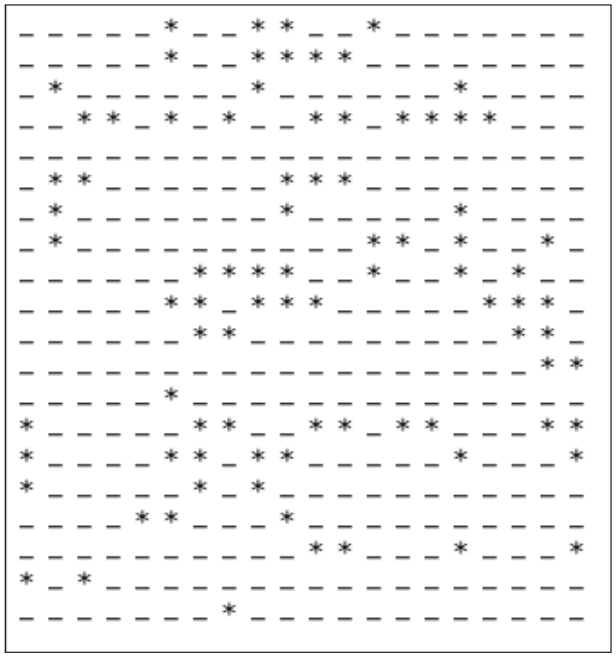
CS161: Introduction to Computer Science
Homework Assignment 9
Due: Monday 4/16 by 11:59pm

Conway's Game of Life

In this assignment, you'll be implementing *The Game of Life* – not the board game but the cellular automata formulated by mathematician John Conway in the 1970s.

The game is represented using a two-dimensional array of cells. Each cell can either be dead or alive. In the picture below, alive cells are represented using asterisks and dead cells are represented using underscores. Depending upon the state of its neighbors, a cell may either die or come to life at each generation.



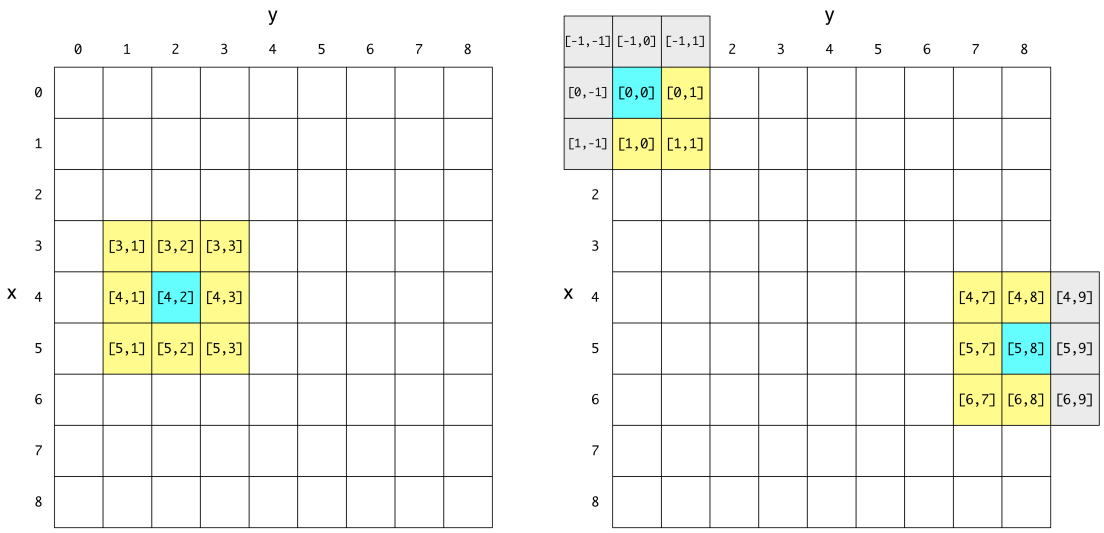
Rules of the Game

In the game of life, you begin with a random configuration of the board. That is, you should randomly initialize each cell to dead or alive. The game is then run for a number of iterations (i.e. "generations"). For each iteration, you determine whether a cell is dead or alive using the following rules:

- A dead cell with exactly three living neighbors becomes a live cell
- A live cell with exactly two or three living neighbors remains a live cell
- In all other cases, the cell dies or remains dead

A cell's neighbors are the cells that are horizontally, vertically, or diagonally adjacent. In other words, a cell's neighbors are the 8 other cells that it touches at edges or corners.

As the picture below shows, most cells have 8 neighbors. The blue cell at `(4,2)` in the first image has exactly 8 neighbors shown in yellow.



However, the cells along the border have fewer than 8 neighbors. For example, `(0,0)` only has 3 neighbors and `(5,8)` only has 5 neighbors. In these cases, we consider the board to be "wrapped around" so that every cell (even the ones on the edge) have 8 neighbors.

For example, consider the cell at `(5,8)`. It's neighbors are: `(4,7)`, `(4,8)`, `(4,0)`, `(5,7)`, `(5,0)`, `(6,7)`, `(6,8)`, `(6,0)`. Or, consider the cell at `(0,0)`. In a wrapped board, this cell has 8 neighbors: `(8,8)`, `(8,0)`, `(8,1)`, `(0,8)`, `(0,1)`, `(1,8)`, `(1,0)`, `(1,1)`.

