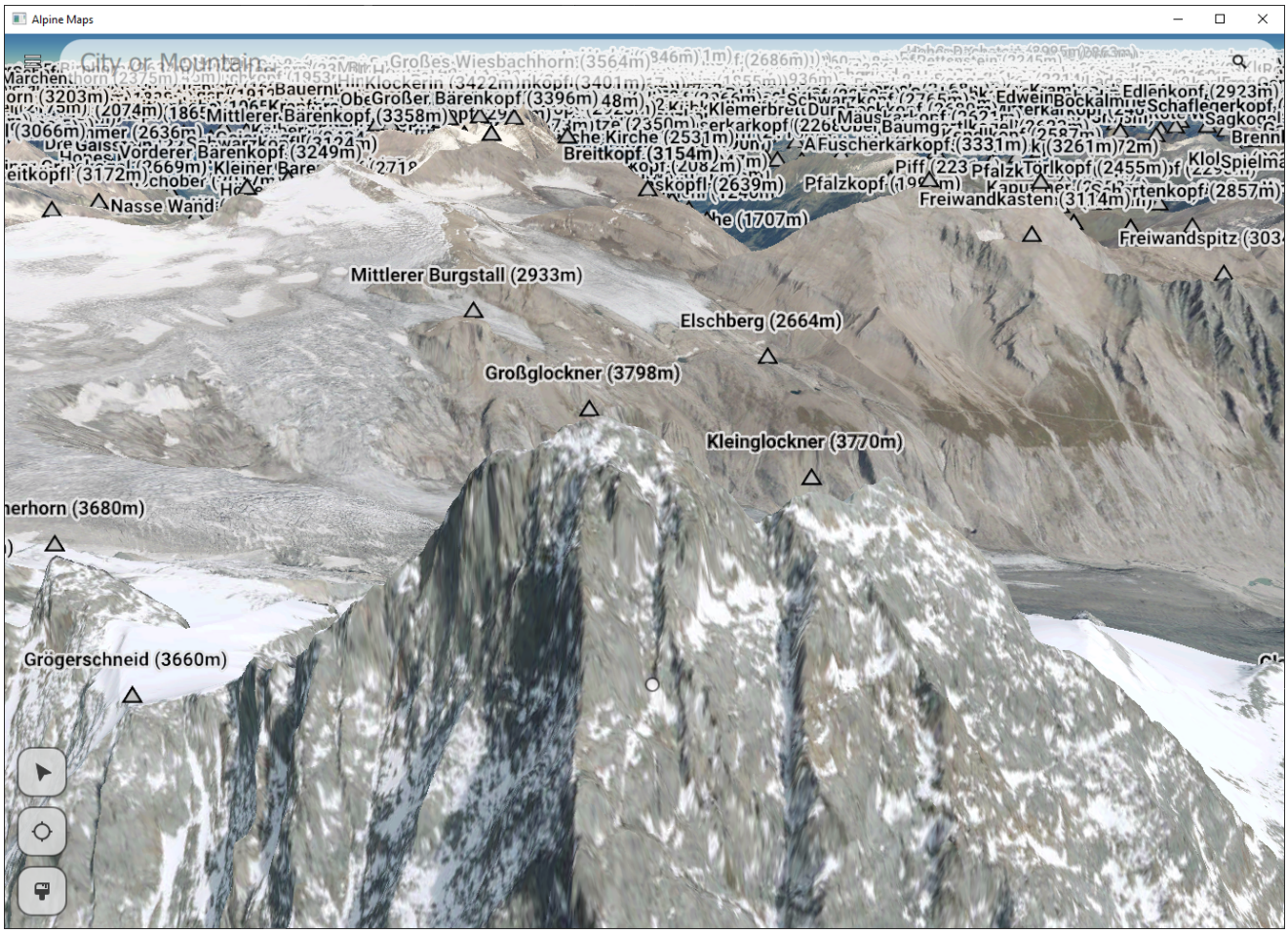
Report - Alpine Maps Labels

Author / Developer: Lucas Dworschak (01225883)
TU Wien



Description

The goal of this project was to receive mountain peak names from a vector tile server, process this information and visualize them on the AlpineMapsOrg application. Initially we agreed on using basemap as the vector tile provider but after encountering some problems (mentioned below), we switched to a custom tile server approach that we developed to fill our needs.

The current tile server is a local server which can be started using the VectorTileService repository. The repository provides multiple scripts which range from downloading the current OSM Data, over extracting relevant features, to providing a network accessible server that provides the final vector tiles in a Mapbox vector tile format (.mvt). Please note that it is currently planned to move away from the VectorTileService program and use a custom server hosted by TU Wien which essentially will do the same thing.

