**使用MapView和MapActivity显示地图**

谷歌地图是谷歌服务中最受欢迎的后台搜索服务之一，它可以让你寻找到一切，比如，从纽约到旧金山 （仅 2,905 英里 ！）方向的最近的比萨饼客厅，还提供街道视图和卫星图像。

大多数的安卓设备，都整合了谷歌地图，这并不奇怪。而这样做的目的是，在主要安卓的发射器中有一个能直接对用户有用的显示地图的活动。但作为一个开发者，更多和你有关联的是MapView类和MapActivity类 ，这让地图融入你自己的应用程序中。你不仅可以显示地图，控制缩放级别，让人们平移，而且你也可以在安卓的基于位置的服务的（包括第32章）配合下，显示设备在哪以及往哪里去。

幸运的是，把基本显示功能纳入你的安卓项目是相当容易的。如果你想要获得想象的功能，你也有权利去增加你想象的功能。

**条款无情**

把谷歌地图纳入你自己的应用程序上一般需要一个相当长的法律术语集。其中包括一些可能使你不愉快的条款。  
 如果你正在考虑使用谷歌地图，请严密地审查这些条款来确定你的使用目的是否将无法运行、是否与其它各个条款相抵触。如果有任何潜在冲突的地方，强烈建议您寻求专业的法律条款。  
 此外，为了其他显示选项，根据其他地图数据的来源，请睁大你的眼睛去选择。如OpenStreetMap的(<http://www.openstreetmap.org/>)。

**添加项问题**

如同安卓l.5版本，谷歌地图不是安卓SDK严格的一部分。相反，它是谷歌应用程序界面的加载项，是普通SDK的延伸部分。安卓附加系统为其他子系统提供了挂钩。这可能是设备一部分而不是设备的全部。

注： 谷歌地图不是安卓开源项目的一部分，由于缺乏一些获得许可的问题有一些设备中并没有谷歌地图。例如，在写这篇文章的时候，人们所用的爱可视5的安卓系统并没有谷歌地图。

大体上，谷歌地图是加载项中的一个事实，这并不影响你日常的开发。然而，牢记以下：

1. 你将需要创建一个合适的项目指标，以确保谷歌地图的应用程序界面将是可用的。
2. 为了测试你完成了的谷歌地图，你还需要安卓运行的虚拟环境（AVD） ，支持谷歌地图API。

**基本要素**

最简单的方式得到一张地图，为您的应用程序创建自己的类MapActivity 的子类。像 MapActivity ListActivity ，在列表视图中占主导地位的活动背后，它进行了一些活动的包装，处理一些细微之处设立一个由MapView主导的活动。

在你为MapActivity子类设计布局时，你需要添加一个自定义的视图，在写这篇文章时用的是，com.google.android.maps.MapView 。这是普通平凑的方式拼出桌面小插件类的名称，包括完整的软件包名称与类名。这是必要的，因为MapView类是不在com.google.android.widget包中。你可以给你想要的任何MapView类部件的android ：id属性赋值，再加上处理所有的布局细节，与你的其它部件正确的呈现出来。

然而，你需要有这两个项目：

1. android：apiKey ，在创建中，将需要一个Google地图API密钥。
2. android:clickable="true"，如果你想让用户能够点击，并能够平移地图，设置clickable为“真”。

例如，从地图/ NooYawk的示例应用程序，这里是主要的布局：

<?xml version="l.0" encoding="utf-8f'?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout height="fill\_parent">

<com.google.android.maps.MapView android:id="@+id/map"

android:layout\_width= fill\_parent"

android:layout\_height="fill\_parent"

android:apiKey="<YOUR API\_KEY>"

android:clickable="true" />

</RelativeLayout>

在这一章的后面，“关键所在”这一节中，我们将讨论那个神秘API密钥。此外，在你的AndroidManifest.xml文件中，您将需要添加几个额外的东西：

（1）获得网络访问权限和获得应用访问范围(如WIFI)性的定位权限（后者与使用的MyLocationOverlay类，在本章后面介绍）

（2）在你的<application> 中，添加一个<uses-library>元素：  
android:name ="com.google.android.maps"，表明你正在使用一个可选的Android应用程序接口。

对于初学者这几乎所有的内容你都需要，再加上从你的创建类继承MapActivity类。如果你是闲来无事做，就新建该项目，并把它在模拟器中运行起来，你就会得到一个漂亮的世界地图。但是请注意，那是抽象的 MapActivity 类。您需要调用的是RouteDisplayed()方法表明你是否找到某个导航方向。

