

John Conway's Game of Life  
– as implement using the C programming language

Shrawan Sreekumar : el20ss : 201398821

University of Leeds

Author Note

This is an initial planning report, created in tandem with attaining credits for Module 1921 :

Programming Project.

### ***Project Design Details***

For this coursework, I plan to implement J.Conway's Game of Life, to the best of abilities, using the C language. The project involves recreating an isolated environment where "cells" exist. The cells are subject to the rules of the game, whereby which they either live, die or reproduce to populate a finite space. The state of each "cell" is recalculated based on the same rules, which gives rise to a new "generation" of cells during each iteration. The original Game of Life has many different implementations that are already present. While most of them are indefinitely scalable, the project I am going to create a variation of the Game whose parameters are finitely defined at the start of the program (read from a file). I will also attempt to utilize SDL or a similar C-compatible graphics library to display the final program.

#### **Module Abstract**

- Main file : loads the state of the environment, initiates the game and saves the state of the game at the end of the process.
- Display file : utilize graphic functions to display the states of the cells as the process iterates through each time step.
- Interface file : handles the menu driven interface that the user interacts with.
- Logic file : the main crux of the program, where the calculations which determine the state of each "cell" is held. **This module maybe split into 2 files if complexity requires it.**

### ***Test Plan***

I plan to unit test my program using the UNITY framework. This will allow me to conduct tests and monitor how my program reacts to a variety of different inputs and situations. This involves creating test cases for each function where the return value cannot be predicted for certain depending on a variety of different situations. Once these tests are created, the program will be run using the test script to verify it can handle the different kinds of errors that may emerge during user interaction. While I may attempt to cover most error cases, it has to be appreciated that not all cases may be visible at first glance to be testable.

#### **List of Functions and their test methodology:**

- `int load_state(FILE *file)`
  - expected return value : 0 . Successfully loads the previously saved state of the program into program memory
  - error code : -1
  - Checked exception : N/A
  - Assertion : the file pointer “file” is not null.
- `int save_state(FILE *file)`
  - expected return value : 0 . Successfully saves the current state of the environment to an output file
  - error code : -1
  - Checked exception : N/A
  - Assertion : the file pointer “file” is not null.

- void main()
  - expected return value : null. On successful competition program ends.
  - error code : N/A
  - Checked exception : N/A
  - Assertion : N/A
  
- int calculate\_state(\*\*\*)
  - expected return value : 0 . Successfully calculates the new state of a cell based on the 9 cells that surround it.
  - error code : -1
  - Checked exception : if cell is an edge cell, assume that cells beyond the pre-determined region is dead.
  - Assertion : cell position is positive.
  
- int update\_state(\*\*\*)
  - expected return value : 0 . Successfully updates the main array with new state of cell.
  - error code : -1
  - Checked exception : N/A
  - Assertion : the cell is in a valid state and has a valid position.
  
- void interface(\*\*\*)
  - expected return value : N/A. Breaks when program successfully ends via user choice

- error code : N/A. Displays error if interface fails and closes application
- Checked exception : User choice is valid
- Assertion : choice is always positive
- `int display_state (***)`
  - expected return value : 0 . Successfully displays the evolving generation of cells till user decides to halt the program.
  - error code : -1. Displays error if the cell states cannot be displayed.
  - Checked exception : Presence of required SDL or similar package definitions and variables required to display the cell states.
  - Assertion : N/A
- `int check_vertical (int row, int col)`
  - expected return value : Count( 1- 2 ) . Successfully counts presence of neighbor 1 step to the top and bottom of current cell
  - Checked exception : If no neighbors are present, return 0
  - Assertion : Count is positive
- `int check_horizontal (int row, int col)`
  - expected return value : Count( 1- 2 ) . Successfully counts presence of neighbor 1 step to the left and right of current cell
  - Checked exception : If no neighbors are present, return 0
  - Assertion : Count is positive
- `int check_diagonal (int row, int col)`
  - expected return value : Count( 1- 2 ) . Successfully counts presence of neighbor 1 step to the top-right and bottom-left of current cell

- Checked exception : If no neighbors are present, return 0
- Assertion : Count is positive
- int check\_antidiagonal (int row, int col)
  - expected return value : Count( 1- 2 ) . Successfully counts presence of neighbor 1 step to the top-left and bottom-right of current cell
  - Checked exception : If no neighbors are present, return 0
  - Assertion : Count is positive

(\*\*\*) : To be determined when the program is being developed

The functions and their tests stated above are the most crucial and core functions of the application. More functions and tests may be added to optimize and improve code handling during development.

### ***Project Schedule***

<b>Week</b>	<b>Task</b>
<b>14/03</b>	Creating planning document
<b>21/03</b>	Finalizing planning document and graphics package research
<b>28/03</b>	Create test code and begin project coding
<b>04/04</b>	Continue with coding
<b>11/04</b>	Ideally should complete first working implementation
<b>18/04</b>	Debug and improvement/ refactoring
<b>25/04</b>	Write project report
<b>02/05</b>	Contingency

### ***Reference***

<b>Description</b>	<b>Date Referred</b>	<b>Link</b>
Practical Simulation of Game of Life	21st March 2021	<a href="https://playgameoflife.com">https://playgameoflife.com</a>

