

TURF

Turf.js

Geoverarbeitung im Browser

Was ist Turf.js?

Turf is a JavaScript library for spatial analysis.

... JavaScript!

Turf: GIS for web maps

By  Morgan Herlocker on December 23 2014 | ● DEVELOPER TOOLS

Turf

business intelligence

Turf is GIS for web maps. It's a fast, compact, and open-source JavaScript library that implements the most common geospatial operations: buffering, contouring, triangular irregular networks (TINs), and more. Turf speaks GeoJSON natively, easily connects to Leaflet, and is now available as a Mapbox.js plugin on our cloud platform. We're also working to integrate Turf into our offline products and next-generation map rendering tools.

<https://www.mapbox.com/blog/turf-gis-for-web-maps/>

Ursprünge

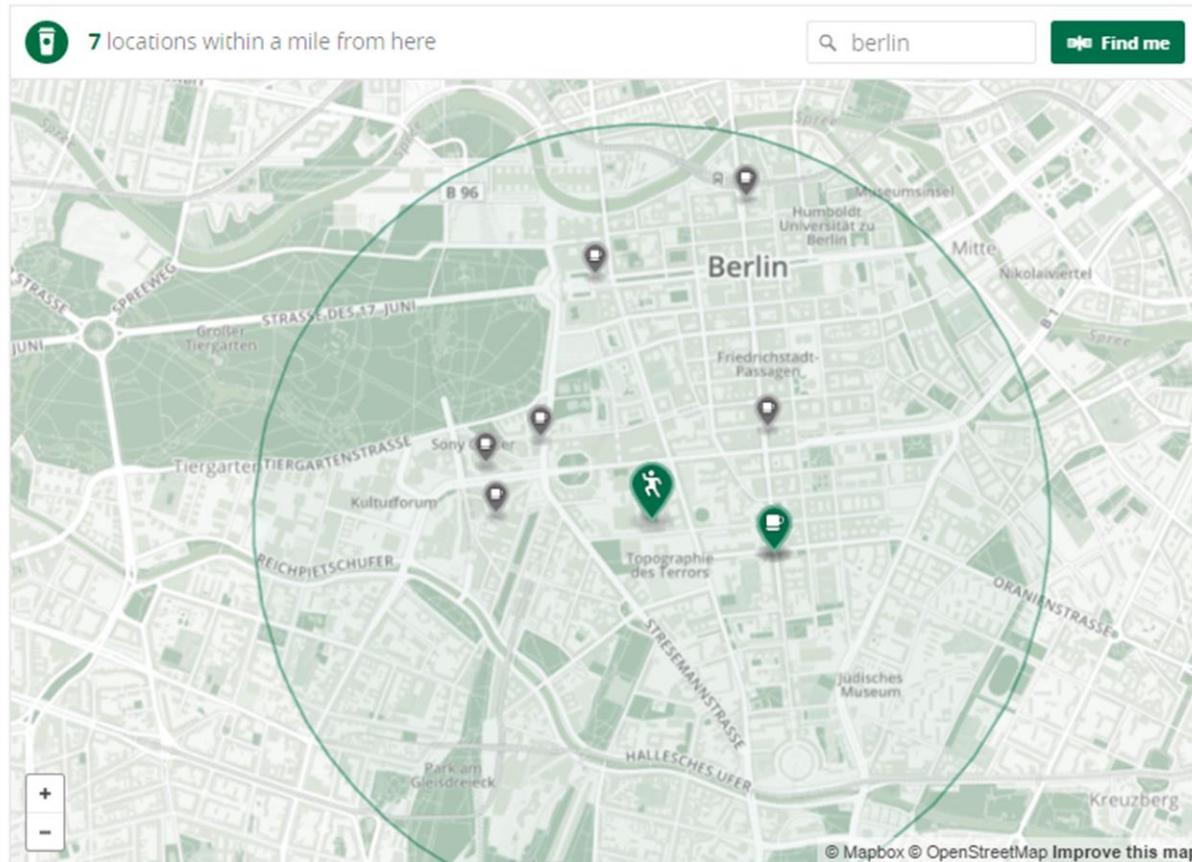
- Von Morgan Herlocker, 2013
 - <http://morganherlocker.com>
 - @morganherlocker
- Mapbox, seit 2014
 - <https://www.mapbox.com>
 - @Mapbox

Find your nearest cup of coffee with Turf

By  Peter Liu on January 14 2015 | [DEVELOPER TOOLS](#)

[Turf](#) [travel](#) [real estate](#) [business intelligence](#) [labs](#)

Turf has opened a wide range of possibilities for apps running entirely in the browser. In this example, we've built a store locator that finds all Starbucks locations near a user's current position. **Drag the little green figure to move it around and mousewheel up and down to adjust the search radius:**





<https://www.mapbox.com/blog/coffee-with-turf/>

GeoJSON in, GeoJSON out

- GeoJSON: das Datenformat in Turf.js
- Alles in einer Datei:
 - Geometrie
 - Attribute
 - Koordinatensystem

GeoJSON



Point

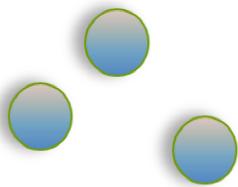


LineString

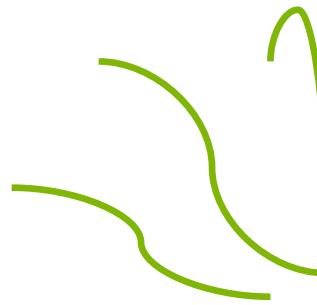


Polygon

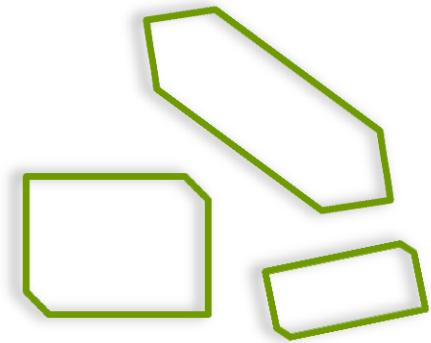
GeoJSON



MultiPoint



MultiLineString



MultiPolygon

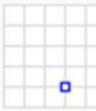
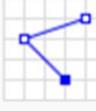
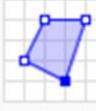
GeoJSON



