CS& 131

Project 4: Group Project: "The Game of Life"

Names:_____

The Game of Life was invented by Cambridge mathematician **John H. Conway** to model the process of birth, survival and death. By the rules of the game, individuals require others to survive, but die as the result of overcrowding.

For our world, we will use two 2-dimensional arrays of chars *(minimum size: 20 x 40)*. One array will be the current world, the other will hold the next generation. Each cell in the array holds either a "*" (a live organism) or a blank--" "--(no organism).

The game starts out with an initial generation, which consists of any mix of stars and blanks. This original array will be read in from a text file specified by the user. This file can be created using any text editor, such as notepad (*do not use a word processor!*).

Three rules govern the transition from one generation to the next:

a. **Birth Rule:** an organism is born into an empty cell that has *exactly* three living neighbors.
b. **Survival Rule:** An organism survives from one generation to the next if it has either 2 or 3 living neighbor.
c. **Death Rule:** An organism dies from loneliness if it has fewer than 2 neighbors. It dies from overcrowding if it has 4 or more neighbors.

	x	x	x
	x	*	x
	x	x	x

A neighbor of a cell is any of the 8 possible cells that touch it.
In the array to the left, the cells labeled "x" are all neighbors of the cell marked "*":
Be careful when looking at the cells at the edge of the world.
Do not go out of bounds!

In your version of the game, you will display the world one generation at a time, clearing the screen between each generation. The user will press <*Enter*> to continue, and some other key to quit.

Make Sure you test fully! Results depend on the initial colony. Some possible results:

- The colony quickly dies out
- The colony rapidly expands *(stopped only by the edge of the world).*
- the colony reaches a set stasis
- the colony reaches an oscillating stasis *(moves back and forth between 2 or more configurations)*

The program, of course, should use top-down design, and must utilize modularly designed functions.

Some suggestions for functions you should utilize:

- a function to read the file into the grid
- a function to count the neighbors of any one cell
- a function to calculate the next generation (*which should call the previous function*)
- a function to print out one generation
- a function to move the next generation to the current one (*unless you swap pointers to the two generations*)

This program is to be created by a *group* process. A *group* should be comprised of 3 - 5 students. When programming as a group, it is even more important that the project be fully **planned** *in advance*. Plan out, in advance, what each function needs as prerequisite conditions, as parameters, what the post-conditions are, and what the function returns. Make sure you test for all 4 possible outcomes, as well as checking that consecutive generations follow the rules. You may use the stand-alone java program **The Game of Life** for comparisons in your testing.
Be user friendly. Make the directions easy to understand. Make the output understandable and usable.

Deliverables: *(one set per group)* *submit **both** physical and electronic*

Physical:

The Project should be turned in inside a clear plastic file folder. This folder should have a simple flap to hold paper in place--NO buttons, strings, Velcro, etc. Pages should be in order, not stapled.

- Assignment Sheet *(this page),* with your **name** on it, as a cover sheet.
- Printed ***Source Code*** with Comments *(including heading blocks. Describe parameters, no line wrapping,)*

Electronic:

- All ***.h, .c, .exe***(Release Version) and all test input and output files
- ***Sample*** Output (as .rtf -- run the program, copy the window using <ALT|PRTSCN>, paste into Paint, invert colors (<Ctrl|Shift|I>), copy, open Wordpad, save.) and Input and Output files.
- A simple ***test plan*** including explanations of any discrepancies and reasons for each test. Show actual input and ALL values output as well as ALL expected output *(make sure to show tests for all four possible end states).* Save as .xls, xlsx, .doc or .docx file
- Zip all of the above files together. Do not use rar or any archive format other than *zip*. Rename the file: "<YourGroupName>_p4.zip".

Submit this single ***zip*** file *(one per group)* by going to **Canvas**, select this class, select the **Assignment** tab on the left, select the ***Assignment 4*** select the **submission** tab at the top, find the file, and **Submit**.

Due

- **A:** Friday**, December 4**, 2014, 9:30 a.m. (*beginning of lab*)
- **B:** Thursday, **December 3**, 2014, 4:30 p.m. (*beginning of lab*)

Informal group presentations will be given at this time