**Ex. No. 5 Android Application using Multithreading**

**Vikash Anand S**

**205001127**

**CSE B**

**Aim:**

Develop an android application to perform multithreading. Define 3 threads to run concurrently when “start” button is clicked.

a. The first thread should change the color of the text indefinitely  
b. The second thread should implement a moving banner  
c. The third thread should display a counter starting from 0 to 1000  
d. When the “Stop” button is pressed all the threads should be stopped

**Additional Points:**

“Resume” button is pressed, all the threads should resume from the stopped point.

**Components used: Android widgets:**

● TextView  
● ImageView ● Button

**Layouts used:**

● ConstraintLayout

**Activity used: Activities:**

● MainActivity JAVA Class

MainActivity.java

**CODE:**

package com.example.threads;

import android.animation.ObjectAnimator;

import android.graphics.Color;

import android.os.Bundle;

import android.os.Handler;

import android.os.Looper;

import android.view.View;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

private Thread colorChangingThread;

private Thread movingBannerThread;

private Thread counterThread;

private boolean isThreadsRunning = false;

private Button start, stop, resume;

private ImageView movingImageView;

private ObjectAnimator animator;

private final Handler colorHandler = new Handler(Looper.getMainLooper());

private final Handler bannerHandler = new Handler(Looper.getMainLooper());

private final Handler counterHandler = new Handler(Looper.getMainLooper());

private int colorIndex = 0, counter = 0;

private String bannerText;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

// Initialize buttons after setContentView

start = findViewById(R.id.start);

stop = findViewById(R.id.stop);

resume = findViewById(R.id.resume);

movingImageView = findViewById(R.id.movingBanner);

animator = ObjectAnimator.ofFloat(movingImageView, "translationX", -100, 900);

animator.setDuration(2000);

animator.setRepeatCount(ObjectAnimator.INFINITE);

// Set OnClickListener for the Start button

start.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

restartThreads(view);

}

});

// Set OnClickListener for the Stop button

stop.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

stopThreads(view);

}

});

// Set OnClickListener for the Resume button

resume.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

resumeThreads(view);

}

});

}

public void restartThreads(View view) {

// Stop threads if they are running

stopThreads(view);

// Reset state variables to their initial values

colorIndex = 0;

bannerText = "This is a moving banner.... ";

counter = 999;

animator.start();

// Start threads with reset state

startThreads(view);

}

public void startThreads(View view) {

if (!isThreadsRunning) {

isThreadsRunning = true;

colorChangingThread = new Thread(() -> {

while (isThreadsRunning) {

colorHandler.post(() -> updateColor());

try {

Thread.sleep(500);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

});

colorChangingThread.start();

movingBannerThread = new Thread(() -> {

while (isThreadsRunning) {

bannerHandler.post(() -> updateBanner());

try {

Thread.sleep(500);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

});

movingBannerThread.start();

counterThread = new Thread(() -> {

while (isThreadsRunning) {

counterHandler.post(() -> updateCounter());

try {

Thread.sleep(100);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

});

counterThread.start();

}

}

public void stopThreads(View view) {

isThreadsRunning = false;

animator.pause();

}

public void resumeThreads(View view) {

startThreads(view);

}

private void updateColor() {

String[] colors = {

"#FF0000", // Red

"#FF4000", // Red-Orange

"#FF8000", // Orange

"#FFBF00", // Orange-Yellow

"#FFFF00", // Yellow

"#BFFF00", // Yellow-Green

"#80FF00", // Green

"#40FF00", // Green-Aqua

"#00FF00", // Aqua

"#00FF40", // Aqua-Green

"#00FF80", // Aqua-Blue

"#00FFBF", // Aqua-Blue (Light)

"#00FFFF", // Cyan

"#00BFFF", // Light Blue

"#0080FF", // Sky Blue

"#0040FF", // Blue

"#0000FF", // Dark Blue

"#4000FF", // Blue-Purple

"#8000FF", // Purple

"#BF00FF", // Purple-Pink

"#FF00FF" // Pink

};

