NOTE: THIS IS STILL WORK IN PROGRESS

SOME INFO IS OUTDATED

Everything hereafter describe the vector tiles obtained from https://basemap.at/standard-5/. While the basemap uses mapbox vector tile specifications, all investigations and insights have only been done with the aforementioned map.

Specification

Basemap uses mapbox vector tile specification version 2.1
https://github.com/mapbox/vector-tile-spec/blob/master/2.1/README.md
The .proto file in the linked github can parse the pbf file obtained from basemap.

Url

Basemap .pbf (protobuf) files can be obtained using the following url pattern
https://mapsneu.wien.gv.at/basemapv/bmapv/3857/tile/{ZOOM}/{Y_TILE}/{X_TILE}.pbf

example:
https://mapsneu.wien.gv.at/basemapv/bmapv/3857/tile/13/2878/4384.pbf
(corresponding location on maptiler: https://www.maptiler.com/google-maps-coordinates-tile-bounds-projection/#13/12.65/47.08)

Basemap uses "Google's" Tile Schema, which uses north as the origin of the y_tiles

.pbf to "usable" format

in order to read the pbf file we are using a fork of https://github.com/mapbox/vector-tile/
This repo provides a basic conversion and helpful utilities that fully suits our needs.

Alternatively it is also possible to use googles protobuf library directly and convert the .proto file of the specification repository (https://github.com/mapbox/vector-tile-spec/blob/master/2.1/vector_tile.proto) into a cpp usable format.

```
./protoc --cpp_out=. test.proto
```

in order to get a "rough" overview of the content of a given protobuf file, the following function might also be useful:

```
cat myfile.pbf | ./protoc --decode=vector_tile.Tile vector_tile.proto > output.txt
```

Please note that this output is not grouped together at all and properties that belong to the same feature may be located on completely different locations.

Layout

A vector tile consists of multiple layer. The layer

Map inaccuracies/duplicates

basemap contains (in its current form) some minor inaccuracies.
on one pbf there exists multiple features with the same name and roughly the same location. The location is not exactly the same which results in cases where we cannot use the text and the position as keys for a hash function to disregard duplicate cases.

On the other hand there also exists some corner cases like a mountain peak having the same name 10 km apart. This means that we also simply cannot just use the text as the only target of our hashing function.

We calculated the position inaccuracy of known false positives and found out that the inaccuracy on zoom level 13 was minor:
e.g. Großglockner had the following world coordinates (in alpinemaps world coordinates)

x1: 1413077.500000
x2: 1413074.500000
difference 3.0

y1: -5957698.500000
y2: -5957699.000000
difference 0.5

For comparison, we used two longitude/latitude coordinates that are a measured 100 m apart and obtained the following difference in alpinemaps world coordinates:

long/lat:
p1 = 48.19955937032757, 16.370708425530825
p2 = 48.19905496720844, 16.369604599426534

world coords difference:
x1:1822378.925855

x2:1822256.048495
difference 122.87

y1:6140118.848787
y2:6140034.608121
difference 84.24

We therefore concluded that we could safely dismiss the last 2 digits of our calculated positions for the hashing function and still can reasonably assume to not have dismissed any mountain peaks that were valid

TODO
test if label_class4 is always present
-> if then only use this for the peaks

other vector tile sources

https://wiki.openstreetmap.org/wiki/Vector_tiles

https://wiki.openstreetmap.org/wiki/OpenStreetMap_Americana

https://tile.ourmap.us/inspect.html#10.96/49.4585/11.8994

https://github.com/onthegomap/planetiler

https://github.com/osm-americana

https://www.openstreetmap.org/user/ZeLonewolf/diary/402227

research:

https://www.openstreetmap.org/user/ZeLonewolf/diary/402227

- describes osm americana
- uses planettiler
- uses pmttiles hosted on AWS (avoids the need of a tileserver since tiles are stored in folders? instead of DB) (this avoids the need for costlier amazon EC2 nodes and just uses AWS lambda)
- 70 GB planet file

Essentially he downloads the OSM file with all the data (updated every 9 hours).
converts them to pmtiles(using planettiler) uploads those tiles to server where they are then used like normal vector tiles

they are using a "free" aws lambda server from amazon where up to 1 million requests are "free".
-> using this server for our needs doesn't seem ethical -> we should strive to use our own server

```
osm:
[out:json][timeout:25];
// gather results
nwr["natural"="peak"]({{bbox}});
// print results
out geom;
```

Planetiler

https://github.com/onthegomap/planetiler
By using planetiler we can create our own vector tiles, using only the data we need for our needs. Planetiler is a java project that converts osm (open street map) data to pmtiles (https://github.com/protomaps/PMTiles). Additionally it allows us to specify an "extractor" to easily convert/extract only the features we want in a form we specify.

Overall the process could be described with the following steps:

1. download your desired osm file (could be the planet or a specific region (see where to get osm.pbf file of austria)
2. write your extractor / adapt the existing one
3. create a jar with the build.sh
4. use extract.sh to create a pmtiles file
5. copy the pmtiles file to a server where it can be used e.g. (with `./pmtiles serve .` command)

By using the `serve` command a server is created that interprets urls
like
`http://localhost:8080/austria.peaks/13/4384/2878.mvt`

On linux only(?):
The above url provides a gziped mvt file. In order to properly use this file we have to unzip it first (e.g. using gunzip `gunzip -d 2878.mvt.gz`) (note that pmtiles serves with only the .mvt extension but gunzip expects .mvt.gz -> rename the file to .gz first before executing the gunzip command)
The resulting file should then be a valid mvt file (that can be processed using protobuf)

where to get osm.pbf file of austria

https://download.geofabrik.de/europe.html
https://download.geofabrik.de/europe/austria-latest.osm.pbf (705MB)

Questions

licensing of planetiler
licensing of osm source data (do we need notice on our map?)

OSM data problems:

some peaks do not have elevation or even name attributes
null ele: Pyringer Höhe || name, natural, source,

null ele: null || natural, summit:cross,
null ele: null || natural,
https://www.openstreetmap.org/search?query=pyringer%20h%C3%B6he#map=17/48.83137/15.36481

those peaks do not appear on americana tilemap at all
https://tile.ourmap.us/inspect.html#15.25/48.829442/15.36899