



Universitat d'Alacant
Universidad de Alicante

Advanced Web Programming

Sergio Luján Mora

**Department of Software and
Computing Systems**

Embedding Google Maps in web pages

GOOGLE MAPS



Index

- Google Maps
- Documentation
- Simple Map
- Markers
- Info Window



Google Maps

- GM is a geospatial viewing tool
- GM is currently a free product and requires no installation or management
 - Google reserves the right to put advertising on the map at any point in the future



Google Maps

- The ability to customize the map display through the addition of application specific data is the true driver of its acceptance



Google Maps

- Sign up for an API key
 - Sign up for a Google Account
 - Specify a web site URL → A single API key is valid for a single directory or domain

Google Maps API - Google Code - Windows Internet Explorer

http://code.google.com/apis/maps/

Archivo Edición Ver Favoritos Herramientas Ayuda

Google Maps API - Google Code

My favorites | English | Sign in

Google code Search

e.g. "ajax apis" or "open source"

Google Maps API


Home Docs FAQ Articles Blog Group Terms

What is the Google Maps API?


The Google Maps API lets you embed Google Maps in your own web pages with JavaScript. The API provides a number of utilities for manipulating maps (just like on the <http://maps.google.com> web page) and adding content to the map through a variety of services, allowing you to create robust maps applications on your website.

New! Set your map to exhibit the same [Default UI](#) as on `maps.google.com!`

New! Use [AdSense for Maps](#) in your Maps API applications!



Sign up for a Google Maps API key, or [read more about the API.](#)



Google I/O Developer Conference

May 27-28, San Francisco

Featured Geo API Session

[Performance Tips for Geo API mashups](#)

This talk will provide tips on reducing latency for your maps mashup, discussing topics like marker management, clustering, custom tiles, static maps, flash maps, encoded polys, light markers, latency oriented features of the JavaScript Maps API, and more.

[Learn more »](#)

How do I start?

1. [Sign up for a Google Maps API key.](#)
2. Read the [Maps API Developer's Guide.](#)
3. Read the [Maps API Reference.](#)

Listo

Internet 100%

Sign Up for the Google Maps API - Google Maps API - Google Code - Windows Internet Explorer

http://code.google.com/apis/maps/signup.html

Archivo Edición Ver Favoritos Herramientas Ayuda

Sign Up for the Google Maps API - Google Maps API - ...

My favorites | English | Sign in

Google code Search

e.g. "ajax apis" or "open source"

Google Maps API

[Home](#) [Docs](#) [FAQ](#) [Articles](#) [Blog](#) [Group](#) [Terms](#)

Sign up for an API key
[Create a KML Sitemap](#)

Maps API

[Home Page](#)
[Documentation](#)

Maps API for Flash

[Home Page](#)
[Documentation](#)

Maps Data API

[Home Page](#)
[Documentation](#)

Mapplets API

[Home Page](#)
[Documentation](#)

Sign Up for the Google Maps API

The Google Maps API lets you embed Google Maps in your own web pages. A single Maps API key is valid for a single "directory" or domain. See this [FAQ](#) for more information. You must have a [Google Account](#) to get a Maps API key, and your API key will be connected to your Google Account.

Here are some highlights from the terms for those of you who aren't lawyers:

- **There is no limit on the number of page views you may generate per day using the Maps API.** However, if you expect more than 500,000 page views per day, please [contact us](#) in advance so we can provision additional capacity to handle your traffic.
- **There is a limit on the number of geocode requests per day.** See this [FAQ](#) for more information on what represents a geocode request and what the exact limits are.
- **The Maps API does not include advertising.** If we ever decide to change this policy, we will give you at least 90 days notice via the [Google Geo Developers Blog](#).
- **Your service must be [freely accessible](#) to end users.** To use Google mapping technology in other types of applications, please use [Google Maps API Premier](#).
- **You may not alter or obscure the logos or attribution on the map.**
- You must [indicate whether your application is using a sensor](#) (such as a GPS locator) to determine the user's location.
- You may use the API (except for the Static Maps API) in websites or in software applications. For websites, please sign up with the URL where your implementation can be found. For other software applications, please sign up with the URL of the page where your application can be downloaded.
- Google will upgrade the API periodically, and you must update your site to use the new versions of the API. The Maps team will notify you of updates on the [Google Geo Developers Blog](#). If we make a non-backwards compatible change, we will give you at least a month's notice to make the transition, during which both versions of the API will be available.
- There are some uses of the API that we just don't want to see. For instance, we do not want to see maps that identify the places to buy illegal drugs in a city, or any similar illegal activity. We also want to respect people's privacy, so the API should

Internet 100%

Google Maps API - API Key Signup - Windows Internet Explorer

http://maps.google.com/maps/api_signup?url=http%3A%2Flocalhost

Archivo Edición Ver Favoritos Herramientas Ayuda

Google Maps API - API Key Signup

slujanmora@gmail.com | [My Account](#) | [Sign out](#)

Google
Code

Search

e.g. "ajax apis" or "open source"

Google Maps API [Home](#) [Docs](#) [FAQ](#) [Articles](#) [Blog](#) [Group](#) [Terms](#)

Sign up for an API key
[Create a KML Sitemap](#)

Maps API
[Home Page](#)
[Documentation](#)

Maps API for Flash
[Home Page](#)
[Documentation](#)

Mapplets API
[Home Page](#)
[Documentation](#)

Google Static Maps API
[Developer's Guide](#)

[Google Maps API Premier](#)

Google Maps API - API Key Signup

Thank You for Signing Up for a Google Maps API Key!

Your key is:

This key is good for all URLs consisting of this registered domain (and directory if applicable):

`http://localhost/`

Note: for more information on the API key system, consult <http://code.google.com/apis/maps/faq.html#keysystem>.

How you use your key depends on what Maps API product or service you use. Your key is valid for use within the entire family of Google Maps API solutions. The following examples show how to use your key within the Maps API product family.

