ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

Object Oriented Programming

Chapter 4: Case Study

ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

System Description

Customer Description

Customer Say’s:
Create a software application, where user can draw simple graphics like Line, Ellipse and Rectangle. User can change the background and line color, also the border thickness. User can move and resize and delete the drawing items. Finally, user can save and load the drawing.

ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

Requirements Analysis

Requirements: (1) Understand

- ➔ Create a software application
- ➔ User can draw simple graphics like Line, Ellipse and Rectangle.
- ➔ User can change the Background, Line Color and Border Thickness.
- ➔ User can Move, Resize and Delete the drawing items.
- ➔ User can Save and Load the drawing.

Requirements: (2) Analyze

- ➔ Create a software application (**Language?**)
- ➔ User (**Who?**) can draw simple graphics like (Line, Ellipse and Rectangle) (**Any More?**)
- ➔ User can change the Background (**More Details?**), Line (**Line Object Only or Border?**) Color and Border Thickness (**Any More?**).
- ➔ User can Move, Resize (**Drag or Dialogs?**) and Delete (**Any More?**) the drawing items.
- ➔ User can (Save and Load) (**Any More? Format**?) the drawing.
- ➔ (**Any More?**)

Requirements: (3) Questioner

- Create a software application.
  - **Which OS is required?**
- User can draw simple graphics like Line, Ellipse and Rectangle.
  - **Students or Children?**
  - **Is there any more drawing items?**
- User can change the Background, Line Color and Border Thickness.
  - **What type of background is required to change?**
  - **(Line Color) do you mean Line Object or any border?**
  - **Is there any more any more drawing properties?**

Requirements: (3) Questioner

- User can Move, Resize and Delete the drawing items.
  - How the user perform the move and resize?
  - Is there any more edit action over the drawing items?
- User can Save and Load the drawing.
  - Is it required to open the saved files by other programs?
  - Is there any more action over the drawing?
- Any More Requirements?

Requirements: (4) Answers

- Create a software application.
  - **Which OS is required? Any → Java**
- User can draw simple graphics like Line, Ellipse and Rectangle.
  - **Students or Children? Students (no need for animation)**
  - **Is there any more drawing items? No**
- User can change the Background, Line Color and Border Thickness.
  - **What type of background is required to change? Color only**
  - **(Line Color) do you mean Line Object or any border? Any border color**
  - **Is there any more any more drawing properties? No**

Requirements: (4) Questioner

- User can Move, Resize and Delete the drawing items.
  - **How the user perform the move and resize? Drag**
  - **Is there any more edit action over the drawing items? No**
- User can Save and Load the drawing.
  - **Is it required to open the saved files by other programs? No → Serialization**
  - **Is there any more action over the drawing? New, Save As and Close**
- Any More Requirements?
  - **Software must show the cursor position and the selected object width and height**

Requirements: (5) Final

- Keywords
  - Java: Programming Language that support any computer with any OS
  - Drawing Object: Drawing Item
  - Drawing: A set of drawing items
  - Drawing System: The drawing software

