**Android传感器专题**

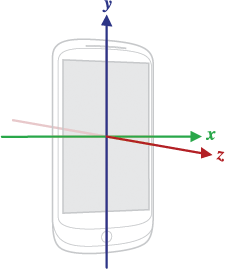
转载 2016年02月29日 10:47:03

* 794

1.方向传感器

1）.三维坐标系的概念：

在Android平台中，传感器框架通常是使用一个标准的三维坐标系来表示一个值的。以本节要讲的方向传感器为例子，确定一个方向也需要一个三维坐标，毕竟我们的设备不可能永远都是水平端着的吧，安卓给我们返回的方向值就是一个长度为3的flaot数组，包含三个方向的值！官方API文档中有这样一个图：[sensors\_overview](http://androiddoc.qiniudn.com/guide/topics/sensors/sensors_overview.html)



如果你看不懂图，那么写下文字解释：

* **X轴的方向**：沿着屏幕水平方向从左到右，如果手机如果不是是正方形的话，较短的边需要水平放置，较长的边需要垂直放置。
* **Y轴的方向**：从屏幕的左下角开始沿着屏幕的的垂直方向指向屏幕的顶端
* **Z轴的方向**：当水平放置时，指向天空的方向

2）.方向传感器的三个值

上一节中说了，传感器的回调方法：onSensorChanged中的参数SensorEvent event，event的值类型是Float[]的，而且最多只有三个元素，而方向传感器则刚好有三个元素，都表示度数！对应的含义如下：

**values[0]：**方位角，手机绕着Z轴旋转的角度。0表示正北(North)，90表示正东(East)，180表示正南(South)，270表示正西(West)。假如values[0]的值刚好是这四个值的话，并且手机沿水平放置的话，那么当前手机的正前方就是这四个方向，可以利用这一点来写一个指南针！

**values[1**]：倾斜角，手机翘起来的程度，当手机绕着x轴倾斜时该值会发生变化。取值范围是[-180,180]之间。假如把手机放在桌面上，而桌面是完全水平的话，values[1](http://androiddoc.qiniudn.com/guide/topics/sensors/sensors_overview.html)的则应该是0，当然很少桌子是绝对水平的。从手机**顶部开始抬起**，直到手机沿着x轴旋转180(此时屏幕乡下水平放在桌面上)。在这个旋转过程中，values[**1**]的值会从**0到-180**之间变化，即手机抬起时，values[1](http://androiddoc.qiniudn.com/guide/topics/sensors/sensors_overview.html)的值会逐渐变小，知道等于-180；而加入从手机**底部开始抬起**，直到手机沿着x轴旋转180度，此时values[**1**]的值会**从0到180**之间变化。我们可以利用value[**1**]的这个特性结合value[**2**]来实现一个平地尺！

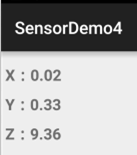
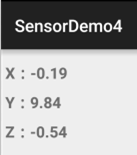
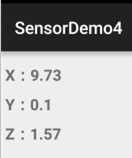
**value[2**]：滚动角，沿着Y轴的滚动角度，取值范围为：[-90,90]，假设将手机屏幕朝上水平放在桌面上，这时如果桌面是平的，values[2](http://www.runoob.com/wp-content/uploads/2015/11/18139494.jpg)的值应为0。将手机从左侧逐渐抬起，values[**2**]的值将逐渐减小，知道垂直于手机放置，此时values[**2**]的值为-90，从右侧则是0-90；加入在垂直位置时继续向右或者向左滚动，values[**2**]的值将会继续在-90到90之间变化！

2.加速度传感器(Accelerometer sensor)

**1）名词概念：**

* 加速度传感器的**单位**：**加速度(m/s^2)**
* 方向传感器获取到的加速度是：**手机运动的加速度与重力加速度(9.81m/s^2)的合加速度**
* 另外重力加速度是**垂直向下**的！

关于这个不同方向合加速度的计算好像蛮复杂的，这里我们就不去纠结这个了！先来看看加速度的value数组中的三个数的值吧~依旧是上节的代码，改下传感器而已~

**水平放置**：**竖直平放**：**竖直横放**：

从上面我们知道value数组的三个值分别对应X，Y，Z轴上的加速度！好的，知道个大概，我们来写个简易计步器来熟悉下用法！

**2）.简易计步器的实现**

**运行效果图**：



**代码实现**：

布局代码：**activity\_main.xml**：

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="5dp">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="30dp"

android:text="简易计步器"

android:textSize="25sp" />

<TextView

android:id="@+id/tv\_step"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="5dp"

android:text="0"

android:textColor="#DE5347"

android:textSize="100sp"

android:textStyle="bold" />

<Button

android:id="@+id/btn\_start"

android:layout\_width="match\_parent"

android:layout\_height="64dp"

android:text="开始"

android:textSize="25sp" />

</LinearLayout>

**MainActivity.java**：

public class MainActivity extends AppCompatActivity implements View.OnClickListener, SensorEventListener {

private SensorManager sManager;

private Sensor mSensorAccelerometer;

private TextView tv\_step;

private Button btn\_start;

private int step = 0; //步数

private double oriValue = 0; //原始值

private double lstValue = 0; //上次的值

private double curValue = 0; //当前值

private boolean motiveState = true; //是否处于运动状态

private boolean processState = false; //标记当前是否已经在计步

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

sManager = (SensorManager) getSystemService(SENSOR\_SERVICE);

mSensorAccelerometer = sManager.getDefaultSensor(Sensor.TYPE\_ACCELEROMETER);

sManager.registerListener(this, mSensorAccelerometer, SensorManager.SENSOR\_DELAY\_UI);

bindViews();

}

private void bindViews() {

tv\_step = (TextView) findViewById(R.id.tv\_step);

btn\_start = (Button) findViewById(R.id.btn\_start);

btn\_start.setOnClickListener(this);

}

@Override

public void onSensorChanged(SensorEvent event) {

double range = 1; //设定一个精度范围

float[] value = event.values;

curValue = magnitude(value[0], value[1], value[2]); //计算当前的模

//向上加速的状态

if (motiveState == true) {

if (curValue >= lstValue) lstValue = curValue;

else {

//检测到一次峰值

if (Math.abs(curValue - lstValue) > range) {

oriValue = curValue;

motiveState = false;

}

}

}

//向下加速的状态

if (motiveState == false) {

if (curValue <= lstValue) lstValue = curValue;

else {

if (Math.abs(curValue - lstValue) > range) {

//检测到一次峰值

oriValue = curValue;

if (processState == true) {

step++; //步数 + 1

if (processState == true) {

tv\_step.setText(step + ""); //读数更新

}

}

motiveState = true;

}

}

}

}

@Override

public void onAccuracyChanged(Sensor sensor, int accuracy) {}

@Override

public void onClick(View v) {

step = 0;

tv\_step.setText("0");

if (processState == true) {

btn\_start.setText("开始");

processState = false;

} else {

btn\_start.setText("停止");

processState = true;

}

}

//向量求模

public double magnitude(float x, float y, float z) {

double magnitude = 0;

magnitude = Math.sqrt(x \* x + y \* y + z \* z);

return magnitude;

}

@Override

protected void onDestroy() {

super.onDestroy();

sManager.unregisterListener(this);

}

}

好的，真的是非常简易的计步器...上面的步数是我坐着拿手撸出来的...http://www.runoob.com/wp-content/uploads/2015/11/93465440.jpg，毕竟写来玩玩而已~

3.陀螺仪传感器(Gyroscope sensor)

**1）名词概念：**

陀螺仪又叫角速度传感器，一般用来检测手机姿态的，好像手机中的陀螺仪传感器一般都是三轴的！体感游戏用得最多，手机拍照防抖，GPS惯性导航，还有为APP添加一些动作感应(比如轻轻晃动手机关闭来电铃声)等等，具体的可以自己去百度下~

