

SimplePaint

Query by Example Editor for the VizIR Project

by

Dalibor Mitrovic

mitrovic@ims.tuwien.ac.at

User Guide

Introduction

SimplePaint is a small painting application that can be used to produce multi-layered sketches. The figure below shows the main window of SimplePaint:

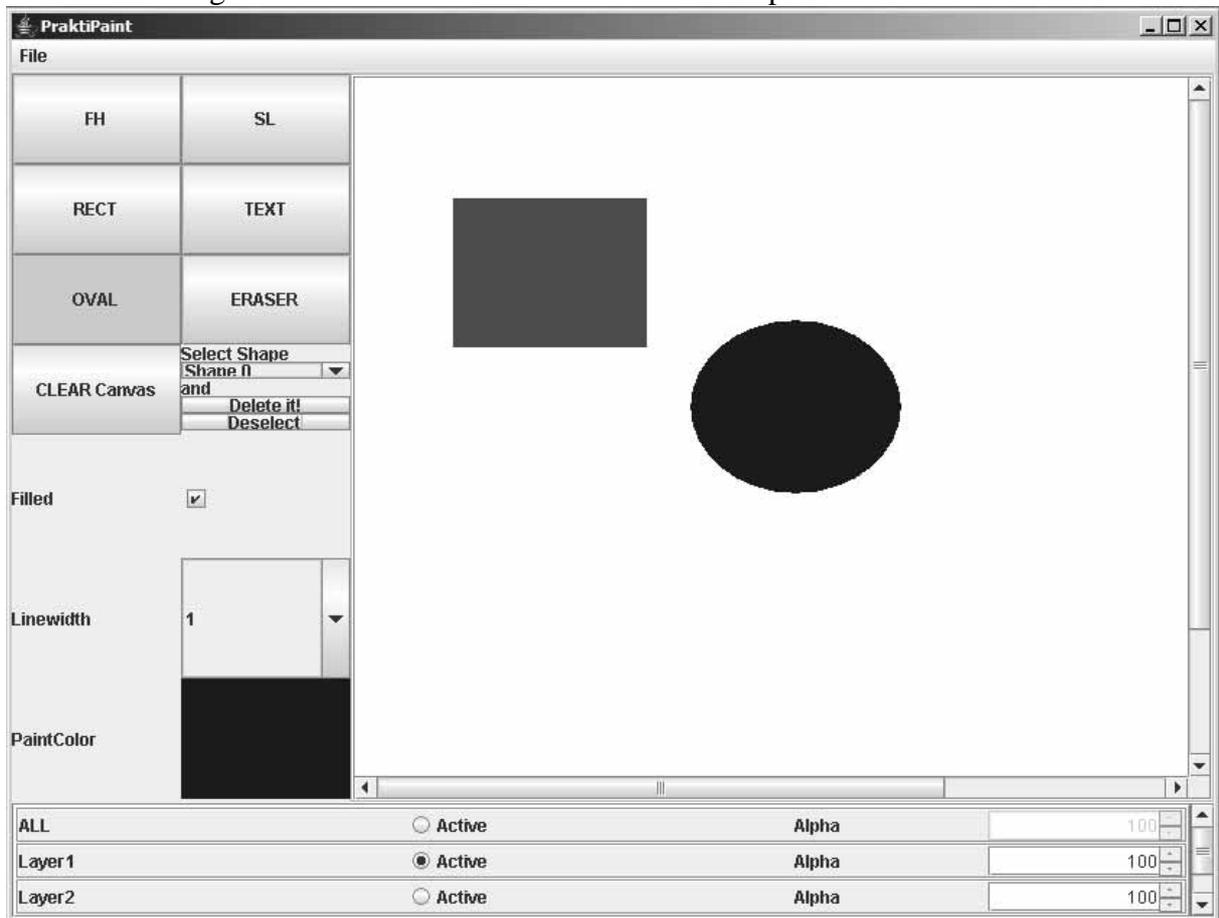


Figure 1 SimplePaint Main Application Window

Using SimplePaint:

SimplePaint extends the `JPanel` class, therefore it can be used as any other `JPanel` would be used. To add a SimplePaint panel to your application you would use this code:

```
SimplePaint simplePaintPanel = new SimplePaint();
JFrame mainFrame = new JFrame("SimplePaint");
mainFrame.setContentPane(simplePaintPanel);
```

Getting the image data:

SimplePaint has two methods to access the image data. The method `getImage()` returns a `BufferedImage` object containing the raster data of the currently active canvas. The method `getAllImage()` returns a `BufferedImage` object containing the raster data of the all canvas. These two methods can be called anytime.

