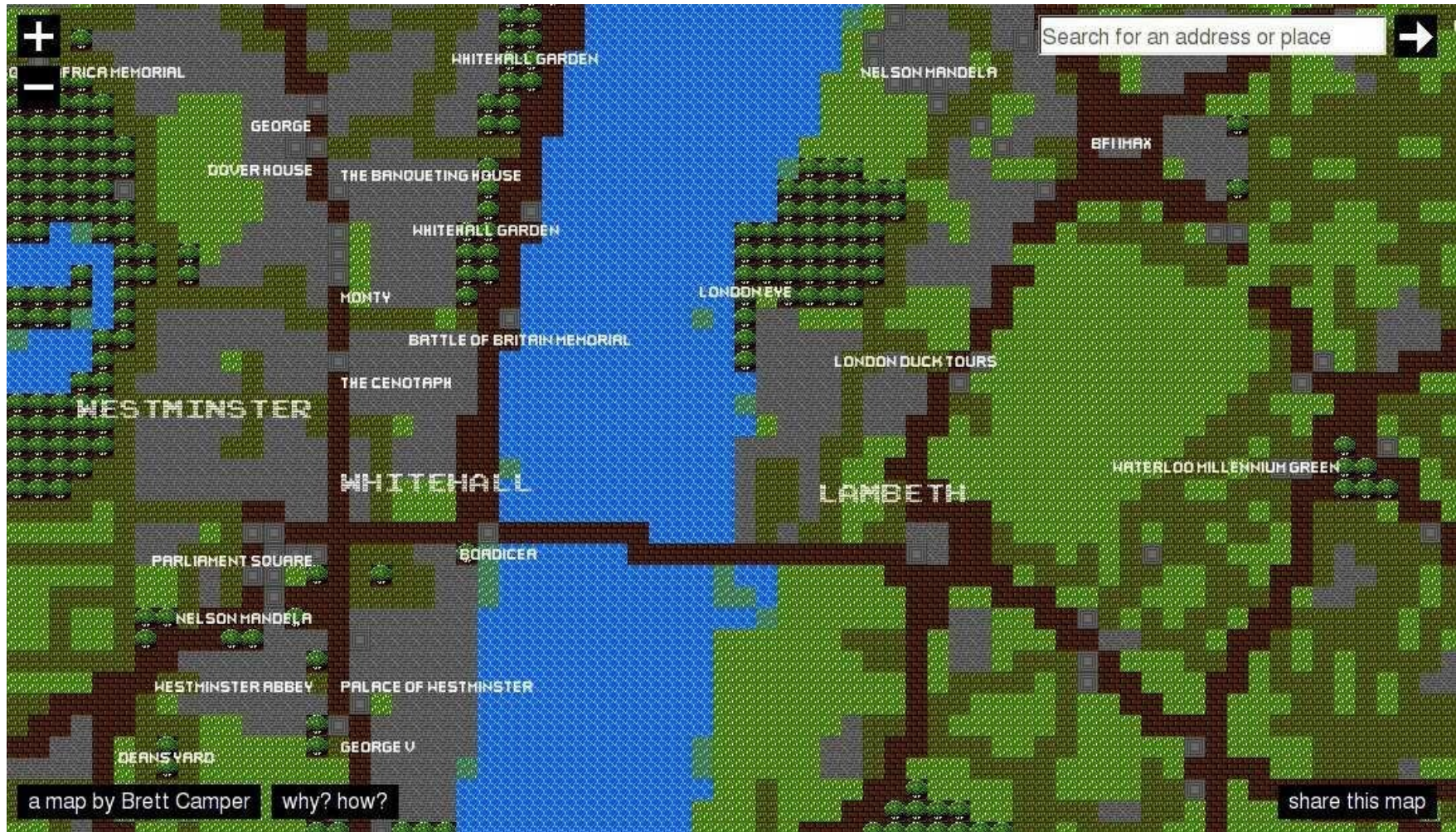


Making Your Own Maps

PGCon 2012



<http://8bitcity.com/map> CC-BY-SA OpenStreetMap & Contributors

Steve Singer
steve@ssinger.info

Steve Singer Amateur Cartographer



<http://www.flickr.com/photos/tinker-tailor/4284883477/>

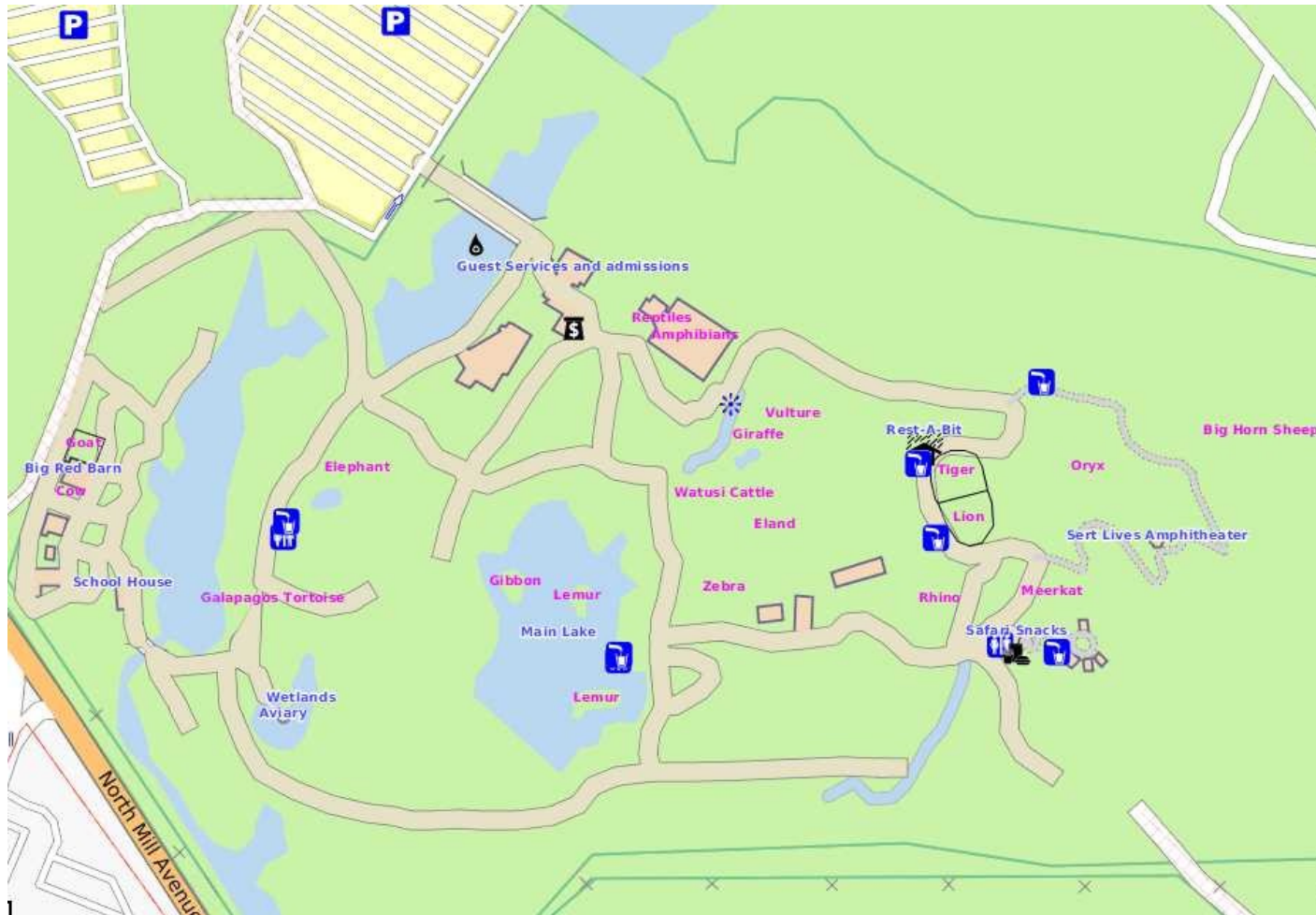
Why Make Your Own Maps

Volume



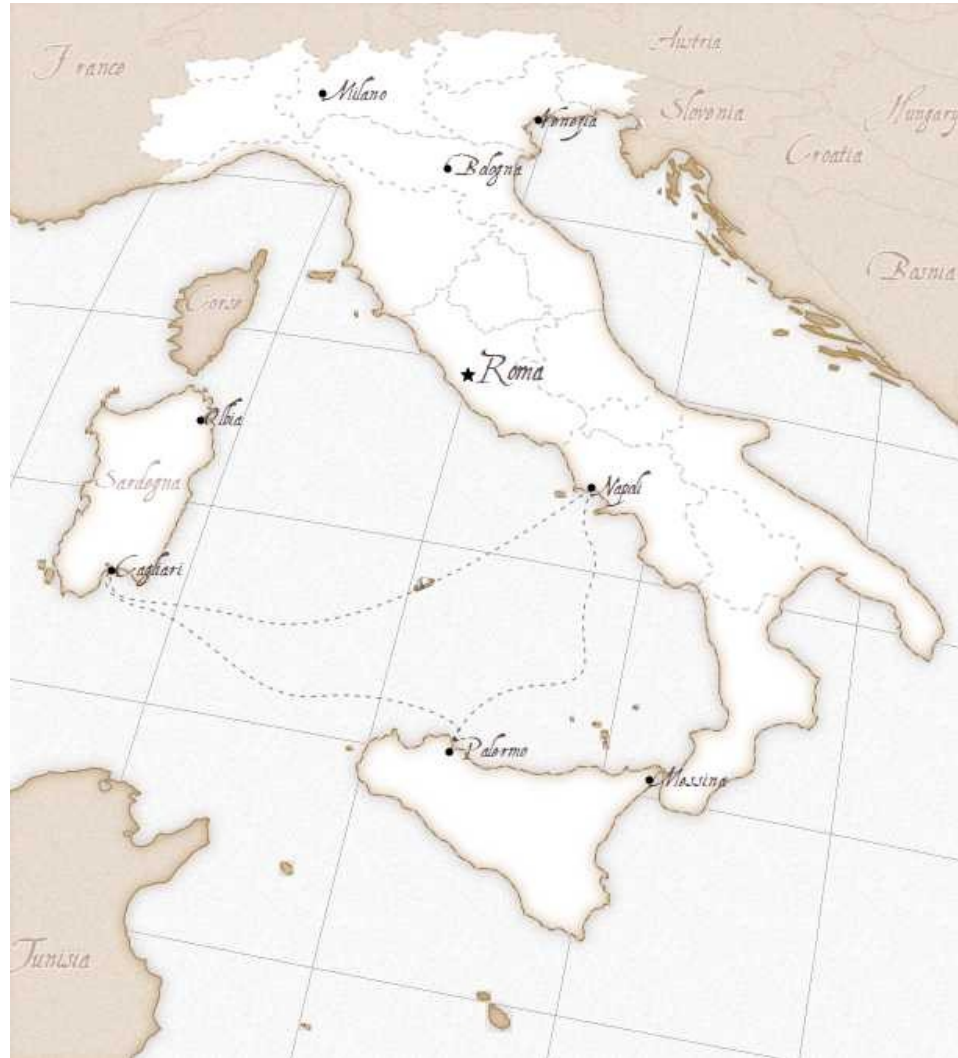
Why Make Your Own Maps

Show things not on other maps



Why Make Your Own Maps

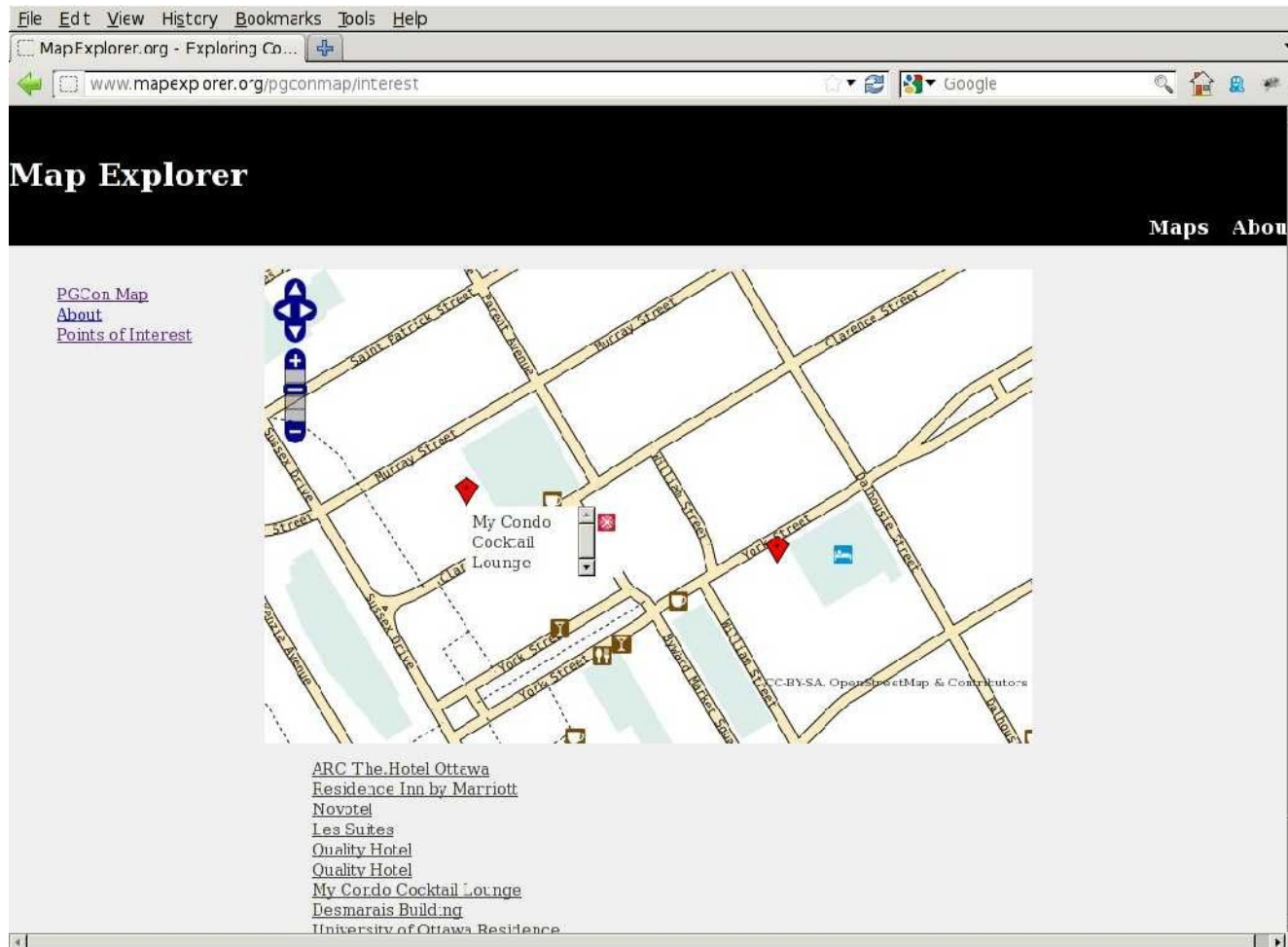
Only Show the things you care about



Making Their Own Maps



A Map to the party

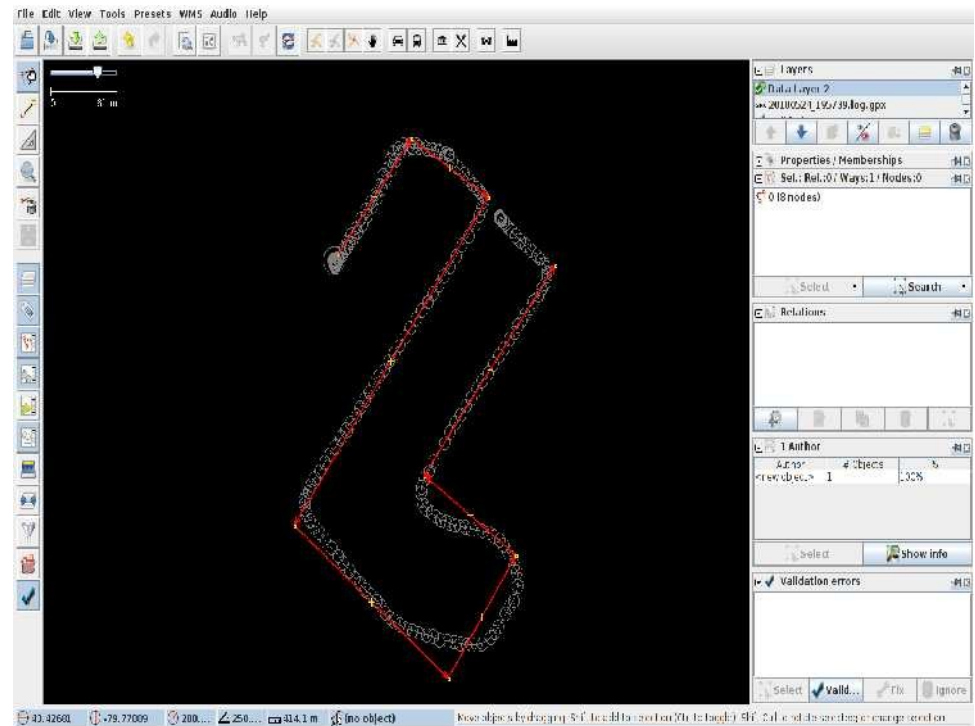


<http://www.mapexplorer.org/pgconmap>

OpenStreetMap