JavaScript Maps API Example

Within the JavaScript Maps API, place the key within the script tag when you load the API:

```
...
// Note: you will need to replace the sensor parameter below with either an explicit true or false
value.
<script src="http://maps.google.com/maps?
file=api&v=2&camp=sensor=true or false&key=ABQIAAAAgc_DdIITVC4h7i10-
```



Documentation

- Google Maps API Concepts:
 - <http://code.google.com/apis/maps/documentation/>
- Google Maps API Examples:
 - <http://code.google.com/apis/maps/documentation/examples/>
- Google Maps API Reference:
 - <http://code.google.com/apis/maps/documentation/reference.html>

Core Class:

[GMap2](#)

This is the most important class within the Maps API. The other classes in this reference are grouped by their purpose.

Base Classes:

[GBounds](#)

[GBrowsersIsCompatible](#)

[GDraggableObject](#)

[GDraggableObjectOptions](#)

[GInfoWindow](#)

[GInfoWindowOptions](#)

[GInfoWindowTab](#)

[GKeyboardHandler](#)

[GLanguage](#)

[GLatLng](#)

[GLatLngBounds](#)

[GLog](#)

[GMapOptions](#)

[GMapPane](#)

[GPoint](#)

[GSize](#)

[GUnload](#)

[G_API_VERSION](#)

Event Classes:

[GEvent](#)

[GEventListener](#)

Control Classes:

[GControl](#)

[GControlAnchor](#)

[GControl](#)

[GControlPosition](#)

[GHierarchicalMapTypeControl](#)

[GMapType](#)

[GMapTypeControl](#)

[GMapTypeOptions](#)

[GMapUIOptions](#)

[GMenuMapTypeControl](#)

[GNavLabelControl](#)

Overlay Classes:

[GCopyright](#)

[GCopyrightCollection](#)

[GGroundOverlay](#)

[GIcon](#)

[GLayer](#)

[GMarker](#)

[GMarkerManager](#)

[GMarkerManagerOptions](#)

[GMarkerOptions](#)

[GMercatorProjection](#)

[GOverlay](#)

[GPolyEditingOptions](#)

[GPolyStyleOptions](#)

[GPolygon](#)

[GPolygonOptions](#)

[GPolyline](#)

[GPolylineOptions](#)

[GProjection](#)

[GScreenOverlay](#)

[GScreenPoint](#)

[GScreenSize](#)

[GTileLayer](#)

[GTileLayerOptions](#)

[GTileLayerOverlay](#)

[GTileLayerOverlayOptions](#)

Service Classes:

[GAdsManager](#)

[GAdsManagerOptions](#)

[GAdsManagerStyle](#)

[GClientGeocoder](#)

[GDirections](#)

[GDirectionsOptions](#)

[GDownloadUrl](#)

[GFactualGeocodeCache](#)

[GGeoAddressAccuracy](#)

[GGeoStatusCode](#)

[GGeoXml](#)

[GGeocodeCache](#)

[GGoogleBar](#)

[GGoogleBarAdsOptions](#)

[GGoogleBarLinkTarget](#)

[GGoogleBarListingTypes](#)

[GGoogleBarOptions](#)

[GGoogleBarResultList](#)

[GPov](#)

[GRoute](#)

[GStep](#)

[GStreetviewClient](#)

[GStreetviewClient.ReturnValues](#)

[GStreetviewData](#)

[GStreetviewLink](#)

[GStreetviewLocation](#)

[GStreetviewOverlay](#)

[GStreetviewPanorama](#)

[GStreetviewPanorama.ErrorValues](#)

[GStreetviewPanoramaOptions](#)

[GTrafficOverlay](#)

[GTrafficOverlayOptions](#)

[GTravelModes](#)

[GXml](#)

[GXmlHttp](#)

[GXslt](#)

Simple Map

```
<script  
  src="http://maps.google.com/maps?file=api&am  
  p;v=2&amp;sensor=false&amp;key=API_KEY"  
  type="text/javascript"></script>
```



Simple Map

```
<script type="text/javascript">
function initialize() {
  if (GBrowserIsCompatible()) {
    var map = new
    GMap2(document.getElementById("map_canvas"));
    // Alicante (Spain)
    // map.setCenter(new GLatLng(37.4419, -122.1419), 13);
    // Lublin (Poland)
    map.setCenter(new GLatLng(51.24, 22.57), 13);
    // Show user interface
    // map.setUIToDefault();
  }
}
</script>
```

