CMSC 447

Software User Manual (SUM)

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Signature** |
| Caroline Cocca | POC/Team Lead |  |
| James Gough | Simulation Dev |  |
| Stuart Reilly | Junior Developer |  |
| Kelley Schmidt | Documentation and  Diagrams Technician |  |
| Ryan Messett | Data Guru |  |
| Sang Nguyen | Visual Specialist |  |
| Dr. Jon Squire | Customer |  |

[1 Scope 3](#_3znysh7)

[1.1 Identification 3](#_1t3h5sf)

[1.2 System overview 3](#_4d34og8)

[1.3 Document overview 3](#_2s8eyo1)

[2 Referenced documents 3](#_3rdcrjn)

[3 Software summary](#_lnxbz9) 4

[3.1 Software application](#_44sinio) 4

[3.2 Software inventory](#_2jxsxqh) 4

[3.3 Software environment](#_3j2qqm3) 4

[3.4 Software organization and overview of operation 4](#_4i7ojhp)

[3.5 Security and privacy](#_2bn6wsx) 5

[3.6 Assistance and problem reporting](#_3as4poj) 5

[4 Access to the software](#_49x2ik5) 6

[4.1 First-time user of the software](#_147n2zr) 6

[4.1.1 Equipment familiarization](#_23ckvvd) 6

[4.1.2 Installation and setup](#_41mghml) 6

[4.2 Initiating a session](#_vx1227) 6

[4.3 Stopping and suspending work](#_1v1yuxt) 6

[5 Processing reference guide 6](#_2u6wntf)

[5.1 Capabilities 6](#_3tbugp1)

[5.2 Conventions](#_nmf14n) 7

[5.3 Processing procedures](#_1mrcu09) 7

[5.3.1](#_2lwamvv) Run 7

5.3.2 Stop 7

5.3.3 Read/Write File, Show 7

[5.4 Related processing 7](#_3l18frh)

[5.5 Recovery from errors, malfunctions, and emergencies](#_1egqt2p) 8

[5.6 Messages](#_2dlolyb) 8

# Scope

## Identification

This document applied to an instruction manual for our Python implementation of Conway’s Game of Life. The application is made to run on Windows 10 via command prompt. The Game of Life is a cellular automaton that is a single-player game with the only user input needed being the initial setup of the starting cell(s) and menu item selections. The cells are contained in squares of a grid, with each square of the grid being a specific x-y coordinate. The cells multiply, shrink, and move to form patterns according to specified algorithms based on their placement and the placement of surrounding cells. The goal of the game is to create interesting patterns through various starting cell coordinates. Title is “Game of Life”, version number one, release number one.

## System overview

The purpose of this application is to demonstrate a classic cellular automaton with client-specific requirements. The input of the initial state of the game is the only input required in order to play the game. The software is a visual program and various aspects can be configured by the user. The system on which the software will run is the Windows 10 operating system. A mouse is required for the system to be able to navigate the software. The users of this software are the developers, graders, and the client. There are no project sponsors as this project will have no monetary cost to produce. The acquirer for our software application is the client, Dr. Jon Squire. The developers consist of our software engineering team - Kelley Schmidt, Caroline Cocca, James Gough, Sang Nguyen, Ryan Messett, and Stuart Reilly. The support agencies for our project would be the teaching assistants, Professor Cain, and the client if support is needed. Planned operating sites not applicable. Other relevant documents are the Software Design Description, Software Requirements Specification, Software Test Description, and Software Test Report.

## Document overview

This document serves as an instruction manual for an intended user of our implementation of Conway’s Game of Life. The document summarizes the software’s capabilities, environment, and organization. An overview of how to install and run the software is provided, along with equipment familiarization. Finally, a reference guide is provided on procedures for using the software, conventions employed by the software, and quick references for menu options in the game. There are no privacy or security concerns regarding the use of this document.

