Test Plan

for

Project 1

Prepared by Ryan Hiner, Dillen Kelton, Ricardo Reyna, Nicholas Shishkoff, and Kaylee Williams

CS3398

10/20/2020

1.1 Contents

1.2 Introduction

This document details the test plan for Team Lrr, Ruler of planet Omicron Persei 8's implementation of Conway's Game of Life (hereinafter referred to as Game of Life). This plan will cover system testing.

1.2.1 Test Plan Objectives

The purpose of this test plan is to provide a clear process to verify and validate all functions of Game of Life and confirm that all requirements have been met.

2.1 Test Strategy

2.1.1 System Test

The system as whole is tested to check for desired outputs and the user's experience with the application. The testers will make sure all of the functionalities of the UI components work in compliance to the requirements outlined in the SRS and SDD.

3.1 Environment Requirements

3.1.1 Environment 1

The Game of Life implementation will run on Windows 10 on a PC.

4.1 System Testing

4.1 Selecting Grid Size

- Condition 1: The game must always display the current size of the grid.
- Input: The user will execute the game.
- Expected Behavior: The system shall present the default grid size.
- Result: Failed. The game does not present the grid size at any state nor the button options to alter it.
- Condition 2: While the simulation is not running. The system always provides button options to increase or decrease the grid size within a minimum and maximum size.
- Input: The user will execute the game.
- Expected Behavior: The grid size shall change to the desired size.
- Result: Failed. The user is never presented with the button options to alter the grid size nor the value of the current grid size.

REQ 4.1.2a

- Condition: The system must provide the user 4 buttons to modify the axis' size.
- Input: The user will select each button.
- Expected Behavior: The grid size shall change by the value the button is labeled as.
- Result: Failed. The user is never presented with the button options to alter the grid size.

REQ 4.1.2b

- Condition: While the simulation is running the buttons must be greyed out and clicking on the shall not alter the state of the system.
- Input: While the simulation is running the user will try to select all of the grid size buttons.
- Expected Behavior: The grid size buttons shall not be able to be press and the grid size shall remain the same.
- Result: Failed. The user is never presented with the button options to alter the grid size nor the value of the current grid size.

REQ 4.1.2c

- Condition 1: If the maximum size for an axis has been reached, the buttons that increase the grid size must be greyed out and unable to alter the grid size.
- Input: The user will try to select the buttons that increase the grid size.
- Expected Behavior: The grid size buttons shall not be able to be press and the grid size shall remain the same.
- Result: Failed. The user is never presented with the button options to alter the grid size nor the value of the current grid size.

- Condition 2: If the minimum size for an axis has been reached, the buttons that increase the grid size must be greyed out and unable to alter the grid size.
- Input: The user will try to select the buttons that decreased the grid size.
- Expected Behavior: The grid size buttons shall not be able to be pressed and the grid size shall remain the same.
- Result: Failed. The user is never presented with the button options to alter the grid size nor the value of the current grid size.

REQ 4.1.2d, e, and f

- Condition 1: The game is at the default grid size.
- Input: The user clicks on the “+1” button.
- Expected Behavior: The grid size will increase by one.
- Result: Failed. The game does not present button options to alter the grid size.

- Condition 2: The game is at the default grid size.
- Input: The user clicks on the “-1” button.
- Expected Behavior: The grid size will decrease by one.
- Result: Failed. The game does not present button options to alter the grid size.

- Condition 3: The game is at the default grid size.
- Input: The user clicks on the “+5” button.
- Expected Behavior: The grid size will increase by one.
- Result: Failed. The game does not present button options to alter the grid size.

- Condition 4: The game is at the default grid size.
- Input: The user clicks on the “-5” button.
- Expected Behavior: The grid size will decrease by one.
- Result: Failed. The game does not present button options to alter the grid size.

REQ 4.1.2g

- Condition 1: The game is at the maximum grid size.
- Input: The user tries to click on the “+5” button.
- Expected Behavior: The user is not able to select the button nor to alter the grid size
- Result: Failed. The game does not present button options to alter the grid size.

- Condition 2: The game is at the maximum grid size.
- Input: The user tries to click on the “+1” button.
- Expected Behavior: The user is not able to select the button nor to alter the grid size
- Result: Failed. The game does not present button options to alter the grid size.

REQ 4.1.2h

- Condition 1: The game is at the minimum grid size.
- Input: The user tries to click on the “-5” button.
- Expected Behavior: The user is not able to select the button nor to alter the grid size
- Result: Failed. The game does not present button options to alter the grid size.

- Condition 2: The game is at the minimum grid size.
- Input: The user tries to click on the “-1” button.
- Expected Behavior: The user is not able to select the button nor to alter the grid size
- Result: Failed. The game does not present button options to alter the grid size.

