

3 March 2015

Jakub Ciecierski

Cellular automaton Requirement specification



Contents

1	Schedule	3
2	Document metric	3
3	Glossary	4
4	Goal	6
5	User stories	7
6	Functional Requirements	8
7	Non Functional Requirements	13
8	Graphical User Interface	14
9	Risk analysis	15
10	History of changes	16

1 Schedule

2015-04-02	Technical project
2015-04-23	Code of modules
2015-04-30	version 0.98
2015-05-07	version 0.99
2015-05-14	version 1.00
2015-05-28	Test report
2015-06-11	Acceptation

2 Document metric

Document metric					
Project:	Cellular Automaton	Company:	WUT		
Name:	Requirement specification				
Topics:	Analysis of the product				
Author:	Jakub Ciecierski				
File:	requirement_specification				
Version no:	0.1	Status:	Under development	Opening date:	2015-03-03
Summary:	This document provides the requirement specifications				
Authorized by:	Wadysaw Homenda Lucjan Stapp	Last modification date:			2015-03-03

3 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.

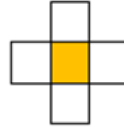
Cell - can be in one of many states, in case of binary cellular automaton we only have two states for each cell, namely on and off. For each cell, a set of cells called its *neighborhood* is defined relative to that cell.

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

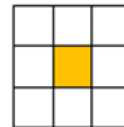
Rule - determines the new state of each cell in terms of the current state of the cell and the states of the cells in its neighborhood. Typically, the rule for updating the state of cells is the same for each cell, and is applied to the whole grid simultaneously, such application of a rule to the entire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

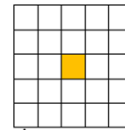
- 4 points neighborhood



- 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.

4 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.

5 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.
- As a user, I want to scroll my mouse roll over the grid, in order to adjust the scale of the grid.
- As a user, I want to select the brush, in order to draw cells on the grid.

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.

Pattern editor

- As a user, I want to save a current state of grid into a pattern, so that later I can load it into the program.
- As a user, I want to browse for my patterns in the browser window, in order to load it to the pattern editor.
- As a user, I want to click change rule in the pattern editor, in order to add a rule of my choosing to that pattern.

6 Functional Requirements

Priority

- 1 - must be implemented
- 2 - can be implemented optionally
- 3 - is a nice addition, but not needed.

ID	Requirement	Comments	Priority
1	The system provides a Grid options allowing for changing the size, colour of cells and enabling/disabling wrapping option	The colour of cells represent a state of the cell. In other words the user can choose in what state to put a cell into.	1
1.1	The system should allow grid maneuvers, zooming in/out and if the entire grid is not visible in one screen, possibility of moving around the grid		1
2	The system provides a Rule editor in which the user can create, edit and save rules.		1
2.1	By clicking create rule button in Rule editor, the application will open a fresh rule creation window		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		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		1

ID	Requirement	Comments	Priority
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
2.5	The application provides special file extension for saving and keeping rules		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 way 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		1

ID	Requirement	Comments	Priority
3.2	The system provides next-N button which computes next N generations, the N must be easily chosen by the user		1
3.3	The system provides a run button which will start the animation of consecutive generations		1
3.4	The system provides way to change speed of which the animation is drawn in the 'run' mode		1
4.1	The application allow user to draw cells on the grid		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.	A pattern editor view component should be created. The grid state consists of its size, states of cells and other grid options.	2
4.3	The application provides Browser window in which the user can browse saved rules and patterns		1
4.4	The application allows the user to have multiple grids opened.		2
4.5	The application should have example of simple game of Life called Conway's Game of Life		2

7 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.
- Pattern editor.

Each of the View components 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.

The user must be informed of any loading time.

- **Performance** The application is assumed not be heavy on computations within the start-up time, thus

the time for the application to start should not exceed 5 seconds.

The response time of loading any view components to the application should not be noticeable by the user (within 1-2 seconds maximal)

8 Graphical User Interface

The GUI should be responsive, clear and editable for user's needs.

