

Map of Corsica

by Thomas Zuberbühler

December 2015 | Visualization of Spatial Data with SVG | UNIGIS

Using an early version of the Atom.io editor, static and animated SVG graphics were created as a portion of the requirements of the UNIGIS Masters of Science (MSc) program at the Department of GeoInformatics - Z_GIS, University of Salzburg, Austria. This page presents two selected SVG works. The first SVG map is a static basemap with some animation and the second one is interactive and allows to show different thematic layers.

Basemap with Animations

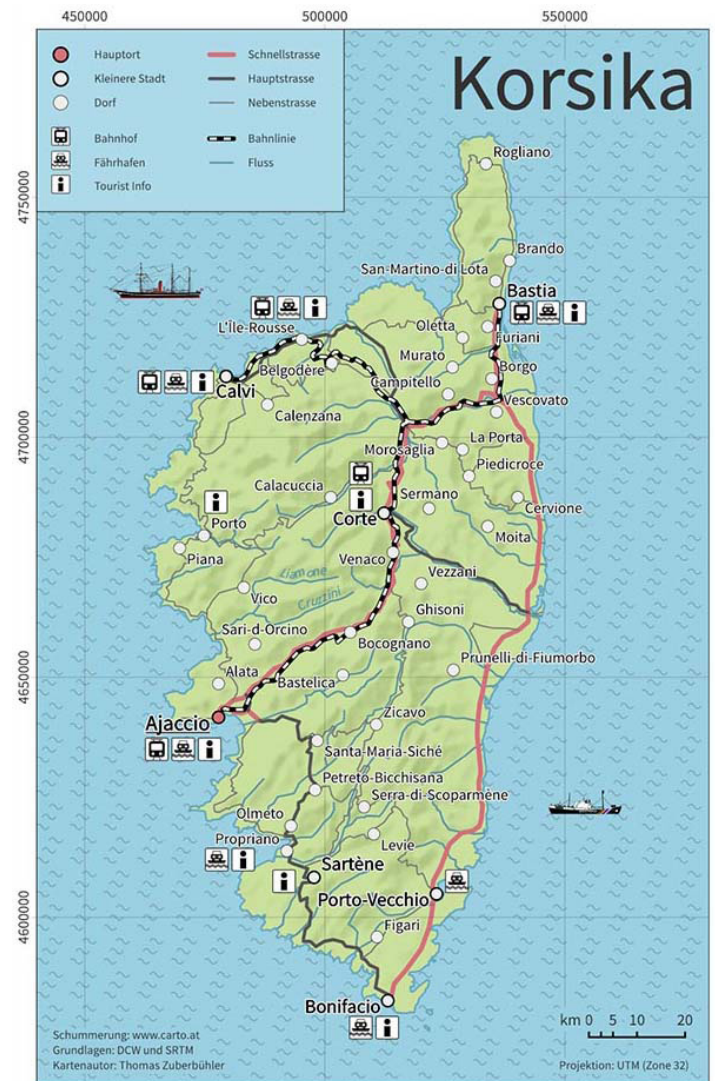
The provided, unedited SVG files had to be cleaned-up in a first step. Then, the features were symbolized using CSS and the label were positioned using SVG transformations such as `translate()`. In order to automatize the halo effect for all labels, TypeScript (which compiles to JavaScript) was used and has allowed to reuse the halo effects in other works as well. And finally, CSS animations were used to animate the waves and boats.

Thematic Map

In this SVG map, thematic layers can be switched through JavaScript. Also, a data handler was implemented in order to load data from CSV and JSON data sources. No external JavaScript libraries were allowed, therefore an own implementation was needed.

Ruby Script

Ruby is a powerful script language and was optionally used in one of the sub assignments. Due to not having had access to tools such as FME, a ruby script was written to transform a CSV file to a JSON file.



Software

Atom.io text editor

Programming Languages

SVG

CSS

JavaScript

Ruby

More Images on:

<https://www.geomo.ch>

```
private _finalizeHaloElement(haloElement:Element, element:Element)
haloElement.setAttribute('fill', color);
haloElement.setAttribute('stroke', color);
haloElement.setAttribute('stroke-width', width + '');
unigis.util.insertBefore(element, haloElement);
haloElement.removeAttribute('id');
unigis.util.addClass(haloElement, 'halo');
}

private _createTextHalo(element:Element, color:string, width:number)
unigis.util.removeClass(element, 'create-halo');
var clone:Element = <Element>element.cloneNode(true);
this._finalizeHaloElement(clone, element, color, width);
}
```