

Google Earth API

开发者指南



制作：[Google Earth 博客](#)

版权申明：本手册的中文原版版权为 Google(TM) 公司所有

转载请采用《[知识共享署名-非商业性使用-相同方式共享 2.5 中国大陆许可协议](#)》

欢迎使用 Google Earth API 开发者文档。通过使用 Google Earth 插件及其 API，您可以在网页中嵌入全功能的 Google Earth 及其三维视图功能。您可以像在二维的 Google Maps 上一样，在三维版本上绘制标记和线条。

由于是全新的 API，所以可能会存在缺陷，文档也不尽完美。与 Google 一起努力使其日臻完善吧！

第 1 章：简单介绍

1、目标读者

本文档专门针对熟悉 JavaScript 编程和面向对象的程序设计概念的人员而设计。Google 地球 API 是以多种 KML 元素为基础，如果您对 KML 不熟悉，请参阅 Google 地球 KML 文档。

2、Google Earth API 简介

您可以单独使用 Google Earth API，也可以将其同 Google Maps API 结合使用。如果您的网页中嵌入了 Google Maps，您可以让用户选择在 Google Maps 中还是在 Google Earth 浏览器中查看内容来以对网页进行增效。您可以在同一网页上执行该操作。例如，Google Maps 网页以二维展示信息。集成 Google Earth 插件后，用户可以选择以三维模式显示信息，而且还可以观看到地形、几何图案及其他众多的 Google Earth 独有的功能。

3、关于本文档中的示例

本文档中的大多数示例仅显示了相关的 JavaScript 代码，并未显示完整的 HTML 文件。您可以将 JavaScript 代码植入自己的 HTML 框架文件，也可以点击每个示例后面的链接，下载完整的 HTML 文件。Google Earth 插件自带的 Sample.html 文件也可作为开始的基点。

4、安装 Google Earth 插件

使用了 Google Earth API 开发的网站需要采用安装了 Google Earth 插件的网页浏览器才可以浏览，在安装 Google Earth 插件之前，请阅读以下编程注释，以确保您的应用程序能够跨平台工作。

本版本的 Google Earth 插件仅支持 Microsoft Windows XP 和 Vista 及 Mac OS 操作系统。对其他操作系统的支持计划在未来版本中实现。

Google Earth 插件当前支持的网页浏览器有:

- IE 6.0+
- IE 7.0+
- Firefox 2.x 或 2.0x 或 Firefox 3.0
- Netscape 7.1+
- Mozilla 1.4+
- Flock 1.0+
- Google Chrome

要安装 Google Earth 插件, 请:

- 1、注册并获取 [Google Earth API 密钥](#)。
- 2、下载并运行 [Google Earth 插件安装程序](#)。
- 3、默认情况下, 插件及相关文件会安装在 C:\Program Files\Google Earth Plugin 目录。

5、 卸载 Google Earth 插件

要卸载 Google Earth 插件, 请从开始菜单选择 "Google Earth Plugin", 然后选择 "Uninstall Google Earth Plugin"。

6、 使用 Samples.html

Samples.html 文件 (<http://www.google.com/earth/plugins/examples/samples/index.html>) 包含很多有用的 JavaScript 示例。您可以看到 JavaScript 代码, 并可从网页上直接运行这些代码。在页面上可以立即看到 Google Earth 3D 视图的变化。也可以使用该页面来测试您的 JavaScript 代码。