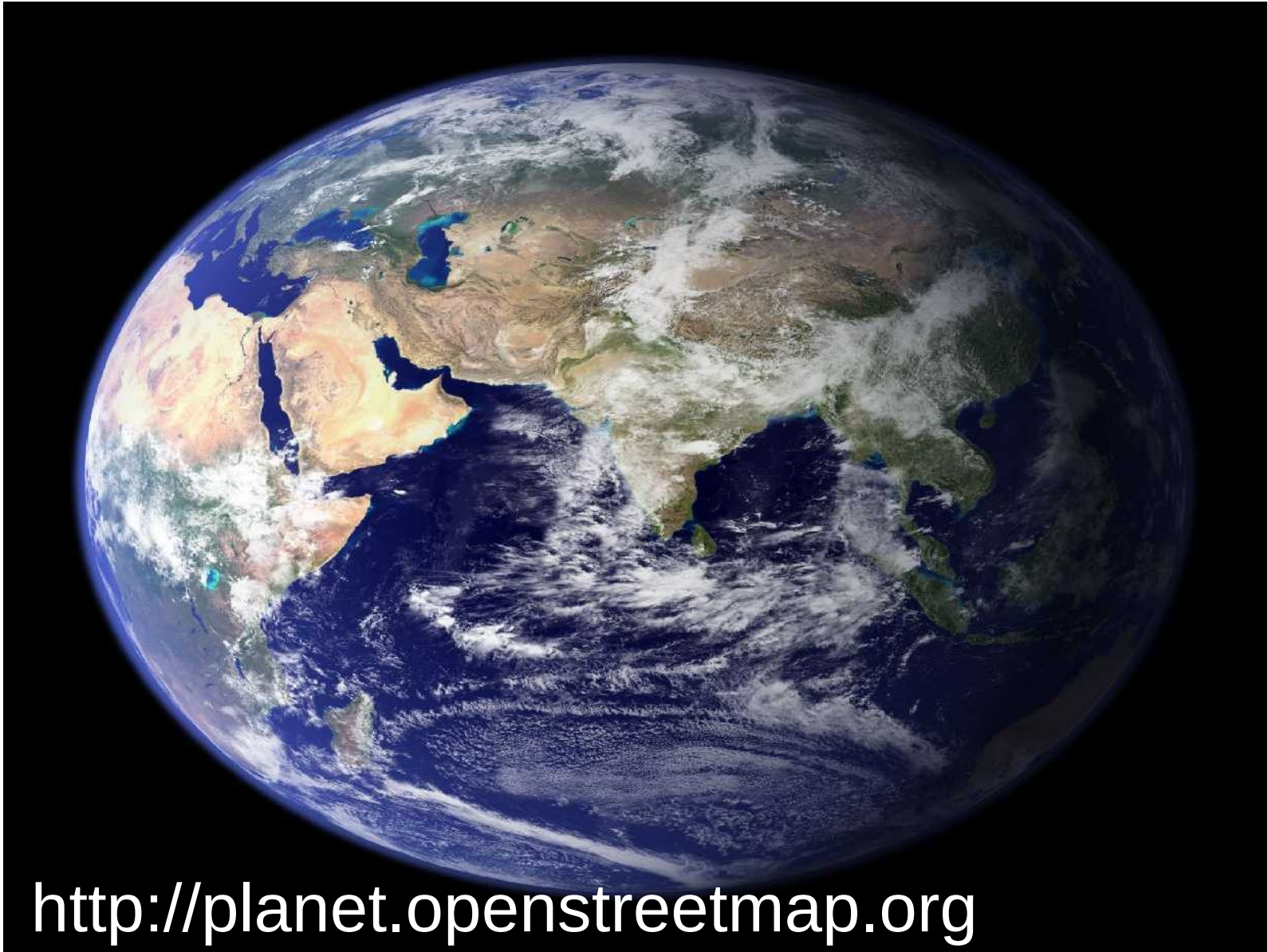
CC-BY Thomas Hasse



Free Data



Planet Dumps



Extracts – By Country

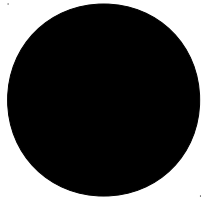
<http://download.geofabrik.de/osm/>

User defined Extracts

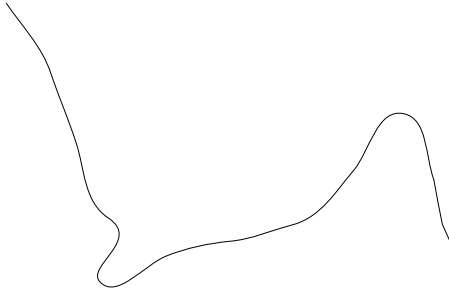
<http://www.overpass-api.de/>

```
wget -O map.osm 'http://overpass.osm.rambler.ru/cgi/interpreter?data=<union>
  <bbox-query s="45.41" n="45.43" w="-75.69" e="-75.67" />
  <recurse type="node-relation" into="rels"/>
  <recurse type="node-way"/>
  <recurse type="way-relation"/>
</union>
<print/>
'~
```


Three Types of Shapes



Points



Lines



Polygons

Schema Overview

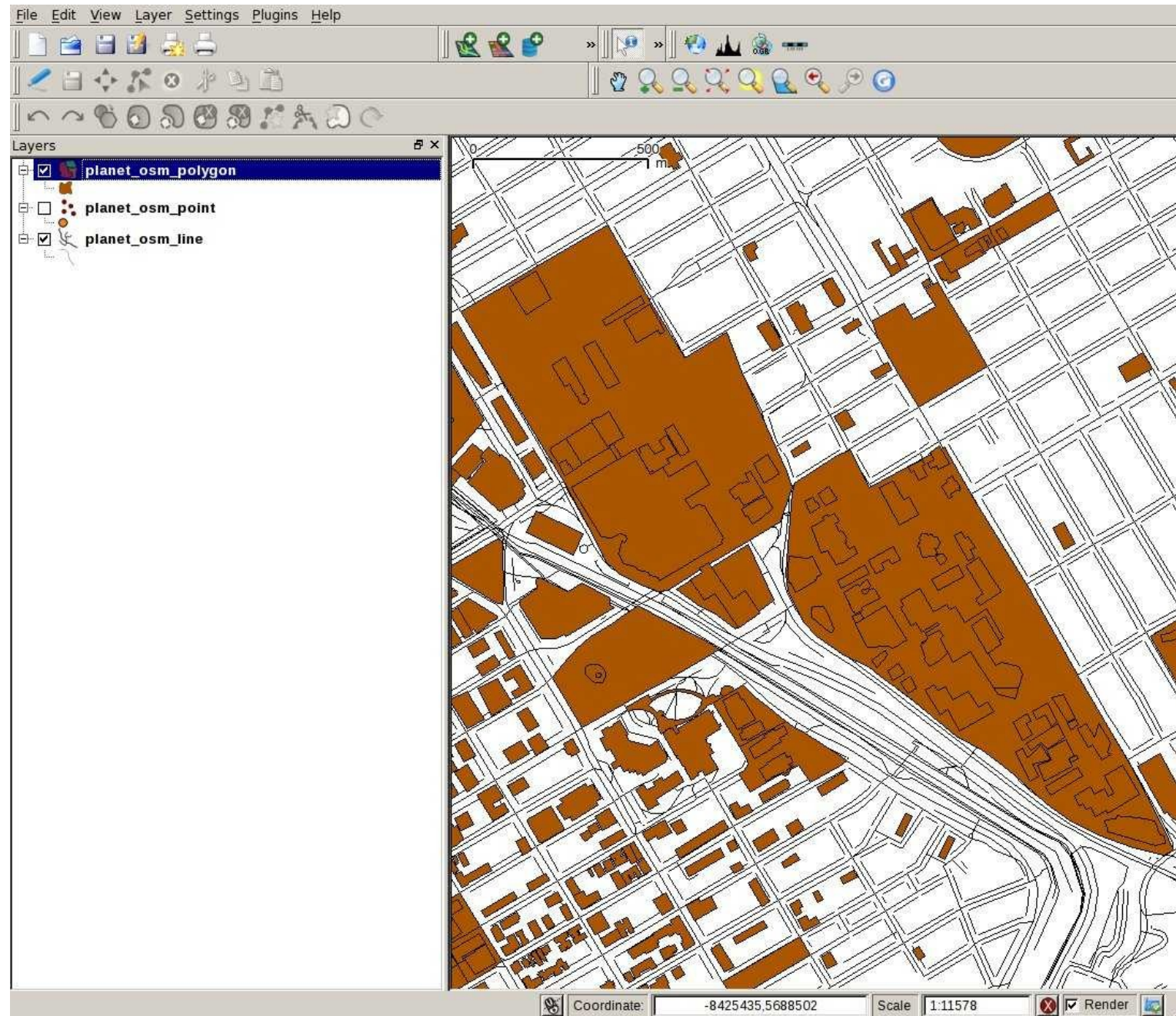
```
File Edit View Terminal Go Help
```

