

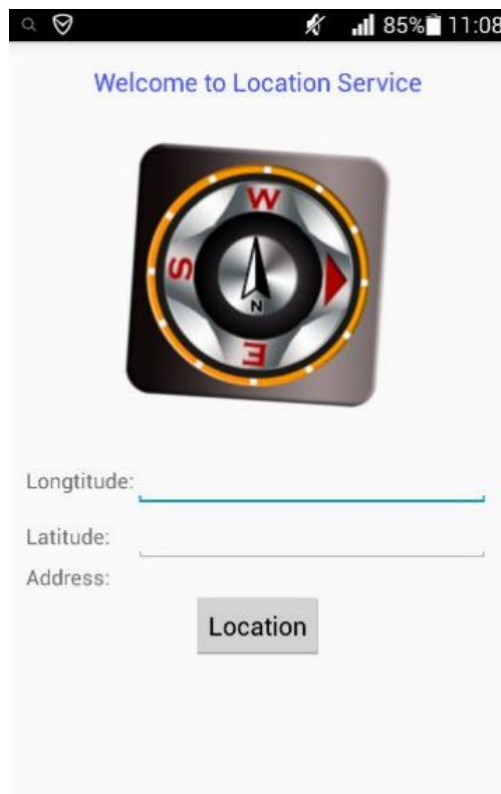
# 传感器应用及位置服务

## 【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) 实验界面如图：



其中，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：

## Welcome to Location Service



Longitude:

Latitude:

Address:

Location

## 【参考内容】

(1) 指南针由两张图片构成，可以使用 RelativeLayout 实现重叠的功能，当然也可以用 FrameLayout

```
<RelativeLayout
    android:layout_height="250dp"
    android:layout_width="250dp"
    android:layout_gravity="center">

    <ImageView
        android:paddingTop="32dp"
        android:paddingBottom="32dp"
        android:id="@+id/compass_img"
        android:layout_height="250dp"
        android:layout_width="250dp"
        android:src="@mipmap/compass" />

    <ImageView
        android:id="@+id/arrow_img"
        android:layout_width="60dp"
        android:layout_height="110dp"
        android:layout_centerInParent="true"
        android:src="@mipmap/arrow" />

</RelativeLayout>
```

(2) 手机作为多媒体平台，其中传感器发挥了非常重要的作用，如何调用手机的传感器呢？首先需要获取传感器的管理器：

```
sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
```

所有的传感器都需要由传感器管理器进行管理，包括注册传感器注销传感器等动作。

(3) 那么，如何获得方向传感器呢，可以使用下面的方法：

```
Sensor sensor = sensorManager.getDefaultSensor(Sensor.TYPE_ORIENTATION);
```

但是现在这种方法已经被废弃了，虽然还能够使用，但是已经不推荐了。现在手机获取旋转的方向都是通过加速度传感器和地磁传感器共同计算得出来的：

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

(4) 获得传感器后，需要注册监听器实现手机方向变化时候图像响应，由于方向传感器精度要求比较高，我们提高传感器更新速率，使用了

SENSOR\_DELAY\_GAME:

```
sensorManager.registerListener(listener, magneticSensor,  
    SensorManager.SENSOR_DELAY_GAME);  
sensorManager.registerListener(listener, accelerometerSensor,  
    SensorManager.SENSOR_DELAY_GAME);
```

其中 listener 为一个监听器，该监听器是一个接口，必须实现接口的函数，才能够实现功能：

```
private SensorEventListener listener = new SensorEventListener() {  
    float[] accelerometerValues = new float[3];  
    float[] magneticValues = new float[3];  
    private float lastRotateDegree;  
  
    @Override  
    public void onSensorChanged(SensorEvent event) {...}  
  
    @Override  
    public void onAccuracyChanged(Sensor sensor, int accuracy) {}  
};
```

可以看到该接口中有两个函数需要实现的，这里我们只需要得到方向变化就行了：

```

        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];
    
```

这里我们可以通过 `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

```

        float[] R = new float[9];
        float[] values = new float[3];

        SensorManager.getRotationMatrix(R, null, accelerometerValues,
            magneticValues);
        SensorManager.getOrientation(R, values);
        // Log.d("MainActivity", "value[0] is " + Math.toDegrees(values[0]));
        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;
        }
    }
}
    
```

怎么计算出来的呢？`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

(5) 如何获取经纬度信息呢？这里我们需要使用位置管理器：

```

locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
    
```

接着我们需要选择一个位置提供者确定设备当前的位置，`android` 中一般有三种位置提供者选择，分别是 `GPS_PROVIDER`，`NETWORK_PROVIDER` 和

PASSIVE\_PROVIDER，定位功能需要用户自己主动去启动，打开手机中的定位服务（这里我们使用网络定位服务）：



这里，实际上，我们可以通过判断哪种服务存在，然后调用该服务即可

```
locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
List<String> providerList = locationManager.getProviders(true);

//Log.d("Test", "" + providerList.get(0).toString());

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;
}
```

调用 `getLastKnownLocation` 能够返回一个位置信息，最后显示出来即可

```
Location loc = locationManager.getLastKnownLocation(provider);

if (loc != null) {
    showLocation(loc);
}

locationManager.requestLocationUpdates(provider, 5000, 1, locationListener);
}
```

其中，`locationListener` 为一个监听器，这个监听器能够感知位置信息的变化，实现更新功能：



```

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());
}

```

(6) 最后不要忘了在退出的时候移除监听器:

```

@Override
public void onDestroy() {
    super.onDestroy();

    if (sensorManager != null) {
        sensorManager.unregisterListener(listener);
    }

    if (locationListener != null) {
        locationManager.removeUpdates(locationListener);
    }
}

```

(7) 这里需要用到一些手机权限:

```

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET" />

```

## 【扩展】

思考如何使用加速传感器实现摇一摇功能。在实验报告中写下思路。