实验三 绘图板APP开发

**——陈洁怡+201906061703**

一、基础实验

（一）实验要求

开发一个绘图板APP，支持线形选择、颜色选择、触点压力感应，支持重做等基本操作。界面要求美观整洁。提交设计APP设计文档及代码。

（二）基本知识与原理

1. 实验内容及步骤

1、更改创建工程自带的activi\_main.xml，作为应用的主界面前端

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"  tools:context=".MainActivity">   <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">   <LinearLayout  android:id="@+id/btnLv\_1"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_alignParentBottom="true"  android:orientation="horizontal">   <Button  android:id="@+id/btn\_shape"  android:layout\_width="0dp"  android:layout\_weight="1"  android:layout\_height="wrap\_content"  android:layout\_margin="5dp"  android:background="@drawable/selector\_btn"  android:text="线形选择"  android:textColor="@color/black"  android:textSize="16sp" />   <Button  android:id="@+id/btn\_color"  android:layout\_width="0dp"  android:layout\_weight="1"  android:layout\_height="wrap\_content"  android:layout\_margin="5dp"  android:background="@drawable/selector\_btn"  android:text="颜色选择"  android:textColor="@color/black"  android:textSize="16sp" />   <Button  android:id="@+id/btn\_redo"  android:layout\_width="0dp"  android:layout\_weight="1"  android:layout\_height="wrap\_content"  android:layout\_margin="5dp"  android:background="@drawable/selector\_btn"  android:text="重做"  android:textColor="@color/black"  android:textSize="16sp" />   </LinearLayout>   <LinearLayout  android:id="@+id/textLv\_1"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  android:layout\_above="@id/btnLv\_1">  <LinearLayout  android:layout\_width="0dp"  android:layout\_weight="1"  android:layout\_height="wrap\_content">  <TextView  android:id="@+id/nowShapeTv\_1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="16sp"  android:textColor="@color/black"  android:maxLines="1"  android:text="当前线形基础半径：5"/>  </LinearLayout>  <LinearLayout  android:layout\_width="0dp"  android:layout\_weight="1"  android:layout\_height="wrap\_content">  <TextView  android:id="@+id/nowColorTv\_1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="16sp"  android:textColor="@color/black"  android:maxLines="1"  android:text="当前颜色：#000000"/>  </LinearLayout>   </LinearLayout>   <LinearLayout  android:id="@+id/textLv\_2"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  android:layout\_above="@id/textLv\_1">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content">  <TextView  android:id="@+id/nowPressureTv\_1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="16sp"  android:textColor="@color/black"  android:maxLines="1"  android:text="当前压力 ：0"/>  </LinearLayout>   </LinearLayout>   <LinearLayout  android:id="@+id/paintLL\_1"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_above="@id/textLv\_2"  android:orientation="horizontal">   <ImageView  android:id="@+id/iv\_1"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"/>   </LinearLayout>   </RelativeLayout>   </LinearLayout> |

2、res->layout中新建color\_dailog.xml作为自定义颜色选择dailog弹窗的样式

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:padding="10dp">   <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">   <LinearLayout  android:id="@+id/shapeDiaLv\_1"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">   <Button  android:id="@+id/btn\_black"  android:layout\_width="30dp"  android:layout\_height="30dp"  android:layout\_margin="5dp"  android:background="@color/black" />   <Button  android:id="@+id/btn\_red"  android:layout\_width="30dp"  android:layout\_height="30dp"  android:layout\_margin="5dp"  android:background="@color/red" />   <Button  android:id="@+id/btn\_yellow"  android:layout\_width="30dp"  android:layout\_height="30dp"  android:layout\_margin="5dp"  android:background="@color/light\_yellow" />   <Button  android:id="@+id/btn\_blue"  android:layout\_width="30dp"  android:layout\_height="30dp"  android:layout\_margin="5dp"  android:background="@color/blue" />   <Button  android:id="@+id/btn\_green"  android:layout\_width="30dp"  android:layout\_height="30dp"  android:layout\_margin="5dp"  android:background="@color/green" />   <Button  android:id="@+id/btn\_orange"  android:layout\_width="30dp"  android:layout\_height="30dp"  android:layout\_margin="5dp"  android:background="@color/orange" />   <Button  android:id="@+id/btn\_grey"  android:layout\_width="30dp"  android:layout\_height="30dp"  android:layout\_margin="5dp"  android:background="@color/dark\_grey" />   </LinearLayout>   <RelativeLayout  android:layout\_marginTop="10dp"  android:id="@+id/colorDiaRL\_1"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_below="@id/shapeDiaLv\_1">   <TextView  android:id="@+id/colorDiaTv\_1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignParentStart="true"  android:text="自定义颜色：#"  android:textColor="@color/black"  android:textSize="20sp" />   <EditText  android:id="@+id/colorDiaEv\_1"  android:layout\_width="120dp"  android:layout\_height="wrap\_content"  android:layout\_toRightOf="@id/colorDiaTv\_1"  android:maxLines="1"  android:textColor="@color/black"  android:textSize="14sp" />  </RelativeLayout>   </RelativeLayout> </LinearLayout> |