# Referenced documents

Conway's Game of Life. (2019, March 24). Retrieved from <https://en.wikipedia.org/wiki/Conway's_Game_of_Life>

Tkinter -- Python interface to Tcl/Tk. (2019). Retrieved from <https://docs.python.org/2/library/tkinter.html>

Pandas: Python Data Analysis Library. (2019). Retrieved from <https://pandas.pydata.org>

Team Scrum-Diddlyumptious Github Repository: <https://github.com/ryanmessett/scrum-diddlyumptious>

# Software summary

## Software application

* **Description**: The Game of Life application is intended as a single-player game where the user can interact through clicks to simulate a cellular automaton.
* **Capabilities**: The Game of Life application can save and load grid states to and from files. Gameplay can be executed with a single step, run for continuous steps, or paused. The rate of gameplay can be sped up or slowed down. Grid lines can be hidden or shown, and the selected color may be changed to red, green, or blue for grid pixel placement. Step and life counts can be viewed as the game executes, along with the word “Stable” if the user has managed to produce a stable oscillating grid.
* **Benefits**: The intended benefit of the software is for the user to enjoy gameplay and feel pride in discovering interesting patterns in the Game of Life.

## Software inventory

In order for the software to operate, the following must be installed:

* **game\_of\_life.exe**: The python file containing the implementation of The Game of Life. There are no security or privacy considerations for this file.

## Software environment

* **Computer equipment**: Hardware that must be present includes a mouse or trackpad for clicking on menu items and interacting with the grid. The memory available should be at least 1GB of RAM, and sufficient memory for storing the game file (< 40 MB) and any saved grid files (< 200 KB).
* **Communications equipment**: None
* **Other software**: Python 3 programming language support must be present. The Windows 10 operating system should be used to run the program.

## Software organization and overview of operation

* **Logical Components**: The program is a GUI (Graphical User Interface) consisting of count/status displays, menus, and the grid.
  + **Count/Status displays**: A life count and step count are displayed, which report the squares active in the grid and the current number of steps executed, respectively. The word “Stable” is displayed when the grid status is currently stable; i.e., the entire grid contains only patterns which oscillate between 1 or 2 shapes and do not move location in the grid.
  + **Menus**: The GUI contains File, Show, Actions, and Grid/Speed menus.
    - **File**: Contains the options Write File and Read File. Write File allows the user to save the grid state to a file. Write File allows the user to load a grid state from a file.
    - **Show**: Contains pre-made grid states that, upon selection, load from a file to the program.
    - **Actions**: Contains the options Clear, Run, Step, Stop, and Quit. Clear removes all active squares from the grid. Run executes steps continuously, updating the life and step counts with each completed step. Step executes a single step. Stop pauses the game if selected while running. Quit terminates the program.
    - **Grid/Speed**: Contains the options No Grid, Show Grid, Faster, Slower, and Color. No Grid hides grid lines if shown. Show Grid displays grid lines if hidden. Faster increases the rate of step execution while running. Slower decreases the rate of step execution while running. Color is a submenu which allows the user to pick from Red, Blue, and Green for grid pixel coloring.
  + **Grid**: The area of the GUI where squares can be clicked by the user. These squares appear, disappear, and move in the grid according to the Game of Life rules when a step is executed.
* **Performance Characteristics**: The following can be expected from the user.
  + **Response Time**: All menu option selections and and grid square selections shall have a response time of less than a second for the associated action. The life and step counts shall update by step completion. The program shall display Stable within completion of the final step in the first stable oscillation of the entire grid.
  + **Limitations**: Stable patterns will be detected up to oscillations between two shapes. Patterns that oscillate between 3 or more shapes will not be detected as stable.
  + **Error rate**: Response times may deviate from the above specifications if there is insufficient RAM to run the program. File save of grid states may fail if there is insufficient memory to store the file.
* **Relationships**: The program interfaces with the Windows 10 operating system in order to save and load files from the system.

## Security and privacy

There are no security or privacy concerns associated with this software.

## Assistance and problem reporting

To obtain assistance or report problems with the Game of Life implementation, contact the Scrum-Diddlyumptious Team with details of the assistance needed or the bug encountered. The point of contact for the Scrum-Diddlyumptious team is Caroline Cocca (ccocca1@umbc.edu).

# Access to the software

## First-time user of the software

### Equipment familiarization

* **Visual Display Dimensions**: The default dimensions of the Graphical User Interface is 500x500 pixels.
* **Cursor appearance**: The cursor appears according to the cursor type selected in the Windows 10 system preferences. The cursor is the only method of interaction with the game, and may be used to click on menu options and interact with the grid.

