User Interaction: Google Maps API

Associate Professor Donald J. Patterson
INF 133 Fall 2013
Why maps?

- Central to much mobile interaction work.
- It can be generalized to interactions with enormous images very easily.
- Mapping data is a natural visualization that helps to reveal patterns.
- It was the first infographic
Google Maps API

Why maps?

http://gigapan.com/galleries/10347/gigapans/66626

http://gigapan.org/viewGigapanFullscreen.php?auth=033ef14483ee899496648c2b4b06233c
Google Maps API

- Google Maps v3 is optimized for mobile
- It used to be just for mobile but is being adapted as the standard Maps API for all platforms
Google Maps API v3

Basics

- Google hosts most of the code in a javascript library
- A developer includes the javascript library
  - Just like jQuery
- A developer makes calls to the library to:
  - display the map
  - display controls
  - display points
  - display other U/I elements
Basics

- You can register event handlers
- So that when a user does something on the map, you can do something on the webpage also (through JavaScript)
Basic Map Objects

- Create a placeholder in your webpage to hold the map object

```html
<!DOCTYPE html>
<html>
  <head>
    <script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
    <script src="//ajax.googleapis.com/ajax/libs/jqueryui/1.9.0/jquery-ui.min.js"></script>
    <script src="Step01.js"></script>
    <style type="text/css">
      html { height: 100% }
      body { height: 100%; margin: 0; padding: 0 }
    </style>
  </head>
  <body>
    <button onClick="myReadyFunction()">Start!</button>
    <div id="map_canvas" style="float:right;width:500px;height:353px;border:1px black">
      <p>
        <a href="//www.flickr.com/photos/spikefi/4064961129/" title="Jack-o'-lantern 2009 by Cheryl Q, on Flickr">
          <img src="//farm3.staticflickr.com/2658/4064961129_fa042a2d4d_n.jpg" width="320" height="213" alt="Jack-o'-lantern 2009"/>
        </a>
      </p>
    </div>
    <div class="dataXML">
      <img src="images/coffeebean.jpg" alt="coffeebean" />
      Replace this with XML
    </div>
    <hr/>
    <div class="dataJSON">
      <img src="images/coffeebean.jpg" alt="coffeebean" />
      Replace this with JSON
    </div>
    <hr/>
    <div class="dataJSONP">
      <img src="images/coffeebean.jpg" alt="coffeebean" />
      Replace this with JSONP
    </div>
  </body>
</html>
```

Credit: http://www.flickr.com/photos/spikefi/4064961129/
Tip

- **Absolute link**

- **Relative link**
  - `picture.jpg`
  - `images/picture.jpg`
  - `/images/picture.jpg`

- **Protocol relative link**
  - `//students.ics.uci.edu/~djp3/images/picture.jpg`
Basic Map Objects

- The size is up to you
- Android/iPhone work best at full screen
- Maximizing to fill browser in HTML 5 is easy
Basic Map Objects

- Figuring out if you are using Android/iPhone etc. is possible, but complicated and changing all the time.
- Quick and Dirty Method:

```javascript
function detectBrowser() {
    var userAgent = navigator.userAgent;
    var mapdiv = document.getElementById("map_canvas");

    if (userAgent.indexOf('iPhone') !== -1 || userAgent.indexOf('Android') !== -1) {
        mapdiv.style.width = '100%';
        mapdiv.style.height = '100%';
    }
}
```

Basic Map Loading

• Loading the Google Maps Library

<script src="//maps.googleapis.com/maps/api/js?sensor=false&language=en&region=US"></script>

• Parameters for the library
  • “sensor”: does your location come from a GPS?
    • mandatory parameter
    • allows Google to respect business agreements
  • “language”: what language are the labels in? (dep?)
  • “region”: bias behavior toward a region (dep?)
  • “Paris” in TX or “Paris” in France?
Basic Map Loading

- Do I need a key?
  - Not for this assignment
  - If you are going to be deploying for real, then yes
    - More than 25,000 loads per day
Putting the map on the web page

```javascript
function initializeMap()
{
    var latlng = new google.maps.LatLng(33.643298, -117.841983);
    var myOptions = {
        zoom: 8,
        center: latlng,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };
    var map = new google.maps.Map($("#map_canvas")[0], myOptions);
}
```
Basic Map Usage

- Make sure that if you use jQuery to get your elements you pass the DOM element to Google

```javascript
map = new google.maps.Map($('<#map_canvas').get(0), myOptions);

map = new google.maps.Map($("#map_canvas")[0], myOptions);
```
Making a Latitude Longitude Pair

```javascript
var latlng = new google.maps.LatLng(33.643298, -117.841983);
```

- This doesn’t put anything on the map
- It’s just an internal reference to a point
- Does not need to be hard coded
Google Maps API v3

Passing options to a map

```javascript
var myOptions = {
  zoom: 8,
  center: centerPoint,
  mapTypeId: google.maps.MapTypeId.ROADMAP
};
```

