

```

public class GraphicsApp extends JFrame {

    public GraphicsApp() {
        setTitle("GraphicsApp");
        setLayout(new BorderLayout());
        JPanel subpanel = new JPanel();
        JButton button1 = new JButton("Fire");
        button1.addActionListener(new FireHandler());
        subpanel.add(button1);
        JButton button2 = new JButton("Water");
        button2.addActionListener(new WaterHandler());
        subpanel.add(button2);
        add(subpanel, BorderLayout.NORTH);
        GraphicsCanvas canvas = new GraphicsCanvas();
        canvas.setPreferredSize(new Dimension(300,300));
        add(canvas, BorderLayout.CENTER);
        pack();
    }

    protected int dx = 1;
    protected int dy = 1;

    protected class GraphicsCanvas extends JComponent {
        public void paintComponent(Graphics g) {
            int w = getWidth();
            int h = getHeight();
            g.drawRect(w/4, h/4, w/2, h/2);
            int x1 = w/2;
            int y1 = h/2;
            int x2 = x1 + dx * w/4;
            int y2 = y1 + dy * h/4;
            g.drawLine(x1, y1, x2, y2);
        }
    }

    protected class FireHandler implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            dx = -dx;
            repaint();
        }
    }

    protected class WaterHandler implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            dy = -dy;
            repaint();
        }
    }

    public static void main(String[] argv) {
        new GraphicsApp().setVisible(true);
    }
}

```

1. Using the window outline below, sketch the appearance when the frame is first displayed. The faint grid lines in the figure are just to help with your sketching; they would not really be drawn by the app.



2. Using the window outline below, sketch the appearance after the “Fire” button is clicked.



```

public class GraphicsApp extends JFrame {

    public GraphicsApp() {
        setTitle("GraphicsApp");
        setLayout(new BorderLayout());
        JPanel subpanel = new JPanel();
        JButton button1 = new JButton("Fire");
        button1.addActionListener(new FireHandler());
        subpanel.add(button1);
        JButton button2 = new JButton("Water");
        button2.addActionListener(new WaterHandler());
        subpanel.add(button2);
        add(subpanel, BorderLayout.NORTH);
        GraphicsCanvas canvas = new GraphicsCanvas();
        canvas.setPreferredSize(new Dimension(300,300));
        add(canvas, BorderLayout.CENTER);
        pack();
    }

    protected int dx = -1;
    protected int dy = -1;

    protected class GraphicsCanvas extends JPanel {
        @Override
        public void paintComponent(Graphics g) {
            int w = getWidth();
            int h = getHeight();
            g.drawOval(w/4, h/4, w/2, h/2);
            int x1 = w/2;
            int y1 = h/2;
            int x2 = x1 + dx * w/4;
            int y2 = y1 + dy * h/4;
            g.drawLine(x1, y1, x2, y2);
        }
    }

    protected class FireHandler implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            dy = -dy;
            repaint();
        }
    }

    protected class WaterHandler implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            dx = -dx;
            repaint();
        }
    }

    public static void main(String[] argv) {
        new GraphicsApp().setVisible(true);
    }
}

```

1. Using the window outline below, sketch the appearance when the frame is first displayed. The faint grid lines in the figure are just to help with your sketching; they would not really be drawn by the app.



2. Using the window outline below, sketch the appearance after the “Fire” button is clicked.



```

public class GraphicsApp extends JFrame {

    public GraphicsApp() {
        setTitle("GraphicsApp");
        setLayout(new BorderLayout());
        JPanel subpanel = new JPanel();
        JButton button1 = new JButton("Fire");
        button1.addActionListener(new FireHandler());
        subpanel.add(button1);
        JButton button2 = new JButton("Water");
        button2.addActionListener(new WaterHandler());
        subpanel.add(button2);
        add(subpanel, BorderLayout.NORTH);
        GraphicsCanvas canvas = new GraphicsCanvas();
        canvas.setPreferredSize(new Dimension(300,300));
        add(canvas, BorderLayout.CENTER);
        pack();
    }

    protected int dx = 1;
    protected int dy = 1;

    protected class GraphicsCanvas extends JComponent {
        public void paintComponent(Graphics g) {
            int w = getWidth();
            int h = getHeight();
            g.drawRect(w/4, h/4, w/2, h/2);
            int x1 = w/2;
            int y1 = h/2;
            int x2 = x1 + dx * w/4;
            int y2 = y1 + dy * h/4;
            g.drawLine(x1, y1, x2, y2);
        }
    }

    protected class FireHandler implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            dx = -dx;
            repaint();
        }
    }

    protected class WaterHandler implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            dy = -dy;
            repaint();
        }
    }

    public static void main(String[] argv) {
        new GraphicsApp().setVisible(true);
    }
}

```

1. Using the window outline below, sketch the appearance when the frame is first displayed. The faint grid lines in the figure are just to help with your sketching; they would not really be drawn by the app.



2. Using the window outline below, sketch the appearance after the “Water” button is clicked.



```

public class GraphicsApp extends JFrame {

    public GraphicsApp() {
        setTitle("GraphicsApp");
        setLayout(new BorderLayout());
        JPanel subpanel = new JPanel();
        JButton button1 = new JButton("Fire");
        button1.addActionListener(new FireHandler());
        subpanel.add(button1);
        JButton button2 = new JButton("Water");
        button2.addActionListener(new WaterHandler());
        subpanel.add(button2);
        add(subpanel, BorderLayout.NORTH);
        GraphicsCanvas canvas = new GraphicsCanvas();
        canvas.setPreferredSize(new Dimension(300,300));
        add(canvas, BorderLayout.CENTER);
        pack();
    }

    protected int dx = -1;
    protected int dy = -1;

    protected class GraphicsCanvas extends JPanel {
        @Override
        public void paintComponent(Graphics g) {
            int w = getWidth();
            int h = getHeight();
            g.drawOval(w/4, h/4, w/2, h/2);
            int x1 = w/2;
            int y1 = h/2;
            int x2 = x1 + dx * w/4;
            int y2 = y1 + dy * h/4;
            g.drawLine(x1, y1, x2, y2);
        }
    }

    protected class FireHandler implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            dy = -dy;
            repaint();
        }
    }

    protected class WaterHandler implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            dx = -dx;
            repaint();
        }
    }

    public static void main(String[] argv) {
        new GraphicsApp().setVisible(true);
    }
}

```

1. Using the window outline below, sketch the appearance when the frame is first displayed. The faint grid lines in the figure are just to help with your sketching; they would not really be drawn by the app.



2. Using the window outline below, sketch the appearance after the "Water" button is clicked.