TextView colorTextView = findViewById(R.id.colorShift);

colorTextView.setTextColor(Color.parseColor(colors[colorIndex]));

colorIndex = (colorIndex + 1) % colors.length;

}

private void updateBanner() {

TextView bannerTextView = findViewById(R.id.banner);

bannerTextView.setText(bannerText);

bannerText = bannerText.charAt(bannerText.length() - 1) +

bannerText.substring(0, bannerText.length() - 1);

animator.resume();

}

private void updateCounter() {

TextView counterTextView = findViewById(R.id.counter);

if (counter == 1000)

counter = 0;

counterTextView.setText(String.valueOf(counter));

counter++;

}

}

**Activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout

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"

tools:context=".MainActivity">

<TextView

android:id="@+id/textView4"

android:layout\_width="413dp"

android:layout\_height="94dp"

android:background="#00FFFF"

android:fontFamily="serif"

android:gravity="center"

android:text="Multithreads"

android:textSize="30dp"

android:textStyle="bold"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.0" />

<Button

android:id="@+id/resume"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="152dp"

android:layout\_marginTop="44dp"

android:text="Resume"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/start"

tools:ignore="MissingConstraints" />

<Button

android:id="@+id/stop"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="76dp"

android:layout\_marginTop="364dp"

android:text="Stop"

app:layout\_constraintStart\_toEndOf="@+id/start"

app:layout\_constraintTop\_toBottomOf="@+id/textView4"

tools:ignore="MissingConstraints" />

<Button

android:id="@+id/start"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="80dp"

android:layout\_marginTop="364dp"

android:text="Start"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView4"

tools:ignore="MissingConstraints" />

<TextView

android:id="@+id/counter"

android:layout\_width="72dp"

android:layout\_height="30dp"

android:layout\_marginStart="320dp"

android:layout\_marginTop="100dp"

android:text=": "

android:textSize="20dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView4"

tools:ignore="MissingConstraints" />

<TextView

android:id="@+id/counterName"

android:layout\_width="116dp"

android:layout\_height="35dp"

android:layout\_marginStart="232dp"

android:layout\_marginTop="100dp"

android:text="Counter: "

android:textSize="20dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView4"

tools:ignore="MissingConstraints" />

<TextView

android:id="@+id/banner"

android:layout\_width="386dp"

android:layout\_height="46dp"

android:layout\_marginStart="12dp"

android:layout\_marginTop="60dp"

android:fontFamily="serif"

android:text="This banner is moving....."

android:textSize="28dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/colorShift"

tools:ignore="MissingConstraints" />

<TextView

android:id="@+id/colorShift"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="16dp"

android:layout\_marginTop="96dp"

android:text="Color change"

android:textSize="30dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView4"

tools:ignore="MissingConstraints" />

<ImageView

android:id="@+id/movingBanner"

android:layout\_width="73dp"

android:layout\_height="57dp"

android:layout\_marginStart="8dp"

android:layout\_marginTop="264dp"

