Submit Assignment

Final Project: Game of Life (With Classes)

Due Friday by 11:59pm **Points** 100 **Submitting** a text entry box, a website url, a media recording, or a file upload
Available until Dec 9 at 11:59pm

Game of Life

With a touch of class

Work in progress

Design the Game of Life (Chapter 7, Programming Project 13) using classes. You will look up the rules of the game from the textbook, but ALL implementation detail listed here MUST be followed.

Use **SFML** for your Graphical User Interface.

const int MAX = 80; This is the size of the grid.

You will need a Grid class:

private member variable: A two dimensional array (MAX by MAX) of integers that will be used to store the live cells.

You will need at least the following functions:

Grid( ) Constructor that will initialize your private member variable.

SetAlive(i, j) and Kill(i, j) that will be used as mutators for the class.

IsAlive(i, j) is a boolean function that will tell you if a cell is alive or dead.

CountNeighbors(i, j) is an integer function that will tell you the number of alive cells for a given cell.

PopulateRandom(r) is a function that will add r live cells to your grid in random locations.

Clear() Will clear the grid.

Save(char fileName[]) will save the current configuration of the grid into a text file.

Load(char fileName[]) will load the saved configuration from a text file.

Step( ) is a function that will generate the next generation on the grid.

Show( ) will print the grid on the screen.

The user will be given the following options to control the program:

Clear: To clear the grid

Random: to place 50 random live cells on the grid

Step: To generate the next generation.

Go: To generate generation after generation without stopping for input.

Save: To Save current configuration.

Load: To load a saved configuration.

A text panel on the right side of the grid will give information about the number of live cells, the number of generations since the start of the program, name of the fie the configuration is being saved, etc.

Once you have implemented these features, you might want to start working on giving the user the ability to turn individual cells on and off and other features like speed, user selectable file names, etc.

You will submit:

- FULLY COMMENTED code.

- a header file named notes.h that explains in detail the features of your program and how they were implemented. What things you had to consider. What math, physics you had to use. Everything a reviewer should know when they are looking at your program. This is not an easy task and is very time consuming. Do not rush this!

- a video (screen capture) where you play your project. Talk about what it does. Once again, go over the challenges you had to overcome and how you did that. After a person listens to your video, they should have a feeling they understand your program.

Upload the link to Youtube and paste the link in the assignment page. You can try to upload a copy to Canvas, but I don't promise this will work.