import android.content.Intent;

import android.database.Cursor;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.graphics.Color;

import android.graphics.drawable.Drawable;

import android.media.Image;

import android.net.Uri;

import android.provider.MediaStore;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.ImageButton;

import android.widget.TableLayout;

import static android.provider.MediaStore.Files.FileColumns.MEDIA\_TYPE\_IMAGE;

public class MainActivity extends AppCompatActivity {

private CanvasView customCanvas;

private TableLayout toolsTableLayout;

private ImageButton xsmallButton;

private ImageButton smallButton;

private ImageButton mediumButton;

private ImageButton largeButton;

private ImageButton xlargeButton;

// private ImageButton colorWhiteButton;

private ImageButton colorBlackButton;

private ImageButton colorRedButton;

private ImageButton colorOrangeButton;

private ImageButton colorYellowButton;

private ImageButton colorGreenButton;

private ImageButton colorBlueButton;

private ImageButton colorPurpleButton;

private ImageButton colorBrownButton;

int cSwitch = 0, sSwitch = 0;

static final int PICK\_IMAGE = 1;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

customCanvas = (CanvasView) findViewById(R.id.signature\_canvas);

toolsTableLayout = (TableLayout) findViewById(R.id.toolsTableLayout);

xsmallButton = (ImageButton) findViewById(R.id.xsmallButton);

smallButton = (ImageButton) findViewById(R.id.smallButton);

mediumButton = (ImageButton) findViewById(R.id.mediumButton);

largeButton = (ImageButton) findViewById(R.id.largeButton);

xlargeButton = (ImageButton) findViewById(R.id.xlargeButton);

// colorWhiteButton = (ImageButton) findViewById(R.id.colorWhiteButton);

colorBlackButton = (ImageButton) findViewById(R.id.colorBlackButton);

colorRedButton = (ImageButton) findViewById(R.id.colorRedButton);

colorOrangeButton = (ImageButton) findViewById(R.id.colorOrangeButton);

colorYellowButton = (ImageButton) findViewById(R.id.colorYellowButton);

colorGreenButton = (ImageButton) findViewById(R.id.colorGreenButton);

colorBlueButton = (ImageButton) findViewById(R.id.colorBlueButton);

colorPurpleButton = (ImageButton) findViewById(R.id.colorPurpleButton);

colorBrownButton = (ImageButton) findViewById(R.id.colorBrownButton);

toolsTableLayout.setVisibility(View.INVISIBLE);

}

public void clearCanvas(View v) {

customCanvas.clearCanvas();

}

public void drawRectangle(View v) {

customCanvas.drawRectangle();

}

public void drawCircle(View v) {

customCanvas.drawCircle();

}

public void drawTriangle(View v) {

customCanvas.drawTriangle();

}

public void toggleToolsVisibility(View v) {

if (toolsTableLayout.getVisibility() == View.VISIBLE) {

toolsTableLayout.setVisibility(View.INVISIBLE);

} else {

toolsTableLayout.setVisibility(View.VISIBLE);

}

}

// public void loadButtonClicked(View view)

// {

// getImageFromGalleryAndLoadToCanvas();

// }

// public void getImageFromGalleryAndLoadToCanvas()

// {

// Intent intent = new Intent();

// intent.setType("image/\*");

// intent.setAction(Intent.ACTION\_GET\_CONTENT);

// startActivityForResult(Intent.createChooser(intent, "Select Picture"), PICK\_IMAGE);

// }

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

// Here we need to check if the activity that was triggers was the Image Gallery.

// If it is the requestCode will match the LOAD\_IMAGE\_RESULTS value.

// If the resultCode is RESULT\_OK and there is some data we know that an image was picked.

if (requestCode == PICK\_IMAGE && resultCode == RESULT\_OK && data != null)

{

// Let's read picked image data - its URI

Uri pickedImage = data.getData();

// Let's read picked image path using content resolver

String[] filePath = { MediaStore.Images.Media.DATA };

Cursor cursor = getContentResolver().query(pickedImage, filePath, null, null, null);

cursor.moveToFirst();

String imagePath = cursor.getString(cursor.getColumnIndex(filePath[0]));

BitmapFactory.Options options = new BitmapFactory.Options();

options.inPreferredConfig = Bitmap.Config.ARGB\_8888;

Bitmap bitmap = BitmapFactory.decodeFile(imagePath, options);

customCanvas.loadImageOntoCanvas(bitmap);

// Do something with the bitmap

// At the end remember to close the cursor or you will end with the RuntimeException!

cursor.close();

}

}

public void setColor(View v) {

if (v instanceof ImageButton) {

ImageButton button = (ImageButton)v;

switch (button.getId()) {

case R.id.colorBlackButton:

customCanvas.getmPaint().setColor(Color.BLACK);

break;

case R.id.colorWhiteButton:

customCanvas.getmPaint().setColor(Color.WHITE);

break;

case R.id.colorRedButton:

customCanvas.getmPaint().setColor(Color.RED);

break;

case R.id.colorOrangeButton:

customCanvas.getmPaint().setARGB(255, 247, 148, 29);

break;

case R.id.colorYellowButton:

customCanvas.getmPaint().setARGB(255, 255, 242, 0);

break;

case R.id.colorGreenButton:

customCanvas.getmPaint().setARGB(255, 57, 181, 74);

break;

case R.id.colorBlueButton:

customCanvas.getmPaint().setARGB(255, 0, 174, 239);

break;

case R.id.colorPurpleButton:

customCanvas.getmPaint().setARGB(255, 133, 96, 168);

break;

case R.id.colorBrownButton:

customCanvas.getmPaint().setARGB(255, 117, 76, 36);

break;

default:

break;

}

}

}

public void setSize(View v) {

if (v instanceof ImageButton) {

ImageButton button = (ImageButton)v;

switch (button.getId()) {

case R.id.xsmallButton:

customCanvas.getmPaint().setStrokeWidth(4f);

break;

case R.id.smallButton:

customCanvas.getmPaint().setStrokeWidth(7f);

break;

case R.id.mediumButton:

customCanvas.getmPaint().setStrokeWidth(10f);

break;

case R.id.largeButton:

customCanvas.getmPaint().setStrokeWidth(13f);

break;

case R.id.xlargeButton:

customCanvas.getmPaint().setStrokeWidth(16f);

break;

default:

break;

}

}

}

public void save(View v) {

customCanvas.save();

}

public void randomColorSwitch(View v){

customCanvas.colorSwitch(cSwitch);

cSwitch++;

}

public void randomSizeSwitch(View v){

customCanvas.sizeSwitch(sSwitch);

sSwitch++;

}

public void change(View v) {

customCanvas.change();

}

}