COM6516 Object Oriented Programming and Software Design

The contents of this module has been developed by Adam Funk, Kirill Bogdanov, Mark Stevenson, Richard Clayton and Heidi Christensen

Practical 7

2D graphics and GUI

- Buttons
- Radio buttons

Creating a frame

Create a quater size window in the centre of the screen:

```
public class MyFrame extends JFrame implements ActionListener {
    …
    public MyFrame() {
        …
        Dimension screenSize =
                       Toolkit.getDefaultToolkit().getScreenSize();
        double width = screenSize.getWidth();
        double height = screenSize.getHeight();
        this.setBounds((int)width/4, (int)height/4, (int)width/2,
                       (int)height/2);

        Container contentPane = this.getContentPane();
        drawingPanel = new MyPanel();
        contentPane.add(drawingPanel, BorderLayout.CENTER);
    }
    …
}
```

Creating buttons

Create a column of buttons using `GridLayout` on the EAST side of the content pane:

```
public class MyFrame extends JFrame implements ActionListener {
    …
    public MyFrame() {
        …
        JPanel columnOfButtons = new JPanel(new GridLayout(8,1));
        for (int i = 3; i < 10; i++) {
            makeButton(columnOfButtons, String.valueOf(i), this);
        }
        makeButton(columnOfButtons, "Exit", this);
        contentPane.add(columnOfButtons, BorderLayout.EAST);
    }
    private void makeButton(JPanel p, String name, ActionListener target) {
        JButton b = new JButton(name);
        p.add(b);
        b.addActionListener(target);
    }
}
```

Link buttons with actions

Identify the button, then decide the action:

```
public class MyFrame extends JFrame implements ActionListener {
    …
   public void actionPerformed(ActionEvent e) {
        String command = e.getActionCommand();
        System.out.println(command);
        if (command.equals("Quit")) {
            System.exit(0);
        }
        else {
            int sides = Integer.parseInt(command);
            drawingPanel.setPolygon(sides);
        }
    }
}
```

Creating a panel

Recall that you need to cast your `Graphics` object to a `Graphics2D` object:

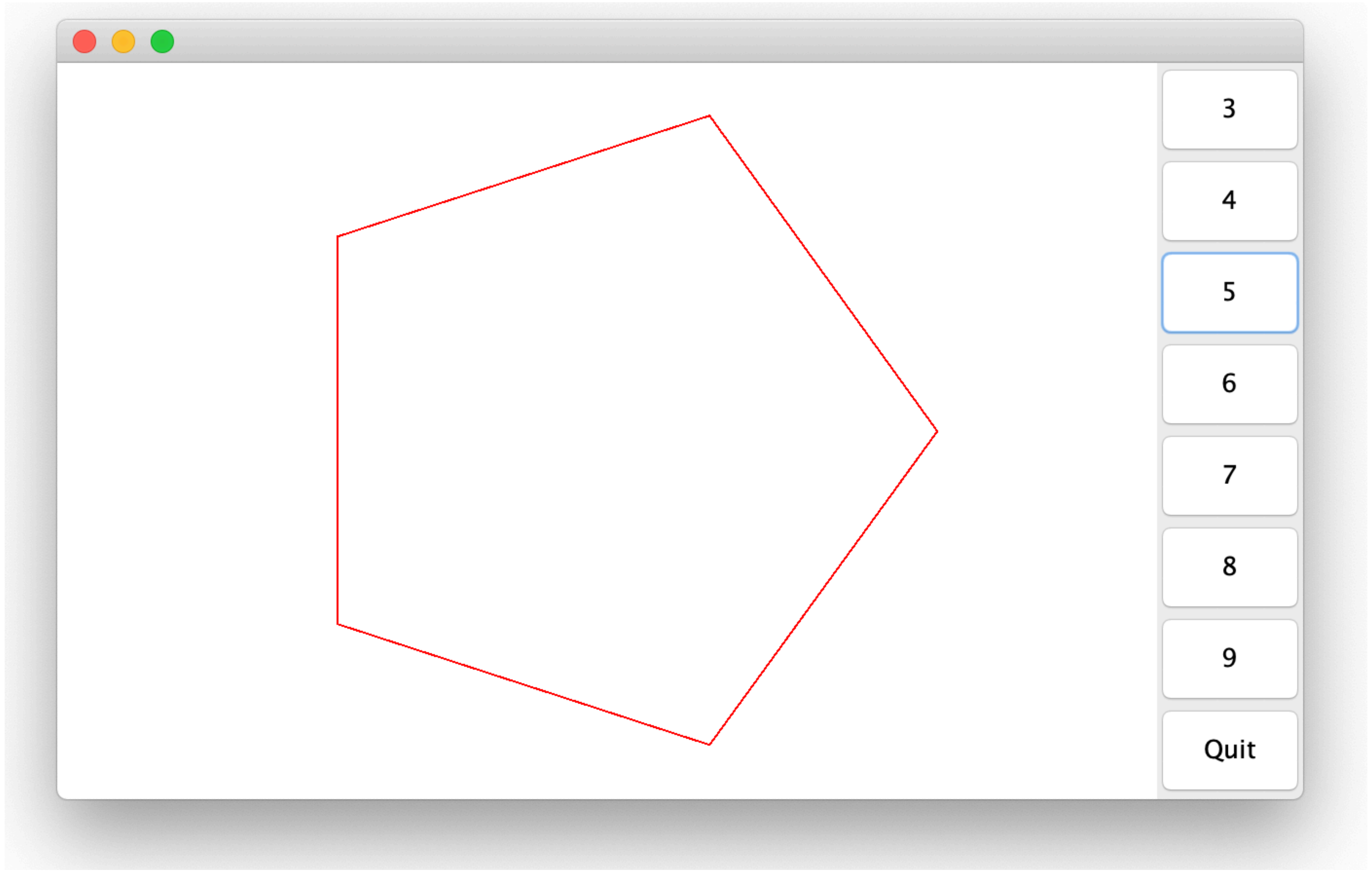
```
public class MyPanel extends JPanel {
    …
    private Polygon polygon;

    public MyPanel() {
        setBackground(Color.white);
        polygon = null;
    }
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2 = (Graphics2D) g;
        g2.setColor(Color.RED);
        if (polygon != null) {
            g2.draw(polygon);
        }
    }
}
```

