

#13 地图应用



地图开发

- 随着移动互联网应用的迅速发展，利用智能手机提供的实时地理位置信息服务功能扩展出众多 LBS(Location Based Service) 应用，比如实时定位、导航、搜索周围好友、基于地理位置的信息推荐等。在android开发中地图和定位是很多软件不可或缺的内容，这些特色功能也给人们带来各种生活便利。
- Google在2013年2月为Android平台推出了Google Maps Android API v2，使用全新的设计方式，改良地图控件的画图效率，增加3D建筑物，并简化了绘图API。Android已经在2013年4月停止使用旧的API。
- 该 API 会自动处理对 Google 地图服务器的访问、数据下载、地图显示以及对地图手势的响应。您还可以利用 API 调用向基础地图添加标记、多边形和叠层，以及更改特定地图区域的用户视图。这些对象可为地图位置提供附加信息，实现用户与地图的交互。

地图开发

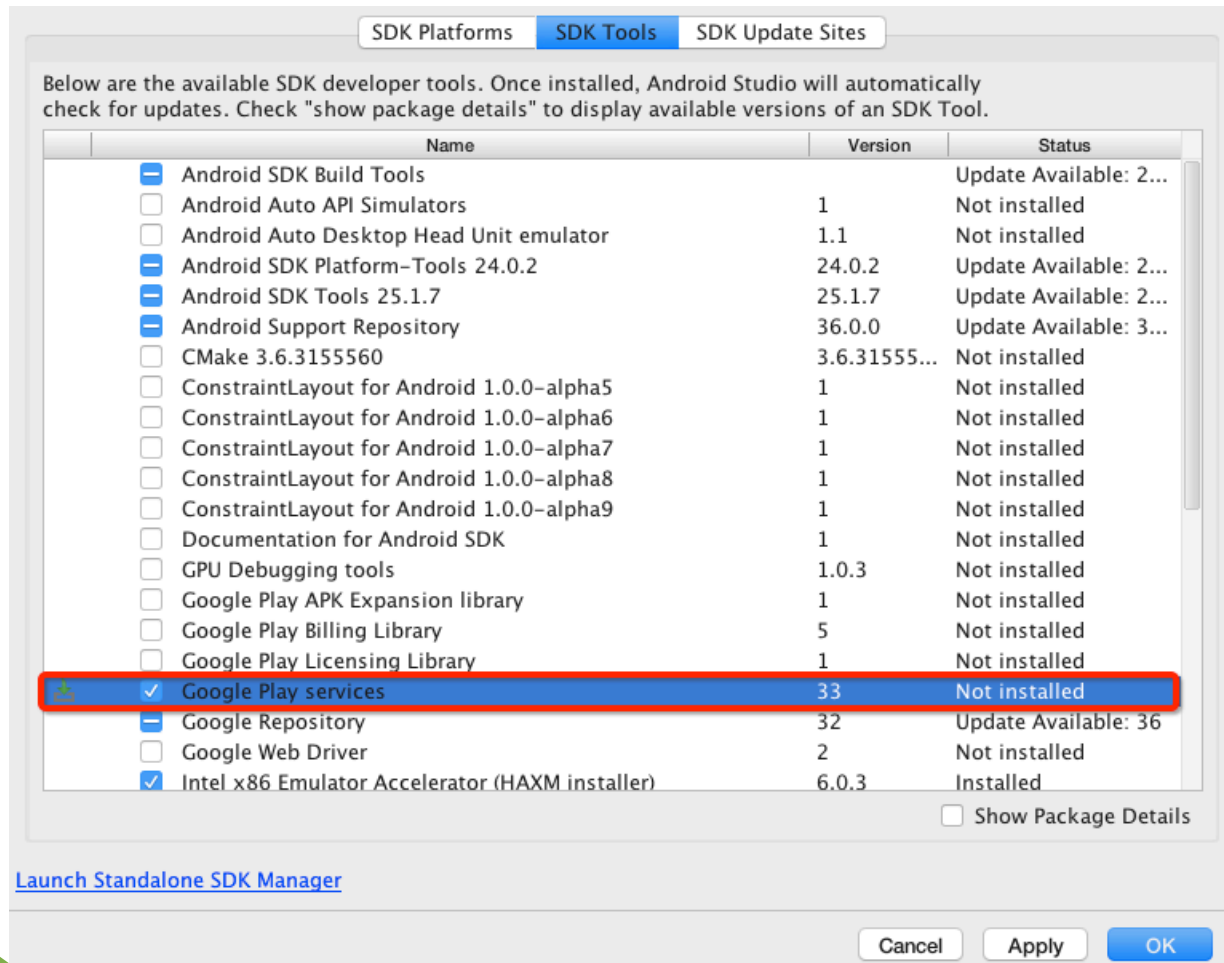
- Hello Map
- 地图属性
- 用户与地图交互
- 位置信息 MyLocation
- 地图标记 Marker
- 添加形状

地图开发

- **Hello Map**
- 地图属性
- 用户与地图交互
- 位置信息 MyLocation
- 地图标记 Marker
- 添加形状

HelloMap 安装API

- Google Maps Android API 包含在Google Play Service SDK中。首先，需要确保在SDK Manager中已下SDK Platform + Google APIs



