3.6 Simple Draw

mainActivity.java

|  |
| --- |
| package com.example.mypen;  import android.content.Context; import android.graphics.Bitmap; import android.graphics.Canvas; import android.graphics.Color; import android.graphics.Paint; import android.graphics.Path; import android.os.Bundle;  import com.google.android.material.floatingactionbutton.FloatingActionButton; import com.google.android.material.snackbar.Snackbar;  import androidx.appcompat.app.AppCompatActivity; import androidx.appcompat.widget.Toolbar;  import android.view.MotionEvent; import android.view.View; import android.view.Menu; import android.view.MenuItem; import android.widget.LinearLayout;  import java.util.ArrayList;  public class MainActivity extends AppCompatActivity {   View mView;  private Paint mPaint;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate( savedInstanceState );  setContentView( R.layout.*activity\_main* );  LinearLayout layout=(LinearLayout) findViewById( R.id.*myDrawing*);   mView=new DrawingView(this);  layout.addView(mView, new LinearLayout.LayoutParams( LinearLayout.LayoutParams.*MATCH\_PARENT*, LinearLayout.LayoutParams.*MATCH\_PARENT*));  init();   }   private void init() {  mPaint= new Paint( );  mPaint.setDither( true );  mPaint.setColor( 0xFFFFFF00);  mPaint.setStyle( Paint.Style.*STROKE*);  mPaint.setStrokeJoin( Paint.Join.*ROUND* );  mPaint.setStrokeCap( Paint.Cap.*ROUND* );  mPaint.setStrokeWidth(3);  }  private class DrawingView extends View {  private Path path;  private Bitmap mBitmap;  private Canvas mCanvas;  public DrawingView(Context context) {  super(context);  path=new Path();  mBitmap= Bitmap.*createBitmap*( 820,480,Bitmap.Config.*ARGB\_8888*);  mCanvas = new Canvas( mBitmap);  this.setBackgroundColor(Color.*BLACK* );  }  private ArrayList<PathWithPaint>\_graphics1=new ArrayList<PathWithPaint>();   @Override  public boolean onTouchEvent(MotionEvent event) {  PathWithPaint pp= new PathWithPaint();  mCanvas.drawPath( path, mPaint );   if (event.getAction()==MotionEvent.*ACTION\_DOWN*){  path.moveTo( event.getX(),event.getY());  path.lineTo( event.getX(),event.getY());  } else if (event.getAction()==MotionEvent.*ACTION\_MOVE*){  path.lineTo( event.getX(),event.getY());  pp.setPath(path);  pp.setmPaint(mPaint);  \_graphics1.add(pp);  }  invalidate();  return true;  }   @Override  protected void onDraw(Canvas canvas) {  super.onDraw( canvas );  if (\_graphics1.size()>0){  canvas.drawPath(\_graphics1.get( \_graphics1.size() - 1 ).getPath(), \_graphics1.get(\_graphics1.size()-1).getmPaint());  }  }  } } |

PathWithPaint.java

|  |
| --- |
| package com.example.mypen;  import android.graphics.Paint; import android.graphics.Path;  import java.nio.file.Path;  public class PathWithPaint {  private Path path;   public Path getPath(){  return path;  }  public void setPath(Path path){  this.path = path;  }  private Paint mPaint;  public Paint getmPaint(){  return mPaint;  }  public void setmPaint(Paint mPaint){  this.mPaint = mPaint;  } } |

Activity\_main.xml

|  |
| --- |
| <?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:id="@+id/myDrawing"  android:orientation="vertical"  android:paddingLeft="@dimen/activity\_horizontal\_margin"  android:paddingRight="@dimen/activity\_horizontal\_margin"  android:paddingTop="@dimen/activity\_vertical\_margin"  android:paddingBottom="@dimen/activity\_vertical\_margin"  tools:context=".MainActivity">  </LinearLayout> |

