# 追踪雷达App的开发

## 需求分析

**主页面：**

* 基于百度地图，自己始终位于地图的中心位置
* 你的好友/敌人位于地图的各个位置（图中带圈的图标），可以显示好友/敌人的一些信息，如距离，昵称/手机号
* 点击刷新时，发送短信给好友的手机
* 好友的手机也运行这个软件时，回复一条包含自己位置（经/纬度）的短信

**好友/敌人页面：**

* 显示好友，同时有添加，删除/编辑功能
* 有昵称和电话号码2项。
* 主界面点击好友的点的时候跑到这个界面
* 好友列表点击红色减号时提示删除。

## Activity的设计

**主页面设计：**

百度地图sdk包的导入

点击刷新，实现animation动画扫描效果

点击刷新，基于SmsManage实现发送短信的功能

设置广播监听状态，一有短信接收立即回复短信

在地图上添加marker和polyline实现标准点和距离连线的添加

**好友及敌人页面的设计：**

Listview的实现

添加内部组件的接口，重写OnClickListener函数，实现listview中内部组件的事件绑定

使用自定义对话框Dialog，实现点击添加或删除的提示框的弹出

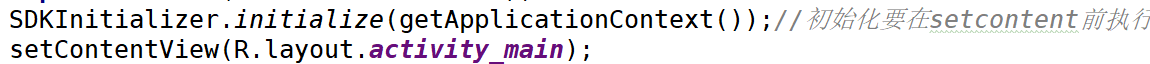
## 关键实现技术

### 百度地图sdk包的导入

在百度地图上注册开发者账号，注册后，然后下载对应得sdk文件，将期间的jar包放在app下的lib文件夹下，其他so文件在src/main/下建立jniLibs文件，然后将剩余的so文件放在此文件下，还要在manifest的配置文件中，需要配置好调用SDK的个人AK值

然后在build gradle里使用implentations files(‘libs/BaiduLBS\_Android.jar’)可以导入百度地图的sdk

在用到百度地图前，需要初始化以提供全局的context信息



### 动画扫描效果

**主类中的java类实现：**

使用Animation anim=new RotateAnimation(),通过设置参数可以实现扫描的那条绿线绕着自己设置的中心点旋转，需要注意的是，除了设置动画时间，动画运动方式，旋转的中心之外，还要设置插入器，这里考虑到可能会旋转多圈，所以设置成线性变换的旋转

Android 动画之Interpolator插入器，比较简单和常用的：

（1）LinearInterpolator：动画从开始到结束，变化率是线性变化。

（2）AccelerateInterpolator：动画从开始到结束，变化率是一个加速的过程。

（3）DecelerateInterpolator：动画从开始到结束，变化率是一个减速的过程。

（4）CycleInterpolator：动画从开始到结束，变化率是循环给定次数的正弦曲线。

（5）AccelerateDecelerateInterpolator：动画从开始到结束，变化率是先加速后减速的过程。

**Xml布局设置：**

另一种方法是在xml实现，使用rotate组件，参数根据需求设置就可以了

### 短信发送

**主要使用下面三个函数：（SmsManage类当中）**

SmsManager：管理短信操作，如发送数据，文本和PDU短信。通过调用静态方法**SmsManager.getDefault()**获取此对象。

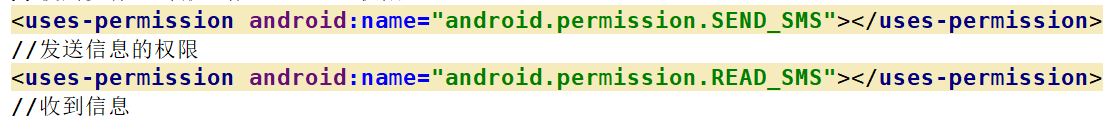
**divideMessage(String text)**

当短信超过SMS消息的最大长度时，将短信分割为几块。

**sendTextMessage(String destinationAddress, String scAddress, String text, PendingIntent sentIntent, PendingIntent deliveryIntent)**