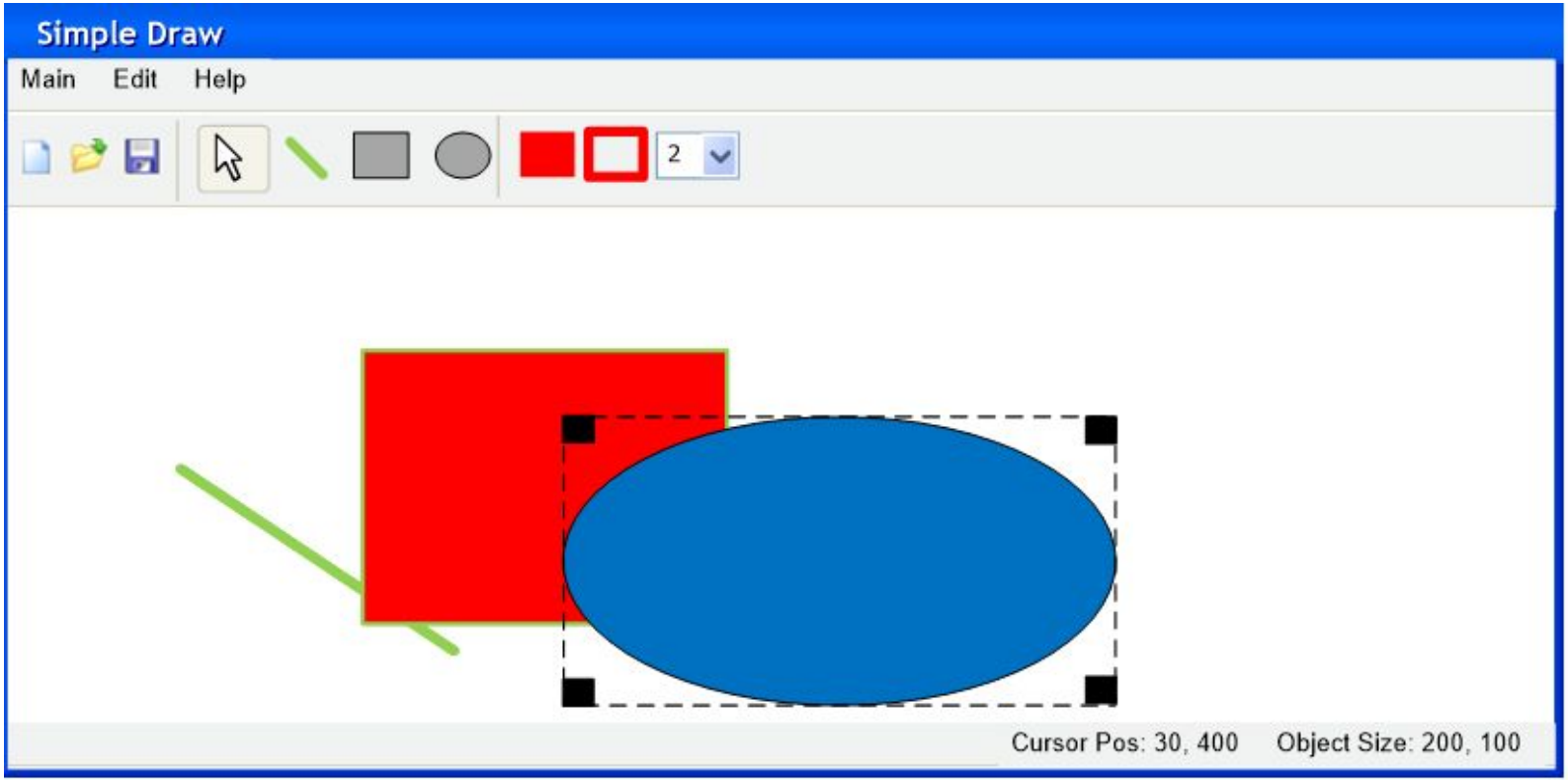
Requirements: (5) Final

1. Create a java software application
2. User can draw Line object
3. User can draw Ellipse object
4. User can draw Rectangle object
5. User can change Drawing Object Background Color
6. User can change Drawing Object Border Color
7. User can change Drawing Object Border Thickness
8. User can Move the drawing object using mouse
9. User can Resize the drawing object using mouse

Requirements: (5) Final

10. User can delete the drawing object
11. User can make New drawing
12. User can Open a previously saved drawing
13. User can Save the drawing using a custom format
14. User can Save As opened drawing to another file
15. User can close the drawing system
16. Show cursor position
17. Show selected drawing object width and height

Requirements: (6) User Interface



ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

Basic Draw Sample

What to do?

Illustrate the moving draw object idea

Objectives

- Study related technologies and techniques
- Focus on complex or new features (high risk)
- Helps to create accurate design
- Helps to create accurate plan and cost estimation

Draw Object Interface

```
import java.awt.*;

interface DrawObject {
    public void draw(Graphics g);
    public void mousePressed(Point p);
    public void mouseDragged(Point p);
    public void mouseReleased(Point p);
    public boolean hitTest(Point p);
    public void moveBy(int dx, int dy);
    public void unSelect();
}
```

Corner Class

```
import java.awt.*;
import java.io.Serializable;

public class Corner implements DrawObject, Serializable {

    private final int radius = 25;
    private Point center;
    private boolean selected = false;
    private Point lastPoint;

    public Corner(Point center) {
        this.center = center;
    }
    public void draw(Graphics g) {
        g.setColor((selected) ? Color.black : Color.white);
        g.fillOval(center.x - radius, center.y - radius, 2 * radius, 2 * radius);
        g.setColor(Color.black);
        g.drawOval(center.x - radius, center.y - radius, 2 * radius, 2 * radius);
    }
    ...
}
```

Corner Class

```
public class Corner implements DrawObject, Serializable {
    ...
    public void mousePressed(Point p) { lastPoint = p;}
    public void mouseDragged(Point p) {
        moveBy(p.x - lastPoint.x, p.y - lastPoint.y);
        lastPoint = p;
    }
    public void mouseReleased(Point p) {}
    public boolean hitTest(Point p) {
        int dx = p.x - center.x, dy = p.y - center.y;
        int d = (int) Math.sqrt(Math.pow(dx, 2) + Math.pow(dy, 2));
        return (selected = d < radius);
    }
    public void moveBy(int dx, int dy) { center.x += dx; center.y += dy;}
    public void unSelect() { selected = false;}
}
```

Draw Panel

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

public class DrawPanel extends JPanel implements MouseListener, MouseMotionListener, KeyListener {

    private ArrayList<DrawObject> drawObjects = new ArrayList<DrawObject>();
    private DrawObject selectedDrawObject = null;
    private enum DrawMode {EDIT, ADDCIRCLE};
    private DrawMode drawMode = DrawMode.ADDCIRCLE;
    public DrawPanel(JFrame frame) {
        setBackground(Color.white);
        setPreferredSize(new Dimension(800, 600));
        addMouseListener(this);
        addMouseMotionListener(this);
        addKeyListener(this);
        setFocusable(true);
    }
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        for(DrawObject o: drawObjects) o.draw(g);
    }
    ...
}
```

Draw Panel

