

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 5 月 10 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.....	7
五、实验结论.....	7

一、实验目的

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

二、实验内容

（一）指南针设计

本实验将演示在 Android 中使用方向传感器开发指南针应用程序的方法。在本实例中首先准备一张指南针素材图片，该图片上方向指南针指向北方。接下来开发一个检测方向的传感器，传感器程序可以检测到手机顶部绕 Z 转过的多少度。在实例中需添加了一张图片，并让图片总是反转方向传感器返回的第一个角度值。



（二）通过方向值判断用户拿着手机是否一直处于直线行走状态。

三、实验步骤及实验结果

（一）实验步骤

1、编写布局文件

编写布局文件 main.xml，功能是插入准备好的素材图片，主要实现代码如下所示。

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    android:background="#fff"
    >

    <TextView
        android:id="@+id/myTextView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="myTextView1"
        android:textSize="20sp" />

    <TextView
        android:id="@+id/myTextView2"
        android:textSize="20sp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="myTextView2"/>

    <TextView
        android:id="@+id/myTextView3"
        android:textSize="20sp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="myTextView3"/>

    <ImageView
        android:id="@+id/znzImage"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/a"
        android:scaleType="centerInside"
        android:layout_gravity="center"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="10dp"/>
```

```

<TextView
    android:id="@+id/myTextView4"
    android:textSize="30sp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="myTextView4"/>
</LinearLayout>

```

2、编写程序文件

编写程序文件，使用传感器获取设备的旋转角度值，并根据这个值返回指南针的角度。

```

package com.example.orientationSensor;

import android.app.Activity;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.view.animation.Animation;
import android.view.animation.RotateAnimation;
import android.widget.ImageView;
import android.widget.TextView;

public class MainActivity extends Activity implements
    SensorEventListener{

    ImageView image; //指南针图片
    float currentDegree = 0f; //指南针图片转过的角度

    TextView myTextView1;//x 方向的方向值
    TextView myTextView2;//y 方向的方向值
    TextView myTextView3;//z 方向的方向值
    TextView myTextView4;//是否直线运动

    SensorManager mSensorManager; //管理器

    Sensor mAccelerometer;
    float[] currentAcceleration = new float[3];
    float[] currentVelocity = new float[3];
    long lastTimestamp = 0;

```

```

float totalSpeed = 0.0f;
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    myTextView1 = (TextView) findViewById(R.id.myTextView1); //得到
myTextView1 的引用
    myTextView2 = (TextView) findViewById(R.id.myTextView2); //得到
myTextView2 的引用
    myTextView3 = (TextView) findViewById(R.id.myTextView3); //得到
myTextView3 的引用
    myTextView4 = (TextView) findViewById(R.id.myTextView4); //得到
myTextView4 的引用
    image = (ImageView) findViewById(R.id.znzImage);
    mSensorManager =
(SensorManager) getSystemService(SENSOR_SERVICE); //获取管理服务
    mAccelerometer =
mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
}

@Override
protected void onResume() {
    super.onResume();
    //注册监听器
    mSensorManager.registerListener(this
        ,
mSensorManager.getDefaultSensor(Sensor.TYPE_ORIENTATION),
SensorManager.SENSOR_DELAY_GAME);
    mSensorManager.registerListener(this, mAccelerometer,
SensorManager.SENSOR_DELAY_GAME);
}
//取消注册
@Override
protected void onPause() {
    mSensorManager.unregisterListener(this);
    super.onPause();
}
@Override
protected void onStop() {
    mSensorManager.unregisterListener(this);
    super.onStop();
}
//传感器值改变

```

