Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

Лабораторная работа №4

по дисциплине "Технологии разработки мобильных приложений"

ТЕМА РАБОТЫ:

Работа с сенсорами на операционной системе Android

Выполнил:

студент гр. ПРИ-120

Парахин К.В.

Принял:

преподаватель кафедры ИСПИ

Рощина А.И.

Владимир 2023 г.

Цель работы:

Изучить работу сенсоров на примере платформы Android и разработать

мобильное приложение, которое будет считывать сигналы сенсоров устройства и представлять их в удобном для восприятия виде.

Задание

Вариант 1

1) Используя акселерометр написать приложение, в котором можно перемещать объект на экране с помощью наклона телефона;

2) Используя магнитный сенсор сделать компас, который бы примерно определял, в каком направлении расположен телефон (север/юг/запад/восток).

Выполнение работы:

Для выполнения задания будем использовать объект типа SensorManager. Из него можно получить сенсор типа акселерометр и сенсор типа магнитный сенсор. Также нужно ввести регистрацию слушателей этих сенсоров по старту приложения и отписку по окончанию.

Ниже показан код класса MainActivity:

public class MainActivity extends AppCompatActivity implements SensorEventListener {  
 TextView mAccelerometerX;  
 TextView mAccelerometerY;  
 TextView mAccelerometerZ;  
 TextView mMagneticX;  
 TextView mMagneticY;  
 TextView mMagneticZ;  
 TextView magnetDirection;  
 TextView accelText;  
 SensorManager sensorManager;  
 Sensor mAccelerometerSensor;  
 Sensor mMagneticSensor;  
 private float[] lastMagnetometerData = new float[3];  
 private float[] lastAccelData = new float[3];  
 private float[] rotationMatrix = new float[9];  
 private float[] orientationAngles = new float[3];  
  
 private float x, y;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 sensorManager = (SensorManager) getSystemService(*SENSOR\_SERVICE*);  
 mAccelerometerSensor = sensorManager.getDefaultSensor(Sensor.*TYPE\_ACCELEROMETER*);  
 mMagneticSensor = sensorManager.getDefaultSensor(Sensor.*TYPE\_MAGNETIC\_FIELD*);  
 accelText = (TextView) findViewById(R.id.*textView1*);  
 mAccelerometerX = (TextView) findViewById(R.id.*textView2*);  
 mAccelerometerY = (TextView) findViewById(R.id.*textView3*);  
 mAccelerometerZ = (TextView) findViewById(R.id.*textView4*);  
 mMagneticX = (TextView) findViewById(R.id.*textView5*);  
 mMagneticY = (TextView) findViewById(R.id.*textView6*);  
 mMagneticZ = (TextView) findViewById(R.id.*textView7*);  
 magnetDirection = (TextView) findViewById(R.id.*textView8*);  
 sensorManager = (SensorManager) getSystemService(*SENSOR\_SERVICE*);  
 mAccelerometerSensor = sensorManager.getDefaultSensor(Sensor.*TYPE\_ACCELEROMETER*);  
 mMagneticSensor = sensorManager.getDefaultSensor(Sensor.*TYPE\_MAGNETIC\_FIELD*);  
 }  
  
 @SuppressLint("Sensor")  
 @Override  
 public void onSensorChanged(SensorEvent event) {  
 if (event.sensor.getType() == Sensor.*TYPE\_ACCELEROMETER*) {  
 System.*arraycopy*(event.values, 0, lastAccelData,  
 0, event.values.length);  
 x = event.values[0];  
 y = event.values[1];  
 accelText.setX(accelText.getX() - (x \* 5));  
 accelText.setY(accelText.getY() + (y \* 5));  
 if (x < 0) {  
 x = 0;  
 }  
 if (y < 0) {  
 y = 0;  
 }  
  