```
public class DrawPanel extends JPanel implements ... {
    ...
    public void mousePressed(MouseEvent e) {
        Point p = new Point(e.getX(), e.getY());
        if (drawMode == DrawMode.ADDCIRCLE) drawObjects.add(new Corner(p));
        else {
            selectedDrawObject = null;
            for (DrawObject o : drawObjects) o.unSelect();
            for (int i = drawObjects.size() - 1; i >= 0; i--) {
                if (drawObjects.get(i).hitTest(p)) { selectedDrawObject = drawObjects.get(i); break; }
            }
            if (selectedDrawObject != null) selectedDrawObject.mousePressed(p);
        }
        repaint();
    }
    public void mouseReleased(MouseEvent e) {
        Point p = new Point(e.getX(), e.getY());
        if (selectedDrawObject != null)
            selectedDrawObject.mouseReleased(p);
        repaint();
    }
    ...
}
```

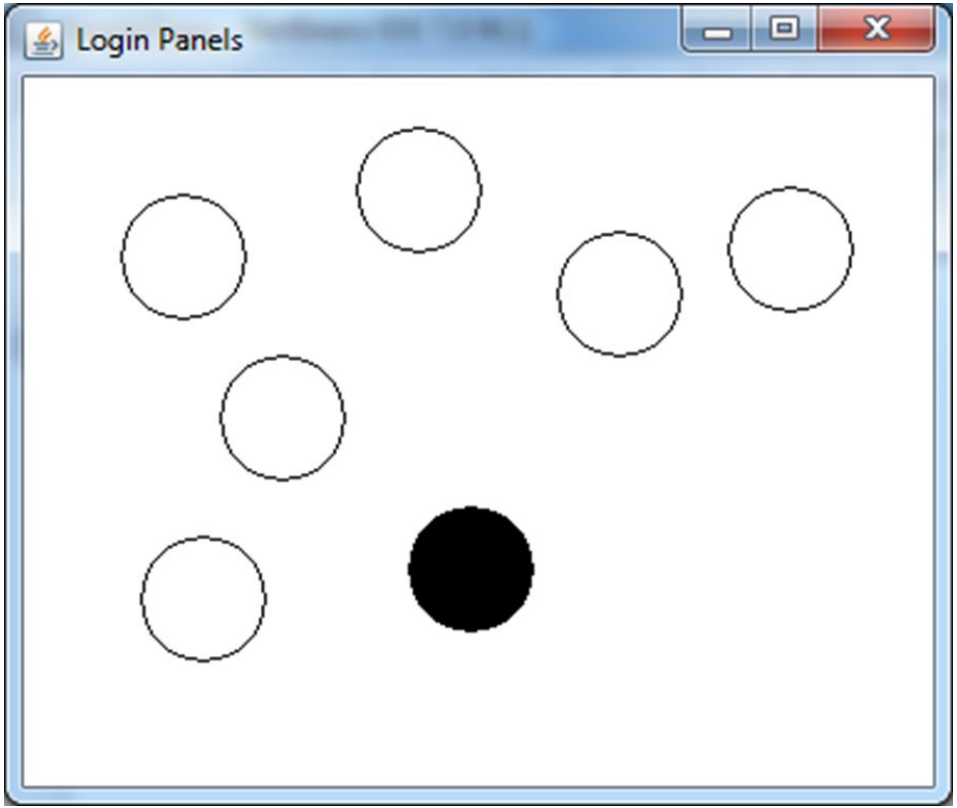
Draw Panel

```
public class DrawPanel extends JPanel implements ... {
    ...
    public void mouseClicked(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}

    public void mouseDragged(MouseEvent e) {
        Point p = new Point(e.getX(), e.getY());
        if (selectedDrawObject != null)
            selectedDrawObject.mouseDragged(p);
        repaint();
    }
    public void mouseMoved(MouseEvent e) {}

    public void keyPressed(KeyEvent e) {
        if (e.getKeyCode() == KeyEvent.VK_A)
            drawMode = DrawMode.ADDCIRCLE;
        else if (e.getKeyCode() == KeyEvent.VK_E)
            drawMode = DrawMode.EDIT;
    }
    public void keyTyped(KeyEvent e) {}
    public void keyReleased(KeyEvent e) {}
}
```

Run Application

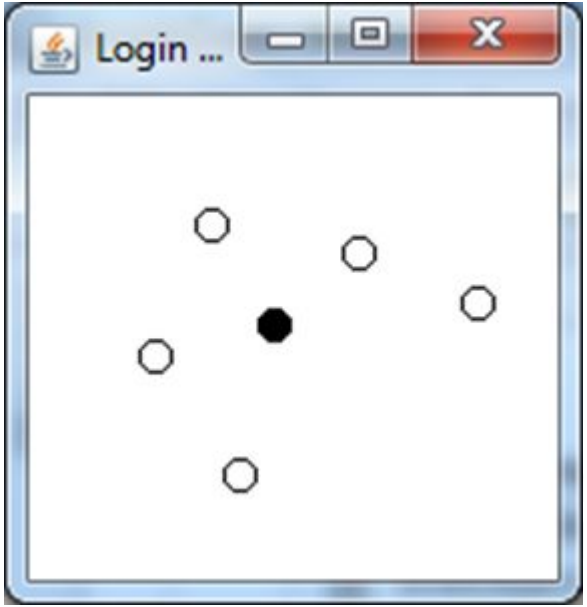
```
import javax.swing.*;

public class App {
    public static void main(String[] args) throws Exception {

        JFrame frame = new JFrame("Basic Draw Sample");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        DrawPanel panel = new DrawPanel(frame);
        frame.getContentPane().add(panel);
        frame.pack();
        frame.setVisible(true);
    }
}
```

Run Application



Radius = 25

Radius = 5



