



**SUN YAT-SEN UNIVERSITY**

中山大學



# 手机应用平台软件开发

---

## 13、地图应用

---

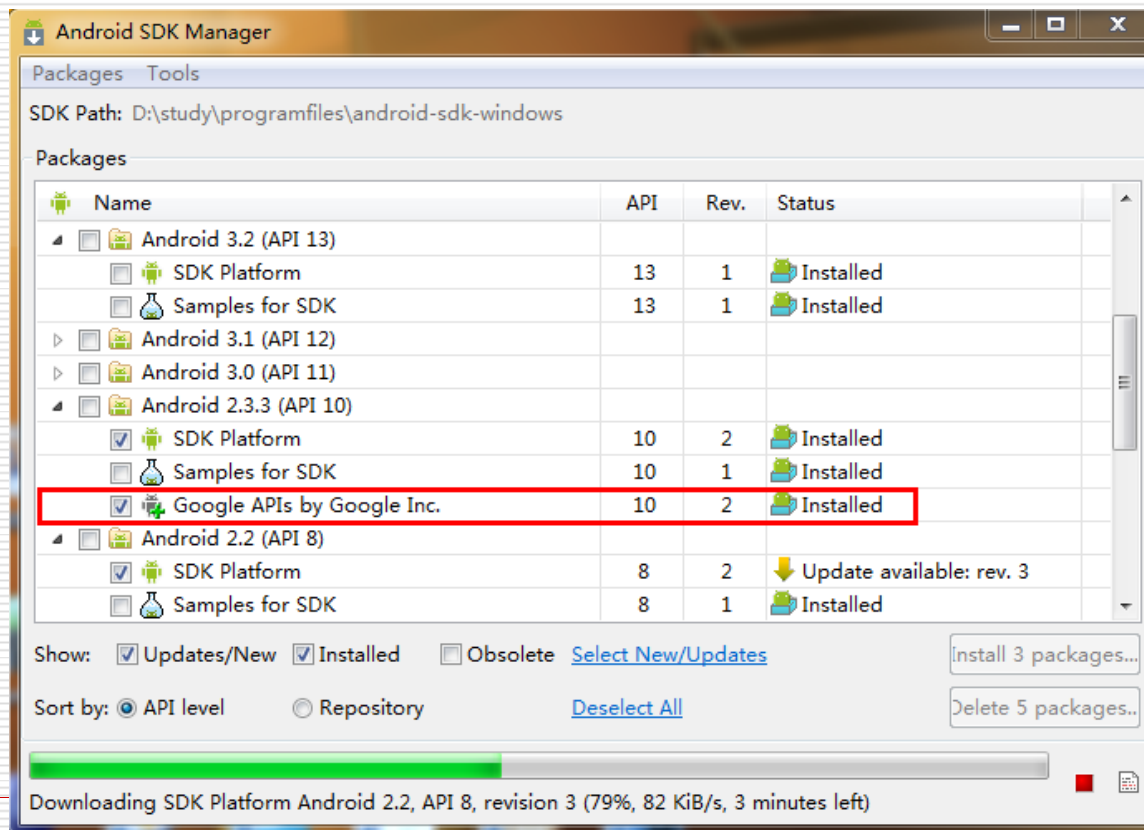


# Android Google Maps API的安装和使用

SUN YAT-SEN UNIVERSITY

## 安装Google APIs

在SDK Manager中确保已下载SDK Platform + Google APIs

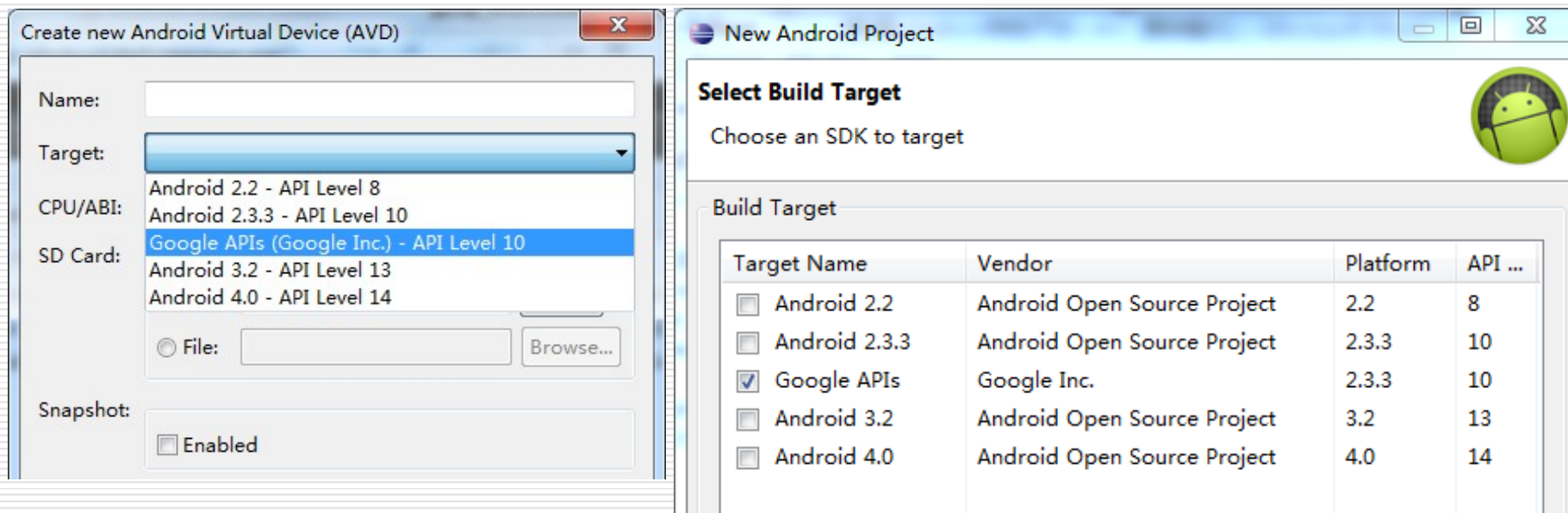




# Android Google Maps API的安装和使用

SUN YAT-SEN UNIVERSITY

## 新建支持Google APIs的模拟器





# Android Google Maps API的安装和使用

SUN YAT-SEN UNIVERSITY

## Google APIs开发文档

- 在<android-sdk>/add-ons目录下找到类似addon\_google\_apis\_google\_inc\_10的目录，该目录下包括了为模拟器所使用的已编译镜像、API类库、示例和API文档。

file:///D:/study/programfiles/android-sdk-windows/add-ons/addon\_google\_apis\_google\_inc\_10/docs/reference/index.html

**All Classes**

- [GeoPoint](#)
- [GeoPoint](#)
- [ItemizedOverlay](#)
- [ItemizedOverlay](#)
- [ItemizedOverlay.OnFocusChangeListener](#)
- [ItemizedOverlay.OnFocusChangeListener](#)
- [MapActivity](#)
- [MapActivity](#)
- [MapController](#)
- [MapController](#)
- [MapView](#)
- [MapView](#)
- [MapView.LayoutParams](#)

**Package Class Tree Deprecated Index Help**  
PREV PACKAGE NEXT PACKAGE [FRAMES](#) [NO FRAMES](#)

**Package com.google.android.maps**  
The maps package allows applications to display and control a Google Map interface.  
**See:** [Description](#)