7、 在网页中嵌入 Google Earth

Google Earth 插件中包含了若干示例文件以及入门使用的模板。以下示例显示了用于在网页中嵌入 Google Earth 3D 视图的 HTML 和 JavaScript。您可以根据以下示例创建自己的应用程序。

```
<html>
<head>
  <title>Google</title>
<!-- *** 使用您自己的 API 密钥替换下面的密钥
(要获取密钥, 请访问 http://code.google.com/apis/maps/signup.html) *** -->
```

```
<script type="text/javascript"
  src="http://maps.google.com/maps?file=api&v=2&key=abcdefg"></script>
<script type="text/javascript">
google.load("earth", "1");
google.load("maps", "2.99"); // 用于 JS 地理编码器

var ge = null;
var geocoder;

function el(e) { return document.getElementById(e); }

function init() {
  geocoder = new GClientGeocoder();
  google.earth.createInstance("map3d", initCB, failureCB);
}

function initCB(object) {
  ge = object;
  ge.getWindow().setVisibility(true);
}

function failureCB(object) {
  alert('未能加载');
}

function submitLocation() {
  var address = el('address').value;
  geocoder.getLatLng(
    address,
    function(point) {
      if (point && ge != null) {
        var la = ge.createLookAt('');
        la.set(point.y, point.x, 100, ge.ALTITUDE_RELATIVE_TO_GROUND,
              0, 0, 4000);
        ge.getView().setAbstractView(la);
      }
    }
  );
}

</script>
</head>
<body onload='init()' id='body'>
  <div style='background:#ffe6e6; font-weight:bold; font-size: 16pt; font-family: arial;'>
```

```
<center>Google 机密</center>

</div>

<div>
  <form name='searchform' id='searchform' action='javascript:submitLocation();void(0);'>
    <input type='text' size=60 id='address'></input>
    <input type='submit' value='转到位置'>
  </form>
</div>

<div id='map3d_container' style='border: 1px solid silver; height: 500px;'>

  <div id='map3d' style='height: 100%;'></div>
</div>
</body>
</html>
```

8、 故障排除

如果您的代码不能正常工作， 可尝试通过以下方法解决问题：

- 确认 API 密钥是否有效。
- 检查有无拼写错误。注意，JavaScript 区分大小写。
- 使用 JavaScript 调试工具。在 Firefox 中，您可以使用 JavaScript 控制台、[Venkman Debugger](#) 或 [Firebug](#) 插件。在 IE 中，您可以使用 [Microsoft Script Debugger](#)。
- 搜索[开发者论坛](#)。如果未找到能够回答您问题的帖子， 请将您的问题张贴到论坛并附上问题演示网页的链接。

9、 更多资源

如果您需要了解更多关于 Google Earth、KML 或者 Google Maps 的信息，请参考：

- <http://code.google.com/apis/kml>
- <http://code.google.com/apis/maps>
- <http://earth.google.com>

第 2 章：基础主题

1、 示例：“你好，地球”

要了解 Google Earth API，最简单方法莫过于观看简单的示例。下面的 JavaScript 代码会创建包含 Google Earth 3D 视图的网页，当运行后，您可以尝试使用操作 Google Earth 的方法操作网页中的视图，看效果是否一样：

```
<html>
  <head>
    <title>你好 Google 地球! </title>
    <script src="http://www.google.com/jsapi?key=ABCDEF"></script>
    <script>
google.load("earth", "1");

var ge = null;

function init() {
  google.earth.createInstance("map3d", initCallback, failureCallback);
}

function initCallback(object) {
  ge = object;
  ge.getWindow().setVisibility(true);
}

function failureCallback(object) {
}

    </script>
  </head>
  <body onload='init()' id='body'>
    <center>

      <div>
        你好，地球!
      </div>

      <div id='map3d_container'
        style='border: 1px solid silver; height: 600px; width: 800px;'>
```

```
<div id='map3d' style='height: 100%;'>&/div>
</div>
</center>
</body>
</html>
```

您可以在线查看本示例，也可以下载、编辑和运行该示例，但您必须将文件中的密钥替换为您自己的地图 API 密钥。（如果您为某个特定的目录注册了密钥，则该密钥同样可用于所有子目录）即使是最简单的示例，您也需要完成以下四步方能在网页中加载 Google Earth 插件：

- 1、使用 **script** 标签包含 Google Earth API 的 JavaScript。
- 2、创建 **div** 元素包含 Google Earth 。
- 3、从 **body** 标签的 **onLoad** 事件初始化对象。
- 4、编写 JavaScript 函数创建“Google Earth”对象。

这些步骤详解如下：

1、加载 Google Earth API 的 JavaScript 文件

<http://www.google.com/jsapi?key=ABCDEF> URL 指向 JavaScript 文件位置，该文件包含使用 Google Earth API 所需的所有符号和定义。您的网页中必须包含一个指向该 URL 的脚本标签（使用注册 API 时获取的密钥）。在此例中，密钥为“ABCDEF”。

2、创建 div 元素包含 Google Earth

```
<div id='map3d_container'
  style='border: 1px solid silver; height: 600px; width: 800px;'>
  <div id='map3d' style='height: 100%;'>&/div>
```