```
public class DrawPanel extends JPanel implements MouseListener,
MouseMotionListener, ActionListener {
    private ArrayList<DrawObject> drawObjects = new ArrayList<DrawObject>();
    private DrawObject selectedDrawObject = null;
    private JToolBar toolBar = new JToolBar();
    private JButton newButton, openButton, saveButton;
    private JToggleButton mouseButton, cornerButton;
    private enum DrawMode {EDIT, ADDCIRCLE};
    private DrawMode drawMode = DrawMode.ADDCIRCLE;
    private JPanel statusBar = new JPanel();
    private JLabel positionStatus = new JLabel();
    ...
}
```

Adding Toolbar

Adding Toolbar

```
public DrawPanel(JFrame frame) {
    setBackground(Color.white);
    setPreferredSize(new Dimension(800, 600));
    addMouseListener(this);
    addMouseMotionListener(this);
    toolBar.add(newButton = new JButton(new ImageIcon("new.png")));
    toolBar.add(openButton = new JButton(new ImageIcon("open.png")));
    toolBar.add(saveButton = new JButton(new ImageIcon("save.png")));
    toolBar.addSeparator();
    toolBar.add(mouseButton = new JToggleButton(new ImageIcon("mouse.png")));
    toolBar.add(cornerButton = new JToggleButton(new ImageIcon("corner.png")));
    setLayout(new BorderLayout());
    statusBar.setPreferredSize(new Dimension(25, 25));
    add(toolBar, BorderLayout.NORTH);
    add(statusBar, BorderLayout.SOUTH);
    ButtonGroup group = new ButtonGroup();
    group.add(mouseButton);
    group.add(cornerButton);
    mouseButton.addActionListener(this);
    cornerButton.addActionListener(this);
    cornerButton.setSelected(true);
    newButton.addActionListener(this);
    openButton.addActionListener(this);
    saveButton.addActionListener(this);
    positionStatus.setText("0, 0");
    statusBar.add(positionStatus);
}
...
public void mouseMoved(MouseEvent e) {
    positionStatus.setText("" + e.getX() + ", " + e.getY());
}
```

Adding Menu

