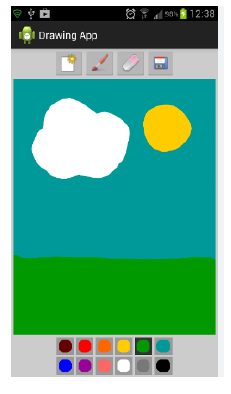
20CYS404 – Android Application Development  
Experiment – 12

Aim:

Create a Free Hand Drawing App exactly as shown  


Code:

**Activity\_main.xml**  
<?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="8dp">  
  
 <!-- Tools row: Save, New, Erase, Brush (at the top) -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:gravity="center"  
 android:background="#cccccc"  
 android:padding="8dp">  
  
 <!-- New Icon -->  
 <ImageButton  
 android:id="@+id/btnNew"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:background="@null"  
 android:src="@drawable/ic\_new"  
 android:contentDescription="New Drawing"  
 android:scaleType="centerInside" />  
  
 <!-- Erase Icon -->  
 <ImageButton  
 android:id="@+id/btnErase"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:background="@null"  
 android:src="@drawable/ic\_erase"  
 android:contentDescription="Erase Drawing"  
 android:scaleType="centerInside" />  
  
 <!-- Brush Icon -->  
 <ImageButton  
 android:id="@+id/btnBrush"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:background="@null"  
 android:src="@drawable/ic\_brush"  
 android:contentDescription="Brush Size"  
 android:scaleType="centerInside" />  
  
  
  
  
 <!-- Save Icon -->  
 <ImageButton  
 android:id="@+id/btnSave"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:background="@null"  
 android:src="@drawable/ic\_save"  
 android:contentDescription="Save Drawing"  
 android:scaleType="centerInside" />  
 </LinearLayout>  
  
 <!-- Canvas for drawing -->  
 <com.example.drawfull.CanvasView  
 android:id="@+id/canvasView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"  
 android:background="#e0f7fa" />  
  
 <!-- Color palette grid layout (at the bottom) -->  
 <GridLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:background="#cccccc"  
 android:columnCount="6"  
 android:gravity="center"  
 android:padding="8dp"  
 android:rowCount="2">  
  
 <!-- Color icons in grid format -->  
 <ImageView  
 android:id="@+id/colorBrown"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_margin="4dp"  
 android:background="#8B4513"  
 android:contentDescription="Brown Color" />  
  
 <ImageView  
 android:id="@+id/colorRed"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_margin="4dp"  
 android:background="#ff0000"  
 android:contentDescription="Red Color" />  
  
 <ImageView  
 android:id="@+id/colorOrange"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_margin="4dp"  
 android:background="#ffa500"  
 android:contentDescription="Orange Color" />  
  
 <ImageView  
 android:id="@+id/colorYellow"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_margin="4dp"  
 android:background="#ffff00"  
 android:contentDescription="Yellow Color" />  
  
 <ImageView  
 android:id="@+id/colorGreen"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_margin="4dp"  
 android:background="#008000"  
 android:contentDescription="Green Color" />  
  
 <ImageView  
 android:id="@+id/colorTeal"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_margin="4dp"  
 android:background="#008080"  
 android:contentDescription="Teal Color" />  
  
 <!-- Second row -->  
 <ImageView  
 android:id="@+id/colorBlue"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_margin="4dp"  
 android:background="#0000ff"  
 android:contentDescription="Blue Color" />  
  
 <ImageView  
 android:id="@+id/colorPurple"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_margin="4dp"  
 android:background="#800080"  
 android:contentDescription="Purple Color" />  
  
 <ImageView  
 android:id="@+id/colorPink"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_margin="4dp"  
 android:background="#ff69b4"  
 android:contentDescription="Pink Color" />  
  
 <ImageView  
 android:id="@+id/colorWhite"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_margin="4dp"  
 android:background="#ffffff"  
 android:contentDescription="White Color" />  
  
 <ImageView  
 android:id="@+id/colorGray"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_margin="4dp"  
 android:background="#808080"  
 android:contentDescription="Gray Color" />  
  
 <ImageView  
 android:id="@+id/colorBlack"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_margin="4dp"  
 android:background="#000000"  
 android:contentDescription="Black Color" />  
 </GridLayout>  
  
 <!-- User's Name -->  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Vigneswaran S [CH.EN.U4CYS21090]"  
 android:gravity="center"  
 android:layout\_gravity="center"  
 android:textStyle="bold"  
 android:textSize="16sp"  
 android:paddingTop="8dp" /> <!-- Add padding to separate from the grid layout -->  
  
</LinearLayout>

**MainActivity.java**

package com.example.drawfull;  
  
import android.Manifest;  
import android.content.pm.PackageManager;  
import android.graphics.Color;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.ImageButton;  
import android.widget.PopupMenu;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.core.content.ContextCompat;  
  
import java.util.HashMap;  
import java.util.Map;  
  
public class MainActivity extends AppCompatActivity {  
  
 private CanvasView canvasView;  
 private Map<Integer, Float> brushSizes; // Brush size map  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 canvasView = findViewById(R.id.*canvasView*);  
  