3、res->layout中新建shape\_dailog.xml作为自定义线性选择dailog弹窗的样式

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"  android:padding="10dp">   <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">   <LinearLayout  android:id="@+id/shapeDiaLv\_1"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">   <Button  android:id="@+id/btn\_1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_margin="5dp"  android:background="@color/medium\_yellow"  android:text="粗"  android:textColor="@color/black"  android:textSize="16sp" />   <Button  android:id="@+id/btn\_2"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_margin="5dp"  android:background="@color/medium\_yellow"  android:text="适中"  android:textColor="@color/black"  android:textSize="16sp" />   <Button  android:id="@+id/btn\_3"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_margin="5dp"  android:background="@color/medium\_yellow"  android:text="细"  android:textColor="@color/black"  android:textSize="16sp" />  </LinearLayout>   <RelativeLayout  android:layout\_marginTop="10dp"  android:id="@+id/colorDiaRL\_1"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_below="@id/shapeDiaLv\_1">   <TextView  android:id="@+id/colorDiaTv\_1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignParentStart="true"  android:text="自定义线形基础半径："  android:textColor="@color/black"  android:textSize="20sp" />   <EditText  android:id="@+id/shapeDiaEv\_1"  android:layout\_width="50dp"  android:layout\_height="wrap\_content"  android:layout\_toRightOf="@id/colorDiaTv\_1"  android:inputType="number"  android:maxLines="1"  android:textColor="@color/black"  android:textSize="14sp" />  </RelativeLayout>   </RelativeLayout>   </LinearLayout> |

1. 在MainActivity.java中实现基本业务逻辑，监听ontouch事件，对于按照触点压力作线的操作。大致思路是根据选择或默认的基础半径，在基础半径50%-150%的粗细范围内随压力的变化而变化；同时根据当前压力和前一次压力比较，如果当前压力比前一次大则增加一个度量的半径，当前压力比前一次小则减少一个度量的半径；每次变化的度量为基础半径的3%：

默认画笔半径为5，画笔颜色为黑色，第一次接触前的压力为0

撤销的大致思路是vector套vector自定义segment类记录所有的笔触然后清空画布按顺序去画

Segment.java

|  |
| --- |
| package com.example.androidtest3;  public class Segment {  float segStartX;  float segStartY;  float segEndX;  float segEndY;  String segColor;  float segRadio;   public Segment() {  this.segStartX = 0;  this.segStartY = 0;  this.segEndX = 0;  this.segEndY = 0;  this.segColor = "#000000";  this.segRadio = 5;  }   public Segment(float segStartX, float segStartY, float segEndX, float segEndY) {  this.segStartX = segStartX;  this.segStartY = segStartY;  this.segEndX = segEndX;  this.segEndY = segEndY;  this.segColor = "#000000";  this.segRadio = 5;  }   public Segment(float segStartX, float segStartY, float segEndX, float segEndY, String segColor, float segRadio) {  this.segStartX = segStartX;  this.segStartY = segStartY;  this.segEndX = segEndX;  this.segEndY = segEndY;  this.segColor = segColor;  this.segRadio = segRadio;  }   public float getSegStartX() {  return segStartX;  }   public void setSegStartX(float segStartX) {  this.segStartX = segStartX;  }   public float getSegStartY() {  return segStartY;  }   public void setSegStartY(float segStartY) {  this.segStartY = segStartY;  }   public float getSegEndX() {  return segEndX;  }   public void setSegEndX(float segEndX) {  this.segEndX = segEndX;  }   public float getSegEndY() {  return segEndY;  }   public void setSegEndY(float segEndY) {  this.segEndY = segEndY;  }   public String getSegColor() {  return segColor;  }   public void setSegColor(String segColor) {  this.segColor = segColor;  }   public float getSegRadio() {  return segRadio;  }   public void setSegRadio(float segRadio) {  this.segRadio = segRadio;  } } |

