Gravity Android App

Introduction:

In this report, I create a Sensor Android App using Android Studio. I'm going to create a sensor application that changes the background color of an activity when the device is shaken.

Types of Sensors that I used:

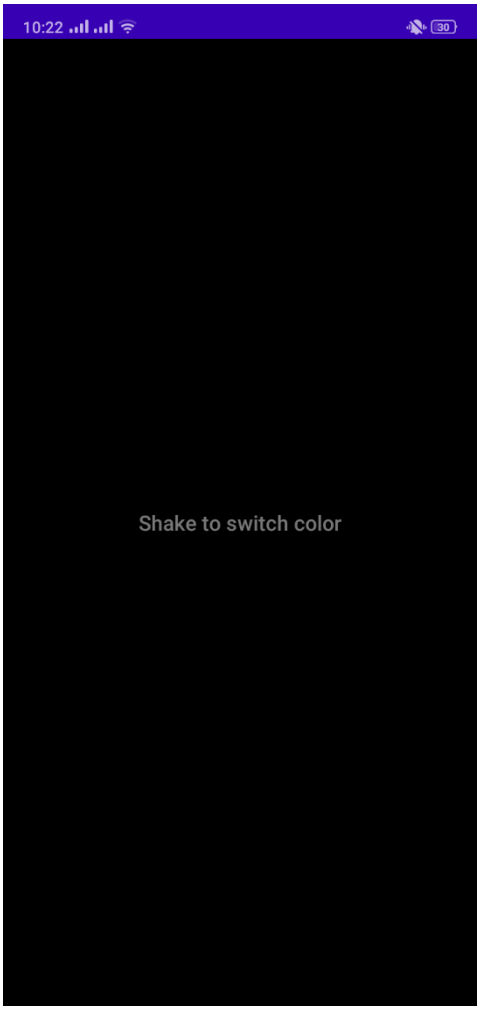
-Motion Sensors

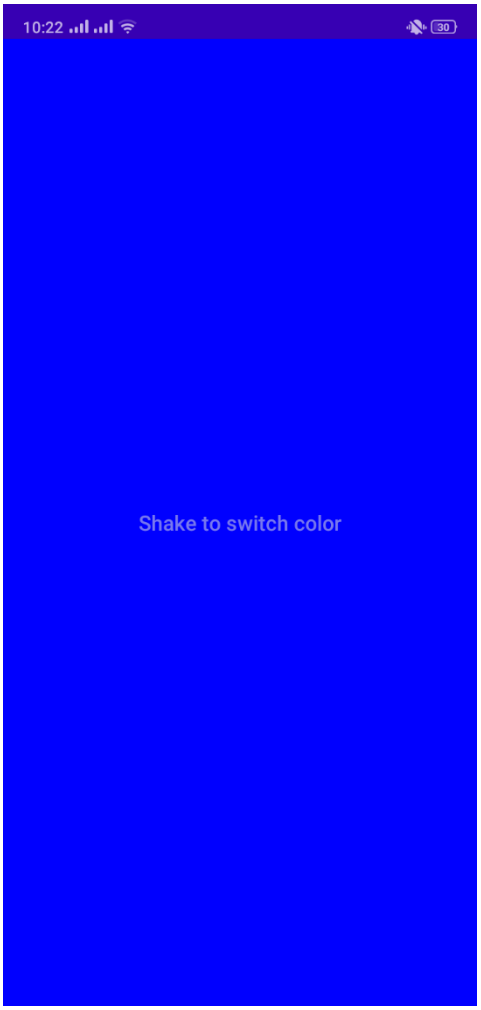
These are used to measure acceleration forces and rotational forces along with three axes.

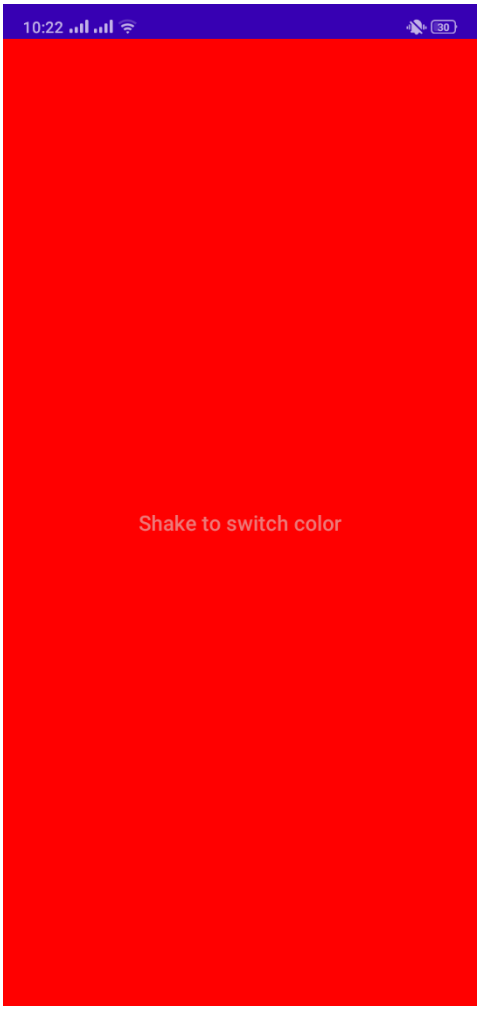
-Position Sensors

These are used to measure the physical position of the device.

Screens from the app:







The XML code:

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:textStyle="bold"
        android:text="Shake to switch color" />

</RelativeLayout>
```

The java code:

```java
package com.example.appsensor;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.graphics.Color;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;

public class MainActivity extends Activity implements SensorEventListener{
    private SensorManager sensorManager;
    private boolean isColor = false;
    private View view;
    private long lastUpdate;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        view = findViewById(R.id.textView);
        view.setBackgroundColor(Color.BLACK);
        sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
        lastUpdate = System.currentTimeMillis();
    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {}
    @Override
    public void onSensorChanged(SensorEvent event) {
        if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
            getAccelerometer(event);
        }
    }

    private void getAccelerometer(SensorEvent event) {
        float[] values = event.values;
```

```
        // Movement
        float x = values[0];
        float y = values[1];
        float z = values[2];

        float accelationSquareRoot = (x * x + y * y + z * z) / (SensorManager.GRAVITY_EARTH *
SensorManager.GRAVITY_EARTH);

        long actualTime = System.currentTimeMillis();

        if (accelationSquareRoot >= 2)
        {
            if (actualTime - lastUpdate < 200) {
                return;
            }
            lastUpdate = actualTime;
            if (isColor){
                view.setBackgroundColor(Color.BLUE);
            } else{
                view.setBackgroundColor(Color.RED);
            }
            isColor = !isColor;
        }
    }

    @Override
    protected void onResume() {
        super.onResume();

sensorManager.registerListener(this,sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
SensorManager.SENSOR_DELAY_NORMAL);
    }

    @Override
    protected void onPause() {
        super.onPause();
        sensorManager.unregisterListener(this);
    }
}
```