UNIVERSITI MALAYSIA TERENGGANU

FACULTY OF COMPUTER SCIENCE AND MATHEMATICS

CSM3123 - NATIVE MOBILE PROGRAMMING

BANCHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS

LAB 6

SEMESTER 5 2024/2025

PREPARED FOR:

DR RABIEI B MAMAT

PREPARED BY:

NUR EZREENA SHUHADA BT EMRAN

S66467

Link Github: https://github.com/NurEzreena/CSM3123_LAB-NATIVE-PROGRAMMING.git

Detecting available sensors

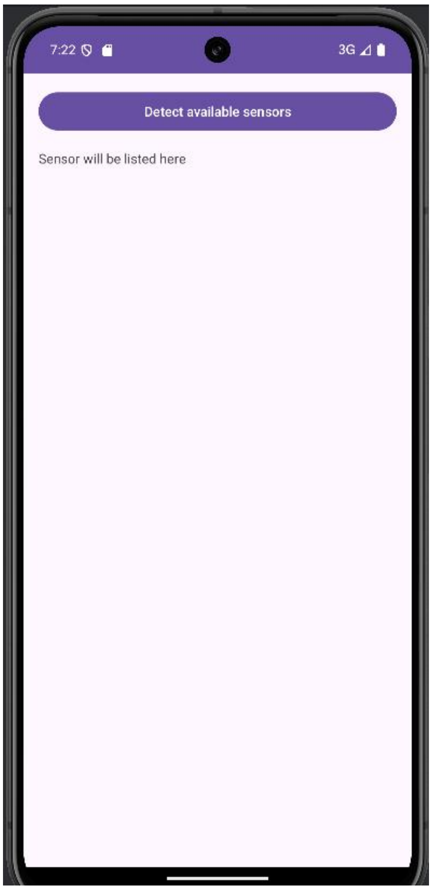
MainActivity.java

```
package com.example.sensorexperimentapp;

import android.hardware.Sensor;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.widget.TextViewa;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;
import java.util.List;

public class MainActivity extends AppCompatActivity {

    private SensorManager sensorManager;
    private TextView sensorListTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button detectSensorsButton = findViewById(R.id.detectSensorsButton);
        sensorListTextView =findViewById(R.id.sensorListTextView);

        sensorManager =(SensorManager) getSystemService(SENSOR_SERVICE);
        detectSensorsButton.setOnClickListener(v-> listAvailableSensors());
    }

    private void listAvailableSensors(){
        List<Sensor>sensorList = sensorManager.getSensorList(Sensor.TYPE_ALL);
        StringBuilder sensorInfo =new StringBuilder("Available sensor: \n");
        for (Sensor sensor :sensorList){
            sensorInfo.append(sensor.getName()).append("(").append(sensor.getType()).append(")\n");
        }
        sensorListTextView.setText(sensorInfo.toString());
    }


}
```

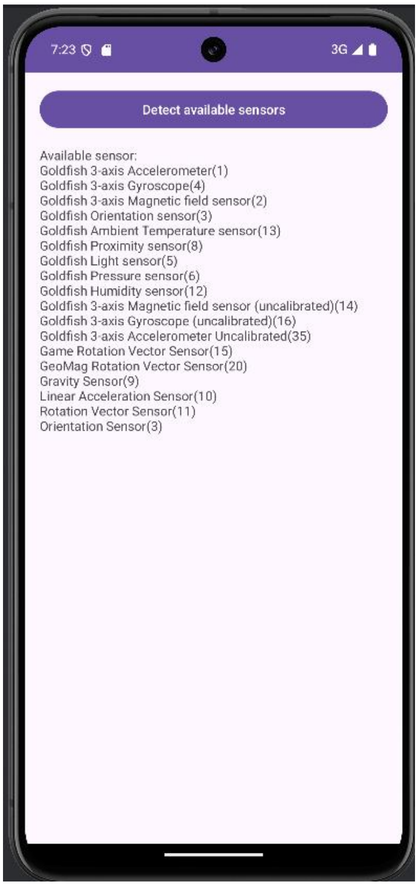
Activity_main.xml

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <Button
        android:id="@+id/detectSensorsButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Detect available sensors" />
    <TextView
        android:id="@+id/sensorListTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Sensor will be listed here"
        android:paddingTop="16dp"/>


</LinearLayout>
```

Output:



