

# CMSC 447

## Software Requirements Specification (SRS)

| Name             | Role  | Signature |
|------------------|---|-----------|
| Mohamed Alkindi  | Point of Contact,<br>Documentation Writer                           |           |
| Eoin Fitzpatrick | Software Developer,<br>Documentation Writer,<br>Program Tester      |           |
| Alex Flaherty    | Lead Software Developer,<br>Documentation Writer,<br>Program Tester |           |
| Evan Hart        | Documentation Writer  |           |
| Gabriel Kilungya | Documentation Writer,<br>Program Tester                             |           |
| Hannah Russell   | Documentation Writer,<br>Program Tester                             |           |

|       |   |   |
|-------|---|---|
| 1     | Scope   | 3 |
| 1.1   | Identification                                      | 3 |
| 1.2   | System overview                                     | 3 |
| 1.3   | Document overview                                   | 3 |
| 2     | Requirements  | 4 |
| 2.1   | Required states and modes                           | 4 |
| 2.2   | CSCI capability requirements                        | 4 |
| 2.2.1 | CSCI capability                                     | 4 |
| 2.3   | CSCI environment requirements                       | 5 |
| 2.4   | Computer resource requirements                      | 5 |
| 2.4.1 | Computer hardware requirements                      | 5 |
| 2.4.2 | Computer hardware resource utilization requirements | 5 |
| 2.4.3 | Computer software requirements                      | 5 |
| 2.5   | Design and implementation constraints               | 5 |
| 3     | Qualification provisions                            | 5 |
| 4     | Requirements traceability                           | 6 |

# 1 Scope

## 1.1 Identification

Team Rocket's Game of Life, Team 3

Version 1, first release

Identifications:

Virus: Infectious system designed for gameplay interactions

Neighbor: The closest squares around the current square

Seed: Number used to generate pseudorandom board layouts

.exe: Windows OS executable program

GUI: Graphical user interface

Left click: User mouse left-click button

Right click: User mouse right-click button

## 1.2 System overview

The purpose of this software is to provide the user with an enhanced experience with Conway's Game of Life. This extension of the game adds features such as generating viruses, alongside living cells and seeing how each living cell and virus thrive or die together in each generation. In addition, users can randomly generate board configurations, change the game speed, and load or save board configurations.

The game comes packed into a single .exe which can be run on windows environments. The user is able to run the application on Windows systems installed with .NET framework 4+.

Team Rocket's Game of Life is sponsored by Ms. Dorothy Kirlew. This project is intended to be used by the public, and is currently being developed by Team Rocket. The open source code is available on github at the link <https://github.com/hrussell898/CMSC447>.

## 1.3 Document overview

Team Rocket's Game of Life is an open source project. This SRS document provides development details, workflow pathways, and various design methodologies for Team Rocket's Game of Life.

## **2 Requirements**

### **2.1 Required states and modes**

The game is a single .exe file, therefore it can only operate in run mode.

### **2.2 CSCI capability requirements**

#### **2.2.1 CSCI capability**

2.2.1.1 The game shall have a GUI

2.2.1.2 The game shall follow the rules listed in Conway's Game of life.

1. A cell with one or no neighbors shall die.
2. A cell with more than four neighbors shall die.
3. A cell with 2-3 neighbors shall live.
4. A cell with 3 neighbors shall become populated.

2.2.1.3 The game shall generate viruses.

1. A virus shall populate one neighbor at a time.
2. A virus with no non-infected neighbors shall die returning to a blank cell.
3. A virus shall infect one neighbor at random, if it has more than one non-infected neighbor.

2.2.1.4 The game shall have a step button that advances the cells forward one generation.

2.2.1.5 The game shall have a run button that continuously advances to the cells next generation.

2.2.1.6 The user shall be able to turn on/off healthy cells and viruses

1. The Game shall have a right click button to generate a virus, which is represented by a green square.
2. The Game shall have a left click button to generate a healthy cell, which is represented by a black square.

2.2.1.7 Game shall have a speed slider that adjusts the speed each generation is generated at.

2.2.1.8 Game shall have the ability to save a current boards state.

2.2.1.9 Game shall have the ability to load a board.

2.2.1.10 User shall be able to change the board size.

1. The minimum dimensions of the board shall be 10 x 10.
2. The maximum dimensions of the board shall be 60 x 140.

2.2.1.11 The game shall have an option to generate cells automatically.

## **2.3 CSCI environment requirements**

- 2.3.1** The program shall run on a computer with Windows operating system.
- 2.3.2** The resolution shall scale with the size of the game board.
- 2.3.3** The user shall have a pointing device such as a mouse, touch screen, keyboard.

## **2.4 Computer resource requirements**

### **2.4.1 Computer hardware requirements**

- 2.4.1.1 The computer shall have an input interface from a pointing device.
- 2.4.1.2 The computer shall have secondary storage in order to save current stage.

### **2.4.2 Computer hardware resource utilization requirements**

- 2.4.2.1 The operating system's requirements are sufficient to accommodate the project's resource needs in a multitasking environment.

### **2.4.3 Computer software requirements**

Given a recent window distribution's ability to accommodate the C# executable, the distribution must provide the project's required facilities.

## **2.5 Design and implementation constraints**

- 2.5.1** The CSCI shall be compiled into a Windows runnable executable file, .exe.

## **2.6 Precedence and criticality of requirements**

All requirements for Team Rocket's Game of Life have equal weight.

## **3 Qualification Provisions**

- a. Demonstration: The operation of the CSCI, or a part of the CSCI, that relies on observable functional operation not requiring the use of instrumentation, special test equipment, or subsequent analysis.
- b. Test: The operation of the CSCI, or a part of the CSCI, using instrumentation or other special test equipment to collect data for later analysis.
- c. Inspection: The visual examination of CSCI code, documentation, etc.

| Requirement | Type of Testing |
|-------------|-----------------|
| 2.2.1.1     | Inspection      |
| 2.2.1.2     | Demonstration   |
| 2.2.1.3     | Demonstration   |
| 2.2.1.4     | Demonstration   |
| 2.2.1.5     | Test            |
| 2.2.1.6     | Test            |
| 2.2.1.7     | Demonstration   |
| 2.2.1.8     | Test            |
| 2.2.1.9     | Test            |
| 2.2.1.10    | Demonstration   |
| 2.2.1.11    | Demonstration   |
| 2.3.1       | Inspection      |
| 2.3.2       | Demonstration   |
| 2.3.3       | Inspection      |
| 2.4.1.1     | Inspection      |
| 2.4.1.2     | Inspection      |
| 2.4.3       | Inspection      |
| 2.5.1       | Inspection      |

## 4 Requirements traceability

All of our requirements were derived from our customer, Ms. Dorothy Kerlew, with the exception of the following:

**Requirement 2.2.1.2 is derived from the original rules of Conway's Game of Life.**

**Requirements 2.2.1.3.1 - 2.2.1.3.3 are derived from requirement 2.2.1.3.**

**Requirements 2.2.1.6.1 and 2.2.1.6.2 are derived from requirement 2.2.1.6.**

**Requirements 2.2.1.10.1 and 2.2.1.10.2 are derived from requirement 2.2.1.10.**

