

# Cellular automata - basic concepts

## Conway's Game of Life

### Preparation

- Download and extract project: Game of Life
- Import project into your IDE.
- Run project - your simple CA framework is ready.

### Definition of Conway's Game of Life

Conway's Game of Life is one of the most popular example of Cellular Automata application. It is a two dimensional, homogeneous CA, based on square lattice with Moore neighborhood. Each cell can be in one of two states: 1 - alive, 0 - dead.

- Dead cell, changes its state to alive in next iteration if there are exactly 3 alive neighbors, otherwise it remains dead.
- Alive cell, remains alive if it has 2 or 3 alive neighbors, otherwise it become dead.

This transition rules can be written as 23/3.

### Implementation (2 pkt)

Fill the missing fragments of source code in classes Board and Point.

- In class Board, in method initialize() initialize neighbors for each cell in table. Use Moore neighborhood. For basic version you don't have to initialize cells on the borders, however implementing periodic boundaries can be a good idea.
- In class Point write a method which returns a number of alive neighbors.
- In class Point, in method calculateNewState(), calculate new state of given cell - according to its actual state and number of alive neighbors. Remember to save cells new state in variable: nextState.
- Run program, and analyze observed behavior of CA automata.

### Alternative rules (2pkt)

One can use different transition rules for Game of Life. Modify your program to use following rules.

- 2345/45678 - cities
- 45678/3 - coral

Try to find other interesting rules.

# Rain (2 pkt)

Try to implement a simulation of rain in CA

- Set neighborhood of cells so that the only neighbor of given cell is the cell one step below.
- In class Point, write method drop(), which change cell state to 6 with some small probability (e.g. 5%).
- In class Board, in method iteration() use method drop() for cells in top row.
- If cell state is higher than zero, set nextState = currentState -1
- If cell state is equals to 0 and its neighbor is higher than 0 set nextState = 6
- In class Board, in method drawNetting() change colors of cells, so that blue color becomes faded while its state number decrease. You can use method:

```
g.setColor(new Color(0.0f, 0.0f, 1.0f, 0.65f));
```

Constructor parameters for class Color are: red, green, blue, alfa.