要在网页上显示 Google Earth，必须为其预留一定的空间。具体是创建一个名为 **div** 的元素，并在浏览器的文档对象模型（DOM）中获取对该元素的引用。以上示例中定义了名为“map_3d_container”的 **div** 元素并使用 **style** 属性定义了其大小。

3、从 body 标签的 onLoad 事件初始化对象

```
<body onload='init()' id='body'>
```

当 HTML 网页呈现时，文档对象模型（DOM）就会建置好，而外部图像和脚本也会接收并合并到 **document** 对象中。为保证地图只在页面完全加载后才置于页面上，我们仅在 HTML 网页的 **<body>** 元素接收到 **onload** 事件时才执行构建 **ge** 对象的函数。这样就可以避免意外的浏览器行为，并且能够更好地控制地图绘制的时间和方式。

4、编写 JavaScript 函数创建“Google Earth”对象

4.1、创建地标

在 Google Earth 中，经常会用到地标这一方便使用的功能。它会使用图钉作为图标，在地球表面标记出位置。通常，地标所标记的是用户正在查看的位置。您可以更改地标的名称，使用自定义的图标，还可以为地标添加其他几何属性。

以下示例创建了一个地标，标出 Google 园区在地球表面上的位置：

```
var placemark = ge.createPlacemark('');
placemark.setName("您正身处 Google");
ge.getFeatures().appendChild(placemark);

// 为地标创建样式映射
var normal = ge.createIcon('');
normal.setHref('http://maps.google.com/mapfiles/kml/paddle/red-circle.png');
var iconNormal = ge.createStyle('');
iconNormal.getIconStyle().setIcon(normal);
var highlight = ge.createIcon('');
highlight.setHref('http://maps.google.com/mapfiles/kml/paddle/red-circle.png');
var iconHighlight = ge.createStyle('');
iconHighlight.getIconStyle().setIcon(highlight);
var styleMap = ge.createStyleMap('');
styleMap.setNormalStyle(iconNormal);
styleMap.setHighlightStyle(iconHighlight);
placemark.setStyleSelector(styleMap);

// 创建点
var la = ge.getView().copyAsLookAt(ge.ALTITUDE_RELATIVE_TO_GROUND);
var point = ge.createPoint('');
point.setLatitude(la.getLatitude());
point.setLongitude(la.getLongitude());
placemark.setGeometry(point);
```

地标在 Google Earth 中显示为：



上例中通过指定了下列属性来创建地标：

- 用于标识地标的名称。在上例中，名称为“`You are at Google`”。
- 定义地标在地球表面位置的点 — 经度、纬度及高度（可选）。在上一个示例中，经度和纬度由 `LookAt` 位置确定。`LookAt` 位置定义了一个虚拟镜头。`LookAt` 方法会针对目前正在查看的对象定位“镜头”。

4.2、添加气泡窗口

创建地标后，您可以使用气泡窗口对其进行描述。气泡窗口里的描述内容可以使用 HTML。您可以创建三种类型的气泡窗口：

- 功能气泡窗口
- 字符串气泡窗口
- DIV 气泡窗口

以下示例创建了功能气泡窗口：

```
var b = ge.createFeatureBalloon('');  
b.setFeature(marker);  
b.setMaxWidth(800);  
ge.setBalloon(b);
```

以下示例创建了标记珠穆朗玛峰的信息窗口：

```
Var b = ge.createHtmlStringBalloon('');  
b.setMaxWidth(300);  
b.setFeature(feature);  
b.setContentString (  
'<b.setContentString (
```

```
'' +  
'<font size=20>珠穆朗玛</font><br><font size=20>世界之巅' + 'window</font>');  
ge.setBalloon(b);
```

上面的示例在 Google 地球中显示为:



以下示例创建了 DIV 气泡窗口:

```
var b = ge.createHtmlDivBalloon('');  
b.setMaxWidth(800);  
b.setFeature(feature);  
var div = document.createElement('DIV');  
div.innerHTML =  
    '<br>' +  
    '<a href="http://www.google.com/googlegulp/">Google Gulp</a>';  
b.setContentDiv(div);  
ge.setBalloon(b);
```