Drawing a polygon

Instance field `polygon` refers to a polygon with the specified number of sides:

```
public class MyPanel extends JPanel {
    …
    public void setPolygon(int sides) {
        xCentre = this.getWidth() / 2.0;
        yCentre = this.getHeight() / 2.0;
        radius = Math.min(xCentre, yCentre) * 0.9;
        int[] xs = new int[sides];
        int[] ys = new int[sides];
        double increment = 2 * Math.PI / sides;
        for (int i = 0; i < sides; i++) {
            xs[i] = (int) (xCentre + radius * Math.cos(i * increment));
            ys[i] = (int) (yCentre + radius * Math.sin(i * increment));
        }
        polygon = new Polygon(xs, ys, sides);
        this.repaint();
    }
}
```

Buttons



Using radio buttons

Some more imports:

```
import java.awt.BorderLayout;
import java.awt.Container;
import java.awt.Dimension;
import java.awt.GridLayout;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ComponentEvent;
import java.awt.event.ComponentListener;
import javax.swing.ButtonGroup;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JRadioButton;

public class MyRadioFrame extends JFrame
        implements ActionListener, ComponentListener {
    …
}
```

Using radio buttons

`ComponentListener` handle `ComponentEvent` / `ButtonGroup` allows only one button:

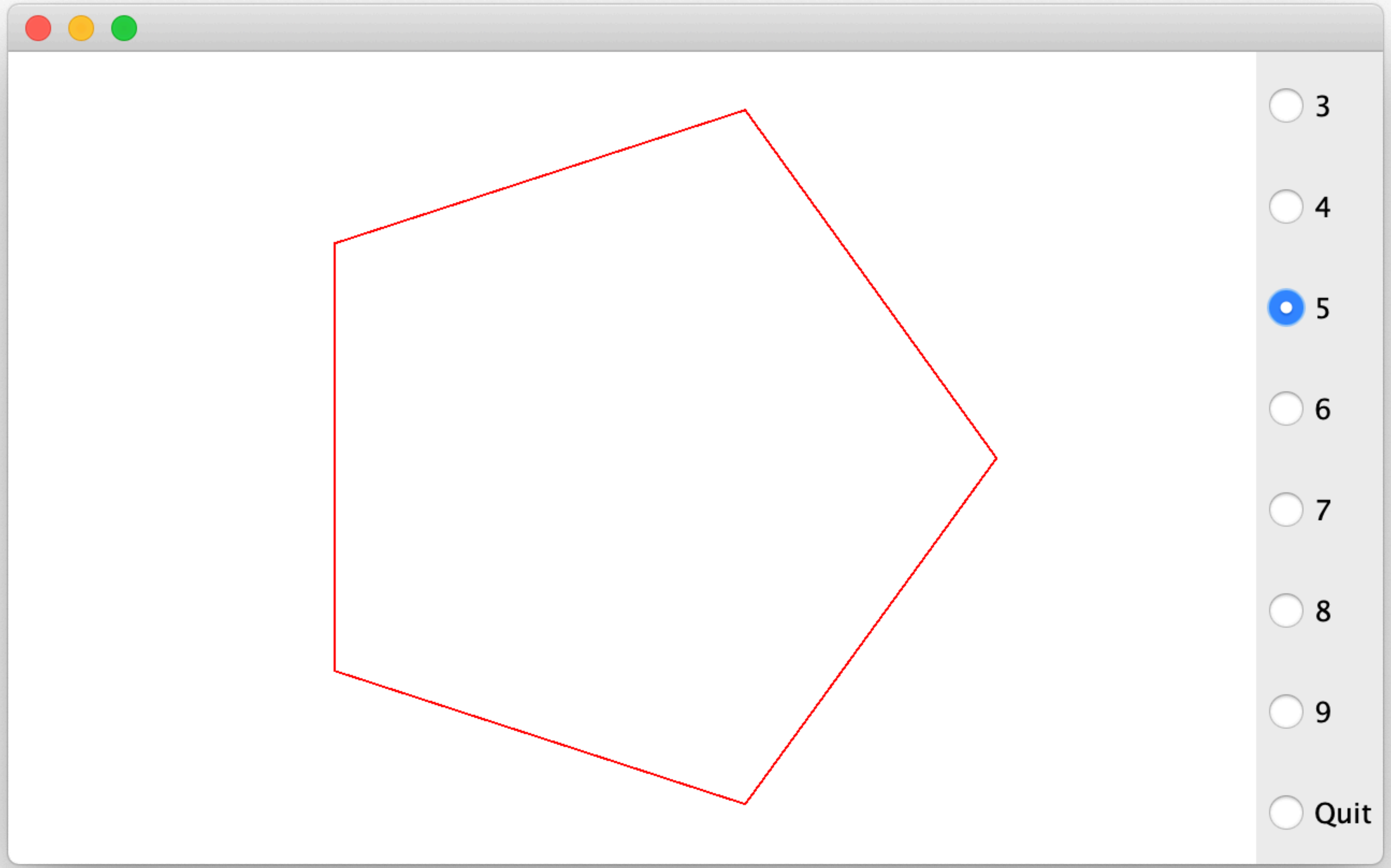
```