GeometryCollection

Spec: <http://geojson.org/geojson-spec.html>

Geometry primitives

Type	Examples
Point	 <pre>{ "type": "Point", "coordinates": [30, 10] }</pre>
LineString	 <pre>{ "type": "LineString", "coordinates": [[30, 10], [10, 30], [40, 40]] }</pre>
Polygon	 <pre>{ "type": "Polygon", "coordinates": [[[30, 10], [40, 40], [20, 40], [10, 20], [30, 10]]] }</pre>
	 <pre>{ "type": "Polygon", "coordinates": [[[35, 10], [45, 45], [15, 40], [10, 20], [35, 10]], [[20, 30], [35, 35], [30, 20], [20, 30]]] }</pre>

```
{  
  "type": "FeatureCollection",  
  "features": [  
    {  
      "type": "Feature",  
      "geometry": {  
        "type": "Point",  
        "coordinates": [  
          13.06036,  
          47.78866  
        ]  
      },  
      "properties": {  
        "Ort": "Salzburg",  
        "Event": "FOSSGIS"  
      }  
    },  
    {  
      "type": "Feature",  
      "geometry": {  
        "type": "Point",  
        "coordinates": [  
          7.12352,  
          50.71855  
        ]  
      },  
      "properties": {  
        "Ort": "Bonn",  
        "Event": "FOSS4G"  
      }  
    }  
  ]  
}
```

```
{  
  "type": "FeatureCollection",  
  "features": [  
    {  
      "type": "Feature",  
      "geometry": {  
        "type": "Point",  
        "coordinates": [  
          13.06036,  
          47.78866  
        ]  
      },  
      "properties": {  
        "Ort": "Salzburg",  
        "Event": "FOSSGIS"  
      }  
    },  
    {  
      "type": "Feature",  
      "geometry": {  
        "type": "Point",  
        "coordinates": [  
          7.12352,  
          50.71855  
        ]  
      },  
      "properties": {  
        "Ort": "Bonn",  
        "Event": "FOSS4G"  
      }  
    }  
  ]  
}
```

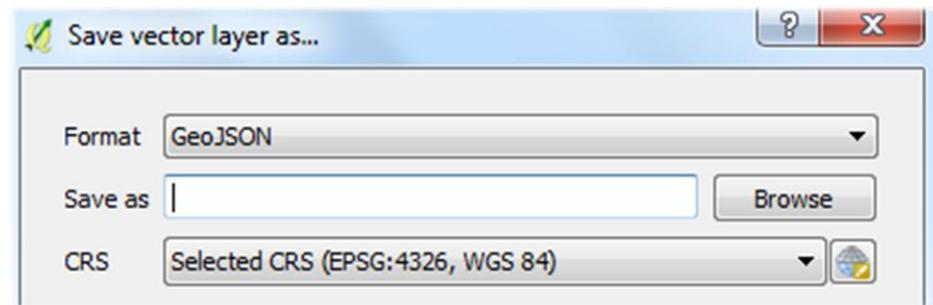
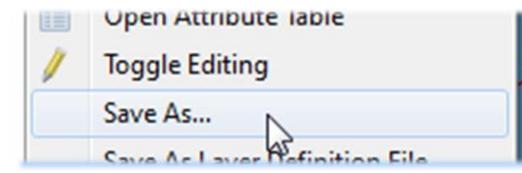
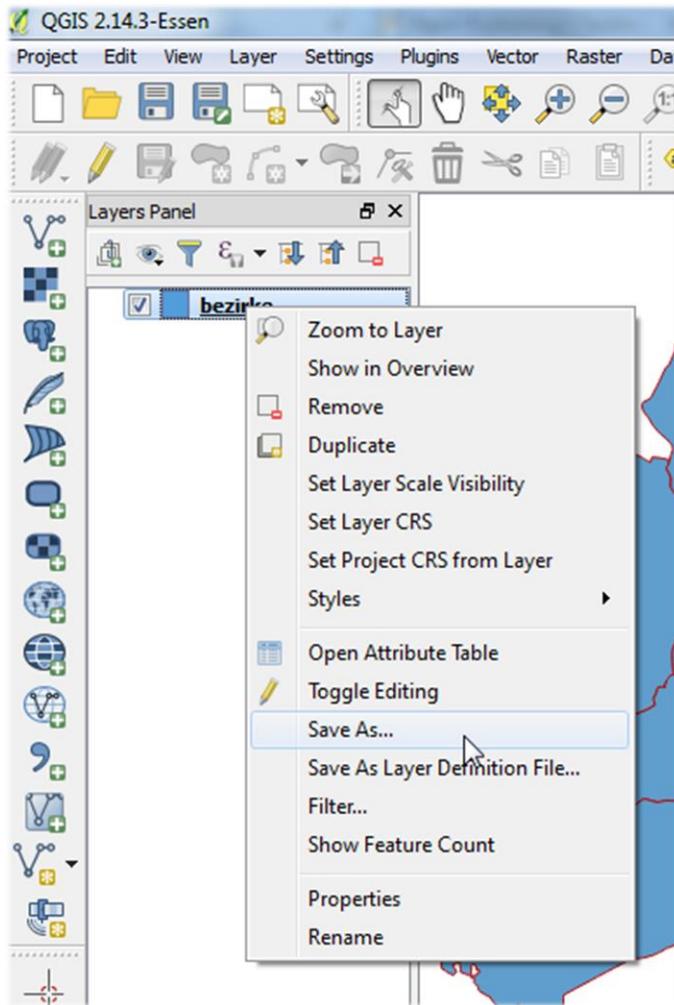
Wie bekomme ich ein GeoJSON?