HelloMap 开发文档

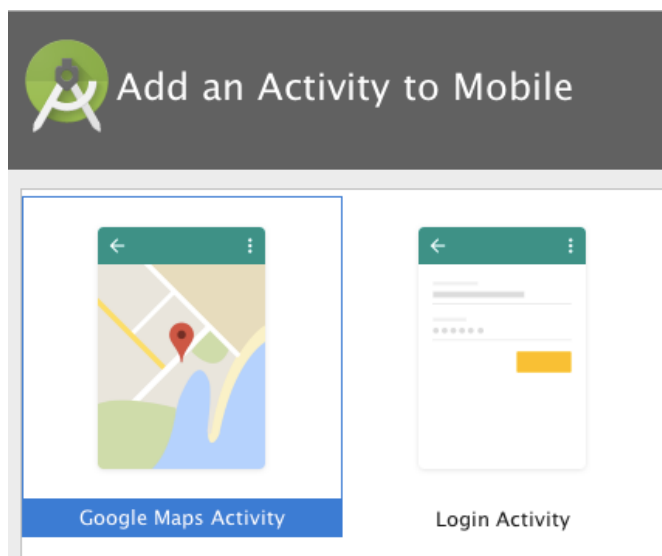
在 `Android/sdk/extras/google/google_play_services` 目录中，包含了Google服务相关的文档和实例代码。

The screenshot shows a web browser window displaying the 'Package Index' for Google Play Services. The address bar shows the file path: `file:///Users/yangqj/Library/Android/sdk/extras/google/google_play_services/docs/reference/packages.html`. The page has a search bar and a list of package names on the left sidebar. The main content area lists the packages and their descriptions.

Package Name	Description
<code>com.google.android.gms</code>	
<code>com.google.android.gms.actions</code>	Contains classes for Google Search Actions.
<code>com.google.android.gms.ads</code>	Contains classes for Google Mobile Ads.
<code>com.google.android.gms.ads.doubleclick</code>	Contains classes for DoubleClick for Publishers.
<code>com.google.android.gms.ads.formats</code>	Contains classes for native ads functionality withi
<code>com.google.android.gms.ads.identifier</code>	Contains classes relating to the Android Advertisi
<code>com.google.android.gms.ads.mediation</code>	Contains classes for Google Mobile Ads mediatic
<code>com.google.android.gms.ads.mediation.admob</code>	Contains classes for the AdMob mediation adap
<code>com.google.android.gms.ads.mediation.customevent</code>	Contains classes for Google Mobile Ads mediatic
<code>com.google.android.gms.ads.purchase</code>	Contains classes for In-App Purchase Ads.

HelloMap 创建 Google Maps 项目

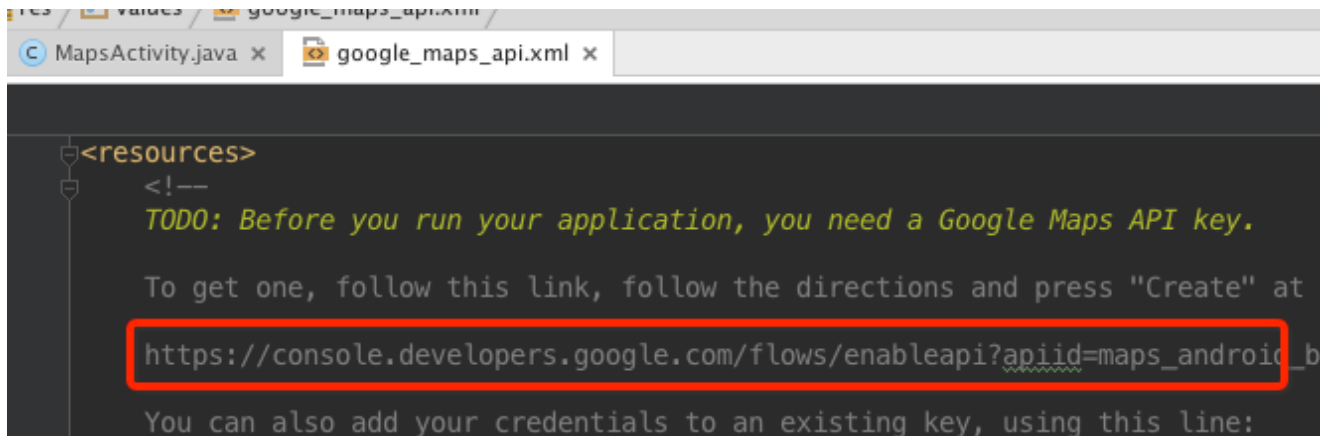
- 按照平常新建项目的步骤新建Google Maps项目，在 “Add an activity to Mobile” 对话框中选择 **Google Maps Activity**。



- 构建完成后，Android Studio 会在编辑器中打开 `google_maps_api.xml` 文件和 `MapsActivity.java` 文件。

