**Table of Contents**

**Introduction2**

Purpose2

References2

**Overall Description3**

Product Perspective3

Product Functions3

User Classes and Characteristics5

Operating Environment5

**External Interface Requirements6**

**System Features19**

Initializing the Application19

Starting a Game19

Rules20

Explanation of Functionality20

Exiting the Game21

Multiplayer mode21

**Other Nonfunctional Requirements22**

Performance Requirements22

Software Quality Attributes22

**Other Requirements22**

Appendix A: Glossary22

**1.Introduction**

**1.1 Purpose**

Decapod-10 is based off of the **Game of Life**, also known simply as **Life**, is a cellular automaton devised by the British mathematician John Horton Conway in 1970. It is a zero-player game, meaning that its evolution is determined by its initial state, requiring no further input. It is Turing complete and can simulate a universal constructor or any other Turing machine. (Wikipedia)

The first goal is the game shall function according to the rules established by [John Conway](https://www.conwaylife.com/wiki/Conway%27s_Game_of_Life). The second goal is to provide a stand-alone program that utilizes a graphical user interface. The third goal is to allow the user to create an initial configuration and launch the program. The final goal will be to provide a working program that allows the user(s) to configure a pattern and the program will playout Conway’s game of life.

**1.2 References**

Website dedicated to John Conway and his game of Life.

<https://www.conwaylife.com/wiki/Conway%27s_Game_of_Life>

Wikipedia’s public site for John Conway <https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life>

Cornell University, Cornell Math Explorer’s Club Website

<http://pi.math.cornell.edu/~lipa/mec/lesson6.html>

Web Based game example of John Conway’s game of Life

<https://bitstorm.org/gameoflife/>

**2. Overall Description**

**2.1 Product Perspective**

**Description**

The *Decapod-10* system shall attempt to replicate the *Game of Life,* by John Horton

Conway,by following the four basic rules of the game.

**Rules**

1. Any live cell with fewer than two live neighbors dies, as if by underpopulation.
2. Any live cell with two or three live neighbors lives on to the next generation.
3. Any live cell with more than three live neighbors dies, as if by overpopulation.
4. Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

**2.2 Product Functions**

Main menu

The system shall display a main menu

Start Game

The system shall allow the player to start the game

Game Rules

The system shall provide a way to display the game rules

Exit Game

The system shall provide a way for the player to exit game

Game Play

The system shall allow the player/s to interact with the game

Cell Locations

The system shall provide a way for the user to decide the starting positions of the cells

Iteration Marker

The system shall display the number of iterations processed

Edit

The system shall allow the user to edit current cell population

New Game

The system shall provide a way for user to start a new game session

Stop Game

The system shall provide a way for user to stop at the current iteration

Iterate

The system shall allow the player to traverse one iteration at a time

Iteration Speed

The system shall allow the player to select the iteration speed

Grid Size

The system shall allow the player some control over the size of the grid

Color/Sprite Selection

The system shall allow the player to select the color of the initial cell

Other Possible Functions

PvP

The system shall allow multiple players

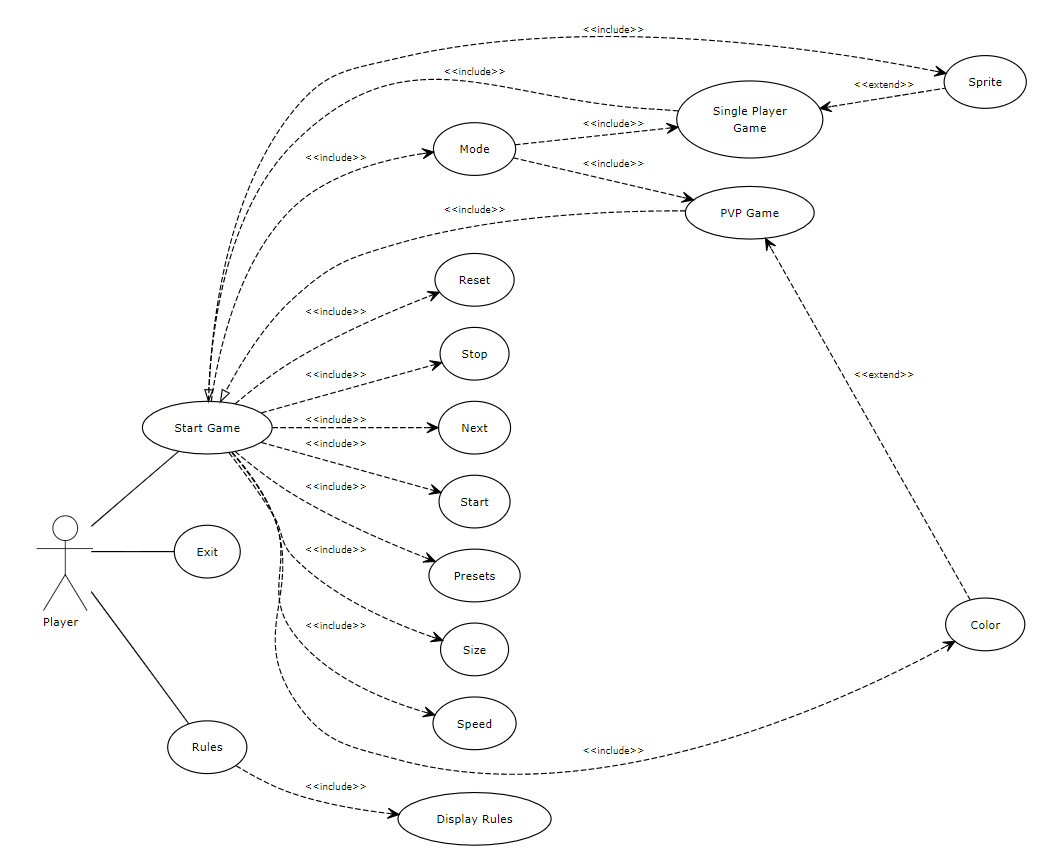
Winner

The system shall be able to display the winning player

Presets

The system shall allow the player to select from a list of preset starting positions

**2.3 User Classes and Characteristics**

****

**2.4 Operating Environment**

Decapod-10 shall be fully functional on any *Windows* or *Mac* operating systems.

**3. External Interface Requirements**

Conway’s Game of Life’s interface will contain a menu where players can learn about game play and functionality provided by the interface. On selecting ‘Start Game’ (via the Start Game button on the menu or after the functionality section), players will be presented a grid with several toggles on the side that they can utilize. The grid will allow players to click on any square to activate the cell and once the game has started, the grid will be updated based on the initial activated cells and toggles selected from the side.