MainActivity.java

|  |
| --- |
| package com.example.androidtest3;  import androidx.appcompat.app.AppCompatActivity;  import android.app.Activity; import android.app.AlertDialog; import android.content.DialogInterface; import android.graphics.Bitmap; import android.graphics.Canvas; import android.graphics.Color; import android.graphics.Paint; import android.os.Bundle; import android.util.Log; import android.view.LayoutInflater; import android.view.MotionEvent; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.ImageView; import android.widget.TextView; import android.widget.ThemedSpinnerAdapter; import android.widget.Toast;  import java.util.Vector;  public class MainActivity extends AppCompatActivity {   private Button btnShape, btnColor, btnUndo, btnRedo;   private Bitmap baseBitmap;  private ImageView iv;  private Canvas canvas;  private Paint paint;  private float baseRadio = 5;  private String baseColor = "#000000";  private float radio;  float prePressure = 0;   private Vector<Vector<Segment>> vec;  private int cnt = 0;   private TextView nowColorTv\_1, nowShapeTv\_1, nowPressureTv\_1;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);   btnShape = findViewById(R.id.btn\_shape);  btnColor = findViewById(R.id.btn\_color);  btnUndo = findViewById(R.id.btn\_undo);  btnRedo = findViewById(R.id.btn\_redo);   OnClick onClick = new OnClick();  btnShape.setOnClickListener(onClick);  btnColor.setOnClickListener(onClick);  btnUndo.setOnClickListener(onClick);  btnRedo.setOnClickListener(onClick);   nowColorTv\_1 = findViewById(R.id.nowColorTv\_1);  nowShapeTv\_1 = findViewById(R.id.nowShapeTv\_1);  nowPressureTv\_1 = findViewById(R.id.nowPressureTv\_1);   iv = findViewById(R.id.iv\_1);  paint = new Paint();  radio = baseRadio;  paint.setStrokeWidth(radio);  iv.setOnTouchListener(touch);    vec = new Vector<>();  }   private View.OnTouchListener touch = new View.OnTouchListener() {  float startX;  float startY;  float mxRadio = baseRadio \* (float) 1.5;  float mnRadio = baseRadio \* (float) 0.5;  float changeRadio = baseRadio \* (float) 0.03;   @Override  public boolean onTouch(View v, MotionEvent event) {   mxRadio = baseRadio \* (float) 1.5;  mnRadio = baseRadio \* (float) 0.5;  changeRadio = baseRadio \* (float) 0.03;   paint.setColor(Color.parseColor(baseColor));  nowPressureTv\_1.setText("当前压力 ：" + prePressure);   switch (event.getAction()) {  case MotionEvent.ACTION\_DOWN:  paint.setStrokeWidth(radio);  if (baseBitmap == null) {  baseBitmap = Bitmap.createBitmap(iv.getWidth(), iv.getHeight(),  Bitmap.Config.ARGB\_8888);  canvas = new Canvas(baseBitmap);  canvas.drawColor(Color.WHITE);  }  startX = event.getX();  startY = event.getY();  Vector<Segment> tmp = new Vector<>();  vec.add(tmp);  break;  case MotionEvent.ACTION\_MOVE:  float stopX = event.getX();  float stopY = event.getY();   Thread t = new Thread(new Runnable() {  @Override  public void run() {  float nowPressure = event.getPressure();  if (nowPressure > prePressure) radio += changeRadio;  else if (nowPressure < prePressure) radio -= changeRadio;  prePressure = nowPressure;  // nowPressureTv\_1.setText("当前压力 ：" + nowPressure);   radio = Math.min(radio, mxRadio);  radio = Math.max(radio, mnRadio);   try {  Thread.sleep(500);  } catch (InterruptedException e) {  e.printStackTrace();  }  }  });  t.start();   paint.setStrokeWidth(radio);  canvas.drawLine(startX, startY, stopX, stopY, paint);   Segment tmpSeg = new Segment(startX, startY, stopX, stopY, baseColor, baseRadio); // vec[cnt].add(tmpSeg);   Vector tmpVec = vec.get(cnt);  tmpVec.add(tmpSeg);   startX = event.getX();  startY = event.getY();  iv.setImageBitmap(baseBitmap);  break;  case MotionEvent.ACTION\_UP:  radio = baseRadio;  cnt++;  break;  default:  break;  }   return true;  }  };   protected void resumeCanvas() {  if (baseBitmap != null) {  baseBitmap = Bitmap.createBitmap(iv.getWidth(),  iv.getHeight(), Bitmap.Config.ARGB\_8888);  canvas = new Canvas(baseBitmap);  canvas.drawColor(Color.WHITE);  iv.setImageBitmap(baseBitmap); // Toast.makeText(MainActivity.this, "清除画板成功", Toast.LENGTH\_LONG).show();  }  }   protected void paintCanvas() {  resumeCanvas();  String s = "in paint canvas";  Log.d(s, "in paint canvas");  for (int i = 0; i < cnt; ++i) {  Vector<Segment> tmpVec = vec.get(i);  int siz = tmpVec.size();  for (int j = 0; j < siz; ++j) {  Segment tmpSeg = tmpVec.get(j);  paint.setStrokeWidth(tmpSeg.segRadio);  paint.setColor(Color.parseColor(tmpSeg.segColor));  canvas.drawLine(tmpSeg.segStartX, tmpSeg.segStartY, tmpSeg.segEndX, tmpSeg.segEndY, paint);  }  }  iv.setImageBitmap(baseBitmap);  }   class OnClick implements View.OnClickListener {   @Override  public void onClick(View v) {  switch (v.getId()) {  case R.id.btn\_shape:  AlertDialog.Builder builderShape = new AlertDialog.Builder(MainActivity.this);  View viewShape = LayoutInflater.from(MainActivity.this).inflate(R.layout.shape\_dialog, null);  Button btnMx = viewShape.findViewById(R.id.btn\_1);  Button btnMed = viewShape.findViewById(R.id.btn\_2);  Button btnMn = viewShape.findViewById(R.id.btn\_3);  EditText etShape = viewShape.findViewById(R.id.shapeDiaEv\_1);  AlertDialog dailogShape = builderShape.setTitle("请选择线形").setView(viewShape).setPositiveButton("确定", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface dialogInterface, int i) {  if (etShape.getText().equals("") || etShape.getText().length() == 0)  return;  String tmpShape = "" + etShape.getText();  float tmpRadio = Float.parseFloat(tmpShape);  baseRadio = tmpRadio;  radio = baseRadio;  nowShapeTv\_1.setText("当前线形基础半径：" + baseRadio);  Toast.makeText(MainActivity.this, "确定选择 : " + etShape.getText(), Toast.LENGTH\_LONG).show();  }  }).setNegativeButton("取消", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface dialogInterface, int i) {  Toast.makeText(MainActivity.this, "取消选择", Toast.LENGTH\_LONG).show();  }  }).setCancelable(false).create();  btnMx.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  baseRadio = 15;  radio = baseRadio;  nowShapeTv\_1.setText("当前线形基础半径：15");  dailogShape.dismiss();  }  });  btnMed.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  baseRadio = 10;  radio = baseRadio;  nowShapeTv\_1.setText("当前线形基础半径：10");  dailogShape.dismiss();  }  });  btnMn.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  baseRadio = 5;  radio = baseRadio;  nowShapeTv\_1.setText("当前线形基础半径：5");  dailogShape.dismiss();  }  });  dailogShape.show();  break;  case R.id.btn\_color:  AlertDialog.Builder builderColor = new AlertDialog.Builder(MainActivity.this);  View viewColor = LayoutInflater.from(MainActivity.this).inflate(R.layout.color\_dialog, null);  Button btn\_black = viewColor.findViewById(R.id.btn\_black);  Button btn\_red = viewColor.findViewById(R.id.btn\_red);  Button btn\_yellow = viewColor.findViewById(R.id.btn\_yellow);  Button btn\_blue = viewColor.findViewById(R.id.btn\_blue);  Button btn\_green = viewColor.findViewById(R.id.btn\_green);  Button btn\_orange = viewColor.findViewById(R.id.btn\_orange);  Button btn\_grey = viewColor.findViewById(R.id.btn\_grey);   EditText etColor = viewColor.findViewById(R.id.colorDiaEv\_1);  AlertDialog dialogColor = builderColor.setTitle("请选择颜色").setView(viewColor).setPositiveButton("确定", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface dialogInterface, int i) {  if (etColor.getText().equals("") || etColor.getText().length() == 0)  return;  String tmpColor = "#" + etColor.getText();  baseColor = tmpColor;  nowColorTv\_1.setText("当前颜色：" + tmpColor);  Toast.makeText(MainActivity.this, "确定选择 : #" + etColor.getText(), Toast.LENGTH\_LONG).show();  }  }).setNegativeButton("取消", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface dialogInterface, int i) {  Toast.makeText(MainActivity.this, "取消选择", Toast.LENGTH\_LONG).show();  }  }).create();  btn\_black.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  baseColor = "#000000";  nowColorTv\_1.setText("当前颜色：#000000");  dialogColor.dismiss();  }  });  btn\_red.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  baseColor = "#880d1e";  nowColorTv\_1.setText("当前颜色：#880d1e");  dialogColor.dismiss();  }  });  btn\_yellow.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  baseColor = "#fcbf49";  nowColorTv\_1.setText("当前颜色：#fcbf49");  dialogColor.dismiss();  }  });  btn\_blue.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  baseColor = "#006ba6";  nowColorTv\_1.setText("当前颜色：#006ba6");  dialogColor.dismiss();  }  });  btn\_green.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  baseColor = "#134611";  nowColorTv\_1.setText("当前颜色：#134611");  dialogColor.dismiss();  }  });  btn\_orange.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  baseColor = "#ff9505";  nowColorTv\_1.setText("当前颜色：#ff9505");  dialogColor.dismiss();  }  });  btn\_grey.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  baseColor = "#454955";  nowColorTv\_1.setText("当前颜色：#454955");  dialogColor.dismiss();  }  });  dialogColor.show();  break;  case R.id.btn\_undo: // vec.remove(cnt);  vec.remove(cnt - 1);  cnt--;  paintCanvas();  break;  case R.id.btn\_redo:  resumeCanvas();  vec.clear();  cnt = 0;  prePressure = 0;  nowPressureTv\_1.setText("当前压力 ：" + prePressure);  break;  }  }  } } |