4.2 Selecting Cell Colors

- Condition: The game must be at the default state or paused.
- Input: The user will select unique colors for a live cell, dead cell, and an empty cell.
- Expected Behavior: The color of each state of the cells shall change to the selected one.
- Result: Passed.

REQ 4.2.2b

- Condition: The game must be at the default state or paused.
- Input: The user will select the same colors for a live cell, dead cell, and an empty cell.
- Expected Behavior: The color of each state of the cells shall not be able to change to the selected one.
- Result: Passed.

REQ 4.2.2c

- Condition 1: The game must not be at stage zero.
- Input: The user will try to change the colors of the cells.
- Expected Behavior: The color of each state of the cells shall not be able to change to the selected one.
- Result: Failed. All types of cells can change color after stage zero.

- Condition 2: The simulation must be running.
- Input: The user will try to change the colors of the cells.
- Expected Behavior: The color of each state of the cells shall not be able to change to the selected one.

- Result: Passed.

4.3 Adding or Removing Live Cells from Grid

REQ 4.3.2a and REQ 4.3.2b.III

- Condition: The game must be open and at stage zero or paused.
- Input: The user will click on an empty cell in the grid where the user wants a live cell to spawn.
- Expected Behavior: Upon clicking, the cell must become alive.
- Result: Passed.

REQ 4.3.2b

REQ 4.3.2b.I

- Condition: The grid must have at least one live cell.
- Input: The user will click on a live cell.
- Expected Behavior: Upon clicking, the cell must become a dead cell.
- Result: Passed.

REQ 4.3.2b.II

- Condition: The grid must have at least one dead cell.
- Input: The user will click on a dead cell.
- Expected Behavior: Upon clicking, the cell must become an empty cell.
- Result: Passed.

REQ 4.3.2c

- Condition 1: The game must be at any stage that is not zero and not paused.
- Input: The user will click on any type of cell in the grid where the user wants a cell state to change.
- Expected Behavior: Upon clicking on a cell, nothing shall happen.
- Result: Passed.

- Condition 2: The simulation must be running.
- Input: The user will click on any cell.
- Expected Behavior: Upon clicking on a cell, nothing shall happen.
- Result: Passed.

4.4 Selecting Preset Grids

REQ 4.4.2a

- Condition: A preset pattern must be selected.
- Input: The user clicks the play button to run the simulation.
- Expected Behavior: While the simulation is running, the preset patterns shall follow Conway's Game of Life rules.
- Result: Passed.

REQ 4.4.2b

- Condition: The program must provide the user with selectable different patterns at stage zero or when the simulation is paused.
- Input: The user will click the arrow buttons to look through all available preset patterns.
- Expected Behavior: The system shall provide a UI which will enable the user to select a preset.
- Result: Passed.

REQ 4.4.2c and REQ 4.4.2d

- Condition: The program must provide the user with selectable different patterns at stage zero or when the simulation is paused.
- Input: The user will click the arrow buttons to look through all available preset patterns.
- Expected Behavior: The system shall provide a UI which will enable the user to select a preset.
- Result: Passed.

REQ 4.4.2e

- Condition: The simulation must be playing.
- Input: The user will try to select a pattern.
- Expected Behavior: The system shall render the preset button non-functional.
- Result: Passed.

4.5 Starting the Simulation

- Condition: The game must be open.
- Input: The user will click on the "Play" button.
- Expected Behavior: The game will start the life, death, or growth simulation of user-selected cells based on Conway's Game of Life.
- Result: Passed.

REQ 4.5.2a

- Condition: The simulation must be running.

- Expected Behavior: The system shall allow for the “Pause” and playback speed features to remain available to be selected from the user if so desired.
- Result: Passed.

REQ 4.5.2b

- Condition: The simulation must be running.
- Expected Behavior: The program shall run continuously, without any additional input required from the user.
- Result: Passed.

REQ 4.5.2c

- Condition: The simulation must be in the paused state.
- Expected Behavior: The game shall allow the user to resume the simulation from the state at which the game is paused.
- Result: Passed.

REQ 4.5.2d

- Condition: The simulation must be running.
- Expected Behavior: The system shall display in real-time the visual life, death, or growth of cells based on user-selected starting positions of live cells.
- Result: Passed.

4.6 Pausing the Simulation

REQ 4.6.2a

- Condition: The game must have recently been opened.
- Expected Behavior: The program will begin the system in a paused state.
- Result: Passed.

REQ 4.6.2b

- Condition: The simulation must be running.
- Input: The user click the “Pause” button
- Expected Behavior: The system shall allow the user to pause a simulation in progress, halting the “Play” operation.
- Result: Passed.

REQ 4.6.2c

- Condition: While the simulation is paused.

- Expected Behavior: The system shall allow the user to select new live cells while in a paused state if so desired.
- Result: Passed.

REQ 4.6.2d

- Condition: While the simulation is paused and there are no live cells in the grid.
- Input: The user will press the “Play Button”.
- Expected Behavior: The system shall automatically pause if no live cells are selected by the user.
- Result: Failed. The simulation is still running. The stage counter keeps increasing.

