CMSC 447

Software Design Description (SDD)

Name	Role	Signature
Daniel Sachs	Junior New Grad Associate Software Development Engineer I	
Innocent Kironji	Junior New Grad Associate Software Development Engineer I	
Sarah Alvarez	Junior New Grad Associate Software Development Engineer I	
Jason Schuler	Junior New Grad Associate Software Development Engineer I	
Pablo Burgos	Junior New Grad Associate Software Development Engineer I	
James Walls	Junior New Grad Associate Software Development Engineer I	

1 Scope

This section shall be divided into the following paragraphs.

1.1 Identification

“Game of Life: The Game”, henceforth referred to as “the game” is a software package developed and distributed by Group 2, also known as “As Long as We Graduate” and “Anything Goes” in Russell Cain’s CMSC 447 Spring ‘19 class. The game was developed to meet the specifications provided by Dr. Susan Mitchell, henceforth referred to as “the customer”.

1.2 System overview

The game is a desktop application. It is intended for use as a standalone tool on a single machine that allows two users to play against each other. The game runs a Game of Life simulation for each user with the object of “painting” a larger area than the opponent.

1.3 Document overview

The purpose of this document is to provide a detailed description of the game’s interface design.

2 Software Architecture

Figure 1 depicts the main code structure of the game. It will consist of three main components:

- Cell – The smallest unit that the game revolves around. It can be placed on the player grids and have the states: alive, dead, and painted.
- Main – The main driver of the game. It contains code for the GUI and game logistics such as game speed, starting/stopping the game, determining when the game is over, etc.
- Tkinter – The main GUI library that the game depends on.

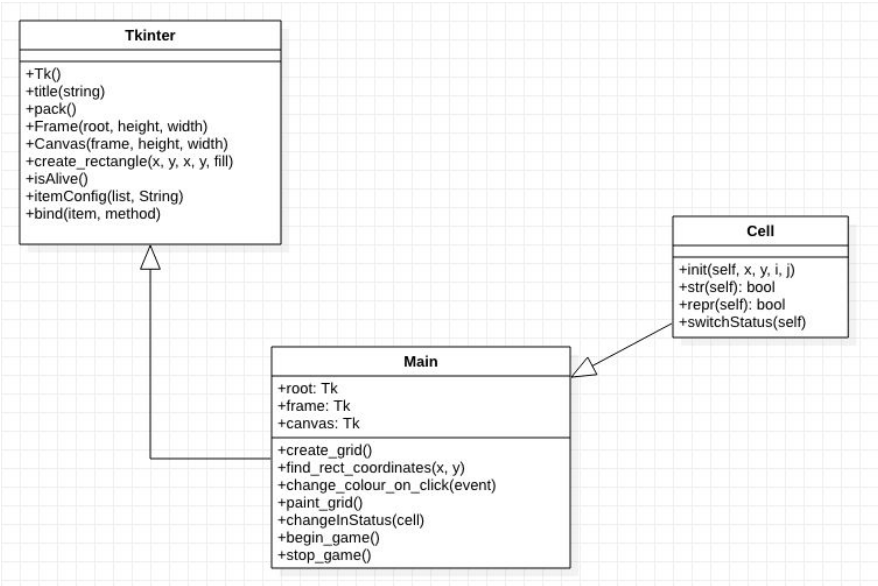

Figure 1. UML diagram.

3 Concept of execution

This section will illustrate the general flow of the game's execution.

Figure 1. Upon execution of the game, this is the first window that appears. Its purpose is to obtain the name that player 1 desires. The text box takes in a string that player 1 types. Once the ENTER button is clicked, the window proceeds to Figure 2.

Figure 2. The purpose of this window is to obtain the number of the color that player 1 desires. If they input 1 into the textbox, their color will be set to Color1. Once the ENTER button is clicked, the window proceeds to two more windows similar to Figure1 and Figure 2. The only difference is that it will now address player 2.

