28 March 2015

Lukasz Wójcik

Individual project
Cellular automaton
Technical project

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 technical documentation				
Topics:	Technical aspect of project				
Author:	Lukasz Wójcik				
File:	cellular_automata_technical_documentation				
Version no:	0.21	**Status:**	Designing	**Opening date:**	2015-03-26
Summary:	Providing technical documentation.				
Authorized by:	Wladyslaw Homenda		**Last modification date:**		2015-04-01

Changes

History of changes			
Version	Date	Who	Description
0.1	2015-03-26	Lukasz Wójcik	Creation of technical documentation sections. Initial planning
0.2	2015-03-27	Lukasz Wójcik	Creation of technical documentation.
0.21	2015-03-27	Lukasz Wójcik	Correction of spelling mistakes.

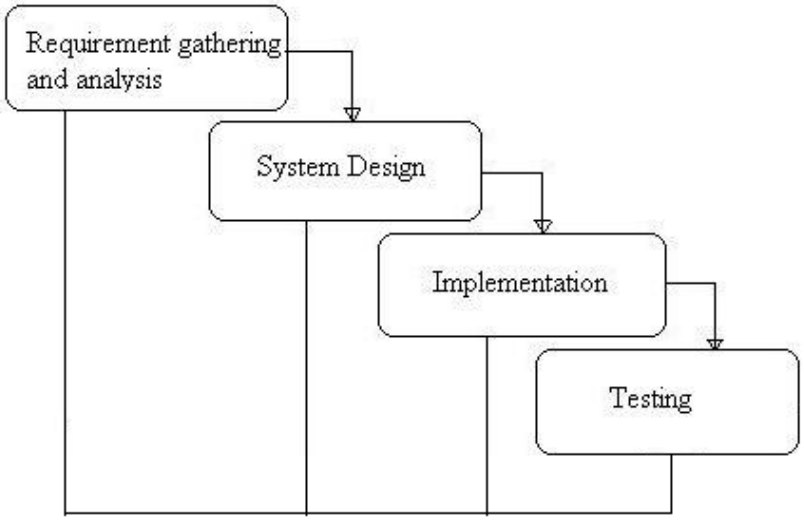
Contents

Short summary of documentation

This document is a technical documentation of Cellular automation program.

1 Production model

Methodology used for implementation is a "Waterfall Model"



Waterfall-model (In our case it end on testing)

This model is easy to understand and use. It is mostly used with relatively small project where all requirements are well understood (returning to earlier stages of project from e.g testing might be very expansive so it is better not to move to another stage without debugging). Each stage is completed one at a time so they are not overlapping.

2 Technologies

Our application will written in C# language. For app development we will use Visual 2013. As a graphic engine we will use WPF (Windows Presentation Foundation) which is based on .NET 3. It gives nice tools to divide project on GUI and logical parts what helps in creating clean code. This application is destined for Windows Vista and higher.

3 used data structures

To store grid data we will use two nested lists of cells object(or just boolean).In C# exploring elements in List is as fast as in simple table but it gives way more tools. List is easy to expand/shrink and has many search functions already implemented

4 Description of algorithm

This product will be based on Conway's Game of Life automaton. It means that is hat two dimensions. It also has two states (ON and OFF). State of the cell will be decided by **rules**. Rules describe which cells in neighborhood (4,8 or 24 point) are important and how many of them have to be in "ON" state to keep cell alive or dead. Overlapping rules are resolved by one of three logical operations (XOR,OR,AND). We need two 2-dimensional matrices and one of which will start as "original" one. We will go through matrix with 2 loops where one will be placed in another. In each iteration of inner loop we will calculate cell state. Depending on neighborhood size, appropriate rule set will be chosen. We have to iterate through all rules in rule set and check which of them can be applied.

If there is more than one of them, than we use predefined logic operation to resolve this conflict (conflict arises since we can also create rule for killing a cell). in case where some of neighbors are outside the matrix, we virtually "stitch" left with right side and top with bottom border. For instance if neighborhood i exceeding top border, lacking neighbors are those adjacent to bottom border.