● Main Menu (Figure 3.1.a)

● Rules (Figure 3.1.b)

● Functionality (Figure 3.1.c)

● Game/Reset Screen (Figure 3.2)

● Main Controls (Figure 3.2.a)

● Iterations (Figure 3.2.b)

● Speed (Figure 3.2.c)

● Resize (Figure 3.2.d)

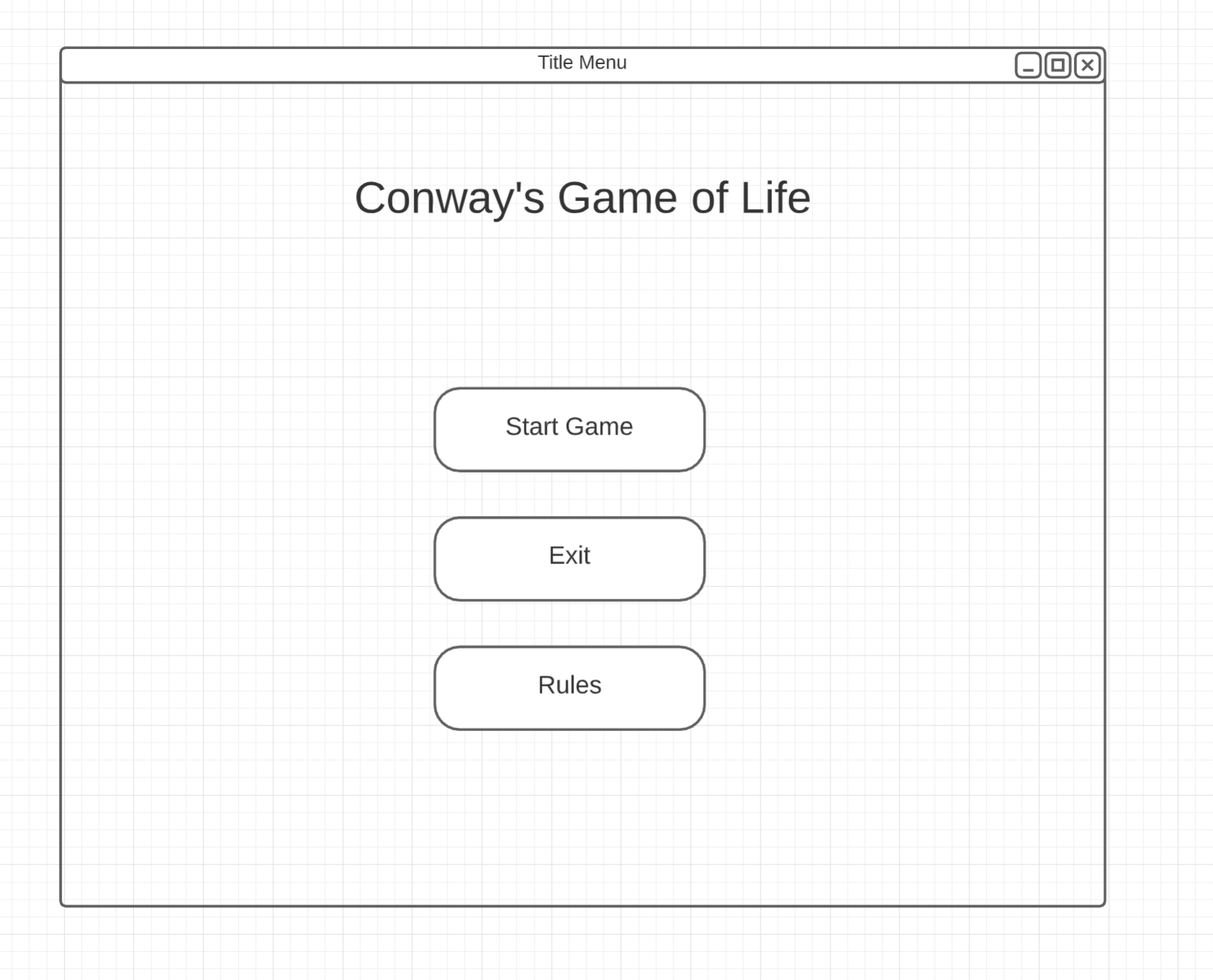
● Preset (Figure 3.2.e)

● Sprite (Figure 3.2.f)

● PvP (Figure 3.2.g)

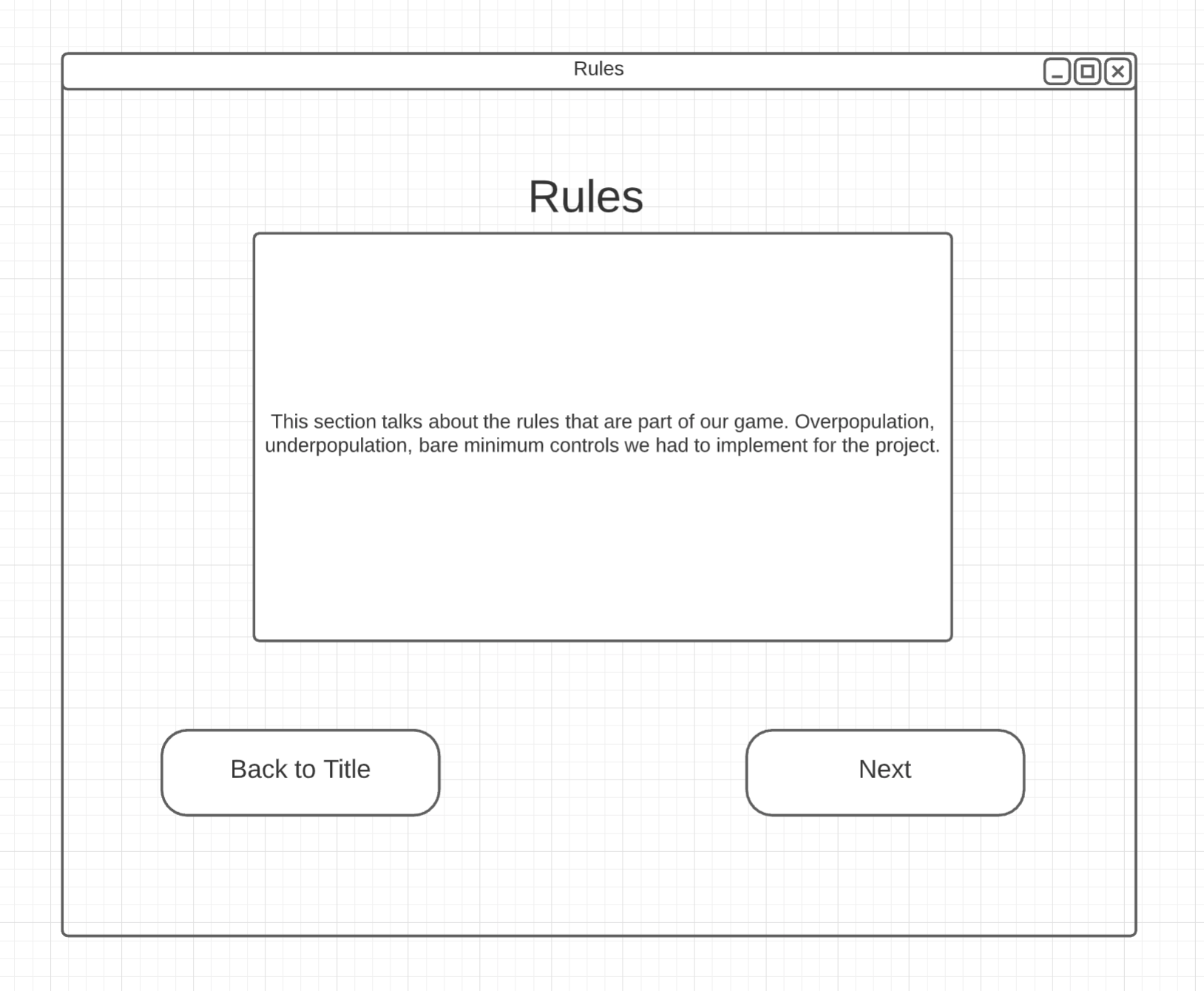
● Colors (Figure 3.2.h)

