

## Programming Assignment 6: Game of Life

Due Date: October 19 (Thursday) at 11:59 pm

Note: This programming assignment is required for ALL students.

### 1. Objective

The purpose of this programming assignment is to simulate the Game of Life. The universe of the Game of Life is a 2-D grid in which each cell has two possible states, alive (1) or dead (0). Every cell interacts with its eight neighbors, which are the cells that are horizontally, vertically, or diagonally adjacent. You are going to simulate the universe for 100 generations. At each generation in time, the following transitions occur:

- Any live cell with fewer than two live neighbors dies, as if caused by under-population.
- Any live cell with two or three live neighbors lives on to the next generation.
- Any live cell with more than three live neighbors dies, as if by over-population.
- Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction; otherwise, stays dead in the next generation.

### 2. Procedure

**Step 0:** After executing `gpu-node4me.sh`, do the following to load `gcc/8.3.1`

```
module remove gcc/8.4.0
module load gcc/8.3.1
```

**Step 1:** Download the Programming Assignment 6 materials from blackboard to your local folder at Karpinski. Unzip it.

```
unzip p6-game-of-life.assignment.zip
```

**Step 2:** Edit `main.cu` and `kernel.cu` to implement host and device kernel code for the Game of Life.

- The allocated global memory space is of twice the size of the universe of the game of life.
- Half of the global memory space is for input and the other half is for output in the same iteration of simulation.
- The program will toggle between the two halves of the global memory for input and output.

**Step 3:** Compile and test your code.

```
make
./gameoflife          # Uses default size of the universe
./gameoflife m        # Universe: m × m
./gameoflife m n      # Universe: m × n
```

**Step 4:** Submit your assignment. You should only submit the following files:

- `kernel.cu`
- `main.cu`

Compress the files and name them after your last name like so:

```
tar -cvf p6_<your last name>.tar kernel.cu main.cu
```

If the last name is alan, the file name is `p6_alan.tar`

Submit the tar file in blackboard.

### 3. Grading:

Your submission will be graded based on the following criteria.

- Functionality: 100 points
  - Correct code and output results
  - Correct handling of boundary cases
  - Use shared memory for performance improvement
  - Check return values of CUDA APIs