发送消息

**开启发送消息的权限：（发收权限）**

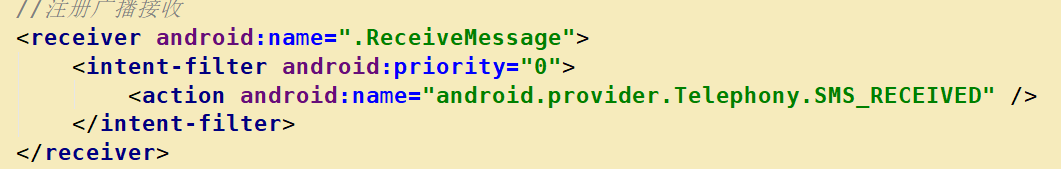


### 短信接收及回发

**通过广播监听短信消息**

（注意：这种方式只对新收到的短消息有效，运行代码，并不会读取收件箱中已读或未读的消息，只有当收到新来的短消息时，才会执行onChange方法）

**注册一个广播（AndroidManifest.xml中注册）**

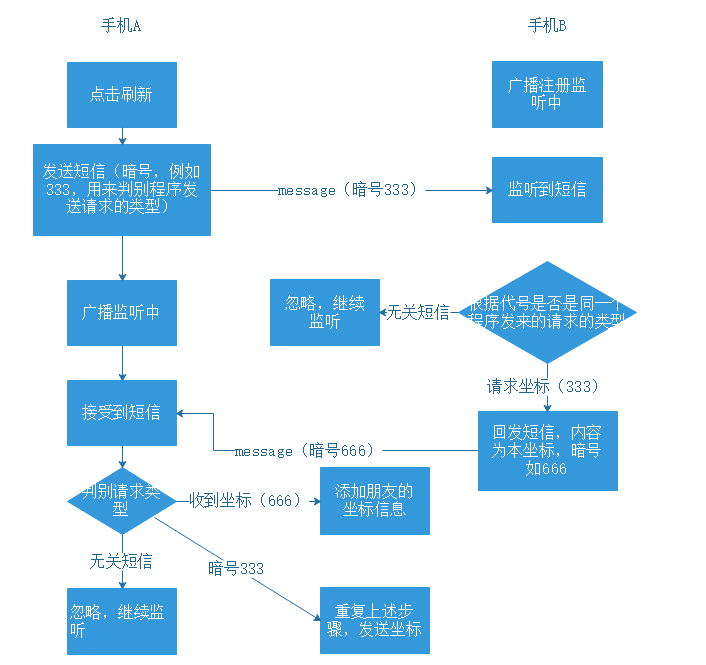


Name中的ReceiveMessage是我们监听短信后需要执行的类，priority是优先级，需要设置，不然现在在一些定制的系统或是有安全软件的情况下，往往短消息都被截取到，并被干掉。网上说是设置999，但是我这里要设置为0才能成功监听，要视不同的系统而定

**接受短信类：**

上面注册广播的所设置的ReceiveMessage继承BroadcastReceiver，重写其中的onReceive（）方法，方法中重新发送消息，发送消息的方法也如上

**下图展示收发消息时程序的处理**



### 在地图上添加标志点及连线

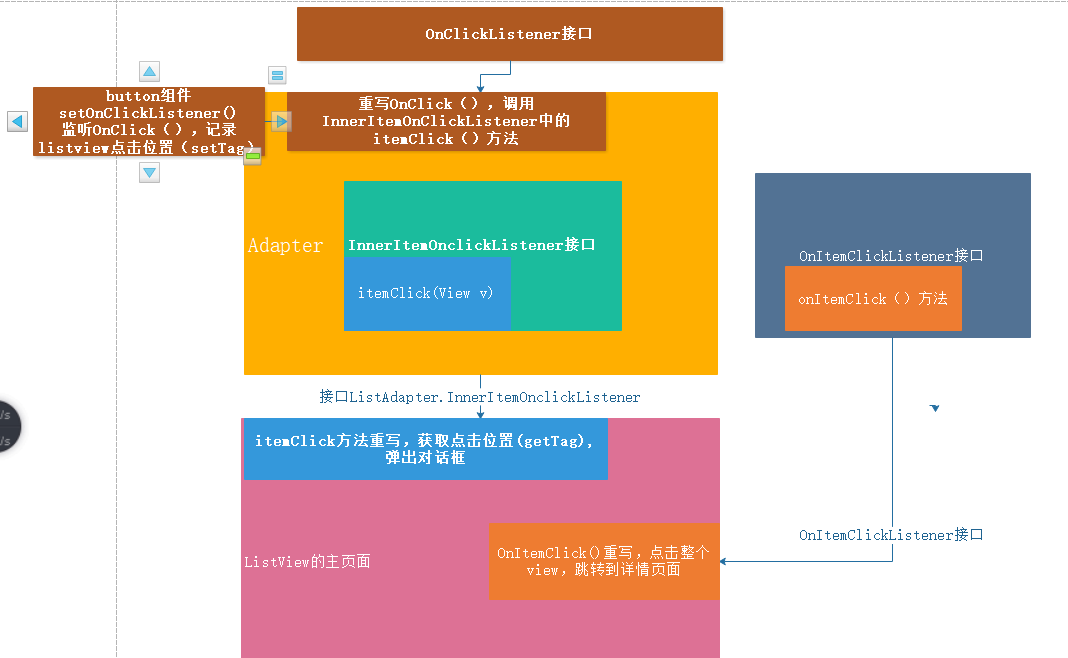
添加marker和polyline，通过markeroptions可以实现，连线则是传入2个坐标，polylineoptions便可以帮助实现，然后连线需要计算2点的距离，所以利用经纬度相见，转换成米单位，利用勾股定理算出来，两个点之间的距离

注意：在设置横线颜色或者是字体颜色的时候，直接设置R.color.XX是识别不出来的，需要ContextCompat.*getColor*(getBaseContext(), R.color.***colorRed***)来调用才能识别出颜色

### Listview的内部组件的事件绑定

列表的点击事件 setOnItemClickListener()方法,，可以实现对列表项的操作，但是如果你想实现列表项的点击事件，又想实现内部按钮的的点击事件,则需要用到接口回调

