CMSC 447

Software Test Report (STR)

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Signature** |
| Caroline Cocca | POC/Team Lead |  |
| James Gough | Simulation Dev |  |
| Stuart Reilly | Junior Developer |  |
| Kelley Schmidt | Documentation and  Diagrams Technician |  |
| Ryan Messett | Data Guru |  |
| Sang Nguyen | Visual Specialist |  |
| Dr. Jon Squire | Customer |  |

[1 Scope 3](#_3znysh7)

[1.1 Identification 3](#_1t3h5sf)

[1.2 System overview 3](#_4d34og8)

[1.3 Document overview 3](#_2s8eyo1)

[2 Referenced documents 3](#_3rdcrjn)

[3 Overview of test results](#_lnxbz9) 4

[3.1 Overall assessment of the software tested](#_1ksv4uv) 4

[3.2 Impact of test environment 4](#_z337ya)

[3.3 Recommended improvements 4](#_3j2qqm3)

[4 Detailed test results 4](#_4i7ojhp)

[5 Test log](#_23ckvvd) 7

# Scope

## Identification

This document applied to the tests we conducted for our Python implementation of Conway’s Game of Life. The application is made to run on Windows 10 via command prompt. The Game of Life is a cellular automaton that is a single-player game with the only user input needed being the initial setup of the starting cell(s) and menu item selections. The cells are contained in squares of a grid, with each square of the grid being a specific x-y coordinate. The cells multiply, shrink, and move to form patterns according to specified algorithms based on their placement and the placement of surrounding cells. The goal of the game is to create interesting patterns through various starting cell coordinates. Title is “Game of Life”, version number 1, release number pending.

## System overview

The purpose of this application is to demonstrate a classic cellular automaton with client-specific requirements. The input of the initial state of the game is the only input required in order to play the game. The software is a visual program and various aspects can be configured by the user. The system on which the software will run is the Windows 10 operating system. A mouse is required for the system to be able to navigate the software. The users of this software are the developers, graders, and the client. There are no project sponsors as this project will have no monetary cost to produce. The acquirer for our software application is the client, Dr. Jon Squire. The developers consist of our software engineering team - Kelley Schmidt, Caroline Cocca, James Gough, Sang Nguyen, Ryan Messett, and Stuart Reilly. The support agencies for our project would be the teaching assistants, Professor Cain, and the client if support is needed. Planned operating sites not applicable. Other relevant documents are the Software Design Description, Software Requirements Specification, Software User Manual, and Software Test Description.

## Document overview

This document serves as a record of the various tests we have conducted to test the requirements and features of the Game of Life. This document separates each test by the requirement number it corresponds to. Each test consists of one or more test cases. Problems as well as deviations from the previously specified tests are documented for each requirement. An overview of the tested software, a description of the test environment, and recommended improvements to the overall process are included. The team member responsible for performing each test is recorded as well as the test date in the log section. There are no privacy or security concerns regarding the use of this document.

# Referenced documents

Conway's Game of Life. (2019, March 24). Retrieved from <https://en.wikipedia.org/wiki/Conway's_Game_of_Life>

Tkinter -- Python interface to Tcl/Tk. (2019). Retrieved from <https://docs.python.org/2/library/tkinter.html>

Pandas: Python Data Analysis Library. (2019). Retrieved from <https://pandas.pydata.org>

# Overview of test results

## Overall assessment of the software tested

The software is steadily approaching its completed stage as shown by the test results. Some functions still need to be implemented but overall the application is cohesive, usable, and meets requirements for each completed function. The reading to file and writing to file functions are not yet implemented, and these account for the 2 requirements (2 and 3) stating that the system shall be able to read in and write to a file. The file reading function shall be implemented to load predefined shapes into the application as well. The beehive, beacon, and glider shape loading requirement (6) has not yet been met; the read file function will account for this.

One error was found while testing life count, wherein active (filled) squares clicked during a pause would occasionally not return to blank squares and instead remained filled. The issue is intermittent, with the source and solution for the error still being examined.

## Impact of test environment

None.

## Recommended improvements

None.

# Detailed test results

**Test 1: Requirement 1**

* **Summary:** 
  + Case 1: Grid squares appeared green as expected when green was selected from the menu.
  + Case 2: Grid squares appeared red as expected when red was selected from the menu.
  + Case 3: Grid squares appeared blue as expected when blue was selected from the menu.
* **Problems:** None.
* **Deviations:** None.

**Test 2: Requirement 2**

* **Summary:**
  + Case 1: When the WRITE FILE button is selected, Windows Explorer is launched as expected.
  + Case 2: When an invalid file name is specified, the game launches a pop-up text box that states the file is invalid as expected.
  + Case 3: The colors of the current grid points match the colors specified in the saved file as expected.
  + Case 4: The save function takes less than 2 seconds to execute as expected.
* **Problems:** None.
* **Deviations:** None.

