

Experiment Number 22 (Exercise 1):

Write a program to changes the background color when device is shuffled.

// Code for activity_main.xml file

```
<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:text="Shake to switch color" />

</RelativeLayout>
</LinearLayout>
```

// Code for MainActivity.java file

```
package com.example.myapplication;
```

```
import androidx.appcompat.app.AppCompatActivity;
import android.graphics.Color;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity 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.GREEN);

        sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
        lastUpdate = System.currentTimeMillis();
    }
    //overriding two methods of SensorEventListener
    @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 accelerationSquareRoot = (x * x + y * y + z * z)
        / (SensorManager.GRAVITY_EARTH * SensorManager.GRAVITY_EARTH);

    long actualTime = System.currentTimeMillis();
    Toast.makeText(getApplicationContext(),String.valueOf(accelerationSquareRoot)+" "+
        SensorManager.GRAVITY_EARTH,Toast.LENGTH_SHORT).show();

    if (accelerationSquareRoot >= 2) //it will be executed if you shuffle
    {

        if (actualTime - lastUpdate < 200) {
            return;
        }
        lastUpdate = actualTime;//updating lastUpdate for next shuffle
        if (isColor) {
            view.setBackgroundColor(Color.GREEN);

        } else {
            view.setBackgroundColor(Color.RED);
        }
        isColor = !isColor;
    }
}

@Override
protected void onResume() {
    super.onResume();
    // register this class as a listener for the orientation and
    // accelerometer sensors

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

@Override
protected void onPause() {
    // unregister listener
    super.onPause();
    sensorManager.unregisterListener(this);
}
}

```

Output:

