CSE2031

Lab 6
Winter 2016

In this lab, you will implement The game of life, for clarification see https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life
The game of life describes a universe that is divided into cells, every cell is either *alive* or *dead*. The universe is infinite and expands in 2-D space. Every cell in the universe is either alive or dead. See the above link for explanation.

The Game of Life

In the original game of life, the universe size is infinite. For this lab, we assume that the universe size is finite and can be descried by 2-D array. The size is an input parameters, so is the initial state.

The game of life is a 0-player game, the game behavior is determined by the initial state, no input is required to play.

The universe (2-D array) evolves in generations. The next generation state is completely depends on the current generation state. Generation 0 is the initial state and is an input to your program.

Rules:

- The state of any cell in the next generation is determined by the state of the that cell neighbors in the current generation.
- If a live cell has less that 2 alive neighbors in the current generation, it dies in the next generation (under population).
- If a live cell has more than 3 live neighbors in the current generation, it dies in the next generation (over population).
- If a live cell has 2 or 3 live neighbors in the current generation, it continues to live in the next generation.
- If a dead cell in the current generation has exactly 3 live cells, it becomes live in the next generation (reproduction).

Some patterns repeat itself after a specific period, some patterns may move to the left/right in a continuous way, some patterns dies off very quickly and every cell is dead in a small number of generations. After you complete you code, you may play around with the initial pattern to produce different scenarios.

C code

Specifications

The input to the program is as follows
two integers, r and c to indicate the size of the array (rows and columns)
one integer that determine the number of generations to calculate and display.
r lines each contains *c* integers with values of either 1 or zero to describe the initial state of the game.
The output is a series of 2-D arrays displays where the game is displayed as 2-D array with every cell is either 1 or 0, where 1 means the cell is alive and 0 means the cell is dead.
A separator of 5 dashes (minus sign) followed by a new line after each display.
For example, if the input is

```
5 5
3
0 0 0 0 0
0 0 0 0 0
0 1 1 1 0
0 0 0 0 0
0 0 0 0 0
```

The output should be

```
0 0 0 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 0 0 0
-----
0 0 0 0 0
0 0 0 0 0
0 1 1 1 0
0 0 0 0 0
0 0 0 0 0
-----
0 0 0 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 0 0 0
-----
```

Exercise

Will be populated soon