Assignment 1 – Program Design and Testing

Due Date: Sunday, 6 July, 23:59:59

Implement Conway's *Game of Life*

Conway's *Game of Life* is a standard example of a cellular automaton. This means that you have an array or matrix of cells. Each turn or generation the value of each cell may change based upon its neighbors. There is a more complete description in the book, Ch 5, p.234, #12. You can also find much information about the game on the Internet. Much information…..

You will design, implement, and test a program that runs the game of life. There are significant initial questions that you must address in your design. For example, we have not covered file operations yet so you cannot read the starting position from a file. Since you may be developing your program through SSH we will limit the "world" to 80 x 22 cells. That's a lot of characters to type in from the keyboard. There are numerous ways to address this problem that may involve hard coding arrays or using the keyboard. That's why it's called design! :-) But you need to give the user some option to start with more than one arrangement in the grid. Remember that a starting configuration will typically consist of less than a dozen cells.

You must also provide some method for the user to have it run a different number of times. Optionally, you may also give them the ability to abort. A starting position may end up with all cells going into some repetitive or static pattern.

Probably the most difficult part of the design is what you will need to do to handle the cells on the edges. Remember this is a window on an infinite grid. You just can't stop the patterns at the edge as that may change the behavior. You design must specify how you will address this problem. Specifically, any pattern should not change as the object moves out of the visible grid and the object should slide out of view.

You will create your design document and submit it BEFORE you start coding. This is up to your schedule but you have 2 weeks for this lab. The TA will grade, in part on how well your implementation matched your design. Remember, in your reflections document you can explain how you had to change the design because your first idea, just didn't work. Just explain what you learned.

You will also write and apply a test plan. Remember, while it's nice to have shooters and other stable configurations you are testing the operation of your software. In this case

you will have “boundary” conditions to test, literally. There are virtual boundaries too. There are several rules that take into account the contents of the neighboring cells. Testing the boundary conditions is not just putting in a stock pattern and verifying that it works. This may be a case where going through your code one generation at a time would be useful. :-)

By the due date, you will submit your program, the test plan, and the reflections document, which will discuss how well your design worked, and the results of the testing.

HINTS-

1. I recommend that you start with a hard coded starting position. That will save time in debugging and also aid debugging since you know it’s the same input each time through. After your code appears to be working correctly, change it to whatever input scheme you want to use.
2. On the Internet you will also find many descriptions and examples of stable patterns. Definitely take advantage of that for your test plan!
3. The actual coding shouldn’t be hard. There are only 4 rules (on p. 235) that you apply to each cell in your array.
4. For cells on the edges you need to handle the case that you are displaying a subset of an infinite grid (according to the original rules). Get your program working first, ignoring the edge condition. Once you know your code is working then implement the edge calculation. You may need to have some ghost columns and rows. If you just stop at 80 columns figures may not behave correctly when the cells they contain disappear. You can continue calculating the movement, but just not display some of the cells. With this approach you will need to adjust the row and column indices that would change your display code.
5. As one website stated, the *Game of Life* is one of the most programmed games in the world. Be very careful about borrowing any code, or ideas you see in someone else’s code. As always, any code submitted must be yours and yours alone.