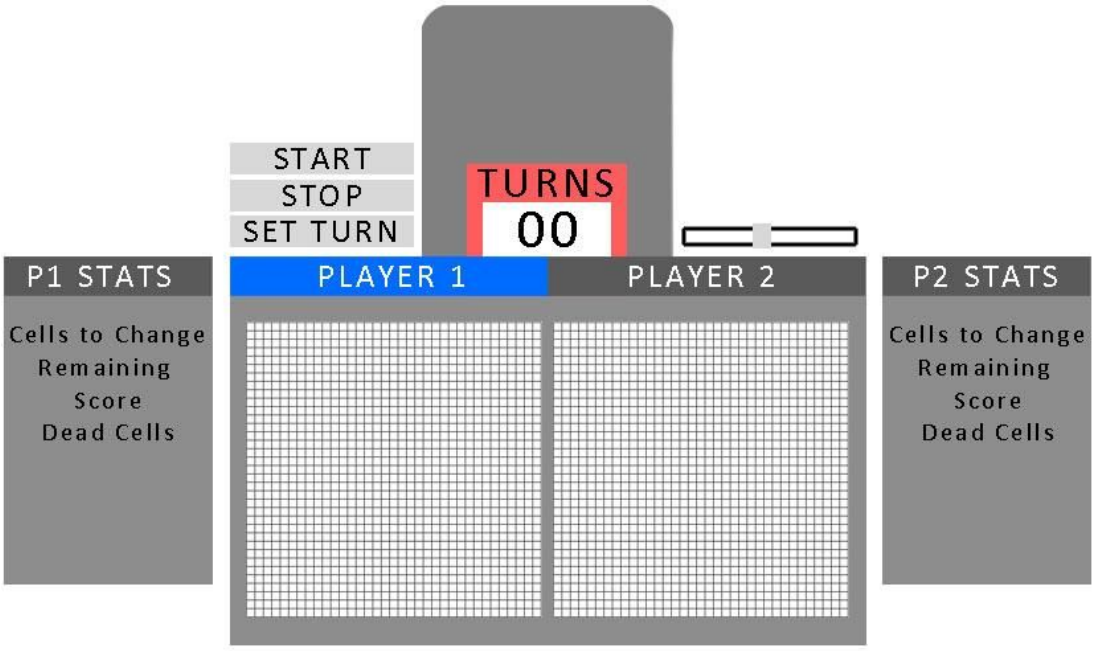

Figure 3. Once all the initial input windows from Figure 1 and Figure 2 have succeeded, they will terminate and the main game window will appear as shown. Player 1 will always go first so their banner will be highlighted in their chosen color. The game boards will be blank and the turns and game speed slider will be set to their respective default values.

