

NAME –Aditya Vatsa

Roll-CH.EN.U4CYS21003

MainActivity.java

```
package com.example.drawingapp;

import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;
import android.widget.LinearLayout;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    private DrawingView drawView;
    private ImageButton currPaint, drawBtn, eraseBtn, saveBtn, newBtn;
    private float mediumBrush, largeBrush;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        drawView = findViewById(R.id.drawing);

        // Initialize color buttons
        LinearLayout paintLayout = findViewById(R.id.paint_colors);
        currPaint = (ImageButton) paintLayout.getChildAt(0); // Default color
        currPaint.setImageDrawable(getDrawable(R.drawable.paint_pressed));

        // Initialize buttons from layout
        drawBtn = findViewById(R.id.draw_btn);
        eraseBtn = findViewById(R.id.erase_btn);
        saveBtn = findViewById(R.id.save_btn);
        newBtn = findViewById(R.id.new_btn);

        // Set click listeners for buttons
        drawBtn.setOnClickListener(this);
        eraseBtn.setOnClickListener(this);
        saveBtn.setOnClickListener(this);
        newBtn.setOnClickListener(this);
    }
}
```

```

        // Get brush sizes from resources
        mediumBrush = getResources().getDimension(R.dimen.medium_size);
        largeBrush = getResources().getDimension(R.dimen.large_size);

        // Set up color button listeners
        setupColorButtons(paintLayout);
    }

    @Override
    public void onClick(View view) {
        if (view == newBtn) {
            drawView.startNew(); // Start new drawing
        } else if (view == drawBtn) {
            drawView.setErase(false); // Switch to drawing mode
            drawView.setBrushSize(mediumBrush);
        } else if (view == eraseBtn) {
            drawView.setErase(true); // Switch to eraser mode
            drawView.setBrushSize(largeBrush);
        } else if (view == saveBtn) {
            // Save functionality (not implemented yet)
            Toast.makeText(this, "Save feature coming soon!", Toast.LENGTH_SHORT).show();
        }
    }

    private void setupColorButtons(LinearLayout paintLayout) {
        for (int i = 0; i < paintLayout.getChildCount(); i++) {
            final ImageButton colorButton = (ImageButton) paintLayout.getChildAt(i);
            colorButton.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    paintClicked(v);
                }
            });
        }
    }

    public void paintClicked(View view) {
        drawView.setErase(false); // Switch to drawing mode
        drawView.setBrushSize(drawView.getLastBrushSize()); // Maintain last brush size

        if (view != currPaint) {
            ImageButton imgView = (ImageButton) view;
            String color = view.getTag().toString();
            drawView.setColor(color); // Set the color in DrawingView
            imgView.setImageDrawable(getDrawable(R.drawable.paint_pressed));
            currPaint.setImageDrawable(getDrawable(R.drawable.paint));
            currPaint = imgView; // Update current paint
        }
    }

```

```
}  
}
```

DrawingView.java

```
package com.example.drawingapp;  
  
import android.content.Context;  
import android.graphics.*;  
import android.util.AttributeSet;  
import android.view.MotionEvent;  
import android.view.View;  
  
public class DrawingView extends View {  
  
    private Path drawPath;  
    private Paint drawPaint, canvasPaint;  
    private int paintColor = 0xFF660000; // Default color  
    private Canvas drawCanvas;  
    private Bitmap canvasBitmap;  
    private float brushSize, lastBrushSize;  
    private boolean erase = false;  
  
    public DrawingView(Context context, AttributeSet attrs) {  
        super(context, attrs);  
        setupDrawing();  
    }  
  
    private void setupDrawing() {  
        drawPath = new Path();  
        drawPaint = new Paint();  
        drawPaint.setColor(paintColor);  
        drawPaint.setAntiAlias(true);  
        drawPaint.setStyle(Paint.Style.STROKE);  
        drawPaint.setStrokeJoin(Paint.Join.ROUND);  
        drawPaint.setStrokeCap(Paint.Cap.ROUND);  
        canvasPaint = new Paint(Paint.DITHER_FLAG);  
  
        // Default brush sizes  
        brushSize = getResources().getDimension(R.dimen.medium_size);  
        lastBrushSize = brushSize;  
        drawPaint.setStrokeWidth(brushSize);  
    }  
  
    @Override  
    protected void onSizeChanged(int w, int h, int oldW, int oldH) {  
        super.onSizeChanged(w, h, oldW, oldH);  
        canvasBitmap = Bitmap.createBitmap(w, h, Bitmap.Config.ARGB_8888);  
    }  
}
```

```

        drawCanvas = new Canvas(canvasBitmap);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        canvas.drawBitmap(canvasBitmap, 0, 0, canvasPaint);
        canvas.drawPath(drawPath, drawPaint);
    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {

        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN:
                drawPath.moveTo(touchX, touchY);
                break;
            case MotionEvent.ACTION_MOVE:
                drawPath.lineTo(touchX, touchY);
                break;
            case MotionEvent.ACTION_UP:
                drawCanvas.drawPath(drawPath, drawPaint);
                drawPath.reset();
                break;
            default:
                return false;
        }
        invalidate(); // Refresh the view
        return true; // Indicate that the event has been handled
    }

    // Set the paint color
    public void setColor(String newColor) {
        invalidate(); // Refresh to reflect color change
        paintColor = Color.parseColor(newColor);
        drawPaint.setColor(paintColor);
        if (erase) {
            setErase(false); // Switch back to drawing mode if erasing was enabled
        }
    }

    // Set the brush size
    public void setBrushSize(float newSize) {
        brushSize = newSize;
        drawPaint.setStrokeWidth(brushSize);
    }