在理论上，用户可以在地图上使用方向键。然而，当他们查看整个世界的地图时，这不是十分的有用。

由于世界地图本身做的不是很好，我们需要添加一些东西，如下面的描述。

**练习控制**

你可以通过findViewById()方法找到你MapView中的部件，就像系统本身部件一样。在桌面插件本身提供了一个getMapController()方法。在MapView 类MapController之间，你有与之相持平的能力来决定地图展示什么以及怎样展示它。以下各节介绍变焦和中心点设置的功能，这些将是你最有可能要使用的特征。

**缩放**

起初你会觉得世界地图是相当宽广。通常，人们在手机上看地图时都希望什么范围变窄点，如一些城市的楼群。

你可以直接通过setZoom()方法对MapController控制缩放的级别。这里需要传一个整型参数，表示缩放的级别，其中1是对整个世界地图的直接显示和21是最精密的变焦镜头，你可以看到。每一个级别是一倍的有效分辨率：1赤道测量256个像素宽，而21赤道测量268435456像素宽。由于你的手机显示屏可能不会有268435456像素的任何尺寸，用户会看到一个小地图集中在一个小小的角落。 16水平将显示几个街区的城市在每一个层面，这可能是一个合理的起点实验。

如果你期望允许用户更改缩放级别，就调用setBuiltInZoomControls()方法并将参数设置为“真”，用户将能够在地图的底部中心通过控制进行变焦来放大和缩小地图。

**居中**

通常情况下，你需要对地图显示进行控制，如超出缩放级别，用户的当前位置或与你的活动中的一些数据保存的位置。要改变地图的位置，在MapController中调用setCenter ()方法，在给定的中心点GeoPoint上设置地图视图 。

将GeoPoint作为setCenter()方法的参数 。通过纬度和经度，一个GeoPoint代表一个位置。捕获的是，GeoPoint纬度和经度为整数，代表实际的纬度和经度乘以1e6。

这样就可以节省一点内存与存储单精度浮点型或双精度浮点型数据的位，这在安卓内部大大加快了对地图的一个位置GeoPoint所需要做的一些转换计算。然而，这就意味着你必须记住由le6乘以现实世界的经度和纬度。

**地形起伏**

正如你在你的全屏电脑上使用谷歌地图服务一样可以显示卫星图像。这在安卓手机地图一样可以实现。

MapView类中提供toggleSatellite()方法，其中，顾名思义，从显示区域进行切换或者关闭卫星的角度查看。你可以让用户通过选项菜单或触发按下监听事件进行切换，在项目NooYawk中，设置按键按下事件方法用onKeyDown()方法。

**层上加层**

如果你曾经使用过的全尺寸版本的谷歌地图，你可能看到地图本身上面覆盖的东西，如指示附近被搜索的企业的位置图钉。在地图界的说法（ 和，对于这个问题，在许多严谨的图形编辑器都有），该图钉与图层分开的地图本身比，你看到的是图层上面的图钉层组成的独立层。

安卓的地图显示允许你创建层为好，这样你可以在你的地图上做标记，正如在你的应用程序中你需要根据用户输入目的地进行选择。例如，NooYawk项目中使用一个图层显示选择的建筑物位于曼哈顿岛。

**Overlay类**

要对任何地图添加覆盖，需要作为一个子类进行覆盖。如果你正在寻找添加图钉或类似的子类进行覆盖的话，有一个可用ItemizedOverlay子类，ItemizedOverlay简化了这一过程。

附加一个覆盖类到你的地图，我们在这里做一个自定义SitesOverlay 类，只需通过实例对象调用你的MapView类中getOverlays()方法和add()方法进行覆盖。

我们将在下一节细看该标记。

**绘制ItemizedOverlay**

顾名思义，在ItemizedOverlay中允许你提供的兴趣点列表上显示地图，具体而言，新建一个OverlayItem实例。为你覆盖处理大部分绘制逻辑。下面是这项工作最基本的步骤：

1. 覆盖ItemizedOverlay <OverlayItem>作为自己的子类（在本文这个例子中用的名字是 SitesOverlay ）。

2. 在构造函数中，创建你项目中OverlayItem类的对象，并调用populate()方法，他们这是准备用于覆盖的。

3. 实现size()方法返回被覆盖处理的项目数。

4. 覆盖createItem()方法返回索引的OverlayItem对象。

5. 当你Itemizedoverlay类实例化对象，它提供一个图像表示默认图标（例如，一个图钉）显示为每个项目。

由NooYawk构造的标记是可绘制用于第5步。它显示一个图钉。

您可能还希望做一个更好的工作覆盖draw()方法来处理标记的阴影。而你将处理地图上的标记的影子，看来你需要提供一点援助，才知道图标的底部，所以可以得出适当绘制阴影。

