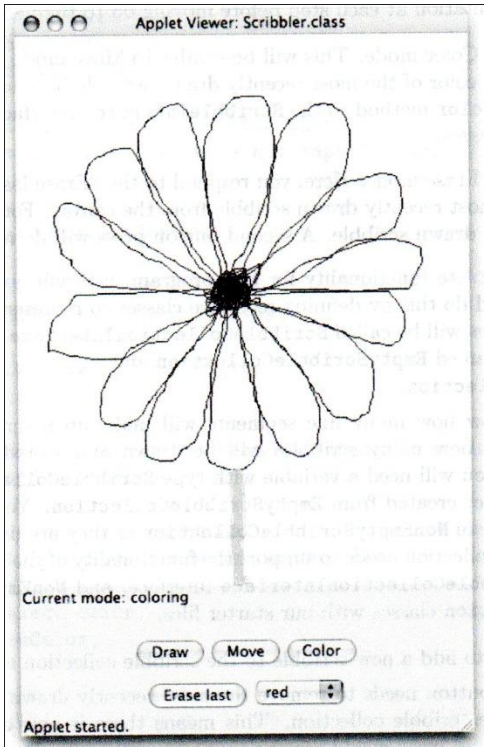
# Programming Exercise

# Scribbler

**Objective**: To gain more experience using arrays and data structures

You will be implementing a drawing program we call "Scribbler". A sample of what your program's interface might look like is show below.




The Scribbler program has three modes: Draw mode, Move mode and Color mode. The modes behave as follows:

Draw mode: as the user drags the mouse around the canvas, the program will draw lines following the path of the mouse.

Move mode: in this mode the user can reposition a scribble by pressing the mouse while touching some part of the scribble and then dragging the scribble to a new location.

Color mode: clicking on any scribble while in this mode should change its color to that selected in the color menu.

The program starts in Draw mode. Draw mode, Move mode and Color mode are selected by pressing buttons, and the color used by Color is selected by choosing a color from a JComboBox filled with color names.

The program also has an “Erase Last” button that will erase the most recently drawn scribble. Pressing the button repeatedly will erase scribbles in the reverse order of which they were drawn.

**How to Proceed** We will provide you with a working but incomplete Scribble program as a starter for this exercise. It supports only Draw mode and a simplified Move mode that is capable of moving only the most recently draw scribble. You will need to add code that will manage your scribbles to allow the various modes and the “Erase Last” button to work correctly.

There are a number of step-by-step approaches you could take to complete this program, but it is important that you have a plan and that you add functionality one step at a time. Here is one possible order of tasks. We recommend that you develop and test your program incrementally – make sure you have a working implementation at each step before moving on to the next.

1. Implement a simplified Color mode. This will be similar to Move mode supported by the starter code, except that you set the color of the most recently drawn scribble instead of moving it. You will need to add a setColor method on the Scribble class and call it from Scribbler. The setColor method will change the color of every line segment in the scribble.

2. Implement a simplified Erase mode. Here you respond to “Erase Last” button’s actionPerformed event by deleting the most recently drawn scribble from the canvas. For now, you will only be able to erase the most recently drawn scribble. A second button press will do nothing. You will need to add an erase method on the Scribble class and call it from Scribbler class. The erase method should call removeFromCanvas for every line in the scribble.

3. To implement the complete functionality of this program, you will need to keep track of a number of scribbles. The ScribbleCollection class will have an array of Scribbles and keep track of the number of scribbles in the array.

- Draw mode needs to add a new scribble to the scribble collection when the dragging is done (in the onMouseRelease method).
- The “Erase Last” button needs to remove the most recently drawn scribble from the canvas and remove it from the scribble collection. Add a removeLast method to the ScribbleCollection class.
- Move mode and Color mode need to determine if a mouse click takes place on any of the scribbles in the collection, not just the most recently drawn scribble. The contains method of ScribbleCollection class should implement this function. Once you have determined which scribble contains the click point, you can move or color that scribble as you did in the simplified version of the program.

