<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 package="nl.alswin.tvart">  
  
 <uses-permission android:name="android.permission.INTERNET" />  
  
 <uses-feature  
 android:name="android.hardware.touchscreen"  
 android:required="false" />  
 <uses-feature  
 android:name="android.software.leanback"  
 android:required="true" />  
  
 <supports-screens android:smallScreens="false"  
 android:normalScreens="false"  
 android:largeScreens="true"  
 android:xlargeScreens="true"/>  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.TVart">  
 <activity  
 android:name=".MainActivity"  
 android:banner="@drawable/app\_icon\_your\_company"  
 android:exported="true"  
 android:icon="@drawable/app\_icon\_your\_company"  
 android:label="@string/app\_name"  
 android:logo="@drawable/app\_icon\_your\_company"  
 android:screenOrientation="landscape"  
 android:theme="@style/Theme.Leanback">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LEANBACK\_LAUNCHER" />  
 </intent-filter>  
 </activity>  
  
 </application>  
  
</manifest>

package nl.alswin.tvart;  
  
import android.graphics.Point;  
import android.os.Build;  
import android.os.Bundle;  
import android.os.Handler;  
import android.util.DisplayMetrics;  
import android.view.View;  
import android.view.Window;  
import android.view.WindowManager;  
import android.widget.Button;  
  
import androidx.fragment.app.FragmentActivity;  
  
  
public class MainActivity extends FragmentActivity {  
 Point[] pointArrayStart = new Point[120];  
 Point[] pointArrayEnd = new Point[120];  
 public static int *a*;  
 public static boolean *zwart* = false;  
 public int scrWidth;  
 public int scrHeight;  
 private LineView mlineView;  
 static int *k* = 0;  
 int xa1 = 30, xa2 = 30, ya1 = 10, ya2 = 2, xb1 = 1920, xb2 = -5, yb1 = 1200, yb2 = -2;  
 public static Button *stopbutton*;  
 public int ms = 75;  
 public int crhulp = 0;  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 DisplayMetrics displayMetrics = new DisplayMetrics();  
 getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);  
 if (android.os.Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 getWindow().setNavigationBarColor(getResources().getColor(R.color.*black*));  
 }  
 if (Build.VERSION.*SDK\_INT* >= 21) {  
 Window window = this.getWindow();  
 window.addFlags(WindowManager.LayoutParams.*FLAG\_DRAWS\_SYSTEM\_BAR\_BACKGROUNDS*);  
 window.clearFlags(WindowManager.LayoutParams.*FLAG\_TRANSLUCENT\_STATUS*);  
 window.setStatusBarColor(this.getResources().getColor(R.color.*black*));  
 }  
  
 if (android.os.Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  
 getWindow().setNavigationBarColor(getResources().getColor(R.color.*black*));  
 }  
  
 scrWidth = displayMetrics.widthPixels;  
 scrHeight = displayMetrics.heightPixels;  
  
 *stopbutton* = findViewById(R.id.*stopbutton*);  
 *stopbutton*.setText("Stop");  
 mlineView = (LineView) findViewById(R.id.*mlineView*);  
 mlineView.*trsp* = 255;  
 een();  
 }  
  
 public void een() {  
 xa1 = Randomizer.*generate*(0, (int) 3 \* scrWidth / 20);  
 xa2 = Randomizer.*generate*(1, (int) scrWidth / 40);  
 ya1 = Randomizer.*generate*(0, (int) 3 \* scrHeight / 10);  
 ya2 = Randomizer.*generate*(0, (int) scrHeight / 100);  
 xb1 = Randomizer.*generate*((int) 10 \* scrWidth / 20, scrWidth);  
 xb2 = Randomizer.*generate*((int) (-0.5 \* scrWidth / 200), 30);  
 yb1 = Randomizer.*generate*((int) 8 \* scrHeight / 10, (int) 12 \* scrHeight / 10);  
 yb2 = Randomizer.*generate*((int) -scrHeight / 100, (int) (0.5 \* scrHeight / 100));  
 crhulp = Randomizer.*generate*(40, scrWidth/10);  
 mlineView.cr = crhulp;  
 mlineView.*trspC* = 0;  
 mlineView.cx = Randomizer.*generate*(crhulp, scrWidth - (crhulp));  
 mlineView.cy = Randomizer.*generate*(crhulp, scrHeight - (crhulp));  
 *a* = 0;  
 Handler h = new Handler();  
 Runnable r = () -> bundel();  
 h.postDelayed(r, 2000);  
 }  
  
