AMA - GPU (TD 1)

Jonàs Martínez

October 15, 2024

Conway's Game of Life (in parallel)

A cellular automaton is a collection of "colored" cells on a grid of specified shape that evolves through a number of discrete time steps according to a set of rules based on the states of neighboring cells. The rules are then applied iteratively for as many time steps as desired.

We consider a cellular automaton that operates on a 2D regular grid of square cells called the "Conway's Game of Life". The grid has size $W \times H$. Each cell has one of two possible states: alive or dead. Every cell interacts with its eight neighbors, which are the cells that are horizontally, vertically, or diagonally adjacent. At each time step, the following transitions occur:

- Any live cell with two or three live neighbors survives.
- Any dead cell with three live neighbors becomes a live cell.
- All other live cells die in the next generation. Similarly, all other dead cells stay dead.

Consider that the initial state of the grid is already given. The first generation is created by applying the above rules simultaneously to every cell in the seed, live or dead; births and deaths occur simultaneously. Each generation is a pure function of the preceding one. The rules continue to be applied repeatedly to create further generations. Note that the computation of each generation can be done in parallel.

1) Implement in OpenCL the Game of Life: a function that computes the state of the grid after evolving N generations. See the next page for some input and expected output examples. For each generation, add an option to plot the grid's state in the terminal. Hence, it becomes easier to visualize the Game of Life.

2) Create an optional parameter to enable a periodic grid: there are no boundaries; when a cell passes through one side, it re-appears on the opposite side.

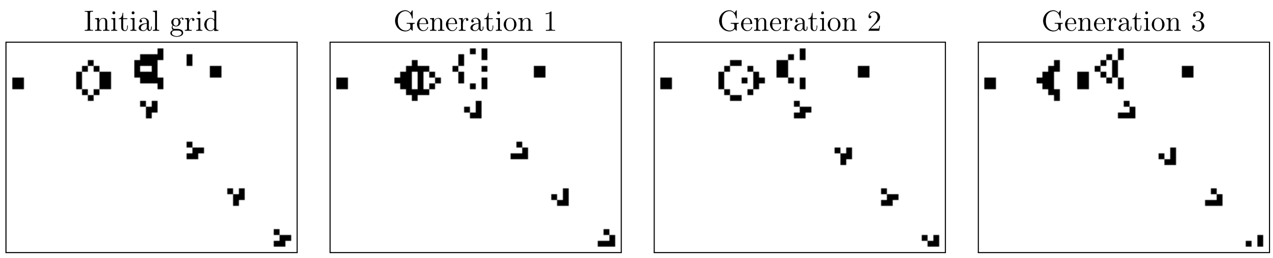

Figure 1: Three generations evolved from a given initial grid.

Examples

For more examples and information, see the following link.

a) Blinker

```
W=5     H=4     generation=0     dead X     alive O
X X X X X
X X X X X
X O O O X
X X X X X
W=5     H=4     generation=1     dead X     alive O
X X X X X
X X O X X
X X O X X
X X O X X
W=5     H=4     generation=2     dead X     alive O
X X X X X
X X X X X
X O O O X
X X X X X
```

b) Glider (periodic boundary)

```
W=5     H=5     generation=0     dead X     alive O
X X X X X
X X O X X
X X X O X
X O O O X
X X X X X
W=5     H=5     generation=1     dead X     alive O
X X X X X
X X X X X
X O X O X
X X O O X
X X O X X
W=5     H=5     generation=2     dead X     alive O
X X X X X
X X X X X
X X X O X
X O X O X
X X O O X
W=5     H=5     generation=3     dead X     alive O
X X X X X
X X X X X
X X O X X
X X X O O
X X O O X
W=5     H=5     generation=4     dead X     alive O
X X X X X
X X X X X
X X X O X
X X X X O
X X O O O
W=5     H=5     generation=5     dead X     alive O
X X X O X
X X X X X
X X X X X
X X O X O
X X X O O
(...)
W=5     H=5     generation=20     dead X     alive O
X X X X X
X X O X X
X X X O X
X O O O X
X X X X X
```