Table "public.planet_osm_line"

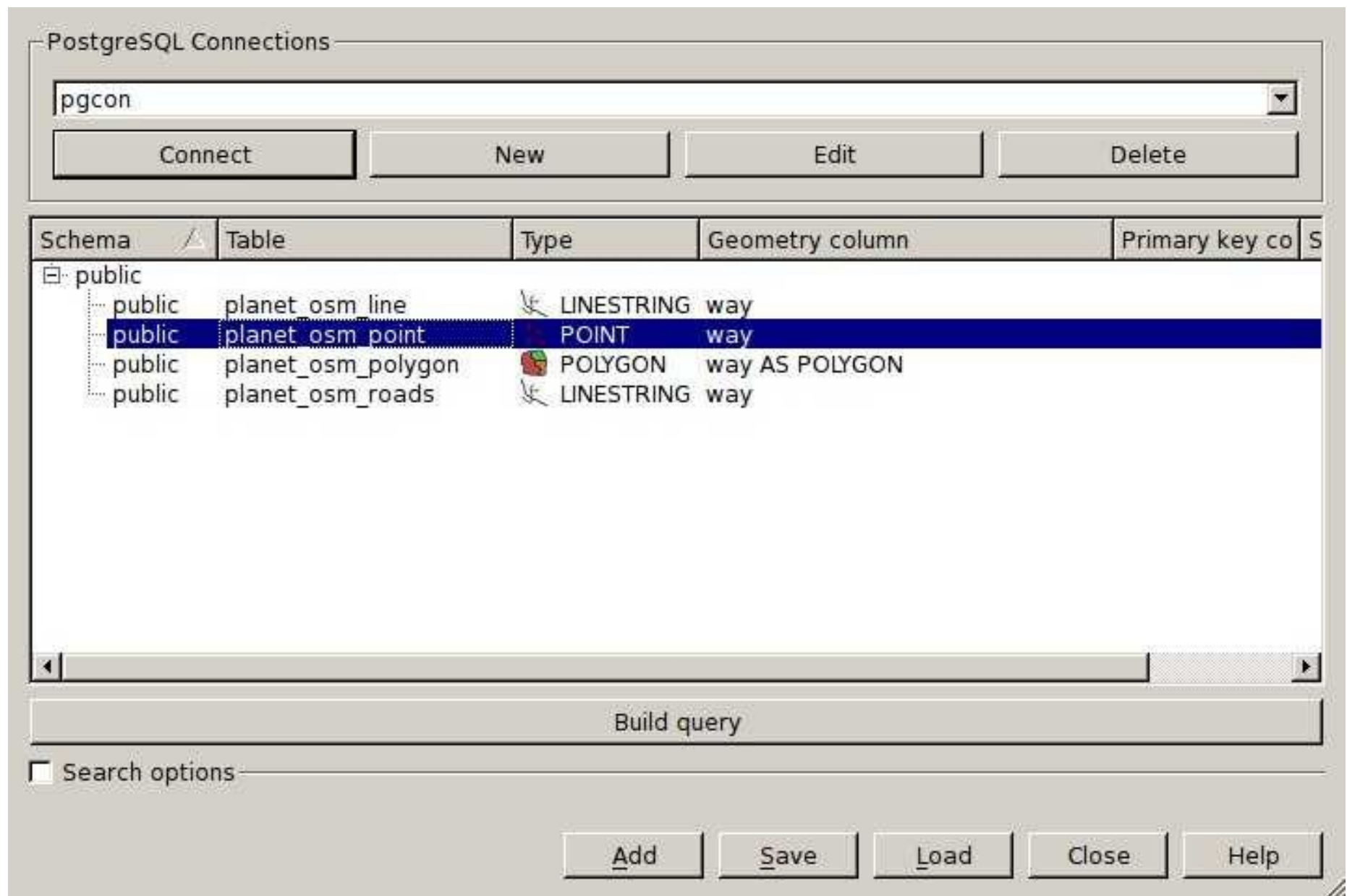
Column	Type	Modifiers
osm_id	integer	
access	text	
addr:flats	text	
addr:housenumber	text	
addr:interpolation	text	
admin_level	text	
aerialway	text	
aeroway	text	
amenity	text	
area	text	
barrier	text	
bicycle	text	
bridge	text	
boundary	text	

--More--

Qgis – A Desktop GIS: www.qgis.org



Exploring your data



Enter Query

dbname='pgcon' host=localhost port=5432 sslmode=disable key='ctid' table="planet_osm_point" (way) sql=amenity in ('bar','pub','cafe','restaurant')

Fields

osm_id
access
addr:flats
addr:housenumber
addr:interpolation
admin_level
aerialway
aeroway
amenity
area
barrier
bicycle
bridge
boundary
building