- Von Hand schreiben ☺
- <http://geojson.io/>
- QGIS

The screenshot shows a map of Europe with various locations marked by dark grey circular pins. The map includes labels for countries like United Kingdom, Ireland, France, Germany, Italy, Spain, Portugal, and others. The interface has a top navigation bar with 'Open', 'Save', 'New', 'Share', 'Meta', and a status 'unsaved'. On the right side, there's a sidebar with a search icon, zoom controls (+, -), a tool icon, and a location pin icon. Below the map, there's a JSON editor window showing a list of features. The JSON code is as follows:

```
9 { "type": "Feature", "properties": { "id": null }, "geometr
10 { "type": "Feature", "properties": { "id": null }, "geometr
11 { "type": "Feature", "properties": { "id": null }, "geometr
12 { "type": "Feature", "properties": { "id": null }, "geometr
13 { "type": "Feature", "properties": { "id": null }, "geometr
14 { "type": "Feature", "properties": { "id": null }, "geometr
15 { "type": "Feature", "properties": { "id": null }, "geometr
16 { "type": "Feature", "properties": { "id": null }, "geometr
17 { "type": "Feature", "properties": { "id": null }, "geometr
18 { "type": "Feature", "properties": { "id": null }, "geometr
19 { "type": "Feature", "properties": { "id": null }, "geometr
20 { "type": "Feature", "properties": { "id": null }, "geometr
21 { "type": "Feature", "properties": { "id": null }, "geometr
22 { "type": "Feature", "properties": { "id": null }, "geometr
23 { "type": "Feature", "properties": { "id": null }, "geometr
24 { "type": "Feature", "properties": { "id": null }, "geometr
25 }
26 }
27 |
```

GeoJSON in QGIS



Geoverarbeitung im Web

Geoverarbeitung im Web: WPS

- Der klassische Fall: **Web Processing Service**



<http://www.opengeospatial.org>

- OGC Standard
- Verlangt:
 - komplexe serverseitige Infrastruktur
 - Requests
 - Internet

Geoverarbeitung im Web: WPS

- User schickt eine Anfrage an den Server
- Der Server bearbeitet die Anfrage in einer DB oder einem GIS
- Der Server schickt das Resultat zurück



Beispiel - WPS: Puffer

... ein paar Minuten später...



Geoverarbeitung im Web: WPS

- User schickt eine Anfrage an den Server
- Server bearbeitet die Anfrage in einer DB oder einem GIS
- Server schickt das Resultat zurück
- (User ist eingeschlafen)
- (User bemerkt einen Fehler, macht das Ganze nochmal)
- (User schläft nochmal ein)
- (User bemerkt schon wieder einen Fehler?)

WPS – ja oder nein?

- WPS macht Sinn, wenn:
 - die Analysen und Berechnungen komplex sind
 - die Daten groß sind
- **Allerdings:** WPS wird öfters benutzt, um sehr simple Werkzeuge oder Berechnungen auszuführen

Die Zeiten sind jetzt vorbei!

TURF

Turf.js

GitHub

Turfjs / turf

Watch 129 Star 2,087 Fork 165

Code Issues 77 Pull requests 6 Wiki Pulse Graphs

A modular geospatial engine written in JavaScript <http://turfjs.org/>

913 commits 10 branches 18 releases 29 contributors

Branch: master New pull request Find file Clone or download ▾

Author	Commit Message	Time
 morganherlocker	Merge branch 'camilleanne-fix-line-slice'	Latest commit <code>1f1f481</code> 9 days ago
 packages	v3.0.11	9 days ago
 .eslintignore	Centralize lint, fix first slew of issues	4 months ago
 .eslintrc.js	Finish eslint sweep	4 months ago
 .gitignore	better tests for turf-point-on-line	26 days ago
 CHANGELOG.md	update changelog	9 days ago
 CONTRIBUTING.md	Start publishing section of contributing	a month ago
 LICENSE	Initial commit	3 years ago
 README.md		

Turf.js Website

Install

node	<code>npm install turf</code>
CDN	<code>//api.tiles.mapbox.com/mapbox.js/plugins/turf/v2.0.0/turf.min.js</code>
Download	turf.min.js

Modular

- Every turf function has been broken into its own separate module, so you can install what you need and nothing else. 05.06.2014, <http://morganherlocker.com/>
- Um die 50 Module

Build your own!

TURF

OUTPUT FILE NAME
turf_MonJun272016
Defaults to turf_dd/MM/YYYY.min.js
No need to include the file extension (eg .min.js)

BUILD

Currently using **Turf 3.0.5**

Available modules

- turf-along
- turf-area
- turf-bbox
- turf-bbox-polygon
- turf-bearing
- turf-bezier
- turf-buffer
- turf-center
- turf-centroid
- turf-collect
- turf-combine
- turf-concave
- turf-convex
- turf-destination
- turf-difference
- turf-distance
- turf-envelope
- turf-isolines
- turf-kinks
- turf-line-distance
- turf-line-slice
- turf-meta
- turf-midpoint
- turf-nearest
- turf-planepoint
- turf-point-grid
- turf-point-on-line
- turf-point-on-surface
- turf-random
- turf-sample
- turf-simplify
- turf-square
- turf-square-grid
- turf-tag

<https://turfjs-builder.herokuapp.com/>

Methoden

center

centroid

distance

destination

envelope

midpoint

bezier

buffer

concave

convex

difference

intersect

simplify

union

combine

explode

flip

kinks

pointOnLine

random

sample

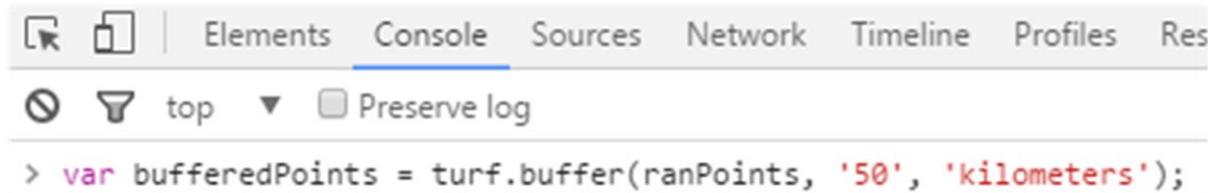
tin

inside

isolines

Mapbox.js?

