Programming Assignment #6

Due Nov 12 by 3pm **Points** 100 **Submitting** a file upload
Available Nov 5 at 7pm - Dec 5 at 12pm 30 days

This assignment was locked Dec 5 at 12pm.

Living Life to the Fullest

Implement Conway's Game of Life using the information on this page: **[1110] The Ingredients of Life**

- Use the algorithm presented in the modules.
- Use a 60 by 40 cell grid with "static" borders.
- The game should initialize with a random set of live cells (20% coverage).
- The game should end when the letter "Q" is pressed.
- The game should re-initialize if the letter "R" is pressed.
- The game should fill up every cell on the grid if the letter "F" is pressed.

Create your application as a Console application that prints generation after generation in a continuous scrolling manner. You will probably need to have the program sleep for ~50 milliseconds after each generation in order to ensure a smooth animation effect.

Use blank spaces and asterisks ("*") to display the cells in your grid.

NOTE FOR WINDOWS USERS: DO NOT USE Console.Write() to display each asterisk or blank space. Console.Write() is _very_ slow in Windows. Instead, for each row, add your asterisks or spaces to a string and then, when you reach the end of the row, using Console.WriteLine() to display that string. See [1110] The Ingredients of Life for an example.

Extra Credit:

- If the letter "Y" is pressed, the game should start adding "Cosmic Rays" randomly to the grid. When "cosmic rays" are enabled, after each new generation is calculated but before it is displayed, you should add a 1-in-20 chance that a randomly selected cell in that new generation "magically" becomes alive. Pressing the letter "Y" a second time should disable cosmic rays.
- If the number "5" is pressed, the game should clear the grid and add an r-pentomino pattern in the center.

Note: This program doesn't need any of the OO stuff we studied recently. Just use the "ingredients" mentioned in the Modules.

Remember: Make your code as readable as possible and don't use stuff we haven't discussed in class such as Lambdas, Delegates, Events or LINQ.

My Grading Guidelines:

30% Does the program compile and run?
50% Does the program generate correct results?
10% Is the program readable / maintainable / flexible?
10% Does the program follow our C# style guide rules?