

[illegible]

Section	Title	Pg Number
1	Scope.	3
1.1	Identification.	3
1.2	System overview..	3
1.3	Document overview..	3
2	Referenced documents.	3
3	Test preparations.	4
3.1	GOL Functional Testing Preparation	4
3.1.1	Hardware preparation.	4
3.1.2	Software preparation.	4
3.1.3	Other pre-test preparations.	4
4	Test descriptions.	4
4.1	GOL Functional Testing	4
4.1.1	Webpage Environment Testing	5
4.1.2	Pattern Customization Testing	5
4.1.3	Rule Customization Testing	7
4.1.4	Graphics Customization Testing	8
4.1.5	Gameflow Customization Testing	9
4.1.6	Grid Testing	11
4.1.7	Information Display Testing	12
5	Requirements traceability.	13
6	Notes.	14
A.	Appendixes.	14

1 Scope

1.1 Identification

This document describes the testing of a web application that simulates a customizable Conway's Game of Life. This application will run on the current version of chrome, 73.0.3683.103 and the current version of Firefox ESR 67.0+.

1.2 System overview

The purpose of this system is to provide users a customizable version of Conway's game of life in the form of a web application. This document applies to the web application. Additionally, the system shall provide the user with a means of customizing the appearance, speed, and functionality of Conway's Game of Life. This system will be developed and tested over a three-month period by a group of six. The operation of the system shall be accessible for the software sponsor, acquirer and user, Geoff Weiss, and Russell Cain. The development team does not have access to a support agency. This system is operable on any computer installed with the current releases of Chrome and Firefox ESR (defined in paragraph 1.1) and will be developed on the UMBC campus.

1.3 Document overview

This document describes the nature and testing of a web-based, customizable Conway's Game of Life application. This application will not record or store any personal information about the user. The system will not have any protection or privacy regarding the user's game information.

2 Referenced documents

This section shall list the number, title, revision, and date of all documents referenced in this document. This section shall also identify the source for all documents not available through normal Government stocking activities.

We will be using the open-source <https://github.com/blaze33/way-of-life> implementation as a start for development, this is licensed under MIT. The STD will refer to our development team's SRS for all of our test cases.

3 Test preparations

This section shall be divided into the following paragraphs. Safety precautions, marked by WARNING or CAUTION, and security and privacy considerations shall be included as applicable.

3.1 GOL Functional Testing Preparations

This segment of testing will describe the GOL (game of life) web application integration testing process. This series of tests will be performed on our web application. The tests will only focus on the functional requirements of the software noted in the SRS document.

3.1.1 Hardware preparation

To prepare the hardware, the team shall connect a personal computer to the internet. Next, we shall open both the earlier stated versions of Chrome and Firefox ESR web browsers. Finally, we'll open the web application on both browsers.

3.1.2 Software preparation

To prepare the software for testing, we have to make sure the C code is compiled correctly for our WASM engine. We will use the earlier stated versions of Chrome and Firefox ESR for testing. To load the software, proceed to the address of our web application.

4 Test descriptions

This section shall be divided into the following paragraphs. Safety precautions, marked by WARNING or CAUTION, and security and privacy considerations shall be included as applicable.

4.1 GOL Functional Testing

The following sections will describe each kind of testing we need to do for the requirements. They will be split into seven sections organized by the specific type of functionality or constraint that will be tested.

❖ 4.1.1 Webpage Environment Testing

Webpage Environment Testing refers to the testing of all environments which are required to support the web application.

4.1.1.1 Requirements addressed

These series of tests address all requirements in section 2.1 of the SRS.

4.1.1.2 Prerequisite conditions

Personal computing device must be connected to the internet and have the versions of Google Chrome and Firefox ESR listed in paragraph 1.1 installed, opened and displaying the webpage.

4.1.1.3 Test inputs

The Game of Life Webpage Environment Testing will not have any inputs. This portion of the testing is to evaluate if the page will load and be viewable on the listed version of Chrome, and the listed version of Firefox ESR, located in paragraph 1.1. This testing will be real and not simulated.

4.1.1.4 Expected test results

Both of the browsers displaying the entire page of our web application.

4.1.1.5 Criteria for evaluating results

Results will be evaluated as a success or failure and if applicable a partial success. The entire web page must be viewable on both web browsers for this to be a success. The pages are allowed to look different as long as both versions of web browsers are able to display the pages.

