**[Paint-Brush](https://github.com/kholoud-mohamed22/Paint-Brush)**

**Overview**

This project is a Java Swing-based Drawing Tool that provides users with an interactive canvas to create various shapes, freehand drawings, and brush strokes. Users can also erase, undo/redo actions, and customize their drawings using features like color selection, line thickness, and dotted or solid shapes.

**Features**

**1. Drawing Modes**

Users can switch between the following drawing modes:

* **Line**: Draw straight lines.
* **Rectangle**: Draw rectangular shapes (filled or outlined).
* **Oval**: Draw oval shapes (filled or outlined).
* **Pencil**: Freehand drawing with customizable thickness.
* **Brush**: Freehand drawing with a larger stroke width.
* **Eraser**: Erase parts of the drawing.

**2. Customization Options**

* **Color Selection**: Use a color chooser dialog to pick any color for shapes and strokes.
* **Stroke Thickness**: Adjust the line thickness using a slider.
* **Fill/Outline Settings**: Draw filled shapes, outlined shapes, or dotted outlines.

**3. Undo/Redo**

* **Undo**: Remove the last drawn shape or stroke.
* **Redo**: Reapply the last undone action.

**4. Canvas Management**

* **Clear Canvas**: Reset the canvas by removing all shapes and strokes.

**Components**

**1. DrawingPanel**

The DrawingPanel class handles all drawing operations and user interactions on the canvas.

**Key Features:**

* Stores shapes in a shapes list and undone shapes in a redoStack list.
* Manages the current drawing mode, color, stroke style, and fill settings.
* Provides the following functionalities:
  + **Mouse Interaction**: Listens to mouse events to draw shapes or freehand strokes.
  + **Undo/Redo**: Manages the lists of shapes to implement undo/redo functionality.
  + **Shape Drawing**: Draws shapes based on the current mode and user input.

**Methods:**

* setCurrentMode(String mode): Sets the drawing mode (e.g., "Pencil", "Line").
* setCurrentColor(Color color): Sets the drawing color.
* setFillShape(boolean fill): Enables or disables shape filling.
* setCurrentStroke(Stroke stroke): Sets the current stroke style.
* undo(): Removes the last shape from the canvas.
* redo(): Reapplies the last undone shape.
* clear(): Clears all shapes from the canvas.

**Mouse Event Handlers:**

* **mousePressed**: Initializes the current shape or freehand stroke.
* **mouseDragged**: Updates the current shape or stroke as the mouse moves.
* **mouseReleased**: Finalizes the current shape or stroke.

**2. ControlPanel**

The ControlPanel class provides a graphical interface for users to select drawing modes, customize settings, and manage the canvas.

**Key Features:**

* Contains buttons for drawing modes, undo/redo, and clearing the canvas.
* Provides a color picker for selecting drawing colors.
* Includes a slider for adjusting stroke thickness.
* Allows users to toggle between solid, outlined, or dotted shapes.

**Components:**

1. **Drawing Mode Buttons**:
   * Line
   * Rectangle
   * Oval
   * Pencil
   * Brush
   * Eraser
2. **Action Buttons**:
   * Undo
   * Redo
   * Clear Canvas
3. **Color Picker**:
   * Button to open a JColorChooser dialog for color selection.
4. **Stroke Thickness**:
   * A slider for adjusting line/stroke thickness.
5. **Shape Style**:
   * Radio buttons for Solid, Outlined, or Dotted shapes.

**Button Configuration:**

Each button is linked to an action listener that updates the DrawingPanel accordingly. For example:

* lineButton.addActionListener(e -> drawingPanel.setCurrentMode("Line"));

**Shapes and Tools**

**Shape Hierarchy**

Each shape in the tool extends a base Shape class (not included in the provided code). The hierarchy may look like this:

* **Shape** (abstract class/interface): Common properties and methods for all shapes.
  + **Line**: Represents a straight line.
  + **Rectangle**: Represents a rectangle (filled or outlined).
  + **Oval**: Represents an oval (filled or outlined).
  + **Pencil**: Represents freehand drawing with points stored as a list.
  + **Brush**: Similar to Pencil but with thicker strokes.
  + **Eraser**: Acts like a white-colored pencil for erasing parts of the canvas.

**Usage Instructions**

**Running the Application**

1. Compile and run the program in a Java IDE (e.g., NetBeans, IntelliJ IDEA).
2. The application will launch with the drawing canvas and control panel.

**Drawing on the Canvas**

1. Select a drawing mode (e.g., Pencil, Rectangle, Line).
2. (Optional) Customize the color, stroke thickness, and shape style.
3. Click and drag on the canvas to draw shapes or strokes.
4. Release the mouse button to finalize the shape or stroke.

**Managing the Canvas**

* **Undo/Redo**: Use the corresponding buttons to revert or reapply changes.
* **Clear Canvas**: Click "Clear" to erase all drawings.
* **Color Selection**: Choose a color using the "Choose Color" button.