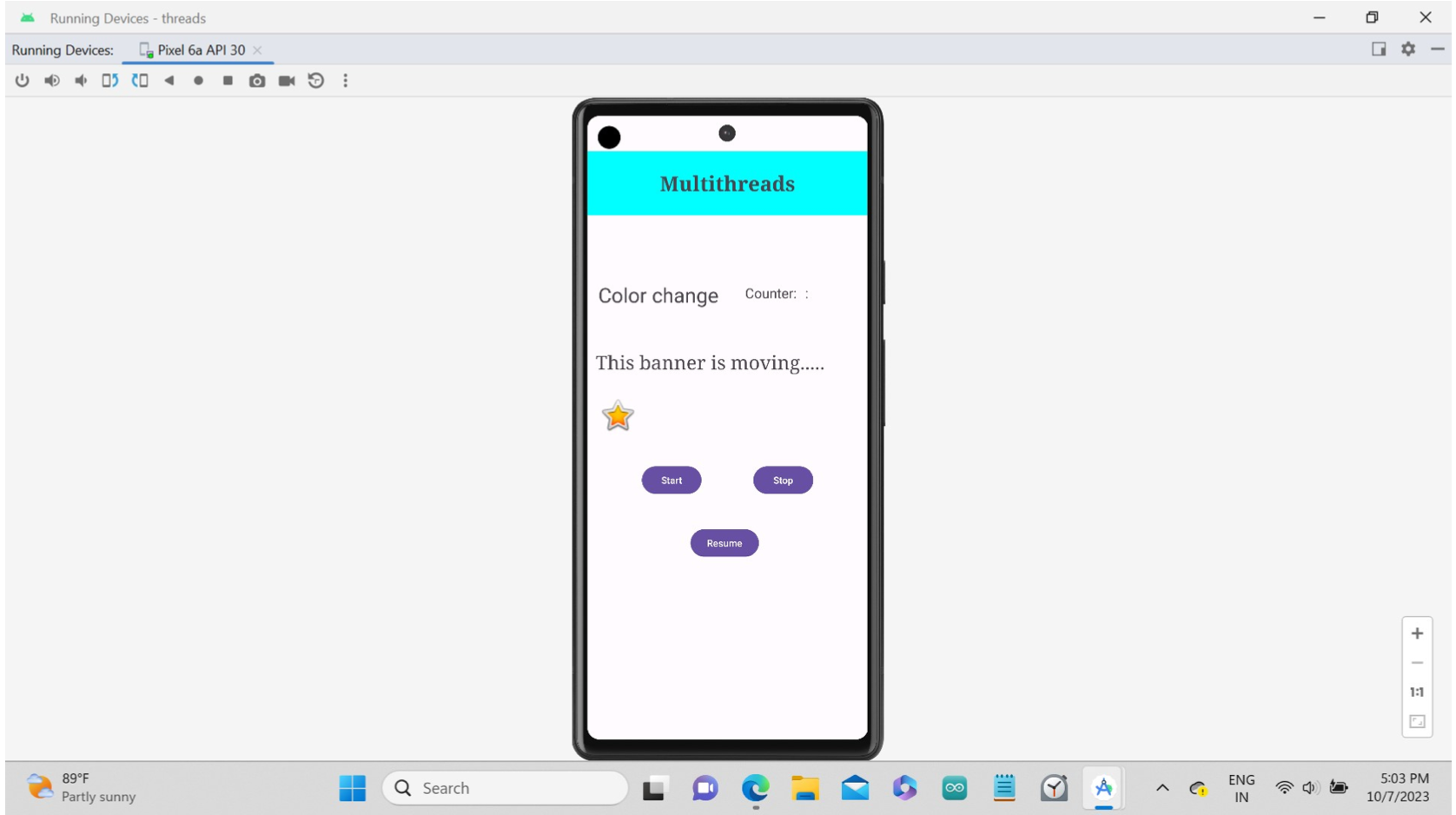
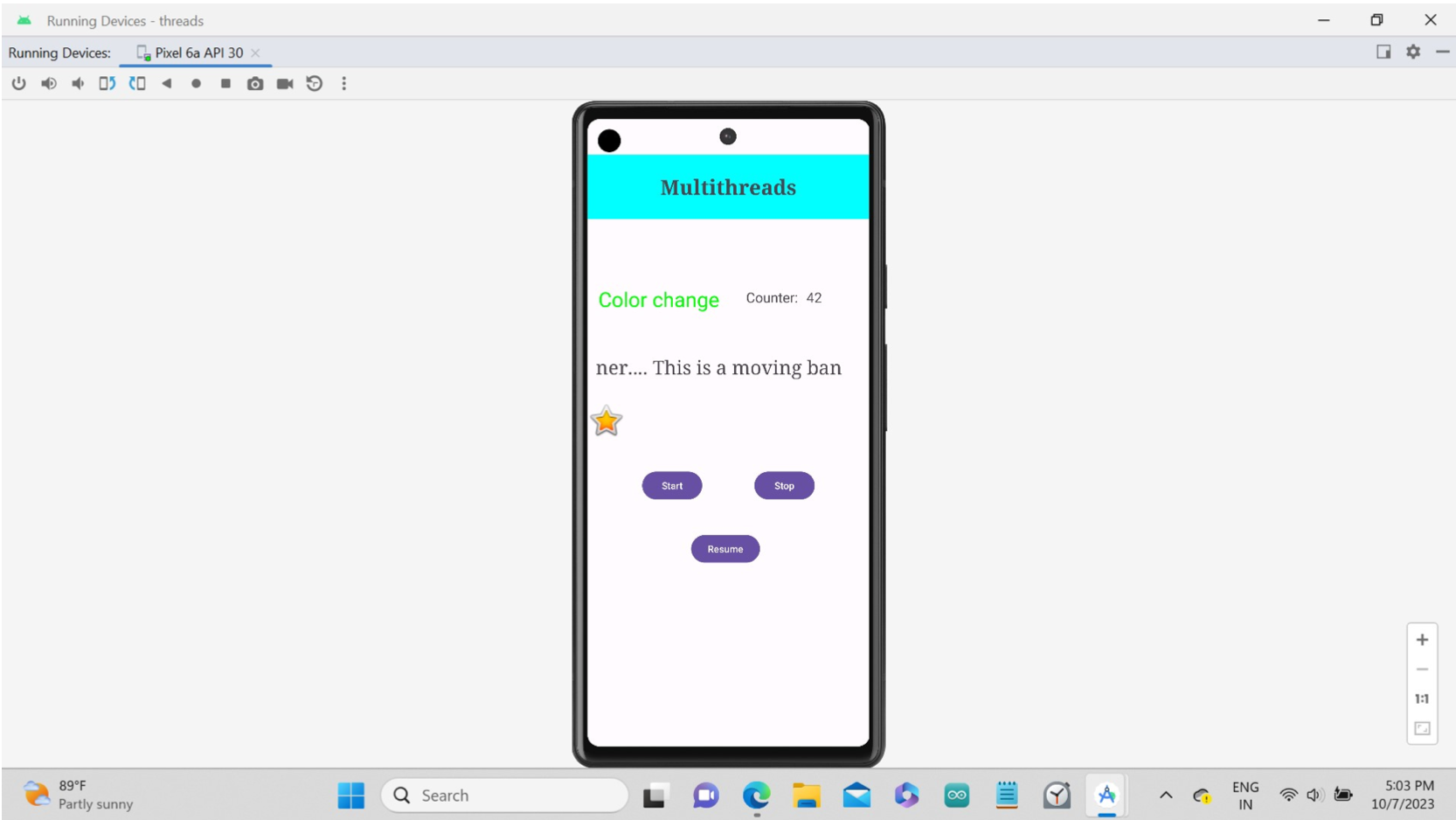
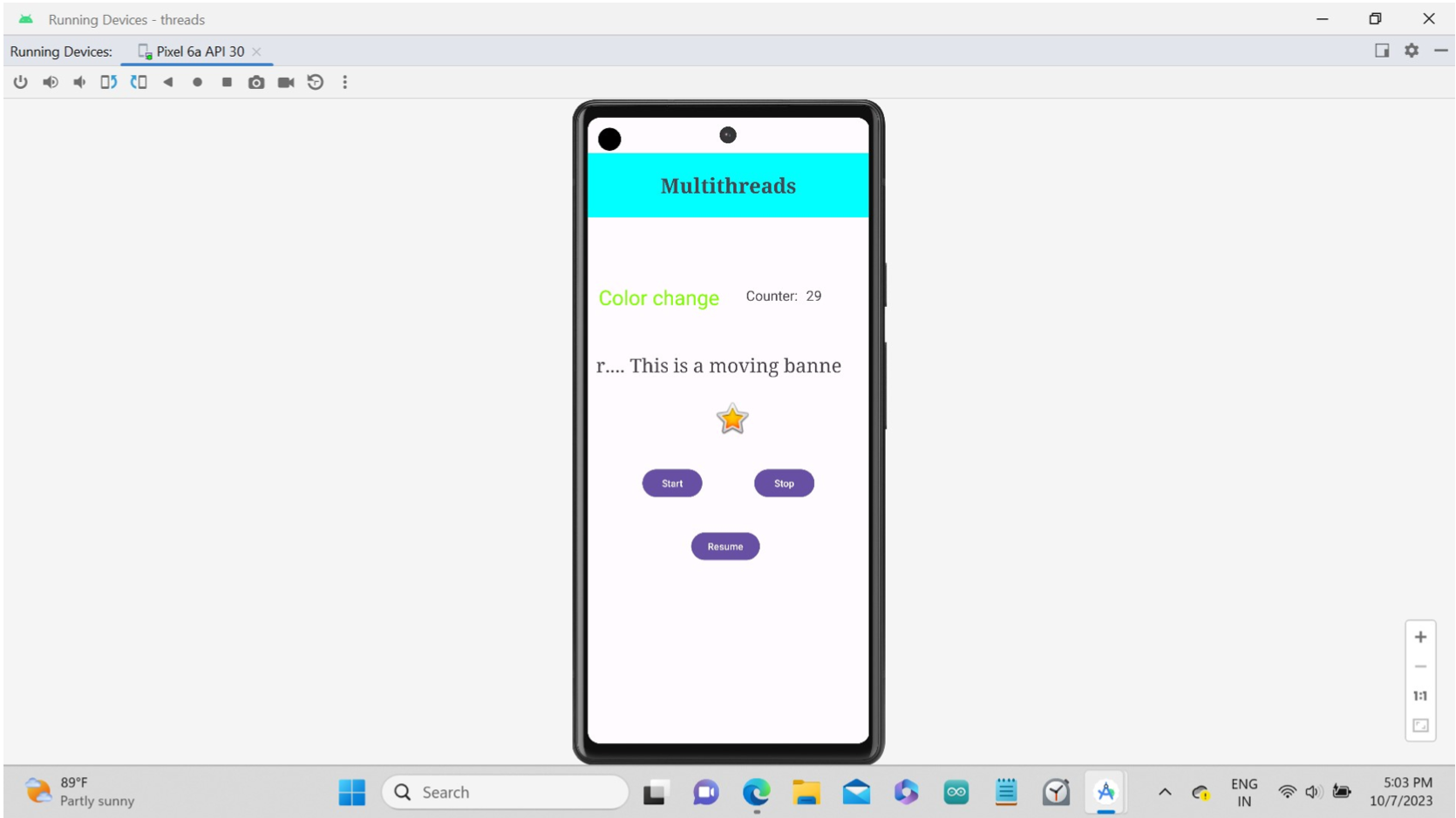
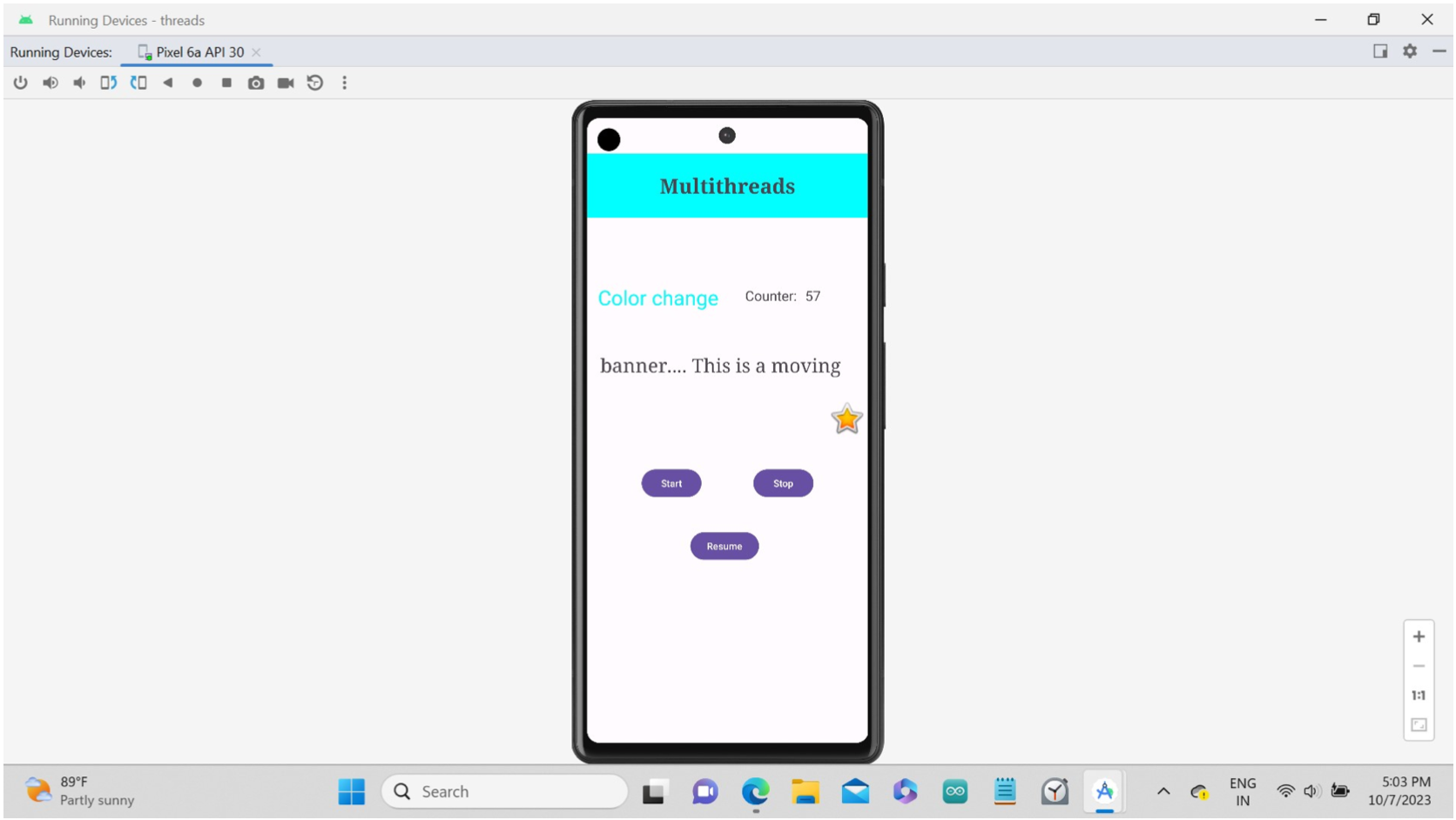
app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView4"

app:srcCompat="@android:drawable/btn\_star\_big\_on" />

</androidx.constraintlayout.widget.ConstraintLayout>

**SCREENSHOTS:**

****

**BEST PRACTICES:**

● Standard naming conventions ● Suitable comments

● Proper indentation  
● Proper user interface which is understandable and easy to navigate ● Use of modularity and functions

**LEARNING OUTCOMES:**

● I learnt the versatility of threads  
● I learnt the working of thread in android studio ● I learnt to run the emulator and debugger.