4.1.1.6 Test procedure

Connect personal computing device to the internet. Open both the version of Google Chrome and Firefox ESR listed in section 1.1. Open the web application with each of the browsers. Once the pages are loaded set the board to the gosper glider gun preset, screenshot both of the web pages to record test results, if the complete page is not viewable for a browser, repeat the test for that browser after an attempt to fix the issue.

❖ 4.1.2 Pattern Customization Testing

Pattern Customization Testing will refer to the testing of all requirements related to the customization of cells' starting position on the board.

4.1.2.1 Requirements Addressed

These series of tests address all requirements in section 2.2 of the SRS.

4.1.2.2 Prerequisite Conditions

Personal computing device must be connected to the internet and have either the version of Google Chrome or Firefox ESR listed in paragraph 1.1 installed, opened and displaying the webpage.

4.1.2.3 Test Inputs

Each repetition of the test requires for the game to be in the beginning state with default settings. The only inputs are the board presets which are the patterns that you can choose from which change the board's initial pattern. These presets will be from a set of ten which will include acorn, gosper glider gun, homer, eden, glider, cow, random, LWSS, frothing puffer and cord. There will also be sample files used to test the upload feature, this file will use a 100 by 35 grid for testing. There will be five different files used for the testing of the upload feature, three of which are text files in the correct format, one is a generic text file, and the last will be a non-text file. The test will evaluate the real performance of the application, no simulations are needed.

4.1.2.4 Expected Test Results

The board responds correctly for the random board feature, draw feature and preset board options. Uploading a file should display a pattern similar to what is in the text file, uploading a improper text file should do nothing, uploading a non text file should do nothing.

4.1.2.5 Criteria for Evaluating Results

The results are evaluated as success or failure. For a test to succeed, the game board will have to respond correctly to the option you choose. The board must generate a random board if the pseudo-random cell generation pattern is selected, for the random generation testing, multiple tests are required to evaluate if it is random. Since we are using five repetitions of the random board feature, then each of the boards generated are likely to be different. The board must let you draw on it to customize the positions of cells. The board must let you select between preset patterns, and it must respond to the selection made. Finally, the upload feature must generate a board pattern when inserting a file with the correct structure, there will be three test files and they all must look different for the test to be a success. The invalid files should not crash the website, nor should it make a pattern. Each of these requirements must have a separate test performed on them, and they must be evaluated individually.

4.1.2.6 Test procedure

Connect personal computing device to the internet. Open either the version of Google Chrome or Firefox ESR listed in section 1.1. Open the web application with the browser. Once the page is loaded, select each preset pattern to see if they work. After doing so, restart the game and select the random option five times to see if the pattern generation is random. Click on the board to test the drawing feature after clearing the board again. Finally, to test the upload features take each of the five sample files provided and upload them. Take note if each of

these features are responsive and behave correctly. If a test fails, attempt a fix and restart the test.

4.1.3.6 Assumptions and Constraints

If a setting was not mentioned for testing, then it should be assumed to be set to default.

◆ 4.1.3 Rule Customization Testing

Rule customization testing refers to the testing of all requirements related to the modifications of Conway's Game of Life rules. The testing will evaluate whether the rules work and if the game keeps those rules in place.

4.1.3.1 Requirements Addressed

These series of tests address all requirements in section 2.3 of the SRS.

4.1.3.2 Prerequisite Conditions

Personal computing device must be connected to the internet and have the versions of Google Chrome and Firefox ESR listed in paragraph 1.1 installed, opened and displaying the webpage.

4.1.3.3 Test Inputs

The test for the standard Game of Life rules will involve using the Gosper Glider Gun preset board. The testing for the custom settings will set the rules for all cells to live for one test, all cells to die for another and all cells to do nothing for the last test. The Gosper Glider Gun will also be used for testing the customized rules. There will also be certain cells that will be set to always alive or dead which will be explained further in the test procedure. All other settings should be the default if not mentioned. This testing will be real and involve direct use of the web application, no simulations will be used.

4.1.3.4 Expected Test Results

The standard rules should perform exactly like the original game, the Gosper Glider Gun board pattern should make it look like the gun is shooting. The customized rules shall be an empty board for testing the cell death. The board shall be full for the testing of the customized cell life setting. And the board should be the initial Gosper Glider Gun preset for the testing of the nothing setting, which should keep the initial board. For the permanently alive and dead cells, those cells should stay alive or dead regardless of current rules.

