid[x][y].open = false:
    grid[x][y].open = true
    if grid[x][y].value = 0:
        neighbors <— getNeighbors(cell)
        for n in neighbors:
            RevealContents(n)
```

```
getNeighbors(cell(x,y)):
    x <— cell.x
    y <— cell.y
    neighbors <— []
    if y-1 >= 0:
        if x-1 >= 0:
            neighbors.add(grid[x-1][y-1])
        if x+1 < grid.length:
            neighbors.add(grid[x+1][y-1])
        neighbors.add(grid[x][y-1])

    if y+1 < grid.length:
        if x-1 >= 0:
            neighbors.add(grid[x-1][y+1])
        if x+1 < grid.length:
            neighbors.add(grid[x+1][y+1])
        neighbors.add(grid[x][y+1])
    if x-1 >= 0:
        neighbors.add(grid[x-1][y])
    if x+1 < grid.length:
        neighbors.add(grid[x+1][y])

    Return neighbors
```


Minesweeper
Game
Help
First Click