● Winning (Figure 3.2.i)

****

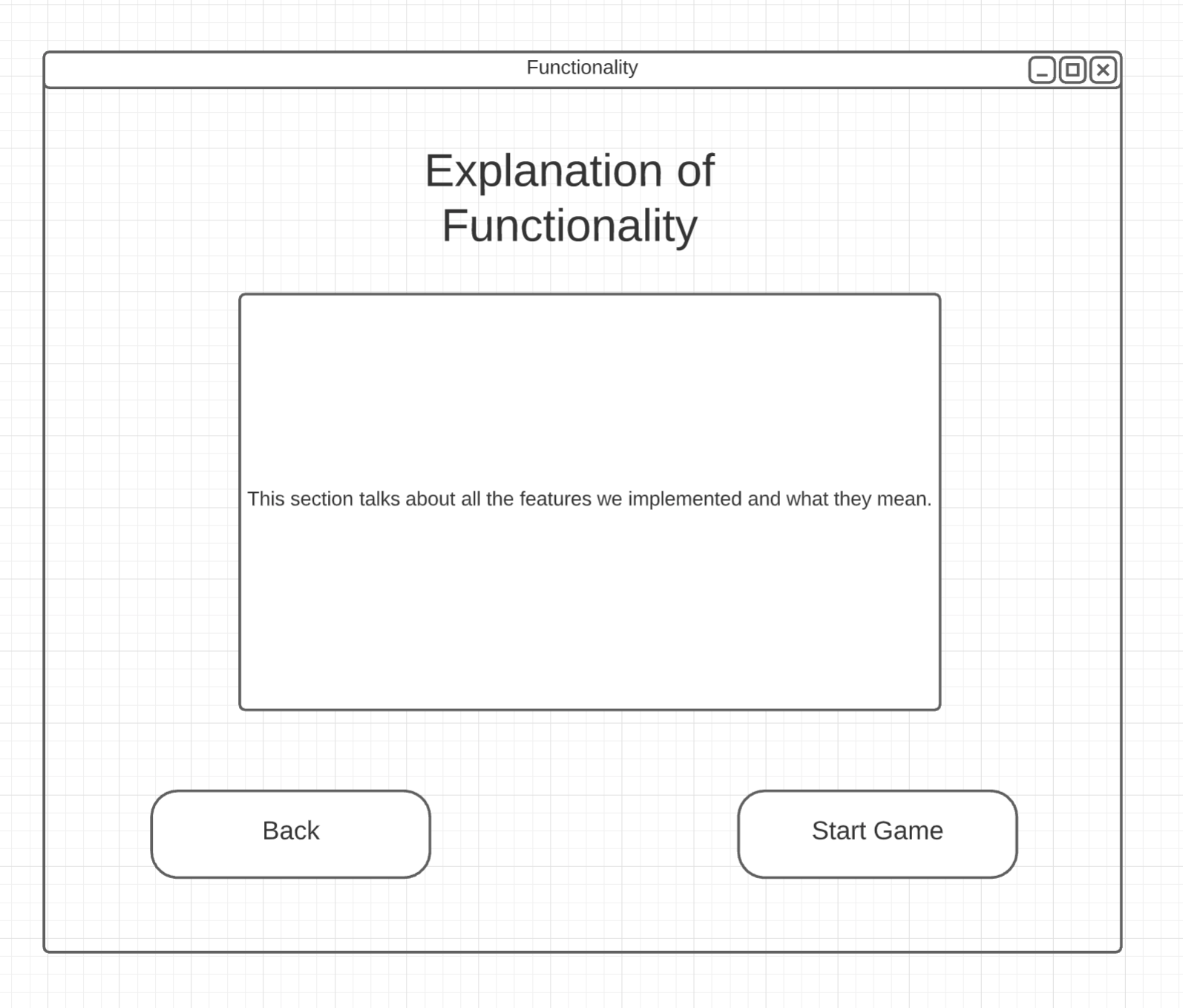
**Figure 3.1.a (Main Menu)**

The system shall display Start Game, Exit, and Rules.

