7 March 2015

Lukasz Wójcik

Individual project
Cellular automaton
Requirement specification

Informations

Scheldue

Date	Stage
2015-03-12	requirement specification
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 1.0
2015-05-21	test report
2015-06-11	acceptation

Document metric

Document metric						
Project:	Cellular automaton		**Company:**	WUT		
Name:	Cellular Automata requirement specification					
Topics:	Features of Cellular Automata program					
Author:	Lukasz Wójcik					
File:	cellular_automata_requirement_specification					
Version no:	0.1	**Status:**	Designing		**Opening date:**	2015-02-26
Summary:	Providing business analysis and requirement specification.					
Authorized by:	Wladyslaw Homenda		**Last modification date:**			2015-03-11

Changes

History of changes			
Version	Date	Who	Description
0.1	2015-03-11	Lukasz Wójcik	Creation of requirements specification

Contents

Short summary of documentation

This document is a requirement specification of Cellular automation program.

1 Goal of the project

A goal of this project is creation of cellular automaton based on Conway's Game of Life which will consist of 2D finite grid with cells, which can have 2 states (black or white,0 or 1). Automaton has to work in 3 environments (4 ,8 and 24 points neighborhood) in which user will create,save,edit set of rules. Software will also help in creating cells states what combining with rules will create new generations (change of state of each cell in the grid).Changing rules and cell states dynamically should also be possible.

2 Concepts vocabulary

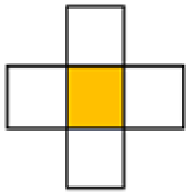
Cellular automaton - consists of a regular grid of cells, each in one of a finite number of states (in our case 0 and 1). The grid is two dimensional. For each cell, a set of cells called its neighborhood is defined relatively to the specified cell. An initial state is selected by assigning a state for each cell. A new generation is created, according to some fixed rule that 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 does not change over time, and is applied to the whole grid simultaneously.

Grid - in this project, grid is a 2D matrix with finite numbers of rows and columns.

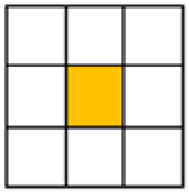
Cell - the smallest part of grid which has its state (0 or 1, black or white) and neighborhood.

Neighborhood - state of cell in next iteration depends on rule and its neighbors. In this project three types neighborhood are considered:

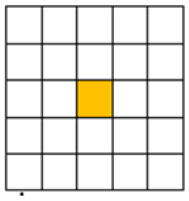
- 4 points neighborhood



- 8 points neighborhood



- 24 points neighborhood



3 User stories

Nr.	User story
1	**As a beginner user**, I want have easy access to help or tutorial.
2	**As a user**, I want to be able to change size of a grid.
3	**As a user**, I want to be able to set initial state for each cell in the grid drawing with my mouse and save it.
4	**As a user**, I want to be able to create and save my own rules.
5	**As a user**,I want to be able to create and save my own cells states.
6	**As a user**,I want to have an easy way to explore files with rules and cells states stored on my disk.
7	**As a user**,I want to make computations step by step or by N steps where N is provided by me.
8	**As a user who has been using this program before**, I want to be able to load previously saved states for cells in a grid.
9	**As a user who has been using this program before**, I want to be able to load previously saved rules.
10	**As a user who want to exit the program**,I want to be able to exit the program in any time. I also want to be informed about ongoing computations or changes which has not been saved