请注意: 您可以通过加入图片、更改字体等操作来自定义信息窗口的外观和风格。

要关闭信息窗口, 请使用以下 JavaScript 代码:

```
ge.setBalloon(null);
```

4.3、移动镜头

您可以使用 LookAt 对象确定地球上正在查看的点、景点与视点间的距离, 以及观看的角度。

以下示例例将镜头向北移动了 7 个纬度, 向东移动了 7 个经度:

```
var lookAt = ge.getView().copyAsLookAt(ge.ALTITUDE_RELATIVE_TO_GROUND);
lookAt.setLatitude(lookAt.getLatitude() + 7.0);
lookAt.setLongitude(lookAt.getLongitude() + 7.0);
ge.getView().setAbstractView(lookAt);
```

4.4、创建路径

在 Google Earth 中可以创建多种类型的路径，并且能够利用您的数据衍生出多种创意。在 `lineString` 对象中定义相连的线段可以创建路径。使用 `LineString` 对象时，您需要指定是否将 `LineString` 连接至地面。`Tessellate` 属性会将线条分割为更小的分段。以下示例创建了一条白色的锯齿状路径：

```
var lineString;

lineString = ge.createLineString('');

var lineStringPlacemark = ge.createPlacemark('');

lineStringPlacemark.setGeometry(lineString);

lineString.setTessellate(true);

ge.getDocument().getFeatures().appendChild(lineStringPlacemark);

addToLineString(lineString, 0, 0, 0);

addToLineString(lineString, .1, .05, 0);

addToLineString(lineString, 0, .10, 0);

addToLineString(lineString, .1, .15, 0);

addToLineString(lineString, 0, .20, 0);

addToLineString(lineString, .1, .25, 0);
```

在 Google Earth 中的路径显示为：



4.5、更改路径样式

您可以通过定义颜色和宽度来更改路径的外观。下面的 JavaScript 代码会检查定义的路径，增加上一示例中路径的宽度，并将颜色改为蓝色。

```
var lineStyle = lineStringPlacemark.getStyleSelector().getLineStyle();  
lineStyle.setWidth(lineStyle.getWidth() + 2);  
lineStyle.getColor().set('66ff0000');
```

新路径在 Google Earth 中显示为：



4.6、创建多边形

您可以使用多边形来创建简单的建筑物及其他图形。以下示例中，先通过设置外框创建了一个白色的正方形，然后再设置正方形的内框。

```
var polygonPlacemark = ge.createPlacemark('');  
polygonPlacemark.setGeometry(ge.createPolygon(''));  
var outer = ge.createLinearRing('');
```

```
polygonPlacemark.getGeometry().setOuterBoundary(outer);
ge.getDocument().getFeatures().appendChild(polygonPlacemark);

// 正方形外框
var center = ge.createLookAt('');
center = ge.getView().copyAsLookAt(ge.ALTITUDE_RELATIVE_TO_GROUND);
var coords = outer.getCoordinates();
var lat = center.getLatitude();
var lon = center.getLongitude();
coords.pushLatLngAlt(lat - .05, lon - .05, 0);
coords.pushLatLngAlt(lat - .05, lon + .05, 0);
coords.pushLatLngAlt(lat + .05, lon + .05, 0);
coords.pushLatLngAlt(lat + .05, lon - .05, 0);

// 创建正方形内框
polygonPlacemark.getGeometry().getInnerBoundaries().
  appendChild(ge.createLinearRing(''));
coords = polygonPlacemark.getGeometry().getInnerBoundaries().getFirstChild().
  getCoordinates();
coords.pushLatLngAlt(lat - .02, lon - .02, 0);
coords.pushLatLngAlt(lat - .02, lon + .02, 0);
coords.pushLatLngAlt(lat + .02, lon + .02, 0);
coords.pushLatLngAlt(lat + .02, lon - .02, 0);
```

多边形在 Google Earth 中显示为:



4.7、更改多边形的样式

与路径一样，您也可以通过指定颜色和宽度来更改多边形的颜色和宽度：

```
var lineStyle = polygonPlacemark.getStyleSelector().getLineStyle();
lineStyle.setWidth(lineStyle.getWidth() + 2);
lineStyle.getColor().set('66ff0000');
polygonPlacemark.getStyleSelector().getPolyStyle().getColor().set('660000ff');
```