```

```

// Set the erase mode
public void setErase(boolean isErase) {
    erase = isErase;
    if (erase) {
        drawPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.CLEAR));
    } else {
        drawPaint.setXfermode(null);
        drawPaint.setColor(paintColor); // Ensure we reset to the last color when
erasing is turned off
    }
}

// Get the last used brush size
public float getLastBrushSize() {
    return lastBrushSize;
}

// Start a new drawing
public void startNew() {
    drawCanvas.drawColor(0, PorterDuff.Mode.CLEAR); // Clear the canvas
    invalidate(); // Refresh the view
}
}

```

XML code :

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#E0E0E0">

    <!-- Toolbar with buttons (at the top) -->
    <LinearLayout
        android:id="@+id/tools"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:background="#607D8B"
        android:padding="10dp"
        android:elevation="5dp"
        android:gravity="center_horizontal"
        android:layout_alignParentTop="true">

        <ImageButton

```

```
    android:id="@+id/new_btn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/sun_icon"
    android:contentDescription="New Canvas"
    android:background="?attr/selectableItemBackground"
    android:tint="#FFFFFF"
    android:padding="10dp"
    android:layout_marginRight="10dp"
    android:elevation="3dp"
    android:scaleType="centerInside" />
```

<ImageButton

```
    android:id="@+id/draw_btn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/pencil_icon"
    android:contentDescription="Draw"
    android:background="?attr/selectableItemBackground"
    android:tint="#FFFFFF"
    android:padding="10dp"
    android:layout_marginRight="10dp"
    android:elevation="3dp"
    android:scaleType="centerInside" />
```

<ImageButton

```
    android:id="@+id/erase_btn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/eraser_icon"
    android:contentDescription="Erase"
    android:background="?attr/selectableItemBackground"
    android:tint="#FFFFFF"
    android:padding="10dp"
    android:layout_marginRight="10dp"
    android:elevation="3dp"
    android:scaleType="centerInside" />
```

<ImageButton

```
    android:id="@+id/save_btn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/save_icon"
    android:contentDescription="Save"
    android:background="?attr/selectableItemBackground"
    android:tint="#FFFFFF"
    android:padding="10dp"
    android:layout_marginRight="10dp"
    android:elevation="3dp"
```

```

        android:scaleType="centerInside" />
</LinearLayout>

<!-- Canvas area for drawing -->
<com.example.drawingapp.DrawingView
    android:id="@+id/drawing"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@id/tools"
    android:background="#FFFFFF"
    android:layout_margin="15dp"
    android:elevation="2dp"
    android:backgroundTint="#E0F7FA" />

<!-- Color palette (at the bottom) -->
<LinearLayout
    android:id="@+id/paint_colors"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center"
    android:background="#607D8B"
    android:elevation="5dp"
    android:padding="10dp"
    android:layout_alignParentBottom="true">

    <ImageButton
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:background="@drawable/paint_color"
        android:layout_margin="5dp"
        android:elevation="3dp"
        android:tag="#F44336"
        android:backgroundTint="#F44336"
        android:contentDescription="Red" />

    <ImageButton
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:background="@drawable/paint_color"
        android:layout_margin="5dp"
        android:elevation="3dp"
        android:tag="#FFEB3B"
        android:backgroundTint="#FFEB3B"
        android:contentDescription="Yellow" />

    <ImageButton
        android:layout_width="40dp"

```

```

        android:layout_height="40dp"
        android:background="@drawable/paint_color"
        android:layout_margin="5dp"
        android:elevation="3dp"
        android:tag="#8BC34A"
        android:backgroundTint="#8BC34A"
        android:contentDescription="Green" />

<ImageButton
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:background="@drawable/paint_color"
    android:layout_margin="5dp"
    android:elevation="3dp"
    android:tag="#03A9F4"
    android:backgroundTint="#03A9F4"
    android:contentDescription="Blue" />

<ImageButton
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:background="@drawable/paint_color"
    android:layout_margin="5dp"
    android:elevation="3dp"
    android:tag="#9C27B0"
    android:backgroundTint="#9C27B0"
    android:contentDescription="Purple" />
</LinearLayout>

</RelativeLayout>

```

Output Screenshot :

Laptop output :



Mobile Screenshot:

