Code Assignment 5
Classes and Objects

CS152 – Computer Programming Fundamentals

Instructor: M. Wolverton

Items Due

- **Completed Code 5**

Instructions

Complete the coding directive in an Intellij IDEA project with the assignment name (e.g. Code 1-2) and package name cs152. Compress each completed project separately into a .zip file and upload it to your account on the file server at cec-code-lab.aps.edu in a folder named CS152. Code will be downloaded and archived for grading on the assignment due date.

Directive

Create a rendition of Conway's Game of Life using Java FX and the following code structure guidelines. Links to the descriptions of the Conway's Game of Life as well a working demonstration executable is linked on on cec-code-lab.aps.edu where this assignment was posted. Briefly, Conway's Game of Life is a 2D grid based simulation to show how complex behavior can emerge from simple rules.

Your program should contain a grid that clearly shows cells alive or dead by color. A timer should step forward the simulation automatically using the following rules:

If a live cell has less than 2 neighbors (out of 8) the cell should die as if by loneliness.

If a live cell has 2 or 3 neighbors (out of 8) it stays alive for the next step.

If a live cell has greater than 3 neighbors (out of 8) the cell should die as if by overcrowding.

If an empty cell has exactly 3 neighbors (out of 8) it should come alive as if by reproduction.

Code 5 Criteria

- Your program should feature the following controls.

Pause – A way to toggle start and stop for the simulation.

Speeds – Simulation speed increase and decrease.

Clear – Set all cells to dead.

Activate Cells – Make a mouse function to set cells somehow.

- You must ***structure your code*** into **four classes** with **fields** and **methods** listed in the Classes section below.
- Each class should be in its own .java file.

Code 5 Classes

Class GameCell

extend Rectangle (javafx.scene.shape.Rectangle)

Fields:

```
boolean alive                int numAliveNeighbors
```

Constructor

```
GameCell(double cellSize)
```

Methods:

```
void makeAlive()                             void makeDead()
```

Class GameGrid

extend GridPane (javafx.scene.layout.GridPane)

Fields:

```
int numColumns                 int numRows                    GameCell[][] cells
```

Constructor

```
GameGrid(int cols, int rows, double cellSize)
```

Methods:

```
void clear()                                  void countCellNeighbors()

void setAliveOrDead()
```

Class GameTimer

extend AnimationTimer (javafx.animation.AnimationTimer)

Fields:

```
int stepsPerSecond             long nanosPerStep              long lastStepTime

boolean paused                 GameGrid grid
```

Constructor

```
GameTimer(GameGrid grid)
```

Overridden Methods:

```
public void handle(long now)                  public void start()
```

Methods:

```
void increaseSpeed()                          void decreaseSpeed()

void togglePause()
```

Class GameOfLife

extend Application (javafx.application.Application)

Fields:

```
GameGrid grid                  GameTimer timer
```

Overridden Methods:

```
public void start(Stage stage)
```

Methods:

```
public static void main(String[] args)       public void start(Stage stage)
```