 // Initialize brush sizes  
 brushSizes = new HashMap<>();  
 brushSizes.put(R.id.*size\_small*, 5f); // Small size  
 brushSizes.put(R.id.*size\_medium*, 10f); // Medium size  
 brushSizes.put(R.id.*size\_large*, 20f); // Large size  
 brushSizes.put(R.id.*size\_extra\_large*, 30f); // Extra large size  
  
  
 // Request necessary permissions  
 if (ContextCompat.*checkSelfPermission*(this, Manifest.permission.*WRITE\_EXTERNAL\_STORAGE*)  
 != PackageManager.*PERMISSION\_GRANTED*) {  
 ActivityCompat.*requestPermissions*(this,  
 new String[]{Manifest.permission.*WRITE\_EXTERNAL\_STORAGE*},  
 1);  
 }  
  
 // Tool buttons  
 ImageButton btnNew = findViewById(R.id.*btnNew*);  
 ImageButton btnErase = findViewById(R.id.*btnErase*);  
 ImageButton btnBrush = findViewById(R.id.*btnBrush*);  
 ImageButton btnSave = findViewById(R.id.*btnSave*);  
  
 btnNew.setOnClickListener(v -> {  
 Log.*d*("MainActivity", "New button clicked");  
 canvasView.eraseAll();  
 });  
  
 btnErase.setOnClickListener(v -> {  
 Log.*d*("MainActivity", "Erase button clicked");  
 canvasView.setEraseMode(true); // Set erase mode  
 canvasView.setBrushSize(50f); // Set larger brush size for erasing  
 });  
  
 btnBrush.setOnClickListener(v -> {  
 Log.*d*("MainActivity", "Brush button clicked");  
 canvasView.setEraseMode(false); // Switch back to drawing mode  
 showBrushSizeMenu(btnBrush); // Show brush size menu  
 });  
  
 btnSave.setOnClickListener(v -> {  
 Log.*d*("MainActivity", "Save button clicked");  
 canvasView.saveImage();  
 });  
  
 // Color palette functionality  
 setUpColorPalette();  
 }  
  
 private void showBrushSizeMenu(View anchor) {  
 // Create a PopupMenu for brush sizes  
 PopupMenu popup = new PopupMenu(this, anchor);  
 popup.getMenuInflater().inflate(R.menu.*brush\_size\_menu*, popup.getMenu());  
  
 // Handle menu item clicks  
 popup.setOnMenuItemClickListener(this::onBrushSizeSelected);  
  
 // Show the popup menu  
 popup.show();  
 }  
  
 // Method to handle brush size selection  
 private boolean onBrushSizeSelected(MenuItem item) {  
 Float size = brushSizes.get(item.getItemId());  
 if (size != null) {  
 canvasView.setBrushSize(size); // Set the brush size from the map  
 return true;  
 }  
 return false;  
 }  
  
 private void setUpColorPalette() {  
 // Set onClickListeners for color ImageViews  
 findViewById(R.id.*colorBrown*).setOnClickListener(v -> {  
 Log.*d*("MainActivity", "Brown color selected");  
 canvasView.setColor(Color.*parseColor*("#8B4513")); // Brown  
 });  
  
 findViewById(R.id.*colorRed*).setOnClickListener(v -> {  
 Log.*d*("MainActivity", "Red color selected");  
 canvasView.setColor(Color.*RED*);  
 });  
  
 findViewById(R.id.*colorOrange*).setOnClickListener(v -> {  
 Log.*d*("MainActivity", "Orange color selected");  
 canvasView.setColor(Color.*parseColor*("#FFA500")); // Orange  
 });  
  
 findViewById(R.id.*colorYellow*).setOnClickListener(v -> {  
 Log.*d*("MainActivity", "Yellow color selected");  
 canvasView.setColor(Color.*YELLOW*);  
 });  
  
 findViewById(R.id.*colorGreen*).setOnClickListener(v -> {  
 Log.*d*("MainActivity", "Green color selected");  
 canvasView.setColor(Color.*parseColor*("#008000")); // Green  
 });  
  
 findViewById(R.id.*colorTeal*).setOnClickListener(v -> {  
 Log.*d*("MainActivity", "Teal color selected");  
 canvasView.setColor(Color.*parseColor*("#008080")); // Teal  
 });  
  
 findViewById(R.id.*colorBlue*).setOnClickListener(v -> {  
 Log.*d*("MainActivity", "Blue color selected");  
 canvasView.setColor(Color.*BLUE*);  
 });  
  
 findViewById(R.id.*colorPurple*).setOnClickListener(v -> {  
 Log.*d*("MainActivity", "Purple color selected");  
 canvasView.setColor(Color.*parseColor*("#800080")); // Purple  
 });  
  
 findViewById(R.id.*colorPink*).setOnClickListener(v -> {  
 Log.*d*("MainActivity", "Pink color selected");  
 canvasView.setColor(Color.*parseColor*("#FF69B4")); // Pink  
 });  
  
 findViewById(R.id.*colorWhite*).setOnClickListener(v -> {  
 Log.*d*("MainActivity", "White color selected");  
 canvasView.setColor(Color.*WHITE*);  
 });  
  
 findViewById(R.id.*colorGray*).setOnClickListener(v -> {  
 Log.*d*("MainActivity", "Gray color selected");  
 canvasView.setColor(Color.*GRAY*);  
 });  
  
 findViewById(R.id.*colorBlack*).setOnClickListener(v -> {  
 Log.*d*("MainActivity", "Black color selected");  
 canvasView.setColor(Color.*BLACK*);  
 });  
 }  
}