 public void bundel(){  
 if (mlineView.*trspC*<245){  
 mlineView.*trspC*+=10;  
 }  
 *a*+=1;  
 verzamel();  
 if (*a*>59){  
 zestig();  
 return;  
 }  
 Handler hh = new Handler();  
 Runnable rr = () -> bundel();  
 hh.postDelayed(rr,75);  
 }  
 public void zestig() {  
 ms=75;  
 *a* = 59;  
 verzamel();  
 if (*zwart* == true) {  
 mlineView.*trsp* -= 10;  
 mlineView.*trspC* = mlineView.*trsp*;  
 if (mlineView.*trsp* < 0) {  
 mlineView.*trsp* = 0;  
 mlineView.*trspC* = mlineView.*trsp*;  
 for (*a* = 0; *a* < 60; *a*++) {  
 pointArrayStart[*a*] = new Point(0, 0);  
 pointArrayEnd[*a*] = new Point(0, 0);  
 mlineView.setLvpointArrayStart(pointArrayStart[*a*]);  
 mlineView.setLvpointArrayEnd(pointArrayEnd[*a*]);  
 mlineView.draw();  
 }  
 *zwart* = false;  
 mlineView.*trsp* = 255;  
 mlineView.*trspC* = mlineView.*trsp*;  
 ms=75;  
 mlineView.refreshDrawableState();  
 een();  
 return;  
 }  
 } else {  
 *zwart* = true;  
 }  
 Handler hh = new Handler();  
 Runnable rr = () -> zestig();  
 hh.postDelayed(rr,75);  
 }  
  
 public void stopbutton(View view) {  
 System.*exit*(0);  
 }  
  
 public void verzamel() {  
 pointArrayStart[*a*] = new Point(xa1 + xa2 \* *a*, ya1 + ya2 \* *a*);  
 pointArrayEnd[*a*] = new Point(xb1 - xb2 \* *a*, yb1 - yb2 \* *a*);  
 mlineView.setLvpointArrayStart(pointArrayStart[*a*]);  
 mlineView.setLvpointArrayEnd(pointArrayEnd[*a*]);  
 mlineView.draw();  
 }  
}

package nl.alswin.tvart;  
  
import android.content.Context;  
import android.graphics.Canvas;  
import android.graphics.Color;  
import android.graphics.Paint;  
import android.graphics.Point;  
import android.util.AttributeSet;  
import android.view.View;  
  
import androidx.annotation.Nullable;  
  
public class LineView extends View {  
  
 private Paint[] paint = new Paint[120];  
 int i;  
 static Point[] *pointArrayStart* = new Point[120];  
 static Point[] *pointArrayEnd* = new Point[120];  
 public static int *trsp*;  
 public static int *trspC* = 0;  
 int cx;  
 int cy;  
 int cr;  
  
 public LineView(Context context) {  
 super(context);  
 }  
  
 public LineView(Context context, @Nullable AttributeSet attrs) {  
 super(context, attrs);  
 }  
  
 public LineView(Context context, @Nullable AttributeSet attrs, int defStyleAttr) {  
 super(context, attrs, defStyleAttr);  
 }  
  