## 申请地图密钥

- 注册“地图密钥”的第一步是申请一个Google账户，也就是Gmail电子邮箱；
  - 找到保存Debug证书的keystore的保存位置，并获取证书的MD5散列值
    - keystore是一个密码保护的文件，用来存储Android提供的用于调试的证书，获取MD5散列值的主要目的是为下一步申请“地图密钥”做准备
-

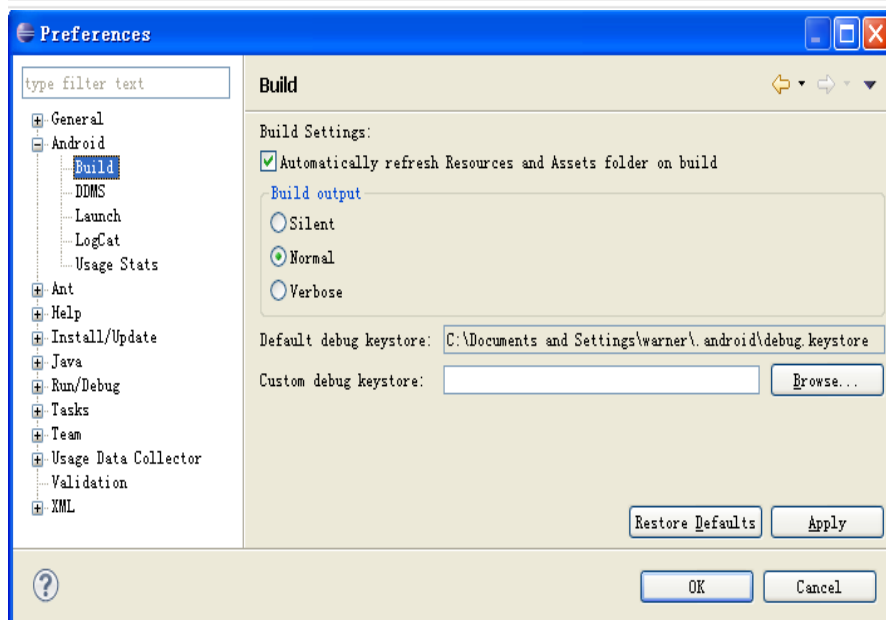


# Android Google Maps应用

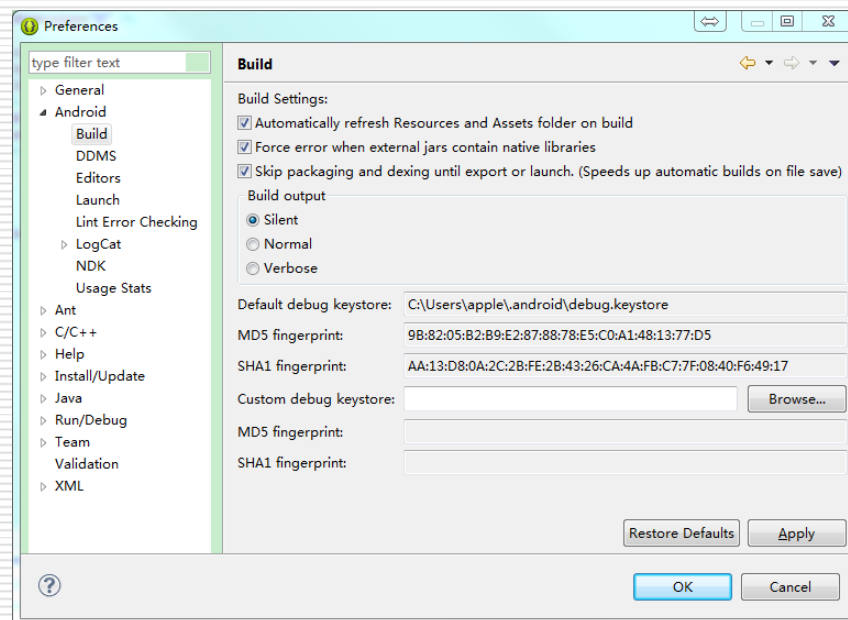
SUN YAT-SEN UNIVERSITY

## 申请地图密钥

- 获取证书的保存地址：首先打开Eclipse，通过Window → Preferences打开配置窗体，在Android → Build栏中的Default debug keystore中可以找到



旧版本



新版本



## 申请地图密钥

- 为了获取Debug证书MD5散列值的，需要打开命令行工具CMD，然后切换到keystore的目录，输入如下命令

```
keytool -list -keystore debug.keystore
```

- 如果提示无法找到keytool，可以将<Java SDK>/bin的路径添加到系统的PATH变量中



# Android Google Maps应用

SUN YAT-SEN UNIVERSITY

## 申请地图密钥

- 在提示输入keystore密码时，输入缺省密码android，MD5散列将显示在最下方。示例的MD5散列值为

68:76:89:C8:A4:24:61:F9:EA:F3:F7:70:CC:FD:C8:15

```
C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\warner\.android>keytool -list -keystore debug.keystore

输入keystore密码:

Keystore 类型: JKS
Keystore 提供者: SUN

您的 keystore 包含 1 输入

androiddebugkey, 2009-6-20, PrivateKeyEntry,
认证指纹 (MD5): 68:76:89:C8:A4:24:61:F9:EA:F3:F7:70:CC:FD:C8:15

C:\Documents and Settings\warner\.android>
```





## 申请地图密钥

- 申请“地图密钥”的最后一步是打开申请页面，输入MD5散列值
  - 申请页面的地址是<http://code.google.com/intl/zh-CN/android/add-ons/google-apis/maps-api-signup.html>
-



# Android Google Maps应用

SUN YAT-SEN UNIVERSITY

## 申请地图密钥

### ★ Google Projects for Android: Google APIs

[Home](#) [Docs](#) [Blog](#) [Group](#)

[Download](#)

[Terms](#)

#### Getting Started

[What is Google APIs Add-on?](#)

[Installing the Add-on](#)

#### Maps

[Overview](#)

[Obtaining a Maps](#)

[API Key](#)

[Maps API Key Signup](#)

[API Reference](#)

#### Resources

[Android SDK >](#)

[Android Developer's](#)

[Guide >](#)

### Maps API Key Signup

Use the form on this page to register with the Google Maps service and obtain a Maps API Key. Registration is free.

If you are developing an Android application that will display Google Maps data using the API provided in the Maps external library, you must register with the service and get an API Key.

A single Maps API key is valid for all applications signed by the corresponding developer certificate. For more information about signing, see [Signing Your Applications](#) on the Android Developers site.

To register for a Key, you also need a [Google Account](#). Once you register, your Key will be associated with your Google Account.

Before you register, read [Obtaining a Maps API Key](#) to understand how your Maps API Key is used in your Android application, why it's needed, and how to generate an MD5 fingerprint based on your developer certificate.

#### Android Maps APIs Terms of Service

Last Updated: October 13, 2008

Thanks for your interest in the Android Maps APIs. The Android Maps APIs are a collection of services (including, but not limited to, the "com.google.android.maps.MapView" and "android.location.Geocoder" classes) that allow you to include maps, geocoding, and other content from Google and its content providers in your Android applications. The Android Maps APIs explicitly do not include any driving directions data or local search data that may be owned or licensed by Google.

