**Conway's Game of Life**

The universe is composed of an **infinite** two-dimensional grid of cells.

Each cell has 8 neighbors which are the cells that are horizontally, vertically or diagonally adjacent.

Each cell has one of two possible statuses: Alive or Dead.

The cells exist in generations. Births and deaths occur simultaneously at a discrete moment. Each generation is a pure function of the preceding one.

Given generation zero, the state of the cells in the next generation is calculated by the following rules:

1. Any live cell with fewer than two live neighbors dies, as if by under population.
2. Any live cell with two or three live neighbors lives on to the next generation.
3. Any live cell with more than three live neighbors dies, as if by overpopulation.
4. Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

Let's see an example:

Generation 0:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Generation 1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Generation 2:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Generation 3:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Let's see more on Wikipedia.

Instructions:

* Decide on the grid representation. Pay attention to the data structures you use.
* Think of the solution for the problem of calculating the next generation.
* Write the code on the computer. Pay attention to design choices, object oriented programming and clean code.