开发文档

## 1.创意来源

两人约定见面时，如果是在不熟悉的地点，经常遇到位置难以描述、找不到路的情况。此时，即使有GPS，也很难跟对方说清自己的位置，更不用说自动规划路径了。因此，需要有一款能够简单的、能够提供实时导航的手机APP来帮助两人见面。

同时使用本APP的两个用户可以互相看见对方的位置，并且实时规划出到达另一个用户的路径。

## 2.实现思路

### (1)客户端

客户端的地图(道路信息)获取使用了百度地图API。定位使用了系统提供的定位方法，在设备支持，并安装安卓7.0（Nougat）系统的环境中，可以使用GPS/Glonass/北斗共同定位。也支持使用WIFI/Cellular定位，但精度要低于卫星定位。

客户端的网络线程会不断将自己的地理位置信息以POST包的形式发送到服务器，获得的Response为另一个用户的位置信息。获取并发送位置信息的频率为3s/次。与此同时，路径规划线程会不断根据自己的位置和目标位置重新规划路径，频率为5s/次。

### (2)服务器端

服务器端对于每个POST请求，都会将收到的位置信息存入数据库（或更新已有的位置信息），并从数据库读入另一个用户的位置信息，作为Response返回。

服务器最初架设在校园网环境内，响应速度快，但对于使用时的网络环境有较多限制。现在已经将服务器端整体迁移至阿里云，响应速度略有下降，但使用范围变得更广，只要连入互联网即可使用。

## 3.运行环境

## (1)客户端

可以使用Android Studio导入。

编写时使用的Android Studio版本为：

Android Studio 2.2.3

Java版本为：

java version "1.8.0\_91"

Java(TM) SE Runtime Environment (build 1.8.0\_91-b14)

Java HotSpot(TM) 64-Bit Server VM (build 25.91-b14, mixed mode)

Android SDK版本为：

Android 7.0(Nougat)

Gradle版本为：

Gradle-2.14.1

只保证在上述环境下可以完成编译。

在文件夹内提供的APK文件在安卓7.0设备上可以完成测试，因资金有限，没有在其他版本的系统上进行测试。

在../app/src/main/AndroidManifest.xml文件中，<meta-data>标签内的android:value项为百度地图API指定的Licence Key，如果客户端的包名/SHA1值发生改变，需要同步更新此项的值。

如果修改了服务器地址，需要在../app/src/main/java/wang/yi-ru/findyou/MainActivity的第96行中，将

**netWork**.setIP(**"www.yi-ru.wang"**,**this**);

修改为

**netWork**.setIP(strURL,**this**);

其中，strURL为一个Stirng对象，内容为新的服务器的URL地址，格式如**"www.yi-ru.wang" ，**请勿添加**"/"**。

## （2）服务器端

服务器仅使用了一个PHP文件，为../login/index.php，使用时也需将此PHP文件放置在服务器根目录下的login文件夹内。

使用前请对变量$host，$username，$password，$database 进行赋值。其中，$host 为数据库服务器的地址，$username 为登陆服务器的用户名，$password 为登陆服务器的密码，$database 为数据库名。数据库内需要有一张 tb\_session 表格，执行 show create table tb\_session; 的结果如下：

+------------+---------------------------------------------------------------+

| Table | Create Table |

+------------+---------------------------------------------------------------+

| tb\_session | CREATE TABLE `tb\_session` (

`session\_key` varchar(100) DEFAULT NULL,

`session\_data` varchar(100) DEFAULT NULL,

`session\_time` bigint(20) NOT NULL,

PRIMARY KEY (`session\_time`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 |

+------------+---------------------------------------------------------------+

其他PHP文件为学习、调试、测试所用，可以删去，留在此处作为记录思路用。

PHP版本为

PHP 5.6

PHP Configure Command 为：

cscript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=c:\php-sdk\oracle\x86\instantclient\_12\_1\sdk,shared" "--with-oci8-12c=c:\php-sdk\oracle\x86\instantclient\_12\_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--without-analyzer" "--with-pgo"

Server API 为：

Apache 2.0 Handler

## 4. 技术细节

（1）客户端使用了HttpURLConnection类作为网络通信的库。

（2）PHP在处理session时，没有使用session\_set\_save\_handler()方法，但是手工实现了其中所需的6个session处理方法并且手工调用。原因是，阿里云的虚拟主机没有开放PHP.ini的全部设置选项，因此无法调整gc机制的触发概率。手工调整可以保证每次都能触发gc，在这样一个高速刷新的环境中很有必要。

## 5. 收获

（1）初步掌握Java、PHP与SQL语言，对于工程方面的程序设计有一些了解。

（2）配置Apache/PHP/SQL环境时，了解了很多系统设置方面的知识。由于之前使用了netsh interface portproxy命令将80端口绑定到了mirrors.ustc.edu.cn，导致Apache安装总是失败，而netstat –ano命令只能查出是svchost.exe占用，使我用尽各种方法才找到了被占用的原因。

（3）编程时开始自发考虑节约系统资源方面的事情，因为之前过高的刷新率导致服务器阻塞以及设备过热，后来慢慢调整，找到了刷新率和资源使用之间的平衡。