Running the Game

You should prompt the user to enter the size of the grid – i.e., the number of rows and columns. After that, you should randomly generate and print a board. Prompt the user to hit enter to continue or "q" to quit. If the user hits enter, you should use the rules of the game to update the board and then print the new configuration. If you hold down the enter button, you'll notice that the printing happens so fast that it seems like an "animation". The next page is an example of a single iteration of my program.

Here are some helpful hints for this assignment:

- Again, you are free to have as many (or a few) classes as you want. The design of this program is up to you.
- You need to keep the grid *intact* and unchanged while you're determining which cells should die or come to life in the next generation. A good idea is to create a new grid for the next generation. After you're done filling in this new grid, you can then overwrite the old grid.
- Although it's random, if your board always stops changing after only 4-5 generations, something is probably wrong. My solution consistently either never converges to a steady state or takes over 20 generations to converge.

Style Guide

Before you submit your assignment, go through the checklist below and make sure your code conforms to the style guide.

Checklist

- ☐ All unused variables are deleted
- ☐ All instance variables are used in more than one method (if not, make them local)
- ☐ All instance variables are declared private
- ☐ All instance variables are initialized in the constructor
- ☐ Javadoc comment for all classes
- ☐ All methods have Javadoc comments (except for the `main` method)
- ☐ Good use of private methods
- ☐ Proper capitalization of variables (final and non-final variables), methods, and classes
- ☐ All numbers have been replaced with constants (i.e. a final variable)
- ☐ Use white space to separate different sections of your code
- ☐ Code is correctly indented to improve readability

See the "Style Guide" (under "Resources" on the course website) for more detailed information.

Submitting your homework assignment

You should submit your `hw9` folder with your all of your code via Moodle.

```
===== Welcome to Conway's Game of Life =====
Enter the size of the grid: 20

Starting board configuration:
* * _ _ * * * _ _ _ * _ * * * _ * _ _ *
_ _ _ * * _ * * * * _ * _ * * * _ * * *
* _ * _ _ * _ _ _ _ _ _ _ * * * * * _ *
* _ * * _ _ _ _ * _ * * * _ * * * _ * _
_ * _ * _ * * _ _ _ _ * * _ * * _ _ _ _
* _ * * _ * * * * _ * _ _ * * _ * * _ *
_ _ _ _ * * * * _ _ * _ * _ * _ * * * *
_ _ _ * * * * * * _ _ * _ * * _ _ _ * _ _
_ * * * * * * _ _ _ * _ _ _ * * * * * * _
_ _ * _ * _ * _ _ * _ _ _ * _ _ _ _ * *
* _ * * * * _ _ _ * * _ * _ * _ * _ _ *
* _ _ _ * _ * * * _ * * * _ * * * _ * *
_ * * _ * * _ * * * * * * _ _ * _ _ * _
* * * _ _ _ _ _ _ * _ * _ * * _ _ * _ _ _
_ _ * _ _ * _ * * * _ _ _ _ * * _ * * * *
* * * * * _ _ _ * _ * _ _ _ * * _ _ _ _
* * _ _ * * _ * _ _ * _ _ _ * * _ * _ *
_ * _ * _ * * _ _ _ * * * * _ * * _ * _
_ * _ _ _ _ _ _ _ _ _ * _ * * * * _ _ * *
_ * * * * _ _ _ _ * * _ * _ _ _ _ * _ _

Press <enter> to continue or "q" to quit:

_ * _ _ _ _ * _ _ _ _ _ _ _ _ _ * _ _ *
_ _ * * _ _ _ * * * * * * _ _ _ _ _ _ _ _
* _ * _ _ * * _ _ _ _ _ _ _ _ _ _ _ _ _ _
* _ _ * _ * * _ _ _ * _ _ _ _ _ _ _ * _
_ _ _ _ _ * _ _ * _ _ _ _ _ _ _ _ _ * _
* * * * * _ _ _ _ * * * _ _ _ _ _ _ _ _ *
* _ * _ _ _ _ _ _ _ * _ * _ * _ _ _ _ *
_ _ _ _ _ _ _ * * * * _ * _ _ _ _ _ _ *
_ * _ _ _ _ _ * * * * _ _ _ _ * * _ _ *
_ _ _ _ _ _ * _ * * _ _ * _ _ _ _ _ _ _
_ _ * _ _ _ _ _ _ _ _ _ * _ * _ * _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ * _ * _ * _
_ _ * _ * * _ _ _ _ _ _ _ _ _ _ _ _ * _
* _ _ _ * * _ _ _ _ _ _ * * _ _ * _ _ _
_ _ _ _ * _ _ * _ _ * * _ _ _ _ _ * * *
_ _ _ _ _ _ _ _ _ _ * _ _ * _ _ _ * _ _
_ _ _ _ _ _ _ * _ _ * _ * _ _ _ _ * * *
_ * _ _ _ * * _ _ * * _ _ _ _ _ _ _ _ _
_ * _ _ _ * _ _ _ _ _ _ _ _ _ * _ _ * *
_ _ _ * * _ _ _ _ * * _ _ _ _ _ * * _ _


Press <enter> to continue or "q" to quit:
```