HelloMap 申请地图密钥

- 应用需要使用 API 密钥来访问 Google 地图服务器。密钥类型是 **Android 应用专用密钥**，这意味着该密钥只能为当前这个应用使用。
- google_maps_api.xml 文件包含有关在您尝试运行应用前获取 Google Maps API 密钥的说明。将文件中提供的链接复制到浏览器中，根据界面提示快速获取API密钥。



```
<resources>
  <!--
    TODO: Before you run your application, you need a Google Maps API key.

    To get one, follow this link, follow the directions and press "Create" at
    https://console.developers.google.com/flows/enableapi?apiid=maps_android_ba

    You can also add your credentials to an existing key, using this line:
```


HelloMap 申请地图密钥

按照上述步骤，获取到应用的API 密钥，将其复制到google_maps_api.xml文件的相应位置。

API 密钥

<input type="checkbox"/> 名称	创建日期 ▾	限制	密钥
<input type="checkbox"/> API key 1	2016年10月13日	Android app	[REDACTED]

```
<string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">
    [REDACTED]
</string>
```

HelloMap 配置文件

检查Android Studio自动生成的相关配置文件

- 在manifest中定义了获取的API 密钥

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyDhUeGwPCi7-VlM5G-8B-35D1-489-A" />
```

- 在manifest中定义了应用需要获取位置信息的权限

```
<!--
    The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
    Google Maps Android API v2, but you must specify either coarse or fine
    location permissions for the 'MyLocation' functionality.
-->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

HelloMap 配置文件

- 如果目标平台是**Android 6.0(API 23)** 以前的版本，还需要申明外部存储器读写权限

```
<!-- EXTERNAL_STORAGE permissions are optional for Android 6.0 onwards. -->
<uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE"
    android:maxSdkVersion="22" />
<uses-permission
    android:name="android.permission.READ_EXTERNAL_STORAGE"
    android:maxSdkVersion="22" />
```

- 地图应用需要使用网络，但是网络权限已经默认声明了，因此不需要在manifest中显式声明。
- 在gradle中，已经自动添加了对Google Play Service 的依赖

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:24.2.0'
    compile 'com.google.android.gms:play-services:9.4.0'
}
```

HelloMap

完成上述步骤后，就可以直接运行程序，在界面上显示出地图。如果使用模拟器运行程序，需要的模拟器映像具有的 Google API 平台应基于 **Android 4.2.2 或更高版本**。此外，将模拟器设置为**x86 目标 AVD**。这可以改善您的模拟器使用体验。



地图开发

- Hello Map
- 地图属性
- 用户与地图交互
- 位置信息 MyLocation
- 地图标记 Marker
- 添加形状

Google地图应用 显示地图

使用地图对象时的关键类是 **GoogleMap** 类。GoogleMap 在应用内为地图对象建模。

GoogleMap 自动处理下列操作：

- 连接到 Google 地图服务
- 下载地图图块。
- 在设备屏幕上显示图块。
- 显示如平移和缩放等各类控件。
- 通过移动和缩放地图响应平移和缩放手势

Google地图应用 显示地图

地图将由 **MapFragment** 对象或 **MapView** 对象表示。

MapFragment

- MapFragment 是 Android Fragment 类的一个子类，用于在 Android Fragment 中放置地图。MapFragment 对象充当地图容器，并提供对 GoogleMap 对象的访问权。
- 与 View 不同，Fragment 表示的是 Activity 中的一种行为或用户界面的某一部分。可以将多个 Fragment 组合在一个 Activity 中来构建多窗格 UI，以及在多个 Activity 中重复使用某个 Fragment。

MapView

- MapView 是 AndroidView 类的一个子类，用于在 View 中放置地图。MapView 与 MapFragment 很相似，它也充当地图容器，通过 GoogleMap 对象公开核心地图功能。

Google地图应用 显示地图

为了显示地图，首先需要向将处理地图的 Activity 添加 **Fragment** 对象。

- 最简单的实现方式是，向 Activity 的布局文件添加 <fragment> 元素。在该元素中，将 android:name 属性设置为 "com.google.android.gms.maps.MapFragment"。此操作会自动将 MapFragment 附加到 Activity。

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    android:name="com.google.android.gms.maps.MapFragment"
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

- 此外，还可以通过代码向 Activity 添加 MapFragment

```
mMapFragment = MapFragment.newInstance();
FragmentManager fragmentManager =
    getFragmentManager().beginTransaction();
fragmentManager.add(R.id.my_container, mMapFragment);
fragmentManager.commit();
```


Google地图应用 显示地图

- 应用内使用地图，需要实现 **OnMapReadyCallback** 接口，并在 **MapFragment** 对象或 **MapView**对象上设置回调实例，这通过使用 **getMapAsync()** 设置。

```
public class MainActivity extends FragmentActivity
    implements OnMapReadyCallback {
    ...
}
```

```
MapFragment mapFragment = (MapFragment) getFragmentManager()
    .findFragmentById(R.id.map);
mapFragment.getMapAsync(this);
```

- 使用 **onMapReady(GoogleMap)** 回调方法获取 **GoogleMap** 对象的句柄。这个回调将在地图做好使用准备时触发。

