Conway's Game of Life

Luke Wood

Project Description

Conway's game of life is a fascinating automaton that follows a basic set of four rules. There is a large grid full of cells which can either be living or dead. The next generation of cells' states is determined by these rules:

1. Any living cell with under 2 live neighbors will die from underpopulation
2. Any living cell with four or more neighbors will die from overpopulation
3. Any living cell with two or three neighbors will survive.
4. Any dead cell with three neighbors will become live from reproduction

Based on these rules some fascinating setups can be formed.

Implementation

I first began by creating a grid of booleans. This serves to tell the state of each cell. I then created a GUI using the javax.swing library. After this, I added a mouse and keyboard listener to allow the user to interact with the grid.

The following commands are implemented:

1. c - kill all cells
2. r - randomize status of of all cells
3. space/enter - pause the simulation
4. click - toggles the state of the cell clicked on.

Code

The code for this project consists of a single java file. The project is fairly simple so I was able to do it all within a single class file. This file is shown below.

```
import javax.swing.JFrame;
import java.lang.Thread;
import java.awt.Graphics;
import java.awt.Color;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import java.awt.event.KeyListener;
import java.awt.event.KeyEvent;
import java.awt.image.BufferedImage;
public class Life extends JFrame implements Runnable
{
	public static void main(String[] args)
	{
		Life lf = new Life();
	}

	private static boolean getRandomBool()
	{
		return Math.random() >.5;
	}
	int width;
	int height;
	int cw, ch;
	int cx = 50, cy= 50;
	boolean running = true;
	boolean[][] living = new boolean[cx][cy];
	BufferedImage toDraw;
	Graphics bg;
	public Life()
	{
		super("Conway's Game of Life");

		width = 750;
		cw = width/cx;
		height = 750;
		ch = height/cy;
		this.setSize(width,height);
		toDraw = new BufferedImage(width,height,BufferedImage.TYPE_INT_RGB);
		bg=  toDraw.getGraphics();
		this.setResizable(false);
		this.setVisible(true);

		this.setDefaultCloseOperation(EXIT_ON_CLOSE);
		this.addKeyListener(new KeyListener()
                                    {
                                    public void keyPressed(KeyEvent e)
                                    {
                                    }
                                    public void keyReleased(KeyEvent e)
                                    {
                                    if(e.getKeyCode() == KeyEvent.VK_ENTER ||
e.getKeyCode() == KeyEvent.VK_SPACE)
                                    {
                                            if(running){running = false;}
                                            else{running = true;}
                                    }
                                    else if(e.getKeyCode() == KeyEvent.VK_C)
```

```
                                {
                                        living = new boolean[cx][cy];
                                        repaint();
                                }
                                else if(e.getKeyCode() == KeyEvent.VK_R)
                                {
                                        randomize();
                                }
                                }
                                public void keyTyped(KeyEvent e){}
                                });
                this.addMouseListener(new MouseListener()
                                {
                                        boolean clicked = false;
                                        public void mousePressed(MouseEvent e)
                                        {
                                                clicked = true;
                                                int xpos = e.getX();
                                                int ypos = e.getY();
                                                xpos = xpos/cw;
                                                ypos = ypos/ch;
                                                if(xpos > 0 && xpos < cx &&ypos > 0 &&ypos <cy)
                                                {
                                                        if(living[xpos][ypos])
                                                        {
                                                                living[xpos][ypos]= false;
                                                        }
                                                        else
                                                        {
                                                                living[xpos][ypos] = true;
                                                        }
                                                        repaint();
                                                }
                                        }
                                        public void mouseReleased(MouseEvent e)
                                        {
                                                clicked = false;
                                        }

                                        public void mouseEntered(MouseEvent e){}
                                        public void mouseExited(MouseEvent e){}
                                        public void mouseClicked(MouseEvent e){}
                                });
                        init();
        }

        private void init()
        {
                for(int i = 0; i < cx; i++)
                {
                        for(int j = 0; j < cy; j++)
                        {
                                living[i][j] = getRandomBool();
                        }
                }
                Thread t = new Thread(this);
                t.start();
        }
        private void randomize()
        {
```

```
                living = new boolean[cx][cy];
                for(int i = 0; i < cx; i++)
                {
                        for(int j = 0; j < cy; j++)
                        {
                                living[i][j] = getRandomBool();
                        }
                }
                repaint();
        }
        public void run()
        {
                while(true)
                {
                        if(running)
                        {
                                tick();
                                repaint();
                                try
                                {
                                        Thread.sleep(1000);
                                }
                                catch(Exception e){}
                        }
                }
        }

        private void tick()
        {
                boolean[][] temp = new boolean[cx][cy];
                for(int i = 0; i < cx; i++)
                {
                        for(int j = 0; j < cy; j++)
                        {
                                int sum=countcell(i,j);
                                if(living[i][j])
                                {
                                        if(sum<2 || sum >=4)
                                        {
                                                temp[i][j] = false;
                                        }
                                        else
                                        {
                                                temp[i][j] = true;
                                        }
                                }
                                else
                                {
                                        if(sum == 3)
                                        {
                                                temp[i][j] = true;
                                        }
                                        else
                                        {
                                                temp[i][j] = false;
                                        }
                                }
                        }
                }
                for(int i = 0; i < cx; i++)
```

```
                {
                        for(int j = 0; j < cy; j++)
                        {
                                living[i][j] = temp[i][j];
                        }
                }
        }

        private void stop()
        {
                running = false;
        }

        private int countcell(int x, int y)
        {
                int sum = 0;

                for(int i = -1; i <= 1; i++)
                {
                        for(int j = -1; j <= 1; j++)
                        {
                                if(checkcell(x+i,y+j)&&!(i==0 && j==0))
                                {
                                        sum++;
                                }
                        }
                }
                return sum;
        }

        private boolean checkcell(int xc, int yc)
        {
                if(xc < 0 || xc>=cx || yc <0||yc >= cy)
                {
                        return false;
                }
                else
                {
                        return living[xc][yc];
                }
        }

        public void paint(Graphics g)
        {

                bg.setColor(Color.black);

                for(int i = 0; i < cx; i++)
                {
                        for(int j = 0; j < cy; j++)
                        {
                                if(living[i][j])
                                {
                                        bg.setColor(Color.black);
                                        bg.fillRect(i*cw,j*ch,cw,ch);
                                }
                                else
                                {
                                        bg.setColor(Color.white);
                                        bg.fillRect(i*cw,j*ch,cw,ch);
```

```
                                }
                        }
                }

                bg.setColor(Color.black);
                for(int i = cw; i < 750; i+=cw)
                {
                                bg.drawLine(i,0,i,750);
                                bg.drawLine(0,i,750,i);
                }

                g.drawImage(toDraw,0,0,null);

        }
}
```

Results

I created a runnable jar file for this project. It is very simple to run, all you need to do is double click the jar and it will then open up conway's game of life.

Concluding Remarks

This project has provided me with very quick results and served as a good refresher on some of the available java classes for GUI development.