213908024 – Kaan Can Kurt

İvme Ölçer ile hareket algılama uygulaması - ödev 1

GENEL OLARAK:

İvmeölçer sensörü kullanılarak cihazın titreşimini ölçüyoruz ve belli bir eşik değerinin üstünde ise kullanıcıya deprem olduğuna dair bildirim ve ekranda yanıp sönen “ !! ACİL DEPREM BİLDİRİMİ ” diye uyarı veriyor.

-Uyarılar cihazı titreştiriyor, bildirim sesi ile uyarıyor ve yanıp sönen bir yazı şeklinde veriliyor.

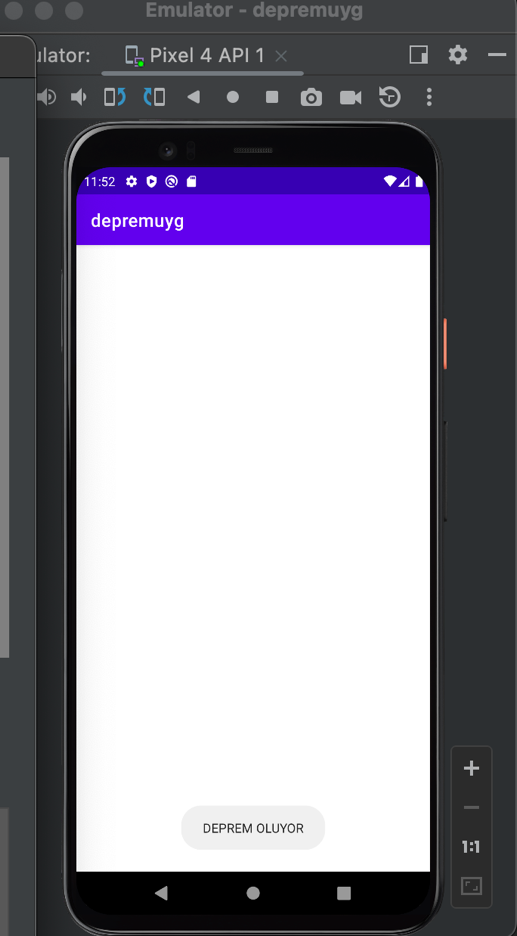
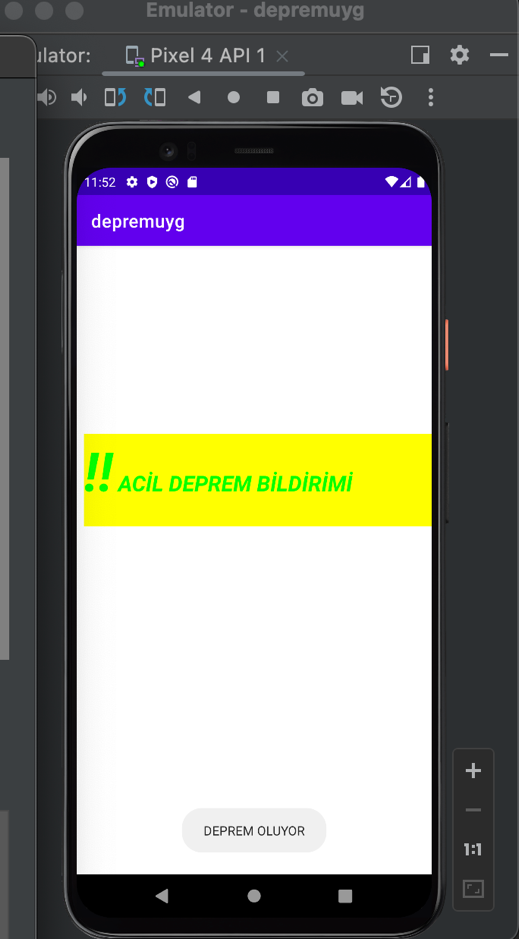
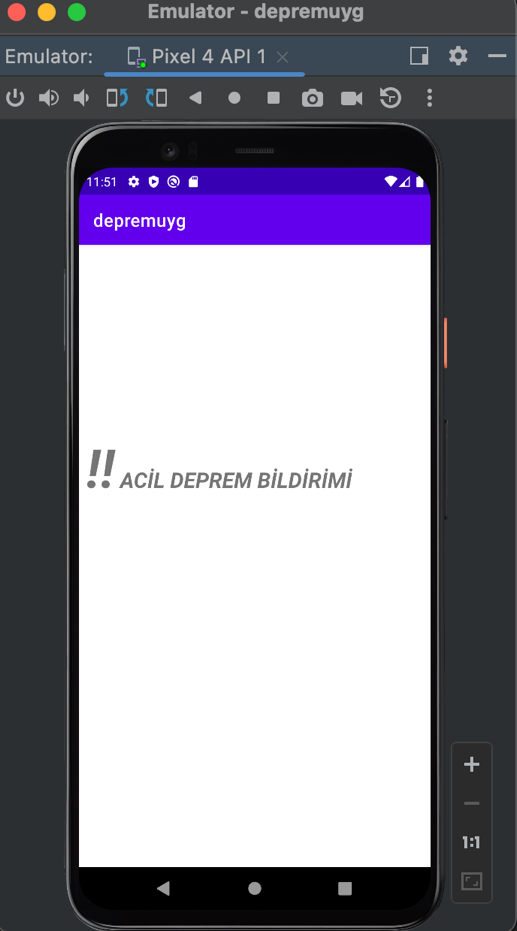
-CountDownTimer sınıfı kullanılarak ekranda yanıp sönen bir yazı efekti oluşturduk.

-Toast sınıfı kullanarak ekrana bir deprem oluyor uyarısı verdik.