```
@Override
public void onMapReady(GoogleMap map) {
    map.addMarker(new MarkerOptions()
        .position(new LatLng(0, 0))
        .title("Marker"));
}
```

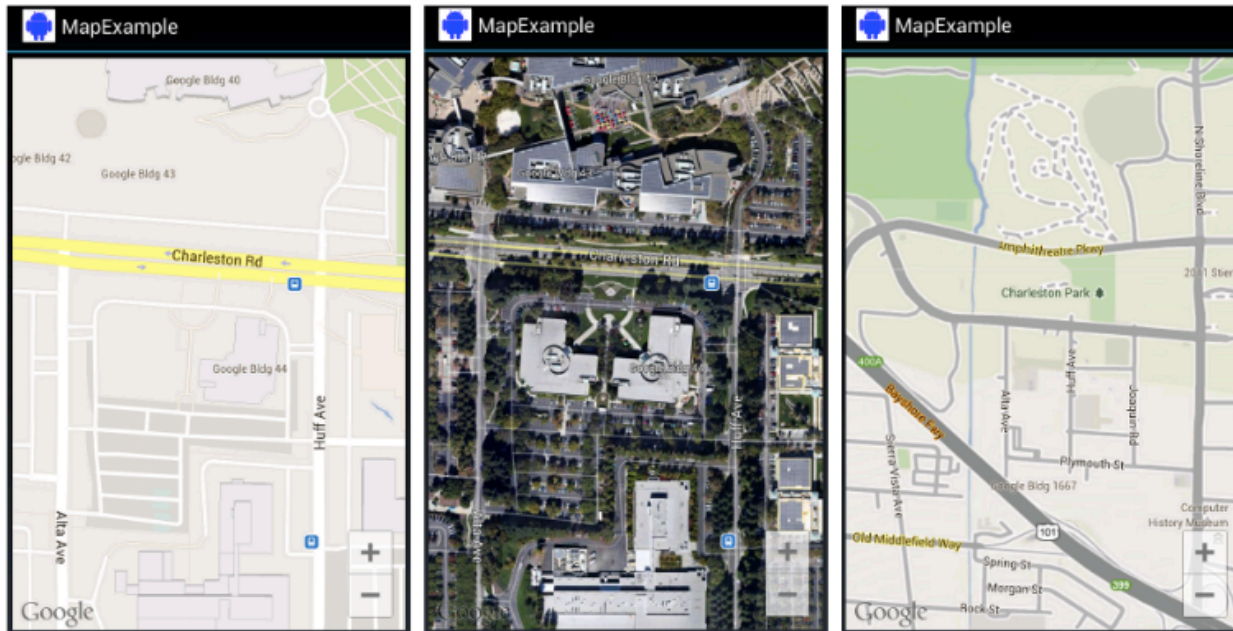
Google地图应用 地图类型

Google Maps Android API 提供了**四种地图**，以及**不显示任何地图的选项**：

- **Normal**：典型道路地图。显示道路、一些人造景观以及河流等重要的自然景观。此外，还会显示道路和景观标签。
- **Hybrid**：添加了道路地图的卫星照片数据。此外，还会显示道路 和景观标签。
- **Satellite**：卫星照片数据。 不显示道路和景观标签。
- **Terrain**：地形数据。 地图包含颜色、轮廓线和标签以及 透视阴影。此外，还会显示一些道路和标签。
- **None**：无图块。 地图将渲染为空网格，不加载任何图块。

Google地图应用 地图类型

- 以下图像显示的是同一位置 normal 地图、hybrid 地图和 terrain 地图的比较



- 如需设置地图类型，可以调用 GoogleMap 对象的 **setMapType()** 方法，传递 GoogleMap 中定义的其中一个类型常量。或者直接在布局文件中设置。

```
mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
```

Google地图应用 配置地图初始状态

- Maps API 允许配置地图的初始状态，可以指定以下内容：
 - 摄像头位置，包括：位置、缩放比例、方位和倾斜角度。
 - 地图类型
 - 缩放按钮和/或指南针是否出现在屏幕上
 - 用户在操纵摄像头时可使用的手势
- 如果已将地图添加到 Activity 的布局文件，可以通过 XML 配置地图的初始状态；如果是编程方式添加地图，则可以编程方式进行配置。

Google地图应用 配置地图初始状态

```
GoogleMapOptions options = new GoogleMapOptions();  
  
options.mapType(GoogleMap.MAP_TYPE_SATELLITE)  
    .compassEnabled(false)  
    .rotateGesturesEnabled(false)  
    .tiltGesturesEnabled(false);
```

编程方式