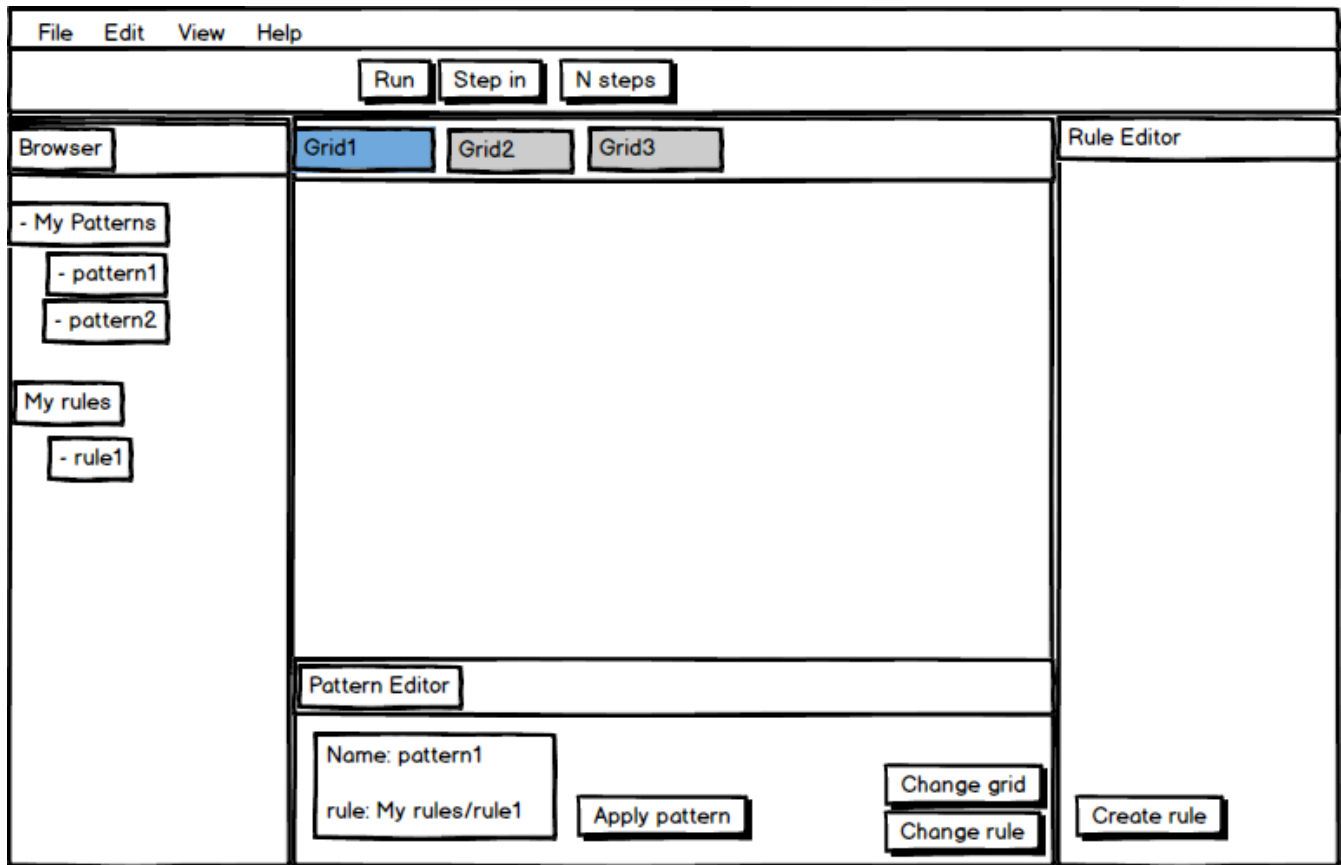
The interface should provide an easy way for the user to position each component of the view as he pleases. There should be places where the user can attach these components (left, bottom, right or center of the main window of the application). These components can also be disconnected and flow as separate windows.

The browser should allow for searching of patterns and rules. The rules can be easily applied using double-click with left mouse button or right click and selecting the apply feature. The rules can also be put into edit from the browser, either by selecting an appropriate 'Edit' button in the right-click menu, or by dragging the rule into the Rule editor window.

The Patterns should work in similar way. After loading a pattern into the editor, user should be able to change a rule and grid associated with that pattern.

In figure 1 we see an example of components alignment

Figure 1: GUI mock-up



9 Risk analysis

The biggest risk of this project is not finishing the application in time. The developer should make sure that all requirements of highest priority are implemented first.

10 History of changes