 mAccelerometerX.setText(Float.*toString*(event.values[0]));  
 mAccelerometerY.setText(Float.*toString*(event.values[1]));  
 mAccelerometerZ.setText(Float.*toString*(event.values[2]));  
 }  
 if (event.sensor.getType() == Sensor.*TYPE\_MAGNETIC\_FIELD*) {  
 System.*arraycopy*(event.values, 0, lastMagnetometerData,  
 0, event.values.length);  
 updateOrientationAngles();  
 magnetDirection.setText(getDirectionFromAngles());  
 mMagneticX.setText(Float.*toString*(event.values[0]));  
 mMagneticY.setText(Float.*toString*(event.values[1]));  
 mMagneticZ.setText(Float.*toString*(event.values[2]));  
 }  
 }  
  
 private void updateOrientationAngles() {  
 SensorManager.*getRotationMatrix*(  
 rotationMatrix, null, lastAccelData, lastMagnetometerData);  
 SensorManager.*getOrientation*(rotationMatrix, orientationAngles);  
 for (int i = 0; i < orientationAngles.length; i++) {  
 orientationAngles[i] = (float) Math.*toDegrees*(orientationAngles[i]);  
 }  
 }  
  
 private String getDirectionFromAngles() {  
 float azimuth = orientationAngles[0];  
 if (azimuth >= -22.5 && azimuth < 22.5) {  
 return "North";  
 } else if (azimuth >= 22.5 && azimuth < 67.5) {  
 return "North-East";  
 } else if (azimuth >= 67.5 && azimuth < 112.5) {  
 return "East";  
 } else if (azimuth >= 112.5 && azimuth < 157.5) {  
 return "South-East";  
 } else if (azimuth >= 157.5 || azimuth < -157.5) {  
 return "South";  
 } else if (azimuth >= -157.5 && azimuth < -112.5) {  
 return "South-West";  
 } else if (azimuth >= -112.5 && azimuth < -67.5) {  
 return "West";  
 } else if (azimuth >= -67.5 && azimuth < -22.5) {  
 return "North-West";  
 } else {  
 return "Direction is undefined";  
 }  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
 sensorManager.registerListener(this, mAccelerometerSensor,  
 SensorManager.*SENSOR\_DELAY\_NORMAL*);  
 sensorManager.registerListener(this, mMagneticSensor,  
 SensorManager.*SENSOR\_DELAY\_NORMAL*);  
 }  
  
 @Override  
 protected void onStop() {  
 super.onStop();  
 sensorManager.unregisterListener(this, mAccelerometerSensor);  
 sensorManager.unregisterListener(this, mMagneticSensor);  
 }  
  
 @Override  
 public void onAccuracyChanged(Sensor sensor, int i) {  
  
 }  
}

Ниже показан код разметки активности activity\_main.xml:

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/textView1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="50dp"  
 android:text="@string/accelText"  
 tools:layout\_editor\_absoluteX="170dp"  
 tools:layout\_editor\_absoluteY="84dp"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"/>  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 android:layout\_marginTop="100dp"  
 tools:layout\_editor\_absoluteX="170dp"  
 tools:layout\_editor\_absoluteY="194dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView1"/>  
  
 <TextView  
 android:id="@+id/textView3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 android:layout\_marginTop="50dp"  
 tools:layout\_editor\_absoluteX="170dp"  
 tools:layout\_editor\_absoluteY="236dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView2"/>  
  
 <TextView  
 android:id="@+id/textView4"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 android:layout\_marginTop="50dp"  
 tools:layout\_editor\_absoluteX="168dp"  
 tools:layout\_editor\_absoluteY="272dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView3"/>  
  
 <TextView  
 android:id="@+id/textView5"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 android:layout\_marginTop="100dp"  
 tools:layout\_editor\_absoluteX="170dp"  
 tools:layout\_editor\_absoluteY="378dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView4"/>  
  
