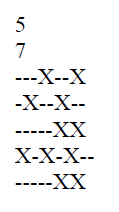
Game of Life

Classes

* Game mode class
  + Classic mode
    - All cells outside grid are considered empty
  + Doughnut mode
    - The grid has the same grid to the right, left, bottom, and top.
  + Mirror mode
    - Each edge cell bounces off
* Map class
  + Loading in map
    - Prompt map file (see configuration to the right)
    - Prompt file path
  + Randomly generated map
    - Prompt dimension for world
    - Prompt decimal value 0 > N <= 1. Represents initial population density of world
    - Randomly generate initial cell based on dimension and density inputs
* General rules class
  + One or fewer neighbors, cell dies in next generation
  + Two neighbors, remains stable
  + If cell has 3 neighbors, all cells remain stable, an empty (random cell?) is born (if there are any empty cells neighboring)
  + If a cell has 4 neighbors, it will be empty in the next generation due to overcrowding

Tip: Since new births and deaths cannot affect other births and deaths in the same generation, switch the pointer reference. If that’s too hard, have two grids (current generation and next generation) and after computing is finished, copy next generation to current generation.

Steps

* Random configuration or flat-file configuration? If random what dimensions would you like?
* What game mode?
* Pause between generations, or output to file?
* Run simulation
* If simulation is infinite, keep running. If world becomes empty or stabilizes, halt simulation and ask user to press “enter” to exit program

g++ -o main \*.cpp

./main