```
private JMenuBar menuBar;
private JMenuItem newMenuItem, openMenuItem, saveMenuItem, closeMenuItem;
public DrawPanel(JFrame frame) {
    JMenu menu;
    menuBar = new JMenuBar();
    menuBar.add(menu = new JMenu("Main"));
    menu.add(newMenuItem = new JMenuItem("New"));
    newMenuItem.addActionListener(this);
    menu.add(openMenuItem = new JMenuItem("Open"));
    openMenuItem.addActionListener(this);
    menu.add(saveMenuItem = new JMenuItem("Save"));
    saveMenuItem.addActionListener(this);
    menu.add(closeMenuItem = new JMenuItem("Close"));
    closeMenuItem.addActionListener(this);
    frame.getRootPane().
    setJMenuBar(menuBar);
    ...
}
```

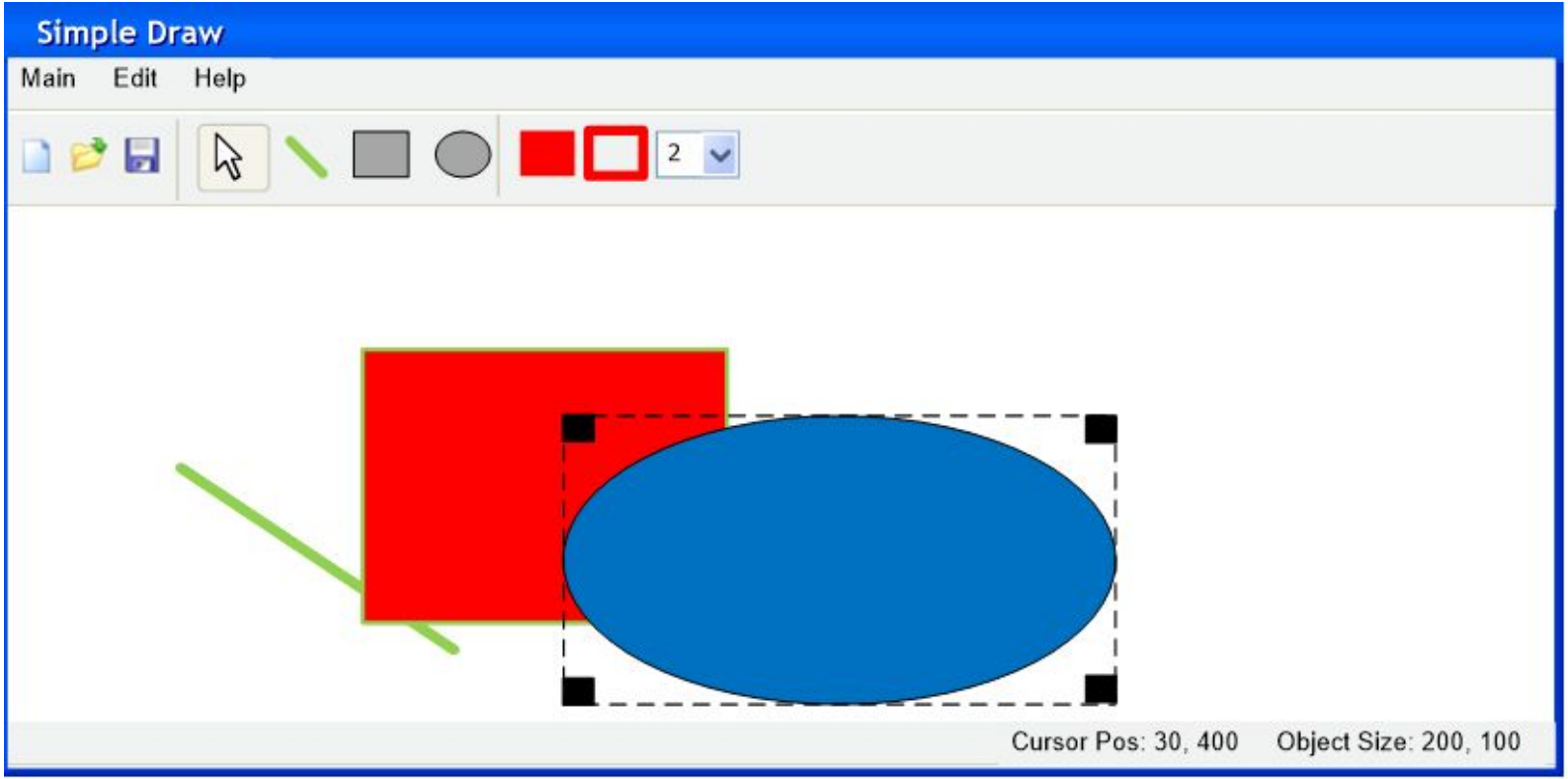
Support Save and Load

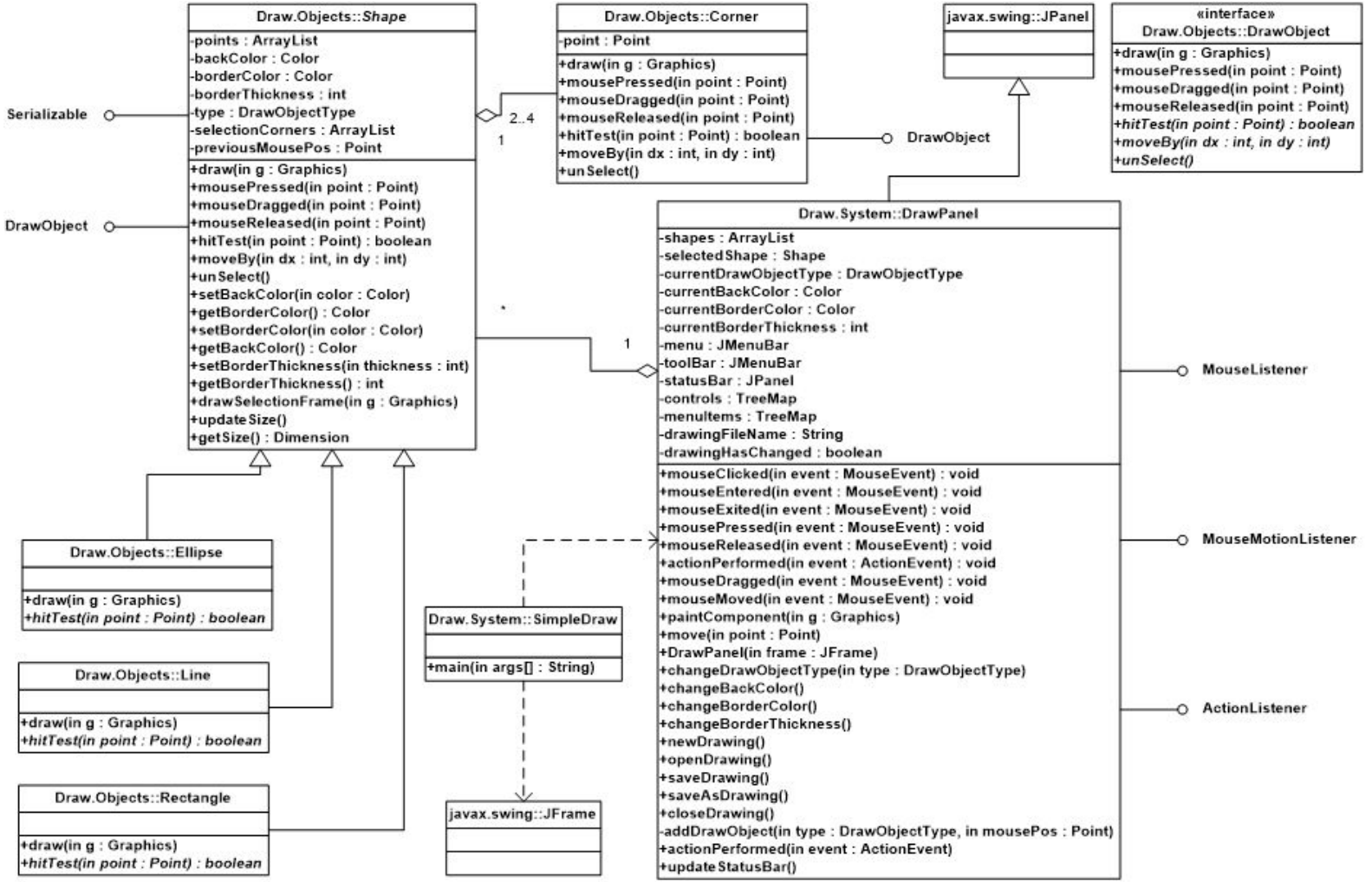
```
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == mouseButton)
        drawMode = DrawMode.EDIT;
    else if (e.getSource() == cornerButton)
        drawMode = DrawMode.ADDCIRCLE;
    else if (e.getSource()==newMenuItem || e.getSource() == newButton)
        drawObjects.clear();
    else if (e.getSource()==openMenuItem || e.getSource() == openButton) {
        FileInputStream fis;
        try {
            fis = new FileInputStream("c:\\serial.dat");
