**The Game of Life**

**Yipeng FU**

**2019110059**

**Summary**

The project I chose is the game of life, which is a two-dimensional grid game. The game of life, also known as Conway game of life, is invented by the British mathematician Conway. The zero-player game takes place in a two-dimensional rectangular world, which is made up of many small squares of equal size. Each square is inhabited by a cell that interacts with the surrounding eight cells centered around itself. Every cell has two states: alive or dead. When the current cell is alive, when the number of living cells around is less than 2, the cell dies due to under population; When there are two or three living cells around, the cell stays alive; When there are more than three living cells around, the cell dies due to overpopulation. Conversely, when a cell is currently dead, it becomes alive when there are three living cells around it (simulated reproduction). On this basis, my project implementation should be divided into two parts, the logical part and the graphical part. In the logic part, I'll set up a checkerboard and put cells in each of the boxes. Then through the judge of the number of cells around the cell, to determine its state, while constantly updating the board, to achieve the effect of iteration. In the graph part, the board is established through SDL. The next step is to establish the shape and size of the cell. Finally the erection of the game interface is completed.

So the main modules are two parts:

1.Logic Part

Complete the game logic, which is to determine the number of cells around each cell.

2.Graph Part

Complete the graphics section, build the game window and the shape of cell.

**Test plan**

The general idea of this test plan is to test each function in a different way to ensure that the entire program runs properly. At the same time, special cases should be considered as much as possible to ensure the applicability of the program when the program runs normally. Perform a self-assessment at the completion of each release, and iterate on this basis to achieve the goal of completing a proper process.

**Function**: int Read\_Map ();

Expected behaviour:

• Reads the initial map from a file

• If the function runs improperly, return -1

• If the function runs properly, return 0

**Function**: void Print\_Map(int Map[X][Y]);

Expected behaviour:

• Print out the map

Assertions:

• The "Map" is a 2-d array

Test cases:

C1:

*Input:*

A normal 2-d array

*Expected results*:

Print the map properly

C2:

*Input:*

A improper array

*Expected results*:

Print something wrong

**Function**: void Init\_ShowMap(int Map[X][Y]);

Expected behaviour:

• Initializes the map to display

Assertions:

• The "Map" is a 2-d array

• If the function Read\_map run properly, then this function can run properly either

Test cases:

C1:

*Input:*

A normal 2-d array

*Expected results*:

Initialize the map properly

**Function**: void Renew\_Map();

Expected behaviour:

• Call the function Calculate, and update the map based on the result of the function

Assertions:

• Each cell is traversed to update the look of the map

**Function**: void Calculate(int Map[X][Y]);

Expected behaviour:

• Based on the incoming map, traverse and calculate the situation of each cell, while generating a new map

• Consider the special cases involving four boundaries first, and if not, calculate the cells around the specific cell

Assertions:

• The "Map" is a 2-d array

Test cases:

C1:

*Input:*

A normal 2-d array

*Expected results*:

Initialize the map properly

**Schedule**

|  |  |
| --- | --- |
| Week5 | Read the assignment documents carefully and make a preliminary judgment |
| Week6 | Think carefully and decide which project to choose |
| Week7 | Start make some preparations (including learning the relevant knowledge and downloading what need to install) |
| Week8 | Start doing coursework and complete the basic requirements |
| Week9 | Check specific requirements, refine and modify the work, and submit Planning Report |
| Week10 | Do the last check and submit the code |