

# 实验十：传感器应用及位置服务

13331388 庄梓嘉

## 实验目的

- 1 学会使用 Android 手机不同的传感器服务
- 2 学会使用 Android 位置服务获得经纬度信息

## 实验内容

- 1 移动手机方向,图片能够实现指南针功能(调用地磁传感器和方向传感器)。
- 2 点击按钮,能够通过联网获得经纬度信息。

### 【拓展项】

思考如何使用加速传感器实现摇一摇功能。

## 实验步骤

(1) 设计页面,指南针由其中两张图片构成,指针和罗盘分开,用 RelativeLayout 重叠。

```
<RelativeLayout
    android:layout_width="250dp"
    android:layout_height="250dp"
    android:layout_gravity="center">
    <ImageView
        android:layout_width="250dp"
        android:layout_height="250dp"
        android:paddingTop="32dp"
        android:paddingBottom="32dp"
        android:id="@+id/compass_img"
        android:src="@drawable/compass"/>
    <ImageView
        android:layout_width="60dp"
        android:layout_height="110dp"
        android:layout_centerInParent="true"
        android:id="@+id/arrow_img"
        android:src="@drawable/arrow"/>
</RelativeLayout>
```

(2) 设置传感器监听器，通过加速度传感器和地磁传感器共同计算获取手机旋转方向。通过 `event.sensor.getType()` 函数来获得不同传感器的类型，然后定义旋转矩阵和重力坐标数组，`getRotationMatrix()` 方法计算出旋转数据，然后赋值到 R 数组之中，然后我们调用 `GetOrientation()` 将 R 数组转换成为空间坐标旋转数据，然后将计算出的旋转角度取反，最后使用动画旋转指南针背景图，实现方向指示功能。

```
sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
Sensor magneticSensor = sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
Sensor accelerometerSensor = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
sensorManager.registerListener(listener, magneticSensor, SensorManager.SENSOR_DELAY_GAME);
sensorManager.registerListener(listener, accelerometerSensor, SensorManager.SENSOR_DELAY_GAME);

//传感器监听
private SensorEventListener listener = new SensorEventListener() {
    float[] accelerometerValues = new float[3];
    float[] magneticValues = new float[3];
    private float lastRotateDegree;
    @Override
    //通过加速度传感器和地磁传感器共同计算获取手机旋转方向
    public void onSensorChanged(SensorEvent event) {
        if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
            accelerometerValues = event.values.clone();
        } else if (event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD) {
            magneticValues = event.values.clone();
        }
        float[] R = new float[9];
        float[] values = new float[3];
        SensorManager.getRotationMatrix(R, null, accelerometerValues, magneticValues);
        SensorManager.getOrientation(R, values);
        float rotateDegree = -(float) Math.toDegrees(values[0]);
        if (Math.abs(rotateDegree - lastRotateDegree) > 1) {
            RotateAnimation animation = new RotateAnimation(lastRotateDegree, rotateDegree,
                Animation.RELATIVE_TO_SELF, 0.5f, Animation.RELATIVE_TO_SELF, 0.5f);
            animation.setFillAfter(true);
            compass.startAnimation(animation);
            lastRotateDegree = rotateDegree;
        }
    }
    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {}
};
```

(3) 设置位置管理器，获取经纬度信息。

```
locationManager = (LocationManager) getSystemService(Context. LOCATION_SERVICE);
List<String> providerList = locationManager.getProviders(true);
if (providerList.contains(LocationManager. GPS_PROVIDER)) {
    provider = LocationManager. GPS_PROVIDER;
} else if (providerList.contains(LocationManager. NETWORK_PROVIDER)) {
    provider = LocationManager. NETWORK_PROVIDER;
} else {
    Toast.makeText(this, "No location provider to use", Toast. LENGTH_SHORT).show();
    return;
}
```

(4) 更新位置变化，并显示到界面中。

```
private LocationListener locationListener = new LocationListener() {
    @Override
    public void onLocationChanged(Location location) {showLocation(location);}
    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {}
    @Override
    public void onProviderEnabled(String provider) {}
    @Override
    public void onProviderDisabled(String provider) {}
};

private void showLocation(Location location) {
    longitude.setText("" + location.getLongitude());
    latitude.setText("" + location.getLatitude());
}
```

(5) 按钮监听事件。

```
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Location loc = locationManager.getLastKnownLocation(provider);
        if (loc != null) {
            showLocation(loc);
        }
        locationManager.requestLocationUpdates(provider, 5000, 1, locationListener);
    }
});
```

(6) 退出时移除监听器。

```
@Override
public void onDestroy() {
    super.onDestroy();
    if (sensorManager != null) {
        sensorManager.unregisterListener(listener);
    }
    if (locationListener != null) {
        locationManager.removeUpdates(locationListener);
    }
}
```

