Conway's Game of Life

Requirements

Write a program that plays Conway's Game of Life. Conway's Game of Life is a cellular automaton. From Wikipedia:

> *The universe of the Game of Life is an infinite two-dimensional orthogonal grid of square cells, each of which is in one of two possible states, alive or dead. Every cell interacts with its eight neighbours, which are the cells that are horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:*
>
> - *Any live cell with fewer than two live neighbours dies, as if caused by underpopulation.*
> - *Any live cell with two or three live neighbours lives on to the next generation.*
> - *Any live cell with more than three live neighbours dies, as if by overcrowding.*
> - *Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.*
> - *The initial pattern constitutes the seed of the system. The first generation is created by applying the above rules simultaneously to every cell in the seed —births and deaths occur simultaneously, and the discrete moment at which this happens is sometimes called a tick (in other words, each generation is a pure function of the preceding one). The rules continue to be applied repeatedly to create further generations.*

Nonfunctional Requirements:

- the program must be implemented in Java and utilize the GridWorld platform
- the program must seed the system by calling a static method defined in the GameOfLife class with the following signature (to facilitate my testing):
  `public static void populateCells(ActorWorld world)`
- do not use GridWorld's execution engine to produce subsequent generations; directly produce and display subsequent generations directly

Artifacts to Produce:

- Requirements Document: Many functional and nonfunctional requirements needs to be defined. You must define additional requirements that are reasonable and document them in a requirements document. I must review your requirements document before you start the design document or test plan. You may change the requirements document throughout development.
- Design Document: You must do some design activity before starting implementation. This may consist of a flow cart, pseudocode, or other design artifact. I must review your design document before you start implementing code. You may change your design document throughout development.

- Test Plan: You must create a test plan with specific test cases before starting implementation. I must review your test plan before you start implementing code. You may change your test plan throughout development.
- Working Software: You must produce working software the meets the requirements and is verified and validated by your test plan.

Files to be Submitted:
- requirements document
- design document (pseudocode, flow charts, etc.)
- GameOfLife.java source file
- test plan (with specific test cases)
- reflection on the lab

Rubric:
The implementation will be evaluated according to the Computer Science Grading Rubric in Canvas. The other artifacts will be evaluated independently.