```
Sensor will be listed here
```



```
Available sensor:
Goldfish 3-axis Accelerometer(1)
Goldfish 3-axis Gyroscope(4)
Goldfish 3-axis Magnetic field sensor(2)
Goldfish Orientation sensor(3)
Goldfish Ambient Temperature sensor(13)
Goldfish Proximity sensor(8)
Goldfish Light sensor(5)
Goldfish Pressure sensor(6)
Goldfish Humidity sensor(12)
Goldfish 3-axis Magnetic field sensor (uncalibrated)(14)
Goldfish 3-axis Gyroscope (uncalibrated)(16)
Goldfish 3-axis Accelerometer Uncalibrated(35)
Game Rotation Vector Sensor(15)
GeoMag Rotation Vector Sensor(20)
Gravity Sensor(9)
Linear Acceleration Sensor(10)
Rotation Vector Sensor(11)
Orientation Sensor(3)
```

Experimenting with 3 available sensors

MainActivity.java

```
package com.example.sensorexperimentapp;

import static android.hardware.SensorManager.SENSOR_DELAY_NORMAL;

import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorManager;
import android.hardware.SensorEventListener;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity implements SensorEventListener {
   private SensorManager sensorManager;
   private Sensor accelerometer, proximitySensor, lightSensor;

   private TextView accelerometerData, proximityData, lightData;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      // Initialize SensorManager
      sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);

      // Initialize sensors
      accelerometer = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
      proximitySensor = sensorManager.getDefaultSensor(Sensor.TYPE_PROXIMITY);
      lightSensor = sensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);

      // Initialize TextViews
      accelerometerData = findViewById(R.id.accelerometerData);
      proximityData = findViewById(R.id.proximityData);
      lightData = findViewById(R.id.lightData);

      // Register listeners for sensors
      sensorManager.registerListener(this, accelerometer, SENSOR_DELAY_NORMAL);
      sensorManager.registerListener(this, proximitySensor, SENSOR_DELAY_NORMAL);
      sensorManager.registerListener(this, lightSensor, SENSOR_DELAY_NORMAL);
   }

   @Override
   public void onSensorChanged(SensorEvent event) {
      if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
         float x = event.values[0];
```

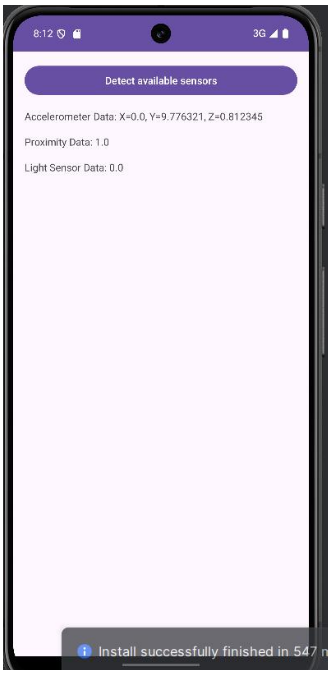
```
            float y = event.values[1];
            float z = event.values[2];
            accelerometerData.setText("Accelerometer Data: X=" + x + ", Y=" + y + ", Z=" + z);
        } else if (event.sensor.getType() == Sensor.TYPE_PROXIMITY) {
            proximityData.setText("Proximity Data: " + event.values[0]);
        } else if (event.sensor.getType() == Sensor.TYPE_LIGHT) {
            lightData.setText("Light Sensor Data: " + event.values[0]);
        }
    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {
        // No action needed for this example
    }
}
```

Activity_main.xml

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <Button
        android:id="@+id/detectSensorsButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Detect available sensors" />
    <TextView
        android:id="@+id/accelerometerData"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Accelerometer data:"
        android:paddingTop="16dp"/>
    <TextView
        android:id="@+id/proximityData"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Proximity data:"
        android:paddingTop="16dp"/>
    <TextView
        android:id="@+id/lightData"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Light sensor data:"
        android:paddingTop="16dp"/>
```

```
</LinearLayout>
```

Output:



