Benjamin Hobbs
CS-165
2/20/2014

PROJECT 5
THE GAME OF LIFE
DESIGN DOCUMENT

REQUIREMENTS

This program will run a version of Conway's Game of Life, specifically it will take a seed and then print the next n iterations of that seed using the following rules:

1. Any live cell with < 2 live neighbors dies
2. Any live cell with 2 or 3 live neighbors lives on
3. Any live cell with > 3 live neighbors dies
4. Any dead cell with exactly 3 live neighbors becomes a live cell

Other requirements include:

World size of at least 80 x 22 cells,

Some kind of user input for the seed ("option to start with more than one arrangement"),

User must be able to rerun the game of life with different seeds,

User must be able to exit during runtime (static seed),

System must be able to handle its edges (ie wrap around).

METHODS:

First a basic design:

There will be a class called GameOfLife, which will contain the following:

-2 Vectors: 1 for the current generation, and 1 for the next generation
Probably vectors of vectors of chars

-A constructor that takes x and y size as args and builds the above array

-A member function that gets the list of seed files.

-A member function that takes a file and uses it as a seed for the first generation (see below)

-A member function that takes the first vector and uses the rules to create the next generation in the other vector

-A copy function that sets the first vector equal to the second vector (this will be used to start the next generation)

-A member function that runs the above three functions and returns a vector to be used in printing outside the class.

I will use the nCurses Library to create visually stunning (or at least smooth and clean) output in the terminal.

-This will include a menu that allows the following:

-Running the game with a random seed

-Running the game with preset seeds

-These will be stored in a folder ("seeds/")

-Each preset will stored in a text file, and formatted as such:
x y
x y
etc

-These presets will be created using MS Excel and a vba script to print coordinates.

-Looking at the settings (Probably just speed, screen size, and the cmd line arg to run the program with options)

-Exit the program (clean up nCurses)

METHODS (CONTINUED):

Runtime

-The seed will run for approximately 5,000 cycles.

-The program will automatically run at 130 columns and 50 row

-if the terminal is too small, the program will prompt the user to resize or use the [-s] option to make the game run at 80 columns by 22 rows.

-The system will take a command line argument to determine how fast each cycle will run. Default will be 250,000 microseconds, or .25 seconds.

-nCurses will also allow for exit during runtime

-exit will return to the menu, which will be printed over the last iteration of the running seed (because it will look cool that way, I think).

Command Line:

./gol [-s] [–w int]

[-s] will set the program size to 80 columns by 22 rows, for little monitors

[-w int] sets the wait time between iterations. In microseconds, def. = 75,000.

NOTES:

To deal with the edges of the board I will use the mod function to create a wrap around effect for the getFriends function (the function that checks for living cells in close proximity). As such I'm thinking its gonna look something like this:

Where X and Y are the position in the 2D Vector

X-1 , Y-1	X , Y-1	X+1 , Y-1
X-1 , Y	X, Y	X+1 , Y
X-1 , Y+1	X , Y+1	X+1 , Y+1

So with the mod functions thrown in it will look like this:

(X or Y) - 1	(X + max(X)-1)%X
(X or Y)	X
(X or Y) +1	(X+1)%max(X)

REFLECTIONS:

In short, the actual mechanics of creating the game of life were not particularly challenging and did not take that much time to get up and working. The methods outlined above worked with minor adjustments, and the code turned out looking pretty good. What wasn't simple, easy, or pretty was getting the nCurses library to print out the game of life and related menus. Interestingly enough the issue was not the ease of use, but rather the shocking level of detail contained in the nCurses library and the variety of ways that it can be used. In reflection, I wish I had learned how to use this library earlier in the term. I think that the ability to write anywhere on the screen would have been nice for the menu driven projects we have been cranking out.