public class MyRadioFrame extends JFrame
        implements ActionListener, ComponentListener {
    …
    public MyRadioFrame() {
        …
        Container contentPane = this.getContentPane();
        drawingPanel = new MyPanel();
        drawingPanel.addComponentListener(this);
        contentPane.add(drawingPanel, BorderLayout.CENTER);
        JPanel columnOfButtons = new JPanel(new GridLayout(8,1));
        ButtonGroup bGroup = new ButtonGroup();
        for (int i = 3; i < 10; i++) {
            makeButton(columnOfButtons, bGroup, String.valueOf(i), this);
        }
        makeButton(columnOfButtons, bGroup, "Quit", this);
        contentPane.add(columnOfButtons, BorderLayout.EAST);
    }
}
```

Using radio buttons

`ComponentListener` handle `ComponentEvent` / `ButtonGroup` allows only one button:

```
public class MyRadioFrame extends JFrame
        implements ActionListener, ComponentListener {
    …
    private void makeButton(JPanel p, ButtonGroup g, String name,
                ActionListener target) {
        JRadioButton b = new JRadioButton(name);
        p.add(b);
        g.add(b);
        b.addActionListener(target);
    }

    @Override
    public void componentResized(ComponentEvent e) {
        if (e.getSource() == drawingPanel) {
            drawingPanel.setPolygon();
        }
    }
}
```

Radio buttons


3
4
5
6
7
8
9
Quit