



Seeing-Eye Dog



—Group 11

group leader: 李聆嘉
group members: 王琢、闫璐、罗雨

Introduction

Status quo: APPs helping the disabled
are restricted in numbers.



Our purpose :

- offer a third eye for the blind people
- name

Our logo :

- Motto
 - Theme
- group members & all round access to the outside world

Our expectations

voice recognition



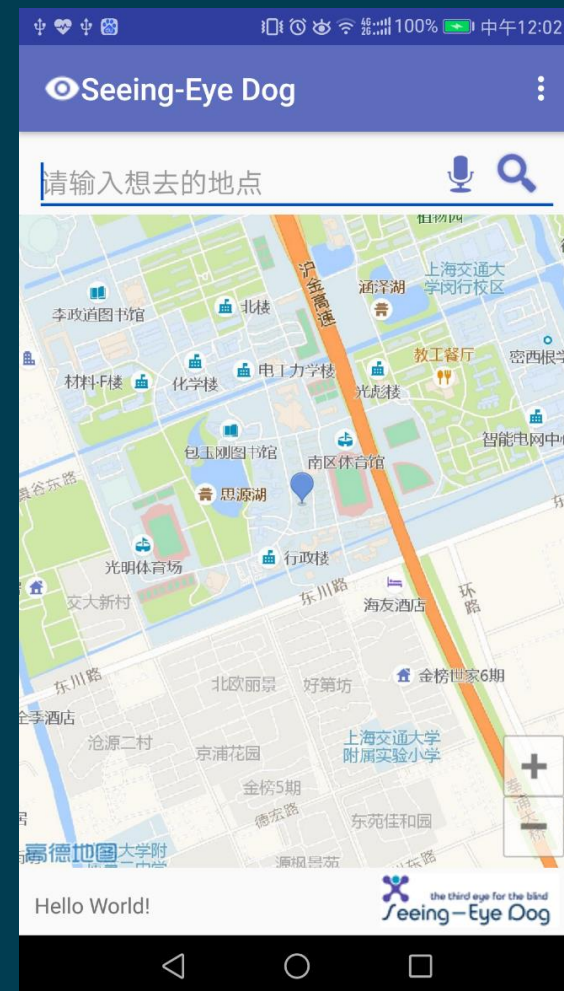
location



vibration output



navigation



Task Allocation



- 1.UI interface
- 2.voice input
- 3.vibration output

by 李聆嘉



- 1.location
- 2.navigation

by 闫璐



- 1.structure design
- 2.apply APIs

by 王琢



- 1.database
- 2.enviroment building

by 罗雨

PART ONE

UI Interface

——李聆嘉

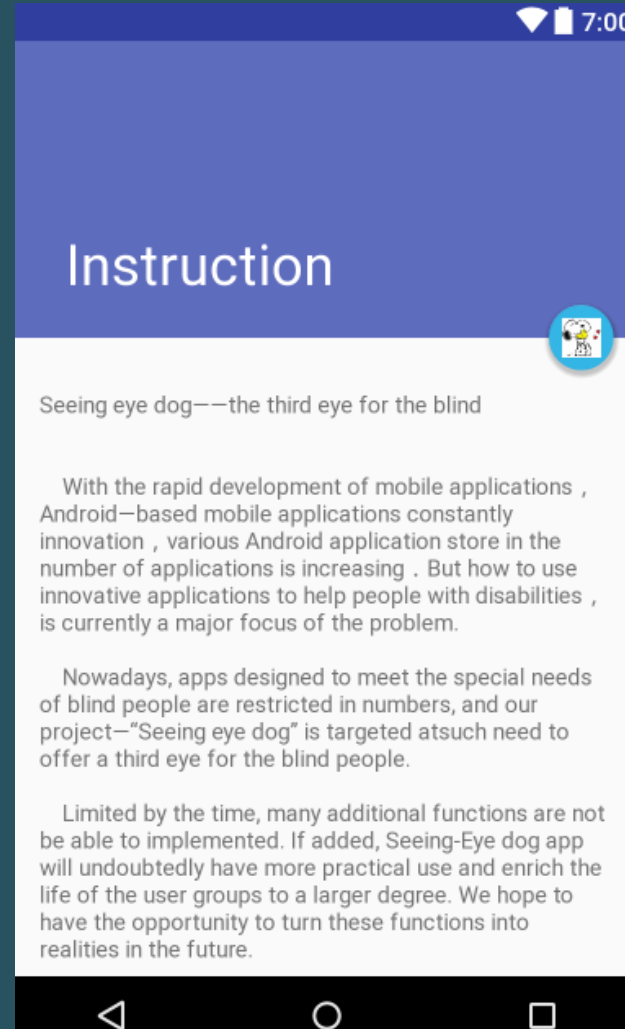
The UI



1. Main page

- ◆ **users division**
 - blind and normal people
- ◆ **2 ways of inputting information**
- ◆ **texts replaced by clear icon buttons**
- ◆ **Menu button**
 - for additional info.
- ◆ **app logo & user's command**
 - voice-input supported.

