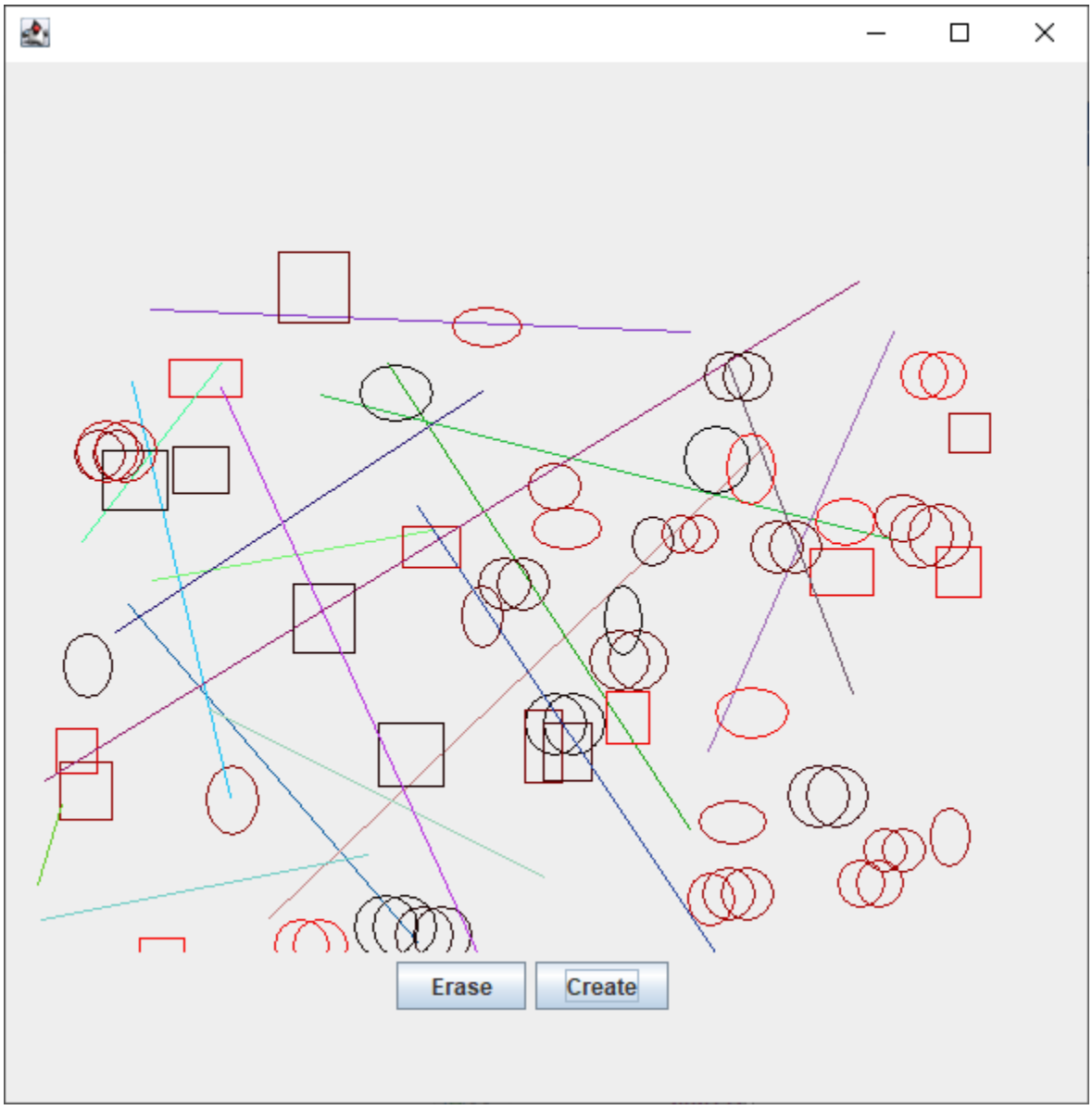
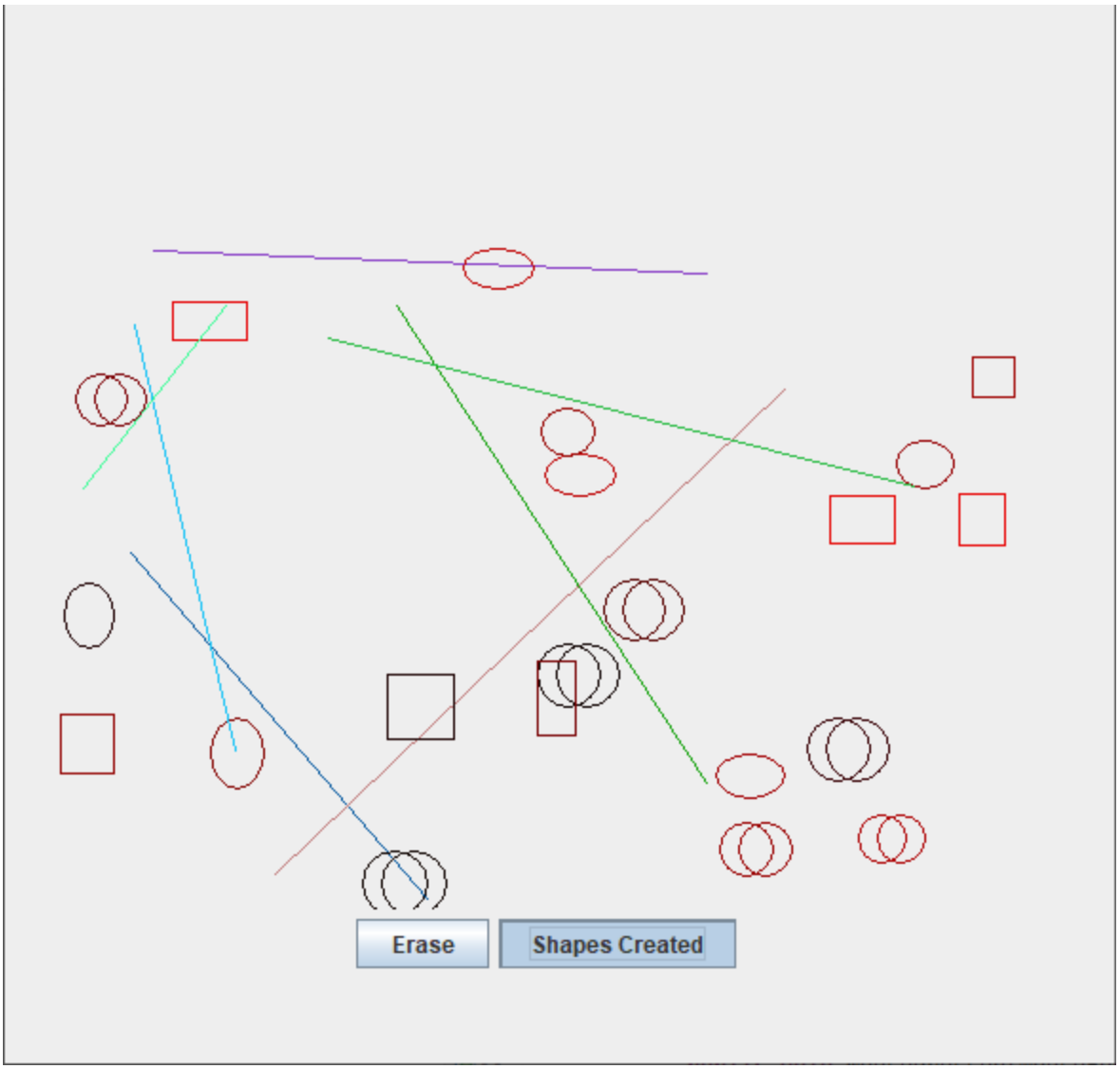
Noah Streveler

Assignment 10






```
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import javax.swing.*;
import javax.swing.border.Border;

import java.util.*;

public interface FrameBuilder {
```

```
    JPanel buildTopPanel();
    JPanel buildBottomPanel();
    JFrame buildAppFrame(String title);
}

class Composite implements FrameBuilder, ActionListener{
        String[] labels = {"Create", "Erase"};
        ArrayList<JButton> buttons = new ArrayList<JButton>();
        JTextArea output;
        ArrayList<IShape> shapes = new ArrayList<IShape>();
        JFrame frame;
        JPanel buttonPanel, displayPanel;

        @Override
        public JPanel buildTopPanel() {
                displayPanel = new JPanel();
                displayPanel.setBounds(10, 100, 550, 380);
                return displayPanel;
        }

        @Override
        public JPanel buildBottomPanel() {
                buttonPanel = new JPanel();
                buttonPanel.setBounds(10, 480, 550, 80);
                JButton eraseB = new JButton("Erase");
                eraseB.setPreferredSize(new Dimension(70,26));
                eraseB.addActionListener(this);
```

```
        IBtn createB = new ShallowBtn(new DeepBtn());
        buttonPanel.add(eraseB);
        buttonPanel.add(createB.create("Button created by an approxy.."));
        return buttonPanel;
}

@Override
public JFrame buildAppFrame(String title) {
        frame = new JFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(null);
        frame.setSize(600, 600);
        frame.setVisible(true);
        return frame;
}

@Override
public void actionPerformed(ActionEvent e) {
        String text = ((JButton) e.getSource()).getText();
        if(text.equalsIgnoreCase("erase")){
                shapes.clear();
        }
        repaint();
}

public void repaint(){
```

```
                buttonPanel.repaint();
        }
}

interface IBtn {
   JButton create(String str);
}

class DeepBtn implements IBtn{
        ArrayList<IShape> shapes = new ArrayList<IShape>();
        @Override
        public JButton create(String str) {
                JButton btn = new JButton(str);
                btn.addMouseListener(new MouseAdapter() {