The .mvt file is requested by the AlpineMaps application, parsed using the Mapbox Vector Tile Library and stored into a useable format. Once the relevant vector tiles are in the

viewport, the program automatically processes all visible mountain peaks and creates quads for each character of the label names. The quad and texture positions are furthermore stored in gpu accessible memory, which will be used to render them in a simple shader program in each render pass.

Libraries

STB_Truetype

STB are single file c++ libraries in the public domain. Each header file can be used separately to accomplish a task. In our application we are using the the stb_truetype.h in order to render .ttf font files into a font atlas file. A font atlas here is an image with each (previously selected) character rendered into certain positions. Those positions are then stored by us and can later be used to render a single characters onto quads. This kind of font rendering algorithm is a similar approach as explained in the book "Learn OpenGL" by De Vries[1].

Mapbox Vector Tile Format

Both Basemap and our internal vector tile server are using the Mapbox vector tile format. The format uses protocol buffers (protobuf) for easily parsable byte encoding. While the file extension .mvt is the most prominent we also encountered .pbf and no file extensions that were encoded that way and provided by a vector tile server.

Mapbox Vector Tile Library

We are using a Vector Tile Library provided by Mapbox in order to parse the vector tile file format into easily accessible c++ classes.

Planetiler

Planetiler is the application we are currently using in order to extract features (like the mountain peaks) from OpenStreetMap files. Using Planetiler allows us to specify which features and what meta information (location, altitude, name, …) we want to retain. The application further allows us to export all the features into pmtiles.

PMTiles

By using PMTiles we are able to easily store and serve vector tile data. A PMTiles archive is a single file which stores all the vector tile data in a binary form. The application in the linked repository can be used to act as a tile server that provides individual vector tiles.

Encountered Problems

Basemap

The main issue we encountered with Basemap were inconsistent data behaviours. The main problem we encountered were duplicate data entries with slightly differing meta data (e.g. two different altitudes and different positions of the same peak). We are assuming that this data error stems from the merging of two (or more) data sources.

While we tried to minimize this behaviour by using custom preprocessing steps (e.g. if multiple mountains with same/similar names are around a certain radius only show the one with the highest altitude), we found that this approach was not ideal. We therefore found a way to create our own vector tile server which additionally allowed us to precisely define what data we want to send, allowing us to reduce the bandwidth and parsing cost significantly. Our approach solely uses the OpenStreetMap data. OSM users are able to provide constant feedback about possible map inaccuracies and each feature also contains their own change history to combat possible vandalisms and make it easier to roll back changes.

VectorTileService

Our own VectorTileService (tile processor and tile server) use the Planetiler tool as the means to extract the features we want. Unfortunately after testing, we found out that the maximal zoom level we could process, using Planetiler, was set to 15 (see GitHub issue). While it is reasonable to argue that a zoom level of 15 is good enough for label placement, there is a problem within the AlpineMaps code itself: Currently the shown tile zoom level in the AlpineMaps application is linked with the provided heightmap and orthographic picture tile zoom level. Since the height and the ortho image are provided/processed using different servers they don't encompass the problem with the zoom level. Our current workaround for this problem stores all vector tiles in the AlpineMaps application and uses a previously cached lower zoom level instead.

The application that provides the vector tiles, that were processed in an earlier step, is PMTiles (using the command line `pmtiles.exe serve` command). Unfortunately during the development of the rest of the AlpineMaps vector tile parser code, we encountered small problems with the reliability of this tile server. Sometimes the tile server wouldn't respond at all for a few seconds. We therefore decided that we would switch to a more specialized server for a production environment. Therefore a move to a TU Hosted server is currently planned, which removes the need of the VectorTileService completely.

How to start the program

In order to start the program with the OpenGL rendered vector tile labels you need to download the VectorTileService and the AlpineMapsOrg application and follow their corresponding installation procedures.

VectorTileService

In order to properly use this application you might need to install the latest Java version and Maven.
After downloading the repository execute the scripts in the `scripts/` folder one by one. By doing this an up-to-date version of Austrias OSM file is downloaded, the java code for the extractor is compiled and relevant features extracted and stored in an easily servable way. After this you only need to execute the `4_server` script in order to provide the vector tiles from your localhost.

AlpineMapsOrg

The current state of the vector tile labels is currently only available in the vector_tiles branch. You need an up to date Qt Creator application in order easily run this program. All other dependencies should automatically download themselves while the initial build is executed using the cmake files.

---

1. De Vries, J. (2020). *Learn OpenGL-Graphics Programming: Learn Modern OpenGL Graphics Programming in a Step-by-step Fashion*. Kendall & Welling. Chapter 48.2↩