4 Functional Requirements

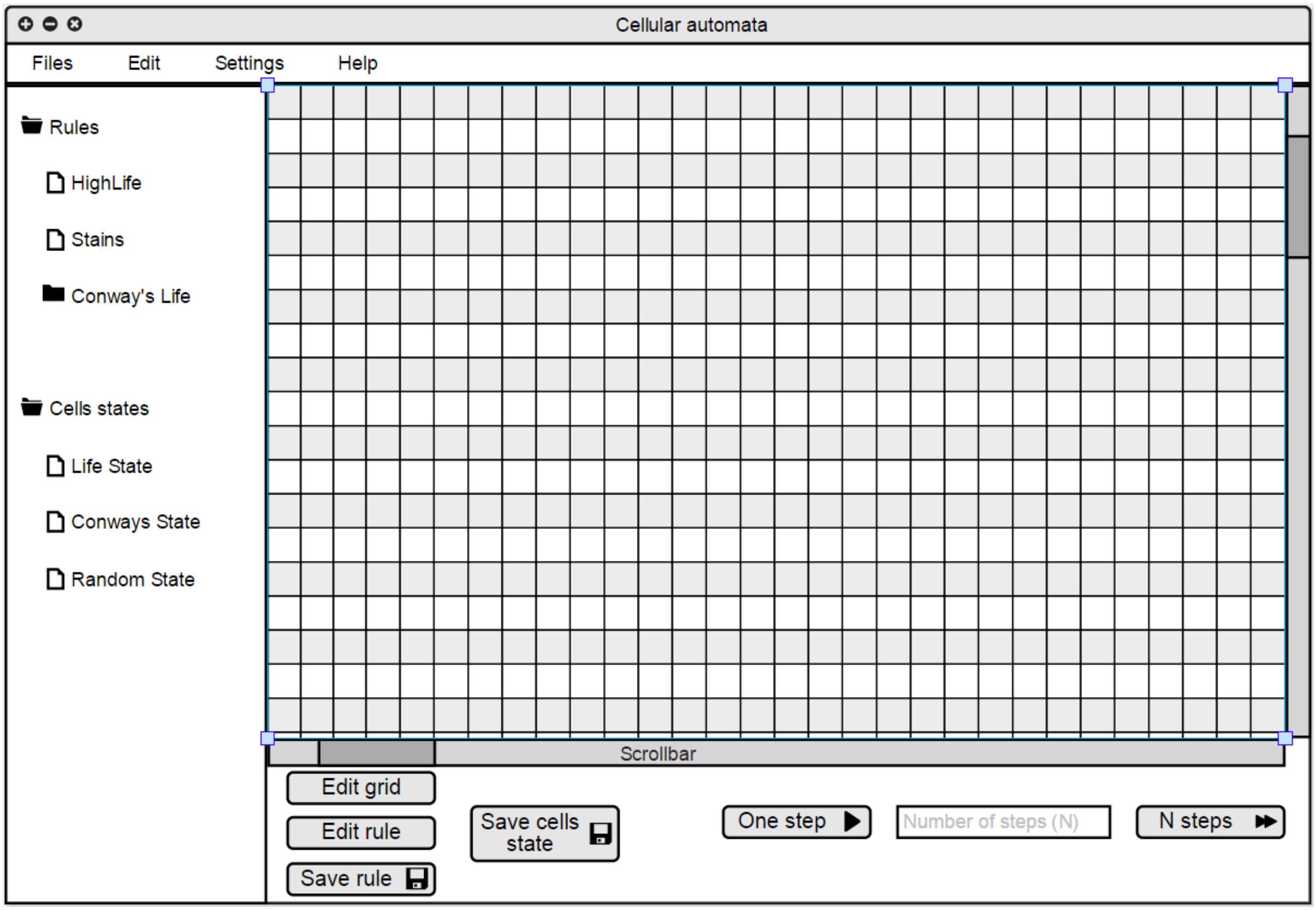
Nr.	Functional requirement
1	Clicking on ”Rules” folder in a left section of the main window of a program should expand a tree with a system of folders and files with previously saved rules.Double click on a file with a rule will load that rule into the program. Next click on a ”Rules folder should collapse a file tree.
2	Clicking on the ”Cells states” folder in left section of main window of program should expand a tree with system of folders and files with previously saved cells states. Double click on file with cells state will load that state into the program. Next click on ”Cells states” should collapse a file tree.
3	Scrolling with mouse scroll or pinching on touch pad should zoom in or zoom out the grid.
4	Clicking on a cell in the grid should change its state to opposite one (since we’re dealing with binary states).
5	Dragging mouse over the grid with left button pressed should change state of cell over which the cursor was hovering to the opposite one.
6	Clicking on ”Edit grid” button will make a popup window appear. In the popup we will be able to set number of rows and number of columns of the grid.
7	Clicking on ”Save cells state” button will save current state of cells in grid to a file. User will be able to specify folder location and name of the file.
8	Clicking ”Edit rule” button will open popup window with tools to edit rule which is currently in use.
9	Clicking on ”Save rule” button will save a current rule to a file. User will be able to specify a folder location and the name of the file.

Nr.	Functional requirement
10	Clicking on ”Files” button in menu bar on top of the program window will open a list of functions such as: create new rule,create new grid cells state. In rule creator, user will be able to choose neighborhood type (4,8 or 24 neighbors) and manually add principles determining state of each cell.
11	Clicking on ”Edit” button in top menu bar will open a list of functions which will provide tools to edit rules and cells states which are not currently in use.
12	Clicking on ”Settings” button in top menu bar will provide tools to change source of folders with rules and cells states.
13	Clicking on ”Help” button in top menu bar will show a list of functions such as: problems, tutorial, contact. ”Problems” function will provide answers to most common problems (if such would exist).”Tutorial” function will be able to show to user how to use ”Cellular automata” program step by step. Contact function will provide a way to contact with person responsible by functionality of this software.
14	Clicking on ”One step” button will execute computation of the next generation. If rule or initial state of cells will not be set, user will be prompted to do so.
15	Clicking on ”N steps” button will execute computation of next generation N times. If rule, initial state of cells or number of steps will not be set, user will be prompted to do so.
16	User will be able to minimize,maximize or change the size of a program window. User will also be able to exit program in any time but he will be alarmed about any ongoing computations or not saved changes.

5 Non Functional Requirements

A user interface should be as clean and user friendly as it's only possible. Intuitive addition of rule sets. Software should be snappy, make use of new hardware (like multi-core processors) but also compatible with older devices. Computations should not block or slow down GUI. In case of some error (computer restart, system crash, etc.) user's work should be stored in a backup file which should be updated during software runtime.

6 GUI prototype



User interface should be kept as simple and as clean as its possible. GUI prototype shown above consist of 4 components. Top menu bar has buttons forwarding to functions for modifying rules and cell states. It also contain application settings and help with tutorial. On the left there are two file trees with default and previously made rules and cells states. On the bottom there is a set of buttons for rules and cells states currently in use and buttons for starting computing of new generations.

User should have possibility to resize all components except menu bar.

7 Evaluation of solution

- Correctness of solution
- Meeting the requirements
- Clarity

8 Risk analysis

One of the risk is not sticking to schedule on one of more stages of software development. Not meeting basic requirements stated on the begging might also be a big problem. Errors made on early stage of development can be very troublesome, since correcting it will most likely involve more work than it would right after creation of such error. Another problem which may occur is compatibility with different software environments (Windows,Linux).