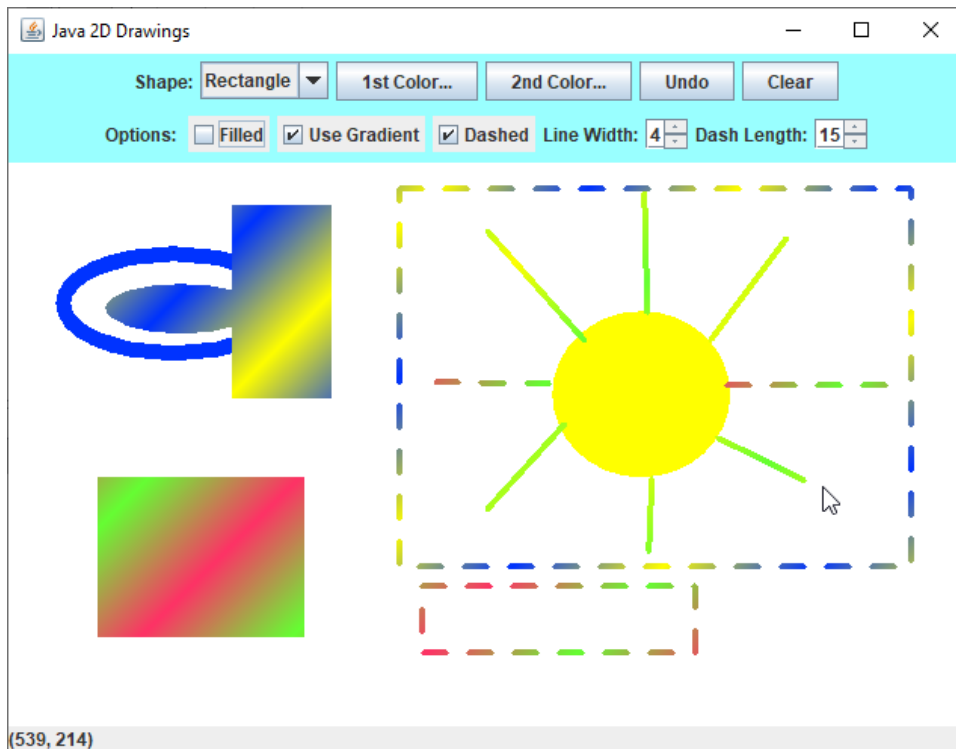


## PA5 Prompt



The application will contain the following elements:

- a) a combo box for selecting the shape to draw, a line, oval, or rectangle.
- b) two JButtons that each show a JColorChooser dialog to allow the user to choose the first and second color in the gradient.
- c) an Undo button to undo the last shape drawn.
- d) a Clear button to clear all shapes from the drawing.
- e) a checkbox which specifies if the shape should be filled or unfilled.
- f) a checkbox to specify whether to paint using a gradient.
- g) a checkbox for specifying whether to draw a dashed or solid line.
- h) a JSpinner for entering the Stroke width.
- i) a JSpinner for entering the Stroke dash length.
- j) a JPanel on which the shapes are drawn.
- k) a status bar JLabel at the bottom of the frame that displays the current location of the mouse on the draw panel.

Note: Your application should look like the example above and you must use the MyShapes class hierarchy provided.

If the user selects to draw with a gradient, set the Paint on the shape to be a gradient of the two colors chosen by the user. If the user does not chose to draw with a gradient, then Paint with a solid color of the 1st Color. The following code can create a gradient

paint object:

```
Paint paint = new GradientPaint(0, 0, color1, 50, 50, color2, true);
```

To set the stroke for a line to be drawn, you can use the following code:

```
        if (dashCheckBox.isSelected())
        {
            stroke = new BasicStroke(lineWidth, BasicStroke.CAP_ROUND,
BasicStroke.JOIN_ROUND, 10, dashLength, 0);
        } else
        {
            stroke = new BasicStroke(lineWidth, BasicStroke.CAP_ROUND,
BasicStroke.JOIN_ROUND);
        }
```

Where the first stroke line creates a dashed line and dashLength is a one element float array with the dash length in the first element. The second stroke line creates an undashed line with the line width specified from the GUI.

Note: When dragging the mouse to create a new shape, the shape should be drawn as the mouse is dragged.

A template project has been provided for you in Canvas in Java2DDrawingApplicationTemplate.zip. This project contains a MyShapes hierarchy that is a complete shape hierarchy for drawing a line, rectangle or oval. You must use this MyShapes hierarchy. A template for the Drawing Application Frame is also provided along with a template for the DrawPanel inner class. You do not need to use these templates if you so choose.

In the paintComponent(Graphics g) method of the DrawPanel, to loop through and draw each shape created by the user, you will loop through an ArrayList of MyShapes, that you built, and call the draw(Graphics2D g2d) method for each shape. The draw method is already implemented in the MyShapes hierarchy.

Note: You do not need to create an event handler for each component in the top two lines of the frame. You only need to create event handlers for the buttons. You can get the values out of all the other components in the top two lines, when the user presses the mouse button on the DrawPanel. At that time, you have all the information you need to create a new Myshapes object.

**Note: Do not use the NetBeans GUI generator for this assignment.**