```
<fragment xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:map="http://schemas.android.com/apk/res-auto"  
    android:name="com.google.android.gms.maps.SupportMapFragment"  
    android:id="@+id/map"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    map:cameraTargetLat="-33.796923"  
    map:cameraTargetLng="150.922433"  
    map:cameraTilt="30"  
    map:cameraZoom="13"  
    map:mapType="normal" />
```

xml方式

地图开发

- Hello Map
- 地图属性
- 用户与地图交互
- 位置信息 MyLocation
- 地图标记 Marker
- 添加形状

Google地图应用 手势控制

Google地图还支持用户使用手势与地图进行交互。比如放大、缩小、移动、水平旋转、倾斜地图等。

zoom:

- Double tap to zoom
- Press and Tap to zoom out
- Press and drag to zoom in / out
- Spread to bring
- Pinch to zoom out

Displacement:

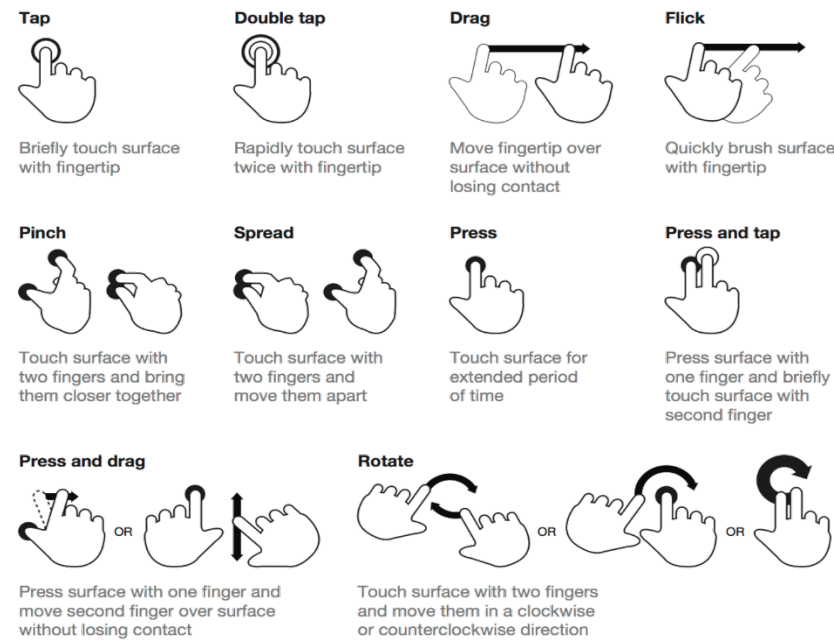
- Drag to continuously scroll the map omnidirectionally
- Flick to move in small portions omnidirectionally map

Inclination:

- Drag x 2 fingers vertically. Up tilts up 90 ° and down declines to 0 °

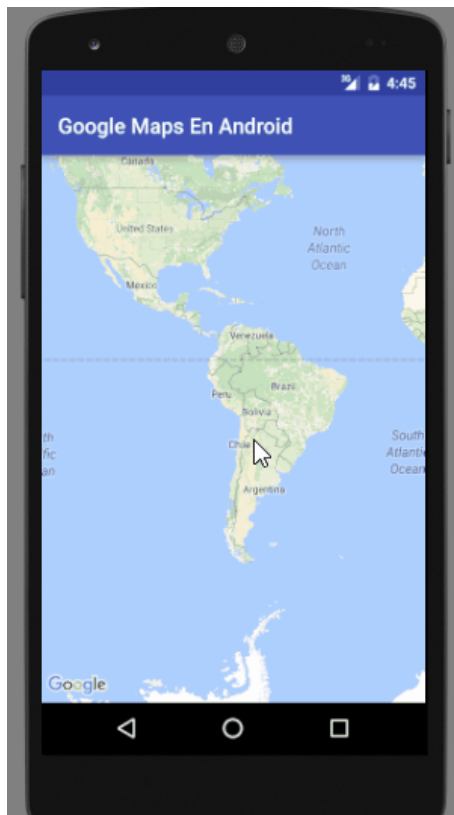
Rotation:

- Rotate to rotate the map for or against clockwise.

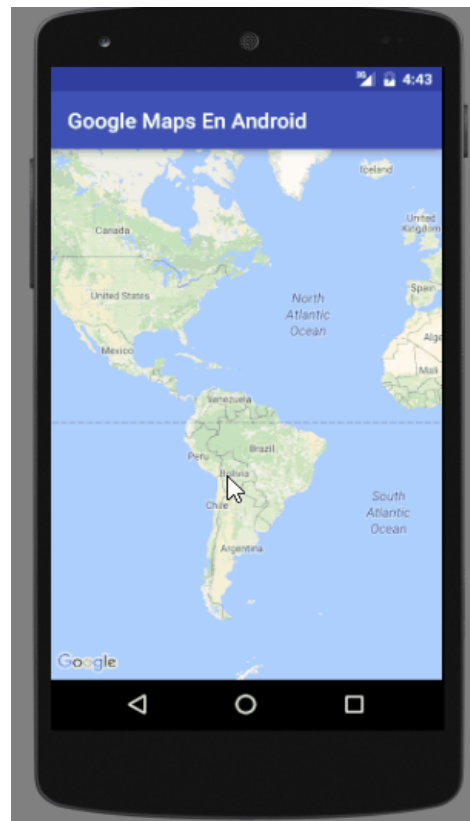


Google地图应用 手势控制

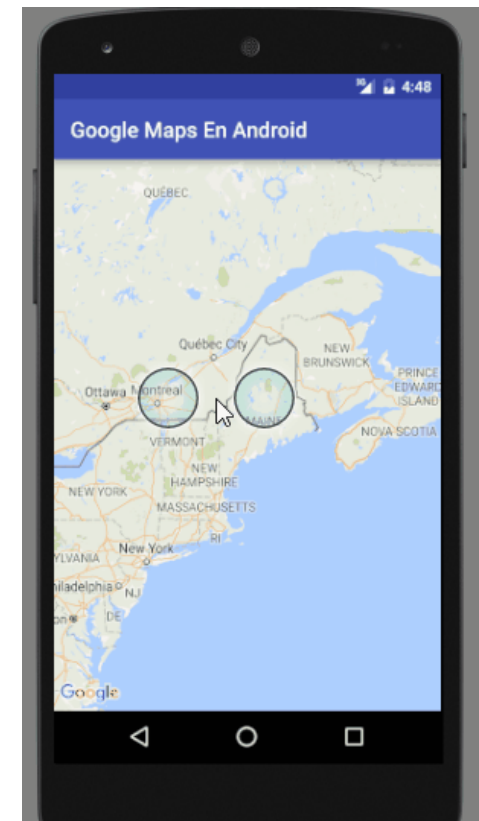
在模拟器中也可以模拟手势控制。以下操作都需要同时按住ctrl键。



放大缩小



旋转



垂直运动

地图开发

- Hello Map
- 地图属性
- 用户与地图交互
- **位置信息 MyLocation**
- 地图标记 Marker
- 添加形状

Google地图应用 当前位置 My Location 层

- Google Maps API 提供给 Android 设备的位置数据包括设备的当前位置（结合使用多种技术确定）、移动的方向和方式，以及设备移动范围是否覆盖预定义的地理边界（或称地理围栏）。
- Google Map API 通过**My Location** 层提供了一种在地图上显示设备位置的简单方式。它并不提供数据。
- 在启用 My Location 层之前，必须确保具备所需的**运行时位置权限**。即在程序运行时动态检查和申请位置权限。

Google地图应用 当前位置My Location 层

- 以下代码示例利用支持库检查权限，然后启用 My Location 层：