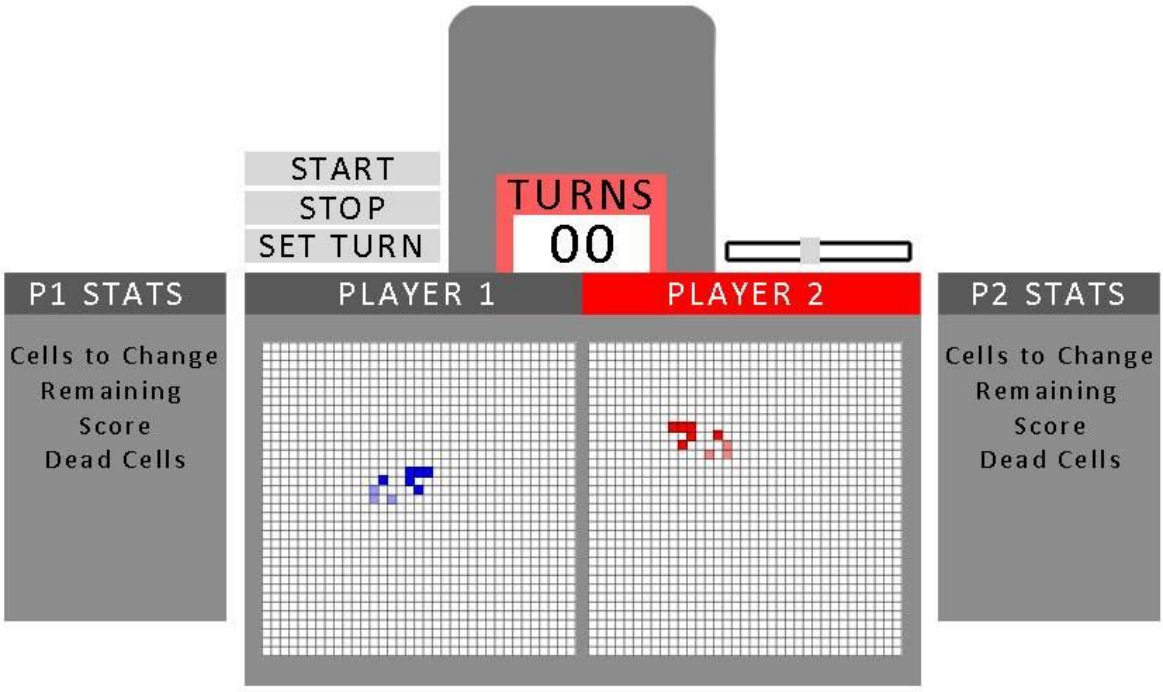

Figure 4. Once player 1 has exhausted all of their "Cells to Change", it will be player 2's turn, hence the red banner highlighting player 2's name. This image also illustrates the alive cells (the darker colored cells) and the dead cells (the lighter colored cells). When the number of turns reaches 0, the winner screen will appear.

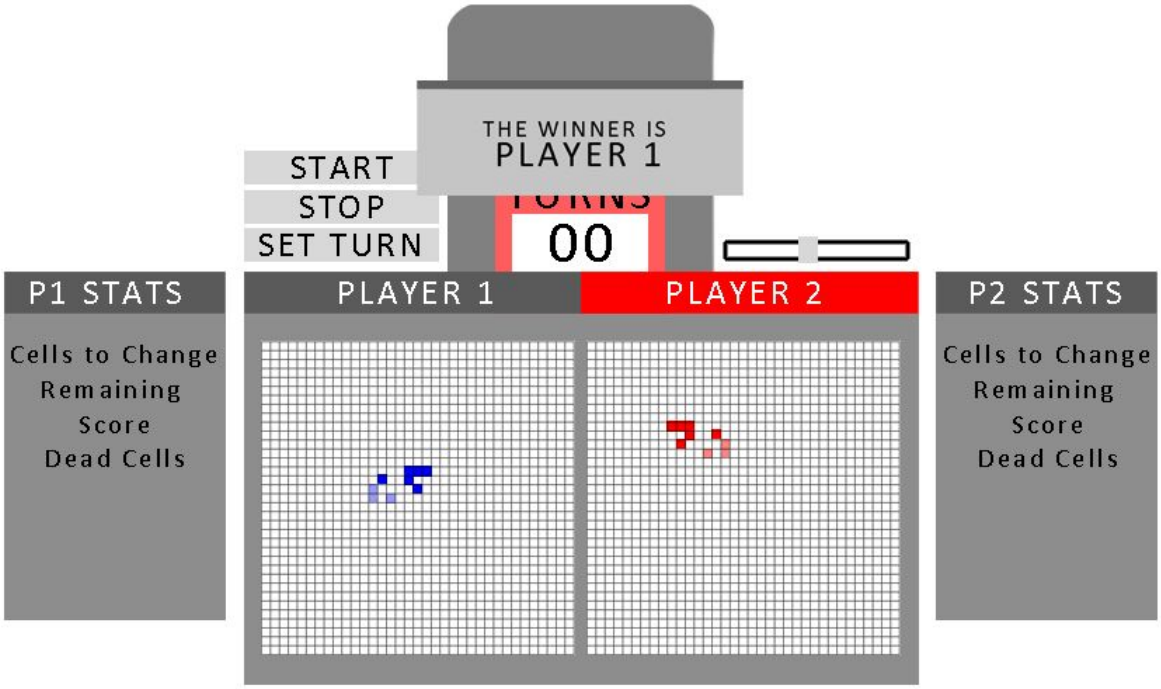

Figure 5. The winning screen will appear above the main game screen once the turns reaches 0. The image shown is for the case where player 1 wins the game. Should player 2 win the game, their name will be displayed instead of player 1.