现在是一个边框为蓝色的红色多边形：



4.8、使用网络链接

网络链接包含用于加载文件且带有 href (超链接引用) 的链接属性。链接中的 href 可指定以下任意对象的位置：

- 图标使用的图像文件，用于图标样式、地面叠加层和屏幕叠加层
- Model 对象中使用的模型文件
- 网络链接加载的 KML 或 KMZ 文件

指定的文件可以是本地文件，也可以是远程服务器上的文件。网络链接的最简单形式是将一个大的 KML 文件分割为同一台计算机上的较小的、更易于管理的多个文件。以下示例创建了网络链接：

```
var networkLink = ge.createNetworkLink("");
networkLink.setDescription("打开已获取内容的网络链接");
networkLink.setName("打开网络链接");
networkLink.setFlyToView(true);
var link = ge.createLink("");
```

```
link.setHref("http://kml-samples.googlecode.com" +
            "/svn/trunk/kml/NetworkLink/placemark.kml");
networkLink.setLink(link);
ge.getFeatures().appendChild(networkLink);
```

4.9、管理事件

浏览器中的 JavaScript 是由事件驱动的，这就是说 JavaScript 会通过生成事件来对交互作出响应，并会找一个程序来对目标事件进行侦听。例如，在浏览器中，用户鼠标和键盘交互操作会创建在 DOM 中传播的事件。捕捉特定事件的程序将为这些事件注册 JavaScript 事件侦听器，并在接收到这些事件时执行代码。

Google Earth API 中的事件会通过 GEvent 命名空间中使用 utility 函数注册事件侦听器来进行处理。每个 Google 地球 API 对象会输出一定数量的命名事件。例如，KmlPlacemark 对象输出单击、双击、移动等一系列事件。每个事件均会在一定的环境中发生，并能够传递参数用于识别环境。例如，当用户在 Google Earth 中移动鼠标时会触发 mousemove 事件。以下示例的代码演示了如何管理事件：

```
function myEventListener(kmlEvent) {
    var targetType = kmlEvent.getTarget().getType();
    var currentTargetType = kmlEvent.getCurrentTarget().getType();
    var button = kmlEvent.getButton();
    var clientX = kmlEvent.getClientX();
    var clientY = kmlEvent.getClientY();
    var screenX = kmlEvent.getScreenX();
    var screenY = kmlEvent.getScreenY();
    var latitude = kmlEvent.getLatitude();
    var longitude = kmlEvent.getLongitude();
    var altitude = kmlEvent.getAltitude();
    var didHitGlobe = kmlEvent.getDidHitGlobe();
    var altKey = kmlEvent.getAltKey();
    var ctrlKey = kmlEvent.getCtrlKey();
    var shiftKey = kmlEvent.getShiftKey();
    var timeStamp = kmlEvent.getTimeStamp();
}

// 用户点击地标时会触发 myEventListener(), 并传入 KmlEvent 对象。
google.earth.addEventListener(placemark, "mousedown", myEventListener);

// 用户点击地球仪时会触发 myEventListener(), 并传入 KmlEvent 对象。
google.earth.addEventListener(ge.getGlobe(), "mousedown", myEventListener);

// 用户点击窗口中任意位置时会触发 myEventListener(), 并传入 KmlEvent 对象。
google.earth.addEventListener(ge.getWindow(), "mousedown", myEventListener);
```

第 3 章：高级主题

1、观看天空

您可以创建应用程序来显示天空中的物体，例如星体、星座、行星、月亮和银河。当应用程序切换至天空模式时，Google Earth 会转换为显示用户当前所处地球位置的天空(天顶)。天文数据会映射至环绕地球的虚拟球体的内面之上。

坐标

天文坐标以赤经 (RA) 和赤维进行描述。赤经对应于经度，代表到春分时太阳和天球赤道交点的距离。赤经按 0 到 24 小时进行度量，一小时 RA 等于地球上某点之上的天空在一小时内的旋转量。零小时 RA 则位于春分点，RA 自该点向东逐步递增。下列示例中，地图从显示地球更改为显示天空。

```
if (!('isSkyMode' in window) || !window.isSkyMode) {
  window.isSkyMode = true;
  ge.getOptions().setMapType(ge.MAP_TYPE_SKY);
} else {
  window.isSkyMode = false;
  ge.getOptions().setMapType(ge.MAP_TYPE_EARTH);
}
```