```
if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION)
    == PackageManager.PERMISSION_GRANTED) {
    mMap.setMyLocationEnabled(true);
} else {
    // Show rationale and request permission.
}
```

- 启用了 My Location 层时，**My Location 按钮**会出现在地图的右上角。当用户点击该按钮时，摄像头将设备的当前位置显示为地图的中心。



Google地图应用 当前位置 Location API

我们还可以使用Google Play services Location API 获取当前位置。

```
public class MainActivity extends ActionBarActivity implements
    ConnectionCallbacks, OnConnectionFailedListener {
    ...
    @Override
    public void onConnected(Bundle connectionHint) {
        mLastLocation = LocationServices.FusedLocationApi.getLastLocation(
            mGoogleApiClient);
        if (mLastLocation != null) {
            mLatitudeText.setText(String.valueOf(mLastLocation.getLatitude()));
            mLongitudeText.setText(String.valueOf(mLastLocation.getLongitude()));
        }
    }
}
```

Google地图应用 当前位置 Location API

获取Google Play services Location API 返回的Location对象，然后将其转化为**LatLng对象**才能够显示在地图上。

```
if (location != null) {  
    // Getting latitude of the current location  
    double latitude = location.getLatitude();  
  
    // Getting longitude of the current location  
    double longitude = location.getLongitude();  
  
    // Creating a LatLng object for the current location  
    LatLng latLng = new LatLng(latitude, longitude);  
}
```

此外，还需要将摄像头移动到LatLng对象所在位置才能正确显示

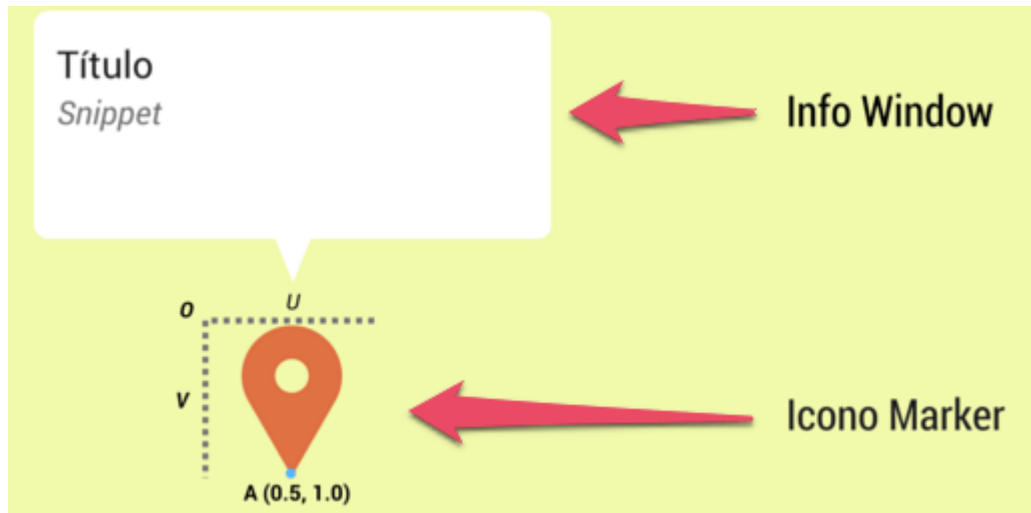
```
googleMap.moveCamera (CameraUpdateFactory.newLatLng (latLng));
```

地图开发

- Hello Map
- 地图属性
- 用户与地图交互
- 位置信息 MyLocation
- **地图标记 Marker**
- 添加形状

Google地图应用 标记 (Marker)

- **标记**用于标识地图上的单个位置。标记包括位置、标记图标、信息窗口、标题、详细信息等内容。



Google地图应用 标记

标记是 **Marker** 类型的对象，通过
GoogleMap.addMarker(markerOptions) 方法向地图添加。

