

Unit Five Assignment Four:

The *Game of Life* is a well-known mathematical game that gives rise to amazingly complex behavior, although it can be specified by a few simple rules. (It is not actually a game in the traditional sense, with players competing for a win.) Here are the rules. The game is played on a rectangular board. Each square can be either empty or occupied. At the beginning, you can specify empty and occupied cells in some way; then the game runs automatically. In each generation, the next generation is computed. A new cell is born on an empty square if it is surrounded by exactly three occupied neighbor cells. A cell dies of overcrowding if it is surrounded by four or more neighbors, and it dies of loneliness if it is surrounded by zero or one neighbors. A neighbor is an occupant of an adjacent square to the left, right, top, or bottom, or in a diagonal direction.

Write a Java project (application) that will program this game. To eliminate the drudgery of computing successive generations by hand, use a 6x6 two dimensional array to store the rectangular configuration. Your program must print, to a JTextArea in a Container attached to a JFrame, Generations 0, 2, & 4 of this game.

Use the code included in the Unit 5 Lecture 4 notes to get started. I would suggest you implement the following two methods in your project...

```
public void ChangeMatrix() -
```

This method uses a temporary 2D array to create the next generation and then allows the reference to the original array to take on the temporary array.

```
public int NumNeighbors(int row, int col) -
```

This method counts the number of occupied neighbors the specified location has.

```
{Be careful, don't create a subscript out of bounds error!}
```

Paste your output into a Word document and turn it in with your code. This Word document should look like the one attached to this assignment.