（四）运行结果截图

1、主页面布局情况

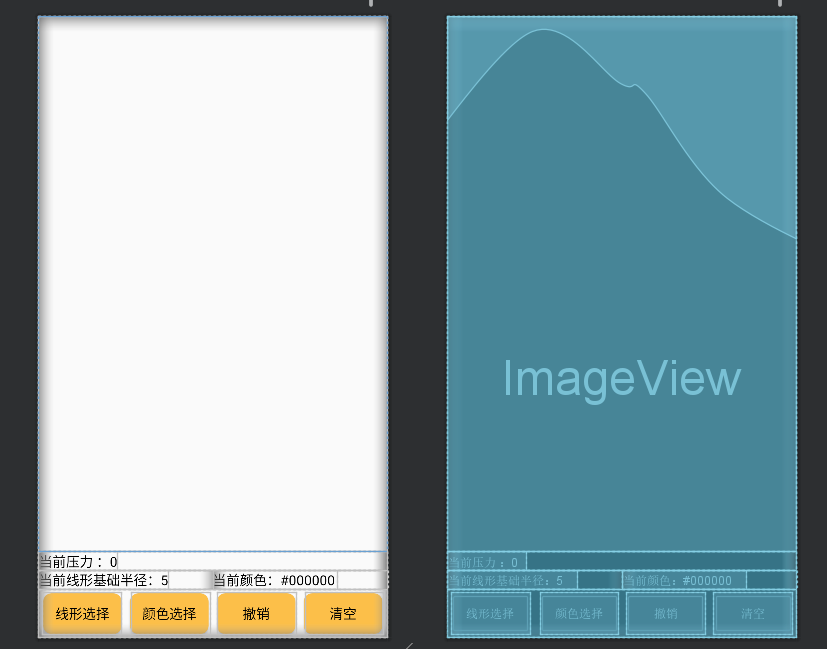


图4-1 主页面布局情况

2、线形选择和颜色选择，自定义dialog布局情况



图4-2-1 线形选择dialog布局

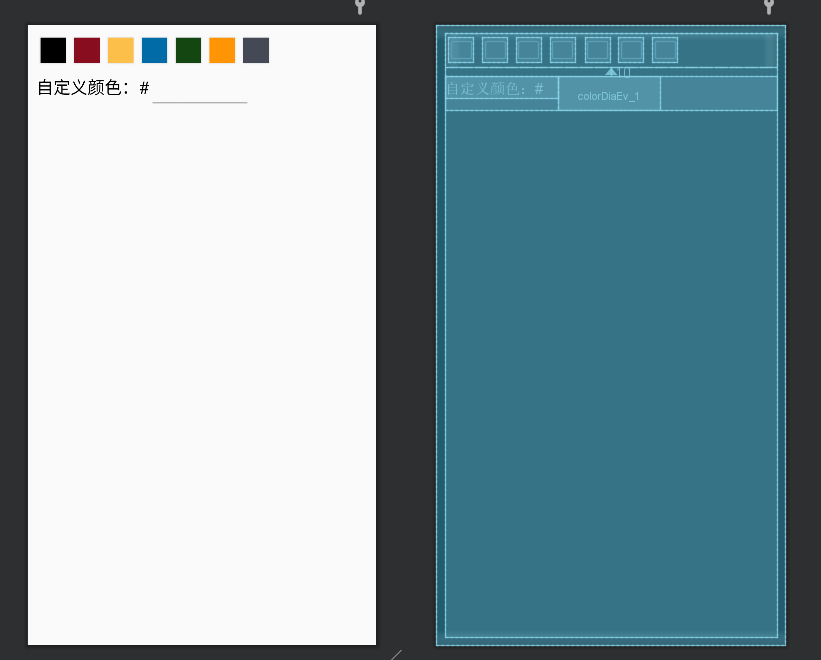


图4-2-2 颜色选择dailog布局

3、基本效果

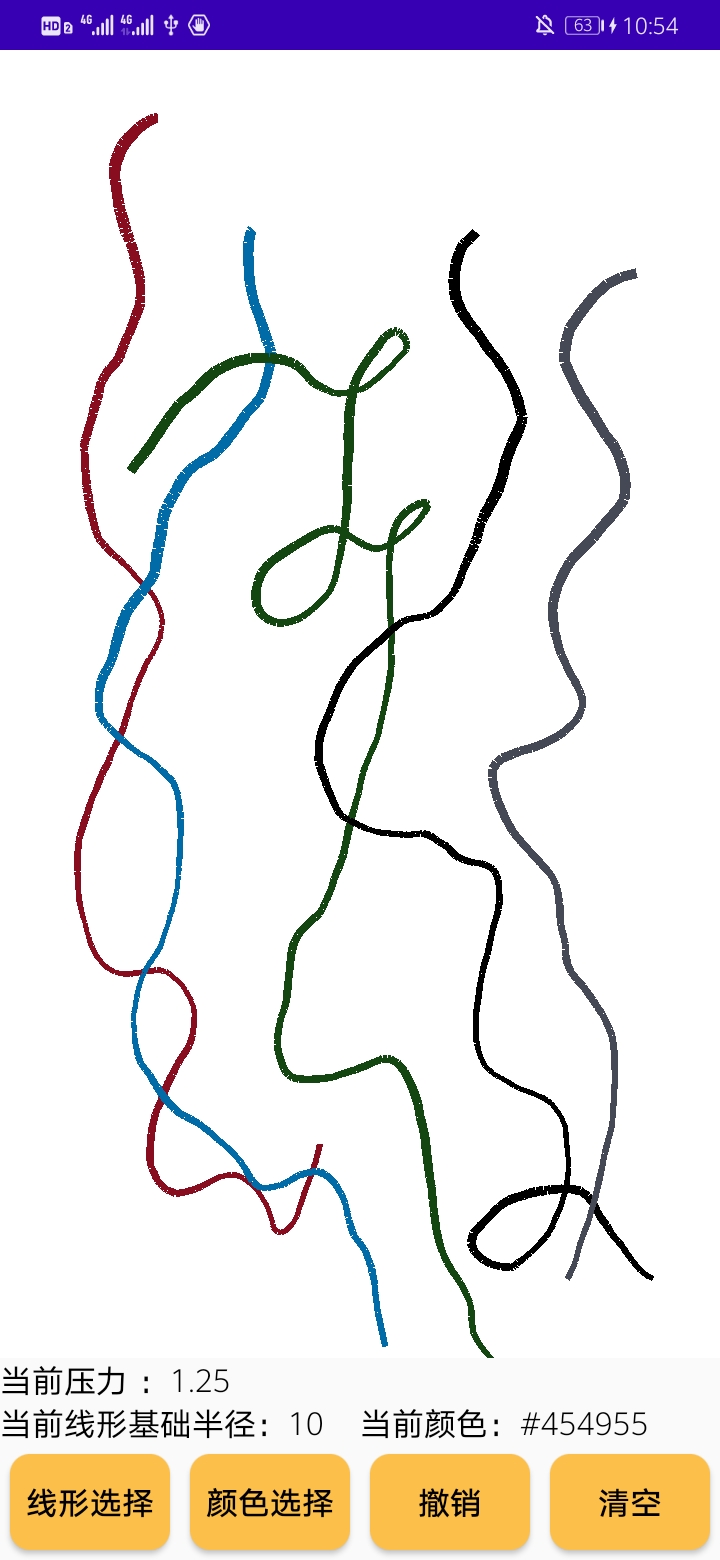