After calculating the state of a cell, it is saved to cell with the same index in second matrix. Calculation of a new generation ends up in setting second matrix as an original one and displaying it on the screen.

5 Functionality

- Display visualization of cell movements - every step of components behavior must be shown on a grid.

- Change environment - by selecting them; there must be at least 3 types of environments: 4, 8, 24 points neighborhood.

- Introduce new rules - by adding new rules determining new behavior of cells.

- Change rules - by dynamically changing existing rules.

- Create new rules - by selecting neighborhood cells and how many of them should be alive in order to make the target cell either alive or dead.

- Select number of steps .

- Control simulation - by choosing options play, pause, stop or reset simulation.

- Observe movements step by step - by rewinding cells movement on click, depending on the selected number of steps.

- Resolve conflicts when there are contradicting rules - by selecting one of them or selecting one of the logical operations (XOR, OR, AND) to be performed on them, for example rule 1 OR rule 2.

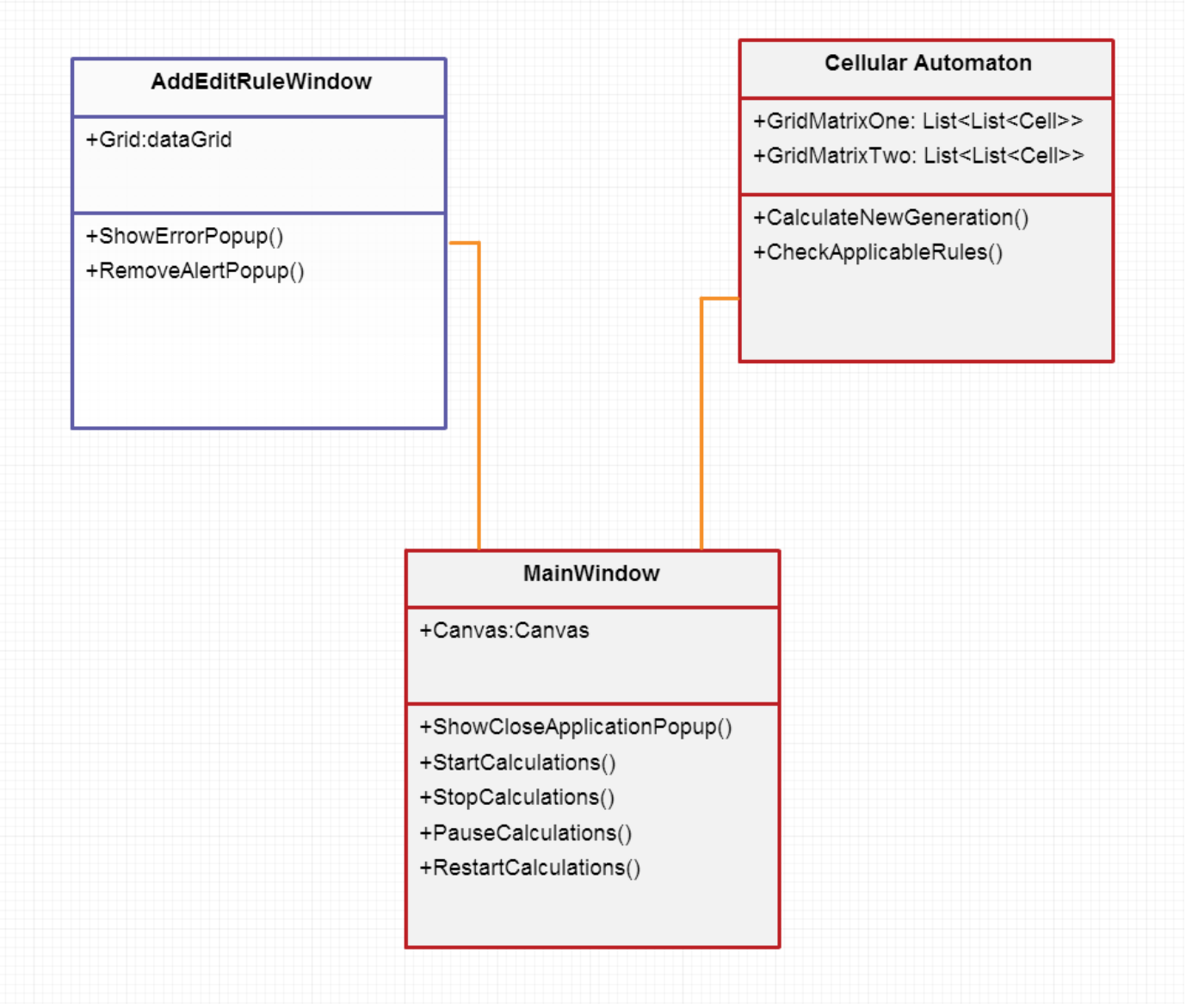
- Change grid's size - by option to zoom in/zoom out the visualisation.