1. Your relationship with Google.
  - 1.1. Your use of any of the Android Maps APIs (referred to in this

☒ I have read and agree with the terms and conditions ([printable version](#))

My certificate's MD5 fingerprint:

[Generate API Key](#)



# Android Google Maps应用

SUN YAT-SEN UNIVERSITY

## 申请地图密钥

■ 输入MD5散列值后，点击Generate API Key按钮，将提示用户输入Google账户，正确输入Google账户后，您的密钥是：

```
0mVK8GeO6WUz4S2F94z52CIGSSlv1TwnrE4DsiA
```

此密钥适用于所有使用以下指纹所对应证书进行验证的应用程序：

```
68:76:89:C8:A4:24:61:F9:EA:F3:F7:70:CC:FD:C8:15
```

下面是一个 xml 格式的示例，帮助您了解地图功能：

```
<com.google.android.maps.MapView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:apiKey="0mVK8GeO6WUz4S2F94z52CIGSSlv1TwnrE4DsiA"
/>
```



## Android Maps API库使用声明

由于Android Maps API库**并不属于**标准的Android库，因此当应用程序需要使用时，需要在项目文件的AndroidManifest.xml文件中进行声明，如下：

```
<uses-library android:name="com.google.android.maps" />
```

需要把这句加在application下作为application的子元素：

```
<application android:icon="@drawable/icon"
android:label="@string/app_name">
    <uses-library android:name="com.google.android.maps" />
    .....
</application>
```



# Android Google Maps API概览

SUN YAT-SEN UNIVERSITY

包名: *com.google.android.maps*

使用了Maps API的应用程序只能运行在支持Google Maps Add-on的Android系统上，模拟器和真机都不例外。在新建Android项目时选择Build Target必须选择含有Google API的Target。如图：

Build Target			
Target Name	Vendor	Platform	API ...
<input type="checkbox"/> Android 1.5	Android Open Source Project	1.5	3
<input checked="" type="checkbox"/> Google APIs	Google Inc.	1.5	3
<input type="checkbox"/> Android 1.6	Android Open Source Project	1.6	4
<input type="checkbox"/> Google APIs	Google Inc.	1.6	4
<input type="checkbox"/> Android 2.1-up...	Android Open Source Project	2.1-up...	7
<input type="checkbox"/> Google APIs	Google Inc.	2.1-up...	7



# Android Google Maps API概览

SUN YAT-SEN UNIVERSITY

## 1. 接口

Interface Summary	
<a href="#">ItemizedOverlay.OnFocusChangeListener</a>	Interface for listeners interested in when the focused Item changes.
<a href="#">Overlay.Snappable</a>	Interface definition for overlays that contain items that can be snapped to (for example, when the user invokes a zoom, this could be called allowing the user to snap the zoom to an interesting point.)
<a href="#">Projection</a>	A Projection serves to translate between the coordinate system of x/y on-screen pixel coordinates and that of latitude/longitude points on the surface of the earth.



# Android Google Maps API 概览

SUN YAT-SEN UNIVERSITY

## 2. 类

Class Summary	
<a href="#">GeoPoint</a>	An immutable class representing a pair of latitude and longitude, stored as integer numbers of microdegrees.
<a href="#">ItemizedOverlay&lt;Item extends OverlayItem&gt;</a>	A base class for an <a href="#">Overlay</a> which consists of a list of OverlayItems.
<a href="#">MapActivity</a>	Base class with code to manage the boring necessities of any activity that displays a <a href="#">MapView</a> .
<a href="#">MapController</a>	A utility class to manage panning and zooming of a map.
<a href="#">MapView</a>	A View which displays a map (with data obtained from the Google Maps service).
<a href="#">MapView.LayoutParams</a>	Per-child layout information associated with MapView.
<a href="#">MyLocationOverlay</a>	An <a href="#">Overlay</a> for drawing the user's current location (and accuracy) on the map, and/or a compass-rose inset.
<a href="#">Overlay</a>	Base class representing an overlay which may be displayed on top of a map.
<a href="#">OverlayItem</a>	The basic component of any <a href="#">ItemizedOverlay</a> .
<a href="#">TrackballGestureDetector</a>	Analyzes a series of MotionEvent and detects gestures.



# Android Google Maps API 概览

SUN YAT-SEN UNIVERSITY

## 3. 枚举类

### Enum Summary

<a href="#">MapView.Reticle DrawMode</a>	Allow the user to specify the mode in which the reticle is drawn.
----------------------------------------------	-------------------------------------------------------------------

利用上述所列的类就可以在应用程序中显示和控制Google Maps。

---





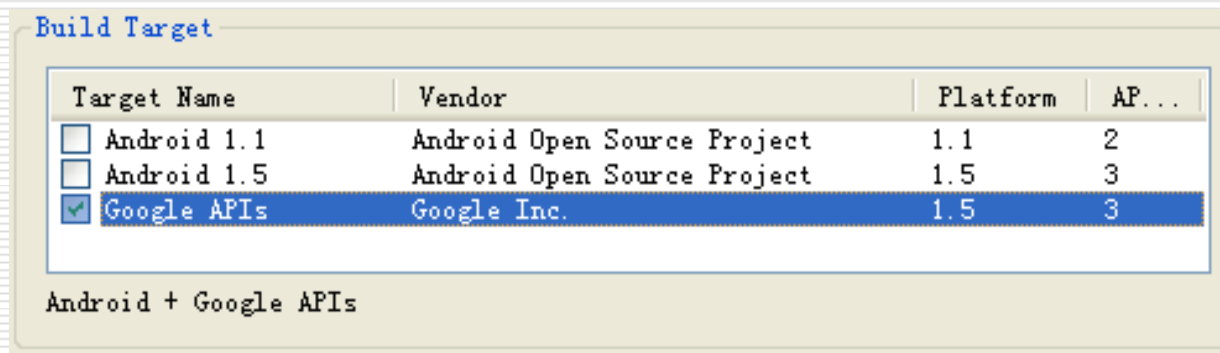
## 使用Google地图

- MapView的地图的显示控件，可以设置不同的显示模式，例如卫星模式、街道模式或交通模式
  - MapController则是MapView的控制器，可以控制MapView的显示中心和缩放级别等功能
-



## 使用Google地图

- 在建立工程时将com.google.android.maps的扩展库添加到工程中，这样就可以使用Google地图的所有功能
- 添加com.google.android.maps扩展库的方式是在创建工程时，在Build Target项中选择Google APIs



- 创建工程后，修改/res/layout/main.xml文件，在布局中加入一个MapView控件，并设置刚获取的“地图密钥”



## 使用Google地图

- 在xml文件中定义MapView及其属性

```
<com.google.android.maps.MapView  
    android:id="@+id/mapview"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    android:enabled="true"  
    android:clickable="true"  
    android:apiKey="0mVK8GeO6WUz4S94z52CIGSSlvITwnrE4DsiA"/>
```



## 使用Google地图

- 仅在布局中添加MapView控件，还不能够直接在程序中调用这个控件，还需要将程序本身设置成MapActivity（`com.google.android.maps.MapActivity`）
- MapActivity类负责处理显示Google地图所需的生命周期和后台服务管理

```
public class GoogleMapDemo extends MapActivity {  
    private MapView mapView;  
    private MapController mapController;  
    ...  
}
```



## 使用Google地图

- 经纬度坐标需要转化为GeoPoint再使用
- 以下代码设定的经度为 $126.676530486 * 1E6$ 、纬度为 $45.7698895661 * 1E6$ 的地理坐标点

```
Double lng = 126.676530486 * 1E6;
```

```
Double lat = 45.7698895661 * 1E6;
```

```
GeoPoint point = new GeoPoint(lat.intValue(), lng.intValue());
```