```
// Sydney的经纬度  
LatLng sydney = new LatLng(-34, 151);  
mMap.addMarker(new MarkerOptions().position(sydney));
```



Google地图应用 标记

开发者可以根据程序需求，自定义标记的相关属性。

- 标记的颜色
- 标记的透明度
- 标记的图像（图标）
- 将标记平面化
- 旋转标记

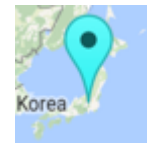
Google地图应用 标记

改变标记的颜色需要使用defaultMarker () 函数，传入的参数就是标记的颜色。其包括如下已定义的颜色。

- float HUE_AZURE
- float HUE_BLUE
- float HUE_CYAN
- float HUE_GREEN
- float HUE_MAGENTA
- float HUE_ORANGE
- float HUE_RED
- float HUE_ROSE
- float HUE_VIOLET
- float HUE_YELLOW

```
@Override
public void onMapReady (GoogleMap googleMap) {

    LatLng position = new LatLng(36.2048, 138.2529);
    googleMap.addMarker (new MarkerOptions ()
        .position (position)
        .title ( "Marker CYAN")
        .icon (BitmapDescriptorFactory.defaultMarker (BitmapDescriptorFactory .HUE_CYAN)));
    googleMap.moveCamera (CameraUpdateFactory.newLatLng (position));
}
```

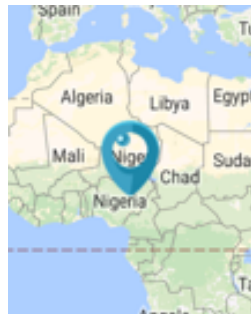


Google地图应用 标记

以下示例说明了如何更改标记的图标。

```
@Override
public void onMapReady (GoogleMap googleMap) {

    LatLng position = new LatLng (10, 10);
    googleMap.addMarker (new MarkerOptions ()
        .position (position)
        .title ( "marker with custom icon")
        .icon (BitmapDescriptorFactory.fromResource (R.drawable.marker_icon_pointer)));
    googleMap.moveCamera (CameraUpdateFactory.newLatLng (position));
}
```



Google地图应用 标记

- 标记设计为具有交互能力。它们默认接收 click 事件，通常与事件侦听器联用以调出**信息窗口**。
- 默认情况下，当用户点按标记时，地图工具栏出现在地图右下角，让用户可以快速访问 Google 地图移动应用。



Google地图应用 标记

此外，还可以使用`draggable()`函数设置标记可拖动。

```
@Override
public void onMapReady (GoogleMap googleMap) {

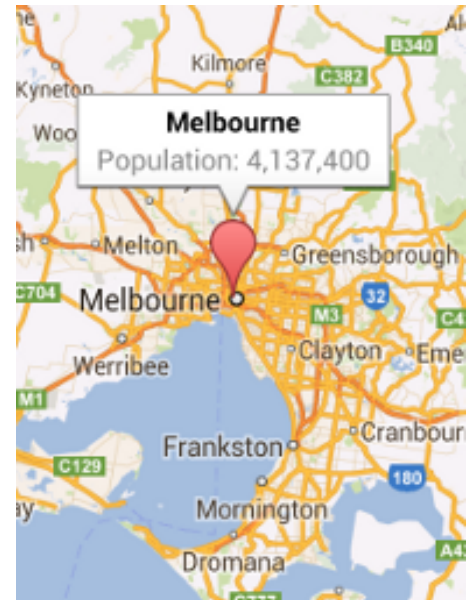
    LatLng position = new LatLng (15, 15);
    googleMap.addMarker (new MarkerOptions ()
        .position (position)
        .title ( "draggable marker")
        .draggable (true));
    googleMap.moveCamera (CameraUpdateFactory.newLatLng (position));
}
```



Google地图应用 信息窗口

- **信息窗口**在标记上方的弹出窗口中显示文本或图像。 信息窗口始终固定在标记上。 其默认行为是在用户点按标记时显示。
- 添加信息窗口最简单的方法是设置相应标记的 `title()` 方法和 `snippet()` 方法。

