CMSC 447

Software Design Description (SDD)

Name	Role	Signature

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. 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 UMBC campus.

2 Referenced documents

Document	Date
SRS - Software Requirements Specification	04/29/2019

3 CSCI-wide design decisions

See SRS document.

4 CSCI architectural design

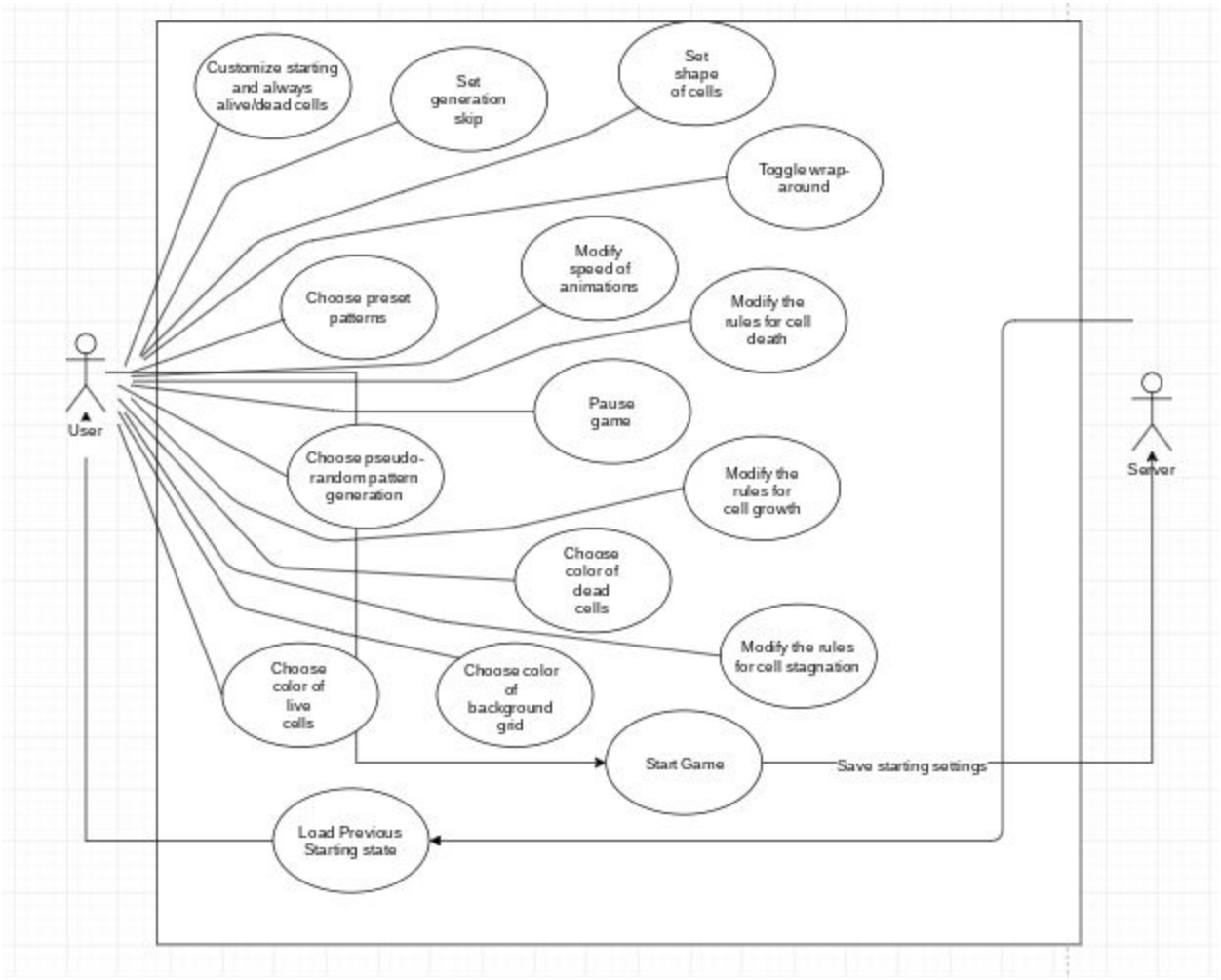
4.1 CSCI components

Software Unit	ID	Purpose	Development Status	Requirement Allocation
Webpage Environment	01	Allows the user to access the application from Firefox current ESR branch browser and/or Chromes current stable release browser.	Recycled from open source implementation	2.1.1 2.1.1.1 2.1.1.2
Pattern Customization	02	Allows the user to customize starting cells by choosing from preset patterns, pseudo-random pattern generation and drawing on the grid.	New Development	2.2.1 2.2.1.1 2.2.1.2 2.2.1.3 2.2.14
Rules Customization	03	Allows user to customize rules by giving them the ability to choose between default Conway's Game of Life rules and their own custom ruleset.	New Development	2.3.1 2.3.1.1 2.3.1.2 2.3.1.3 2.3.1.4 2.3.1.5 2.3.1.6
Graphics Customization	04	Allows the user to customize application graphics by giving them the ability to choose their cell's avatar, and grid display color.	New Development	2.4.1 2.4.1.1 2.4.1.2 2.4.1.3 2..4.2 2.4.1.3 2.4.2 2.4.2.1 2.4.2.2 2.4.2.3 2.4.3
Gameflow Control	05	Allows the user to control the Gameflow by modifying the speed, skipping frames, pausing and starting a new game.	Recycled from open source implementation	2.5.1 2.5.2 2.5.3 2.5.4