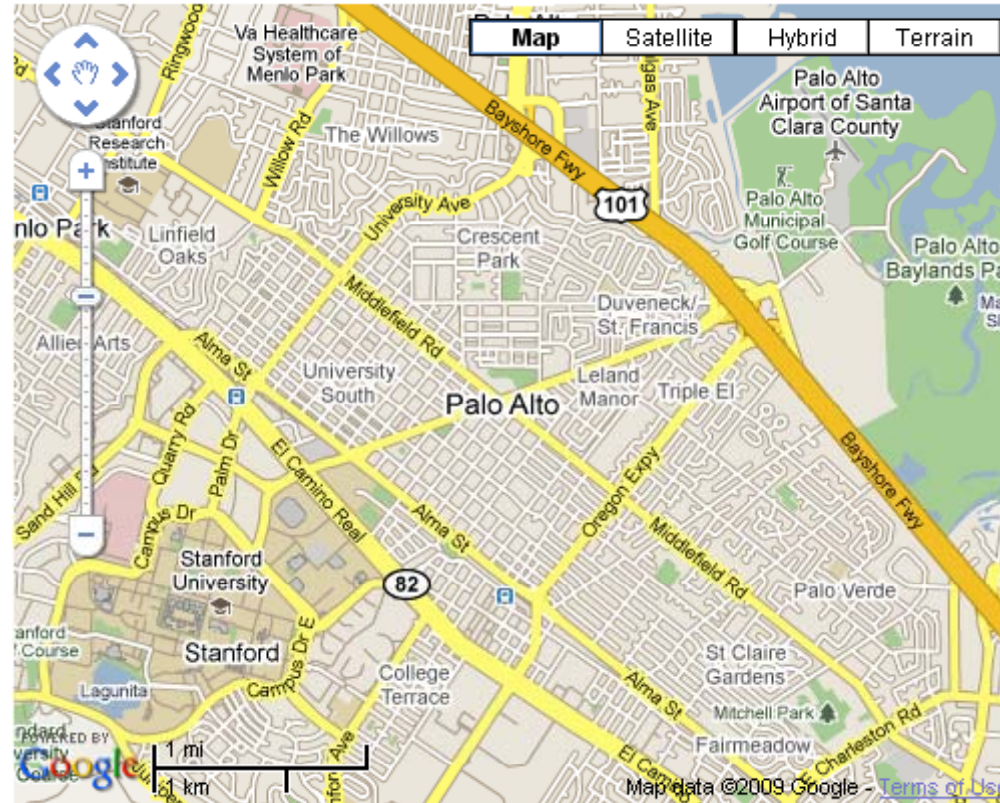
Simple Map

- `var map = new GMap2()` → We pass the `<div>` specifying where the map will appear
 - The size of the map will default to the size specified in the `<div>` tag
- `map.setCenter()` → We center the map at a particular latitude, longitude, and zoom level
 - This method must be called first after construction to set the initial state of the map
 - 17 point scale: 0 (entire world) ... 16 street level
- `map.setUIToDefault()` → Adds the default behaviour and UI elements

Simple Map

```
<body onload="initialize()" onunload="GUnload()">  
<div id="map_canvas" style="width: 500px; height:  
    400px">  
</div>  
</body>
```

We have to use
the same **id** in the
JavaScript code





Simple Map

```
// map.setUIToDefault();  
map.addControl(new GLargeMapControl());
```

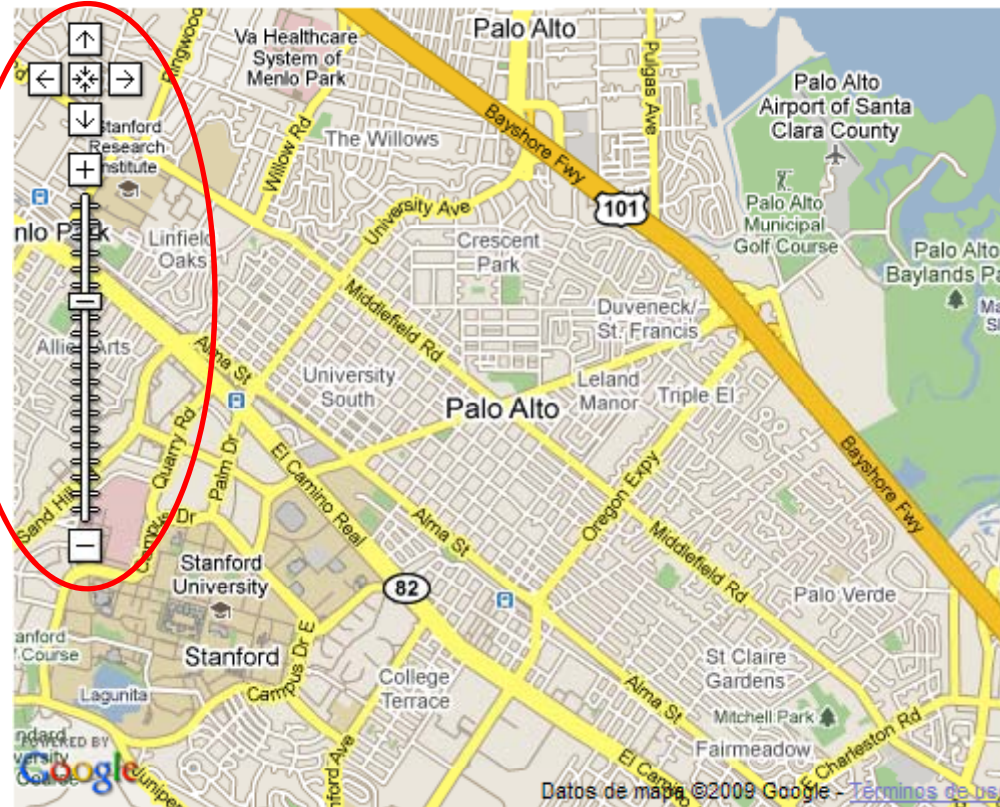
`GSmallMapControl()`: Creates a control with buttons to pan in four directions, and zoom in and zoom out.

`GLargeMapControl()`: Creates a control with buttons to pan in four directions, and zoom in and zoom out, and a zoom slider.

`GSmallZoomControl()`: Creates a control with buttons to zoom in and zoom out.

`GLargeMapControl3D()`: Creates a new 3D-style control with buttons to pan in four directions, and zoom in and zoom out, and a zoom slider.

`GSmallZoomControl3D()`: Creates a new 3D-style control with buttons to zoom in and zoom out.