-RingtoneManager ve MediaPlayer sınıflarını kullanarak default bildirim sesini loop’a aldık.

-Yazının başındaki iki ünlemi Spannable sınıfı kullanarak büyülttük index-index+2 ilk ikisi bu şekilde büyüdü

İLK AÇILDIĞINDA TİTREŞİM ALGILADIĞINDA BU ŞEKİLDE YANIP SÖNÜYOR



package com.example.depremuyg;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Context;  
import android.graphics.Color;  
import android.graphics.Typeface;  
import android.hardware.Sensor;  
import android.hardware.SensorEvent;  
import android.hardware.SensorEventListener;  
import android.hardware.SensorManager;  
import android.media.MediaPlayer;  
import android.media.Ringtone;  
import android.media.RingtoneManager;  
import android.net.Uri;  
import android.os.Build;  
import android.os.Bundle;  
import android.os.CountDownTimer;  
import android.os.VibrationEffect;  
import android.os.Vibrator;  
import android.text.Spannable;  
import android.text.SpannableString;  
import android.text.style.RelativeSizeSpan;  
import android.view.View;  
import android.widget.TextView;  
import android.widget.Toast;  
  
public class MainActivity extends AppCompatActivity implements SensorEventListener {  
  
 SensorManager sensorManager\_kck;  
 Sensor ivme\_kck;  
 float ivmelenme\_kck;  
 float simdikiIvmelenme\_kck;  
 float birOncekiIvmelenme\_kck;  
 TextView txt\_kck;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
  
 Vibrator vibrator\_kck = (Vibrator) getSystemService(Context.*VIBRATOR\_SERVICE*);  
  
 txt\_kck = findViewById(R.id.*txtGoster\_kck*);  
  
  
 sensorManager\_kck = (SensorManager) getSystemService(*SENSOR\_SERVICE*);  
 ivme\_kck = sensorManager\_kck.getDefaultSensor(Sensor.*TYPE\_ACCELEROMETER*);  
  
 sensorManager\_kck.registerListener(this, ivme\_kck, SensorManager.*SENSOR\_DELAY\_NORMAL*);  
  
 TextView textView\_kck = findViewById(R.id.*txtGoster\_kck*);  
 textView\_kck.setTextSize(22);  
 textView\_kck.setTypeface(null, Typeface.*BOLD*);  
 textView\_kck.setTypeface(null, Typeface.*BOLD\_ITALIC*);  
 String text\_kck = textView\_kck.getText().toString();  
  
 int index\_kck = text\_kck.indexOf("!!");  
 if(index\_kck >= 0){  
  
 SpannableString spannableString = new SpannableString(text\_kck);  
 spannableString.setSpan(new RelativeSizeSpan(2.5f), index\_kck,index\_kck+2, Spannable.*SPAN\_EXCLUSIVE\_EXCLUSIVE*);  
 textView\_kck.setText(spannableString);  
 }  
  
 //ilk değerler  
 ivmelenme\_kck = 0.00f;  
 simdikiIvmelenme\_kck = SensorManager.*GRAVITY\_EARTH*;  
 birOncekiIvmelenme\_kck = SensorManager.*GRAVITY\_EARTH*;  
 }  
  
 @Override  
 public void onSensorChanged(SensorEvent sensorEvent) {  
  
 float x = sensorEvent.values[0];  
 float y = sensorEvent.values[1];  
 float z = sensorEvent.values[2];  
  
 birOncekiIvmelenme\_kck = simdikiIvmelenme\_kck;  
 simdikiIvmelenme\_kck = (float) Math.*sqrt*(x\*x+y\*y+z\*z);  
  
 float degisim\_kck = simdikiIvmelenme\_kck - birOncekiIvmelenme\_kck;  
  
 ivmelenme\_kck = ivmelenme\_kck \* 0.9f + degisim\_kck;  
  
 if (ivmelenme\_kck > 2){  
 txt\_kck.setBackgroundColor(Color.*YELLOW*);  
 txt\_kck.setTextColor(Color.*GREEN*);  
  
 Vibrator vibrator\_kck = (Vibrator)getSystemService(Context.*VIBRATOR\_SERVICE*);  
 if (Build.VERSION.*SDK\_INT*>=26){  
 vibrator\_kck.vibrate(VibrationEffect.*createOneShot*(2000,VibrationEffect.*DEFAULT\_AMPLITUDE*));  
 }  
 else{  
 vibrator\_kck.vibrate(2000);  
 }  
  
  
 Toast.*makeText*(this, "DEPREM OLUYOR", Toast.*LENGTH\_SHORT*).show();  
  
 Uri notificationSound = RingtoneManager.*getDefaultUri*(RingtoneManager.*TYPE\_NOTIFICATION*);  
 Ringtone r\_kck = RingtoneManager.*getRingtone*(getApplicationContext(), notificationSound);  
 r\_kck.play();  
  
 MediaPlayer mediaPlayer = MediaPlayer.*create*(this, notificationSound);  
 mediaPlayer.setLooping(true);  
 mediaPlayer.start();  
  
  
  
 CountDownTimer timer\_kck = new CountDownTimer(Long.*MAX\_VALUE*, 1000) {  
  
 boolean visible=true;  
 @Override  
 public void onTick(long l) {  
 txt\_kck.setVisibility(visible ? View.*VISIBLE* : View.*INVISIBLE*);  
 visible = !visible;  
 }  
  
 @Override  
 public void onFinish() {  
  
 }  
 };  
 timer\_kck.start();  
 }  
 }  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 sensorManager\_kck.unregisterListener(this);  
 }  
  
 @Override  
 public void onAccuracyChanged(Sensor sensor, int i) {  
  
 }  
}