- Load known automaton - for example Langton's ant.

- Display information - by showing for example time of simulation (in seconds), number of cells alive.

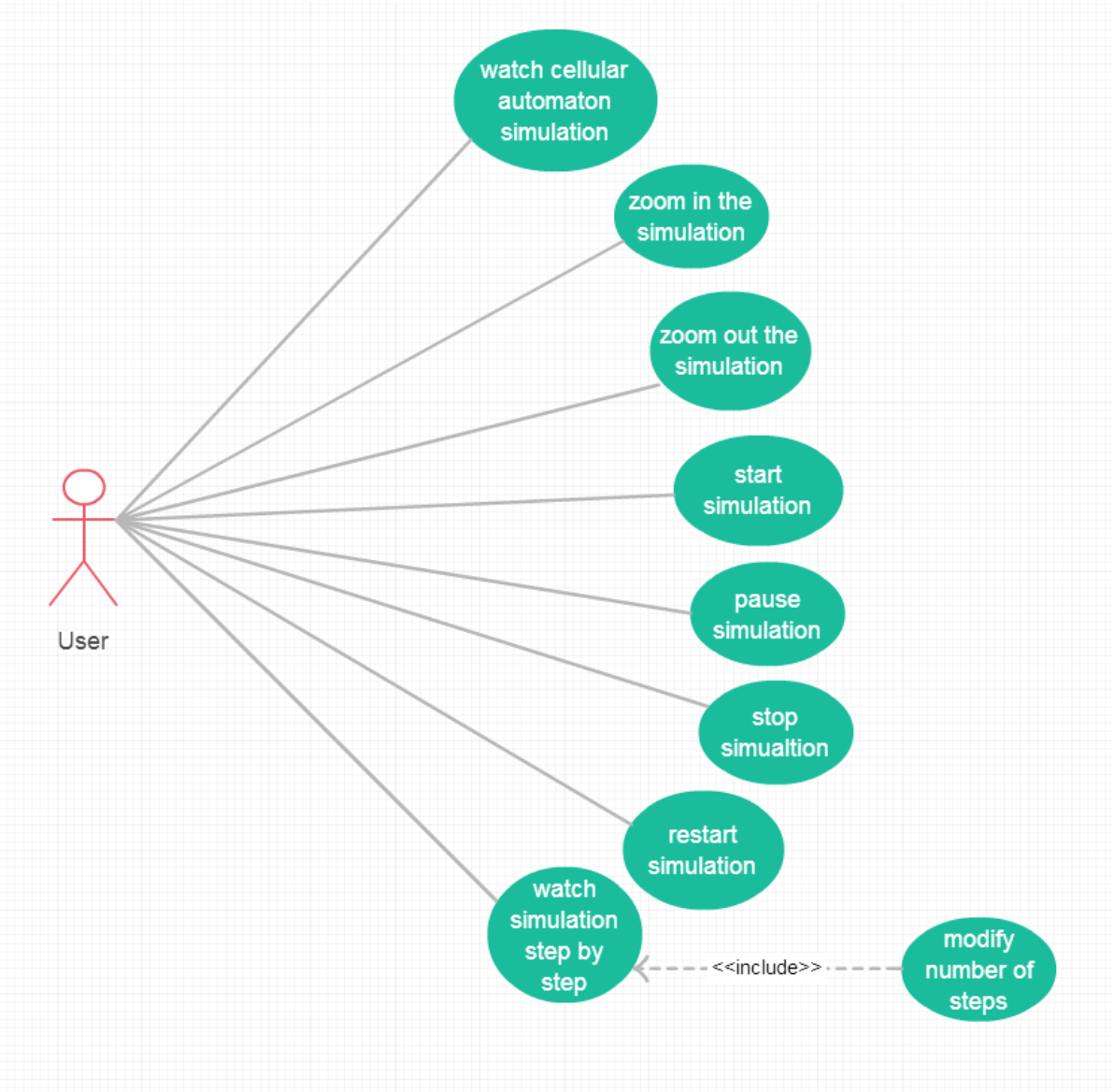
6 Class diagram / program structure