**Test 3: Requirement 3**

* **Summary:** 
  + Case 1: File name input text box appears when READ FILE is selected as expected.
  + Case 2: An error message appears as expected when an invalid file name is specified.
  + Case 3: The color and grid points in the saved file match what is displayed on the board after reading the file as expected.
  + Case 4: The load function takes less than 2 seconds to execute as expected.
  + Case 5: The current state of the board is cleared when a new file is loaded as expected.
* **Problems:** None.
* **Deviations:** None.

**Test 4: Requirement 4**

* **Summary:** 
  + Case 1: When STOP is selected, the game pauses as expected and the life and step counts stop incrementing. No changes in current number or position of grid squares occurs.
* **Problems:** None.
* **Deviations:** None.

**Test 5: Requirement 5**

* **Summary:**
  + Case 1: The rate of gameplay speeds up when FASTER is selected from the Grid/Speed menu as expected. The life count and step count increment at a faster rate and match up with the grid square movements.
  + Case 2: The rate of gameplay slows down when SLOWER is selected from the Grid/Speed menu as expected. The life count and step count increment at a slower rate and match up with the grid square movements.
* **Problems:** None.
* **Deviations:** None.

**Test 6: Requirement 6**

* **Summary:**
  + Case 1: The Beehive test case pattern was placed in the grid, and the RUN option was selected. The lifecount accurately reported 6 for all steps executed.
  + Case 2: The Beacon test case pattern was placed in the grid, and the RUN option was selected. The lifecount accurately alternated from 6 to 8 for all steps executed.
  + Case 3: The Glider test case pattern was placed in the grid, and the RUN option was selected. The lifecount accurately alternated from 6 to 8 for all steps executed.
* **Problems:** None.
* **Deviations:** None.

**Test 7: Requirement 7**

* **Summary:**
  + Case 1: After a step was executed, the step count increased by one as expected.
  + Case 2: After the RUN option was executed, the step count increased by one with each step for the ten steps observed.
* **Problems:** None.
* **Deviations:** None.

**Test 8: Requirement 8**

* **Summary:** 
  + Case 1: Grid lines appear as expected when the SHOW GRID option is clicked.
  + Case 2: Grid lines disappear as expected when the NO GRID option is clicked.
* **Problems:** None.
* **Deviations:** None.

**Test 9: Requirement 9**

* **Summary:**
  + Case 1: The color options were observed to be present on the COLOR menu.
* **Problems:** None.
* **Deviations:** None.

**Test 10: Requirement 10**

* **Summary:**
  + Case 1: The Beehive test case from the Game of Life Wikipedia was placed in the grid and the RUN option was selected. After two steps, the word “Stable” appeared as expected.
  + Case 2: The Beacon test case from the Game of Life Wikipedia was placed in the grid and the RUN option was selected. After two steps, the word “Stable” appeared as expected.
  + Case 3: The Beacon test case from the Game of Life Wikipedia was placed in the grid and the RUN option was selected. After two steps, the word “Stable” appeared as expected. The program was paused after ten steps and the Glider test case was added to another area of the grid. After two steps, the word “Stable” disappeared as expected.
* **Problems:** None.
* **Deviations:** None.

**Test 11: Requirement 11**

* **Summary:**
  + Case 1: Grid squares are highlighted prior to performing the test. When RUN is selected, the game proceeds as expected and the life and step counts continue incrementing. RUN also works as expected after performing STOP test.
* **Problems:** None.
* **Deviations:** None.

**Test 12: Requirement 12**

* **Summary:**
  + Case 1: The Beehive test case from the Game of Life Wikipedia was placed in the grid and the RUN option was selected. The cells were observed to remain still and active for 10 steps.
  + Case 2: The Beacon test case from the Game of Life Wikipedia was placed in the grid and the RUN option was selected. The cells were observed to oscillate between the 2 specified shapes from the Wikipedia for 10 steps.
  + Case 3: The Glider test case from the Game of Life Wikipedia was placed in the grid and the RUN option was selected. The cells were observed to oscillate between the 4 specified shapes from the Wikipedia and move continuously about the grid for 10 steps.
* **Problems:** None.
* **Deviations:** None.

# Test log

This section shall present, possibly in a figure or appendix, a chronological record of the test events covered by this report. This test log shall include:

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement No. | Tested By | Date | Python Version |
| 1 | Kelley | 4/24/2019 | 3.7.3 |
| 2 | Ryan | 4/29/2019 | 3.7.3 |
| 3 | Ryan | 4/29/2019 | 3.7.3 |
| 4 | Kelley | 4/24/2019 | 3.7.3 |
| 5 | Kelley | 4/24/2019 | 3.7.3 |
| 6 | Caroline | 4/24/2019 | 3.7.3 |
| 7 | Sang | 4/24/2019 | 3.7.3 |
| 8 | Caroline | 4/24/2019 | 3.7.3 |
| 9 | Caroline | 4/24/2019 | 3.7.3 |
| 10 | Stuart | 4/24/2019 | 3.7.3 |
| 11 | James | 4/26/2019 | 3.7.3 |
| 12 | James | 4/26/2019 | 3.7.3 |