				2.5.5 2.5.6
Grid	06	Allows the user to modify the grid size and “wrap around” functionality.	Recycled from open source implementation	2.6.1 2.6.2
Displaying Information	07	Displays game information on the screen.	Recycled from open source implementation	2.7.1 2.7.2 2.7.3 2.7.4 2.7.4.1 2.7.4.2 2.7.4.2.1 2.7.4.2.2 2.7.4.2.3 2.7.4.2.4 2.7.4.3

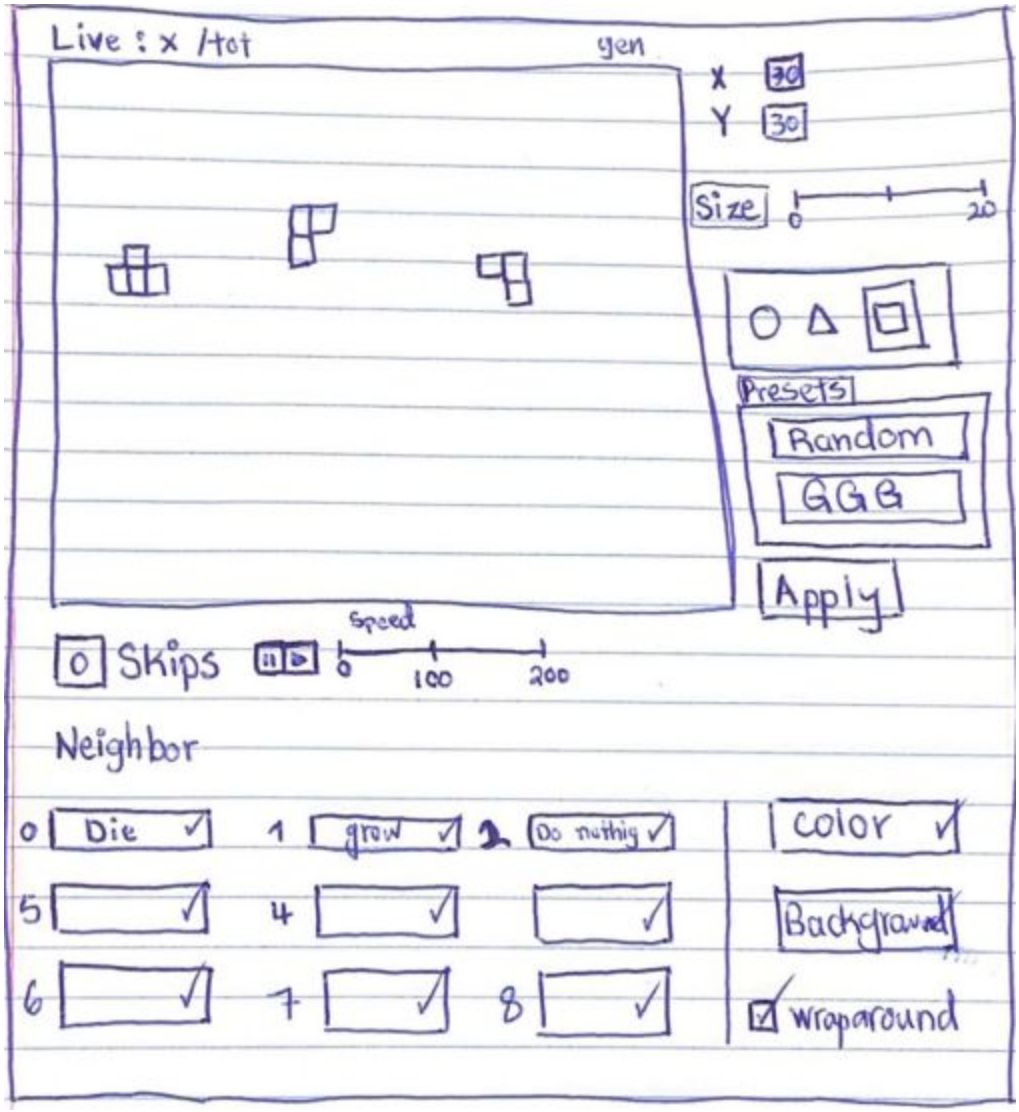
4.2 Concept of execution

The following UML diagram shows the possible scenarios that can be performed by actors in the system



4.3 Interface design

The following diagram shows the planned user interface for this system.



5 CSCI detailed design

5.1 (Project-unique identifier of a software unit, or designator of a group of software units)

Software Unit	ID	Constraints/Special Features	Programming Language	Inputs /Outputs
Webpage Environment	01	Chrome and Firefox	JS & WASM	N/A
Pattern Customization	02	N/A	JS	**Inputs:** 2.2.3 - Pattern from Mouse Strokes **Outputs:** 2.2.1.3 - Image of custom starting cells drawing
Rules Customization	03	N/A	JS & WASM	**Inputs**: 2.3.1.1 - 2.3.1.3 - Number of adjacent alive cells required to die, survive or be born. 2.3.1.4-2.3.1.6 - Specified cells **Outputs**: N/A
Graphics Customization	04	N/A	JS	**Inputs**: Customization choice **Outputs:** Customized avatar/cells
Gameflow Control	05	N/A	JS & WASM	**Inputs:** Gameflow control preferences **Outputs**: Modified Gameflow

Grid	06	N/A	JS & WASM	**Inputs:** 2.6.1 - Grid starting size 2.6.2 - Toggle preference **Outputs** 2.6.1 - Modified grid 2.6.2 - N/A
Displaying Information	07	N/A	JS & WASM	**Inputs**: N/A **Outputs** Displayed game information.

6. Requirement traceability

All requirements are traceable directly to the customer unless stated otherwise.

7 Notes

7.1 Abbreviations

Firefox ESR - Firefox Extended Support Release

UML - Uniform Modeling Language

WASM - WebAssembly

JS - Javascript