CS 100 Programming I
Project 3: Description

John C. Lusth

Revision Date: October 2, 2015

Preamble

You may develop your code anywhere, but you must ensure it runs correctly on a Linux distribution before submission.

Minesweeper!

> "They blowed up good, blowed up *real* good" – *The Farm Film Report* with Big Jim McBob and Billy Sol Hurok

Your task is to implement the classic game, *minesweeper*. The game consists of a board with the board divided into a grid. Throughout the grid, mines are scattered. At the beginning of the game, all squares of the grid are covered, obscuring the location of the mines. The object of the game is to uncover all the squares on the board that do not hold mines, avoiding uncovering those squares that do. Squares are uncovered one at a time.

When a square is uncovered, three things can happen. If the square holds a mine, the mine blows up and the game ends. If the square does not hold a mine, but is neighbors with a square that does, then the number of neighbors that hold mines is shown in the uncovered square. If there are no adjacent mines, then all neighboring squares are also uncovered. If any of the newly uncovered squares are not adjacent to mines, then their neighbors are uncovered, and so on. Each square has eight neighbors, unless it is on the edge of the board. In that case, it will have fewer neighbors.

Graphical representation of a board

You are to use Cheesy ASCII graphics (TM) to represent the board. Here is a possible, but much too boring, representation of a board with five rows and nine columns:

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

In this representation, a dash represents a covered square. Suppose one square with no adjacent mines is uncovered. In that case, the board might look like:

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

Be creative with your Cheesy ASCII drawings, especially when a mine is uncovered.

Human-to-program communication

The number of rows in grid, the number of columns in the grid, and the number of mines are passed as command line arguments. The program then randomly (or not) places the mines and prints out a representation of the board with all squares covered. It then goes into a loop, prompting for a row and column of the first/next square to be uncovered. For each square location entered by the user, the program displays the updated board if the square did not hold a mine or an impressive pyrotechnic display if it did.

Compliance Instructions

Your program should run like this, but look better:

```
$ python3 minesweeper.py 5 9 5

Here is the minefield:

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

Which square do you wish to uncover?
Enter a row and column: 1 7

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

Which square do you wish to uncover?
Enter a row and column:
```

Submission Instructions

Change to the directory containing your program and run the command:

```
    submit cs100 YYY project3
```

Replace YYY with your instructor's name.