- Uses standard Javascript Object Notation
- zoom 0, is the whole earth, anything higher is closer
- center is where the map should be centered when drawn
- mapTypeId is what kind of map you want
  - ROADMAP, SATELLITE, HYBRID, TERRAIN
Advanced Map Options

- 45 degree imagery control
- Changing the U/I controls that are available to the user
- Custom map styling
- Custom maps (like the inauguration photo)
- Other projections (Mercator being the default)
Putting the things on the web page

- Markers

```javascript
var marker = new google.maps.Marker(
    {position: latlng,
    map: map,
    title: "Hello World!"
});
```
Recentering the map

- A javascript call to the map object

```html
<button onClick="map.setCenter(centerPoint)">Recenter Map</button>
```
Putting the custom icons on the map

- Icons

```javascript
var image = new google.maps.MarkerImage('images/icon.png', 
  // This marker is 32 pixels wide by 32 pixels tall.
  new google.maps.Size(32, 32),
  // The origin for this image is 0,0.
  new google.maps.Point(0,0),
  // The anchor for this image is at 0,16.
  new google.maps.Point(0, 16));

// The shadow image is larger in the horizontal dimension
// while the position and offset are the same as for the main image.
var shadow = new google.maps.MarkerImage('images/shadow.png',
  new google.maps.Size(41, 18),
  new google.maps.Point(0,0),
  new google.maps.Point(0,0));

var marker = new google.maps.Marker({
  position: latlng,
  map: map,
  shadow:shadow,
  icon: image,
  title: "Hello World!"
});
```
Google Maps API v3

Putting the things on the web page

• Polylines
Putting the things on the web page

- Polygons
Google Maps API v3

Putting the things on the web page

- Info Windows

Uluru, also referred to as Ayers Rock, is a large sandstone rock formation in the southern part of the Northern Territory, central Australia. It lies 335 km (208 mi) south west of the nearest large town, Alice Springs; 450 km (280 mi) by road. Kata Tjuta and Uluru are the two major features of the Uluru - Kata Tjuta National Park. Uluru is sacred to the Pitjantjatjara and Yankunytjatjara, the Aboriginal people of the area. It has many springs, waterholes, rock caves and ancient paintings. Uluru is listed as a World Heritage Site.

Google Maps API v3

Putting the things on the web page

- Info Windows

```javascript
var contentString = '<div id="content">' +
    '<div id="siteNotice">' +
    '<div id="bodyContent">Uluṟu</div>' +
    '</div>' +
    '<h1 id="firstHeading" class="firstHeading">Uluṟu</h1>' +
    '<div id="content" role="tabpanel" aria-labelledby="firstHeading">' +
    '<p><b>Uluṟu</b>, also referred to as <b>Ayers Rock</b>, is a large sandstone rock formation in the southern part of the Northern Territory, central Australia. It lies 335 km (208 mi) south west of the nearest large town, Alice Springs; 450 km (280 mi) by road. Kata Tjuta and Uluṟu are the two major features of the Uluru – Kata Tjuta National Park. Uluru is sacred to the Pitjanṭjatjara and Yankunytjatjara, the Aboriginal people of the area. It has many springs, waterholes, rock caves and ancient paintings. Uluru is listed as a World Heritage Site.</p>' +
    '</div>' +
    '</div>' +
    '</div>' +
    '</div>);

var infowindow = new google.maps.InfoWindow({
    content: contentString
});

var marker = new google.maps.Marker({
    position: myLatlng,
    map: map,
    title: "Uluṟu (Ayers Rock)"
});

google.maps.event.addListener(marker, 'click', function() {
    infowindow.open(map, marker);
});
```
Geocoding Addresses

- Geocoding means translating a place into a (Lat,Long) pair

```javascript
function codeAddressHelper(results, status) {
    if (status == google.maps.GeocoderStatus.OK) {
        map.setCenter(results[0].geometry.location);
        var marker = new google.maps.Marker({
            map: map,
            position: results[0].geometry.location
        });
    } else {
        alert("Geocode was not successful for the following reason: " + status);
    }
}

function codeAddress() {
    var geocoder = new google.maps.Geocoder();
    var myAddress = $("#address").val();
    geocoder.geocode({ address: myAddress }, codeAddressHelper);
}
Google Maps API v3

More info

- Google Maps API v3
  - https://developers.google.com/maps/documentation/javascript/tutorial
    - HeatMaps
    - Weather
    - Street view
    - Ads
    - Utilities
    - Events
    - Traffic
    - Infographics (FusionTables)
How do you debug?

- Web Developer Toolbar
- Firefox Error Console
- Internet Explorer Script Debugger
- Firebug
- Google’s GLog library
  - For adding your own debugging statements
- screencast tutorials here:
Task 19

• Add a map to Task 14
• AJAX call
• Make your table entries link to the map (center on click)
• Plot the table entries on the map
• Enhance the map in one interesting way
  • For example, multiple map marker icons
  • More than a custom marker or an info window
• There will be two bragging rights prizes awarded
  • 50% on data novelty
  • 50% on presentation