Values

Sample

All

Operators

=

<

>

LIKE

%

IN

NOT IN

<=

>=

!=

ILIKE

AND

OR

NOT

SQL where clause

amenity in ('bar','pub','cafe','restaurant')

OK

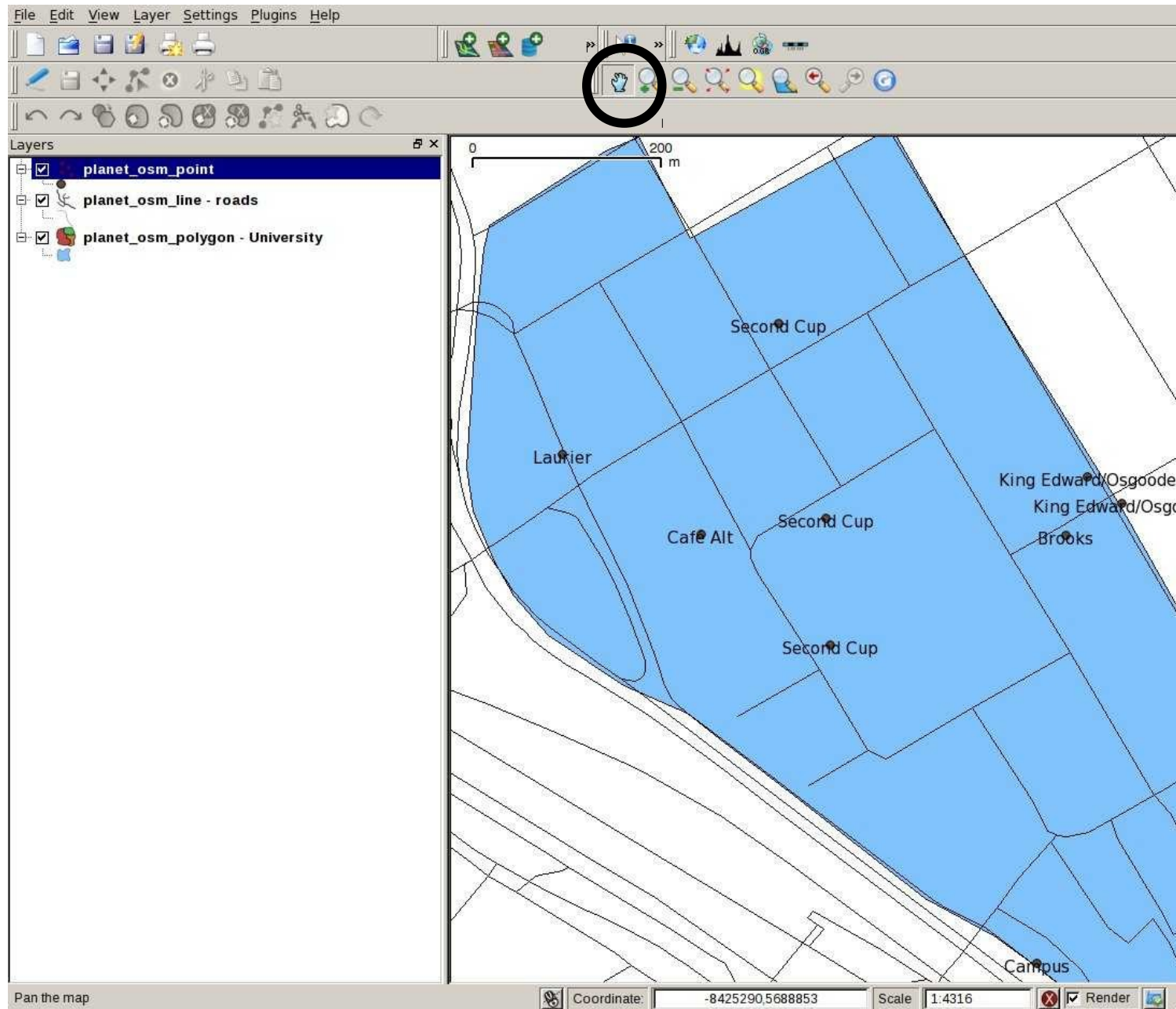
Test

Clear

Cancel

Help

Explore Data(Panning)



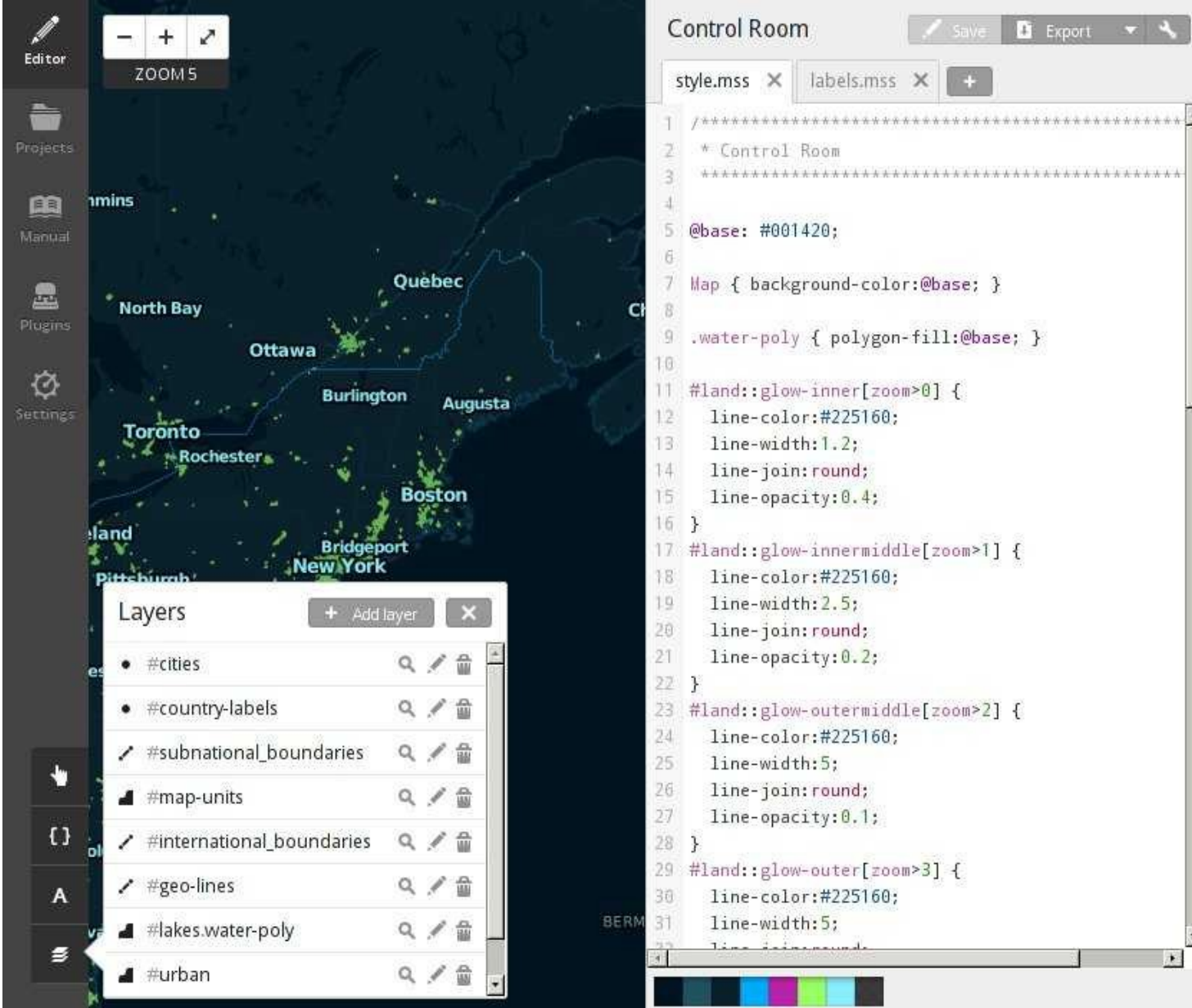
Explore Data (Attributes)

	aeroway	amenity	area	barrier	bicycle	bridge
0	NULL	NULL	NULL	NULL	NULL	NULL
1	NULL	NULL	NULL	NULL	NULL	NULL
2	NULL	NULL	NULL	NULL	NULL	NULL
3	NULL	NULL	NULL	NULL	NULL	NULL
4	NULL	NULL	NULL	NULL	NULL	NULL
5	NULL	NULL	NULL	NULL	NULL	NULL
6	NULL	pub	NULL	NULL	NULL	NULL
7	NULL	pub	NULL	NULL	NULL	NULL
8	NULL	NULL	NULL	NULL	NULL	NULL
9	NULL	fuel	NULL	NULL	NULL	NULL
10	NULL	NULL	NULL	NULL	NULL	NULL
11	NULL	bank	NULL	NULL	NULL	NULL
12	NULL	hospital	NULL	NULL	NULL	NULL
13	NULL	NULL	NULL	NULL	NULL	NULL
14	NULL	pub	NULL	NULL	NULL	NULL
15	NULL	NULL	NULL	NULL	NULL	NULL
16	NULL	NULL	NULL	NULL	NULL	NULL
17	NULL	NULL	NULL	NULL	NULL	NULL
18	NULL	bus_station	NULL	NULL	NULL	NULL
19	NULL	bus_station	NULL	NULL	NULL	NULL
20	NULL	NULL	NULL	NULL	NULL	NULL
21	NULL	school	NULL	NULL	NULL	NULL

Look for in

Show selected records only Search selected records only

Tilemill: mapbox.com/tilemill



The screenshot displays the Tilemill web application interface. On the left, a dark sidebar contains navigation icons for Editor, Projects, Manual, Plugins, and Settings. The main map area shows a dark-themed map of the Great Lakes region with city labels like Toronto, Ottawa, and Quebec. A 'Layers' panel is open in the bottom-left, listing layers such as #cities, #country-labels, #subnational_boundaries, #map-units, #international_boundaries, #geo-lines, #lakes.water-poly, and #urban. The right side features a 'Control Room' panel with a code editor showing map style code (MSS) for 'Control Room'. The code includes a base color and various glow effects for land features.