**CanvasView.java**

package com.example.drawfull;  
  
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.graphics.PorterDuff;  
import android.graphics.PorterDuffXfermode;  
import android.os.Environment;  
import android.util.AttributeSet;  
import android.view.MotionEvent;  
import android.view.View;  
import android.widget.Toast;  
  
import java.io.File;  
import java.io.FileOutputStream;  
  
public class CanvasView extends View {  
  
 private Paint mPaint;  
 private Path mPath;  
 private Bitmap mBitmap;  
 private Canvas mCanvas;  
 private boolean isEraseMode = false; // Track erase mode  
  
 public CanvasView(Context context, AttributeSet attrs) {  
 super(context, attrs);  
 init();  
 }  
  
 private void init() {  
 mPaint = new Paint();  
 mPaint.setAntiAlias(true);  
 mPaint.setColor(Color.*BLACK*); // Default color is black  
 mPaint.setStyle(Paint.Style.*STROKE*);  
 mPaint.setStrokeWidth(10f); // Default brush size  
  
 mPath = new Path();  
 }  
  
 @Override  
 protected void onDraw(Canvas canvas) {  
 super.onDraw(canvas);  
 if (mBitmap != null) {  
 canvas.drawBitmap(mBitmap, 0, 0, null); // Draw the saved bitmap  
 }  
 canvas.drawPath(mPath, mPaint); // Draw the current path being drawn  
 }  
  
 @Override  
 public boolean onTouchEvent(MotionEvent event) {  
 float x = event.getX();  
 float y = event.getY();  
  
 switch (event.getAction()) {  
 case MotionEvent.*ACTION\_DOWN*:  
 mPath.moveTo(x, y); // Start a new path  
 break;  
 case MotionEvent.*ACTION\_MOVE*:  
 mPath.lineTo(x, y); // Add points to the path  
 break;  
 case MotionEvent.*ACTION\_UP*:  
 // Finish the path and draw it on the bitmap  
 if (mBitmap == null) {  
 mBitmap = Bitmap.*createBitmap*(getWidth(), getHeight(), Bitmap.Config.*ARGB\_8888*);  
 mCanvas = new Canvas(mBitmap);  
 }  
 mCanvas.drawPath(mPath, mPaint); // Draw the path to the bitmap  
 mPath.reset(); // Reset the path for a new one  
 break;  
 }  
 invalidate(); // Redraw the view  
 return true;  
 }  
  
 public void setColor(int color) {  
 if (!isEraseMode) { // Change color only if not in erase mode  
 mPaint.setColor(color); // Set the paint color  
 }  
 }  
  
 public void setBrushSize(float size) {  
 mPaint.setStrokeWidth(size); // Set the brush size  
 if (isEraseMode) {  
 mPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.*CLEAR*)); // Use clear mode for erasing  
 } else {  
 mPaint.setXfermode(null); // Reset to normal drawing mode  
 }  
 }  
  
 public void setEraseMode(boolean erase) {  
 isEraseMode = erase;  
 if (erase) {  
 mPaint.setColor(Color.*TRANSPARENT*); // Set paint color to transparent for erasing  
 mPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.*CLEAR*)); // Use clear mode for erasing  
 } else {  
 mPaint.setXfermode(null); // Reset to normal drawing mode  
 mPaint.setColor(Color.*BLACK*); // Set paint color back to default  
 }  
 }  
  
 public void eraseAll() {  
 mBitmap = null; // Clear the bitmap  
 invalidate(); // Redraw the view  
 }  
  
 public void saveImage() {  
 if (mBitmap == null) {  
 Toast.*makeText*(getContext(), "Nothing to save", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 // Get external storage directory  
 File directory = Environment.*getExternalStorageDirectory*();  
 String filename = "Drawfull\_" + System.*currentTimeMillis*() + ".png";  
 File file = new File(directory, filename);  
  
 try (FileOutputStream fos = new FileOutputStream(file)) {  
 mBitmap.compress(Bitmap.CompressFormat.*PNG*, 100, fos); // Compress and save bitmap  
 Toast.*makeText*(getContext(), "Image saved to " + file.getPath(), Toast.*LENGTH\_SHORT*).show();  
 } catch (Exception e) {  
 e.printStackTrace();  
 Toast.*makeText*(getContext(), "Error saving image", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
}

AndriodManifest.xml  
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Drawfull"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
 <!-- Permission to write to external storage -->  
 <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  
</manifest>

Output:  