## 使用Google地图

- 第1行代码获取了MapController，并通过MapController 进行各项设置

```
mapController = mapView.getController();  
mapController.setCenter(point); //设置MapView的“显示中点”  
mapController.setZoom(11); //设置放大层级  
mapController.animateTo(point); //将MapView显示区域的中心移动到第2行设置的“显示中心”
```



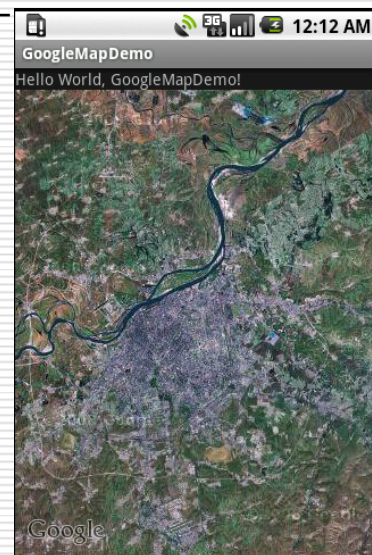
## 使用Google地图

- 设定MapView的地图显示模式是否为卫星模式，设置true则为卫星模式，设置false则为普通模式

1. `mapView.setSatellite(false)`



■ 地图模式



■ 卫星模式



### 使用Google地图

- 由于获取Google地图是需要使用互联网的，所以在运行前还需要在AndroidManifest.xml文件中，添加允许访问互联网的许可

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





## 使用Overlay

- 通过在MapView上添加覆盖层，可以在指定的位置添加注解、绘制图像或处理进行鼠标事件等
  - Google地图上可以加入多个覆盖层，所有覆盖层均都在地图图层之上，每个覆盖层均可以对用户的点击事件做出响应
  - 创建覆盖层继承Overlay类的子类，并通过重载draw()方法为指定位置添加注解，重载onTap()方法处理用户的点击操作
-



## 使用Overlay

- 下面的代码是创建Overlay的最小代码集合

```
public class TextOverlay extends Overlay {  
    @Override  
    public void draw(Canvas canvas, MapView mapView, boolean shadow) {  
        if (shadow == false){//false表示在覆盖层上进行绘制  
        }  
        else{  
        }  
        super.draw(canvas, mapView, shadow);  
    }  
    @Override  
    public boolean onTap(GeoPoint p, MapView mapView) {  
        return false;  
    }  
}
```



## 使用Overlay

- 代码中draw()方法，shadow变量是用来区分绘制图层
    - false表示在覆盖层上进行绘制
    - true则表示在隐藏层上进行绘制
  - 代码是onTap()方法的返回值
    - 返回false表示覆盖层不处理点击事件
    - 返回true则表示已经处理了点击事件
  - 在覆盖层绘制图形或文字需要使用“画布”（Canvas）来实现，绘制的位置是屏幕坐标，这就需要~~将地图上的物理坐标与屏幕坐标~~进行转换
-



## 使用Overlay

- Projection类提供了物理坐标和屏幕坐标的转换功能，可在经度和纬度表示的GeoPoint点和屏幕上Point点进行转换
  - toPixels()方法将物理坐标转换为屏幕坐标
  - fromPixels()方法将屏幕坐标转换为物理坐标
- 两个方法的具体使用方法可以参考下面的代码

```
1. Projection projection = mapView.getProjection();  
2. projection.toPixels(geoPoint, point);  
3. projection.fromPixels(point.x, point.y);
```



## 使用Overlay

- 以MapOverlayDemo示例，说明如何在Google地图上添加覆盖层，并在预订的物理坐标上显示提示信息
- 右图是MapOverlayDemo示例的运行结果





## 使用Overlay

- TextOverlay类是MapOverlayDemo示例的覆盖层，主要重载了draw()方法，在指定的物理坐标上绘制了标记点和提示文字
- 建立覆盖层文件的核心代码如下

```
public class TextOverlay extends Overlay {  
    private final int mRadius = 5; //定义了绘制半径变量mRadius  
    @Override  
    public void draw(Canvas canvas, MapView mapView, boolean shadow) {  
        Projection projection = mapView.getProjection();
```



### 使用Overlay

```
if (shadow == false){  
    Double lng = 126.676530486 * 1E6;  
    Double lat = 45.7698895661 * 1E6;  
    GeoPoint geoPoint = new GeoPoint(lat.intValue(), lng.intValue());  
    //使用Projection完成了从物理  
    //坐标到屏幕坐标的转换  
    Point point = new Point();  
    projection.toPixels(geoPoint, point); //oval设定标记点的大小  
    RectF oval = new RectF(point.x - mRadius, point.y - mRadius, point.x + mRadius, point.y +  
mRadius);  
    Paint paint = new Paint();  
    paint.setARGB(250, 250, 0, 0); //设置了绘制颜色  
    paint.setAntiAlias(true); //开启了平滑设置, 防止文字出现锯齿  
    paint.setFakeBoldText(true);  
    canvas.drawOval(oval, paint); //绘制了圆形的标记点  
    canvas.drawText("标记点", point.x+2*mRadius, point.y, paint);}  
    super.draw(canvas, mapView, shadow); }  
@Override  
public boolean onTap(GeoPoint p, MapView mapView) {  
    return false;  
}  
}
```



## 使用Overlay

- 建立了覆盖层后，还需要把覆盖层添加到MapView上，核心代码如下

//实例化了TextOverlay对象

```
textOverlay = new TextOverlay();
```

//通过getOverlays()方法，获取MapView已有的覆盖层

```
List<Overlay> overlays = mapView.getOverlays();
```

//使用add()方法将TextOverlay对象对象添加到MapView中

```
overlays.add(textOverlay);
```

---





# 运行MapsDemo示例

SUN YAT-SEN UNIVERSITY

## MapsDemo样例:

在Eclipse中可以通过New→Android Project→选择Create project from existing sample→选择Google APIs→选择MapsDemo→Finish完成项目的创建。并需要将前面获取到的API Key加入到mapview.xml中。如下。

```
<com.google.android.maps.MapView  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:enabled="true"  
    android:clickable="true"  
    android:apiKey="0k-swCMzyAu0tDwVytvRtZi5--YM34YpwGVyq1Q"  
/>
```

---



# 运行MapsDemo示例

SUN YAT-SEN UNIVERSITY

示例初步效果





# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

## 1、添加GPS位置获取功能

首先需要实现LocationListener接口，该接口所包含的方法如下：

```
public abstract void onLocationChanged (Location location); //当地点发生改变时被调用
public abstract void onProviderDisabled (String provider); //当GPS的provider被禁用时被调用
public abstract void onProviderEnabled (String provider); //与上一方法相反
public abstract void onStatusChanged (String provider, int status, Bundle extras); //当GPS状态发生改变时被调用
```

---



# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

## 添加**GPS**位置获取功能

实现一个名为MyLocationListener的类使其实现LocationListener接口，该类的功能是当GPS数据更新时，在手机界面上显示一个Toast消息框，消息内容为新的位置经纬度，并且将地图定位至新的GPS数据所代表的地点。

---



# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

## 添加**GPS**位置获取功能--注册MyLocationListener

```
context = getBaseContext(); //获取应用程序上下文环境
locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE); //获取位置管理器
locationListener = new MyLocationListener(context, mapView,
    mapController); //新建位置监听器对象
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0,
    locationListener); //注册位置监听器
```