### 【拓展项】

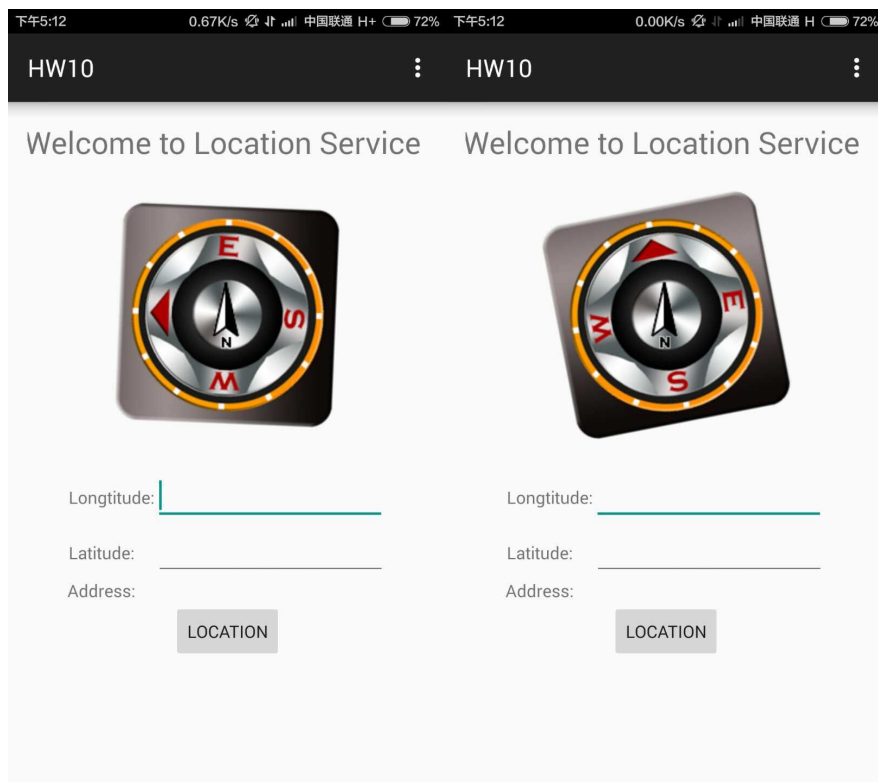
摇一摇思路：

通过设置加速度传感器，计算获取手机旋转角度，设定一个初始值，当大于该值时，判定手机摇动了，跳转到相应界面。

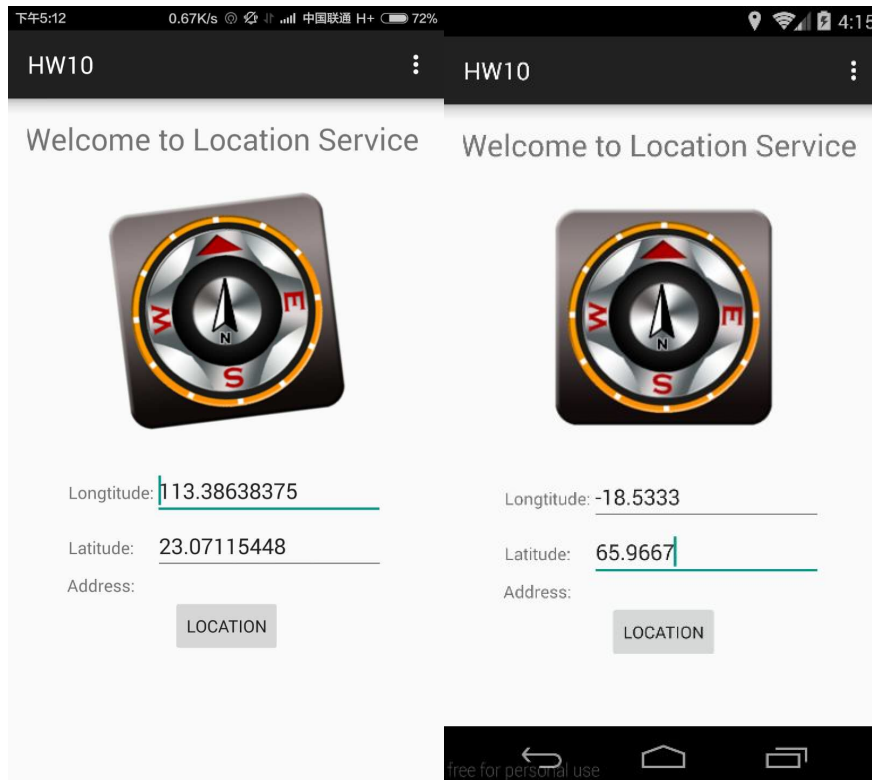
## 实验结果

初始页面

移动手机



真机：点击按钮获取经纬度      虚拟机：点击按钮获取经纬度



## 参考资料

Android 实现摇一摇功能：

<http://www.2cto.com/kf/201408/328408.html>

Android 手机加速度传感器和方向传感器的应用：

[http://blog.sina.com.cn/s/blog\\_45e6be0801013mzm.html](http://blog.sina.com.cn/s/blog_45e6be0801013mzm.html)

Android 使用基于位置的服务（一）：

<http://blog.csdn.net/qiannuo/article/details/6791873>

## 实验总结

这次实验要求使用传感器和位置服务，基本代码按照实验文档就能实现了所需功能，手机上传感器还有很多种，这次只是接触到两种传感器，还需要学习一下其他传感器的使用。在虚拟机上运行时，获取的经纬度跟真机上的不一样，可

能是虚拟机上定位在其他地方上。还有就是部分代码有红线警告，查询后得知，是需要手机上运行时允许其访问位置服务，所以不影响使用。在虚拟机上运行的话就要把 GPS 开关打开才可以获取到经纬度。本次实验还是比较简单易懂的。