The UI



2. About us & Instruction page

PART TWO

Voice input

——李聆嘉

**initspeech* —initializing voice recognition

- ◆ invoking of hearing Fei voice SDK
- ◆ accurate and high-efficiency voice recognition



```
public void open(View view) { initspeech(this); }

/**
 * 初始化语音识别
 */
public void initspeech(final Context context) {
    //1. 创建RecognizerDialog对象
    RecognizerDialog mDialog = new RecognizerDialog(context, null);
    //2. 设置accent、language等参数
    mDialog.setParameter(SpeechConstant.LANGUAGE, "zh_cn");
    mDialog.setParameter(SpeechConstant.ACCENT, "mandarin");
    //3. 设置回调接口
    mDialog.setListener(new RecognizerDialogListener() {
        @Override
        public void onResult(RecognizerResult recognizerResult, boolean isLast)
            if (!isLast) {
                //解析语音
                String result = parseVoice(recognizerResult.getResultString());
                tv.setText(result);
            }
    });

    @Override
    public void onError(SpeechError speechError) {

    }
});
//4. 显示dialog, 接收语音输入
mDialog.show();
}
```

```

/**
 * 解析语音json
 */
public String parseVoice(String resultString) {
    Gson gson = new Gson();
    Voice voiceBean = gson.fromJson(resultString, Voice.class);

    StringBuffer sb = new StringBuffer();
    ArrayList<Voice.WSBean> ws = voiceBean.ws;
    for (Voice.WSBean wsBean : ws) {
        String word = wsBean.cw.get(0).w;
        sb.append(word);
    }
    return sb.toString();
}

/**
 * 语音对象封装
 */
public class Voice {

    public ArrayList<WSBean> ws;

    public class WSBean {
        public ArrayList<CWBean> cw;
    }

    public class CWBean {
        public String w;
    }
}

```

—functions for voice parsing & packaging



PART THREE

Vibration Output

——李聆嘉

- User groups—sensitive **tactile** and **auditory ability**
- Disadvantages of **voice** output
 - a) **noise** generated by busy traffic
 - b) Difficult to identify **environment**

Therefore, the method of **difference-vibration** is applied :

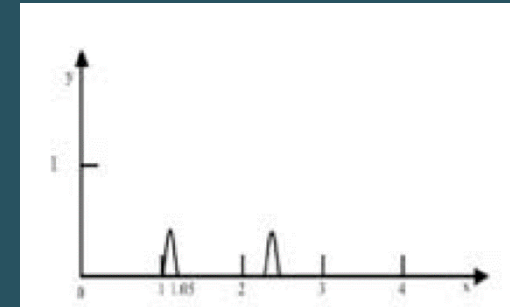
- ✓ in-time.
- ✓ noise proof.

```
Vibrator myVibrator = (Vibrator) getSystemService(Service.VIBRATOR_SERVICE);
```

```
myVibrator.vibrate(new long[]{1000, 50, 1000, 50}, 0);
```

Short vibration

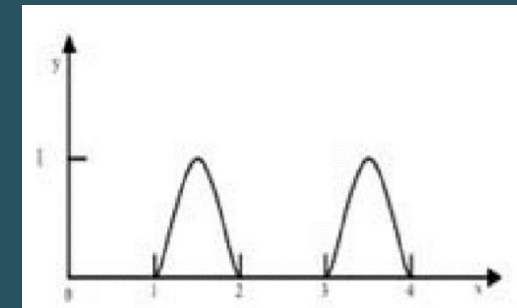
- **Waiting time:**1s
- **Lasting time:**0.05s



```
myVibrator.vibrate(new long[]{1000, 1000, 1000, 1000}, 0);
```

Long vibration

- **Waiting time and lasting time:** 1s
- **the vibration can achieve its peak.**