2、创建地面叠加层

通过地面叠加层，可在 Google Earth 的地形上叠加图像。Icon 对象包含了指向叠加图像 JPEG 文件的链接。

```
var groundOverlay = ge.createGroundOverlay('');
groundOverlay.setIcon(ge.createIcon(''))
groundOverlay.setIcon().
  setHref("http://www.google.com/intl/en_ALL/images/logo.gif");
groundOverlay.setLatLonBox(ge.createLatLonBox(''));

var center = ge.getView().copyAsLookAt(ge.ALTITUDE_RELATIVE_TO_GROUND);
var north = center.getLatitude() + .85;
var south = center.getLatitude() - .85;
var east = center.getLongitude() + .55;
var west = center.getLongitude() - .55;
```

```
var rotation = 0;
var latLonBox = groundOverlay.getLatLonBox();
latLonBox.setBox(north, south, east, west, rotation);

ge.getFeatures().appendChild(groundOverlay);
```

在本例中，Google 的 Logo 被放置在加利福尼亚州的山景城总部上。如下图：



3、创建屏幕叠加层

屏幕叠加层是固定于屏幕上的图像叠加。ScreenOverlay 可用于在 Google Earth 中创建罗盘、徽标和预览显示。ScreenOverlay 的大小由 size 属性决定。叠加层的定位是通过将 overlayXY 属性指定的图像中的点映射到 screenXY 属性指定的屏幕上的点来处理的。然后，图像会绕屏幕相关点进行旋转，旋转角度由 rotationXY 属性指定。以下示例会在屏幕上叠加 Google 徽标：

```
var screenOverlay = ge.createScreenOverlay('');
screenOverlay.setIcon(ge.createIcon(''));
screenOverlay.getIcon().
    setHref("http://www.google.com/intl/en_ALL/images/logo.gif");

// 设置屏幕位置（像素）
screenOverlay.getOverlayXY().setXUnits(ge.UNITS_PIXELS);
screenOverlay.getOverlayXY().setYUnits(ge.UNITS_PIXELS);
screenOverlay.getOverlayXY().setX(400);
screenOverlay.getOverlayXY().setY(200);

// 绕对象中心点旋转
screenOverlay.getRotationXY().setXUnits(ge.UNITS_FRACTION);
screenOverlay.getRotationXY().setYUnits(ge.UNITS_FRACTION);
screenOverlay.getRotationXY().setX(0.5);
screenOverlay.getRotationXY().setY(0.5);

// 设置对象尺寸（像素）
screenOverlay.getSize().setXUnits(ge.UNITS_PIXELS);
```

```
screenOverlay.setSize().setYUnits(ge.UNITS_PIXELS);
screenOverlay.setSize().setX(300);
screenOverlay.setSize().setY(75);

// 旋转 45 度
screenOverlay.setRotation(45);

ge.getFeatures().appendChild(screenOverlay);
```

在 Google Earth 中显示为:



4、管理地形、道路和边界

默认情况下, Google Earth 插件第一次加载时仅会显示地形层。在 Google Earth 3D 视图中可用的层在插件中并非都同样可用。以下 JavaScript 代码展示了如何遍历插件中可用的层。

```
function TraverseLayers(list) {
  for (var i = 0; i < list.getLength(); ++i) {
    var child = list.item(i);
    var type = child.getType();
    try {
      alert("FOLDER=" + child.getName() + " UUID=" + "" +
        child.getId() + "");
      TraverseLayers(child.getChildNodes());
    } catch(err) {
    }
  }
}

var layers = ge.getLayerRoot();
var tree = layers.getChildNodes();
TraverseLayers(tree);
```

下面的 JavaScript 示例使用了插件的预定义宏 LAYER_BUILDINGS:

```
function getInheritedVisibility(layer) {
  if (layer.getVisibility() == false) {
    return false;
  } else {
    var parent = layer.getParentNode();
    if (!parent) {
      return true;
    }
    return getInheritedVisibility(parent);
  }
}

var buildingsLayer = ge.getLayerRoot().getLayerById(ge.LAYER_BUILDINGS);
var inheritedVisibility = getInheritedVisibility(buildingsLayer);
ge.getLayerRoot().enableLayerById(ge.LAYER_BUILDINGS, !inheritedVisibility);
```

您可以通过在 dbRoot 中指定 UUID 来取代宏 LAYER_BORDERS。

5、 使用样式映射

StyleMap 可在两种不同的样式之间进行映射。通常情况下，StyleMap 对象用于为地标提供独立的普通和突出显示样式，以使用户将鼠标移到 Google 地球中的图标上时会显示突出显示的版本。