4.1.3.5 Criteria for Evaluating Results

These tests will be evaluated as a success or failure, and if applicable a partial success. All tests should work as described in section 4.1.3.4. If they do not have the proper displayed patterns, then it will be evaluated as a failure. If a permanently alive cell dies, then that feature

is a failure, if a permanently dead cell becomes live, then that feature is a failure. These tests will be evaluated on a success/failure metric and the basic rules will be evaluated independently from the customized rules. Tests can be deemed as a partial success if there is some problem observed in the implementation while still mainly working correctly.

4.1.3.6 Test procedure

Connect personal computing device to the internet. Open either the version of Google Chrome or Firefox ESR listed in section 1.1. Open the web application with the browser. Once the page is loaded, keep all settings default and use the Gosper Glider Gun preset pattern to test the standard rules for Conway's Game of Life, run the game for fifty generations and take note of the board state. Then, compare the board state to the expected one in section 4.1.3.4. Do three separate runs for the customized rules. First test setting all rules such that all cells die, run the game and the next iteration should have an empty board. Next, test setting all cells to live, and run the game, the board should be full. Last, test setting all cells to do nothing, the board should not change when running the game. For the testing of the permanently alive cell and the permanently dead cell, start a new game with the Gosper Glider Gun preset. Then set one of the inhabited cells on the square to the right to always be dead. Then set the cell in the top leftmost corner to always be alive. Run the game for fifty generations and take note if those cells stay alive or dead. Take note of the board states after finishing all of these tests.

4.1.3.7 Assumptions and Constraints

Test assumes that the default board size is used. If a setting was not mentioned for testing, then it should be assumed to be set to default.

❖ 4.1.4 Graphics Customization Testing

Graphics customization testing refers to all of the requirements which change the visual appearance of the board and cells. Requirements 2.4.1 and 2.4.2 will have a different test.

4.1.4.1 Requirements Addressed

These series of tests address all requirements in section 2.4 of the SRS, except for requirement 2.4.3, which cannot be tested for.

4.1.4.2 Prerequisite Conditions

Personal computing device must be connected to the internet and have either the version of Google Chrome or Firefox ESR listed in paragraph 1.1 installed, opened and displaying the webpage.

4.1.4.3 Test Inputs

The game shall have all default settings prior to the test unless stated otherwise. The inputs for the cell avatar requirements 2.4.1 will simply be the square avatar, the circle avatar, and the

triangle avatar. The inputs for the colored grid shall be red, live cells should be green, dead cells shall be blue. This test will be real and not a simulation, default rules for the game and the Gosper Glider Gun board should be selected for testing.

4.1.4.4 Expected Test Results

For testing the cell shape feature, the cells must respond to the option selected immediately. Changing the color of the grid, cells and dead cells should also respond when the option is selected.

4.1.4.5 Criteria for Evaluating Results

These tests will be evaluated as a failure or a success, or if applicable, a partial success. The testing of the cell avatar and the grid color will be evaluated independently. The avatar feature has to change the shape of live cells to the avatar you select. If the avatar takes more than a few seconds to change, or if it doesn't change to the one you select, then the test is a failure. Testing the color feature is similar. When selecting a color for the grid, live cells, or dead cells, the color has to change to the one you select and it must be immediate. If the color change is not correct, then the test is a failure.

4.1.4.6 Test procedure

There will be a different procedure testing for requirement 2.4.1 and requirement 2.4.2. For 2.4.1, start the game with the Gosper Glider Gun as the board preset pattern and change the avatar to a circle. Afterwards, change the avatar to a triangle, and back to square. If the shapes of the live cells changed, the test is complete and a success. If the shapes do not change for any reason, attempt to fix the solution and test again. For testing 2.4.2, start the game with the standard preset board. Change the grid background to be red, change the color of live cells to be green, and change the colors of the dead cells to be blue. Check and see if the colors are correct, if so, the test is a success and complete.

4.1.4.7 Assumptions and Constraints

If a game setting was not mentioned, it should be left at the default option and will not be considered for the test.

◆ 4.1.5 Gameflow Customization Testing

Gameflow customization testing refers to the testing of all of the requirements that relate to the customization of the current state of the game animations of the game.

4.1.5.1 Requirements Addressed

These series of tests address all requirements in section 2.5 of the SRS.

4.1.5.2 Prerequisite Conditions

Personal computing device must be connected to the internet and have either the version of Google Chrome or Firefox ESR listed in paragraph 1.1 installed, opened and displaying the webpage.

4.1.5.3 Test Inputs

The test for the gameflow customization tests the state of the game and shall have default settings prior to the test. The testing of the animation speed will have inputs for the game speed, the test will use 1 generation set per second and 60 generation sets per second (30 is the default). For the generation-skip testing, the inputs of skipping zero generations and one generation will be used. When we test for edge cases, negative numbers, blank input and the attempt of using letters will be used as well. For testing for a steady state, the acorn preset will be used.

