画板 实验报告

1. 类简述

有Myshape、Pen、Palette、DMouseListener、drawListener以及主类Main

Myshape用于存储需要画出的图像的特征，比如形状、颜色、笔触

Pen继承了Myshape类，用于画笔模式，需要将所画图形存储成多段直线，同样用于存储需要画出的图形

Palette继承了JFramelei用于生成画板界面，例如背景色、选项框等并且调用各种绘画模式

DMouseListener、drawListener都是Palette的内部类，用于监听鼠标的动作，确定鼠标行为模式所对应的绘画类型，并且进行绘图

1. 代码分析

MyShape类：

用于存储图形的特征信息

包括构造函数、默认构造函数、以及将图像画出来的draw()方法

含有shape（形状）color（颜色）stroke（笔触）三个变量

Pen类：

继承了MyShape类，用于画笔模式，可以将所画图像看作是许多小线段构成的，将这些线段存储在一个ArrayList中，含有一个将图像画出的方法

Palette类：

继承了JFrame类来生成窗体以及图形框架

其中含有一个ArrayList<MyShape>存储需要绘制的图像

一个JPanel类的对象window作为主要窗体

两个javax.swing.JComboBox<String>类的选项框分别用于选择画笔和笔触

该类含有两个内部类：

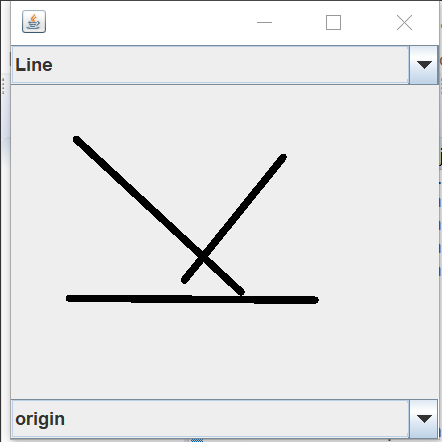
* DMouseListener

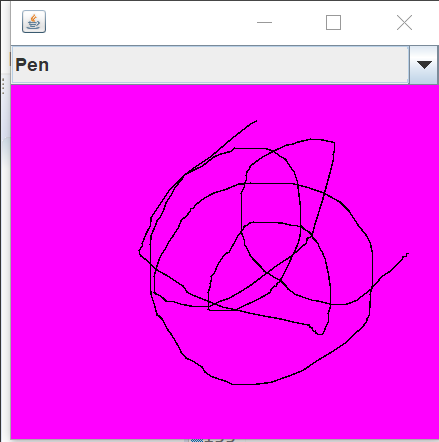
用于监听跟踪鼠标移动的动作，获取每一次鼠标在屏幕上的坐标，在鼠标移动的过程中往数组里添加新的线段，并且重新画出。

* drawListener

用于监听鼠标点击释放的动作，在点击位置生成形状，然后在释放位置终结

1. 结果截图





1. 源码

package sort;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.HeadlessException;

import java.awt.Shape;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.awt.event.MouseMotionListener;

import java.awt.geom.Line2D;

import java.util.ArrayList;

import javax.swing.JComboBox;

import javax.swing.JFrame;

import javax.swing.JPanel;

class MyShape {

Shape s;

Color c;

public MyShape(Shape s, Color c) {

super();

this.s = s;

this.c = c;

}

public MyShape() {

super();

}

void draw(Graphics2D g)

{

g.setColor(c);

g.draw(s);

}

}

class penshape extends MyShape{

ArrayList<Shape> ll=new ArrayList<Shape>();

Line2D.Float cl;

void draw(Graphics2D g)

{

g.setColor(c);

for(Shape ms:ll)

{

g.draw(ms);;

}

if(cl!=null)

{

g.draw(cl);

}

}

}

public class DrawPanel extends JFrame{

ArrayList<MyShape> sl=new ArrayList<MyShape>();

JPanel draw=new JPanel(){

@Override

public void paint(Graphics arg0) {

// TODO Auto-generated method stub

super.paint(arg0);

Graphics2D g=(Graphics2D )arg0;

for(MyShape ms:sl)

{

ms.draw(g);

}

if(tempShape!=null)

{

tempShape.draw(g);

}

}

};

javax.swing.JComboBox<String> shapetype=new JComboBox<String>();

MyShape tempShape;

Color currc=Color.black;

class drawMotionListener implements MouseMotionListener

{

@Override

public void mouseDragged(MouseEvent arg0) {

// TODO Auto-generated method stub

DrawPanel dp=DrawPanel.this;

switch (dp.shapetype.getSelectedIndex()){

case 0:

{

Line2D.Float cl=(Line2D.Float)tempShape.s;

cl.x2=arg0.getX();

cl.y2=arg0.getY();

break;

}

case 1:

{

penshape ps=(penshape)tempShape;

ps.cl.x2=arg0.getX();

ps.cl.y2=arg0.getY();

ps.ll.add(ps.cl);

Line2D.Float cl=new Line2D.Float();

cl.x1=arg0.getX();

cl.y1=arg0.getY();

cl.x2=arg0.getX();

cl.y2=arg0.getY();

ps.cl=cl;

break;

}

}

JPanel j=(JPanel)(arg0.getSource());

j.repaint();

}

@Override

public void mouseMoved(MouseEvent arg0) {

// TODO Auto-generated method stub

}

}

class drawListener implements MouseListener

{

@Override

public void mouseClicked(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mouseEntered(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mouseExited(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mousePressed(MouseEvent arg0) {

// TODO Auto-generated method stub

DrawPanel dp=DrawPanel.this;

switch (dp.shapetype.getSelectedIndex()){

case 0:

{

Line2D.Float cl=new Line2D.Float();

cl.x1=arg0.getX();

cl.y1=arg0.getY();

cl.x2=arg0.getX();

cl.y2=arg0.getY();

tempShape=new MyShape(cl,currc);

break;

}

case 1:

{

penshape ps=new penshape();

ps.c=currc;

Line2D.Float cl=new Line2D.Float();

cl.x1=arg0.getX();

cl.y1=arg0.getY();

cl.x2=arg0.getX();

cl.y2=arg0.getY();

ps.cl=cl;

tempShape=ps;

break;

}

}

}

@Override

public void mouseReleased(MouseEvent arg0) {

// TODO Auto-generated method stub

DrawPanel dp=DrawPanel.this;

switch (dp.shapetype.getSelectedIndex()){

case 0:

{

Line2D.Float cl=(Line2D.Float)tempShape.s;

cl.x2=arg0.getX();

cl.y2=arg0.getY();

break;

}

case 1:

{

penshape ps=(penshape)tempShape;

ps.cl.x2=arg0.getX();

ps.cl.y2=arg0.getY();

ps.ll.add(ps.cl);

Line2D.Float cl=new Line2D.Float();

cl.x1=arg0.getX();

cl.y1=arg0.getY();

cl.x2=arg0.getX();

cl.y2=arg0.getY();

ps.cl=cl;

break;

}

}

dp.sl.add(tempShape);

tempShape=null;

JPanel j=(JPanel)(arg0.getSource());

j.repaint();

}

}

public DrawPanel() throws HeadlessException {

super();

shapetype.addItem("Line");

shapetype.addItem("Pen");

this.getContentPane().add(BorderLayout.NORTH,shapetype);

this.getContentPane().add(draw);

draw.addMouseListener(new drawListener());

draw.addMouseMotionListener(new drawMotionListener());

draw.setBackground(Color.magenta);

this.setSize(300,300);

this.setDefaultCloseOperation(3);

this.setVisible(true);

}

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

DrawPanel dp=new DrawPanel();

}

}