图4-3 基本效果

1. 点击线形选择按钮，弹出对话框，可以点击已有按钮或自定义：



图4-4 线形选择

5、点击确定后提示当前选择的线性粗细，可以点击已有按钮或自定义：

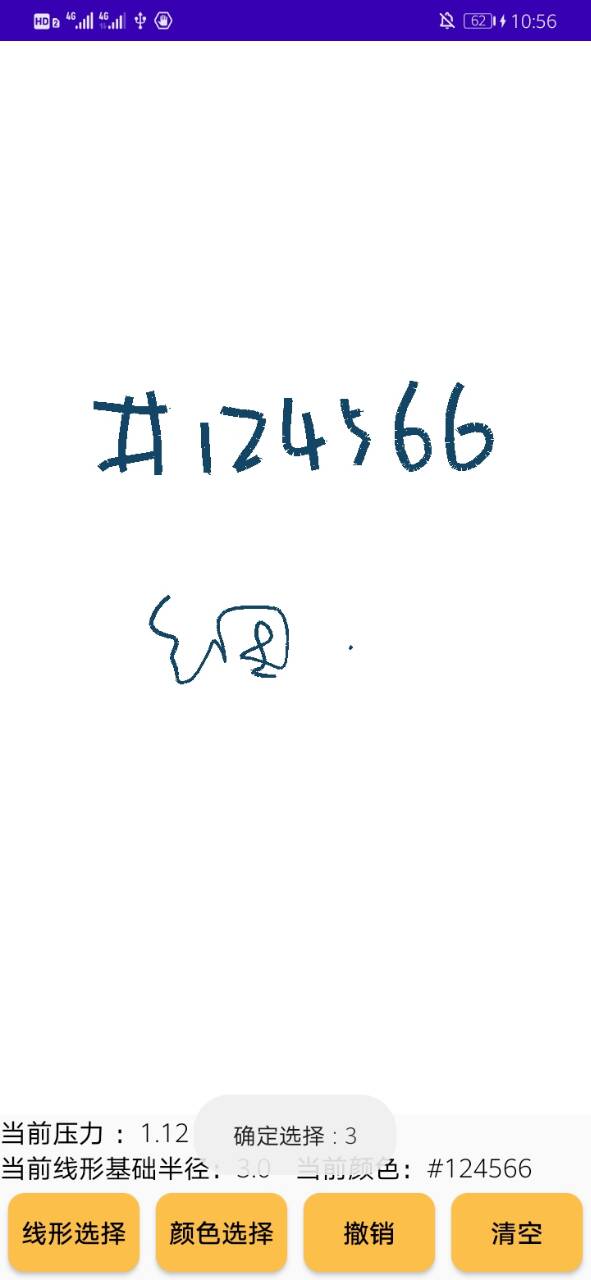


图4-5 确定选择

6、点击选择颜色按钮，弹出对话框：



图4-6 颜色选择

1. 点击确定弹出提示



图4-7 确定选择

1. 有实时触点压力提示和线条粗细变化，测试撤销效果，附清空和撤销demo