 <TextView  
 android:id="@+id/textView6"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 android:layout\_marginTop="50dp"  
 tools:layout\_editor\_absoluteX="170dp"  
 tools:layout\_editor\_absoluteY="427dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView5"/>  
  
 <TextView  
 android:id="@+id/textView7"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 android:layout\_marginTop="50dp"  
 tools:layout\_editor\_absoluteX="170dp"  
 tools:layout\_editor\_absoluteY="474dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView6"/>  
  
 <TextView  
 android:id="@+id/textView8"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 android:layout\_marginTop="100dp"  
 tools:layout\_editor\_absoluteX="168dp"  
 tools:layout\_editor\_absoluteY="586dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView7"/>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Попробуем провести тестирование приложения в эмуляторе (из-за отсутствия устройства Android):

Надпись, положение которой зависит от расположения акселерометра – имеет название «Поплавок» - при запуске на эмуляторе скользит с верха в низ экрана (случайные положения на рис.1 и рис.2).

Надпись направления компаса всегда соответствует значению North (так как по умолчанию, угол равен нулю и его нельзя сместить).

На остальных надписях указаны значения акселерометра по осям x, y, z,

затем значения магнитного сенсора по осям x, y, z, соответственно.

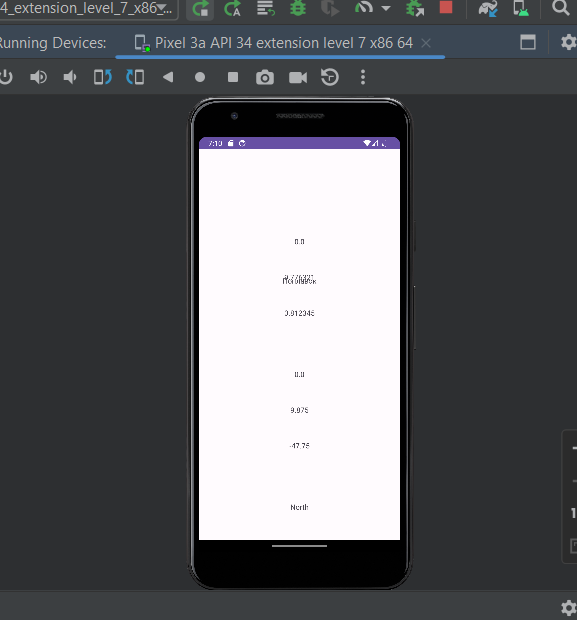


Рисунок 1. Скриншот в момент №1 после запуска

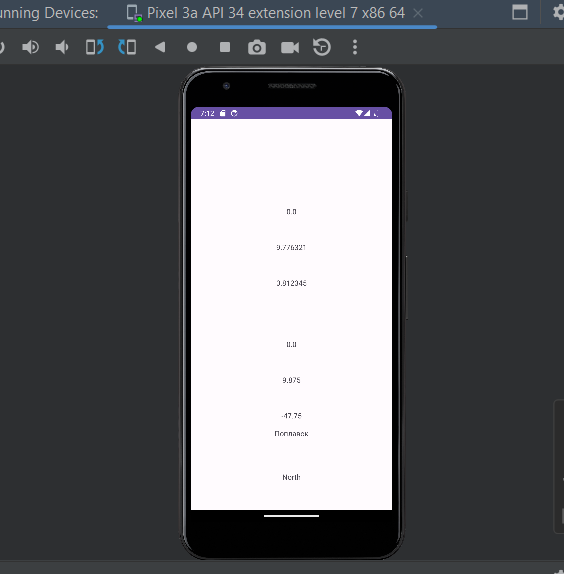


Рисунок 2. Скриншот в момент №2 после запуска, где момент №2 произошел позже момента №1

Вывод

В результате выполнения работы я изучил работу сенсоров на примере платформы Android, а также разработал мобильное приложение, которое будет считывать сигналы сенсоров устройства и представлять их в удобном для восприятия виде.