CT255 [2D games in Java]
Week#6 Sample Solution – Starting Conway’s Game of Life

The main application class (single instance)

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.awt.image.*;

public class ConwaysLife extends JFrame implements Runnable, MouseListener {

  // member data
  private BufferStrategy strategy;
  private Graphics offscreenBuffer;
  private boolean gameState[][] = new boolean[40][40];

  // constructor
  public ConwaysLife () {

        //Display the window, centred on the screen
        Dimension ss = java.awt.Toolkit.getDefaultToolkit().getScreenSize();
        int x = ss.width/2 - 400;
        int y = ss.height/2 - 400;
        setBounds(x, y, 800, 800);
        setVisible(true);
        this.setTitle("Conway's game of life");

        // initialise double-buffering
        createBufferStrategy(2);
        strategy = getBufferStrategy();
        offscreenBuffer = strategy.getDrawGraphics();

        // register the Jframe itself to receive mouse events
        addMouseListener(this);

        // initialise the game state
        for (x=0;x<40;x++) {
         for (y=0;y<40;y++) {
           gameState[x][y]=false;
         }
        }

        // create and start our animation thread
        Thread t = new Thread(this);
        t.start();
  }

  // thread's entry point
  public void run() {
    while ( 1==1 ) {
      // 1: sleep for 1/5 sec
      try {
        Thread.sleep(200);
      } catch (InterruptedException e) { }
      // 2: animate game objects [nothing yet!]

      // 3: force an application repaint
      this.repaint();

    }
  }

  // mouse events which must be implemented for MouseListener
     public void mousePressed(MouseEvent e) {
       // determine which cell of the gameState array was clicked on
       int x = e.getX()/20;
       int y = e.getY()/20;
       // toggle the state of the cell
       gameState[x][y] = !gameState[x][y];
       // request an extra repaint, to get immediate visual feedback
       this.repaint();
     }

      public void mouseReleased(MouseEvent e) { }

      public void mouseEntered(MouseEvent e) { }

      public void mouseExited(MouseEvent e) { }

      public void mouseClicked(MouseEvent e) { }
  //

  // application's paint method
  public void paint(Graphics g) {
    g = offscreenBuffer; // draw to offscreen buffer

    // clear the canvas with a black rectangle
    g.setColor(Color.BLACK);
    g.fillRect(0, 0, 800, 800);

    // redraw all game objects
    g.setColor(Color.WHITE);
    for (int x=0;x<40;x++) {
       for (int y=0;y<40;y++) {
          if (gameState[x][y]) {
              g.fillRect(x*20, y*20, 20, 20);
          }
       }
    }

    // flip the buffers
    strategy.show();

  }

  // application entry point
  public static void main(String[] args) {
    ConwaysLife w = new ConwaysLife();
  }

}
```