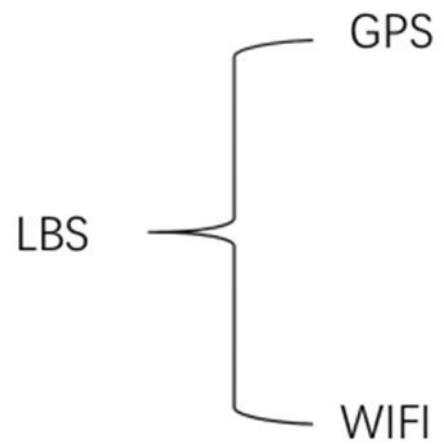


PART FOUR

location and navigation

——闫璐

location and navigation



Maybe the third
choice:
SDK of third
company

location and navigation



My main reference —

location and navigation

Apply for API key



Prepare LBS SDK



Location

Construction

- Ⓒ 🔒 AMapThread
- Ⓒ 🔒 BGThread
- Ⓒ 🔒 LocationThread
- Ⓒ 🔒 Main2Activity
- Ⓒ 🔒 Main3Activity
- Ⓒ 🔒 Main4Activity
- Ⓒ 🔒 MainActivity
- Ⓒ 🔒 MyLocationListener
- Ⓒ 🔒 Util_http

location and navigation



Display Result

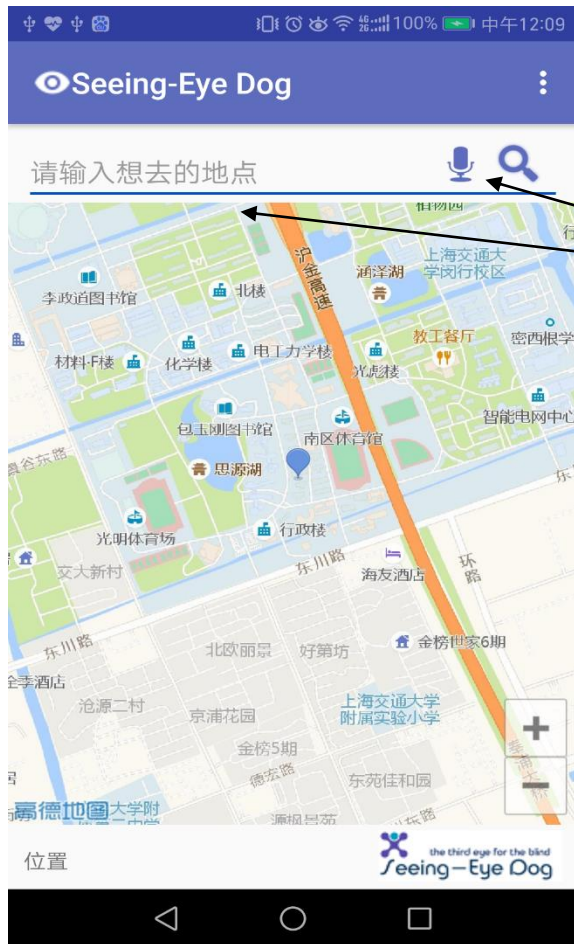
Specific to street location

Specific to latitude and longitude location

Set time interval to update location

Show map and stop GPS button

location and navigation



Two ways to input OR search your destination

location and navigation



The real-time map provided by AMAP

Press these two buttons OR sliding your finger to Zoom in and out

PART FIVE

Structure design and API

——王琢

part 3

structure design

our app is divided into 4 parts:

- 1.voice recognition
- 2.vibration output
- 3.location
- 4.navigation.

As the project is complex, we have to import many libraries to support the system.

Integration and apply APIs

To make the whole app run, I did the integration for different parts.
For UI part , I have to apply to APIs to connect the interface with our main algorithm.
For database part, there's also some connection tasks to intergrate.