---



# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

## 添加GPS位置获取功能--初始化MapView

```
mapView = (MapView)findViewById(R.id.map);
mapView.setBuiltInZoomControls(true);//使用默认的缩放按钮
mapView.displayZoomControls(true);//显示缩放按钮
mapController = mapView.getController();//得到地图控制器
final int defaultZoomLevel = 17;
mapController.setZoom(defaultZoomLevel);//缩放等级调至默认
final double dLong = 103.9242; //默认地点经纬度
final double dLati = 30.75777;
GeoPoint defaultPoint = new GeoPoint((int)(dLati*1E6),(int)(dLong*1E6));
mapController.animateTo(defaultPoint);//将地图中心移至默认地理位置
```

---



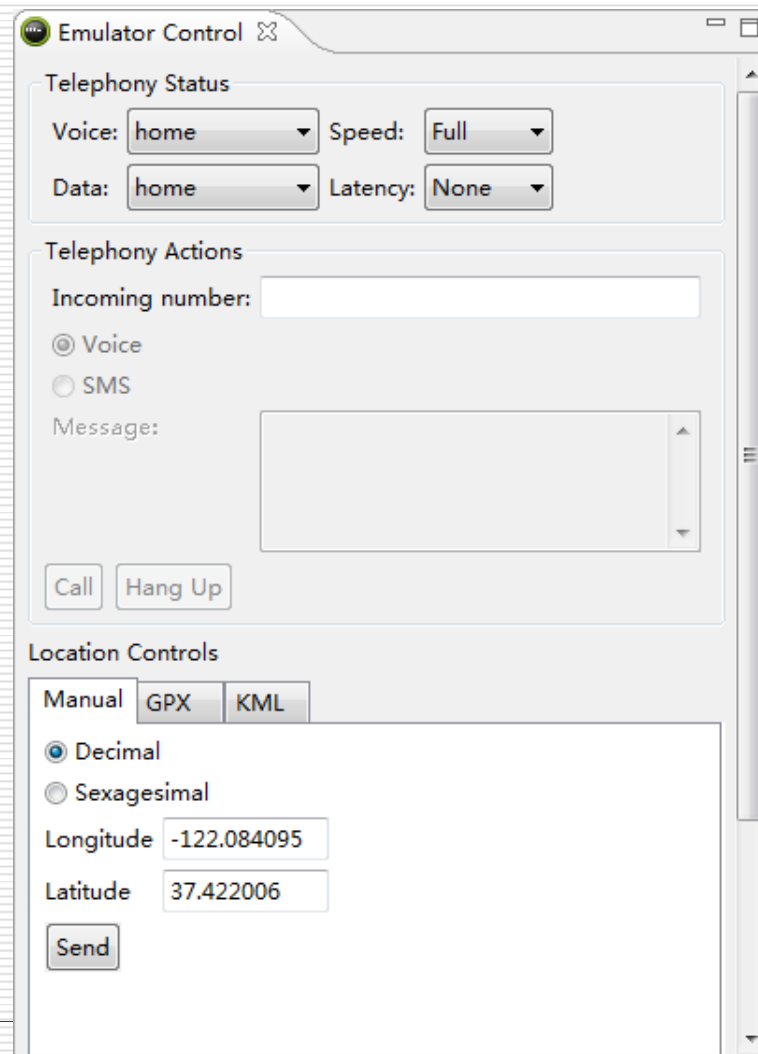
# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

## 添加GPS位置获取功能

使用DDMS发送GPS数据模拟位置获取功能

使用DDMS对模拟器的一些状态的设置，例如调整音量、电量、模拟来电和短信等等，在最下方则是用于模拟地理位置的Location Controls，为模拟器发送地理位置包括发送一个单独的位置（Manual）、以及发送一串保存在文件中的多个位置（GPX，KML）这几种方式。

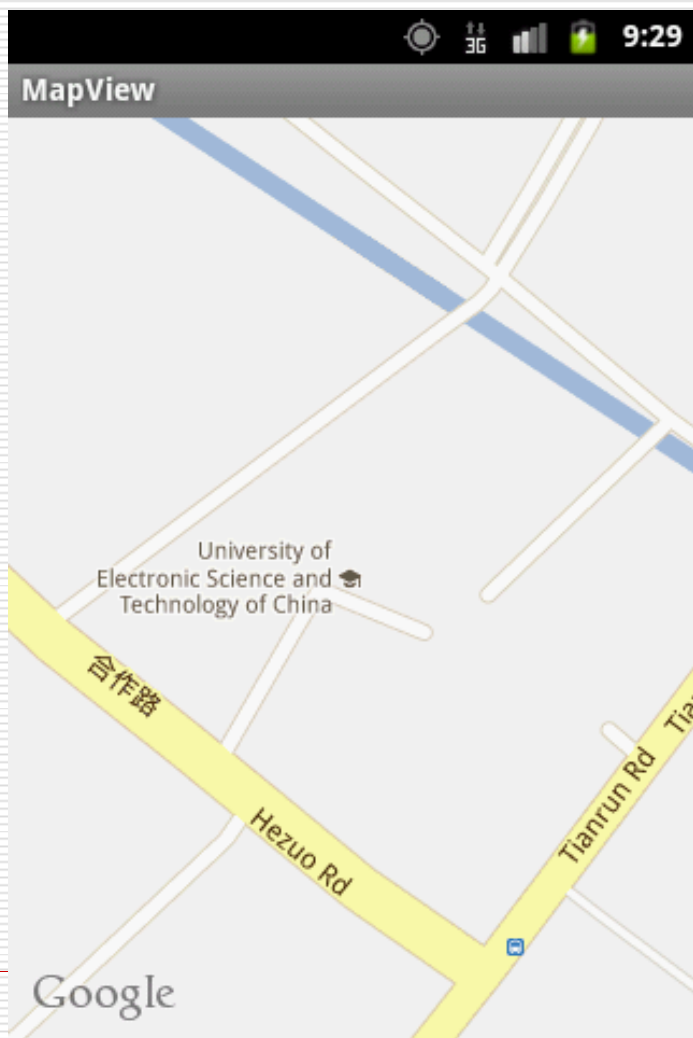




# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

## 添加GPS位置获取功能







# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

## 在MapView上标记位置

在实际应用中经常需要在地图上进行标记，包括使用地标标记地点，或者使用弹出气泡来显示相关信息等。实现在地图上显示地标（图片），需要使用Map API中的`OverlayItem`类，就是在MapView上覆盖一层视图。为此需要实现一个地标类并继承`OverlayItem`类，代码如下：

```
public class PlaceMarker extends OverlayItem {  
    private static int placeID = 1;  
    private int myID = 0;  
    public PlaceMarker(GeoPoint point, String title,  
        String snippet, Drawable marker) {  
        super(point, title, snippet);  
        myID = placeID++;  
        this.setMarker(marker);  
    }  
    public int getID(){  
        return myID;  
    }  
}
```



# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

## 在MapView上标记位置--OverlayItem类的方法

类 名	描 述
Drawable getMarker(int stateBitset)	获取指定状态（stateBitset）的标志图
GeoPoint getPoint()	返回该OverlayItem的地理位置
String getSnippet()	返回该OverlayItem的简介
String getTitle()	返回该OverlayItem的标题
String routableAddress()	以map-routable格式返回该OverlayItem的位置
void setMarker(Drawable marker)	设置该OverlayItem需要被绘制时所用的标记
void setState(Drawable drawable, int stateBitset)	设置多个状态的标记



# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

## 在MapView上标记位置--实现PlaceMarkerList类

```
private static PlaceMarkerList theInstance = null;

Context mContext;

public PlaceMarkerList(Drawable defaultMarker, Context context) {
    super(boundCenterBottom(defaultMarker));
    mContext = context;}

public static PlaceMarkerList getInstance(Drawable defaultMarker, Context
context)
{   if (theInstance == null)
    { theInstance = new PlaceMarkerList(defaultMarker, context);}
    return theInstance;
}
```

---



## 在MapView上标记位置--修改MapViewDemo

SUN YAT-SEN UNIVERSITY

### 1、将PlaceMarkerList图层添加到MapView的Overlay列表中

//添加新的图层用于显示地标

```
List<Overlay> mapOverlays = mapView.getOverlays();
```

```
final Drawable defaultMarker =
```

```
this.getResources().getDrawable(R.drawable.markera);
```

```
PlaceMarkerList placeMarkerList = PlaceMarkerList.getInstance(defaultMarker,  
this);
```

```
placeMarkerList.setOnFocusChangeListener(onFocusChangeListener);
```

```
mapOverlays.add(placeMarkerList);
```



## 在MapView上标记位置--修改MapViewDemo

SUN YAT-SEN UNIVERSITY

### 2、通过手动添加标记的方式，在地图上显示两个标记

//添加地点A

```
final double dLongA = 103.9242;
```

```
final double dLatiA = 30.75777;
```

```
GeoPoint geoPointA = new GeoPoint((int)(dLatiA*1E6),(int)(dLongA*1E6));
```

```
final Drawable markera = this.getResources().getDrawable(R.drawable.markera);
```

```
PlaceMarker placeMarkerA = new PlaceMarker(geoPointA, "学校", "电子科技大学", markera);
```

```
placeMarkerList.addPlace(placeMarkerA);
```

//添加地点B

```
final double dLongB = 103.9258;
```

```
final double dLatiB = 30.7547;
```

```
GeoPoint geoPointB = new GeoPoint((int)(dLatiB*1E6),(int)(dLongB*1E6));
```

```
final Drawable markerb = this.getResources().getDrawable(R.drawable.markerb);
```

```
PlaceMarker placeMarkerB = new PlaceMarker(geoPointB, "公交站", "阳光地带", markerb);
```

```
placeMarkerList.addPlace(placeMarkerB);
```



# 在MapView上标记位置——实现弹出式气泡

## SUN YAT-SEN UNIVERSITY

气泡的显示是通过在MapView上添加覆盖在其上View的方式来实现，为此，定义View类型的对象popView用于实现气泡视图的显示，首先需要实现的是气泡内部的界面布局，与Activity的布局实现一样，气泡内部的布局也是通过xml文件来实现的。

```
<?xml version="1.0" encoding="UTF-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/bubble"
    android:orientation="vertical"
    android:paddingBottom="10px"
    android:paddingLeft="5px"
    android:paddingRight="5px"
    android:paddingTop="5px" >
```

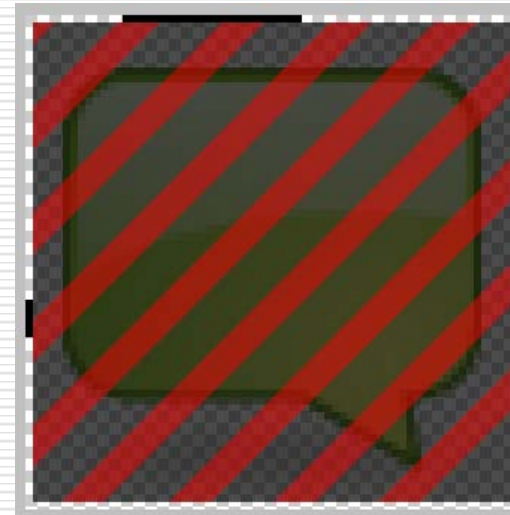
.....



# 在MapView上标记位置——实现弹出式气泡

## SUN YAT-SEN UNIVERSITY

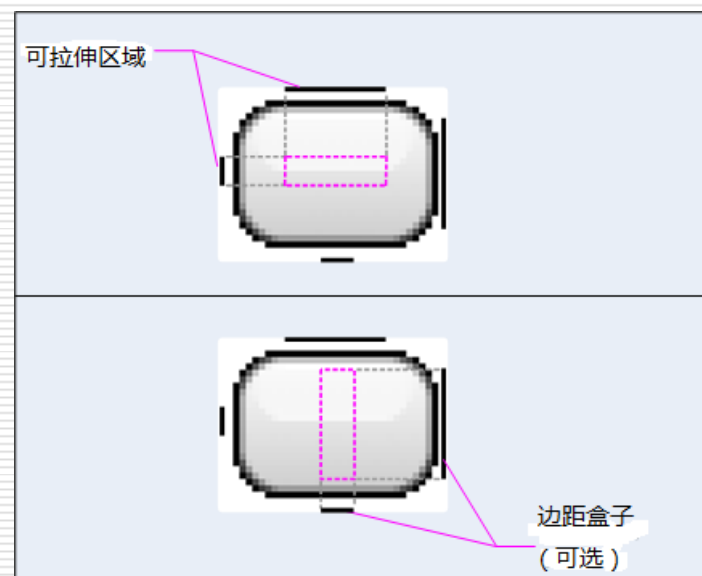
Draw 9-patch工具的使用，避免的简单缩放所造成的效果失真





### □ 背景自适应且不失真问题

自适应背景图片是UI开发的广泛问题。如列表的背景图固定，列表高度随着列表数据项发生变化时，保证背景不失真。**Android**专门制作了**9.PNG**格式来解决该问题。**9.PNG**是标准**PNG**格式，只是在最外面一圈额外增加**1px**边框，用来定义图片中可扩展和静态不变的区域。**left**和**top**边框中交叉部分是可拉伸部分，未选中部分是静态区域部分。**right**和**bottom**边框中交叉部分则是内容部分（变形的相当于定义看一个内边距），该参数可选的。







# 在MapView上标记位置——实现弹出式气泡

## SUN YAT-SEN UNIVERSITY

初始化气泡视图，实现了气泡的布局之后，通过方法：

```
popView = getLayoutInflater().inflate(R.layout.bubble, null);
```

来实现对气泡视图的填充。并通过如下代码将popView添加到mapView中：

```
01    mapView.addView(popView,  
02        new MapView.LayoutParams(MapView.LayoutParams.WRAP_CONTENT,  
03        MapView.LayoutParams.WRAP_CONTENT, null,  
04        MapView.LayoutParams.BOTTOM | MapView.LayoutParams.RIGHT));  
05    popView.setVisibility(View.GONE);  
06    bubbleTitle = (TextView) findViewById(R.id.map_bubbleTitle);  
07    bubbleSnippet = (TextView) findViewById(R.id.map_bubbleSnippet);
```

---



# 在MapView上标记位置—实现弹出式气泡

## SUN YAT-SEN UNIVERSITY

实现监听器，该监听器的作用是监听被**Focus**的**item**的变化，代码实现如下：

```
private final ItemizedOverlay.OnFocusChangeListener onFocusChangeListener =
    new ItemizedOverlay.OnFocusChangeListener() {
        @Override
        public void onFocusChanged(ItemizedOverlay overlay, OverlayItem newFocus) {
            if (newFocus != null) {
                //将气泡在被新点击的地点处显示出来
                MapView.LayoutParams geoLP = (MapView.LayoutParams)
                    popView.getLayoutParams();
                geoLP.point = newFocus.getPoint();
                popView.setVisibility(View.VISIBLE);
                bubbleThread = new BubbleThread(newFocus, mHandler);
                bubbleThread.start();
            }
        }
    };
```



