

感知技术与应用

实 验 报 告

学 院 网安学院
年 级 大二
班 级 物联网工程
学 号 2212005
姓 名 秦泽斌

2024 年 3 月 22 日

目录

一、实验目标.....	1
二、实验内容.....	1
三、实验步骤.....	1
四、实验遇到的问题及其解决方法.....	4
五、实验结论.....	5

一、实验目的

本次实验的目的是让大家了解 Android 中光线传感器的基本知识，掌握 Android 中光线传感器的使用方法。

二、实验内容

(一)、获取设备中光线传感器的值

1、编写布局文件 activity_main.xml。

2、编写程序文件 MainActivity.java。

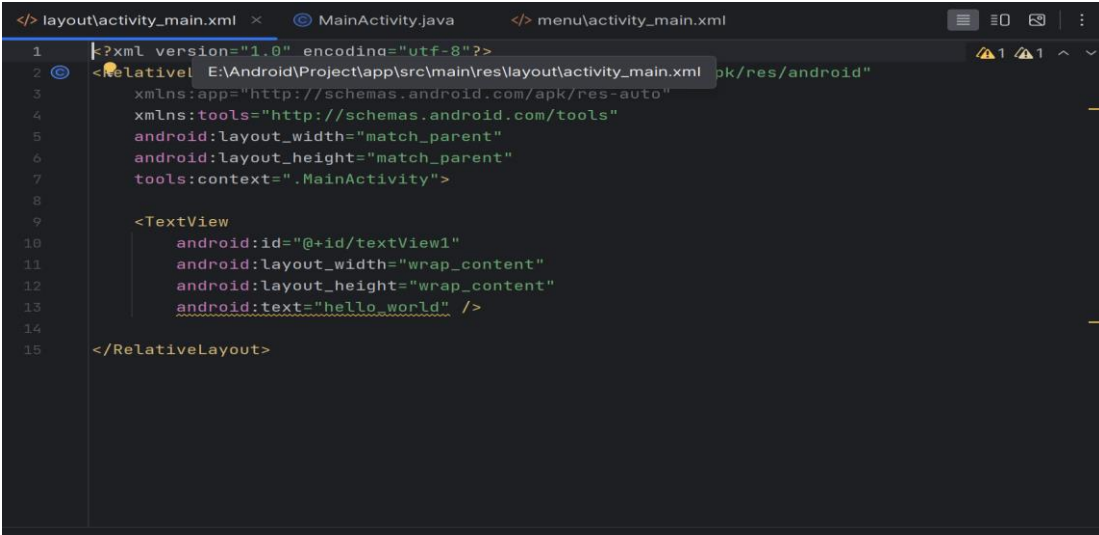
(二)、按照实验的内容进行操作，掌握 Android 中光线传感器的使用方法等内容。

(三)、完成实验内容后，设计一个光线传感器的 app，至少包含如下功能：

- 显示当前光线传感器的光照强度；
- 显示当前光线传感器的名称，耗电量，以及最大测量范围；
- 能自动调整屏幕的亮度。

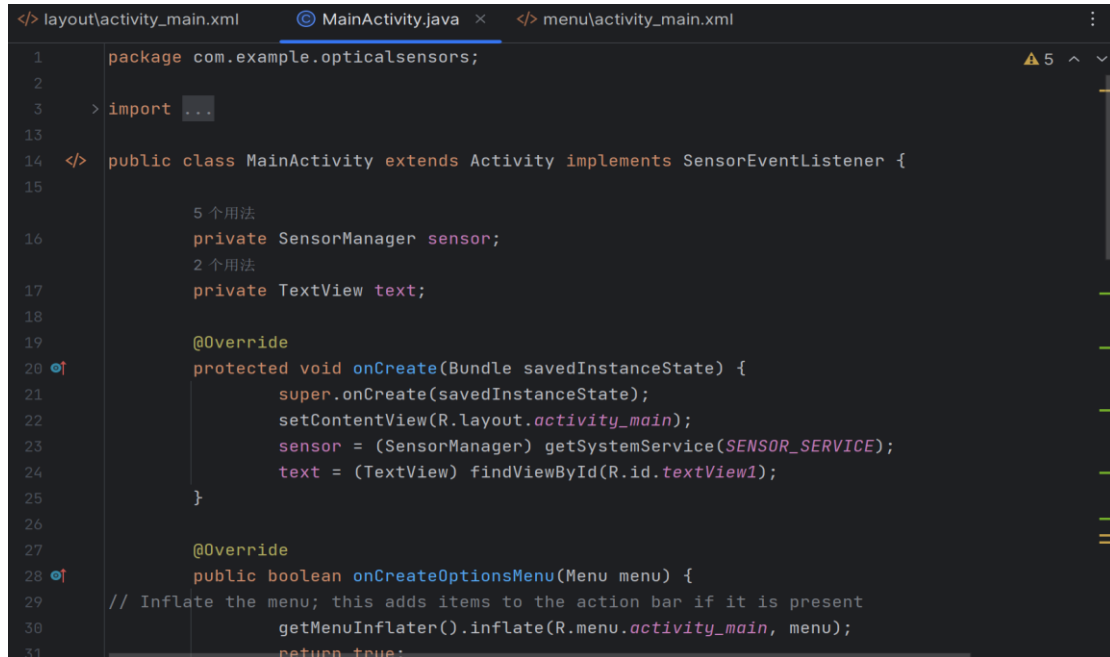
三、实验步骤及实验结果

(一) 编写布局文件 activity_main.xml。



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:text="hello_world" />
</RelativeLayout>
```

(二) 编写程序文件 MainActivity.java。



