Assignment 2

Overview:

We will implement Conway's game of life. Conway's game of life is a famous simulation. Yours will include a self-updating display.

Conway's game of life is simple. The simulation consists of a grid. Each grid is referred to as a cell. The simulation moves forward in discrete time steps. At each time step a cell is marked as either alive or dead according to the following rules:

- If the cell is alive, then it stays alive if it has either 2 or 3 live neighbors
- If the cell is dead, then it springs to life only in the case that it has 3 live neighbors

See this website for more information: playgameoflife.com/

Your code:

Your code will be a script written in Python. The script will contain, at least, two functions. One function will take the entire board representation, advance it one step, and return a new board state. The second function will take the entire board representation and plot it. A third will take a board representation and update the plot dynamically (update the plot data but do not destroy and recreate the figure). You can include as many supporting functions as you wish as long as these three functions are present so these portions may be viewed seperately.

In your plot, a living cell will be a blue x and a dead cell will be nothing. Your grid will be defined as a list of lists in your implementation. Your code will be able to handle different-sized grids (e.g. 50x50 or 200x200).

Your script will begin by offering the user to select from the following initial states:

- blinker
- glider gun
- random (cells are set to living with some percentage of probability).
- another pattern which interests you

Your script will then ask for a grid size (grid will always be square so just ask for a single number). Finally, your script will ask the user how many time steps to run the simulation for.

When you script has finished, some statistics will be displayed. You will show the start date/time of the simulation, the end date/time of the simulation, the number of milliseconds elapsed, the number of frames processed, the total number of living cells during processing (just sum up the number of living cells across all timesteps of the simulation), and the number of cells that died across all timesteps of the simulation.

Grading

Successful completion of the simulation: 70%

Successful completion of the visualization: 25%

Groups ranked by number of frames displayed and simulation time steps completed in an allotted amount of time: 5%

The slowest project will receive at most a 95%; the fastest will receive, at most, a 100%. Failure to complete the visualization will disqualify your project from receiving these points. If no TA is found for this course, these 5% points will be added to the visualization portion of the project.