* 陀螺仪传感器的**单位**：**角速度(弧度/秒)radians/second**
* 获得传感器用的是：**Sensor.TYPE\_GYROSCOPE**

他的三个值依次是沿着X轴，Y轴，Z轴旋转的角速度，手机逆时针旋转，角速度值为正，顺时针则为负值！经常用于计算手机已经转动的角度！这是网上的一段代码~

private static final float NS2S = 1.0f / 1000000000.0f;

private float timestamp;

public void onSensorChanged(SensorEvent event)

{

if (timestamp != 0)

{

// event.timesamp表示当前的时间，单位是纳秒（1百万分之一毫秒）

final float dT = (event.timestamp - timestamp) \* NS2S;

angle[0] += event.values[0] \* dT;

angle[1] += event.values[1] \* dT;

angle[2] += event.values[2] \* dT;

}

timestamp = event.timestamp;

}

通过陀螺仪传感器相邻两次获得数据的时间差（dT）来分别计算在这段时间内手机延X、 Y、Z轴旋转的角度，并将值分别累加到angle数组的不同元素上

4.磁场传感器(Magnetic field sensor)

**作用**：该传感器主要用于读取手机附近的磁场变化

**传感器的值采集到的值**：有三个，分别是：X，Y，Z轴上方向上的磁场值

**数值单位**：T，微特斯拉

**传感器获取**：Sensor.**TYPE\_MAGNETIC\_FIELD**

5.距离传感器(Proximity sensor)

**作用**：用于感应手机与人体的距离，用得最多的就是手机通话时候，脸部贴近屏幕时，屏幕会熄灭，当脸部离开屏幕一段距离后，屏幕又会亮起，这样可以避免通过过程脸部误碰挂断按钮，从而导致通话中断~我们可以利用这个传感器来做一些交互型的App~

**传感器的值采集到的值**：有一个，物体与设备间的距离

**数值单位**：cm，厘米

**传感器获取**：Sensor.**TYPE\_PROXIMITY**

**其他**：

* ①关于距离传感器可能有两种，一种是能直接给出距离的，而另一种则是给出靠近或者远离！就是只返回两个值，0.0或者最大值！我们可以通过对比解析度和最大值是否相等进行判断！假如相等说明是后者，假如不等说明是前者！
* ②调用sensor.getResolution()方法获得解析度，调用getMaximumRange()获得最大值！

6.光线传感器(Light sensor)

**作用**：用来读取光度值，即光线强度

**传感器的值采集到的值**：有一个，光亮度值

**数值单位**：lux，1流明每平方米面积，就是1勒克斯(lux)，最大值是：120000.0f，Android中把光线强度分了不同的等级，可以自行查看SensorManager类~

**传感器获取**：Sensor.**TYPE\_LIGHT**

7.气压传感器(Pressure sensor)

**作用**：用于测量大气压力，常用于测量海拔高度

**传感器的值采集到的值**：有一个，大气压值

**数值单位**：hPa，百帕

**传感器获取**：Sensor.**TYPE\_PRESSURE**

8.温度传感器（Temperature sensor）

**作用**：测量手机内部的温度或者外部环境的问题

**传感器的值采集到的值**：有一个，温度值

**数值单位**：℃，摄氏度

**传感器获取**：Sensor.**TYPE\_TEMPERATURE**(手机内部)/**TYPE\_AMBIENT\_TEMPERATURE**(手机外部)

9.传感器模拟工具——SensorSimulator

如题，当我们的真机不具备某种传感器的时候，而又需要进行开发~关于具体用法可见下面的文章：[Android设备上的传感器模拟工具：SensorSimulator](http://www.cnblogs.com/mengdd/archive/2013/05/18/3085703.html)

文章出处：http://www.runoob.com/w3cnote/android-tutorial-sensor4.html