            ObjectInputStream ois = new ObjectInputStream(fis);
            drawObjects = (ArrayList<DrawObject>) ois.readObject();
            ois.close();
        } catch (Exception ex) {
        }
    } else if (e.getSource()==saveMenuItem || e.getSource() == saveButton) {
        try {
            FileOutputStream fos = new FileOutputStream("c:\\serial.dat");
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeObject(drawObjects);
            oos.flush();
            oos.close();
        } catch (Exception ex) {
        }
    }
    repaint();
}
```

ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

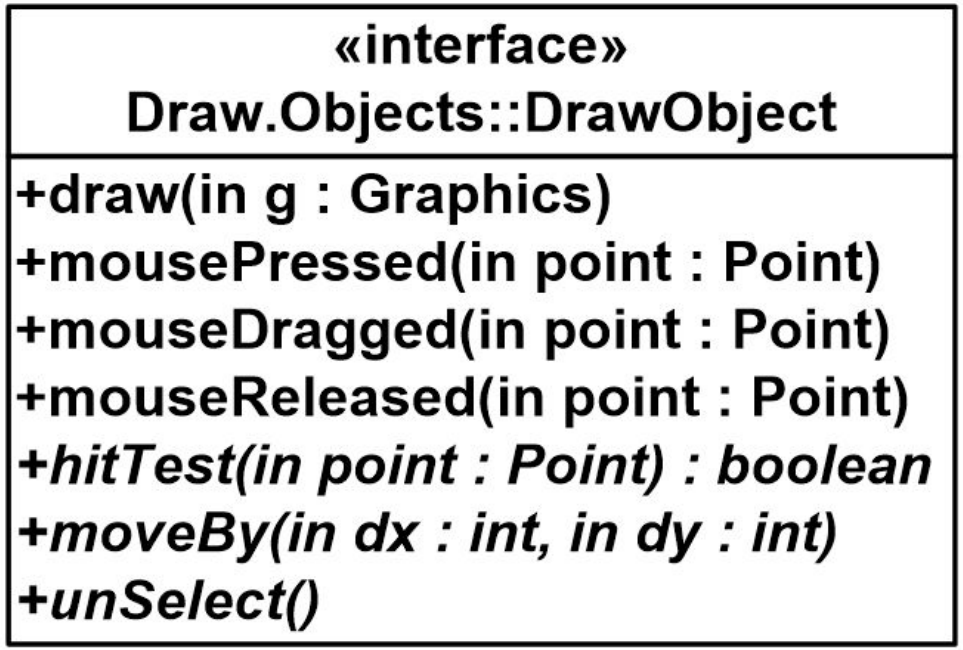
System Design

Remember, UI

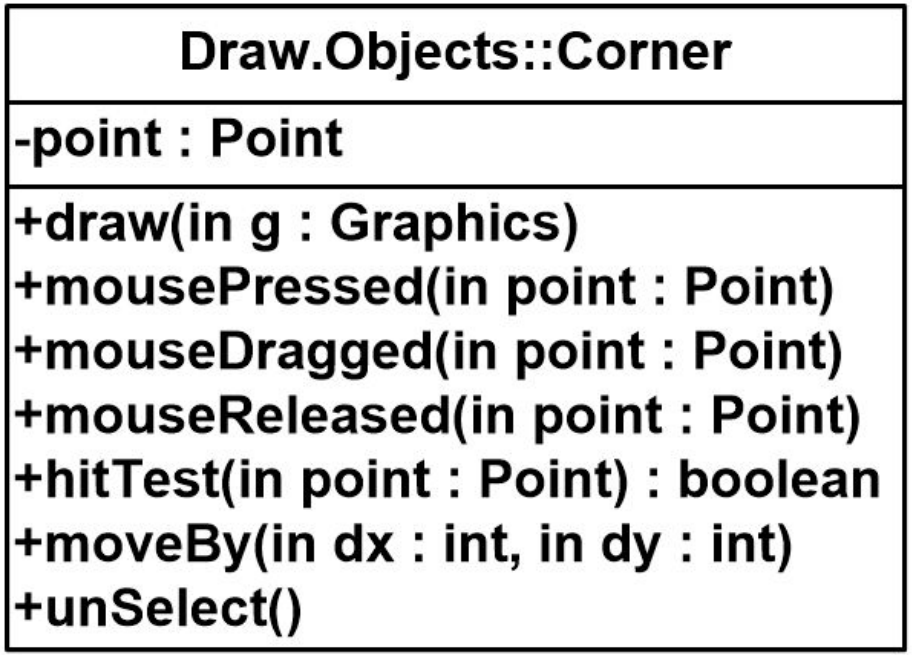

Simple Draw
Main
Edit
Help
2
Cursor Pos: 30, 400
Object Size: 200, 100


Draw.Objects::Shape
-points : ArrayList
-backColor : Color
-borderColor : Color
-borderThickness : int
-type : DrawObjectType
-selectionCorners : ArrayList
-previousMousePos : Point
+draw(in g : Graphics)
+mousePressed(in point : Point)
+mouseDragged(in point : Point)
+mouseReleased(in point : Point)
+hitTest(in point : Point) : boolean
+moveBy(in dx : int, in dy : int)
+unSelect()