Creating (= drawing) a shape:

First the layer is selected where the shape is to be drawn. Then the type of the shape is selected by clicking the appropriate button e.g. FH for freehand line. Now the width of the stroke and the color are selected. At last the user needs to press a mouse button while over the

canvas and drag the mouse while the button is pressed to paint. This procedure works for all shapes.

Deleting a shape:

The user has to select a shape and delete it by pressing the button labeled “delete it!”.

Clearing the active canvas:

Pushing the button labeled “clear canvas” the currently active canvas is cleared.

Starting SimplePaint as an application (used mostly for testing):

SimplePaint is started by typing `java org.vizir.sketching.SimplePaint` in the console if the folder is the root of the hierarchy `org/vizir/sketching/`.

LayerManager

The LayerManager panel is used to select the active canvas (= layer) and to control the alpha values. An alpha of zero means that the layer is completely transparent, an alpha value of 100 means that the layer is opaque. The canvas with the label “all” displays a combination of all layers rendered with their respective alpha, this layer is not editable.



Figure 2 SimplePaint LayerManager

ToolBox

The ToolBox offers a set of basic sketching tools. It empowers the user to draw straight lines, freehand lines, rectangles, ovals and text. The eraser tool behaves similarly to the freehand line tool except that it automatically uses the canvas’ background color. Clicking the button “clear canvas” clears the currently active canvas. Note that this button is disabled when the active canvas is the “all” canvas. Deleting a certain shape is done by selecting the shape and pressing the “delete it!” button. Deselecting is performed by clicking on the “deselect” button. The checkbox “filled” and the combobox “line width” behave as expected. By selecting “filled” the next shape to be drawn is filled. “Line width” controls the width of the stroke. Clicking on the colored area labeled “PaintColor” opens a new window that contains the standard color chooser dialog. Note the issues with this color chooser window.

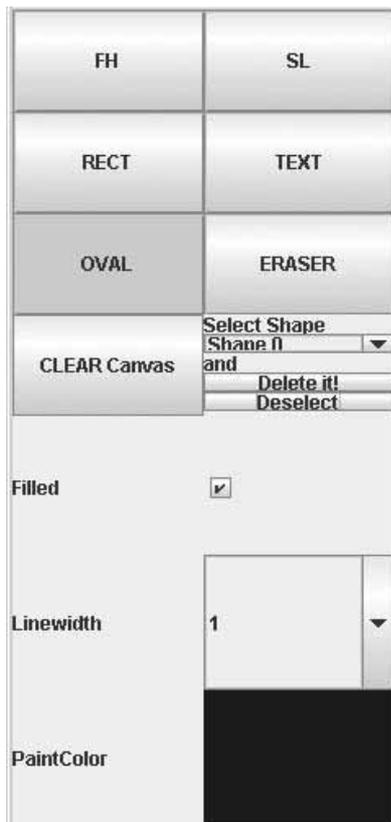


Figure 3 SimplePaint ToolBox

Text is created by selecting “text” as the current shape. A new window is opened where the user can enter the font, the font size, the font style and the text. When finished the user has to click the “done” button, the window is closed, now the user can simply place the text on the canvas by clicking.



Figure 4 Text Creation Window

Known Issues

In this section known issues are listed and explained. Although the application was tested extensively there is the possibility that some bugs were not found. Therefore the following list may be incomplete:

- no possibility of filling a region with a specified color**
 It is not possible to fill a contiguous region with a certain color. This is due to the object oriented character of the painting. A region that should be filled would have to be separate iMyShape object.
- no point and click selection**
 It is not possible the select a shape by clicking on it. Selection has to be performed via the selection combobox.

- **strange behavior of color chooser window**

The color chooser window loses focus the moment the mouse leaves it. Furthermore if the application's main window is maximized the color chooser window is hidden the moment it is entered by the mouse cursor. One has to reduce the application's main window in order to access the color chooser window.

- **shape deletion of unselected shapes possible**

It is possible to delete a shape that is not marked as selected. The shape deletion routine does not check whether a shape is selected or not prior to deleting it.

- **selection visible in `BufferedImage`**

If a shape is selected and the `SimplePaint`'s methods `getImage()` or `getAllImage()` are called the returned `BufferedImage` object shows the selected shape rendered in its highlighted mode.