```

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {
    // TODO Auto-generated method stub
}
//精度改变
@Override
public void onSensorChanged(SensorEvent event) {
    // TODO Auto-generated method stub
    //添加三个方向的加速度
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
        float accelerationX = event.values[0]; //x 方向
        float accelerationY = event.values[1]; //y 方向
        float accelerationZ = event.values[2]; //z 方向
        long currentTimestamp = System.currentTimeMillis();
        float totalAcceleration = (float) Math.sqrt(accelerationX *
accelerationX
                + accelerationY * accelerationY + accelerationZ *
accelerationZ);
        if (totalAcceleration < 10.0 && totalAcceleration > 9.5) {
            totalSpeed = 0; //如果总加速度小于 10.0 且大于 9.5，则将总速度设
置为 0
        }
        else if (totalAcceleration >= 10.0 || totalAcceleration <= 9.5) {
            if (lastTimestamp != 0) {
                float deltaTime = (currentTimestamp - lastTimestamp) *
0.001f;

                //如果上一个时间戳不为 0，则计算时间差并将其转换为秒

                // 更新速度
                currentVelocity[0] = accelerationX * deltaTime;
                currentVelocity[1] = accelerationY * deltaTime;
                currentVelocity[2] = accelerationZ * deltaTime;
            }
            totalSpeed = (float)
Math.sqrt(Math.pow(currentVelocity[0], 2) +
                Math.pow(currentVelocity[1], 2) +
Math.pow(currentVelocity[2], 2));
            //计算总速度
            lastTimestamp = currentTimestamp;
        }
    }
    //方向
    if (event.sensor.getType() == Sensor.TYPE_ORIENTATION) {
        float threshold = 2.0f; //设置阈值
    }
}

```

```

        float degree1 = event.values[0]; //获取 z 转过的角度
        float degree2 = event.values[1]; //获取 y 转过的角度
        float degree3 = event.values[2]; //获取 x 转过的角度

        //截断角度到小数点后三位
        float degreeX = (int)(degree3 * 1000.0f) / 1000.0f;
        float degreeY = (int)(degree2 * 1000.0f) / 1000.0f;
        float degreeZ = (int)(degree1 * 1000.0f) / 1000.0f;

        myTextView1.setText("x 方向上的方向值为: " + degreeX);
        myTextView2.setText("y 方向上的方向值为: " + degreeY);
        myTextView3.setText("z 方向上的方向值为: " + degreeZ);

        //如果速度不为零的情况
        if(totalSpeed != 0){
            if(Math.abs(degreeX) > threshold && Math.abs(degreeY) >
threshold){
                myTextView4.setText("当前未处于直线运动状态\n 速度为: " +
totalSpeed);
            }
            else{
                myTextView4.setText("当前处于直线运动状态\n 速度为: " +
totalSpeed);
            }
        }
        //如果速度为零
        else{
            myTextView4.setText("当前处于静止状态");
        }

        //穿件旋转动画
        RotateAnimation ra = new
RotateAnimation(currentDegree,-degree1,Animation.RELATIVE_TO_SELF,0.5
f
            ,Animation.RELATIVE_TO_SELF,0.5f);
        ra.setDuration(100); //动画持续时间
        image.startAnimation(ra);
        currentDegree = -degree1;
    }
}
}

```


（二）实验结果

9:10 9:10 75%
x方向上的方向值为: 1.078
y方向上的方向值为: -11.796
z方向上的方向值为: 246.078



当前处于直线运动状态
速度为: 0.1832311

9:10 9:10 75%
x方向上的方向值为: 10.375
y方向上的方向值为: -13.234
z方向上的方向值为: 190.343



当前处于静止状态

四、实验遇到的问题及其解决方法

在本次实验中,本人遇到了一点点的小困难,包括处理图片素材导入的问题等,根据 IDE 给出的问题报错,我找到对应的文件夹并将图片正确导入。另外,在设计检测是否处于直线运动时,在算法层面曾一度陷入困难,后参考网络上的办法,完成了算法的设计。

五、实验结论

通过本次实验,我了解了 Android 中方向传感器的基本知识,掌握了 Android 中方向传感器的使用方法,熟悉了 Android Studio 的使用,对软件开发有了初步的了解。