# 在MapView上标记位置—实现弹出式气泡

SUN YAT-SEN UNIVERSITY

气泡内的信息是从网络获取，为了防止用户界面被阻塞，从而使用线程的方式。

BubbleThread的实现如下：

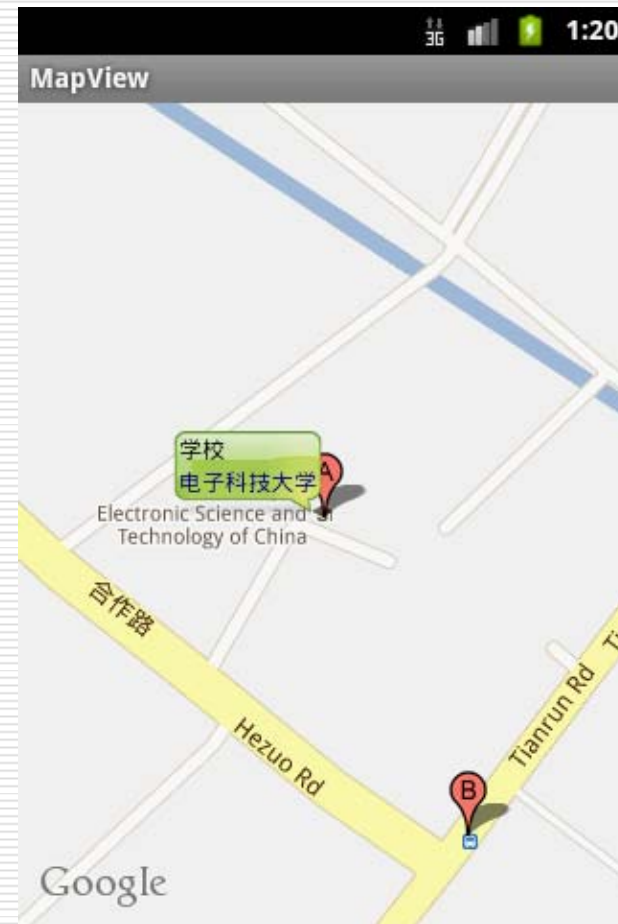
```
01     public class BubbleThread extends Thread {
02         private OverlayItem newFocus;
03         private Handler mHandler;
04
05         public BubbleThread(OverlayItem newFocus, Handler mHandler) {
06             this.newFocus = newFocus;
07             this.mHandler = mHandler;
08         }
09
10         public void run() {
11             Message msg = new Message();
12             msg.what = MapViewDemo.MESSAGE_TITLE;
13             msg.obj = newFocus.getTitle();
14             mHandler.sendMessage(msg);
15             msg = new Message();
16             msg.what = MapViewDemo.MESSAGE_SNIPPET;
17             msg.obj = newFocus.getSnippet();
18             mHandler.sendMessage(msg);
19         }
20
21         public void setNewFocus(OverlayItem newFocus) {
22             this.newFocus = newFocus;
23         }
24     }
```



# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

在MapView上标记位置





# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

## 在MapView上绘制轨迹

- 类似于标记效果的实现原理，在MapView上绘制轨迹也可以利用在其上添加Overlay的方法来实现，为了实现轨迹的折线效果，可以把轨迹分解成一段一段的线段，每一个线段根据两个GPS返回的地理位置来确定即可。
  - 可以采用两种方式来获取一系列的地理位置经纬数据，第一种方法是通过Google Earth生成kml文件的方法来模拟一连串的地理位置数据；第二种方式是借助于Google Map提供的Web Service接口来获取代表两个地点间路径的xml数据。
-



# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

在**MapView**上绘制轨迹---Google Earth生成kml文件

首先，安装Google Earth

Google earth

English (US)

Home

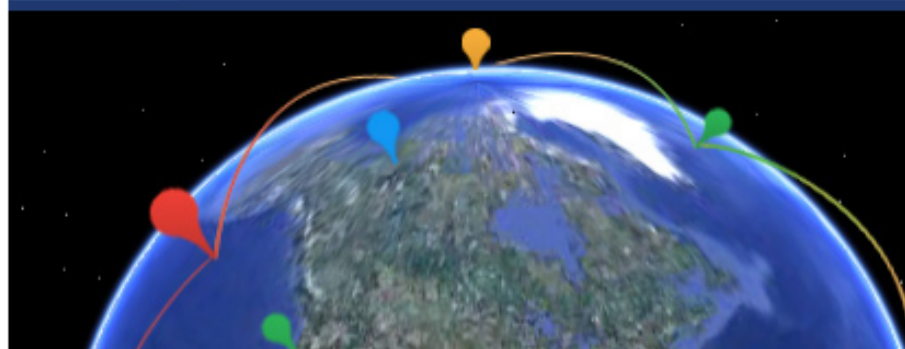
Explore

Download

Learn

Connect

Help



Get the world's geographic information at your fingertips.