Control Room Save Export

style.mss x labels.mss x +

```
1 /*****  
2 * Control Room  
3 *****/  
4  
5 @base: #001420;  
6  
7 Map { background-color:@base; }  
8  
9 .water-poly { polygon-fill:@base; }  
10  
11 #land::glow-inner[zoom>0] {  
12   line-color:#225160;  
13   line-width:1.2;  
14   line-join:round;  
15   line-opacity:0.4;  
16 }  
17 #land::glow-innermiddle[zoom>1] {  
18   line-color:#225160;  
19   line-width:2.5;  
20   line-join:round;  
21   line-opacity:0.2;  
22 }  
23 #land::glow-outermiddle[zoom>2] {  
24   line-color:#225160;  
25   line-width:5;  
26   line-join:round;  
27   line-opacity:0.1;  
28 }  
29 #land::glow-outer[zoom>3] {  
30   line-color:#225160;  
31   line-width:5;  
32   line-join:round;  
33 }
```

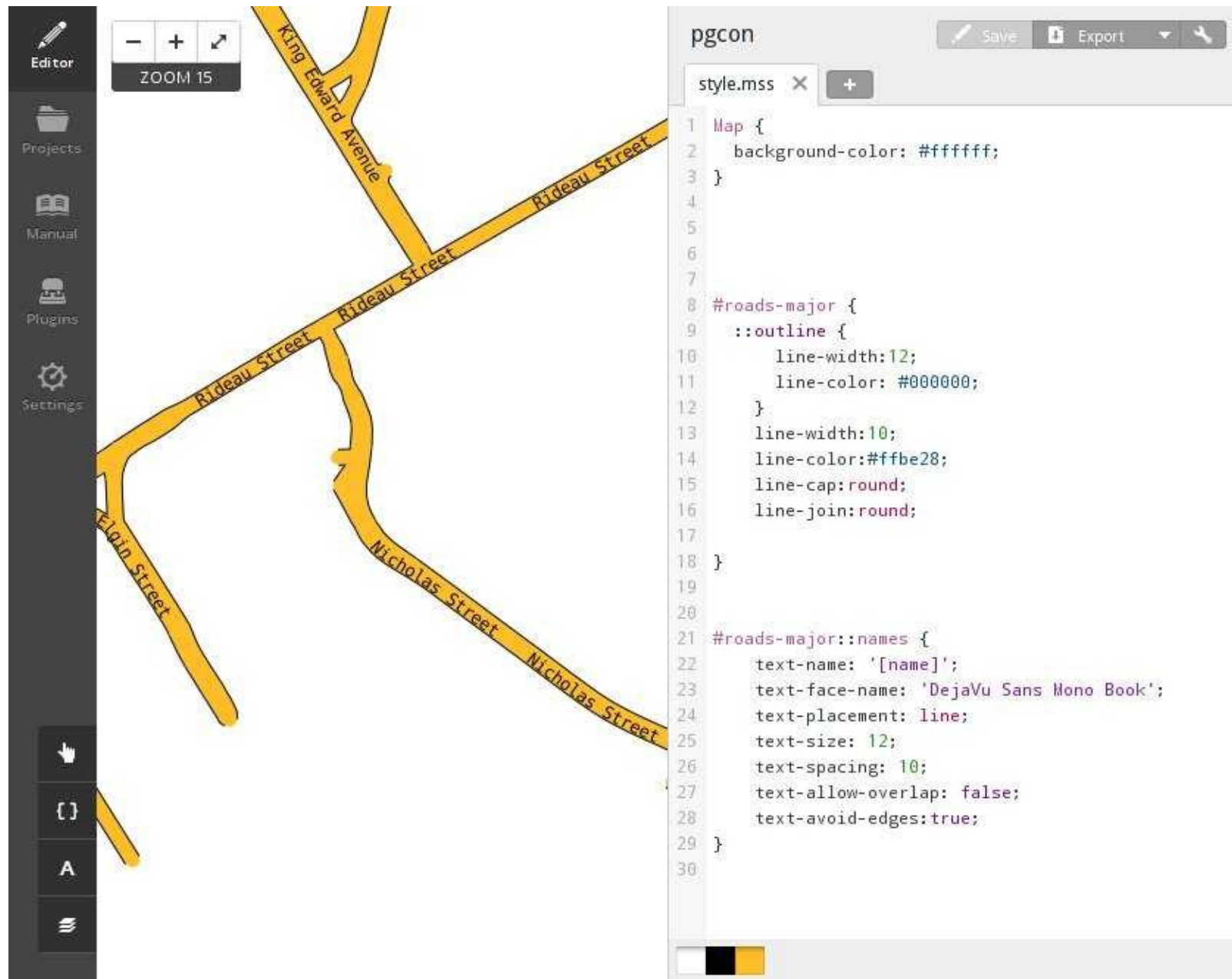

Define A Layer

The screenshot shows the 'Edit roads-major' dialog box in QGIS. The dialog is titled 'Edit roads-major' and has a close button (X) in the top right corner. It features a tabbed interface with 'File', 'SQLite', and 'PostGIS' tabs. The 'PostGIS' tab is active. The dialog contains the following fields and options:

- ID:** roads-major (select in Carto #id)
- Class:** roads (select in Carto .class)
- * Connection:** dbname=pgcon host=localhost (Provide your PostGIS authentication and connection parameters. Includes a 'Browse' button and a star icon).
- * Table or subquery:** (SELECT * FROM planet_osm_line where highway in ('primary','motorway','motorway_link','primary_link') as data)
- Extent:** (limit the query by this bounding box)
- Unique key field:** osm_id (SQL field containing a unique key for each feature)
- Geometry field:** way (SQL field containing feature geometry)
- SRS:** 900913 (+proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +) (SRS projection string for this datasource. TileMill can often autodetect this value.)
- Advanced:** option1=? option2=? (Optional, advanced arguments to pass to Mapnik.)

At the bottom of the dialog are 'Save' and 'Cancel' buttons. In the bottom-left corner of the QGIS interface, the 'Layers' icon is circled in red.

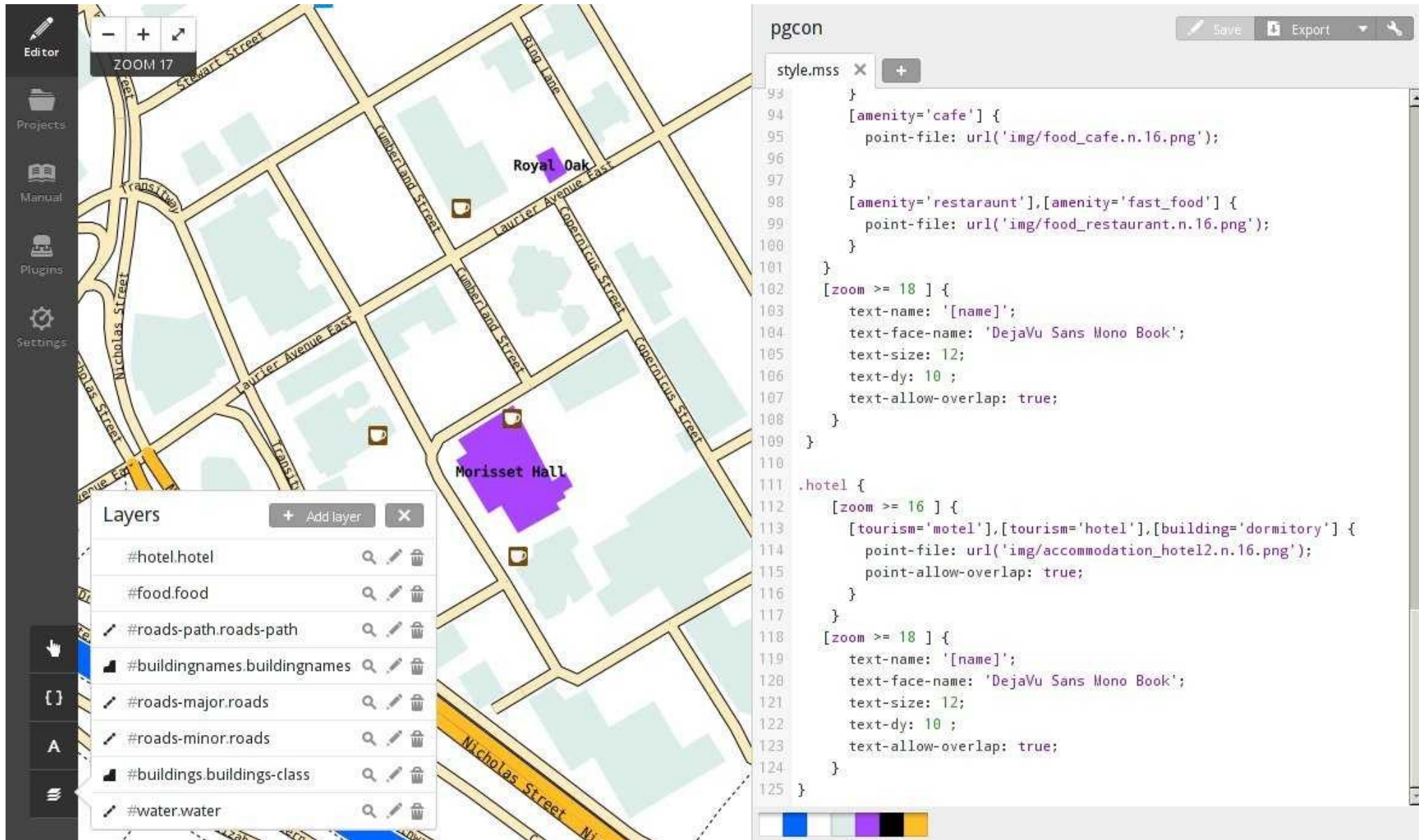
Define A Style



The image shows a map editor interface. On the left is a dark sidebar with icons for Editor, Projects, Manual, Plugins, and Settings. The main map area displays a network of roads: King Edward Avenue, Rideau Street, Elgin Street, and Nicholas Street. The roads are styled with thick yellow lines and black outlines. A zoom control at the top left shows 'ZOOM 15'. On the right, a window titled 'pgcon' displays a style definition file named 'style.mss'. The code defines a map background and styles for major roads, including line width, color, and text labels.

```
pgcon
style.mss
1 Map {
2   background-color: #ffffff;
3 }
4
5
6
7
8 #roads-major {
9   ::outline {
10    line-width:12;
11    line-color: #000000;
12  }
13  line-width:10;
14  line-color:#ffbe28;
15  line-cap:round;
16  line-join:round;
17 }
18 }
19
20
21 #roads-major::names {
22   text-name: '[name]';
23   text-face-name: 'DejaVu Sans Mono Book';
24   text-placement: line;
25   text-size: 12;
26   text-spacing: 10;
27   text-allow-overlap: false;
28   text-avoid-edges:true;
29 }
30
```

Define Some More Layers + Styles



The image displays a map editor interface. On the left, a vertical sidebar contains icons for Editor, Projects, Manual, Plugins, and Settings. The main map area shows a street grid with buildings highlighted in purple. Two buildings are labeled: "Royal Oak" and "Morisset Hall". A zoom control at the top left indicates "ZOOM 17".