+setBackColor(in color : Color)
+getBorderColor() : Color
+setBorderColor(in color : Color)
+getBackColor() : Color
+setBorderThickness(in thickness : int)
+getBorderThickness() : int
+drawSelectionFrame(in g : Graphics)
+updateSize()
+getSize() : Dimension
Draw.Objects::Corner
-point : Point
+draw(in g : Graphics)
+mousePressed(in point : Point)
+mouseDragged(in point : Point)
+mouseReleased(in point : Point)
+hitTest(in point : Point) : boolean
+moveBy(in dx : int, in dy : int)
+unSelect()
javax.swing::JPanel
«interface»
Draw.Objects::DrawObject
+draw(in g : Graphics)
+mousePressed(in point : Point)
+mouseDragged(in point : Point)
+mouseReleased(in point : Point)
+hitTest(in point : Point) : boolean
+moveBy(in dx : int, in dy : int)
+unSelect()
Serializable
DrawObject
2..4
1
DrawObject
Draw.System::DrawPanel
-shapes : ArrayList
-selectedShape : Shape
-currentDrawObjectType : DrawObjectType
-currentBackColor : Color
-currentBorderColor : Color
-currentBorderThickness : int
-menu : JMenuBar
-toolBar : JMenuBar
-statusBar : JPanel
-controls : TreeMap
-menuItems : TreeMap
-drawingFileName : String
-drawingHasChanged : boolean
+mouseClicked(in event : MouseEvent) : void
+mouseEntered(in event : MouseEvent) : void