Download Google Earth 6



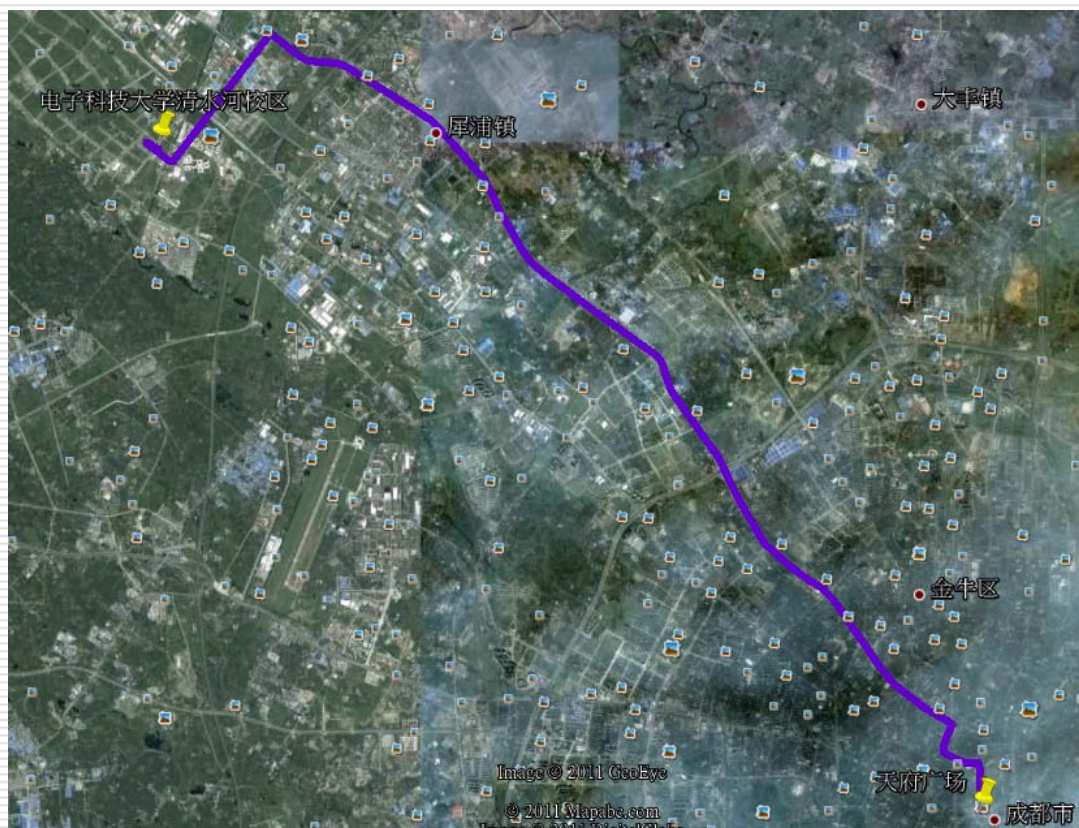


# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

在**MapView**上绘制轨迹---Google Earth生成kml文件

在Google Earth上选择起点和终点，并生成kml文件





# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

## 在MapView上绘制轨迹---Google Earth生成kml文件

### kml文件的结构

```
01      <?xml version="1.0" encoding="UTF-8"?>
02      <kml xmlns="http://www.opengis.net/kml/2.2"
03            xmlns:gx="http://www.google.com/kml/ext/2.2"
04            xmlns:kml="http://www.opengis.net/kml/2.2"
05            xmlns:atom="http://www.w3.org/2005/Atom">
06      <Placemark>
07          <name>路线</name>
08          <visibility>0</visibility>
09          <description><![CDATA[路程: 21.6&#160;公里 (大约 34 分钟)<br/>
10              地图数据 ©2011 Mapabc]]></description>
11          <styleUrl>#roadStyle</styleUrl>
12          <MultiGeometry>
13              <LineString>
14                  <coordinates>
15                      103.92328, 30.75685, 0 .....
16                  </coordinates>
17              </LineString>
18          </MultiGeometry>
19      </Placemark>
20  </kml>
```





# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

## 在MapView上绘制轨迹---Google Earth生成kml文件 解析kml文件

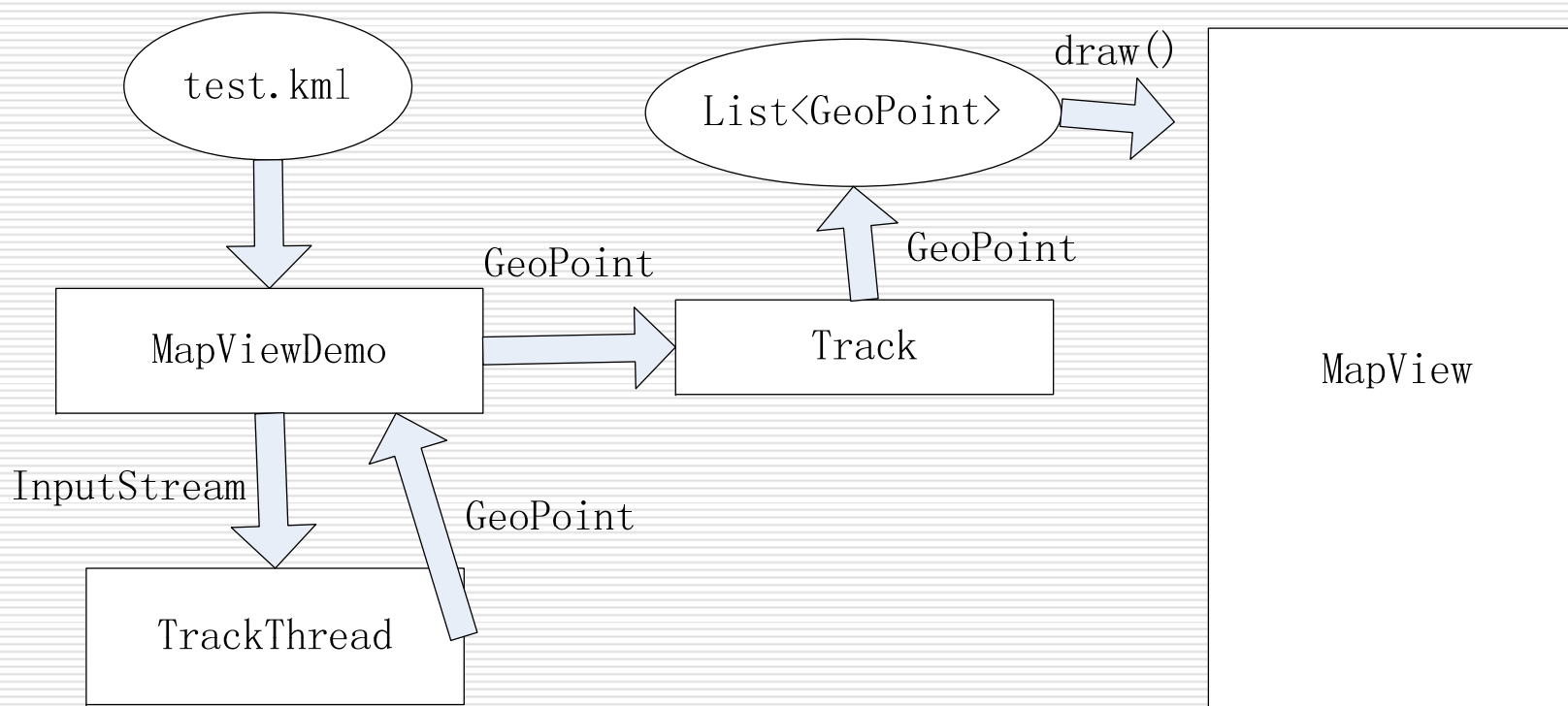
```
11  NodeList geoPointList = doc.getElementsByTagName("LineString");
12  for(int indexOfLine=0; indexOfLine< geoPointList.getLength();
    indexOfLine++) {
13      Node coordinatesNode = geoPointList.item(indexOfLine);
14      String[] coordinates =
          coordinatesNode.getNodeTextContent().split(" ");
15      for(int index = 0; index < coordinates.length - 1; index++) {
16          String lon_lat_alt= coordinates[index];
17          int lon=(int) (Double.parseDouble(lon_lat_alt.split(",")[0])*1e6);
18          int lat=(int) (Double.parseDouble(lon_lat_alt.split(",")[1])*1e6);
19          currentPoint = new GeoPoint(lat,lon);
20          Message msg = new Message();
21          msg.what = MapViewDemo.MESSAGE_TRACK;
22          msg.obj = currentPoint;
23          mHandler.sendMessage(msg);
24      }
25  }
```



# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

## 在MapView上绘制轨迹---实现绘制





# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

## 在MapView上绘制轨迹--使用Google Map Web Service获取路径

```
String urlString =  
    "http://maps.google.com/maps?f=d&hl=en&saddr=30.756850,103.923280" +  
        "&daddr=30.659960,104.062280&ie=UTF&om=0&output=kml";  
URLConnection urlConnection = null;  
URL url = null;  
url = new URL(urlString.toString());  
urlConnection = (URLConnection) url.openConnection();  
urlConnection.setRequestMethod("GET");  
urlConnection.setDoOutput(true);  
urlConnection.setDoInput(true);  
urlConnection.connect();  
trackThread = new TrackThread(urlConnection.getInputStream(), mHandler);
```



# 为MapsDemo示例添加功能

SUN YAT-SEN UNIVERSITY

在MapView上绘制轨迹



# Questions?



ANDROID