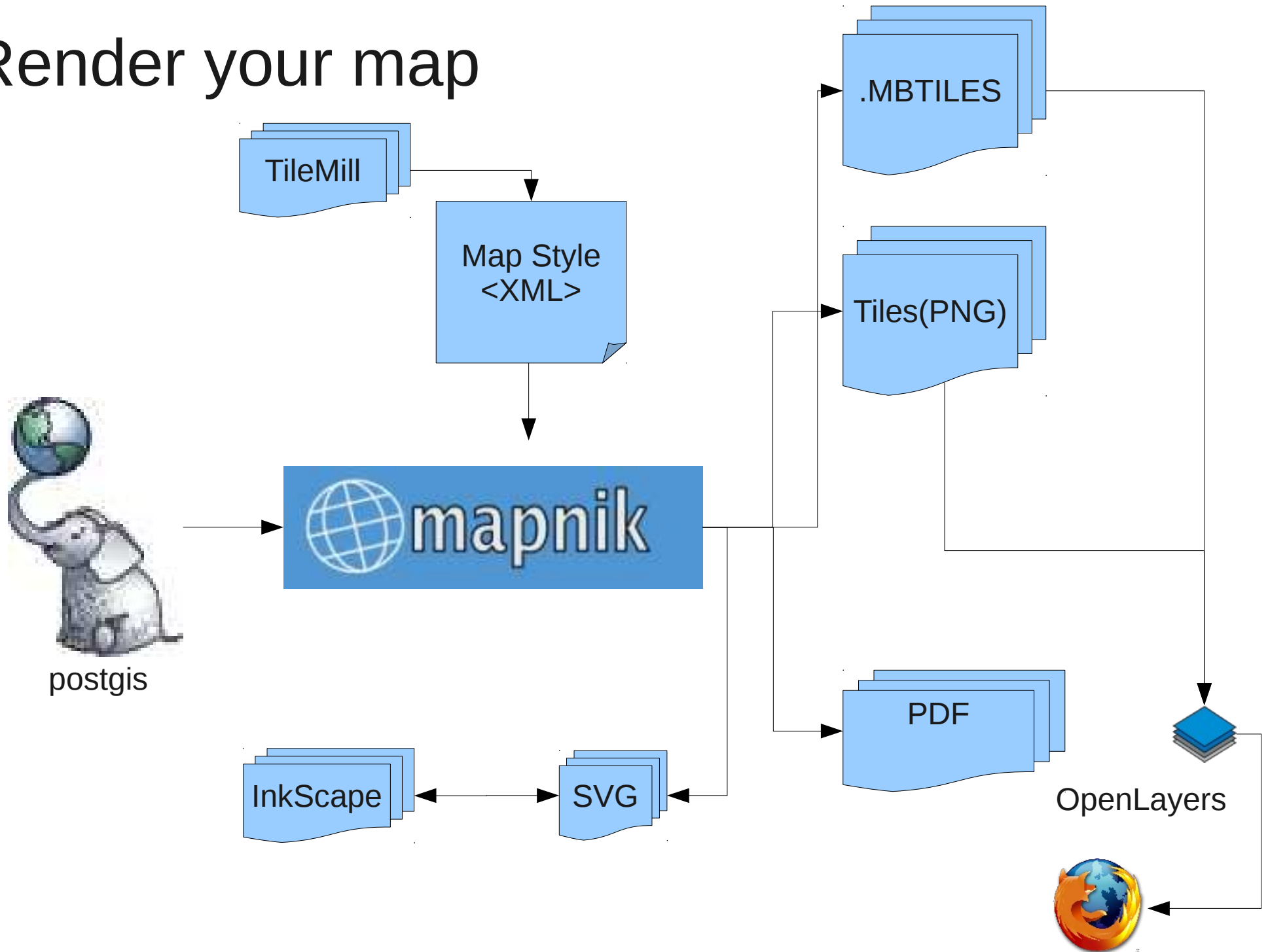
In the bottom-left corner, a "Layers" panel is visible, listing several layers with search, edit, and delete icons:

- #hotel.hotel
- #food.food
- #roads-path.roads-path
- #buildingnames.buildingnames
- #roads-major.roads
- #roads-minor.roads
- #buildings.buildings-class
- #water.water

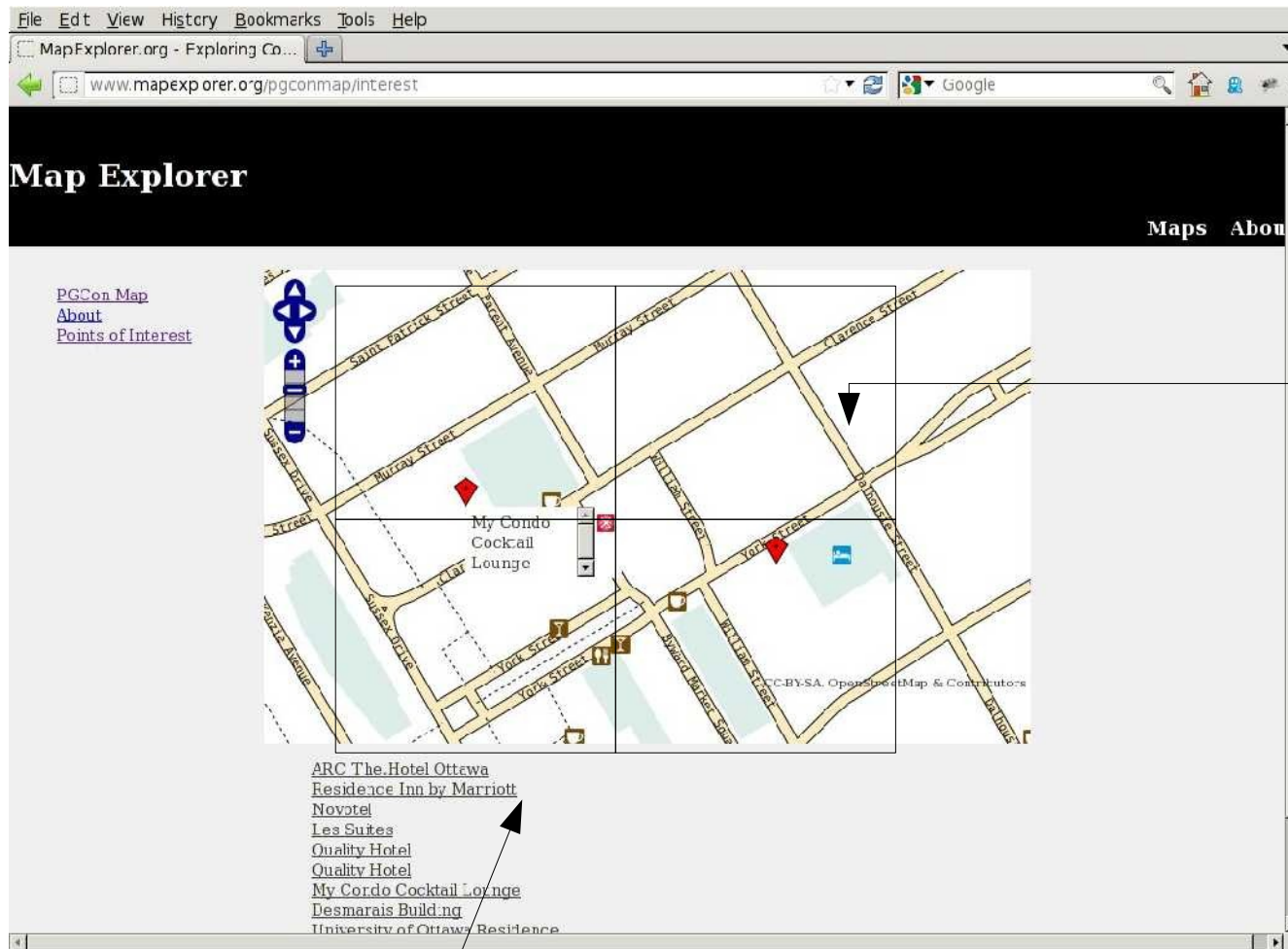
On the right, a "pgcon" window displays a style definition file named "style.mss". The code defines styles for points and text based on amenity, zoom level, and building type.

```
93 }
94 [amenity='cafe'] {
95   point-file: url('img/food_cafe.n.16.png');
96 }
97 }
98 [amenity='restaraunt'],[amenity='fast_food'] {
99   point-file: url('img/food_restaurant.n.16.png');
100 }
101 }
102 [zoom >= 18 ] {
103   text-name: '[name]';
104   text-face-name: 'DejaVu Sans Mono Book';
105   text-size: 12;
106   text-dy: 10 ;
107   text-allow-overlap: true;
108 }
109 }
110 }
111 .hotel {
112   [zoom >= 16 ] {
113     [tourism='motel'],[tourism='hotel'],[building='dormitory'] {
114       point-file: url('img/accommodation_hotel2.n.16.png');
115       point-allow-overlap: true;
116     }
117   }
118 }
119 [zoom >= 18 ] {
120   text-name: '[name]';
121   text-face-name: 'DejaVu Sans Mono Book';
122   text-size: 12;
123   text-dy: 10 ;
124   text-allow-overlap: true;
125 }
```


Render your map



Web Mapping 101



Tiles

Javascript

Pre-Render in Tilemill

- Editor
- Projects
- Manual
- Plugins
- Settings

- +
ZOOM 15



Export MBTiles

Name:

Description:

Attribution:

Version:

Filename: .mbtiles

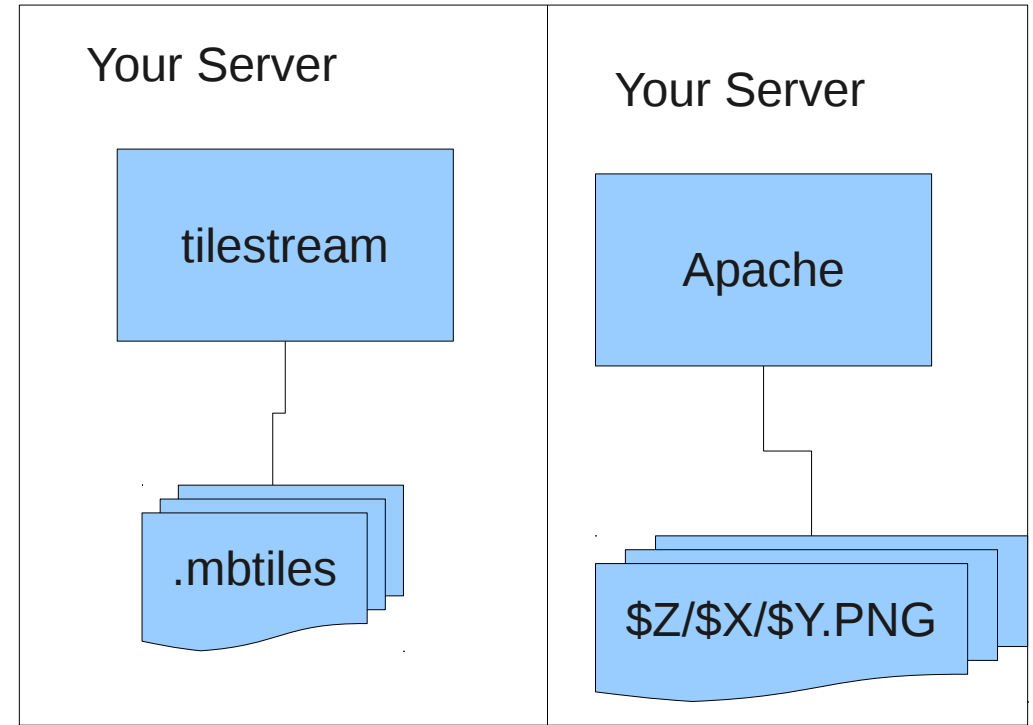
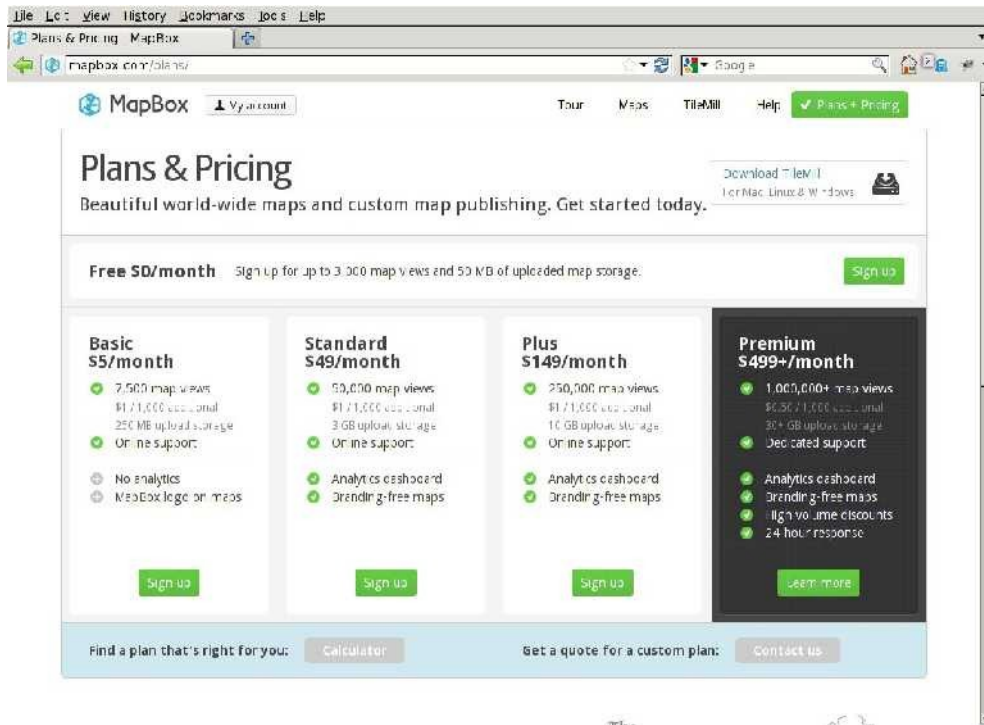
Zoom:
519 tiles (1 MB)

Center:
Click to move starting center point.

Bounds:
Shift + drag to select bounds.

Save settings to project

Host your tiles



<http://mapbox.com>

<https://github.com/mapbox/tilestream>

So you want to render planet

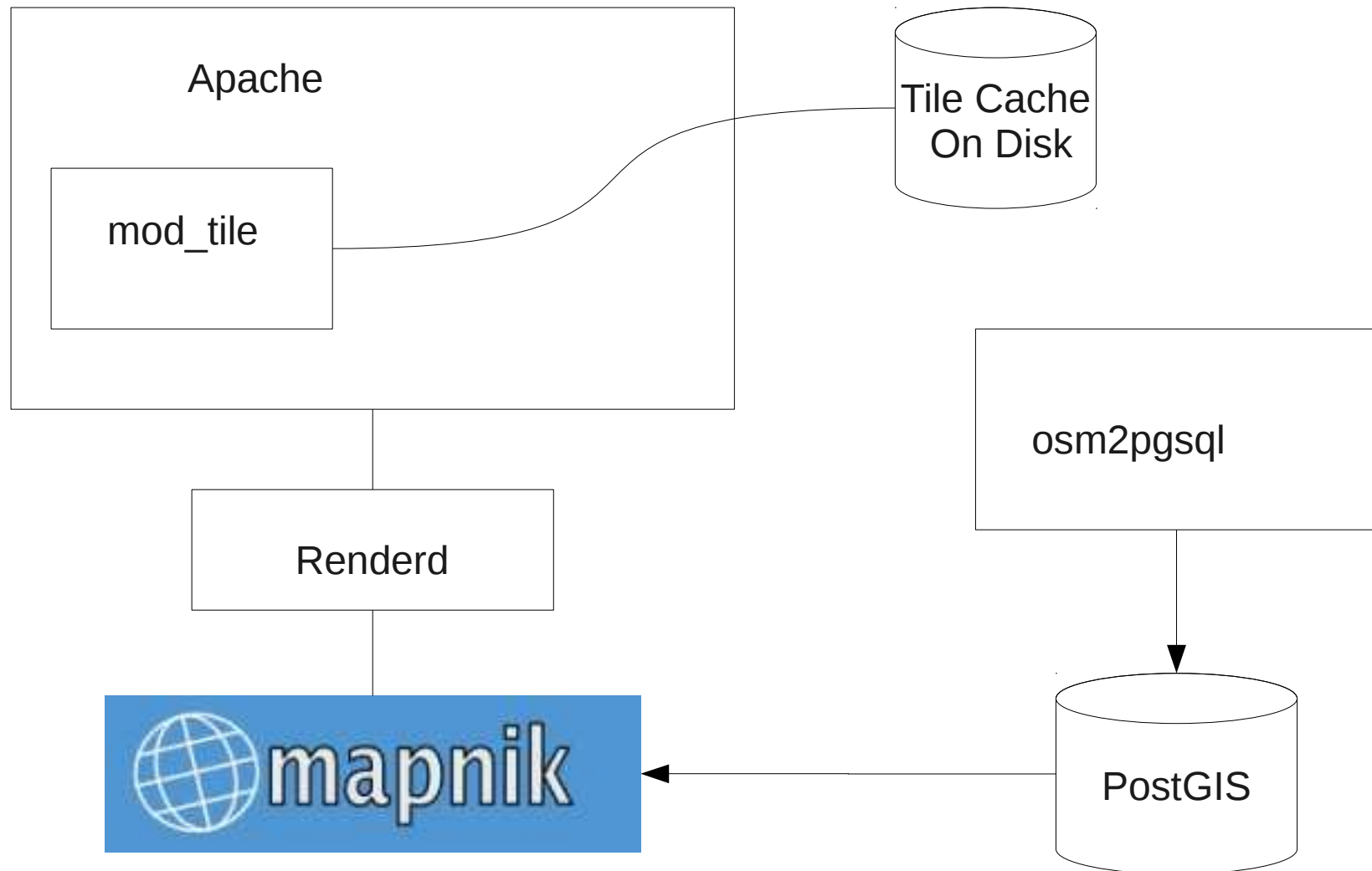
Render Tiles On Disk (PNG)

Zoom Level	Cumulative Storage
0	2 KB
5	3 MB
10	3 GB
14	683 GB
18	171 TB

Source: OpenStreetMap Using and Enhancing the Free Map of the World. Ramm, Topf, 2011

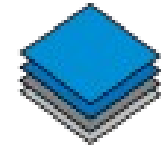
PostGIS Database Server	
Disk Space Required	600 GB
Import Time	8 – 60 Hours
Recommended RAM to apply diffs	24 GB

Tile Servers



<http://switch2osm.org> for instructions

OpenLayers: www.openlayers.org



Javascript Map Display Library

File Edit View History Bookmarks Tools Help

MapExplorer.org - Exploring Co...

www.mapexplorer.org/pgconmap

Map Explorer

Maps About

[PGCon Map](#)
[About](#)
[Points of Interest](#)