import libs and modify something

```
apply plugin: 'com.android.library'

android {
    compileSdkVersion 23
    buildToolsVersion '25.0.3'

    defaultConfig {
        //applicationId "com.mirraico.bluetoothindoorlocation"
        minSdkVersion 15
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
    sourceSets.main {
        jniLibs.srcDirs = ['libs']
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.2.0'
    compile files('libs/ab-sdk-beta1.6.6.jar')
```

```
<excludeFolder url="file://$MODULE_DIR$/build/intermediates/manifests" />
<excludeFolder url="file://$MODULE_DIR$/build/intermediates/pre-dexed" />
<excludeFolder url="file://$MODULE_DIR$/build/intermediates/res" />
<excludeFolder url="file://$MODULE_DIR$/build/intermediates/rs" />
<excludeFolder url="file://$MODULE_DIR$/build/intermediates/shaders" />
<excludeFolder url="file://$MODULE_DIR$/build/intermediates/symbols" />
<excludeFolder url="file://$MODULE_DIR$/build/intermediates/transforms" />
<excludeFolder url="file://$MODULE_DIR$/build/outputs" />
<excludeFolder url="file://$MODULE_DIR$/build/tmp" />
</content>
<orderEntry type="jdk" jdkName="Android API 23 Platform" jdkType="Android SDK" />
<orderEntry type="sourceFolder" forTests="false" />
<orderEntry type="library" exported="" name="support-v4-23.2.0" level="project" />
<orderEntry type="library" exported="" scope="TEST" name="hamcrest-core-1.3" level="project" />
<orderEntry type="library" exported="" name="ab-sdk-beta1.6.6" level="project" />
<orderEntry type="library" exported="" scope="TEST" name="junit-4.12" level="project" />
<orderEntry type="library" exported="" name="support-annotations-23.2.0" level="project" />
<orderEntry type="library" exported="" name="appcompat-v7-23.2.0" level="project" />
<orderEntry type="library" exported="" name="animated-vector-drawable-23.2.0" level="project" />
<orderEntry type="library" exported="" name="support-vector-drawable-23.2.0" level="project" />
<orderEntry type="library" exported="" name="fengmap-beta-1.2.0" level="project" />
</component>
</module>
```

PART SIX

*enviroment building
&
database dedsign*

—— 罗雨

database

data base




Map **API** Provider : **Amap**




Apply official **Demo** to our app

Apply for API key

 我的应用(2) + 创建新应用

您可以在这里创建、设置并管理您的应用及Key

 SeeingEyeDog 2017-05-20创建 🗑️ ✎️ + 添加新Key ^			
Key名称	Key	绑定服务	操作 ^①
SeeingEyeDog_01	4e9ffb4ad4ea33082d7fc7296154f9a7	Android平台	设置 删除

```

MainActivity demos
    if (convertView instanceof FeatureView) {
        featureView = (FeatureView) convertView;
    } else {
        featureView = new FeatureView(getContext());
    }
    DemoDetails demo = getItem(position);
    featureView.setTitleId(demo.titleId, demo.activityClass!=null);
//    featureView.setDescriptionId(demo.descriptionId);
    return featureView;
}

private static final DemoDetails[] demos = {
//    创建地图
    new DemoDetails("创建地图", "", null),
//    显示地图
    new DemoDetails("基本地图", "介绍如何创建一个基本地图",
        BasicMapActivity.class),
//    OSM地图
    new DemoDetails("OSM地图", "介绍如何替换高德底图, 显示诸如osm之类的底图",
        OsmMapActivity.class),
//    Fragment创建地图
    new DemoDetails("SupportMapFragment创建地图", "介绍SupportMapFragment方式创建地图",
        BaseMapFragmentActivity.class),
//    地图多实例
    new DemoDetails("地图多实例", "",
        TwoMapActivity.class),
//    amapoptions实现地图
    new DemoDetails("AMapOptions实现地图",
        "介绍用AMapOptions展示一个地图", MapOptionActivity.class),
//-----与地图交互-----
    new DemoDetails("地图交互", "", null),
//    缩放控件、定位按钮、指南针、比例尺等的添加

