How to display a map using Polymaps JS library

Introduction

Polymaps is a JavaScript library for generating “slippy” maps in the style of Google Maps, Modest Maps, CloudMade and OpenLayers.

Most mapping libraries focus on 256×256-pixel image tiles, with only limited support for dynamic overlays such as county boundaries and point clouds. These libraries assume that data needed to produce the desired overlay can be loaded into memory all-at-once, making it difficult to visualize large datasets. Furthermore, while image tiles automatically adjust in resolution as the map zooms in or out, the overlay resolution remains constant; this greatly limits multi-scale exploration of data, as the resolution must be fixed either for macro- (e.g., state-level) or micro- (e.g., block-level) observation.

The goal of Polymaps is to better support rich, large-scale data overlays on interactive maps by extending the tile metaphor to vector graphics: in addition to standard image tiles, Polymaps supports vector tiles that are rendered with SVG. The vector geometry is loaded as GeoJSON via asynchronous XMLHttpRequest; Point geometry objects are rendered as SVG circle elements, Polygons as paths, and so on. By loading geometry at known tile boundaries, requests can be issued efficiently on-demand by the client, and responses trivially cached on the server. When the user zooms in, geometry can be seamlessly refined to show greater detail, while on zoom out, geometry can be simplified to improve performance.

Prerequisites

1. You will need the Polymaps JavaScript library for the example and it can be downloaded from the main site http://polymaps.org/
2. Full documentation for this library is at the Polymaps.org main site.
3. A phone device supporting with Web Runtime support installed.
4. You can use a code editor of your choice, but this example is build with Aptana Studio and the WRT S60 plugin.

Note: Aptana has been superseded by the Nokia Web Tools, an IDE and Simulator based on Aptana and Eclipse.

Example code

For this example we will create a default WRT S60 widget, with the following files

index.html basic.css basic.js

index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
 <head>
  <title> Sample web app</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <script language="javascript" type="text/javascript" src="basic.js"></script>
  <script language="javascript" type="text/javascript" src="simplegeo-polymaps-4f09be6/polymaps.js"></script>
     <link rel="stylesheet" href="basic.css" type="text/css" />
     <meta name="Generator" content="Nokia WDE 3.0.0" />
 </head>
 <body onload="init();">
  <div id="map"></div>
 </body>
</html>
```

basic.css

```
//The stylesheet is not implemented for this example.
```

basic.js

```
function init()
{

var po = org.polymaps;

var map = po.map()
    .container(document.getElementById("map").appendChild(po.svg("svg")))
    .add(po.interact())
```


(C) Copyright Nokia 2013. All rights reserved.

```
    .add(po.hash());

map.add(po.image()
    .url(po.url("http://{S}tile.cloudmade.com"
    + "/1a1b06b230af4efdbb989ea99e9841af" // http://cloudmade.com/register
    + "/999/256/{Z}/{X}/{Y}.png")
    .hosts(["a.", "b.", "c.", ""]));

map.add(po.compass().pan("none"));
map.add(po.arrow());
map.add(po.drag());
map.add(po.dblclick());
map.add(po.wheel());
}
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.