**Submitting your work** Before turning in your work, be sure to double check its logical organization and your style of presentation. Make your code as clear as possible and include appropriate comments describing major sections of code and declarations.

```
// A very simple drawing program.
public class Scribbler extends WindowController
                       implements ActionListener {

   // user modes
   // using ints rather than boolean to allow for extension to
   // other modes
   private static final int DRAWING = 1;
   private static final int MOVING = 2;
   private static final int COLORING = 3;


   // the current scribble
   private Scribble currentScribble;

   // the collection of scribbles
   // private ScribbleCollection scribbles;

   // stores last point for drawing or dragging
   private Location lastPoint;

   // whether the most recent scribble has been selected for moving
   private boolean draggingScribble;

   // buttons that allow user to select modes
   private JButton setDraw, setMove, setErase, setColor;

   // Choice JButton to select color
   private JComboBox chooseColor;

   // new color for scribble
   private Color newColor;

   // label indicating current mode
   private JLabel modeLabel;

   // the current mode -- drawing mode by default
   private int chosenAction = DRAWING;

   public Scribbler ( ) {
       startController(700,500);
    }
   // create and hook up the user interface components
   public void begin() {

      setDraw = new JButton("Draw");
      setMove = new JButton("Move");
      setColor = new JButton("Color");

      JPanel buttonPanel = new JPanel();
      buttonPanel.add(setDraw);
      buttonPanel.add(setMove);
      buttonPanel.add(setColor);

      chooseColor = new JComboBox();
      chooseColor.addItem("red");
      chooseColor.addItem("blue");
      chooseColor.addItem("green");
      chooseColor.addItem("yellow");

      setErase = new JButton("Erase last");
      JPanel choicePanel = new JPanel();
```

```
   choicePanel.add(setErase);
   choicePanel.add(chooseColor);

   JPanel controlPanel = new JPanel(new GridLayout(3,1));
   modeLabel = new JLabel("Current mode: drawing");
   controlPanel.add(modeLabel);
   controlPanel.add(buttonPanel);
   controlPanel.add(choicePanel);


   getContentPane().add(controlPanel, BorderLayout.SOUTH);

   // add listeners
   setDraw.addActionListener(this);
   setMove.addActionListener(this);
   setErase.addActionListener(this);
   setColor.addActionListener(this);

   // current scribble is empty
   currentScribble = null;

   validate();
}

// if in drawing mode then start with empty scribble
// if in moving mode then prepare to move
public void onMousePress(Location point) {
   if (chosenAction == DRAWING) {
          // start with an empty scribble for drawing
      currentScribble = new Scribble();

   } else if (chosenAction == MOVING) {
          // check if user clicked on current scribble
      draggingScribble = currentScribble.contains(point);
   }

   // remember point of press for drawing or moving
   lastPoint = point;
}

// if in drawing mode, add a new segment to scribble
// if in moving mode then move it
public void onMouseDrag(Location point) {
   if (chosenAction == DRAWING) {
      // add new line segment to current scribble
      Line newSegment = new Line(lastPoint, point, canvas);

      currentScribble.addLine(newSegment);

   } else if (chosenAction == MOVING) {
      if (draggingScribble) {// move current scribble
         currentScribble.move(point.getX() - lastPoint.getX(),
                              point.getY() - lastPoint.getY());
      }
   }
   // update for next move or draw
   lastPoint = point;
}

public void onMouseRelease(Location point) {
   // Add code when have collection of scribbles
}
```

```
   // Set mode according to JButton pressed.
   public void actionPerformed(ActionEvent e) {
      if (e.getSource() == setDraw) {
         chosenAction = DRAWING;
         modeLabel.setText("Current mode: drawing");
      } else if (e.getSource() == setMove) {
         chosenAction = MOVING;
         modeLabel.setText("Current mode: moving");
      }
   }

}
```

```
// this class holds an array of line segments
// that make up one scribble
public class Scribble {

    private static final int MAX_LINES=10000;   // size of array

    private int numberLines = 0;                // count of line segments in scribble
    private Line[] lines = new Line[MAX_LINES]; // array to hold line segments

    public void addLine(Line segment){
        if (numberLines==MAX_LINES){
            // too many lines
            System.out.println("ERROR: MAX LINES EXCEEDED");
            throw new RuntimeException("MAX LINES EXCEEDED");
        } else {
            lines[numberLines]=segment;
            numberLines++;
        }
    }

    public boolean contains(Location pt){
        for (int i=0; i<numberLines; i++){
            if (lines[i].contains(pt)){
                return true;
            }
        }
        return false;
    }

    public void move(double dx, double dy){
        for (int i=0; i<numberLines; i++){
            lines[i].move(dx, dy);
        }
    }

}
```

```
// this class contains an array of Scribbles
public class ScribbleCollection {

    // private int number of scribbles
    // private Scribble[] scribbles = new Scribble[MAX];

    /*
     * add a scribble to the array
     * if the array is full, create and throw a RuntimeException.
     */
    public void addScribble (Scribble s){
    }

    /*
     * return the scribble that contains the input location
     * return null if no scribble contains the location
     */
    public Scribble contains (Location pt){
        return null;
    }

    /*
     * remove the last scribble from the canvas
     * remove the scribble from the collection array
     * if there are no scribbles, then do nothing.
     */
    public void removeLast(){
    }
}
```