

20 May 2015

Jakub Ciecierski

**Cellular automaton
Test documentation**



Contents

1	Document metric	3
2	History of changes	3
3	Introduction	4
4	Description of tests	4
4.1	Test Scenarios	4
5	Test log	5
5.1	Test results	5
5.2	Summary	7
6	Incident report	7
7	Summary test protocol	8

1 Document metric

Document metric					
Project:	Cellular Automaton	Company:	WUT		
Name:	Test Documentation				
Topics:	Test Documentation				
Author:	Jakub Ciecierski				
File:	test_report.pdf				
Version no:	1.0	Status:	Being Tested	Opening date:	2015-03-03
Summary:	Technical documentation for cellular automaton				
Authorized by:	Wadysaw Homenda Lucjan Stapp	Last modification date:			2015-05-20

2 History of changes

History of Changes			
Version	Date	Who	Description
1.0	2015-05-20	Jakub Ciecierski	Definition of the main purpose of the document

3 Introduction

The following test documentation contains evaluation of Cellular Automaton project designed by Alicja Kostrzewa and developed by Hubert Hunia. For vocabulary, refer to Business Analysis.

4 Description of tests

Presented here are test cases grouped into several test scenarios. The idea for test scenarios is to chain test cases into sequences where each previous test case's result(output) starts next test case(input).

4.1 Test Scenarios

1. Test scenario of basic application behaviour.
Estimated time to complete: *0.5hour*
 - 1.1. Change size of the grid.
Expected: The grid should resize.
 - 1.2. Draw few cells on the grid.
Expected: Clicked cells should have changed color.
 - 1.3. Add and apply a rule.
Expected: Notification of applied rule.
 - 1.4. Inspect chosen rule applied on the grid.
Expected: The rules previously applied to the grid should be inspected.
 - 1.5. Play and pause the automaton simulation.
Expected: The simulation should start after pressing play, and stop for pause.
 - 1.6. Delete a rule
Expected: Rule removed from a list, and its logic should not be included in the automaton generation computations.
2. Test Scenario for correctly creating, applying and saving new rules. Even though the automaton is being ran in this tests the actual correctness of computing new generation is tested in later scenarios.
Estimated time to complete: *1hour*
 - 2.1. Choose a neighborhood - 4, 8, 24. Repeat for each neighborhood.
Expected: Window with visual representation of given neighborhood where a position and states of cells can be selected.
 - 2.2. Add two new rules
Expected: Two rules added to a list which can then be selected.

- 2.3. Applying rules to the automaton
Expected: Automaton grid should inform user of changed rule and show which rules are currently applied.
- 2.4. Draw few cells on the grid
Expected: Cells clicked should be colored
- 2.5. Run 4 steps
Expected: Automaton grid should react to applied rules.
- 2.6. Save rule to a file
Expected: Rule should be saved to file system correctly.
- 2.7. Close the application and open it again.
Expected: The application should close without error.
- 2.8. Load a previously saved rule to the application
Expected: The same rule set which was saved to be loaded.
- 2.9. Repeat drawing cells(the same ones) and Run 4 steps.
Expected: The same results should be expected as in previous test in this scenario.
- 3. Test scenario for correctness of computing generations.
Estimated time to complete: *0.5hour*
 - 3.1. Choose neighborhood, create and apply few rules. Repeat for each neighborhood.
 - 3.2. Draw few cells on the grid and few single steps.
Expected: The generation should be computed correctly with respect to the applied rules.

5 Test log

The results of each test case described in previous section. Each test is labelled either **OK** or **NOK** i.e. test was successful or it failed, respectively. The tests are enumerated the same as in the previous section thus the description is left out, leaving only the id of each test.

5.1 Test results

- 1. Test scenario of basic application behaviour.
Actual time to complete: *0.5hour*
 - 1.1. **NOK**
Reason: Not implemented.
 - 1.2. **OK**
 - 1.3. **OK**

- 1.4. **NOK**
Reason: Double clicking on the rule elements in the list does not do anything. No matter what was done, the rule could not have been inspected.
- 1.5. **OK**
- 1.6. **OK**
- 2. Test Scenario for correctly creating, applying and saving new rules. Even though the automaton is being ran in this tests the actual correctness of computing new generation is tested in later scenarios.
Actual time to complete: *2hours*
 - 2.1. **NOK**
Reason: Selecting 'Neighborhood mode' in combo box, changes the number but the neighborhood window in the rule editor does not change, it is a hard coded 24 neighborhood.
 - 2.2. **OK**
 - 2.3. **NOK**
Reason: Even though the rule can be added, it exists in the list, it does not appear to be applied to the grid.
 - 2.4. **OK**
 - 2.5. **OK**
 - 2.6. **NOK**
Reason: The 'Save to file' button does not work. The file called 'constrains.txt' exists but appending new rules to it does not work.
 - 2.7. **OK**
 - 2.8. **NOK**
Reason: Since saving did not work, loading was impossible. After clicking 'Load', three rules appear which were prepared as an example.
 - 2.9. **OK**
- 3. Test scenario for correctness of computing generations.
Actual time to complete: *0.5hour*
 - 3.1. **NOK**
Reason: The rules did no apply to the grid.
 - 3.2. **NOK**
Reason: The grid did not react correctly to newly created rules.

5.2 Summary

The prepared tests proved to fail on most of the functionalities, especially on the most important ones, i.e. creating, applying and saving rules. These tests lasted for about 3 hours. The Test scenario number 2 took more time than it was estimated. The reason for that was the hard to understand interface for rule manipulation. Till the very end of running tests, it was not clear if a given functionality was difficult to initiate or simply did not work.

6 Incident report

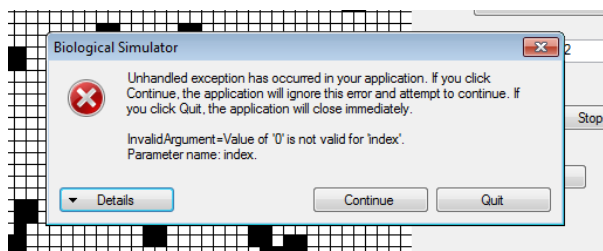
Events that require investigation. Grouped into three categories.

1. Critical

- 1.1. Saving rules to file does not work. The file exists, and has three examples saved but no rules are added to it after clicking 'save to file'.
- 1.2. Adding new rules either does not work or is extremely counter intuitive. The manual states that the rule does not compute the position of neighbors but its number. Yet the rule editor uses a 5 by 5 grid where each neighbor can be triggered dead or alive.

2. Major

- 2.1. No way to save current pattern of the grid.
- 2.2. Some actions can cause the application to show an error message with unhandled exception:



Recorded actions that caused the error: double clicking many times on the elements of Rule List, clicking 'Delete' button when no rule is selected.

3. Minor

- 3.1. The graphics of the grid flickers with every generation.

7 Summary test protocol

The application being tested was clearly not finished. The only functionalities that seemed to work was loading prepared rules from the example and running them on the grid. The grid it self, gave a horrible flickering visual bug with each generation. Even though adding rules appended the list of rules, there was no functionality to edit it again or even to apply it to the grid. Saving rules to file does not work. Some actions caused the application to show an error message. Resizing the grid and saving the pattern of a grid was not even implemented.

Running tests lasted three hours. It was difficult to tell if a given functionality was counter intuitive or simply not working.

Given the state of the application the final mark must be non other than: **Exclusion**. The application needs a lot of work to be accepted. Correctness of its functionalities must be the highest priority. However the application lacks a lot in being intuitive and clean, especially when it comes to rule manipulation and automaton visual generation steps.