****

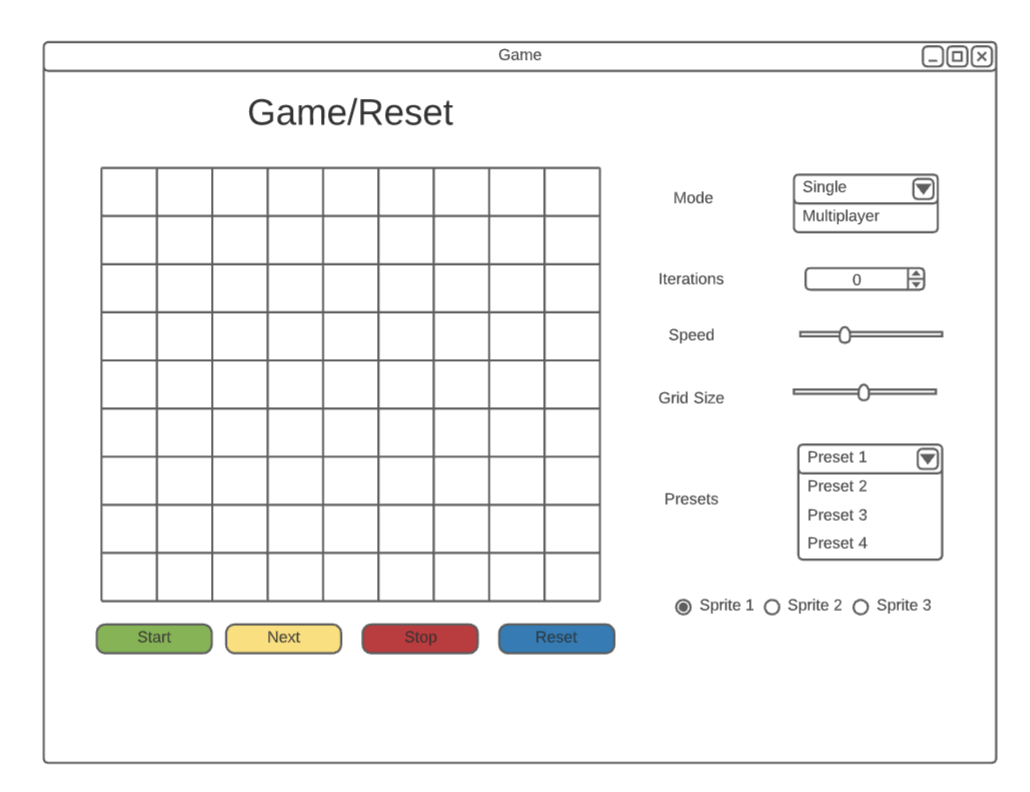
**Figure 3.1.b (Rules)**

The system shall show the rules page which explains how the game of life works (based on the Rules portion of Section 2).

****

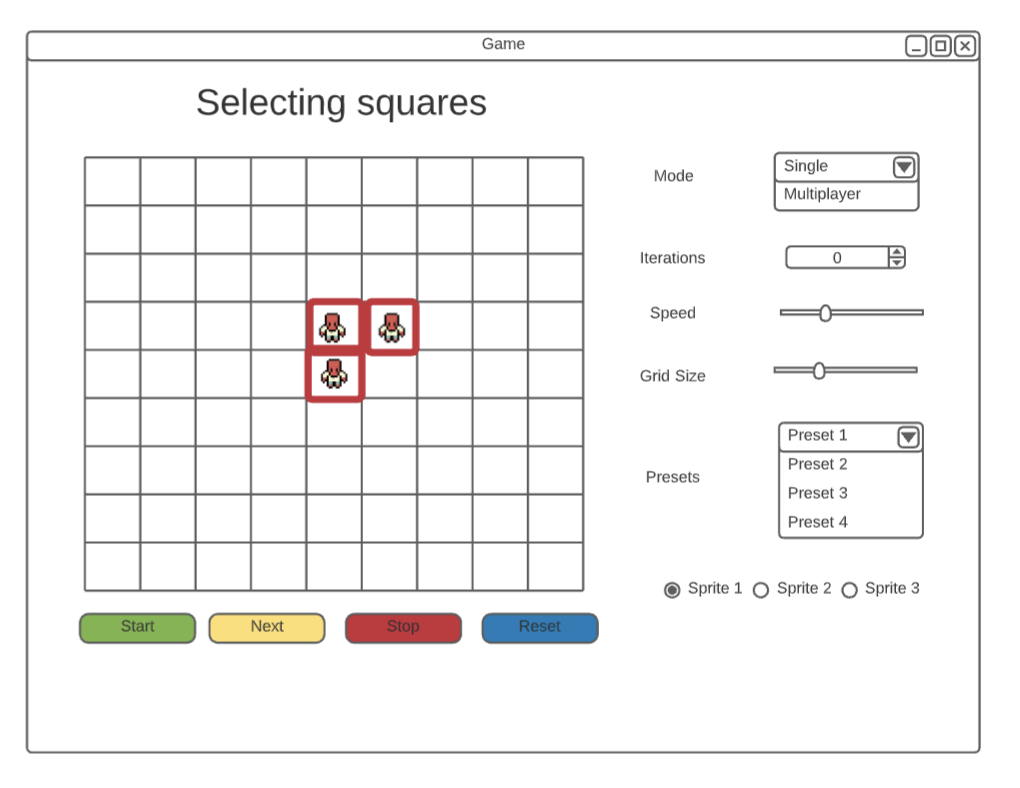
**Figure 3.1.c (Functionality)**

The system shall show a guide with information on functionality on the game screen.

****

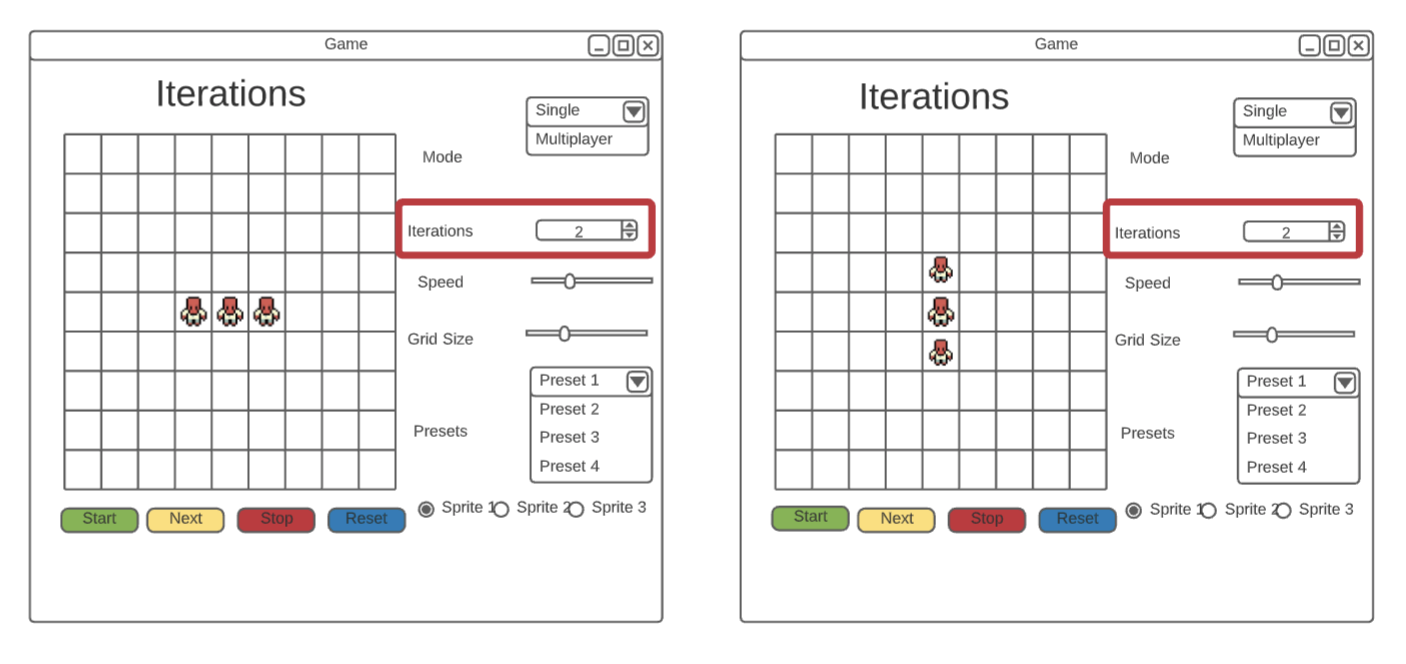
**Figure 3.2 (Game/Reset Screen)**

The system shall show the game interface.