```
LatLng MELBOURNE = new LatLng(-37.813, 144.962);  
Marker melbourne = mMap.addMarker(new MarkerOptions()  
    .position(MELBOURNE)  
    .alpha(0.7f)  
    .title("Melbourne")  
    .snippet("Population: 4,137,400")  
    .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_AZURE)));
```



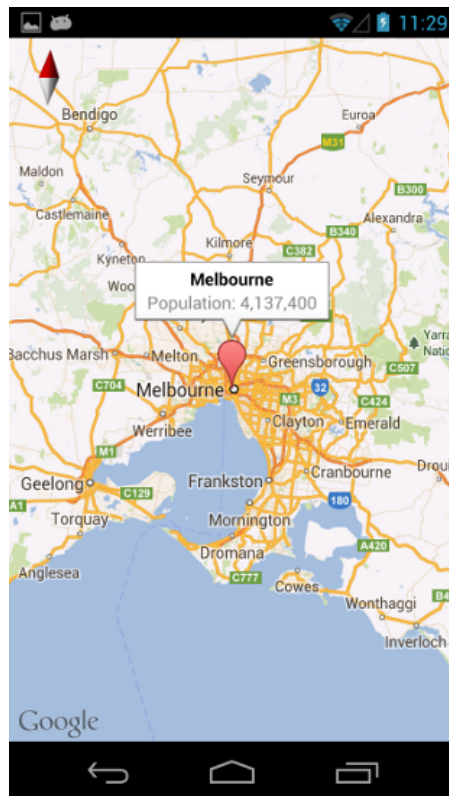
Google地图应用 信息窗口

- 在用户点按标记时，通过信息窗口向其显示信息。一次只能显示一个信息窗口。如果用户点击另一个标记，将隐藏当前信息窗口，并显示新的信息窗口。
- 可以通过对目标标记调用 **showInfoWindow()** 以编程方式显示信息窗口,调用 **hideInfoWindow()** 隐藏信息窗口。

```
LatLng MELBOURNE = new LatLng(-37.813, 144.962);
Marker melbourne = mMap.addMarker(new MarkerOptions()
    .position(MELBOURNE)
    .alpha(0.7f)
    .title("Melbourne")
    .snippet("Population: 4,137,400")
    .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_AZURE)));
melbourne.showInfoWindow();
melbourne.hideInfoWindow();
```

Google地图应用 信息窗口

还可以定制信息窗口的内容和设计。下图像显示的是一个默认信息窗口、一个包含定制内容的信息窗口以及一个包含定制框架和背景的信息窗口。



地图开发

- Hello Map
- 地图属性
- 用户与地图交互
- 位置信息 MyLocation
- 地图标记 Marker
- 添加形状

Google地图应用 形状

Google Maps API for Android 提供了一些简单的方法,向地图添加形状。包括：

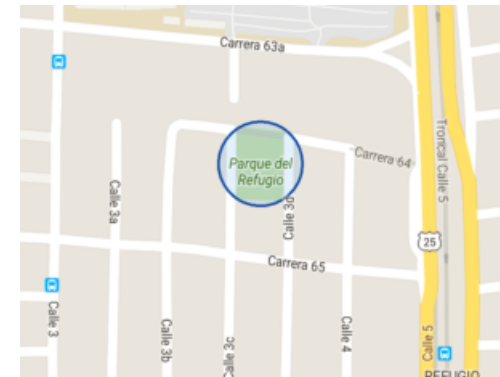
- **Polyline** 是一系列相连的线段，可组成您想要的任何形状，并可用于在地图上标记路径和路线
- **Polygon** 是一种封闭形状，可用于在地图上标记区域
- **Circle** 是投影在绘制于地图上的地球表面上地理位置准确的圆圈



Polyline



Polygon



Circle

Google地图应用 形状

以Polyline为例，说明向地图添加形状的方法：

- Polyline 对象包含一组LatLng 位置，它创建的一系列线段以有序方式将这些位置连接起来。
- 如需创建多段线，首先需要创建一个 **PolylineOptions** 对象并为其添加位置点，位置点以LatLng 对象形式表示。然后Google Maps API 按照向PolylineOptions对象添加点的顺序在各点之间绘制线段。

Google地图应用 形状

下面这段代码说明了如何向地图添加矩形：

```
// Instantiates a new Polyline object and adds points to define a rectangle
PolylineOptions sudamericaRect = new PolylineOptions ()
    .add (new LatLng (12.897489, -82.441406)) // P1
    .add (new LatLng (12.897489, -32.167969)) // P2
    .add ( new LatLng (-55.37911, -32.167969)) // P3
    .add (new LatLng (-55.37911, -82.441406)) // P4
    .add (new LatLng (12.897489, -82.441406)) // P1
    .color (Color.parseColor ( "# f44336")); // Red 500

Polyline polyline = mMap.addPolyline (sudamericaRect);
// Move camera
mMap.moveCamera(CameraUpdateFactory.newLatLng(new LatLng (-20.96144, -61.347656)));
```



参考资料

官网：<https://developers.google.com/maps/documentation/android-api/>

Tutorial：<http://www.hermosaprogramacion.com/2016/05/google-maps-android-api-v2/>

Google地图应用 v1 to V2

Google 在2012年发布了Google Maps API v2，其与v1有很大的不同。

- 设置地图的相关属性

v1中设置地图的相关属性需要通过**MapController**设置，MapController是MapView的控制器，可以控制MapView的显示中心和缩放级别等功能

- 增加标记

在v1中添加标记需要使用**Overlay**。Overlay是一个可显示于地图之上的可绘制的对象。

- 定义地图上的位置

在v1中需要将经纬度封装在**GeoPoint**类对象中。

- 等等。

QUESTIONS ?