Simple Map

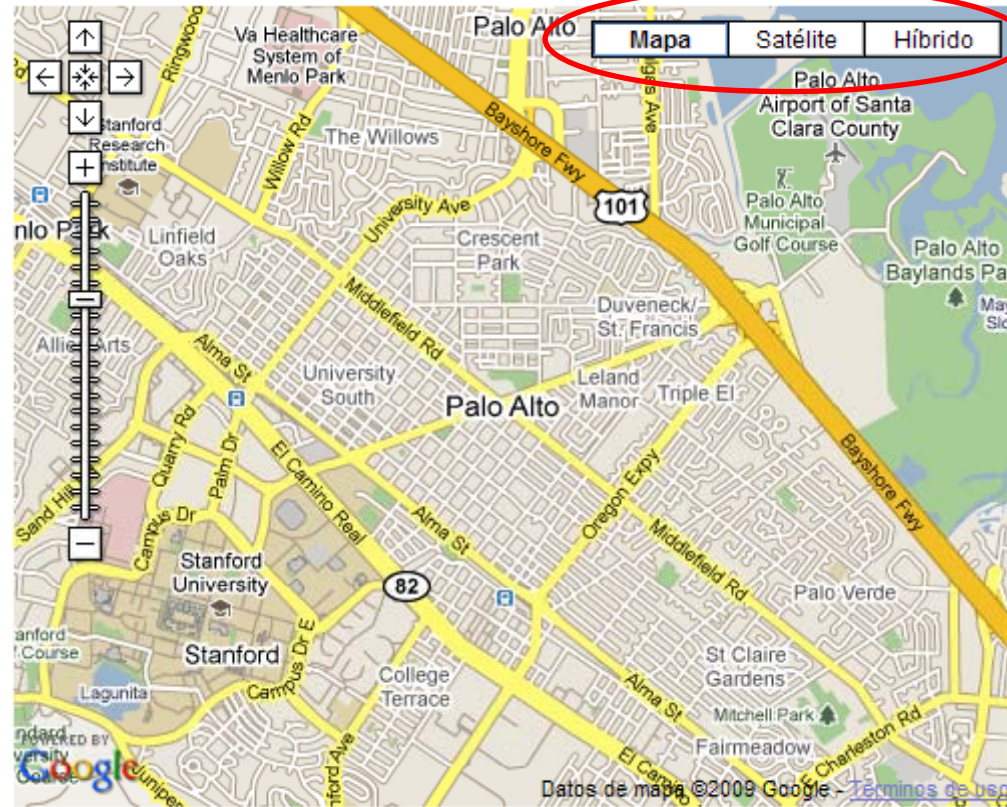
```
// map.setUIToDefault();  
map.addControl(new GLargeMapControl());  
map.addControl(new GMapTypeControl());
```

GMapTypeControl(): Creates a standard map type control for selecting and switching between supported map types via buttons.

GMenuMapTypeControl(): Creates a drop-down map type control for switching between supported map types.

GHierarchicalMapTypeControl(): Creates a "nested" map type control for selecting and switching between supported map types via buttons and nested checkboxes.

Advanced Web Programming





Simple Map

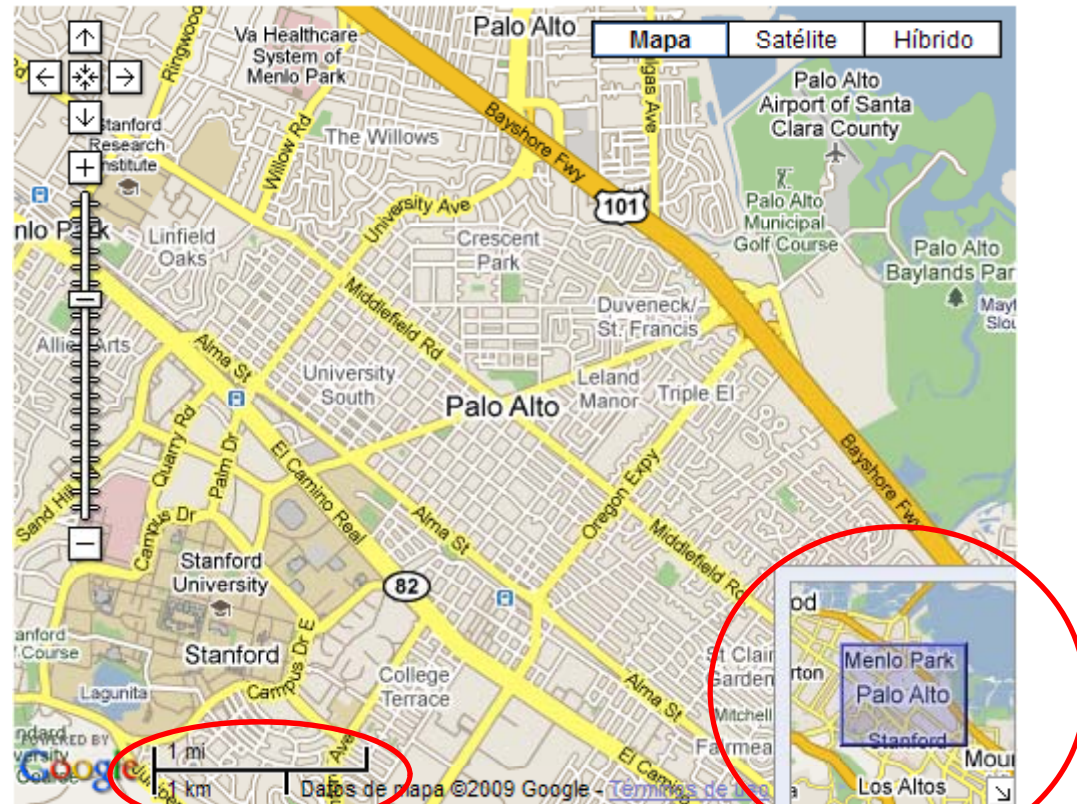
```
// map.setUIToDefault();  
map.addControl(new GLargeMapControl());  
map.addControl(new GMapTypeControl());  
map.addControl(new GScaleControl());  
map.addControl(new GOverviewMapControl());
```

`GScaleControl()`: Creates a control that displays the map scale.

`GOverviewMapControl()`: Creates a collapsible overview mini-map in the corner of the main map for reference location and navigation (through dragging). The `GOverviewMapControl` creates an overview map with a one-pixel black border. Note: Unlike other controls, you can only place this control in the bottom right corner of the map

(`G_ANCHOR_BOTTOM_RIGHT`).

Advanced Web Programming



Simple Map

- By default, the Normal view will be displayed
- However, the map type can be changed to several map types with `map.setMapType()`:
 - `G_NORMAL_MAP` This map type (which is the default) displays a normal street map.
 - `G_SATELLITE_MAP` This map type displays satellite images.
 - `G_HYBRID_MAP` This map type displays a transparent layer of major streets on satellite images.
 - `G_PHYSICAL_MAP` This map type displays maps with physical features such as terrain and vegetation. This map type is not displayed within map type controls by default.