****

**Figure 3.2.a (Main Controls)**

The system shall show when a player has activated or deactivated a cell on the grid. Start will start the simulation, Stop will stop the simulation at the current iteration, Next will display the next iteration, and Reset will clear the grid.

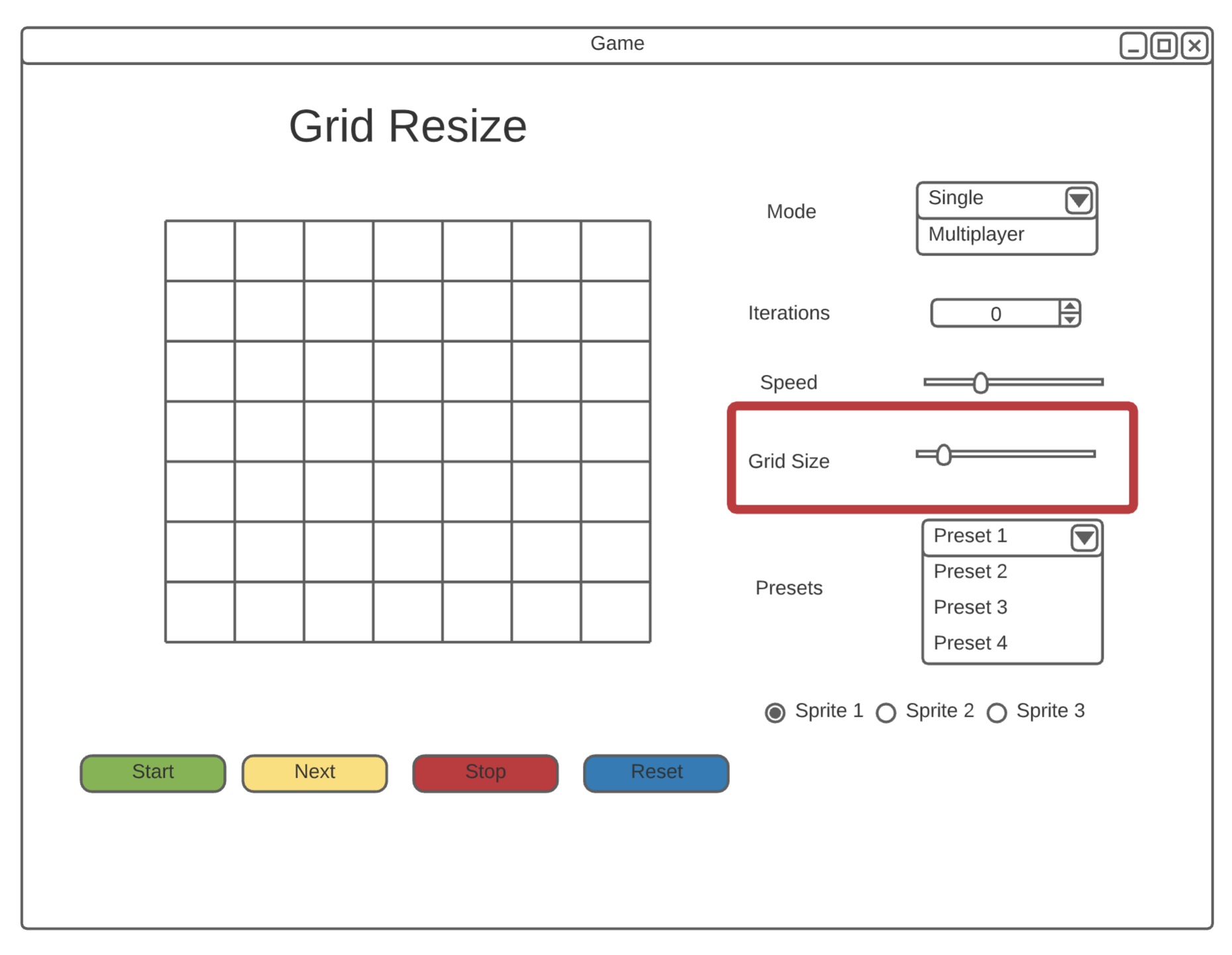


**Figure 3.2.b (Iterations)**

The system shall run the game up to iteration specified and stop.