4.1.5.4 Expected Test Results

The application responds to all actions correctly, pause, start a new game, and modifications of game speed. The generation skip counter should work as well, if improper input is used, it should functionally work like zero (applies to negative numbers, non-number characters, and blank inputs. When using the acorn preset, the game should eventually lead to a steady state at around generation 210.

4.1.5.5 Criteria for Evaluating Results

The results are evaluated as success or failure, or if applicable, a partial success. For a test to succeed, the game should respond to the option that the user chose in the correct nature. The application will be paused if the pause option is selected. The application must end the game when all cells are dead or if a steady state is detected. The speed of the game will change based on the speed the user shall select, 1 generation set per second must make the game show 1 set of generations each second. 60 generation sets per second must make the game show a maximum of 60 generation sets per second (backend computations can slow this down). For the generation skip option, the selection for generation skips must apply as the user selects it. To evaluate the generation skip feature, the generation skip of one has to skip every other generation to pass, the generation skip of zero has to skip none to pass. Any invalid input such as negative numbers, or non-integer characters, To check the steady state protection feature, run the preset pattern and the game will eventually detect it. If the preset does not generate a steady state after two minutes, then it is a failure

4.1.5.6 Test procedure

Connect personal computing device to the internet. Open either the version of Google Chrome or Firefox ESR listed in section 1.1. Open the web application with the browser. Once the page is loaded, keep all settings default. Start the game, then select the pause option to see if the game paused. Modify the speed between frames to 1 generation set per second by clicking on

the speed option slider and dragging it to the left side of the bar. Then drag the slider to the right side and take note if the speed changes to be around 60 generation sets per second. Then, check the generation skip feature, first set it to one. Take note if the game skips every other generation. After doing that set it back to zero and take note if the game skips generations or not. See if the changes have been made after a set. Click on start new game and take note if the game actually restarts. To test if the game ends with all cells dying, select the draw option for board generation and make sure the board is empty with standard rules. Starting a new game should make the current game end, take note if it does. To test the steady state feature, choose the acorn and let it run for two minutes, if it doesn't detect a steady state then it is a failure.

4.1.5.7 Assumptions and Constraints

The test assumes that the default speed between frames is used if not stated for a procedure. If a game setting was not mentioned, it should be left at the default option.

◆ 4.1.6 Grid Testing

Grid Testing refers to the testing of the requirements that relate to the customization of the size and functionalities of the game board.

4.1.6.1 Requirements Addressed

These series of tests address all requirements in section 2.6 of the SRS.

4.1.6.2 Prerequisite Conditions

Personal computing device must be connected to the internet and have the versions of Google Chrome and Firefox ESR listed in paragraph 1.1 installed, opened and displaying the webpage.

4.1.6.3 Test Inputs

The test of the grid shall have default settings prior to the test. There are two inputs, one is grid size which is the x and y fields that determine the size of the grid. The other is the wrap-around. We will use negative sizes and very large sizes to test the limitations of the system and logic.

4.1.6.4 Expected Test Results

For testing of the grid size feature, the grid must have as much boxes as $x * y$. The toggleable option, "wrap-around" must respond immediately when it's selected.

4.1.6.5 Criteria for Evaluating Results

The results are evaluated as success or failure, or if applicable, a partial success. For a test to succeed, the game should show a starting grid size in each iteration which is correlated to what you set it to. If the game does not change the grid size when another size is entered for x or y, then the test is a failure. For the testing of wrap-around, the game must treat cells at opposite

edges like they are neighbors. So if the cells on opposite borders have not become neighbors with the option on, the test is a failure.

4.1.6.6 Test procedure

Connect personal computing device to the internet. Open either the version of Google Chrome or Firefox ESR listed in section 1.1. Open the web application with the browser. Once the page is loaded, select a starting grid size. Start the game with a $x=50$ and $y=50$ grid size and take note of the actual size after starting the game. Then end that game and start a new one with $x=30$ and $y=30$, take note of the actual size of the grid after starting the game. To make sure the logic is correct, start another game with negative sizes and zero, and make sure there is no board. To test the limitations of the application, first test different values of x , increase the amount entered by 100 until the site or grid display breaks. Take note of the upper boundary the system allows. Upon encountering this, reduce the value of x until the website works normally again to get a good rough estimate of the maximum workable horizontal size of the grid. Next do the same with the y , while the x field is $x=50$. To test wrap around, select the Gosper Glider Gun preset and see if the gun shoots projectiles that form more than one line, make sure x and y are set to 100 during this to see this better, during this test, turn off wrap around and see if the bullets stop looping. Take note if the game behaves normally.