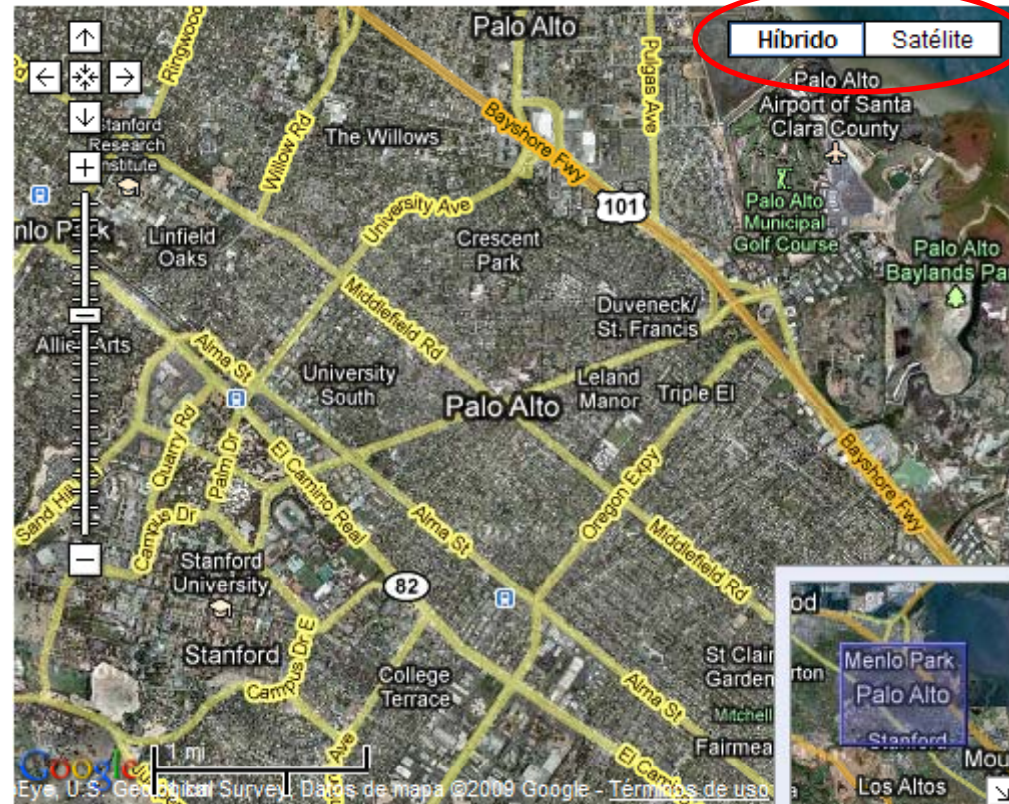
Simple Map

- Special map types:
 - G_MOON_ELEVATION_MAP: This map type displays a shaded terrain map of the surface of the Moon, color-coded by altitude. This map type is not displayed within map type controls by default.
 - G_MOON_VISIBLE_MAP: This map type displays photographs taken from orbit around the moon. This map type is not displayed within map type controls by default.
 - G_MARS_ELEVATION_MAP: This map type displays a shaded relief map of the surface of Mars, color-coded by altitude. This map type is not displayed within map type controls by default.
 - G_MARS_VISIBLE_MAP: This map type displays photographs taken from orbit around Mars. This map type is not displayed within map type controls by default.
 - G_MARS_INFRARED_MAP: This map type displays a shaded infrared map of the surface of Mars, where warmer areas appear brighter and colder areas appear darker.
 - G_SKY_VISIBLE_MAP: This map type shows a mosaic of the sky, covering the full celestial sphere.

Simple Map

- By default, the Map, Satellite, and Terrain map types are available
 - We can control the available map types through a list that is provided in the GMap2 constructor:

```
var map = new  
  GMap2(document.getElementById("map_canvas"),  
        {mapTypes: [G_HYBRID_MAP, G_SATELLITE_MAP]});
```



Markers

- We can add some point data to a map
- The GMarker class is used to create icons showing points of interest





Markers

```
// Latitude and longitude of the new marker  
var point = new GLatLng(lat, lon);  
// Create a new marker  
var marker = new GMarker(point);  
// Add the marker to the map  
map.addOverlay(marker);
```



Markers

- Delete a marker:

```
var point = new GLatLng(lat, lon);  
var marker = new GMarker(point);  
map.addOverlay(marker);  
// ...  
// Delete a specific marker  
map.removeOverlay(marker);  
// Delete all the markers  
map.clearOverlays();
```



Markers

- Exercise:
 - Show a map of Lublin
 - Show a list (`<select>` with different places in Lublin)
 - When the user selects a place, show a marker in the map
 - Show a button to delete the markers on the map

Markers

- GMarkerManager class is used to efficiently manage hundreds of markers on a map
 - Without the use of this class, the performance of the application can be very poor
 - This class can also be used to reduce the clutter of these markers when viewed at certain map scales

Markers

- `addMarkers(markers:GMarker[], minZoom:Number, maxZoom?:Number)`: Adds a batch of markers to this marker manager. The markers are not added to the map, until the `refresh()` method is called.
- `addMarker(marker:GMarker, minZoom:Number, maxZoom?:Number)`: Adds a single marker to a collection of markers controlled by this manager. If the marker's location falls within the map's current viewport and the map's zoom level is within the specified zoom level range, the marker is immediately added to the map.
- `refresh()`: Forces the manager to update markers shown on the map. This method must be called if markers were added using the `addMarkers` method.