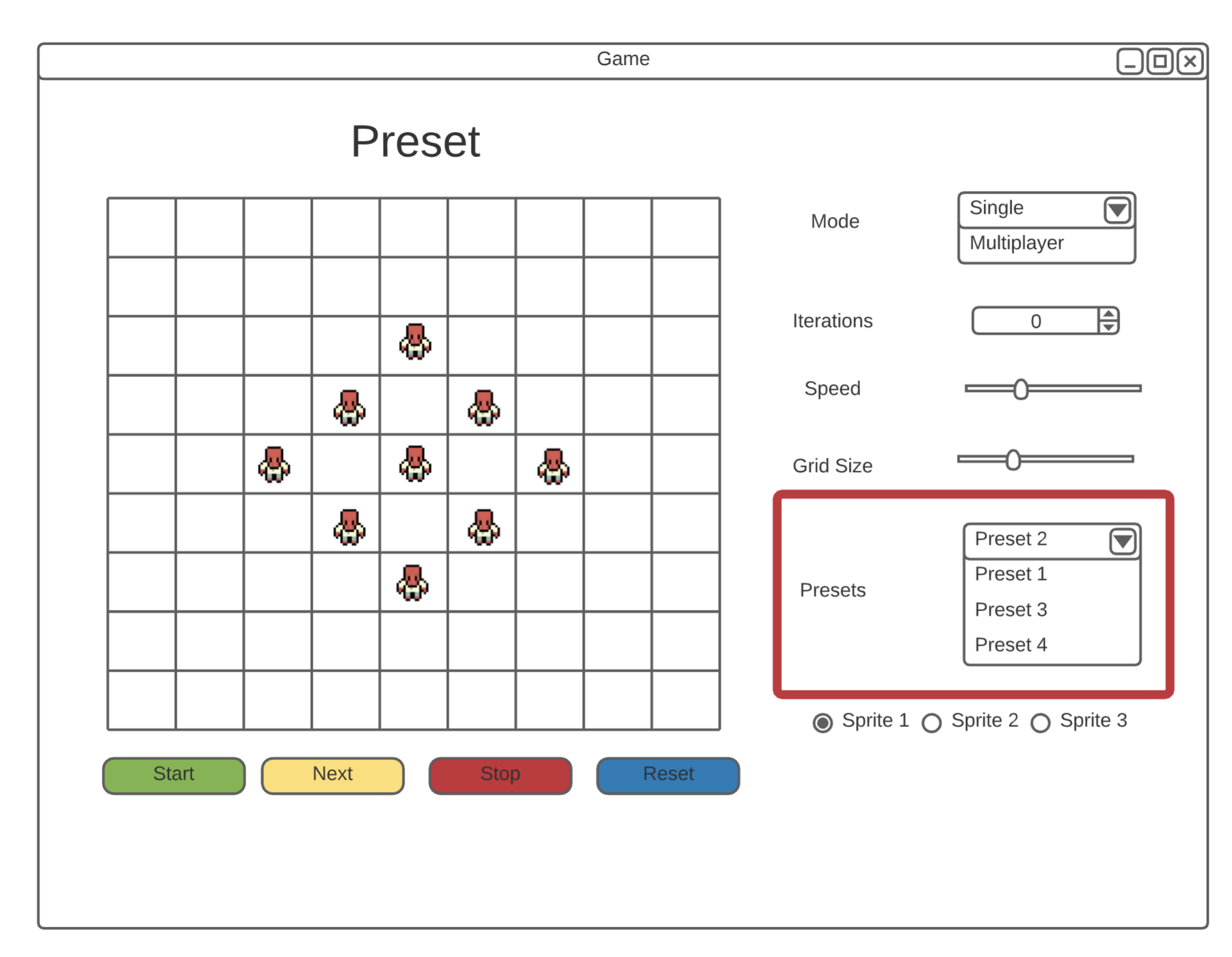
**Figure 3.2.c (Speed)**

The system shall adjust the speed of the game based on the speed slider.

****

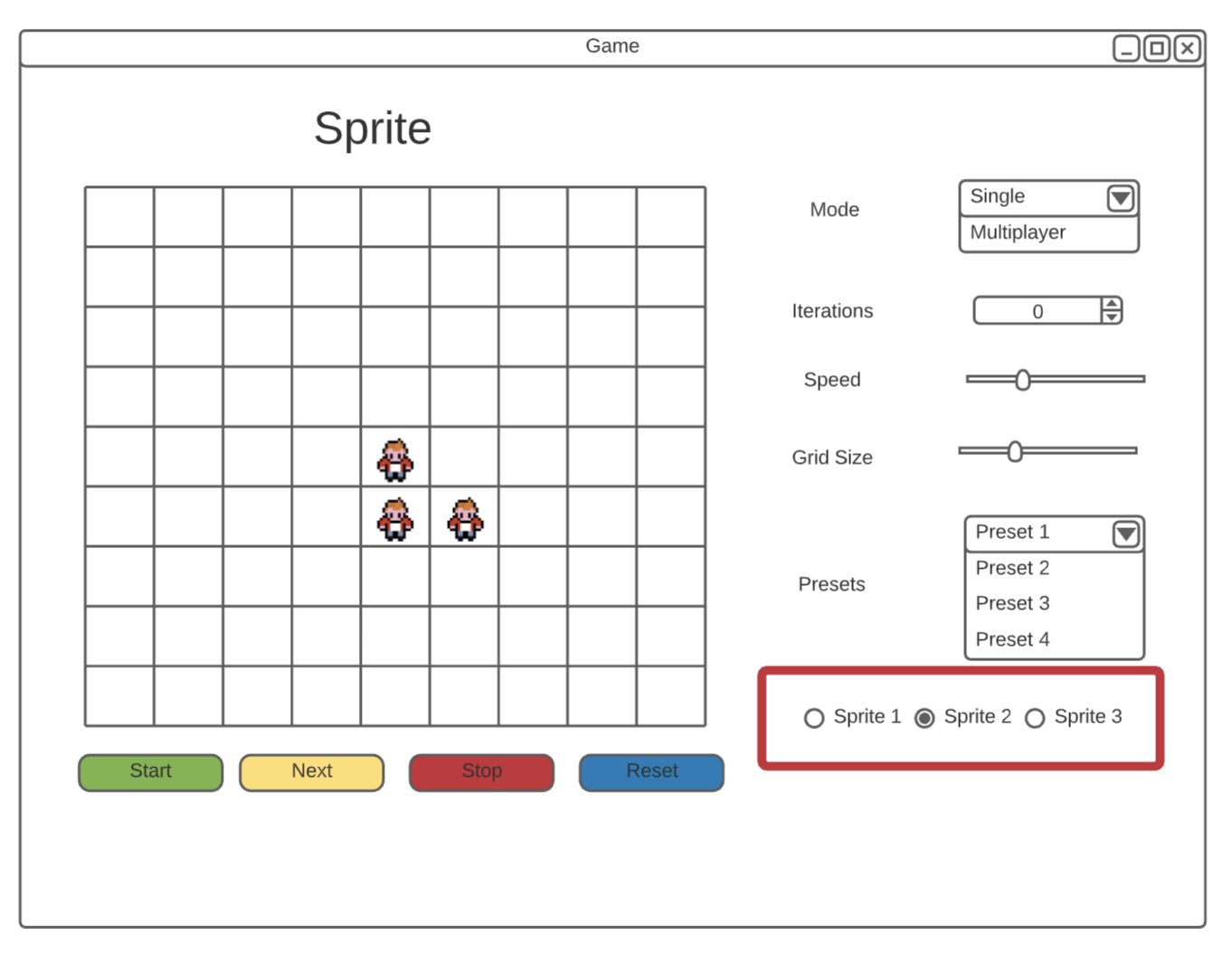
**Figure 3.2.d (Grid Resize)**

The system shall adjust the grid to be NxN, where N is the desired width selected by the number on the grid size slider (min: 5, default: 20, max: 50).