```

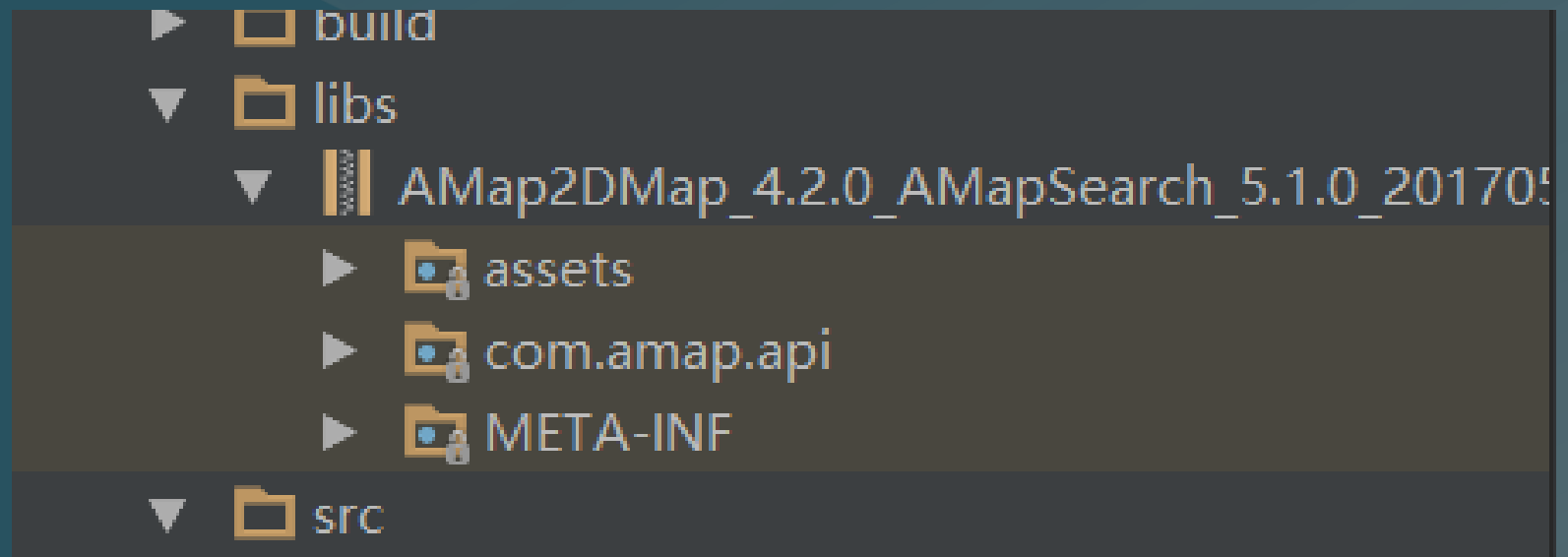
Demo codes of MainActivity

implementation codes:

```
private void init() {  
    if (aMap == null) {  
        aMap = mapView.getMap();  
        setUpMap();  
    }  
  
    mLocationErrText = (TextView)findViewById(R.id.location_errInfo_text);  
    mLocationErrText.setVisibility(View.GONE);  
}  
  
/**  
 * 设置一些amap的属性  
 */  
private void setUpMap() {  
    aMap.setLocationSource(this); // 设置定位监听  
    aMap.getUiSettings().setMyLocationButtonEnabled(true); // 设置默认定位按钮是否显示  
    aMap.setMyLocationEnabled(true); // 设置为true表示显示定位层并可触发定位，false表示隐藏定位层并不可触发定位，默认是false  
    setupLocationStyle();  
}  
  
private void setupLocationStyle() {  
    // 自定义系统定位蓝点  
    MyLocationStyle myLocationStyle = new MyLocationStyle();  
    // 自定义定位蓝点图标  
    myLocationStyle.myLocationIcon(BitmapDescriptorFactory.  
        fromResource(R.drawable.gps_point));  
    // 自定义精度范围的圆形边框颜色  
    myLocationStyle.strokeColor(STROKE_COLOR);  
}
```

After importing the libs support,
we can use the implementation codes in our app.

Libs:



Our codes:

This is a part of it, which shows how we use the map directly:

```
Intent intent = this.getIntent();
double latitude = intent.getDoubleExtra("latitude", 0);
double longitude = intent.getDoubleExtra("longitude", 0);

//获取地图控件引用
mMapView = (MapView) findViewById(R.id.map);
//在activity执行onCreate时执行mMapView.onCreate(savedInstanceState), 创建地图
mMapView.onCreate(savedInstanceState);

//mMap
aMap = mMapView.getMap();

LatLng latLng = new LatLng(latitude, longitude);
final Marker marker = aMap.addMarker(new MarkerOptions().position(latLng).title("position").snippet("DefaultMarker"));
aMap.moveCamera(CameraUpdateFactory.changeLatLng(latLng));
}

@Override
protected void onDestroy() {
    super.onDestroy();
}
```

*Summary
&
Future perspectives*

- **Additional services** related to **daily life**
- **Personalized** service (user log in/out)
- **Path sharing** on socializing platforms
- **gesture commands**

.....

➤ Technology

—satisfy the special need of the underprivileged in a more humanizing way

➤ change the world into a better place for all humanity.

ANY QUESTION

THANK YOU

