МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Звіт

з практичної роботи № 5 з дисципліни

Програмування під Android

Виконав: Перевірив:

ст. гр. ПЗПІ-23-4 ст. викладач кафедри ПІ

Цисельський Андрій Сокорчук Ігор Петрович

Харків 2024

ЗАВДАННЯ

Ознайомитися з основними типами сенсорів у Android.

Опанувати процес зчитування даних із сенсорів та їх обробку.

Навчитися використовувати клас SensorManager для керування сенсорами.

ХІД РОБОТИ

Читання даних акселерометра [ДОДАТОК А]

1. Ініціалізація компонента акселерометра:
   1. У файлі AccelerometerActivity.java спочатку ініціалізуємо SensorManager для доступу до сенсорів.
   2. Використовуємо метод getDefaultSensor(Sensor.TYPE\_ACCELEROMETER), щоб отримати доступ до акселерометра.
   3. Перевіряємо, чи є акселерометр на пристрої. Якщо його немає, виводимо відповідне повідомлення.
2. Обробка даних акселерометра:
   1. Реєструємо слухача подій сенсора за допомогою sensorManager.registerListener().
   2. У методі onSensorChanged() обробляємо отримані дані (значення прискорення по осях X, Y і Z) і відображаємо їх у текстовому полі.
3. Робота з життєвим циклом:
   1. В методі onResume() реєструємо слухача подій сенсора, а в onPause() скидаємо реєстрацію слухача, щоб зберегти ресурси пристрою.

Читання даних з гіроскопа [ДОДАТОК Б]

1. Ініціалізація компонента гіроскопа:
   1. Аналогічно акселерометру, ініціалізуємо SensorManager і отримуємо доступ до гіроскопа через getDefaultSensor(Sensor.TYPE\_GYROSCOPE).
2. Обробка даних гіроскопа:
   1. У методі onSensorChanged() отримуємо значення кутових швидкостей по осях X, Y і Z.
   2. Для візуалізації руху, переміщаємо ImageView на екрані відповідно до отриманих значень.
3. Робота з життєвим циклом:
   1. Як і з акселерометром, слухач подій для гіроскопа реєструється в методі onResume() та скидається в методі onPause().

Порівняння сенсорів [ДОДАТОК В]

1. Інтерфейс вибору сенсора:
   1. У SensorComparisonActivity.java створюємо Spinner для вибору типу сенсора (акселерометр, гіроскоп або магнітометр).
   2. Реєструємо обробник вибору для спінера, який дозволяє перемикати сенсор, що зчитується.
2. Обробка вибору сенсора:
   1. В залежності від вибору сенсора, змінюємо активний сенсор, реєструючи відповідний слухач для нового сенсора.
   2. Виводимо дані обраного сенсора (акселерація для акселерометра, кутові швидкості для гіроскопа, магнітне поле для магнітометра).
3. Робота з життєвим циклом:
   1. Для кожного сенсора при його зміні виконується скидання старого слухача подій та реєстрація нового для обраного сенсора.

ВИСНОВКИ

Під час цієї практичної роботи ми освоїли процес зчитування та обробки даних з сенсорів. Ми створили програму для роботи з сенсорами на платформі Android, яка дозволяє:

* Виводити дані акселерометра.
* Виводити дані гіроскопа.
* Порівнювати дані акселерометра, гіроскопа та магнітометра в реальному часі.

Посилання на відеозвіт завантажений на YouTube. URL: <https://youtu.be/BuG8pEEtuCM>

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Документація AndroidDevelopers для роботи з сенсорами. URL: <https://developer.android.com/guide/topics/sensors> (дата звернення: 25.11.2024)
2. Документація AndroidDevelopers для SensorManager. URL: <https://developer.android.com/reference/android/hardware/SensorManager> (дата звернення: 25.11.2024)
3. Документація AndroidDevelopers для Sensor. URL: <https://developer.android.com/reference/android/hardware/Sensor> (дата звернення: 25.11.2024)

ДОДАТОК А

Код програми AccelerometerActivity.java

package tsyselskyi.andrii.nure;  
  
import android.content.Context;  
import android.hardware.Sensor;  
import android.hardware.SensorEvent;  
import android.hardware.SensorEventListener;  
import android.hardware.SensorManager;  
import android.os.Bundle;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class AccelerometerActivity extends AppCompatActivity {  
  
 private SensorManager sensorManager;  
 private Sensor accelerometer;  
 private TextView accelerometerText;  
  
 private final SensorEventListener accelerometerListener = new SensorEventListener() {  
 @Override  
 public void onSensorChanged(SensorEvent event) {  
 if (event.values != null) {  
 String formattedText = String.*format*("Accelerometer Data:\nX: %.2f\nY: %.2f\nZ: %.2f",  
 event.values[0], event.values[1], event.values[2]);  
 accelerometerText.setText(formattedText);  
 }  
 }  
  
 @Override  
 public void onAccuracyChanged(Sensor sensor, int accuracy) {  
  
 }  
 };  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_accelerometer*);  
  
 initializeComponents();  
 }  
  
 private void initializeComponents() {  
 accelerometerText = findViewById(R.id.*accelerometerText*);  
  
 sensorManager = (SensorManager) getSystemService(Context.*SENSOR\_SERVICE*);  
 if (sensorManager != null) {  
 accelerometer = sensorManager.getDefaultSensor(Sensor.*TYPE\_ACCELEROMETER*);  
 if (accelerometer == null) {  
 accelerometerText.setText("Accelerometer not available on this device.");  
 }  
 }  
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
 if (accelerometer != null) {  
 sensorManager.registerListener(accelerometerListener, accelerometer, SensorManager.*SENSOR\_DELAY\_NORMAL*);  
 }  
 }  
  