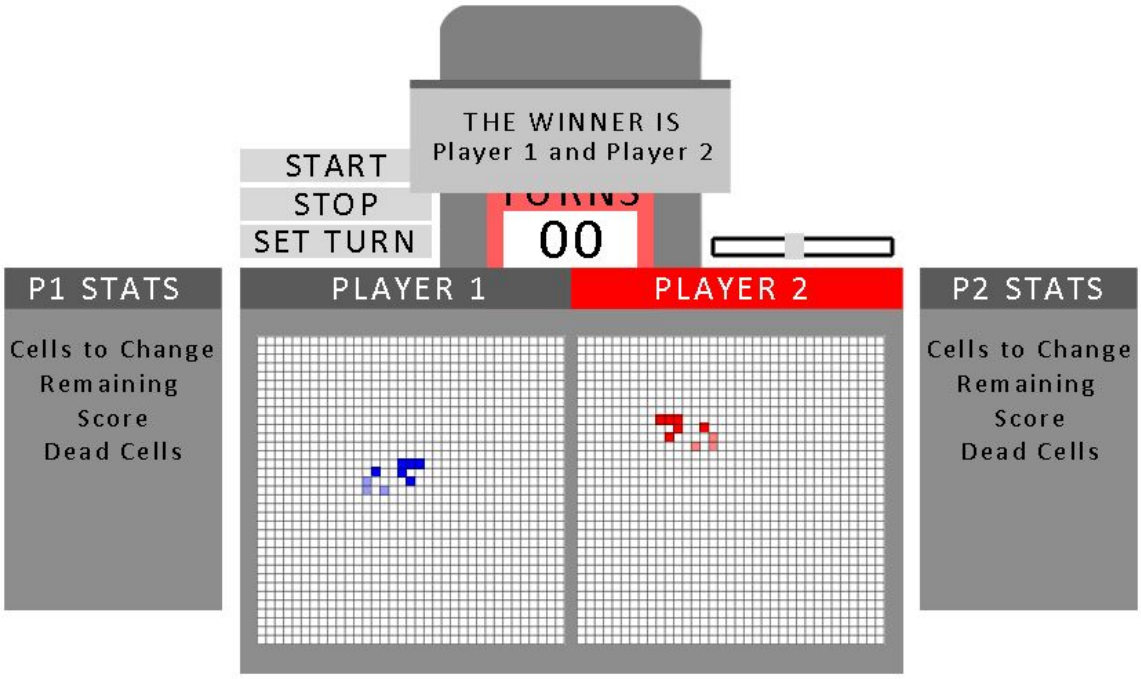

Figure 6. Should the game end in a tie, this version of the winning screen will appear instead of the one seen in Figure 5.

3.1 Interface design

This paragraph shall be divided into the following subparagraphs to describe the interface characteristics of the software units. It shall include both interfaces among the software units and their interfaces with external entities such as systems, configuration items, and users. If part or all of this information is contained in Interface Design Descriptions (IDDs), in section 5 of the SDD, or elsewhere, these sources may be referenced.

3.1.1 Interface identification and diagrams

This paragraph shall state the project-unique identifier assigned to each interface and shall identify the interfacing entities (software units, systems, configuration items, users, etc.) by name, number, version, and documentation references, as applicable. The identification shall state which entities have fixed interface characteristics (and therefore impose interface requirements on interfacing entities) and which are being developed or modified (thus having interface requirements imposed on them). One or more interface diagrams shall be provided, as appropriate, to depict the interfaces.

3.1.2 Player 1 Name Input Interface

The interfacing entity of this interface is the user, specifically player 1. This interface includes a text box which will take any keyboard input from the user. However, it will only accept a string that is 10 characters or less in length. Input will only be processed when the Enter key on the keyboard is

pressed. Invalid input will cause player 1 to be assigned the default name of “Player 1”. The interface will also include a button with the label “Enter”. Once this button is pressed, the interface will terminate and the Player 1 Color Input Interface will appear.

3.1.3 Player 1 Color Input Interface

The interfacing entity of this interface is the user, specifically player 1. This interface includes a text box which will take any keyboard input from the user. However, it will only accept an integer between the range of [1, 4]. Input will only be processed when the Enter key on the keyboard is pressed. Invalid input will cause player 1 to be assigned the default color of blue. The interface will also include a button with the label “Enter”. Once this button is pressed, the interface will terminate and the Player 2 Name Input Interface will appear.

3.1.4 Player 2 Name Input Interface

The interfacing entity of this interface is the user, specifically player 2. This interface includes a text box which will take any keyboard input from the user. However, it will only accept a string that is 10 characters or less in length. Input will only be processed when the Enter key on the keyboard is pressed. Invalid input will cause player 2 to be assigned the default name of “Player 2”. The interface will also include a button with the label “Enter”. Once this button is pressed, the interface will terminate and the Player 2 Color Input Interface will appear.

3.1.5 Player 2 Color Input Interface

The interfacing entity of this interface is the user, specifically player 2. This interface includes a text box which will take any keyboard input from the user. However, it will only accept an integer between the range of [1, 4]. Input will only be processed when the Enter key on the keyboard is pressed. Invalid input will cause player 2 to be assigned the default color of blue. The interface will also include a button with the label “Enter”. Once this button is pressed, the interface will terminate and the main game screen will appear.