Markers

```
var mkmgr = new GMarkerManager(map);  
...  
mkmgr.addMarker(new GMarker(point1), 13, 17);  
mkmgr.addMarker(new GMarker(point2), 13, 17);  
mkmgr.addMarker(new GMarker(point3), 13, 17);  
...  
mkmgr.refresh();
```



Markers

- **Exercise:**

- Show a map of Lublin
- Show a list (<select> with different places in Lublin)
- When the user selects a place, show a marker in the map
- Show a button to delete the markers on the map
- **Center the map in the marker:**

```
var point = new GLatLng(lat, lon);
```

```
var marker = new GMarker(point);
```

```
...
```

```
map.panTo(point);
```



Markers

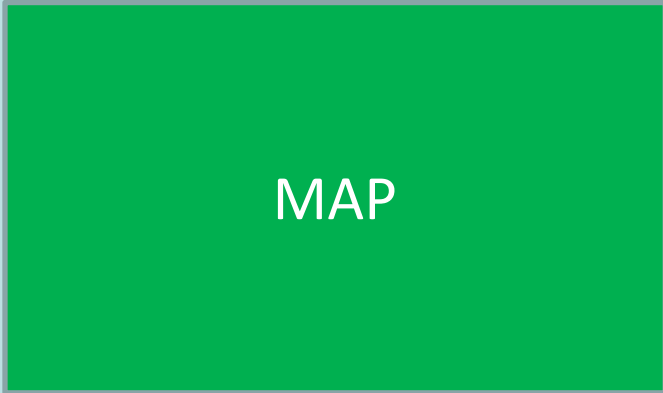
- Exercise:
 - Create a web page that shows the different places a person has visited during a trip
 - User interface:
 - Button “Star trip”: the code starts to show a marker for every place every three seconds
 - Button “Stop trip”
 - Use `setInterval()` to execute some code after a specified time-interval
 - Use `clearInterval()` to stop the timer



Markers

- Exercise:
 - Create a web page that allows the user to add markers to a map
 - User interface:
 - Latitude input box
 - Longitude input box
 - Button “Add marker”
 - Button “Remove markers”

Markers



MAP

Latitude:

Longitude:



Markers

- Show a small map over a marker:

```
marker.showMapBlowup( );
```



Markers

- A marker has options, e.g. allow a marker to be moved:

```
var options = {draggable: true};  
marker = new GMarker(point, options);  
marker.enableDragging();
```



Markers

- How to change the icon of a marker:

```
var myIcon = new GIcon(G_DEFAULT_ICON);  
myIcon.image = "my_custom_icon.png";  
myIcon.iconSize = new GSize(22, 31);  
myIcon.shadow = "my_custom_icon_shadow.png";  
myIcon.shadowSize = new GSize(42, 31);  
myIcon.iconAnchor = new GPoint(10, 29);  
myIcon.infoWindowAnchor = new GPoint(10, 14);  
myIcon.printImage = "my_custom_icon_print.gif";  
myIcon.mozPrintImage = "my_custom_icon_mozPrint.gif";  
myIcon.printShadow = "my_custom_icon_printShadow.gif";  
myIcon.transparent = "my_custom_icon_transparent.png";  
myIcon.imageMap = [ 10,29, 1,16, 0,5, 5,0, 12,4, 18,2,  
21,12, 21,16 ];
```

Info Window

- We can add an info window to display information about the markers
 - But info window can be placed anywhere on a map
- Events are external stimulus to the map and are usually triggered by the user
 - We can write code that responds to any of the defined events



Info Window

- Write an info window with HTML in the middle of the map:

```
var html = "A simple <b>text</b>";  
map.openInfoWindowHtml(map.getCenter(), html);
```

Info Window

- Write a tabbed info window with HTML in the middle of the map:

```
var info = [  
  new GInfoWindowTab("School", "Lublin"),  
  new GInfoWindowTab("Hospital", "Warsaw"),  
  new GInfoWindowTab("Culture", "Krakow")  
];  
  
map.openInfoWindowTabsHtml(map.getCenter(),  
  info);
```



Info Window

- Show information in a marker:

```
var point = new GLatLng(lat, lon);  
map.addOverlay(new GMarker(point));  
marker.openInfoWindowHtml("Something we want  
to show");
```

```
// We can also use:
```

```
// marker.openInfoWindowTabsHtml(tabs);
```



Events

- We want to show an info window when we click the map



Events

- We have to add an event handler for a “click”:

```
GEvent.addListener(map, "click", function(overlay, latlng) {  
    map.openInfoWindowHtml(latlng, "Something we want  
    to show");  
});
```

Events

- We want to show latitude and longitude coordinates:

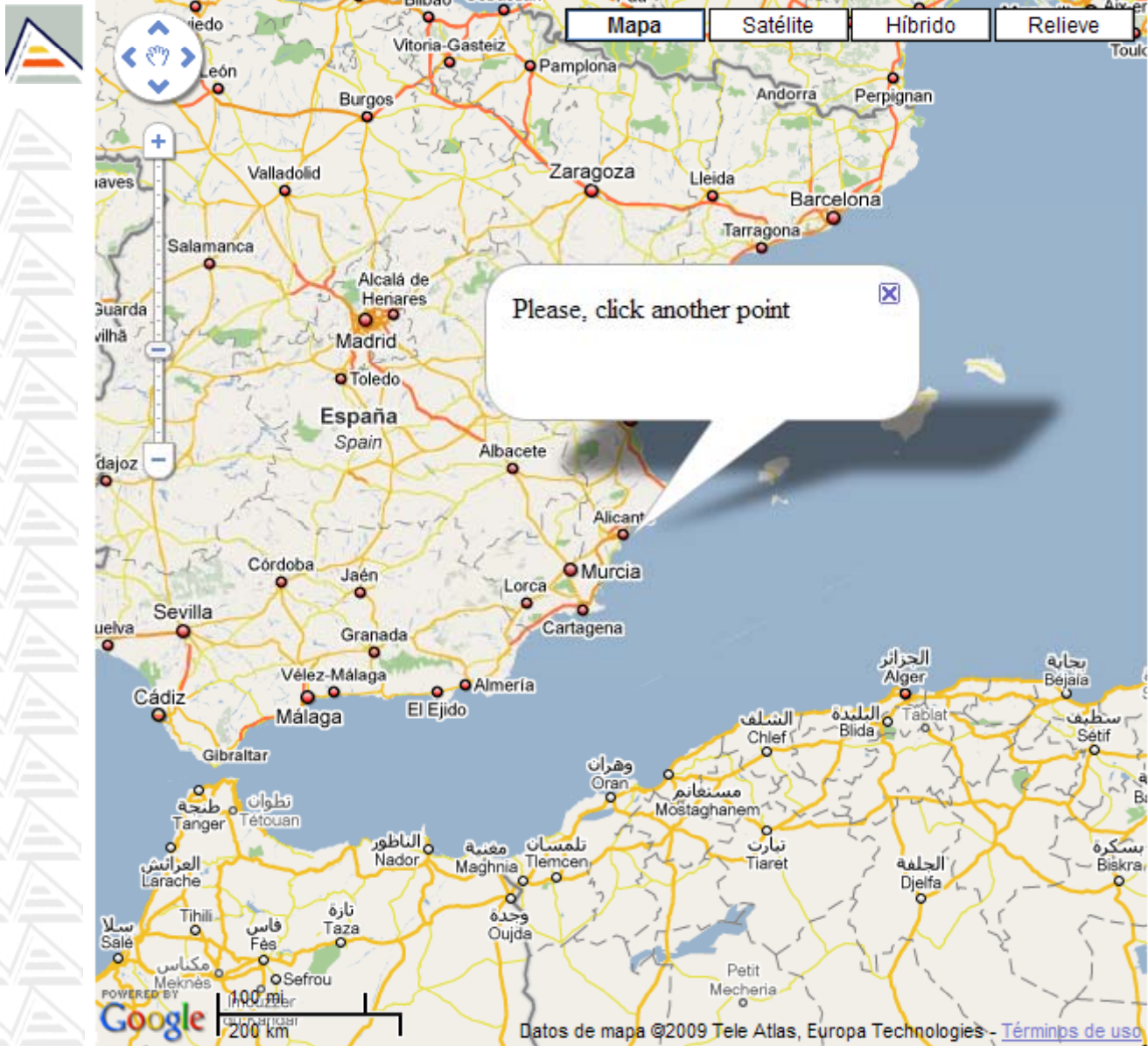
```
GEvent.addListener(map, "click", function(overlay,
    latlng) {
    map.openInfoWindowHtml(latlng, "Latitude: " +
        latlng.lat() + "<br />Longitude: " +
        latlng.lng());
});
```



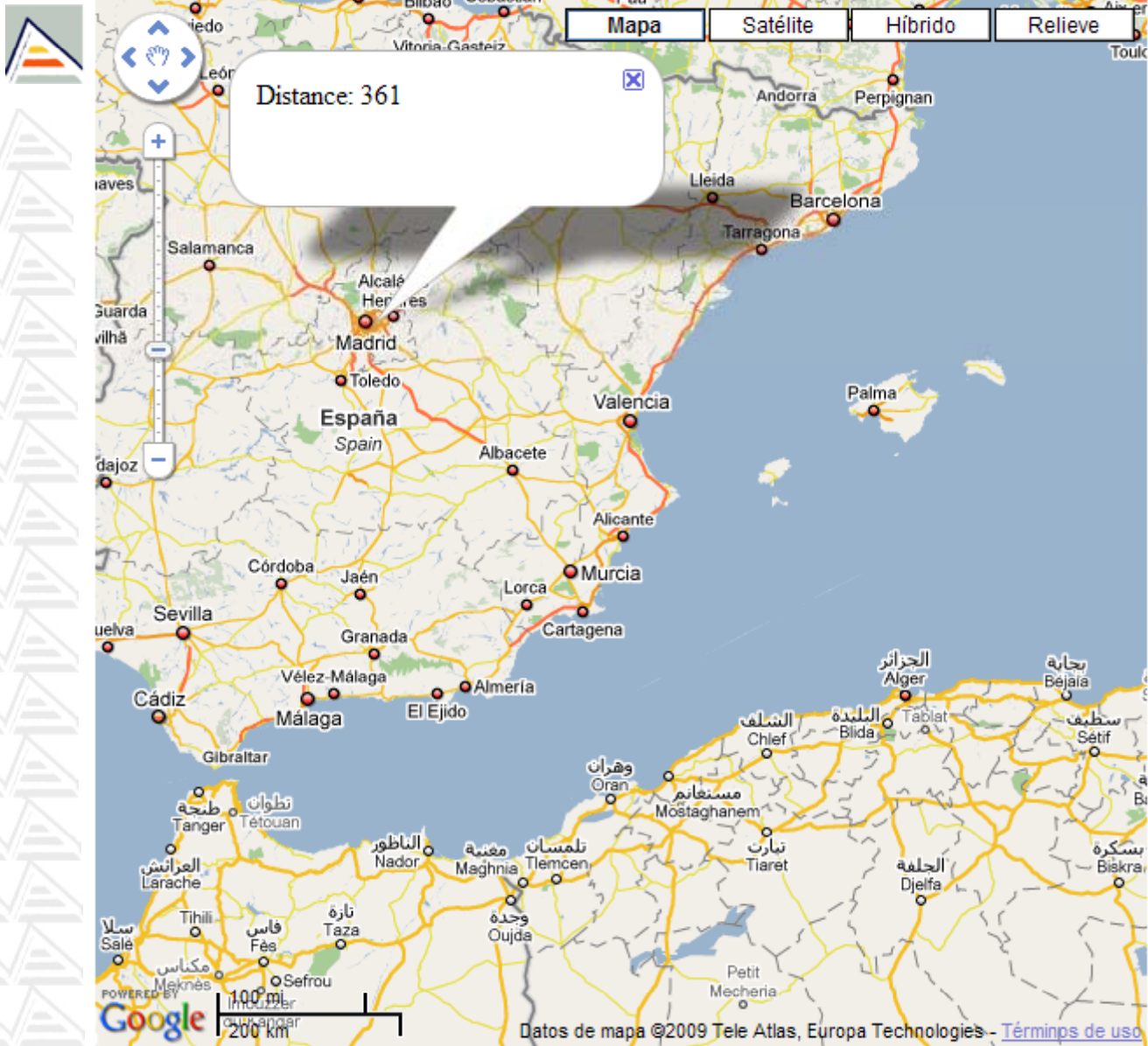
Events

- Exercise:
 - The user clicks two points in a map
 - Use function `LatLng2.distanceFrom(LatLng1)` to calculate the distance in meters between two points