Accelerometer Data: X=0.0, Y=9.776321, Z=0.812345

Proximity Data: 1.0

Light Sensor Data: 0.0

Install successfully finished in 547 m

Sensor Fusion Experimentation

MainActivity.java

```
package com.example.sensorexperimentapp;

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.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity implements SensorEventListener {

    private SensorManager sensorManager; // SensorManager declaration
    private Sensor rotationVectorSensor; // Sensor declaration
    private TextView orientationData; // TextView declaration

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);

        // Initialize SensorManager
        sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);

        // Initialize TextView
        orientationData = findViewById(R.id.orientationData);

        // Initialize Rotation Vector Sensor
        rotationVectorSensor = sensorManager.getDefaultSensor(Sensor.TYPE_ROTATION_VECTOR);

        // Register Sensor Listener
        if (rotationVectorSensor != null) {
            sensorManager.registerListener(this, rotationVectorSensor,
SensorManager.SENSOR_DELAY_NORMAL);
        } else {
            orientationData.setText("Rotation Vector Sensor not available");
        }
    }

    @Override
    public void onSensorChanged(SensorEvent event) {
        if (event.sensor.getType() == Sensor.TYPE_ROTATION_VECTOR) {
            float[] rotationMatrix = new float[9];
            SensorManager.getRotationMatrixFromVector(rotationMatrix, event.values);
```

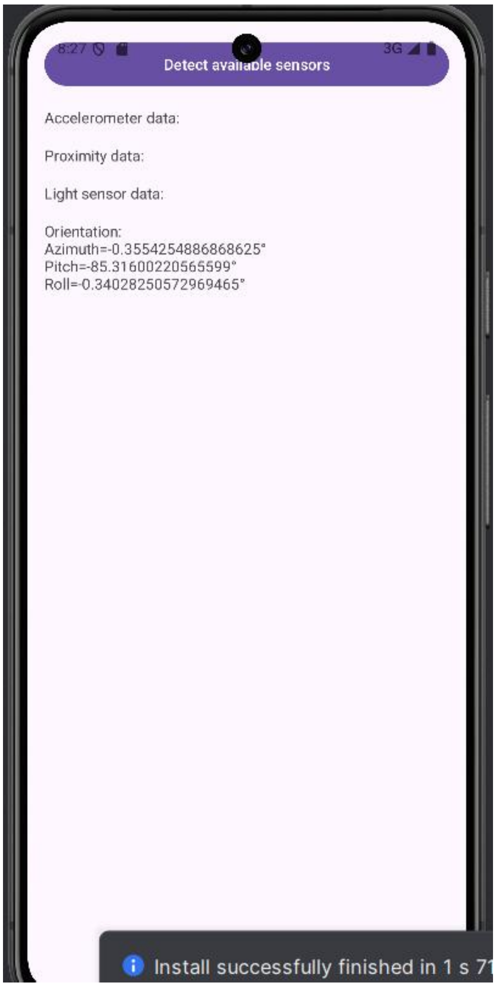
```
         float[] orientation = new float[3];
         SensorManager.getOrientation(rotationMatrix, orientation);

         orientationData.setText("Orientation:\n" +
               "Azimuth=" + Math.toDegrees(orientation[0]) + "°\n" +
               "Pitch=" + Math.toDegrees(orientation[1]) + "°\n" +
               "Roll=" + Math.toDegrees(orientation[2]) + "°");
      }
   }

   @Override
   public void onAccuracyChanged(Sensor sensor, int accuracy) {
      // Not used in this example
   }

   @Override
   protected void onDestroy() {
      super.onDestroy();
      // Unregister listener to avoid memory leaks
      sensorManager.unregisterListener(this);
   }
}
```

Activity_main.xml

```
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_height="match_parent"
   android:layout_width="match_parent"
   android:orientation="vertical"
   android:padding="16dp">

   <Button
      android:id="@+id/detectSensorsButton"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="Detect available sensors" />
   <TextView
      android:id="@+id/accelerometerData"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="Accelerometer data:"
      android:paddingTop="16dp"/>
   <TextView
      android:id="@+id/proximityData"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="Proximity data:"
      android:paddingTop="16dp"/>
   <TextView
```

```
        android:id="@+id/lightData"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Light sensor data:"
        android:paddingTop="16dp"/>

    <TextView
        android:id="@+id/orientationData"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Orientation data:"
        android:paddingTop="16dp"/>



</LinearLayout>
```

Output:



```
Accelerometer data:

Proximity data:

Light sensor data:

Orientation:
Azimuth=-0.3554254886868625°
Pitch=-85.31600220565599°
Roll=-0.34028250572969465°
```

Install successfully finished in 1 s 71