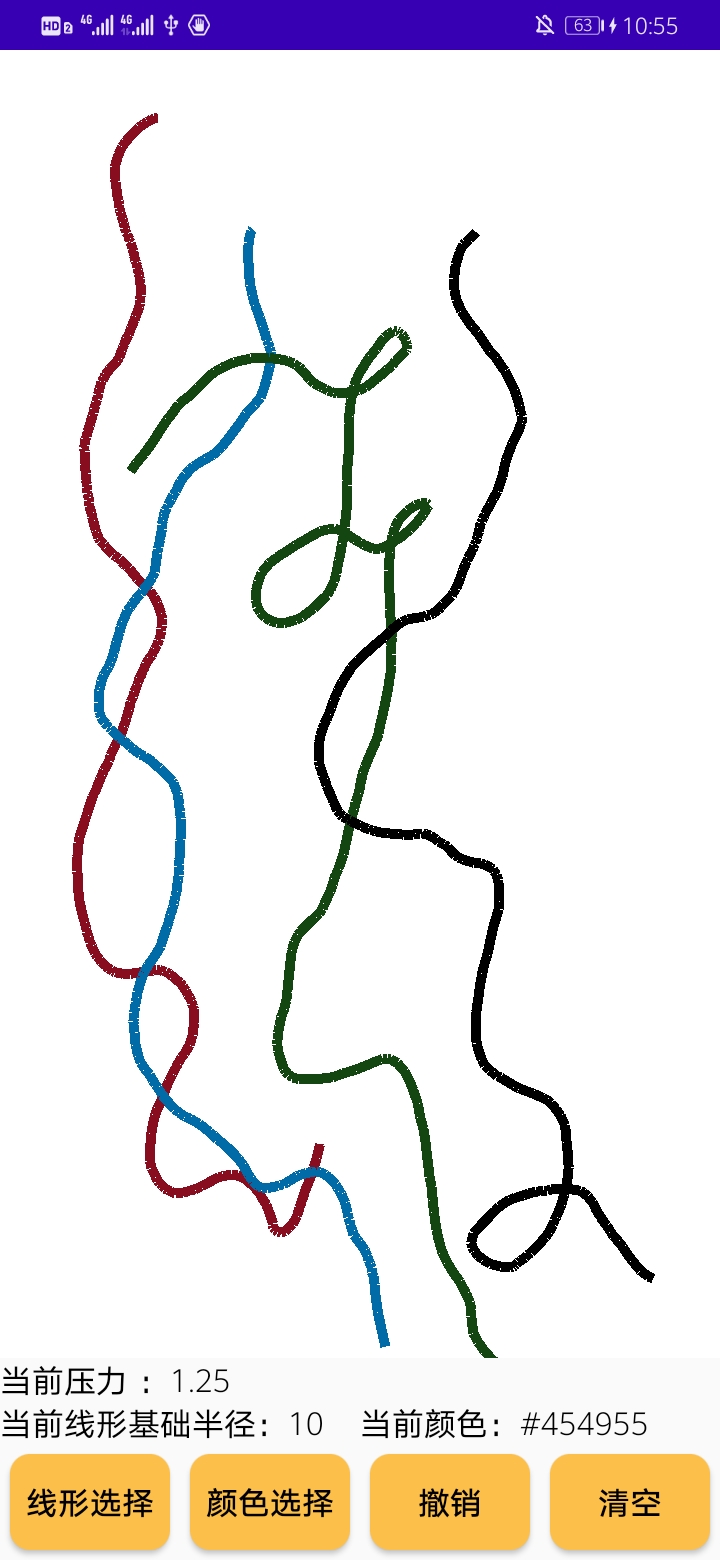
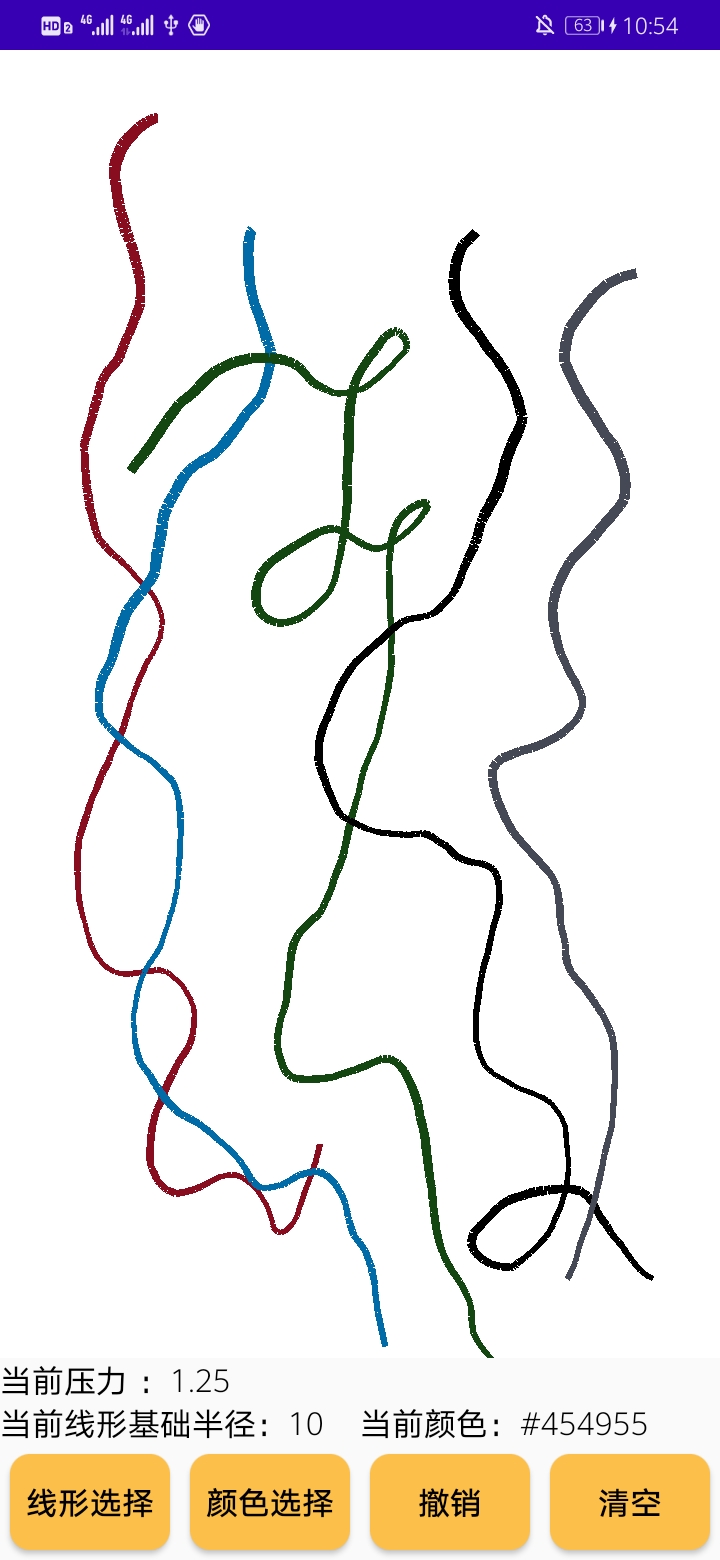


图4-8 真机撤销运行效果（可以看右上角时间）

（五）实验收获及总结

在对简单搜索网上代码后没有找到非常合适的参考，决定自己写。近期在自学dailog于是把自定义dailog的内容也加入了，过程中涉及到调参。刚开始以为要求中的重做是指清空画布，和老师交流后才知道是撤销，于是又加了个撤销。刚开始以为撤销会很困难，有想过偷懒用背景色覆盖，后来想到这样的话可能存在最新线条覆盖原有线条直接用背景色会导致原有线条不完整，于是用vector套vector来存线条，最外层vector存线条，内层存每条线的组成的线段。复习了我的java编写能力。

有考虑过那种pad上的笔记软件是怎么完成撤销这件事的，如果笔记太多不方便从头开始做，有些软件并不支持很多步撤销可能是存了一定基础上的画布，如果支持很多很多撤销可能可以对总线条分块，每块存一个画布，大大减少绘制笔触的次数，不过还要调参，具体实现效果也不知道如何。