在以下示例中，地标先设置为三角形，然后当用户的鼠标移过时，地标会变为正方形。样式映射可指定用于地标的普通图标和突出显示图标：

```
map = ge.createStyleMap('styleMap' + counter);

// 为样式映射创建普通图标
normal = ge.createIcon('');
normal.setHref('http://maps.google.com/mapfiles/kml/shapes/triangle.png');
iconNormal = ge.createStyle('styleIconNormal' + counter);
iconNormal.getIconStyle().setIcon(normal);

// 为样式映射创建突出显示图标
highlight = ge.createIcon('');
highlight.setHref('http://maps.google.com/mapfiles/kml/shapes/square.png');
iconHighlight = ge.createStyle('styleIconHighlight' + counter);
iconHighlight.getIconStyle().setIcon(highlight);
```



```

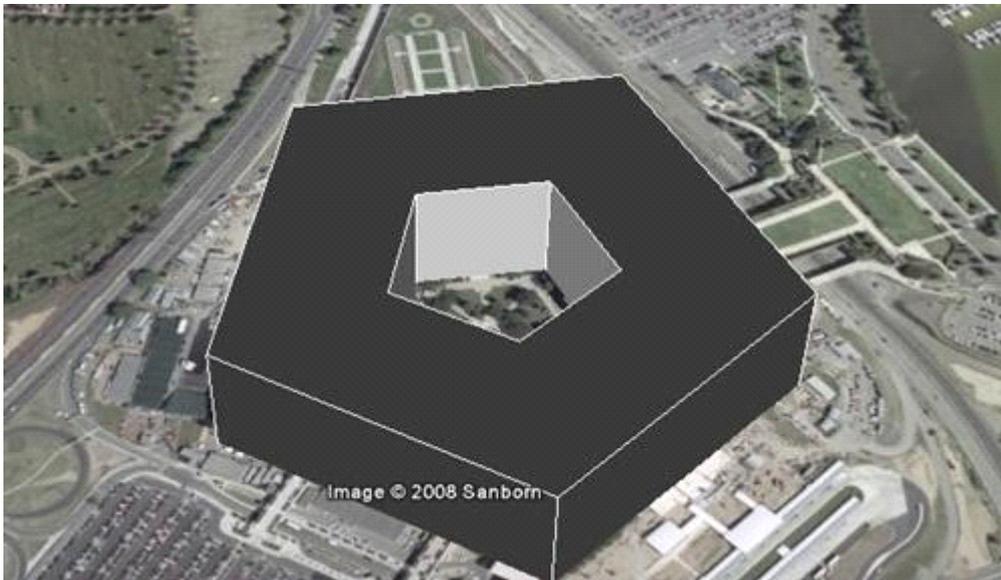
    ,      , +
    ,      , +
    ,      , +
    ,      , +
    '');

ge.getFeatures().appendChild(pentagon);

var la = ge.createLookAt('');
la.set(38.867, -77.0565, 500, ge.ALTITUDE_RELATIVE_TO_GROUND, 0, 45, 900);
ge.getView().setAbstractView(la);

```

本例在 Google Earth 中显示为：



7、显示模型

Google Earth 插件支持情景模型(以三维重建的场景)。三维模型导入至 Google Earth 后，模型会进行转换、旋转并缩放，以符合地球坐标系。以下示例会打开和关闭旧金山的三维构建层。

```

placemark = ge.createPlacemark('');
placemark.setName('model');
model = ge.createModel('');
ge.getFeatures().appendChild(placemark);
loc = ge.createLocation('');
model.setLocation(loc);
link = ge.createLink('');

