Game of Life exercise

Definition:

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.

Rules:

At each step in time, the following transitions occur:

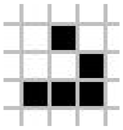
1. Any **live cell with fewer than two live neighbours dies**, as if caused by under-population.
2. Any **live cell with two or three live neighbours lives** on to the next generation.
3. Any **live cell with more than three live neighbours dies**, as if by overcrowding.
4. Any **dead cell with exactly 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.

Objective:

1. Implement game of life data structures and algorithm
2. Demonstrate that game of life algorithm works

   Note: the program has to run and work properly (working program is better than in-progress design).
   Note: use a 'Glider' pattern placed in the middle of 25x25 cell universe for this exercise.



Guidelines:

1. Please limit yourself to no more than 2 hours for this exercise.
2. Use any language and/or frameworks you'd like.

3. No actual “UI” is required - console output is fine.
4. Be ready to explain your design decisions and how you might improve/expand upon your solution.
5. Please submit your solution using Github or Dropbox or Google Drive or etc.
6. Include any setup details as needed to make your solution run.
7. Please email us if you have any questions.