REQ 4.6.2e

- Condition: While the simulation is paused and there are live cells in the grid.
- Input: The user will press the “Play Button”.
- Expected Behavior: The system shall automatically pause given the situation in which all live cells have died and there is no further life or growth to simulate.
- Result: Failed. The simulation is still running. The stage counter keeps increasing.

4.7 Altering the Speed of the Simulations

- Condition: The game must have recently been opened.
- Expected Behavior: The system shall allow variable playback speeds as available options for the user.
- Result: Failed. The simulation is still running. The stage counter keeps increasing.

REQ 4.7.2a

- Condition: The simulation must be running.
- Input: The user will select the “.5X” speed button.
- Expected Behavior: The system shall operate as if the “Play” function has been selected by the user. While “.5X” is selected, the system shall display the visual simulation at half the rate as would be
- displayed by the “Play” function
- Result: Passed

REQ 4.7.2b

- Condition: The simulation must be running.

- Input: The user will select the “2X” speed button.
- Expected Behavior: The system shall operate as if the “Play” function has been selected by the user. While “2X” is selected, the system shall display the visual simulation at double the rate as would be displayed by the “Play” function.
- Result: Passed

REQ 4.7.2c

- Condition: The simulation must be running.
- Input: The user will alternate between the half speed and double speed.
- Expected Behavior: The system shall allow for the user to freely switch between “.5X,” “Play,” and “2X,” features, so long as the game is in a state in which the “Play” function may operate
- Result: Passed

4.8 Simulating a Single Generation (next)

- Condition: While playing the game, the user must have the option to select the “Next” button.
- Input: The user will select the “Next” button.
- Expected Behavior: The system shall simulate the next generation, which consists of new live or dead cells.
- Result: Passed

4.8.2 Functional Requirements

REQ 4.8.2a

- Condition: In the game make a living cell that has fewer than two live neighbors.
- Input: The user will click on the grid and create only one living cell.
- Expected Behavior: The live cell dies and is not present in the next generation
- Result: Passed

REQ 4.8.2b

- Condition: In the game creating a live cell that has more than three live neighbors.
- Input: The user will click on the grid and create 3 living cells that are neighbors
- Expected Behavior: The live cells die and are not present in the next generation.
- Result: Passed

REQ 4.8.2c

- Condition: In the game creating a live cell that has exactly two or three live neighbors.
- Input: The user will click on the grid and create a living cell that has two or three neighbors
- Expected Behavior: The live cell stays the same in the next generation.
- Result: Passed

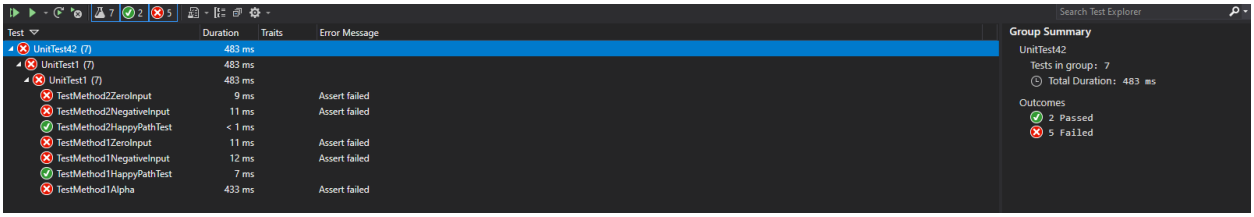
REQ 4.8.2d

- Condition: In the game creating a live cell that has exactly three live neighbors.
- Input: The user will click on the grid and create 3 living cells that are neighbors
- Expected Behavior: The dead cell will come to life and will be present in the next generation.
- Result: Passed

4.9 Resetting the Simulation

- Condition: While the simulation is running.
- Input: The user will click on the “Reset” button.
- Expected Behavior: The system shall halt the “Play” operation, and shall revert to the default state of the game.
- Result: Passed

5.0 Unit Tests



5.1 setWidth()

5.1.1 Happy Path Test

- Input: 1
- Expected Behavior: width will be set to 1.
- Result: Passed

5.1.2 Numerical Zero Input

- Input: 0
- Expected Behavior: Error will be thrown
- Result: failed

5.1.3 Alpha Input

- Input: ‘a’
- Expected Behavior: Error will be thrown
- Result: failed

5.1.4 Negative Input

- Input: -10
- Expected Behavior: Error will be thrown
- Result: failed

5.2 setHeight()

5.2.1 Happy Path Test

- Input: 1
- Expected Behavior: height will be set to 1.
- Result: Passed

5.2.3 Numerical Zero Input

- Input: 0
- Expected Behavior: Error will be thrown
- Result: failed

5.2.3 Negative Input

- Input: -10
- Expected Behavior: Error will be thrown
- Result: failed