- Alle offiziellen Beispiele benutzen Mapbox.js
- <https://www.mapbox.com/mapbox.js/api/v2.4.0/>
- **Aber:** Turf.js kann vollkommen unabhängig benutzt werden, sogar ohne Karte



The screenshot shows a browser's developer tools console tab selected. The interface includes icons for selection, copy, and paste, followed by tabs for Elements, Console, Sources, Network, Timeline, Profiles, and Res. Below the tabs, there are filter and log level controls (error, warning, info, top, preserve log). The main area displays a command-line input and output. The input shows a JavaScript variable declaration using the Turf.js library to create buffered points.

```
> var bufferedPoints = turf.buffer(ranPoints, '50', 'kilometers');
```

Mapping = optional

- Turf.js funktioniert auch **ohne** Karte!
- In den meisten Fällen wird eine Karte benutzt, um die Ergebnisse darzustellen aber die Darstellung ist vollkommen optional



- Turf.js kann auch alleine benutzt werden, als pure GIS Bibliothek, ohne eine Darstellungskomponente
- Turf.js funktioniert auch mit Node.js

Kartenclients

- Leaflet
 - OpenLayers
 - Mapbox.js
 - Mapbox GL JS
 - Google Maps JavaScript API
 - usw.
- ... solange GeoJSON unterstützt wird, kann auch Turf.js integriert werden!

Keine API Keys

- Turf.js funktioniert ganz ohne API Schlüssel
- Herunterladen, einbinden und los!

Syntax

- Sehr simpel:

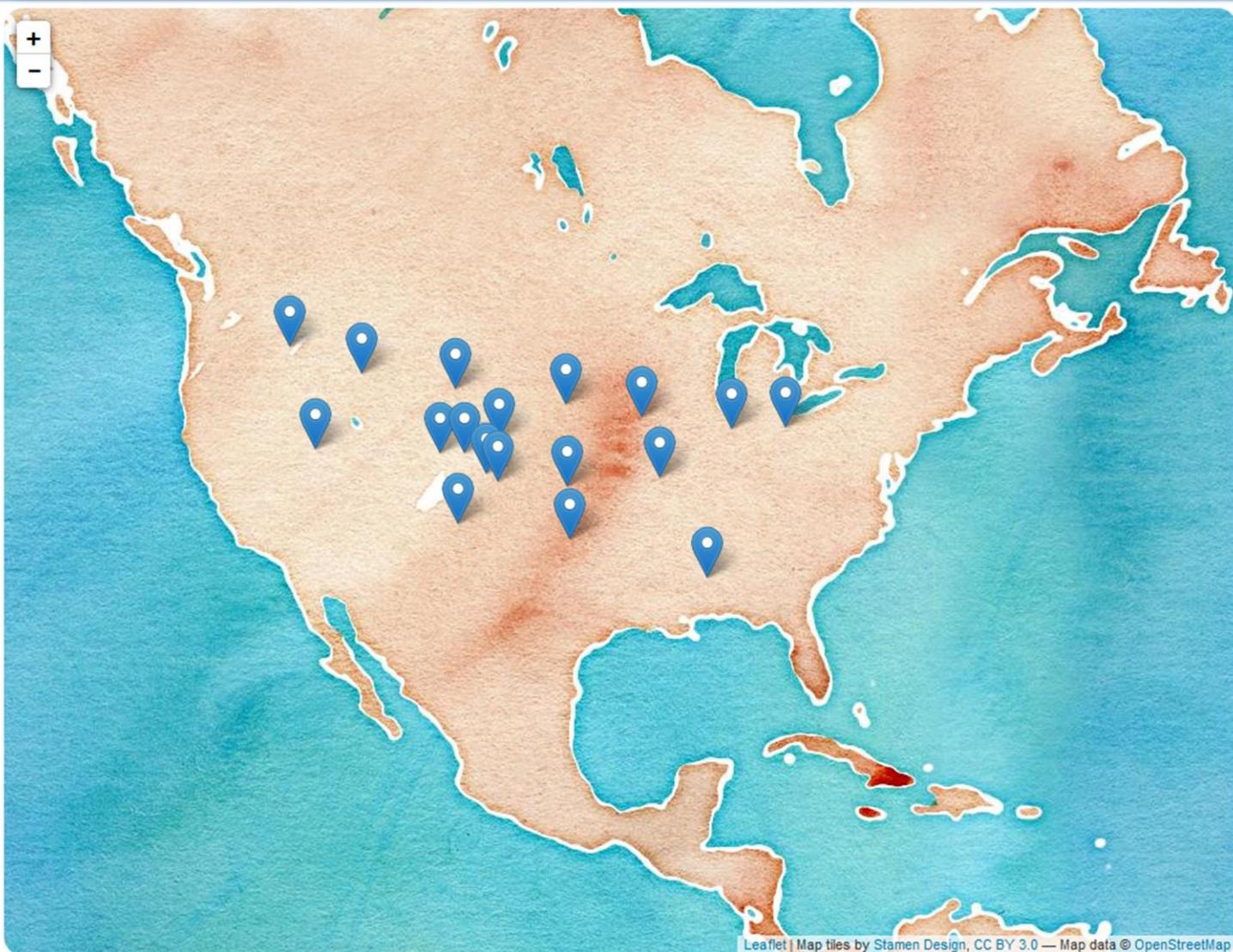
```
turf.werkzeug(params);
```

- Beispiel, Puffer:

```
turf.buffer(features, 45, 'kilometers');
```

DEMO

Point in Polygon



Point in Polygon



Leaflet | Map tiles by Stamen Design, CC BY 3.0 — Map data © OpenStreetMap

Point 18 is in **Nevada**

Point 7 is in **Ohio**

Point 11 is in **Colorado**

Point 12 is in **Colorado**

Point 13 is in **Colorado**

Point 14 is in **Colorado**

Point 15 is in **Colorado**

Point 17 is in **Idaho**

Point 6 is in **Indiana**

Point 4 is in **Iowa**

Point 2 is in **Kansas**

Point 8 is in **Mississippi**

Point 3 is in **Missouri**

Point 5 is in **Nebraska**

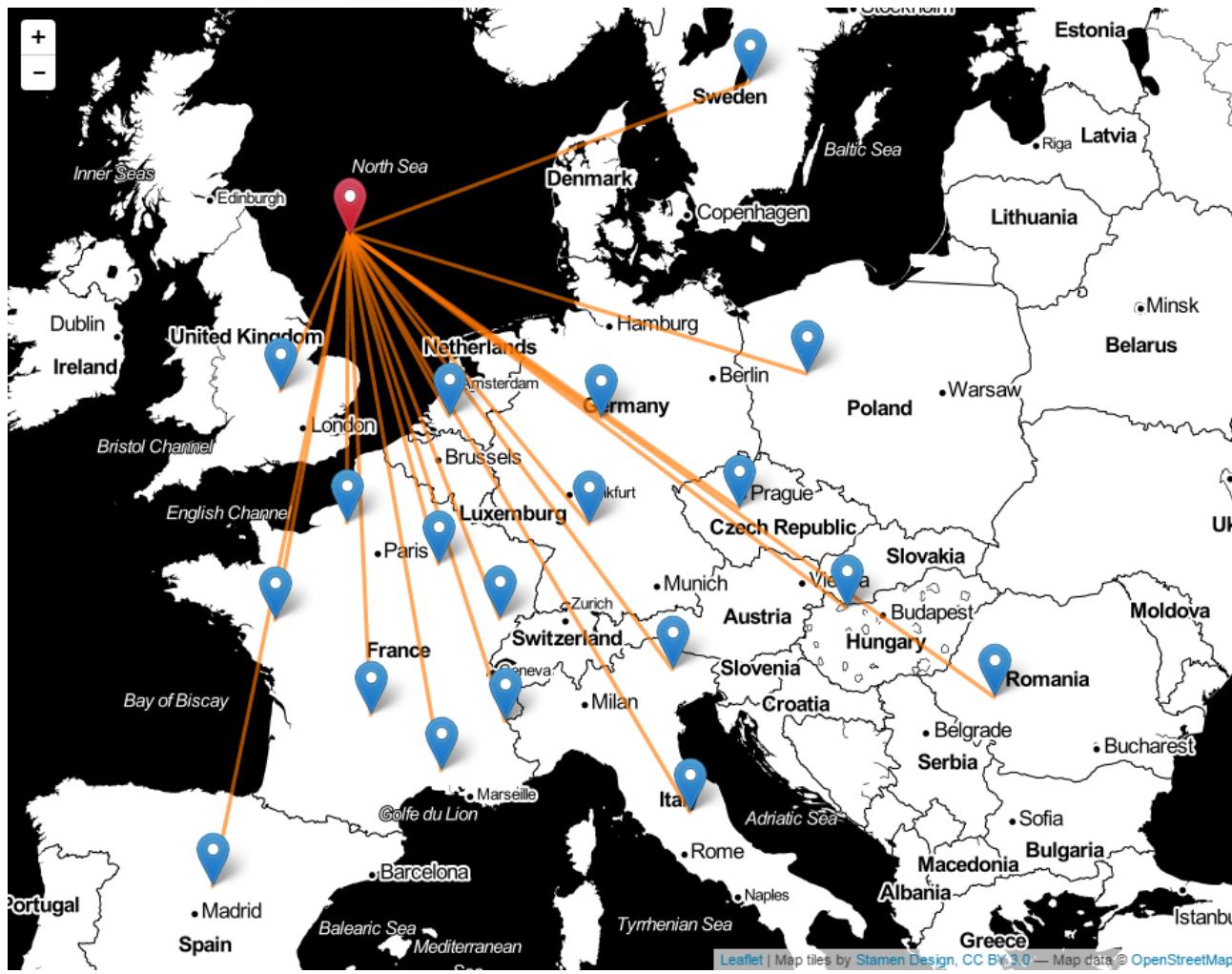
Point 10 is in **New Mexico**

Point 9 is in **Oklahoma**

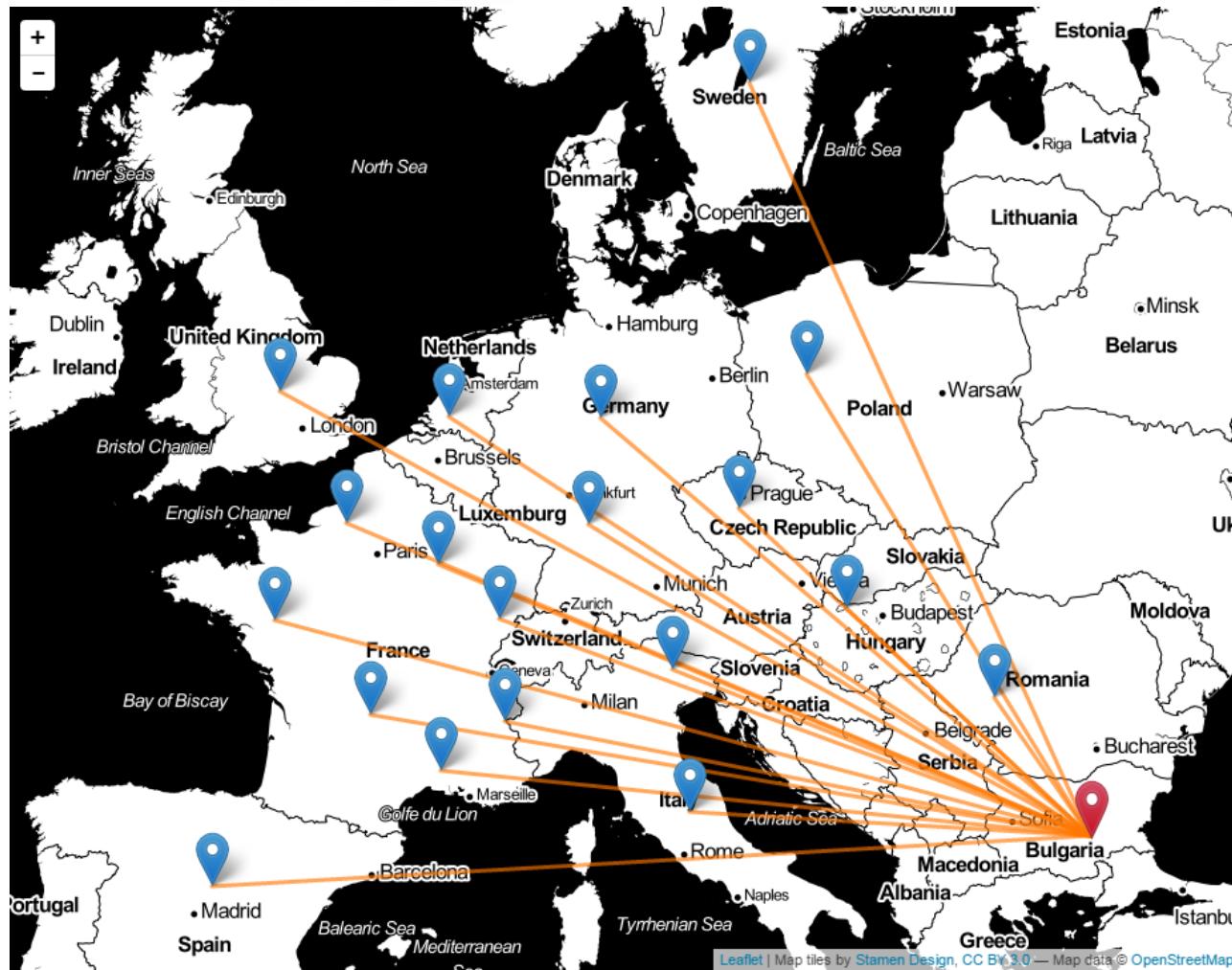
Point 1 is in **Oregon**

Point 16 is in **Wyoming**