**总体接口设计思路：**



**说明：**

**点击整个列表：OnItemClick()监听可以获得listview的位置，进而获取当前的当前列表的数据**

**点击删除（列表中的内部组件）：**

点击删除按钮，执行OnClickListener接口的OnClick()方法，这个是adapter的点击监听，然后在OnClick()中执行InnerItemOnClickListener接口中的itemClick()，这个方法在ListView的主页面被重写了，所以执行重写后的itemClick()方法

**为什么不直接在Adapter中的Onclick（）监听的时候直接在此方法中写事件，还要在搞个内部接口：**

因为要实现事件所需要的数据在listview的主页面中，在主页面中重写方法可以直接使用这些数据，这里接口回调就是自己写InnerItemOnClickListener接口中的itemClick()，然后Adapter的OnClick（）监听来调用。

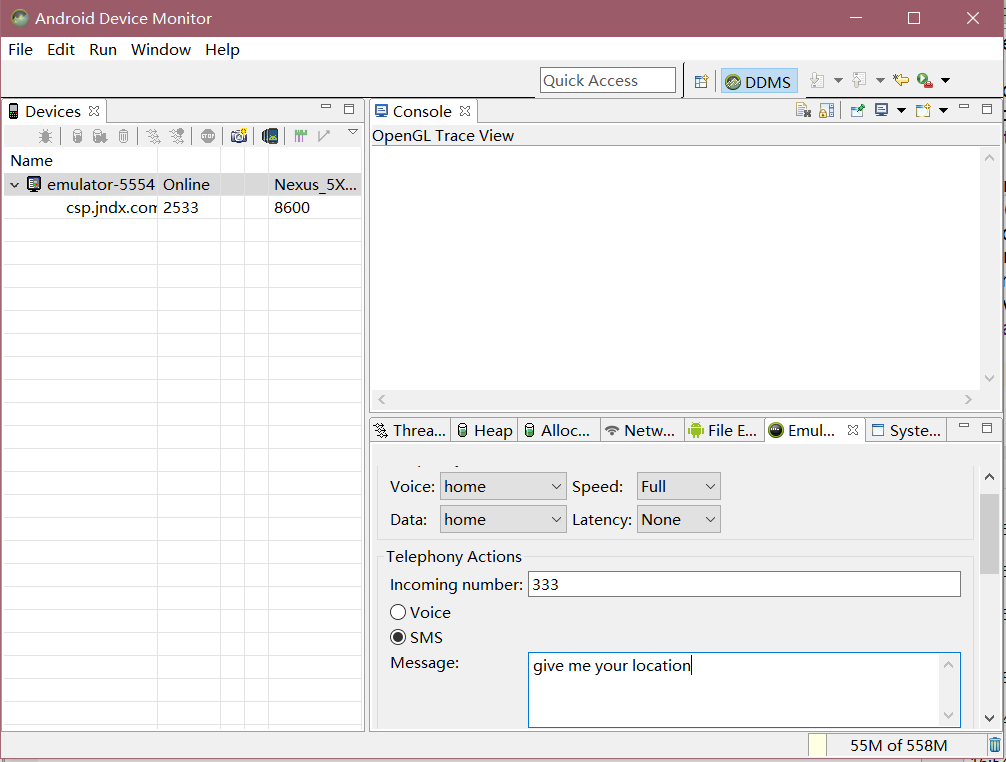
### Dialog自定义的对话框

设计思路类似于项目一的自定义对话框，自定义确认和取消按钮，绑定对应的事件，点击确定，则在适配器中删除对应位置的数据，然后刷新适配器。

## 测试及使用说明

### 测试消息发送

消息的发送及监听，因为这里只开启了一个模拟器，所以如果发送短信会直接在本机收到，如果想要模拟手机接收到短信需要打开DDMS，模拟短信发送，在Android的sdk文件下有一个tools的文件夹，点击monitors，便可以进入



点击发送手机就会收到信息，可以检验短信的功能

**短信功能检验：（测试方法）**

**点击刷新图标**：会自动发送短信，可以看到收件箱有一个号码为333的短信

**回复坐标：**模拟收到来电333的短信，自动回复一个号码为666，内容是自己坐标的短信，故在收件箱可以看到2个短信，一个是收到的333，一个是刚发出去带自己坐标信息的666

**收到坐标信息：**手机监听到666的短信，Toast显示短信内容，即坐标信息

### 使用说明

因为导入了外部包，使用的时候，自己新建一个可以运行的新项目，先拷贝覆盖src文件和libs文件，然后在build.gradle中的dependencies{}下添加导入百度地图的sdk



如果要使用自己的百度sdk，则在上面步骤后，下载的sdk的jar包替换到libs文件下，然后其他的so文件放到src/main/jniLibs下

## 待完善的地方

布局的样式尚不美观，需要进一步调整

未放到真机测试，只是用模拟器测试了一下，但基本原理基本实现

## 项目代码

https://github.com/harbour-hao/Radar