                        String text;
                        @Override
                        public void mousePressed(MouseEvent e) {
                                JButton btn = (JButton) e.getSource();
                                text = btn.getText();
                                btn.setText("Shapes Created");
                                if(text.equalsIgnoreCase("create")) {
                                        shapes.add(new SimpleCircle());
                                        shapes.add(new SimpleRectangle());
                                        shapes.add(new SimpleLine());
                                        shapes.add(new DoubleCircle());
                                }
                                else if(text.equalsIgnoreCase("erase")){
```

```
                                shapes.clear();
                        }
                }

                @Override
                public void mouseReleased(MouseEvent e) {
                        JButton btn = (JButton) e.getSource();
                        btn.setText(text);
                }
        });
        return btn;
    }
}

class ShallowBtn implements IBtn{
    IBtn proxyBtn;
    public ShallowBtn(IBtn real) {
        proxyBtn = real;
    }

    @Override
    public JButton create(String str) {
        return proxyBtn.create("Create");
    }
}
```

```
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import javax.swing.*;
import java.util.*;

public class FrameDriver{
	Composite comp;
	JFrame frame;
	JPanel buttons, display;

	public static void main(String[] args) {
		FrameDriver start = new FrameDriver();
	}

	FrameDriver(){
		comp = new Composite();
		frame = comp.buildAppFrame("Shapes");
		display = comp.buildTopPanel();
		buttons = comp.buildBottomPanel();
		frame.add(buttons);
		frame.add(display);
	}
}
```

```
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import javax.swing.*;
import java.util.*;

import javax.swing.*;

public interface IShape {
	FrameDriver start = new FrameDriver();
	void draw(Graphics g);

	Color getColor();
}

class SimpleCircle implements IShape {

	SimpleCircle(){
		draw(start.display.getGraphics());
	}

	@Override
	public void draw(Graphics g) {
		Graphics2D g2 = (Graphics2D) g;
		g2.setColor(getColor());
		Random rand = new Random();