**处理屏幕单击事件**

子类也可以覆盖实现onTop()方法 ，当用户点击地图时得到通知，所以可以调整对它的绘制覆盖。例如，在全尺寸的“谷歌地图”中，点击一个图钉弹出一个与图钉针尖的位置相关的信息。与onTop()方法类似 ，在安卓中你可以做很多类似的设置。

onTop()方法是ItemizedOverlay收到的被点击OverlayItem的事件处理。带着这些事件，这取决你来做一些值得事情。

在SitesOverlay的情况，如前一节中所示，onTop()方法看起来像这样：

protected boolean onTap(int i) {

Toast.makeText(NooYawk.this, items.get(i).getSnippet(),Toast.LENGTH\_SHORT).show();

return(true),}

在这里，我们只是设置了一个简短的片段从OverlayItem上弹出式显示，返回真值，表示我们处理的响应器。

**MyLocationOverlay**

安卓有一个内置的叠加处理有两种常见的情景：

（1）Gps显示你在地图上或其他位置提供逻辑基础。

（2）显示你在哪里标记，其中根据内置可用的罗盘传感器。

所以您需要做的就是创建一个MyLocationOverlay类的实例化对象，将它添加到您的MapView类的覆盖列表，在适当的时候，启用和禁用所需的功能，“在适当的时刻”的概念是电池最长使用时间。有没有暂停时更新位置或方向的活动方法，所以建议你用onResume ()方法在启用这些功能，并禁止他们时使用onPause ()方法 。

例如，在NooYawk项目MyLocationOverlay将显示一个罗盘使用。要做到这一点，我们首先需要创建一个对象添加实现覆盖方法，并把它添加到列表中：

me=new MyLocationOverlay(this, map);

map.getOverlays().add(me);

然后我们用onResume()方法和onPause()方法启用和关闭罗盘:

**关键所在**

如果你已经下载了本书的源代码，并编译NooYawk项目，把上它安装在你的模拟器中，并运行该项目，你可能会看到一个网格屏幕和一个图钉，但没有实际的地图。

这是因为在源代码中的API密钥在你的开发机器上是无效的。相反，你需要生成你自己的API密钥（S）使用在你的应用程序中。

可以发现在安卓的网站（http://code.google.com/android/add-ons/google-apis/ mapkey.html）上有开发和生产使用的API密钥生成的完整说明。在简洁起见，让我们更专注于狭窄的情况下获得NooYawk项目在模拟器上运行所需的密钥。这样做需要以下步骤：

（1）访问的API密钥的注册页面，并查看服务条款。

（2）重读这些服务条款，让你确定你是否真的确定要同意他们的条款。

（3）找到MD5摘要签署调试模式的应用程序使用的证书。

（4）API密钥的注册页面上，粘贴在该MD5签名并提交表格。

（5）在结果页上，复制的API密钥，将其粘贴在您的MapView使用布局的 apiKey的值上。

最棘手的部分是找到MDS签名用于签名的证书的调试应用程序。事实上，许多复杂的事情仅仅是在意义上的概念。

所有的安卓应用程序都使用数字签名生成证书。当您设置你的SDK时会自动给出一个调试证书 ，并有一个单独的进程，创建了在开发应用程序中使用自签名的证书。这个签名的过程中涉及使用的密钥和证书管理工具和JDK中包含的用于JAR文件签名和验证的工具公共的工具。这些的宗旨就是，你只需要担心你的密钥就可以了。

你的调试证书的MD5摘要，如果你的电脑是苹果操作系统（Mac OS X）或Linux ，使用下面的命令：

keytool -list -alias androiddebugkey –keystore ~／.android/debug.keystore -storepass

android -keypass android

在其他开发平台上，您将需要更换你的平台和用户的密钥的位置：

(1) 在Windows XP操作系统中，使用命令C：\文件和设置\ <用户> \android 根 目录\ debug.keystore 。  
  (2) 在Windows Vista / Windows 7操作系统在中，使用C：\用户\ <user> \android根目录\ debug.keystore(<user>是您的账户名）。

输出的第二行就包含你的MD5的摘要，由冒号分隔的一系列十六进制数字对作为一个MD5。