4.1.6.7 Assumptions and Constraints

The test assumes that grid size is using the default size. Test also assumes settings that aren't mentioned are left at default settings.

◆ 4.1.7 Information Display Testing

Information Display testing refers to the testing of all of the requirements related to the information about the current game that is displayed to the user.

4.1.7.1 Requirements Addressed

These series of tests address all requirements in section 2.7 of the SRS.

4.1.7.2 Prerequisite Conditions

Personal computing device must be connected to the internet and have the versions of Google Chrome and Firefox ESR listed in paragraph 1.1 installed, opened and displaying the webpage.

4.1.7.3 Test Inputs

The testing for these requirements will use a preset and pairs of dimensions for the board in the form of $x=\#$ and $y=\#$. Another input used will be the amount of generations that the game will run for. The board preset that will be used is the Gosper Glider Gun preset. The sizes used will be $x=100$ and $y=40$, $x=-50$, $y=-50$, $x=50$ and $y=-50$, $x=0$ and $y=50$, $x=$ and $y=10$. The different sizes will be used as different test cases to make sure the counters operate logically. The only

test where we take the generation counter into account is the one where we have $x=100$ and $y=40$ as the board dimensions. That test will use 50 generations as the duration of the game.

4.1.7.4 Expected Test Results

The board size for the 100 by 40 board should be 4000, the rest should display 0 for their size. The 100 by 40 board should have an initial live counter for cells of 36. After 50 generations, the game should have 59 live cells.

4.1.7.5 Criteria for Evaluating Results

The results are evaluated as a success or failure, or if applicable, a partial success. For a test to succeed, the application should show a counter that displays an accurate number of live cells. Also, another counter that should display the total number of the cells in the grid. Another counter is going to display the total number of generations that occurred since the beginning of the game. If any of the counters do not show up, they are failures, if they show up and illogically count things, they are partial successes.

4.1.7.6 Test procedure.

Connect personal computing device to the internet. Open either the version of Google Chrome or Firefox ESR listed in section 1.1. Open the web application with the browser. Once the page is loaded, choose the Gosper Glider Gun preset while the game is paused and set the board dimensions to $x=100$ and $y=40$. Compare the displayed text on live cells and total cells and see if they are correct. Start the game and pause it after 50 generations passed, take note of the live cell counter in the top left. Now, end the game by selecting different board sizes, first set the board to $x=-50$ and $y=-50$ and take note of the displayed statistics for live cells and total cells. Then set it to $x=-50$ and $y=50$ and take note of the displayed statistics. Then set it to $x=50$ and $y=0$ and take note of the displayed statistics. The last test will involve changing the board preset while the game is running, run a standard Gosper Glider Gun game for 50 generations, then try switching to the acorn preset. Take note if the generation counter gets set to zero or not.

4.1.7.7 Assumptions and Constraints

The test assumes default settings for any settings not mentioned.

5 Requirements traceability

Each test description listed in Section contains a “Requirements Addressed” subsection. The subsection identifies the Computer Software Configuration Item (CSCI) requirement that is addressed in the CSCI Software Requirements Specification (SRS) and is in the form of: “This series of tests addresses Conway’s Game of Life SRS requirement ##, where ## is the requirement number”.

6 Notes

6.1 Abbreviations

SRS - Software Requirement Specification

CSCI - Computer Software Configuration Item

Firefox ESR - Extended Support Release

A. Appendixes

DESCRIPTION/PURPOSE

The Software Test Description (STD) describes the test preparations, test cases, and test procedures to be used to perform qualification testing of a Computer Software Configuration Item (CSCI) or a software system or subsystem.

The STD enables the acquirer to assess the adequacy of the qualification testing to be performed.

APPLICATION/INTERRELATIONSHIP

Portions of this plan may be bound separately if this approach enhances their usability.

Examples include plans for software configuration management and software quality assurance.

The Contract Data Requirements List (CDRL) should specify whether deliverable data are to be delivered on paper or electronic media; are to be in a given electronic form (such as ASCII, CALS, or compatible with a specified word processor or other support software); may be delivered in developer format rather than in the format specified herein; and may reside in a computer-aided software engineering (CASE) or other automated tool rather than in the form of a traditional document.