****

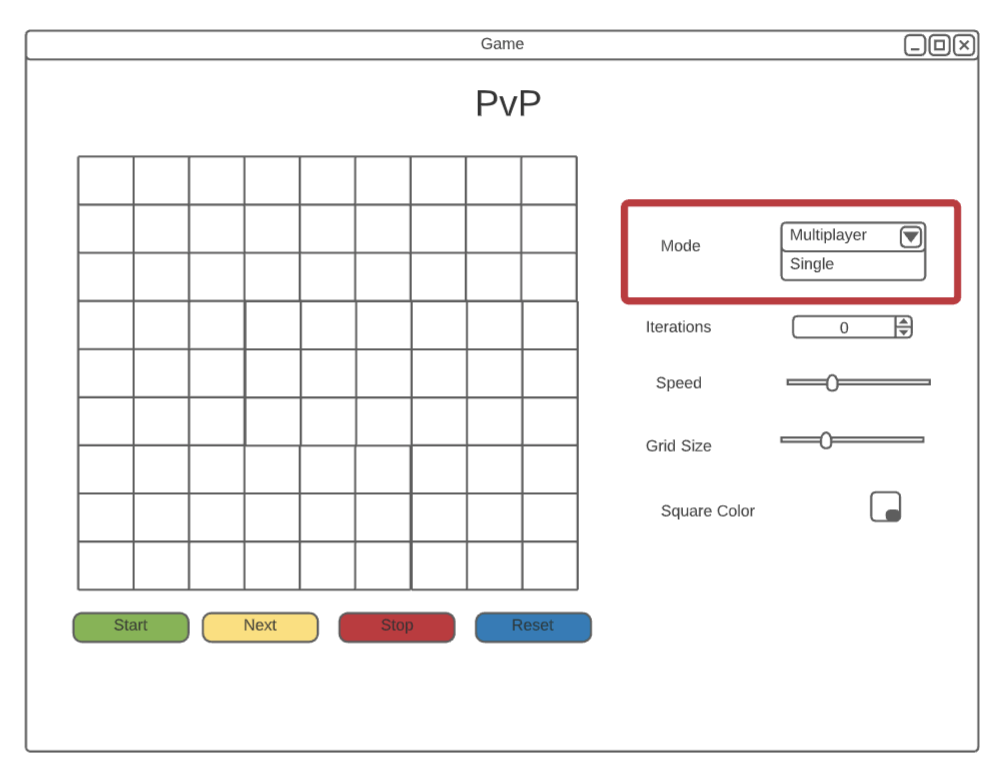
**Figure 3.2.e (Presets)**

The system shall display a preset pattern on the grid.



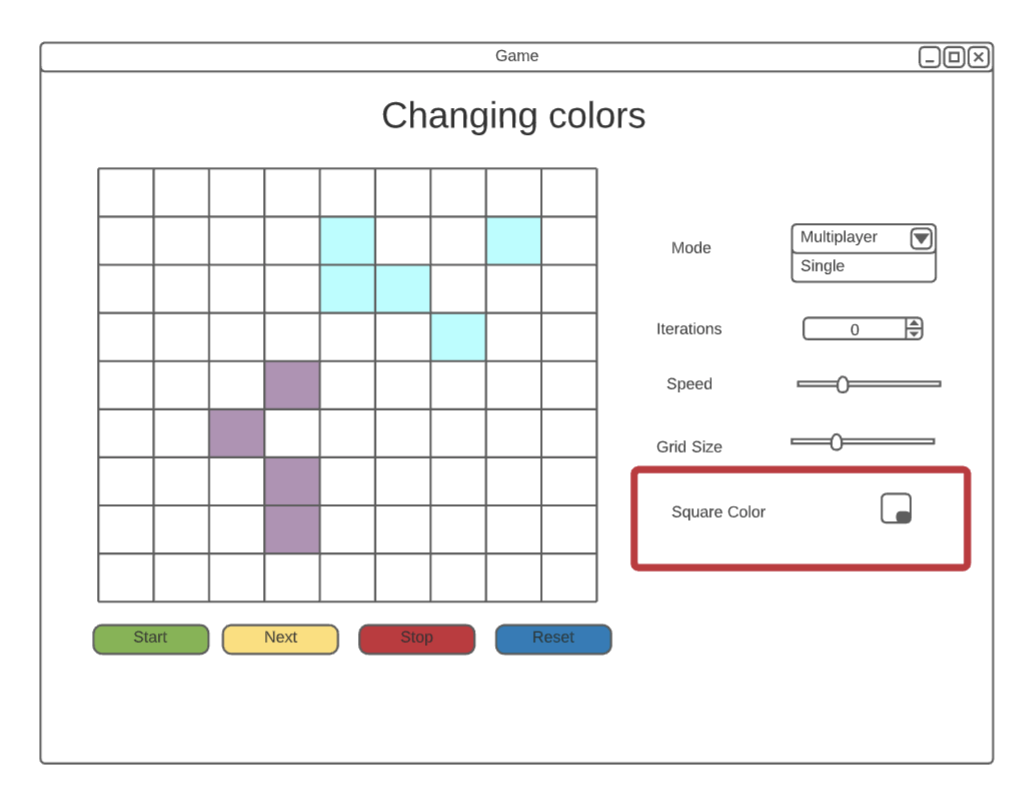
**Figure 3.2.f (Sprites)**

The system shall change the sprite on the grid to the selected sprite.