 protected void onDraw(Canvas canvas) {  
 for (int i = 0; i < 60; i++) {  
 paint[i] = new Paint();  
 paint[i].setStrokeWidth(3);  
 if (i < 26) {  
 paint[i].setColor(Color.*argb*(*trsp*, 255, 10 \* i, 0));  
 }  
 if (i > 25 && i < 51) {  
 paint[i].setColor(Color.*argb*(*trsp*, 255, 255 - (i - 25) \* 10, (i - 25) \* 10));  
 }  
 if (i > 50 && i < 60) {  
 paint[i].setColor(Color.*argb*(*trsp*, 255 - (i - 50) \* 15, (i - 50) \* 5, 255 - (i - 50) \* 10));  
 }  
 try {  
 canvas.drawLine(*pointArrayStart*[i].x, *pointArrayStart*[i].y, *pointArrayEnd*[i].x, *pointArrayEnd*[i].y, paint[i]);  
 } catch (Exception e) {  
 //do nothing  
 }  
 if (i < 26) {  
 paint[i].setColor(Color.*argb*(*trspC*, 255, 10 \* i, 0));  
 }  
 if (i > 25 && i < 36) {  
 paint[i].setColor(Color.*argb*(*trspC*, 255, 255 - (i - 25) \* 10, (i - 25) \* 10));  
 }  
 if (i > 35 && i < 60) {  
 paint[i].setColor(Color.*argb*(*trspC*, 255 - (i - 50) \* 10, 0, 255));  
 }  
 if (i > 0 && i < 60) {  
 canvas.drawCircle(cx, cy, cr - i \* 15, paint[i]);  
 }  
 super.onDraw(canvas);  
 }  
 }  
 public void setLvpointArrayStart(Point lvpointArrayStart) {  
 LineView.*pointArrayStart*[MainActivity.*a*] = lvpointArrayStart;  
 }  
  
 public void setLvpointArrayEnd(Point lvpointArrayEnd) {  
 LineView.*pointArrayEnd*[MainActivity.*a*] = lvpointArrayEnd;  
 }  
  
 public void draw() {  
 invalidate();  
 requestLayout();  
 }  
  
}

package nl.alswin.tvart;  
  
public class Randomizer {  
  
 public static int generate(int min,int max) {  
 return min + (int)(Math.*random*() \* ((max - min) + 1));  
 }  
  
}

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout 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:background="@color/black"  
 tools:context="nl.alswin.tvart.MainActivity">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <nl.alswin.tvart.LineView  
 android:id="@+id/mlineView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
 <Button  
 android:id="@+id/stopbutton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="30dp"  
 android:layout\_marginBottom="10dp"  
 android:background="@color/transparant"  
 android:onClick="stopbutton"  
 android:text=""  
 android:textColor="@color/gray"  
 app:layout\_constraintBottom\_toBottomOf="@+id/mlineView"  
 app:layout\_constraintEnd\_toEndOf="parent" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
  
</RelativeLayout>

<resources>  
 <color name="background\_gradient\_start">#000000</color>  
 <color name="background\_gradient\_end">#DDDDDD</color>  
 <color name="fastlane\_background">#0096a6</color>  
 <color name="search\_opaque">#ffaa3f</color>  
 <color name="selected\_background">#ffaa3f</color>  
 <color name="default\_background">#3d3d3d</color>  
 <color name="purple\_200">#FFBB86FC</color>  
 <color name="purple\_500">#FF6200EE</color>  
 <color name="purple\_700">#FF3700B3</color>  
 <color name="teal\_200">#FF03DAC5</color>  
 <color name="teal\_700">#FF018786</color>  
 <color name="black">#FF000000</color>  
 <color name="white">#FFFFFFFF</color>  
 <color name="darkred">#FF990000</color>  
 <color name="darkgray">#66111111</color>  
 <color name="gray">#FF333333</color>  
 <color name="darkorange">#FFAA5500</color>  
 <color name="red">#FFFF0000</color>  
 <color name="orange">#FFFF6600</color>  
 <color name="transparant">#00000000</color>  
</resources>