

# **User Manual for the Game of Life Simulation**

# Table Of Contents

Chapter.....	pg#
1.1 Introduction.....	3
2.1 Usage Instructions.....	5
2.1.1 Prerequisites:.....	5
2.2.1 Running the Program:.....	5
2.2.2 Example:.....	5
2.3 Helper Scripts.....	6
2.3.1 GameOfLife.bat:.....	6
2.3.2 GameOfLife.sh:.....	6
3.1 Error Messages.....	7
4.1 Input and Output Formats.....	8
4.1.1 Seed File:.....	8
4.1.2 Example Seed File:.....	8
4.1.3 Output Files:.....	9
4.1.4 Example Output File:.....	9

## 1.1 Introduction

The Game of Life is a cellular automaton devised by mathematician John Conway. This “Game” consists of a two-dimensional orthogonal grid of square cells, each of which is in one of two possible states, alive or dead, or “populated” or “unpopulated”. Every cell interacts with its eight neighbors, which are the cells that are horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:

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

This Java program simulates Conway's Game of Life. It allows users to load an initial configuration (seed) from a file, simulate the evolution of the grid over a specified number of steps, and save the state of the grid after each step.

This manual will provide a comprehensive guide to using the Game of Life simulation program. It includes descriptions of features, input and output formats, error messages, and examples.

## 1.2 Features

- **Load Seed File:** Initialize the grid based on a seed file.
- **Simulate Evolution:** Simulate the grid's evolution for a specified number of steps.
- **Save Grid State:** Save the state of the grid to a file after each simulation step.

## 2.1 Usage Instructions

### 2.1.1 Prerequisites:

- Java Development Kit (JDK) installed on your system.
- A text editor to create and modify seed files.

### 2.2.1 Running the Program:

1. Open a terminal or command prompt and make sure you are in the correct directory.

Compile the Java program using the following command:

```
javac GameOfLife.java
```

2. Run the program with the following command:

```
java GameOfLife <seed_file> <output_name> <steps>
```

3. Replace **<seed\_file>** with the path to your seed file, **<output\_name>** with the desired output file name, and **<steps>** with the number of steps to simulate.

### 2.2.2 Example:

Given a seed file named **seed.txt** with output name **out.txt** and you want to simulate 10 steps:

```
java GameOfLife seed.txt out.txt 10
```

## 2.3 Helper Scripts

### 2.3.1 GameOfLife.bat:

1. Open a terminal or command prompt and make sure you are in the scripts directory.
2. Run the script with the following command:

```
GameOfLife.bat
```

Available options will now be displayed:

```
How to use GameOfLife.bat:
GameOfLife.bat javadoc
    - this compiles all of the Javadocs in the program into ./docs/
GameOfLife.bat build
    - this compiles the java code into bytecode into
"C:\Users\nicow\OneDrive\Documents\JAVAR\wangn4\src\main\java"
GameOfLife.bat run ...
    - this runs the compiled butecode
GameOfLife.bat all
    - runs all of the previous options in order
GameOfLife.bat help
    - displays this help message
```

When using all or run options, remember to use the correct arguments as such:

```
GameOfLife.bat run <seed_file> <output_name> <steps>
```

Or

```
GameOfLife.bat all <seed_file> <output_name> <steps>
```

## 3.1 Error Messages

The program may output the following error messages:

```
Usage: java GameOfLife <seed_file> <output_name> <steps>
```

This error occurs if the program is run without the required command-line arguments.

```
java.lang.NumberFormatException
```

This error occurs if the seed file contains non-numeric values where numbers are expected.

```
java.io.FileNotFoundException
```

This error occurs if the specified seed file cannot be found.

```
java.io.IOException
```

This error occurs if there is an issue reading from the seed file or writing to the output file.

## 4.1 Input and Output Formats

### 4.1.1 Seed File:

The seed file defines the initial state of the grid. It should be a plain text file with the following format:

1. **First line:** Contains two integers separated by a comma and a space, representing the number of rows and columns in the grid.
2. **Subsequent lines:** Each line represents a row in the grid, with cell values (0 for dead, 1 for alive) separated by a comma and a space.

### 4.1.2 Example Seed File:

```
5, 7
0, 0, 0, 0, 0, 0, 0
0, 0, 0, 1, 0, 0, 0
0, 0, 0, 1, 1, 1, 1
1, 0, 0, 0, 1, 1, 1
1, 1, 1, 1, 1, 1, 1
```



### 4.1.3 Output Files:

The program saves the state of the grid after each step to a file. The output file name is based on the specified name and the tick number. Each output file has the following format:

1. **First line:** Contains two integers separated by a comma and a space, representing the number of rows and columns in the grid.
2. **Subsequent lines:** Each line represents a row in the grid, with cell values (0 for dead, 1 for alive) separated by a comma and a space.

### 4.1.4 Example Output File:

```
5, 7
1, 1, 0, 0, 0, 1, 1,
0, 0, 0, 1, 0, 1, 0,
1, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0,
0, 1, 1, 1, 0, 0, 0,
```