Jakub Ciecierski

Cellular automaton
Requirement specification

1 Glossary

Cellular automaton - consists of a *grid* of *cells*, each in one of a finite number of *states* (e.g. on and off). A new *generation* is created, according to some fixed *rule* that determines the new state of each cell.

Cell - can be in one of many states, in case of binary cellular automaton we only have two states for each cell, namely on and off. For each cell, a set of cells called its *neighborhood* is defined relative to that cell.

Grid - can be in any finite number of dimensions. Consists of cells. An initial state is selected by assigning a state for each cell

Rule - determines the new state of each cell in terms of the current state of the cell and the states of the cells in its neighborhood. Typically, the rule for updating the state of cells is the same for each cell, and is applied to the whole grid simultaneously, such application of a rule to the entire grid, creates a new *generation*

Neighborhood - have possibility to introduce rules which determine the new state of each cell. The transition of a cell is based on the states of his neighbors which are defined under different environments such as:
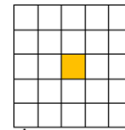
- 4 points neighborhood
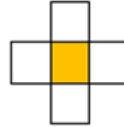
- 8 points neighborhood
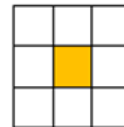
- 24 points neighborhood

Pattern - is a combination of specific layout of cells on the grid and the set of rules to be applied for them.

2 What is the project about

What is cellular automaton?

3 Goal

The goal of this project is to create an application for cellular automaton. The automaton will consist of grid of cells that can be in either of two states (e.i. on / off) and will operate in 3 different environments:

- 4 points neighborhood.
- 8 points neighborhood.
- 24 points neighborhood.

The application will allow user to create, save and edit rules based on this environments and of course observe each generation on the two dimensional finite in size grid of cells (the grid will have wrapping option to simulate infinite size - the left edge will be connected with right edge).

The user will be able to draw cells on the grid and save the state of grid into patterns. The user can easily move around the grid, zoom in and out.

4 User stories

Grid editor

- As a user, I want to open grid editor, in order to change the grid size.

- As a user, I want to open grid editor, in order to change color of each state of a cell.

- As a user, I want to open grid editor, in order to enable/disable wrapping option.

Rule editor

- As a user, I want to open rule editor, in order to create new rule.

- As a user, I want to choose neighborhood environment, in order to add new rule.

- As a user, I want to define specific transition for a given state of cell, in order to generate new state.

- As a user, I want to click save/save as button in rule editor, in order to save current rule.

- As a user, I want to click load button in rule editor, in order to load current rule and possible edit it.

Application option

- As a user, I want to move View components (e.g. rule editor / grid editor / browser), in order to position them in different location.

- As a user, I want to click next generation button to compute next generation

- As a user, I want to click next N generations button, to compute next N generations.

- As a user, I want to set the number of generation to skip by clicking next N generations button, in order to compute next N generations.

- As a user, I want set the speed of computation of next generation in running mode, to customize the speed of which the automaton is transitioning.

5 Functional Requirements

ID	Requirement	Comments	Priority
1	The system provides a Grid options allowing for changing the size, colour of cells and enabling/disabling wrapping option	N/A	1
2	The system provides a Rule editor in which the user can create, edit and save rules.	N/A	1
2.1	By clicking create button in Rule editor, the application will open a fresh rule creation window	N/A	1
2.2	By clicking load button in Rule editor, the application will open a browser which will allow the user to find saved rules	N/A	1
2.3	By clicking save button in Rule editor, the application will make sure that name for the rule is provided and then will save the rule in specified by the user location	N/A	1
2.4	The system provides three different neighborhood environments in which the user can create rules, 4-point, 8-point, 24-point	See Glossary / Neighborhood for more information	1

ID	Requirement	Comments	Priority
2.5	The application provides special file extension for saving and keeping rules	N/A	1
2.6	For 4-point and 8-point environments the system should provide a way to create rules in which positions of neighbors relative to the cell are considered. If a transition is not defined then this transition does not change the state of current cell	The user can choose to what state current cell transitions, based on this cell's state and states of his neighbors	1
2.7	For 24-point environment system should provide a may of creating rules in which the user specifies number of neighbors in each column	This environment can be represented as a 5 by 5 matrix with the current cell in the middle	1
2.8	For 4-point, 8-point and 24-point environments the system should provide a simplified mode of creating rules in which the user inputs only number of neighbors in given state in the neighborhood for a current cell state.	The user inputs number of neighbors with given state which should appear for the cell to transition to another specified state	2
3.1	The system provides a step-by-step button which computes next generation	N/A	1
3.2	The system provides next-N button which computes next N generations, the N must be easily chosen by the user	N/A	1

ID	Requirement	Comments	Priority
3.3	The system provides a run button which will start the animation of consecutive generations	N/A	1
3.4	The system provides way to change speed of which the animation is drawn in the 'run' mode	N/A	1
4.1	The application allow user to draw cells on the grid	N/A	1
4.2	The application provides a way for user to save grid state into patterns, additionally the pattern can have a rule attached to it, which later can be loaded into the grid.	The grid state consists of its size, states of cells and other grid options.	1
4.3	The application provides Browser window in which the user can browse saved rules and patterns	N/A	1
4.4	The application allows the user to have multiple grids opened.	N/A	2
4.5	The application should have example of simple game of Life	N/A	2

6 Non Functional Requirements

- **Usability**

  The user interface should contain the following View components but not limited to:

  - Browser of saved rules and patterns.
  - Grid.
  - Rule editor.

  Each of View component should be able to be disconnected from the main application and moved around freely. These components should also be able to be attached to the sides of the main window of the application(left, center, right, bottom).

  The main window should provide easily accessible menu and buttons for options, guides and running the grid animations features. The application should provide a way for the user to set up key binds(short cuts) for all buttons in the system.

  Guides for setting up the rules are to be thoroughly explained.

  In the first launch of the application the user should be welcomed with a window which can lead the user through the application and load some examples provided in the system.

- **Reliability**

  reliability what

7 GUI mock-up

8 Evaluation of solution