+mouseExited(in event : MouseEvent) : void
+mousePressed(in event : MouseEvent) : void
+mouseReleased(in event : MouseEvent) : void
+actionPerformed(in event : ActionEvent) : void
+mouseDragged(in event : MouseEvent) : void
+mouseMoved(in event : MouseEvent) : void
+paintComponent(in g : Graphics)
+move(in point : Point)
+DrawPanel(in frame : JFrame)
+changeDrawObjectType(in type : DrawObjectType)
+changeBackColor()
+changeBorderColor()
+changeBorderThickness()
+newDrawing()
+openDrawing()
+saveDrawing()
+saveAsDrawing()
+closeDrawing()
-addDrawObject(in type : DrawObjectType, in mousePos : Point)
+actionPerformed(in event : ActionEvent)
+updateStatusBar()
*
1
MouseListener
MouseMotionListener
ActionListener
Draw.Objects::Ellipse
+draw(in g : Graphics)
+hitTest(in point : Point) : boolean
Draw.Objects::Line
+draw(in g : Graphics)
+hitTest(in point : Point) : boolean
Draw.Objects::Rectangle
+draw(in g : Graphics)
+hitTest(in point : Point) : boolean
Draw.System::SimpleDraw
+main(in args[] : String)
javax.swing::JFrame

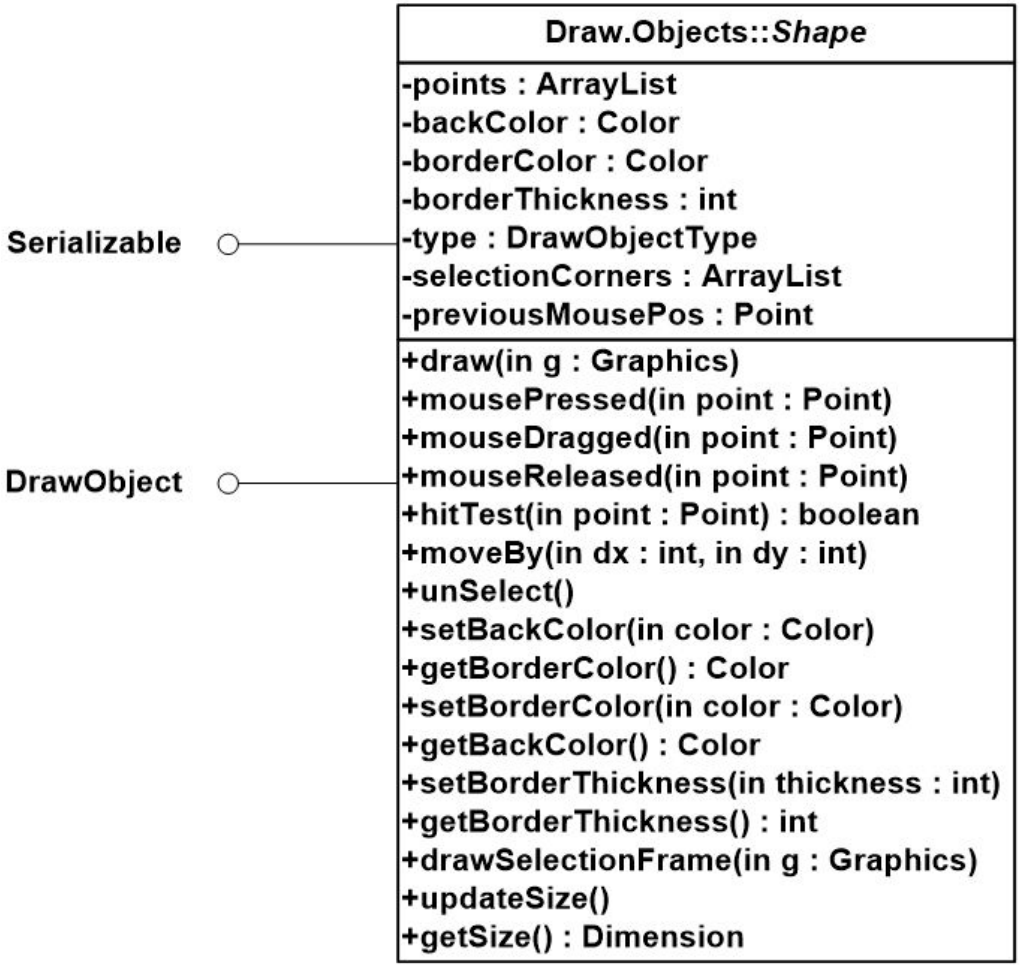
Draw Object Interface



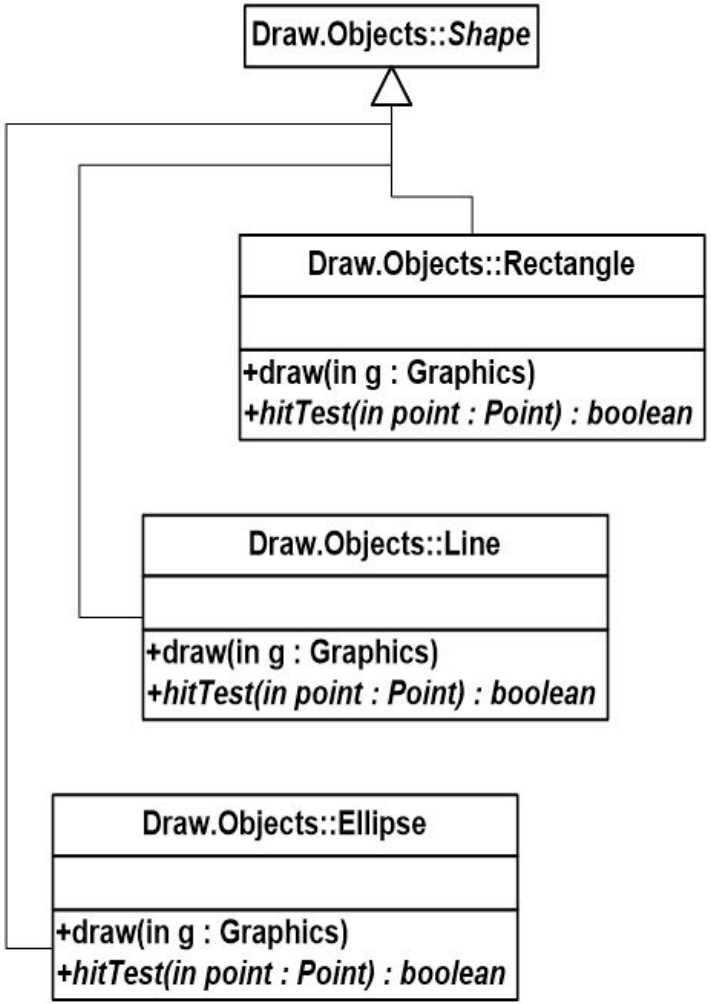
Corner Object



Shape Object



Drawing Objects



Drawing System