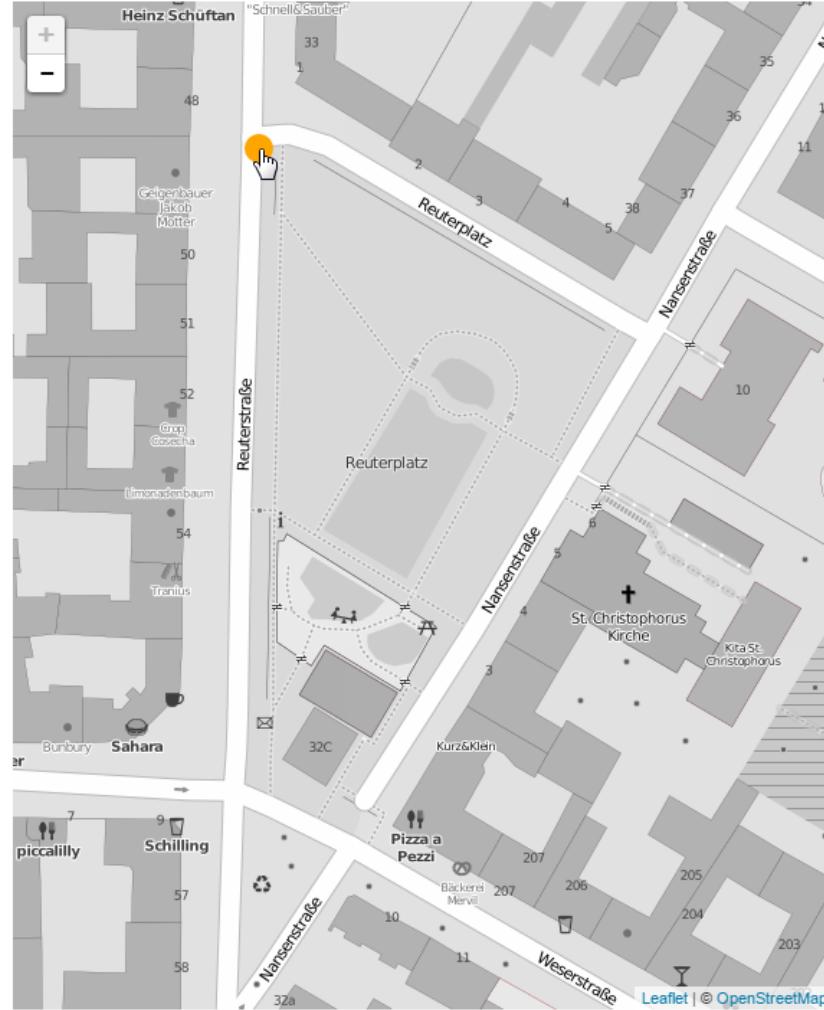
Lines to Points



Lines to Points



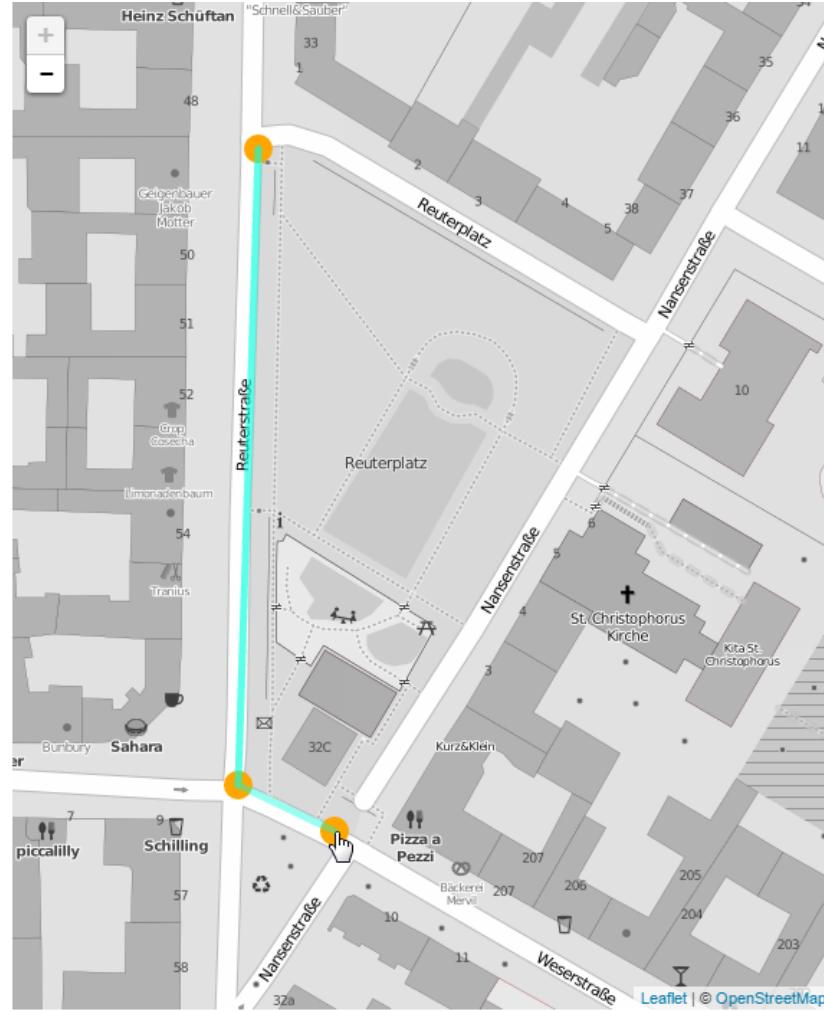
Digitizing



Digitizing



Digitizing



Digitizing



Digitizing



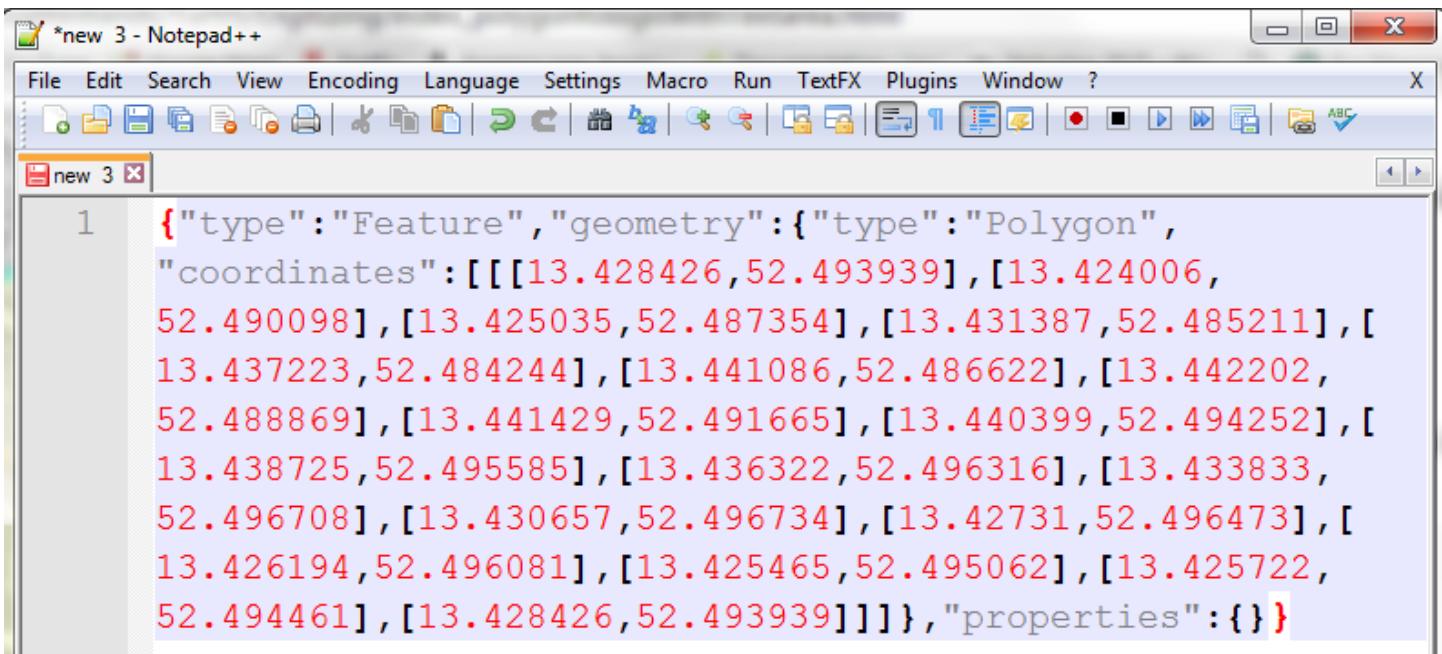
Digitizing





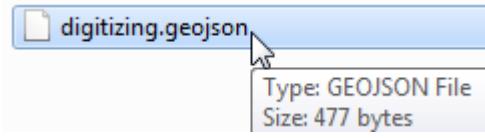
Show Result

```
{"type": "Feature", "geometry": {"type": "Polygon", "coordinates": [[[13.428426, 52.493939], [13.424006, 52.490098], [13.425035, 52.487354], [13.431387, 52.485211], [13.437223, 52.484244], [13.441086, 52.486622], [13.442202, 52.488869], [13.441429, 52.491665], [13.440399, 52.494252], [13.438725, 52.495585], [13.436322, 52.496316], [13.433833, 52.496708], [13.430657, 52.496734], [13.42731, 52.496473], [13.426194, 52.496081], [13.425465, 52.495062], [13.425722, 52.494461], [13.428426, 52.493939]]]}, "properties": {}}
```



The screenshot shows a Notepad++ window titled "new 3 - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Macro, Run, TextFX, Plugins, Window, and ?. The toolbar below the menu has various icons for file operations like Open, Save, Find, and Copy. The main editor area contains a single line of JSON code:

```
1 {"type": "Feature", "geometry": {"type": "Polygon", "coordinates": [[[13.428426, 52.493939], [13.424006, 52.490098], [13.425035, 52.487354], [13.431387, 52.485211], [13.437223, 52.484244], [13.441086, 52.486622], [13.442202, 52.488869], [13.441429, 52.491665], [13.440399, 52.494252], [13.438725, 52.495585], [13.436322, 52.496316], [13.433833, 52.496708], [13.430657, 52.496734], [13.42731, 52.496473], [13.426194, 52.496081], [13.425465, 52.495062], [13.425722, 52.494461], [13.428426, 52.493939]]]}, "properties": {}}
```