```

```
		Shape oval = new Ellipse2D.Double(rand.nextInt(500), rand.nextInt(400),
(int)(Math.random() * 20 + 20),(int)(Math.random() * 20 + 20));
		g2.draw(oval);
	}

	@Override
	public Color getColor() {
		Color c = new Color(((int)(Math.random() * 256 + 1)), ((int)(Math.random() * 5 +
1)), ((int)(Math.random() * 5 + 1)));
		return c;
	}
}

class SimpleRectangle implements IShape{

	SimpleRectangle(){
		draw(start.display.getGraphics());
	}

	@Override
	public void draw(Graphics g) {
		Graphics2D g2 = (Graphics2D) g;
		g2.setColor(getColor());
		Random rand = new Random();
		Shape rect = new Rectangle2D.Double(rand.nextInt(500), rand.nextInt(400),
(int)(Math.random() * 20 + 20),(int)(Math.random() * 20 + 20));
		g2.draw(rect);
	}
```

```
	@Override
	public Color getColor() {
		Color c = new Color(((int)(Math.random() * 256 + 1)), ((int)(Math.random() * 5 + 1)), ((int)(Math.random() * 5 + 1)));
		return c;
	}
}

class SimpleLine implements IShape{

	SimpleLine(){
		draw(start.display.getGraphics());
	}

	@Override
	public void draw(Graphics g) {
		Graphics2D g2 = (Graphics2D) g;
		Random rand = new Random();
		g2.setColor(getColor());
		g2.drawLine(rand.nextInt(500), rand.nextInt(400), rand.nextInt(500),rand.nextInt(400));
	}

	@Override
	public Color getColor() {
		Color c = new Color(((int)(Math.random() * 255)), ((int)(Math.random() * 255)), ((int)(Math.random() * 255)));
```

```
            return c;
        }
}

class DoubleCircle implements IShape{

        DoubleCircle(){
            draw(start.display.getGraphics());
        }

        @Override
        public void draw(Graphics g) {
            Graphics2D g2 = (Graphics2D) g;
            g2.setColor(getColor());
            Random rand = new Random();
            int x = rand.nextInt(500);
            int y = rand.nextInt(400);
            int size = (int)(Math.random() * 20 + 20);
            Shape oval = new Ellipse2D.Double(x, y, size,size);
            g2.draw(oval);
            oval = new Ellipse2D.Double(x + 10, y, size,size);
            g2.draw(oval);
        }

        @Override
        public Color getColor() {
```

```
                Color c = new Color(((int)(Math.random() * 256 + 1)), ((int)(Math.random() * 5 +
1)), ((int)(Math.random() * 5 + 1)));
                return c;
        }
}
```