 @Override  
 protected void onPause() {  
 super.onPause();  
 if (sensorManager != null) {  
 sensorManager.unregisterListener(accelerometerListener);  
 }  
 }  
}

ДОДАТОК Б

Код програми GyroscopeActivity.java

package tsyselskyi.andrii.nure;  
  
import android.content.Context;  
import android.hardware.Sensor;  
import android.hardware.SensorEvent;  
import android.hardware.SensorEventListener;  
import android.hardware.SensorManager;  
import android.os.Bundle;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class GyroscopeActivity extends AppCompatActivity {  
  
 private SensorManager sensorManager;  
 private Sensor gyroscope;  
 private TextView gyroscopeText;  
 private ImageView imageView;  
  
 private final SensorEventListener gyroscopeListener = new SensorEventListener() {  
 @Override  
 public void onSensorChanged(SensorEvent event) {  
 if (event.values != null) {  
 String formattedText = String.*format*("Gyroscope Data:\nX: %.2f\nY: %.2f\nZ: %.2f",  
 event.values[0], event.values[1], event.values[2]);  
 gyroscopeText.setText(formattedText);  
  
 float x = imageView.getX() + event.values[0] \* 10;  
 float y = imageView.getY() + event.values[1] \* 10;  
 imageView.setX(x);  
 imageView.setY(y);  
 }  
 }  
  
 @Override  
 public void onAccuracyChanged(Sensor sensor, int accuracy) {  
  
 }  
 };  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_gyroscope*);  
  
 initializeComponents();  
 }  
  
 private void initializeComponents() {  
 gyroscopeText = findViewById(R.id.*gyroscopeText*);  
 imageView = findViewById(R.id.*imageView*);  
  
 sensorManager = (SensorManager) getSystemService(Context.*SENSOR\_SERVICE*);  
 if (sensorManager != null) {  
 gyroscope = sensorManager.getDefaultSensor(Sensor.*TYPE\_GYROSCOPE*);  
 if (gyroscope == null) {  
 gyroscopeText.setText("Gyroscope not available on this device.");  
 }  
 }  
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
 if (gyroscope != null) {  
 sensorManager.registerListener(gyroscopeListener, gyroscope, SensorManager.*SENSOR\_DELAY\_NORMAL*);  
 }  
 }  
  
 @Override  
 protected void onPause() {  
 super.onPause();  
 if (sensorManager != null) {  
 sensorManager.unregisterListener(gyroscopeListener);  
 }  
 }  
}

ДОДАТОК В

Код програми SensorComparisonActivity.java

package tsyselskyi.andrii.nure;  
  
import android.content.Context;  
import android.hardware.Sensor;  
import android.hardware.SensorEvent;  
import android.hardware.SensorEventListener;  
import android.hardware.SensorManager;  
import android.os.Bundle;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Spinner;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class SensorComparisonActivity extends AppCompatActivity {  
  
 private SensorManager sensorManager;  
 private Sensor activeSensor;  
 private TextView sensorDataText;  
  
 private final SensorEventListener sensorListener = new SensorEventListener() {  
 @Override  
 public void onSensorChanged(SensorEvent event) {  
 if (event.values != null) {  
 String formattedText = String.*format*("Sensor Data:\nX: %.3f\nY: %.3f\nZ: %.3f",  
 event.values[0], event.values[1], event.values[2]);  
 sensorDataText.setText(formattedText);  
 }  
 }  
  
 @Override  
 public void onAccuracyChanged(Sensor sensor, int accuracy) {  
  
 }  
 };  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_sensor\_comparison*);  
  
 initializeComponents();  
 }  
  
 private void initializeComponents() {  
 sensorDataText = findViewById(R.id.*sensorDataText*);  
 Spinner sensorSelector = findViewById(R.id.*sensorSpinner*);  
  
 sensorManager = (SensorManager) getSystemService(Context.*SENSOR\_SERVICE*);  
  
 String[] sensorsList = {"Accelerometer", "Gyroscope", "Magnetometer"};  
 ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.*simple\_spinner\_item*, sensorsList);  
 adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
 sensorSelector.setAdapter(adapter);  
  
 sensorSelector.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
 @Override  
 public void onItemSelected(AdapterView<?> parent, android.view.View view, int position, long id) {  
 switchSensor(position);  
 }  
  
 @Override  
 public void onNothingSelected(AdapterView<?> parent) {  
 sensorDataText.setText("No sensor selected.");  
 }  
 });  
 }  
  
 private void switchSensor(int index) {  
 if (sensorManager != null) {  
 sensorManager.unregisterListener(sensorListener);  
  
 switch (index) {  
 case 0:  
 activeSensor = sensorManager.getDefaultSensor(Sensor.*TYPE\_ACCELEROMETER*);  
 break;  
 case 1:  
 activeSensor = sensorManager.getDefaultSensor(Sensor.*TYPE\_GYROSCOPE*);  
 break;  
 case 2:  
 activeSensor = sensorManager.getDefaultSensor(Sensor.*TYPE\_MAGNETIC\_FIELD*);  
 break;  
 default:  
 activeSensor = null;  
 }  
  
 if (activeSensor != null) {  
 sensorManager.registerListener(sensorListener, activeSensor, SensorManager.*SENSOR\_DELAY\_NORMAL*);  
 } else {  
 sensorDataText.setText("Sensor not available.");  
 }  
 }  
 }  
  
 @Override  
 protected void onPause() {  
 super.onPause();  
 if (sensorManager != null) {  
 sensorManager.unregisterListener(sensorListener);  
 }  
 }  
}