```
1 package com.example.opticalsensors;
2
3 > import ...
13
14 </> public class MainActivity extends Activity implements SensorEventListener {
15
16     5 个用法
17     private SensorManager sensor;
18     2 个用法
19     private TextView text;
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_main);
25         sensor = (SensorManager) getSystemService(SENSOR_SERVICE);
26         text = (TextView) findViewById(R.id.textView1);
27     }
28
29     @Override
30     public boolean onCreateOptionsMenu(Menu menu) {
31         // Inflate the menu; this adds items to the action bar if it is present
32         getMenuInflater().inflate(R.menu.activity_main, menu);
33         return true;
34     }
35 }
```

具体代码如下：

```
package com.example.opticalsensors;

import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.renderscript.Sampler.Value;
import android.app.Activity;
import android.view.Menu;
import android.widget.TextView;

public class MainActivity extends Activity implements
SensorEventListener {

    private SensorManager sensor;
    private TextView text;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        sensor = (SensorManager) getSystemService(SENSOR_SERVICE);
        text = (TextView) findViewById(R.id.textView1);
    }
}
```

```

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it is present
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }

    @Override
    protected void onPause() {
// TODO Auto-generated method stub
        sensor.unregisterListener(this);
        super.onPause();
    }

    @Override
    protected void onResume() {
        // TODO Auto-generated method stub
        sensor.registerListener(this,
sensor.getDefaultSensor(Sensor.TYPE_LIGHT),
        SensorManager.SENSOR_DELAY_GAME);
        super.onResume();
    }

    @Override
    protected void onStop() {
        // TODO Auto-generated method stub
        sensor.unregisterListener(this);
        super.onStop();
    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onSensorChanged(SensorEvent event) {
        // TODO Auto-generated method stub
        float[] values = event.values;
        int sensorType = event.sensor.TYPE_LIGHT;

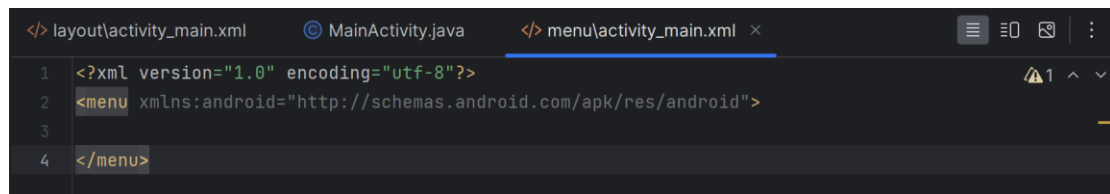
```

```

        if (sensorType == Sensor.TYPE_LIGHT) {
            String light_intensity = "LightIntensity: " +
String.valueOf(values[0]) + '\n';
            light_intensity += "SensorName: " +
event.sensor.getName() + '\n';
            light_intensity += "PowerDraw: " +
event.sensor.getPower() + '\n';
            light_intensity += "MaximumRange: " +
event.sensor.getMaximumRange() + '\n';
            text.setText(light_intensity);
        }
    }
}

```

（三）编写 menu 文件中的 activity_main.xml 文件



（四）运行程序并检查结果

程序运行结果如下：



经过检查，实验结果符合要求。

四、实验遇到的问题及其解决方法

在本次实验中，本人遇到了不少的困难，但都一一解决，具体问题如下：

问题一： activity_main.xml 文件无法进入输入代码的界面，只有可视化设计界面。

解决办法：通过查询 Android Studio UI 界面指导，找了 design 和 code 的切换按键，如下图所示



问题二：从实验指导文档中复制的代码出现大量报错，基本每行都有报错。

解决办法：上述现象出现的原因是从 word 文档复制过来的代码中带有中文字符中的空格，而 Android Studio 无法识别这种编码的空格。只需将代码中所有的空格删除并填充为英文空格即可

问题三：MainActivity.Java 文件中出现“无法解析字符 textView1”的报错。

解决办法：layout 文件中的 activity_main.xml 文件代码仍然存在问题。再采用实验离线包 Experiment1 中的 layout\activity_main.xml 中的代码替换，报错提示消失。

问题四：MainActivity.Java 文件中出现“无法解析字符 activity_mian”的报错。

解决办法：通过观察代码，发现此 activity_mian 并不是上一个问题中 layout 文件中的布局文件，而是 menu 文件中的 activity_mian.xml 文件。对比实验离线包 Experiment1 和本工程，发现本工程中缺少 menu 文件，故在 res 文件目录下新建一个 menu 文件夹，并在其目录下新建 activity_main.xml 文件，填充代码。至此，所有报错消失，成功运行程序。

五、实验结论

通过本次实验，本人学习到了很多，首先了解了 Android 中光线传感器的基本知识，掌握了 Android 中光线传感器的使用方法，其次我也熟悉了解了 Android Studio 的基本使用方法，并掌握了一部分的语法及使用技巧。另外，我也锻炼了我的个人排错能力，做到了独立地发现并解决问题，最终完成实验。