This class diagram is a simple visualization of classes in our program. Person implementing these classes is free to divide them to smaller ones.



7 Use cases

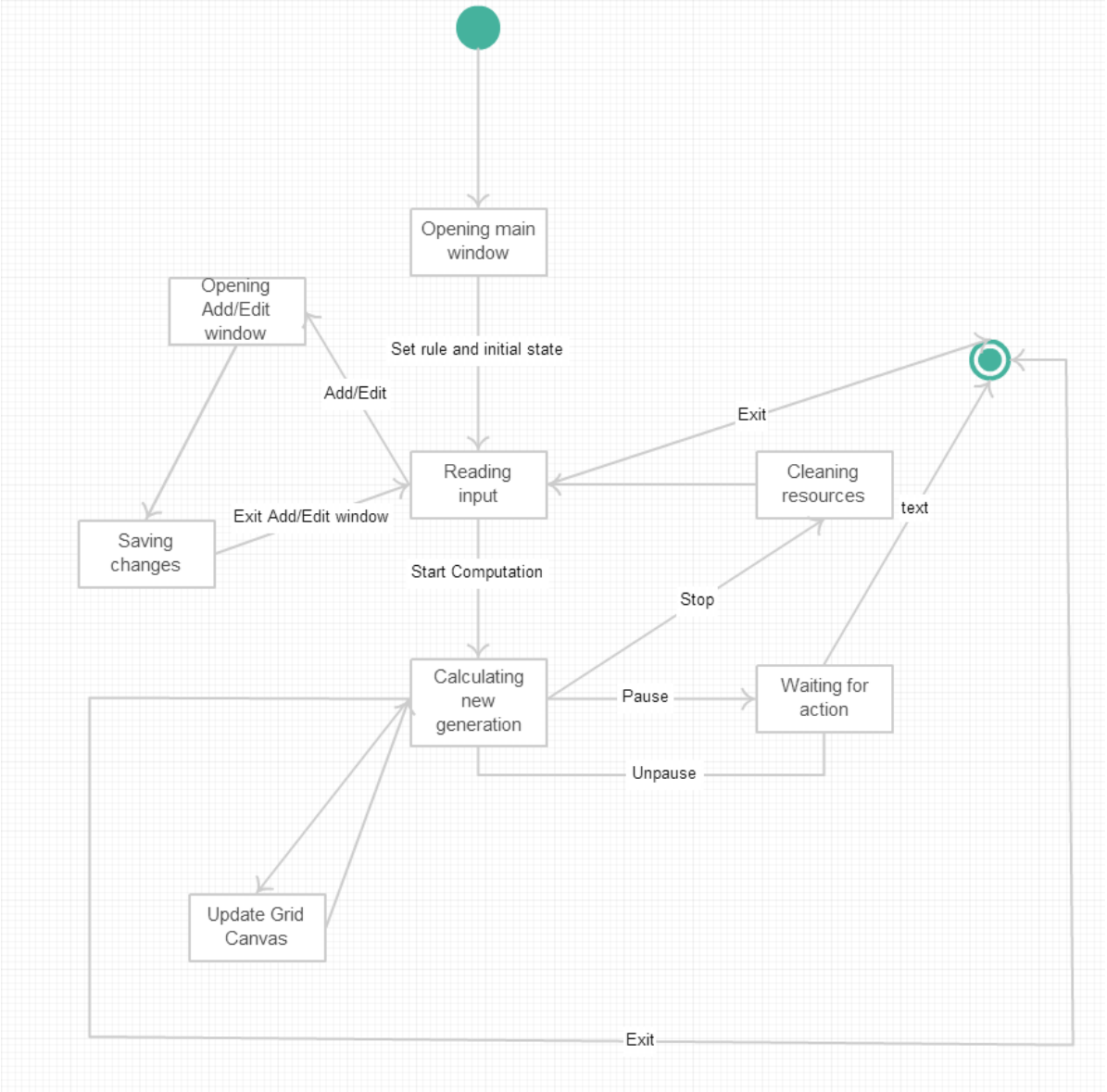
Use case for user in main menu.