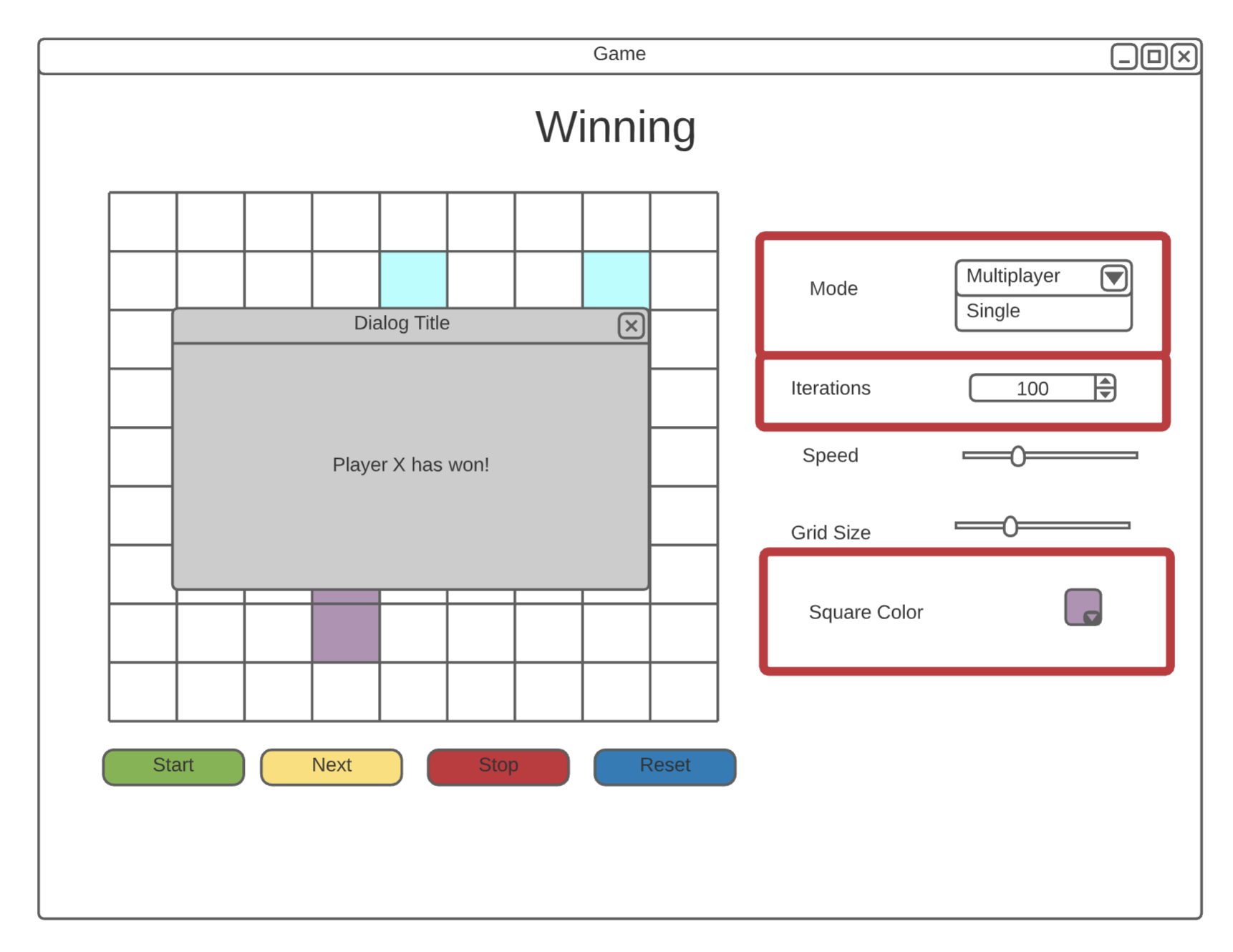


**Figure 3.2.g (PvP)**

The system shall provide a two-player mode. The system shall show the square color and hide the preset/sprite options.

**Figure 3.2.h (Colors)**

The system shall change the color of the cell based on the square color selected



**Figure 3.2.i (Winning)**

The system shall display an alert declaring the winner.

**4. System Features**

**4.1 Initializing the Application**

4.1.1 Functional Requirements

REQ 4.1.1 Upon initialization of application the user shall be presented with a main menu.(see figure 3.1.a)

If the player selects “Start” Game continue to 4.2

If the player selects “Exit” continue to 4.5

If the player selects “Rules” continue to 4.3

**4.2 Starting a Game**

4.2.1 Description

The game interface will present the player with a clickable grid, buttons to start/stop/step/reset/, and other various buttons that allow the user to interact with the grid

4.2.2 Functional Requirements

REQ 4.2.2a The player shall be able to select any given squares on the grid to activate/deactivate the cell.

REQ 4.2.2b The system shall present a button to start the Game

REQ 4.2.2c The system shall present a button to step through the different phases of life in the game

REQ 4.2.2d The system shall present a button to stop the game at the current phase of life.

REQ 4.2.2e The system shall present a button to reset the game to its initial iteration.

REQ 4.2.2f The system shall present an input box that allows the player to set how many generations the system will iterate through.

REQ 4.2.2g The system shall present a speed toggle that will adjust the time in between each generation.

REQ 4.2.2h The system shall present a grid toggle that allows the user to adjust the size of the grid. Default is set to 20, minimum will be 5, maximum will 50.

REQ 4.2.2i The player shall be presented with a drop down menu that allows the user to select from a list of preset patterns for the live cells of the grid box. Available only in solo mode.

REQ 4.2.2j The system shall present a radio button for preset sprites available only in solo mode.

REQ 4.2.2k The system shall present a dropdown to select game mode (solo or multiplayer), the iterations and color will be required. An alert will pop up at the end to display the winner (the player with most active cells based on color).

REQ 4.2.2l The system shall present a dropdown to select from a preset list of colors. Available only in multiplayer mode.

REQ 4.2.2m The system shall present the user with a dropdown box to choose between single or multiplayer. If multiplayer is selected, continue to 4.6.

**4.3 Rules**

4.3.1 Description

In the rules section the system shall present the player with the rules page which presents the user with text that explains how The Game of Life works.

4.3.2 Functional Requirements

REQ 4.3.2a The system shall present a button that returns the player back to the main menu.

REQ 4.3.2b The system shall present a button that allows the player to go next to the Explanation of Functionality (continue to 4.4)

REQ 4.3.2c The system shall present a button that allows the player to Start the Game. (continue from 4.2)

**4.4 Explanation of Functionality**

4.4.1 Description

In the Explanation of Functionality section, the user shall be presented with text that explains the features that the team implemented and how each feature affects the game.

4.4.2 Functional Requirements

REQ 4.4.2a The system shall present a back button that allows the player to go back to the Rules section. (continue from 4.3)

REQ 4.4.2b The system shall present a Start Game button that takes the player to the game interface. (continue from 4.2)

**4.5 Exiting the Game**

4.5.1. Description

The exit button, after prompting the player for confirmation that they do indeed want to exit, will exit the player from the game GUI.

4.5.2 Functional Requirements

REQ 4.5.2a The system shall prompt the player to confirm they want to exit the game. If yes the system will exit. If no, they will be returned to the Functional Requirements page. (continue from 4.4)

**4.6 Multiplayer mode**

4.6.1 Description

When multiplayer mode is selected(PVP), the players will have all the functional capabilities of the single player mode except for preset and sprite options. Additionally, the system shall give the option to change colors of cells to distinguish between players.

4.6.2 Functional Requirements

REQ 4.6.2a The system shall present the players with a button to choose their team color.

REQ 4.6.2b Upon the victory of a player, the system shall alert the player who has won with a popup.

**5. Other Nonfunctional Requirements**

* 1. **Performance Requirements**

REQ 5.1.1 The system shall be a console-based application

* 1. **Software Quality Attributes**

REQ 5.2.1 **Maintainability -** All fixes, updates and added features shall be done by the development team

REQ 5.2.2 **Adaptability -** The system shall be able to run on Mac and Windows operating systems

REQ 5.2.3 **Reliability -** The system shall act as a true game of life program, following the 4 rules for each iteration consistently

**6. Other Requirements**

**Appendix A: Glossary**

PvP Player vs. Player

GUI Graphical User Interface

Player Current system user

Cell Name given to a single square on the grid