```

```
// 在 Sketchup 中创建并导出为 Gollada 文件的纹理模型。
var href = window.location.href;
var pagePath = href.substring(0, href.lastIndexOf('/') + '/') + '/';
link.setHref(pagePath + 'spotchy_box.dae');
model.setLink(link);

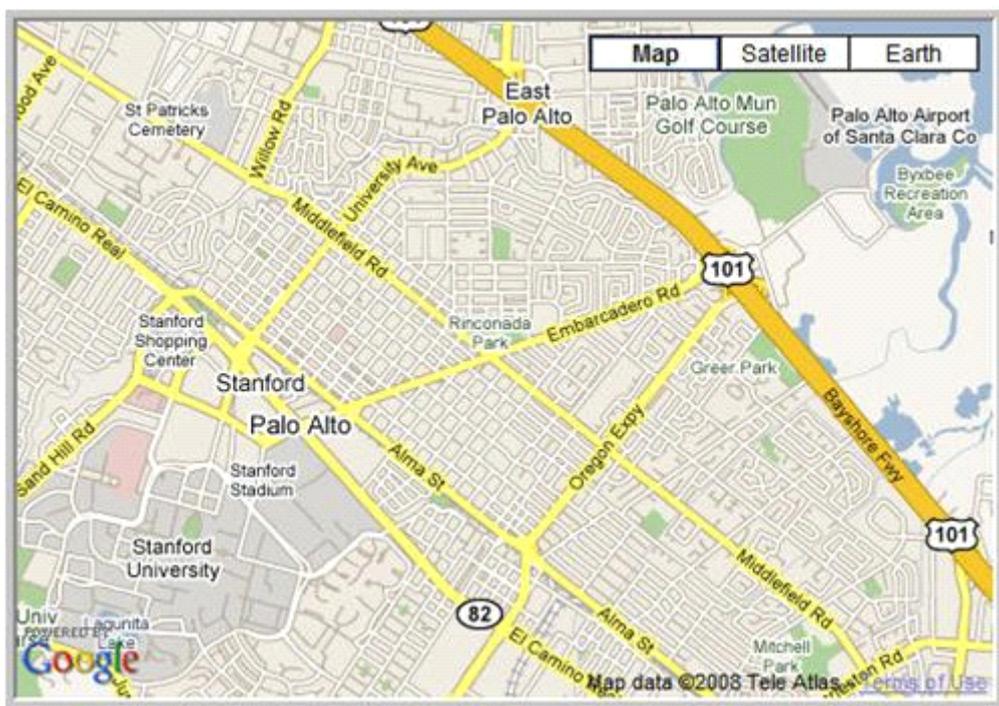
la = ge.getView().copyAsLookAt(ge.ALTITUDE_RELATIVE_TO_GROUND);
loc.setLatitude(la.getLatitude());
loc.setLongitude(la.getLongitude());

placemark.setGeometry(model);

la.setRange(300);
la.setTilt(80);
ge.getView().setAbstractView(la);
```

8、与 Google Maps 集成

如果您是 Google Maps 开发者，则可以使用 Google Earth 来为您的地图增效。Google Maps API 在 Gmap2 类中加入了一个新的方法，并在 GMapType 中加入了一个新的常量，这样可以让您在现有的地图中加入 Google Earth 的三维功能。以下示例中，标准 Google Maps 提供了新的地球按钮。



点击地球时，地图会像在 Google Earth 中的一样显示出来。



下面的代码演示了如何创建地图：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:v="urn:schemas-microsoft-com:vml">
<head>
<title>参考实施：通过地图 API 使用 Tumbler

<script src="http://maps.google.com/maps?file=api&v=2&key=abcdefg"
    type="text/javascript"></script>
var gMap;
function initialize() {
    var map = new GMap2(document.getElementById("map"));
    map.setCenter(new GLatLng(37.4419, -122.1419), 13);
    map.addMapType(G_SATELLITE_3D_MAP);
    map.addControl(new GHierarchicalMapTypeControl());
    gMap = map;

    // 取消注释下面一行，可在启动地图时启用三维模式。
    // map.setMapType(G_SATELLITE_3D_MAP);
}
</script>

</head>
<body onload="initialize()" onunload="GUnload">
```

```
<div id="map" class="map" style="width:500px;height:350px"></div>
</body>
</html>
```

上例中您可以调用 `gMap.addMapType(G_SATELLITE_3D_MAP)` 来将新的地图类型添加至现有的 Google Maps。如果您加入了地图控件，这个新的地图类型则会显示为标记地球的按钮。当用户点击此按钮时，Google 地球便会显示。添加至地图中的标记、线条、多边形和球形框可以在 Google 地球中正常工作。您也可以调用 `gMap.setMapType(G_SATELLITE_3D_MAP)` 以直观的方式将地图类型更改为地球。要隐藏 Google 地球，可调用 `gMapType.setMapType()` 更改至其他地图类型。