Use case for user in Add/Edit menu.

8 States diagram

Simple state diagram describing behavior of program in general.



9 Graphical user interface mock-up

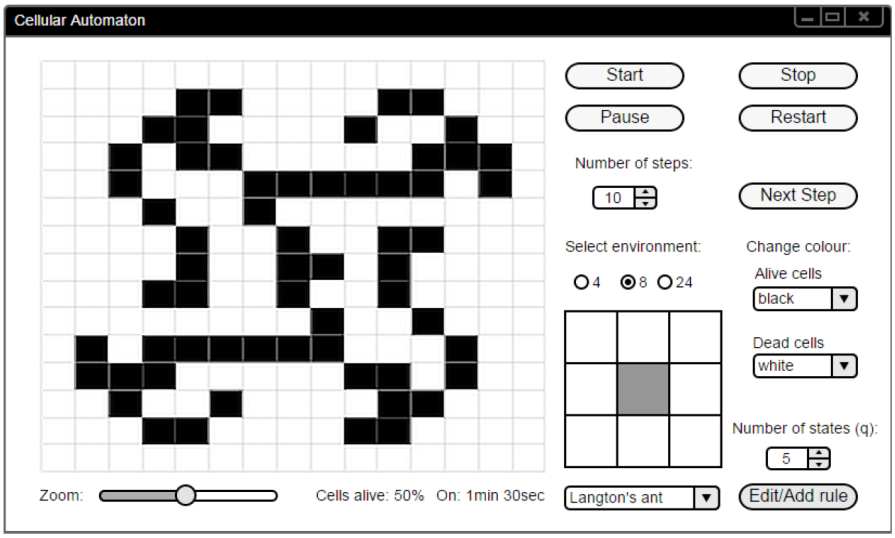


Figure 1. Main window of the application

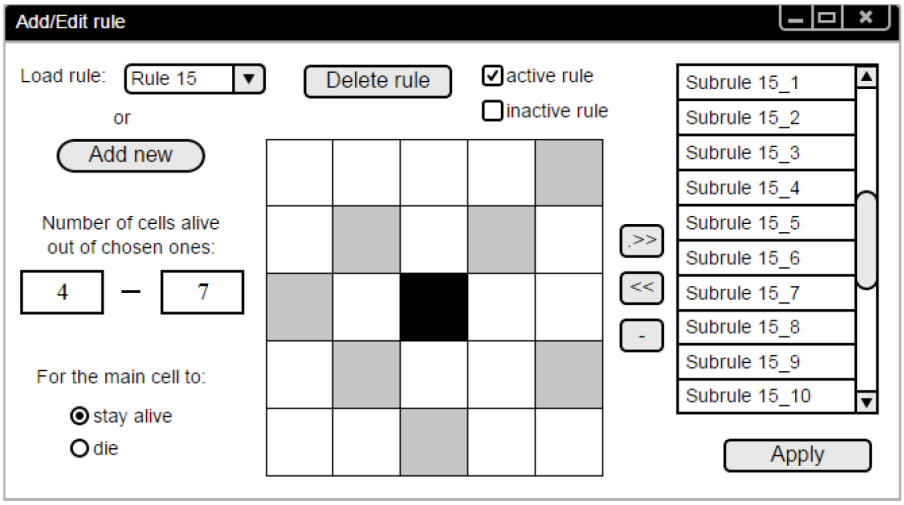


Figure 2. Edit/Add new rule window

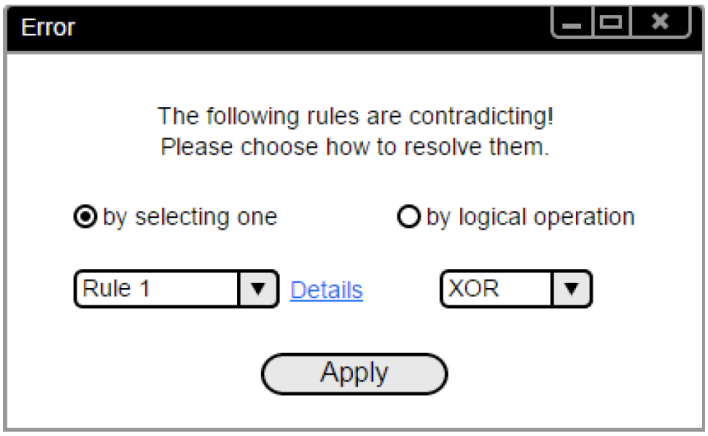


Figure 3. Pop-up window for resolving conflicts in rules

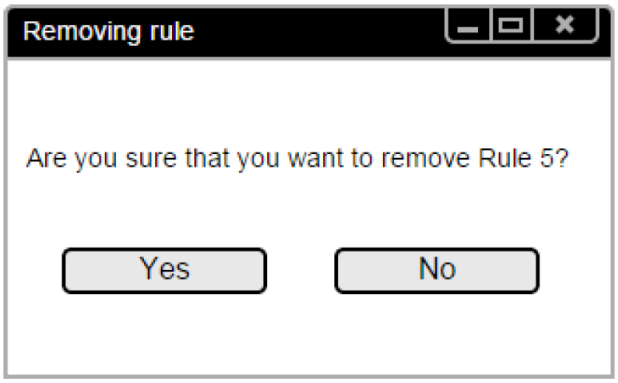


Figure 4. Pop-up window warning about deleting a rule or subrule

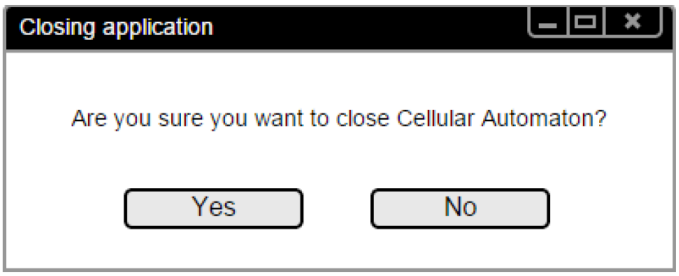


Figure 5. Pop-up window warning about closing the application

10 Summary

This technical documentation has been made basing on Business analysis made by Mai Viet Ba. "Functionality" and "Graphical user interface mock-up" has been copied since these informations are very important from technical aspect.

Application created basing on this documentation should be very fast, user friendly and eye pleasing. The biggest drawback will be support for only one computer system (it also might be open through "Wine" on Linux but it's not a official solution so it might not work properly).