Advanced Web Programming



Advanced Web Programming



Events

- We want to show an info window over a marker when we click the marker



Events

- We have to add an event handler for a “click”:

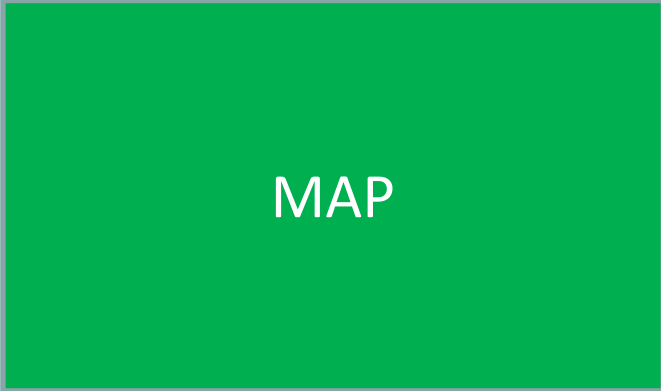
```
GEvent.addListener(marker, "click", function() {  
    marker.openInfoWindowHtml("Something we want to  
    show");  
});
```



Events

- Exercise:
 - Add “Content” input box to the previous exercise

Markers



MAP

Latitude:

Longitude:

Content:

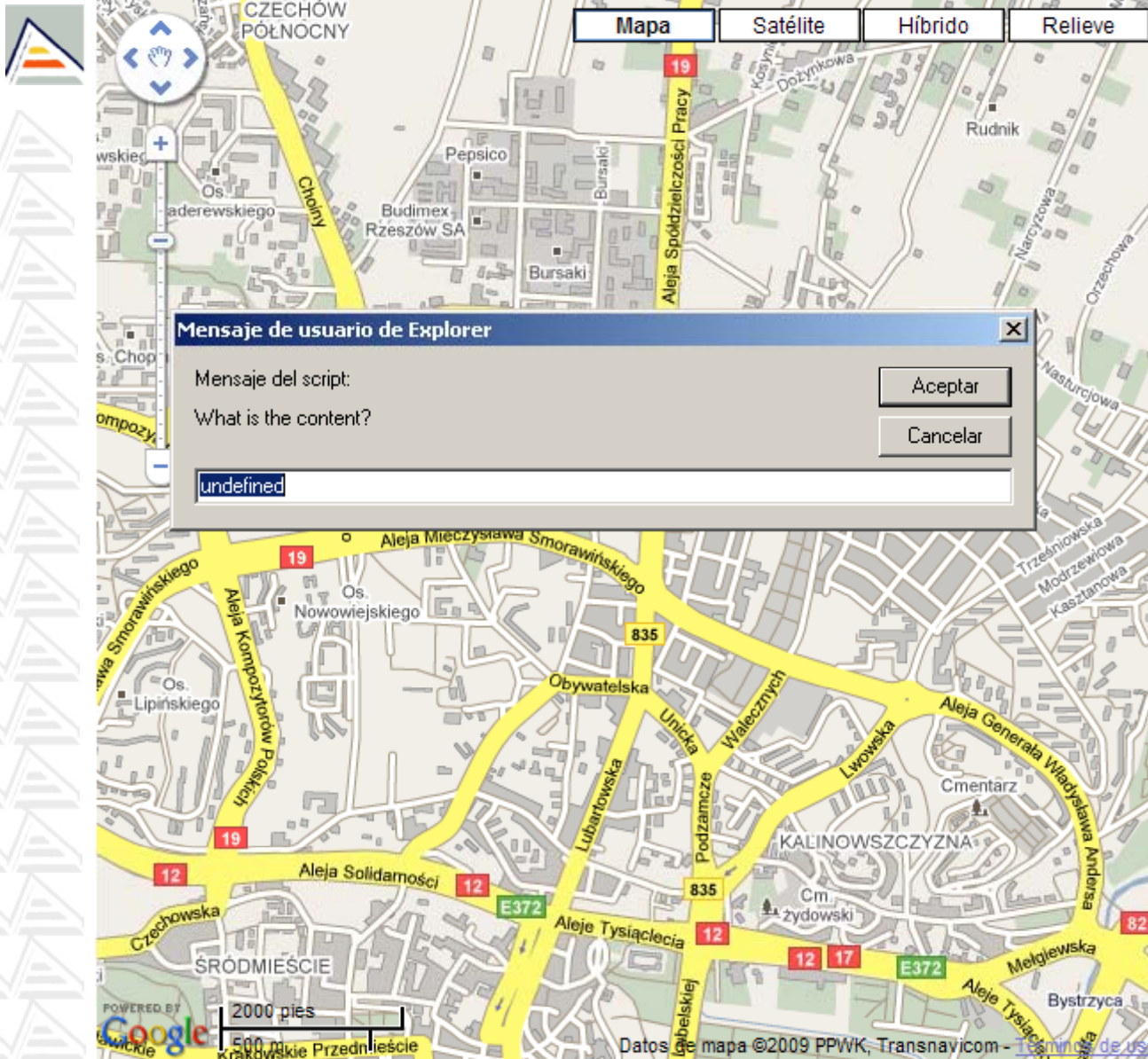




Markers

- Exercise:
 - We want to add a new marker with content to a map when we click the map
 - Define addListener for the map to handle the “click event”
 - Create a new marker
 - Ask for the content of the info window of the marker, use JavaScript function prompt()
 - Define addListener for the marker to handle the “click event” to show the info window
- Add the new marker to the map

Advanced Web Programming



Advanced Web Programming