Desmarais Building

A Map for PGCon Attendees

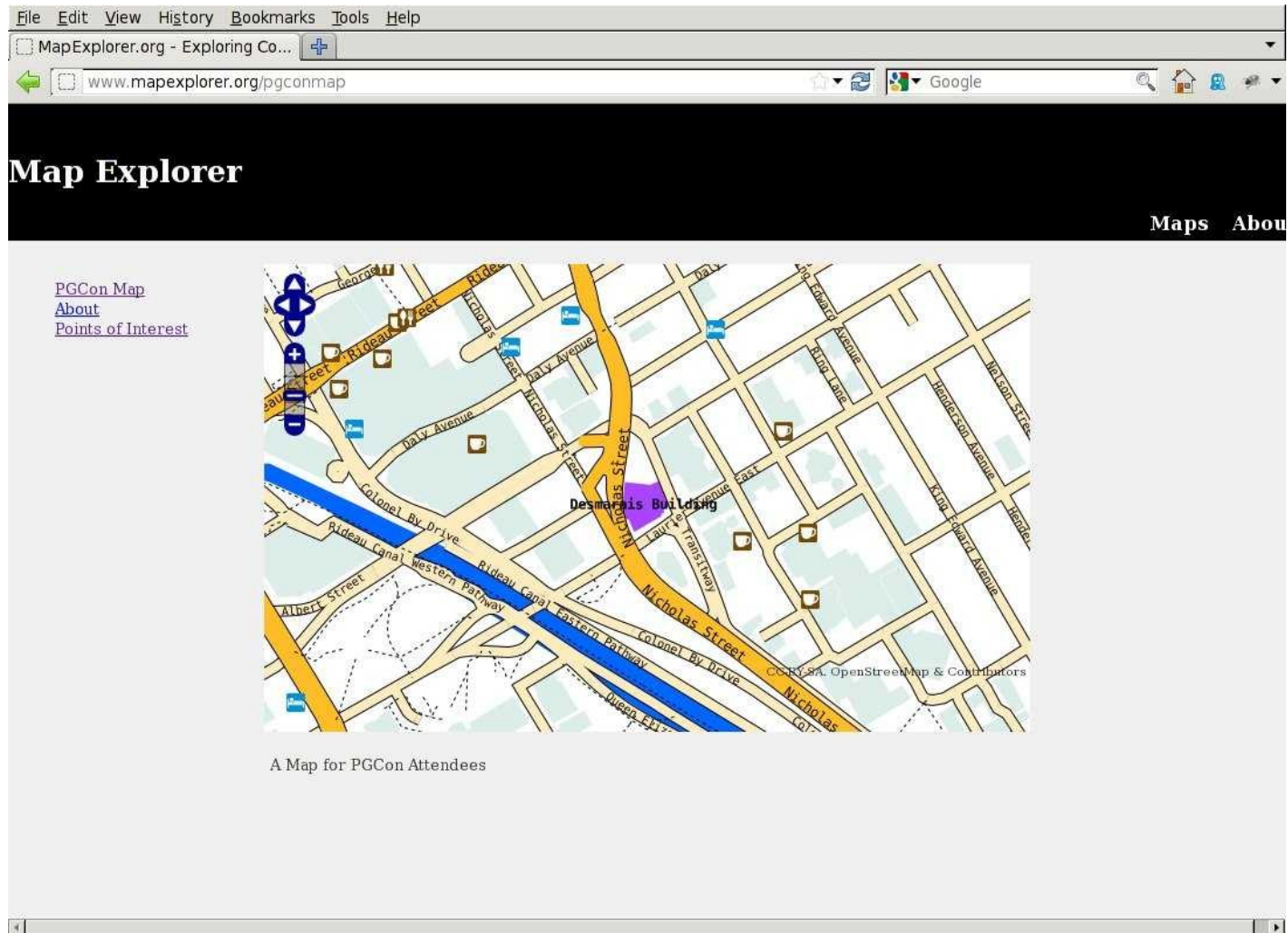
OpenLayers – Create Your Map

```
<div id="map" ></div></div>
<script defer="defer" type="text/javascript">
  var map, layer;
  var bounds = new OpenLayers.Bounds(-75.7001, 45.4166,
    -75.674, 45.435).transform(new OpenLayers.Projection
    ("EPSG:4326"),
    new OpenLayers.Projection("EPSG:900913"))
  map = new OpenLayers.Map( 'map',
    {
      maxExtent: bounds,
      projection: "EPSG:900913",
      units: "degree",
      restrictedExtent: bounds,
      controls : [new OpenLayers.Control.Attribution(),
        new OpenLayers.Control.PanZoomBar(),
        new OpenLayers.Control.Navigation()]
    });
```

OpenLayers: Add Your Layer

```
layer = new OpenLayers.Layer.XYZ("PGCon Map",
  ["http://a.tiles.mapbox.com/v3/ssinger.pgcon/{z}/{x}/{y}.png",
  "http://b.tiles.mapbox.com/v3/ssinger.pgcon/{z}/{x}/{y}.png",
  "http://c.tiles.mapbox.com/v3/ssinger.pgcon/{z}/{x}/{y}.png"]
  , { attribution: "CC-BY-SA. OpenStreetMap & Contributors",
    sphericalMercator: true,
    minZoomLevel: 15,
    maxZoomLevel: 18,
    numZoomLevels:4,
    zoomOffset: 15,
    resolutions: [ 4.777314266967774, 2.388657133483887,
                  1.1943285667419434, 0.597164283],
    units:"degree",
    projection: "EPSG:900913",
  }
  map.addLayer(layer)
```


PGCON Map



https://github.com/ssinger/mapexplorer/blob/master/app/views/pgconmap/_map.html.erb

Adding Markers

XMLHttpRequest()

JSON Response

```
[{"lat":45.4259345,"lon":-75.6896723,"description":"Novotel"}  
{43,"lon":-75.6869527,"description":"Desmarais Building"}]
```

HTTP GET - /pgconmap/interests



mapexplorer.org

Table: points

lat	lon	point_type	description
45.42593	-75.689	pgcon	Novotel

Adding Markers:

```
var points=JSON.parse(client.responseText);
```


Adding Markers:

```
var markers = new  
OpenLayers.Layer.Markers("markers");
```

```
map.addLayer(markers);
```

Adding Markers:

```
for(var i=0; i < points.length; i++) {  
    var p=new OpenLayers.LonLat(points[i]['lon'],  
                                points[i]['lat']).  
        transform(  
            new OpenLayers.Projection  
                ("EPSG:4326"),  
            new OpenLayers.Projection("EPSG:900913"));  
    var marker=new OpenLayers.Marker(p,icon.clone());  
    markers.addMarker(marker);  
}
```


Adding Popups



```
PointClickHandler.prototype.onEvent = function(evt) {  
  
  var popup=new OpenLayers.Popup(null,this.latlon,  
  new OpenLayers.Size(100,50),this.description,true);  
  map.addPopup(popup);  
}
```



Printed Maps



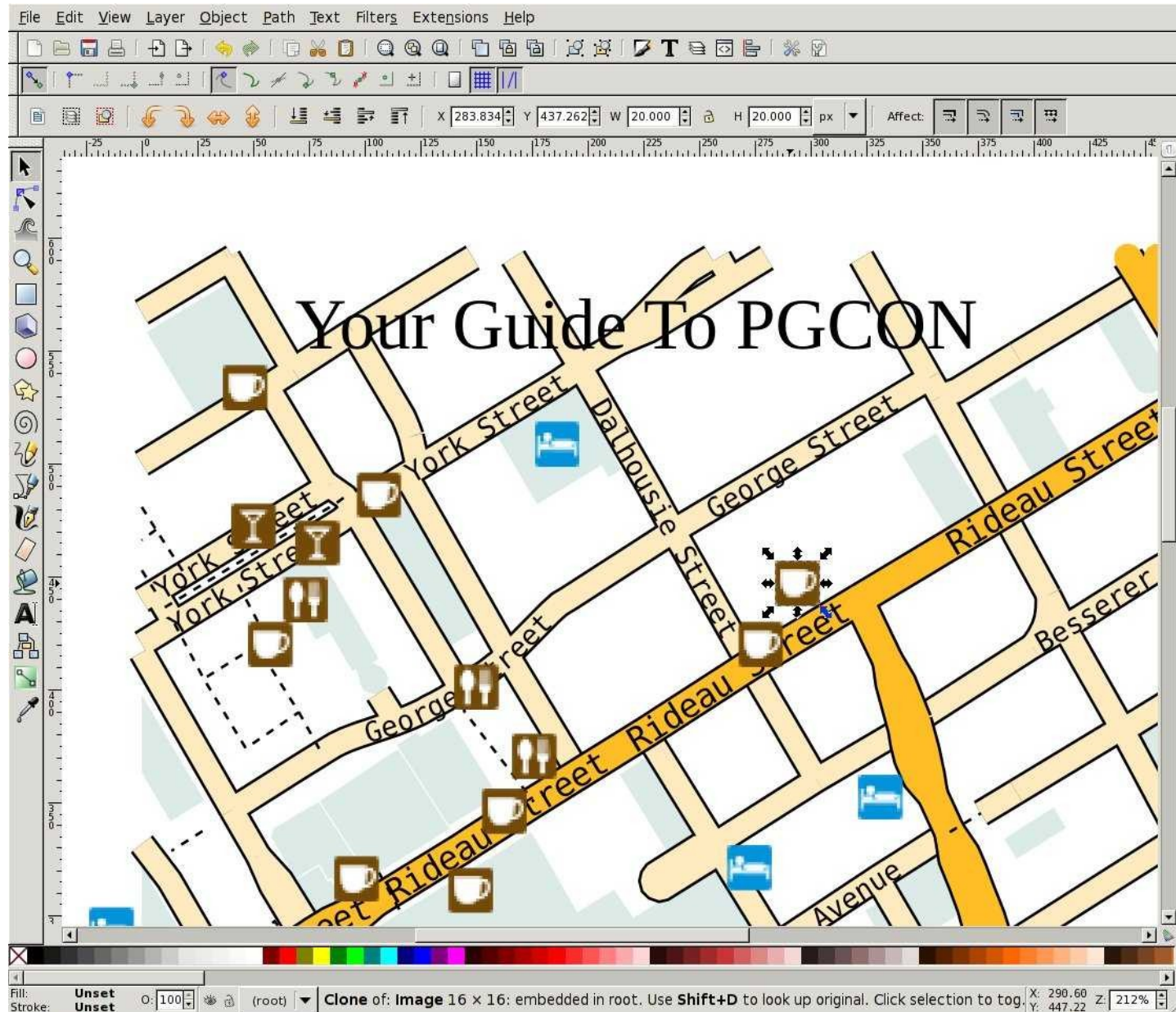
SVG Export

The image shows a map application interface with a dark sidebar on the left containing icons for Editor, Projects, Manual, Plugins, and Settings. The main map area displays a street grid with several buildings highlighted in purple. Labels on the map include "Royal Oak", "Second Cup", "Café Alt", and "Morisset Hall". Street names visible are "Laurier Avenue East", "Cobden Street", "Cumberland Street", "Université Privée", "King Edward Avenue", "Louis J. Bastien Private", and "Trinity Street".

On the right side, an "Export SVG" dialog box is open. It contains the following fields and controls:

- Filename: .svg
- Size: x
- Bounds: 
Shift + drag to select bounds.
- Buttons:

Editing Your Map – inkscape



Printing



Additional Information

<http://www.openstreetmap.org>

<http://www.switch2osm.org>

<http://www.mapexplorer.org/pgconmap>

<http://github.com/ssinger/pgcon.map>

<http://github.com/ssinger/mapexplorer>

Steve Singer

<http://scanningpages.wordpress.com>

steve@ssinger.info