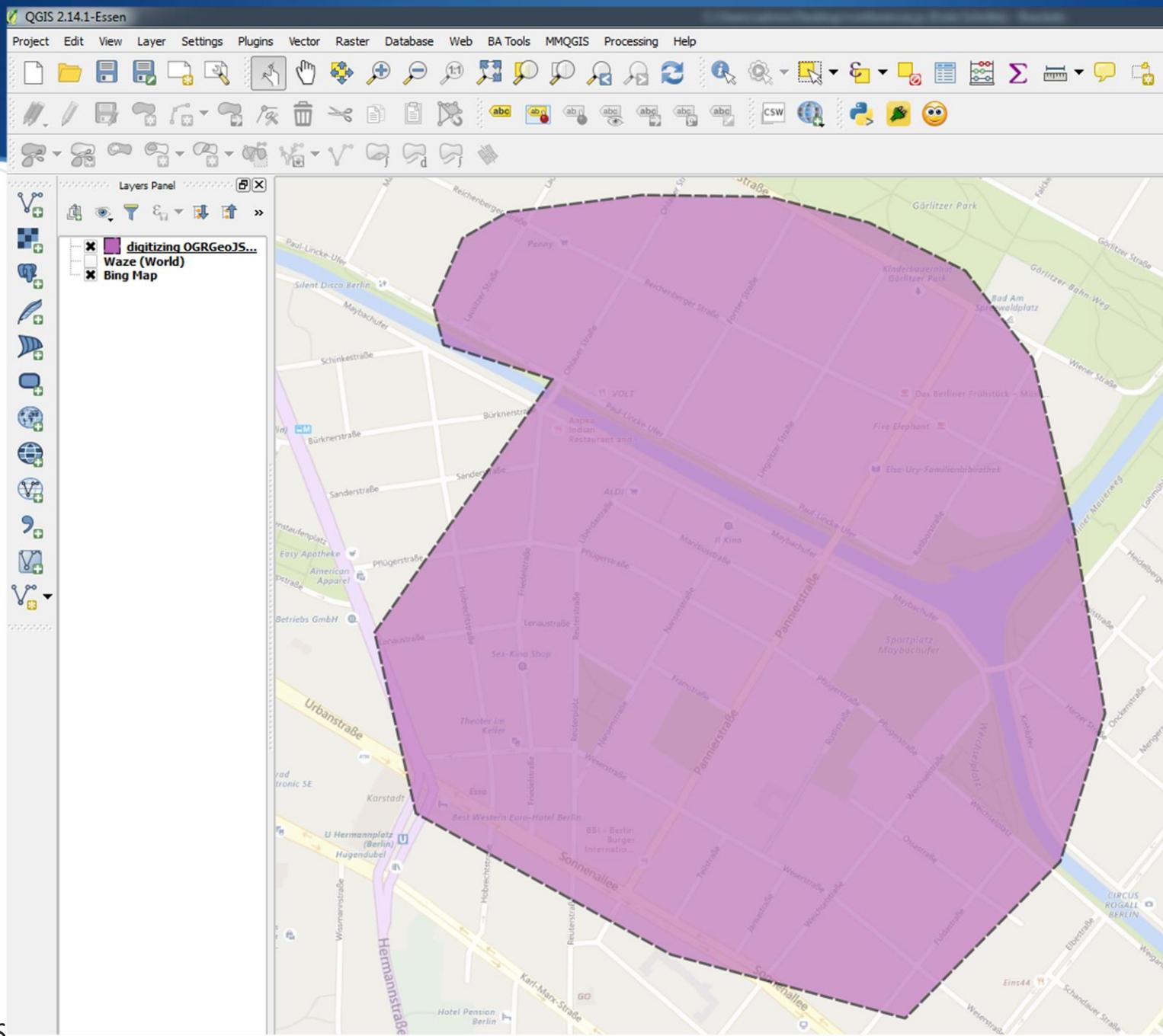


Alle Dateien (*) (*.*)
GDAL/OGR-VSI-Dateihandler (*.zip *.gz *.tar *.t
Arc/Info ASCII Coverage (*.e00 *.E00)
Atlas BNA (*.bna *.BNA)
AutoCAD DXF (*.dxf *.DXF)
ESRI-Personal-GeoDatabase (*.mdb *.MDB)
ESRI-Shapedateien (*.shp *.SHP)
GPS-Austauschformat [GPX] (*.gpx *.GPX)
Generic Mapping Tools [GMT] (*.gmt *.GMT)
GeoJSON (*.geojson *.GEOJSON)
GeoPackage (*.gpkg *.GPKG) ↗
GeoRSS (*.xml *.XML)
Geoconcept (*.gxt *.txt *.GXT *.TXT)
Geography Markup Language [GML] (*.gml *.GML)
INTERLIS 1 (*.itf *.xml *.ili *.ITF *.XML *.ILI)
INTERLIS 2 (*.itf *.xml *.ili *.ITF *.XML *.ILI)
Keyhole Markup Language [KML] (*.kml *.KML)
Komma-separierte Werte [CSV] (*.csv *.CSV)
Mapinfo-Datei (*.mif *.tab *.MIF *.TAB)
Microstation DGN (*.dgn *.DGN)
S-57 Base-Datei (*.000 *.000)
SQLite/Spatialite (*.sqlite *.db *.SQLITE *.DB)
Spatial Data Transfer Standard [SDTS] (*.catd *.SDTS)
Storage and eXchange Format (*.sxf *.SXF)
Systematic Organization of Spatial Information
VRT - Virtuelle Datenquellen (*.vrt *.VRT)
X-Plane/Flightgear (apt.dat nav.dat fix.dat awy)

Alle Dateien (*) (*.*)

Open

Cancel



Dashboard

Input

Input layer
Layer 1

Akureyri
Ísland
Eskifjörður
Höfn
Reykjavík
Reykjavík
Keflavík

Leaflet | © OpenStreetMap contributors | Mapbox basemap

Output

Akureyri
Vopnafjörður
Egilsstaðir
Höfn
Reykjavík
Keflavík

Leaflet | © OpenStreetMap contributors | Mapbox basemap

+ -

Intersect Intersect 2

Erase

Union

Merge

Combine

Buffer distance unit

Simplify

Value:

Flip

Random

points/polylines no. xmin ymin xmax ymax

Sample Number of features

Kinks

Add to map Clear input

Generate output Clear output

Europe Game



Next Round

Countries left: 3

Find

Germany

CORRECT

You get 8 points.

8 seconds left.

Nicht vergessen!

Workshop!

Morgen um 10:30 Uhr

Raum HS432

<http://frab.fossgis-konferenz.de/de/2016/public/events/5083>

Danke, Kontakt, Fragen

```
console.log('Danke! :-)!');
```

- gis.stackexchange.com/users/23263/britishsteel



- numa.gremling@geosysnet.de

- twitter.com/Gremling89

- geosysnet.de

Impressum

geoSYS
Pflügerstr. 56
12047 Berlin, Germany

Telefon: 030/82070657
Telefax: 030/82070658

Email: info@geosysnet.de
Website: www.geosysnet.de