### Installation and setup

To install the software, the zipped game\_of\_life.exe file must be downloaded from the link at the Team Scrum-Diddlyumptious GitHub repository, linked above in Section 2: Referenced Documents. From the home page of the repository, scroll down to view the README. There is a section in the README titled “Download” with a link to the zipped exe file. After downloading and unpacking this, simply run the exe. No further installation is required.

## Initiating a session

To initiate a session of the Game of Life application, the game\_of\_life.exe file may be double clicked to run. This will launch the Graphical User Interface.

## Stopping and suspending work

To terminate a session of the Game of Life application, the Quit option may be selected from the Actions menu. The Close button, marked by an “X” in the upper right corner of the window, may also be clicked to end the application. To pause a session of the Game of Life application that is actively running steps, the Stop option may be selected from the Actions menu. To save a session of the Game of Life application for future continuation, the Write File option may be selected from the File menu.

# Processing reference guide

## Capabilities

* **File Menu**: This menu may be used to read grid states from files and write grid states to files. These functions interface with the Windows 10 operating system, as well as the grid dataframe.
* **Show Menu**: This menu may be used to load premade patterns to the grid. These functions interface with the grid dataframe.
* **Actions menu**: This menu may be used to Clear, Run, Step, Stop, or Quit. Clear will remove all active pixels from the grid. Run will execute multiple steps in succession. Step will execute one single step. Stop will halt step execution if the program is in Run mode. Quit will terminate the program. These functions interface with the grid dataframe, as well as the stored life count, step count, and stability.
* **Grid/Speed Menu**: This menu may be used to show or hide grid lines in the Grid area, as well as to speed up or slow down the rate of step execution while the program is in Run mode. This menu also contains the Color submenu that allows the user to select from Red, Green, and Blue for the color of square selection in the Grid area. These functions interface with the grid dataframe, as well as the stored color selected and current speed.
* **Grid**: The area of the Graphical User Interface that displays active squares in the Game of Life simulation. Squares appear, disappear, and move according to the rules of Conway’s Game of Life. This interfaces with the stored color selected as well as the grid dataframe.

## Conventions

* **Color**: Red is the default “red” option in TkInter, with RGB value (255, 0, 0). Blue is the default “blue” option in TkInter, with RGB values (0, 0, 255). Green is the default “green” option in TkInter, with RGB values (0, 255, 0). The default color is the default “black” option in TkInter, with RGB value (0, 0, 0).
* **Stability**: A stable grid is defined as a grid that contains patterns which only oscillate between 1 or 2 shapes and do not change position in the Grid. A pattern which oscillates between 3 or more shapes is not detected as stable.

## Processing procedures

The following sections describe steps that have an inherent order to them in order for selected actions to be properly executed.

### Run

In order for the subsequent action to be taken, the program must be in Run mode, where steps are being executed continuously according to the current speed selected. This can be initiated by selecting the Run option within the Actions menu.

* + 1. **Stop**

In order to pause the program, the program must already be in Run mode. The program can be paused by selecting the Stop option within the Actions menu. This takes the program out of Run mode. If Stop is selected while the program is not in Run mode, nothing will happen.

* + 1. **Read/Write File, Show**

In order to successfully read and write files of the grid state or use the Show menu to load premade grid states, the program should not be in Run mode. This should be achieved by using the Stop option within the Actions menu to pause the game if it is currently in Run mode. The Read File and Write File options can be selected from the File menu in order to save and load grid states through the Windows 10 operating system. Pattern options can be selected from the Show menu in order to load premade grid states. The current session does not necessarily have to previously have been in Run mode to use the actions in this paragraph.

## Related processing

The program performs processing in the background through using the rules of Conway’s Game of Life to compute pixel placement in the Grid at the end of an executed step. The user has no responsibilities in supporting this background processing.

## Recovery from errors, malfunctions, and emergencies

In the event of an error in visual display or Grid state, such as the Grid becoming unresponsive to clicks, the user may choose to restart the program through terminating the session and initiating a new one as described in sections 4.3 and 4.2 above. If necessary to revert to a previous grid state, a saved state can be loaded as described in sections 3.4 and 5.1.

## Messages

The message “Invalid save file!” is displayed to the user in the event that the user selects Read File and attempts to load a file with an improper format. If this message is encountered, the user should ensure their saved file is in the expected format; i.e., that which is output by the Write File option.