Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

IKG
Institute of Cartography
and Geoinformation

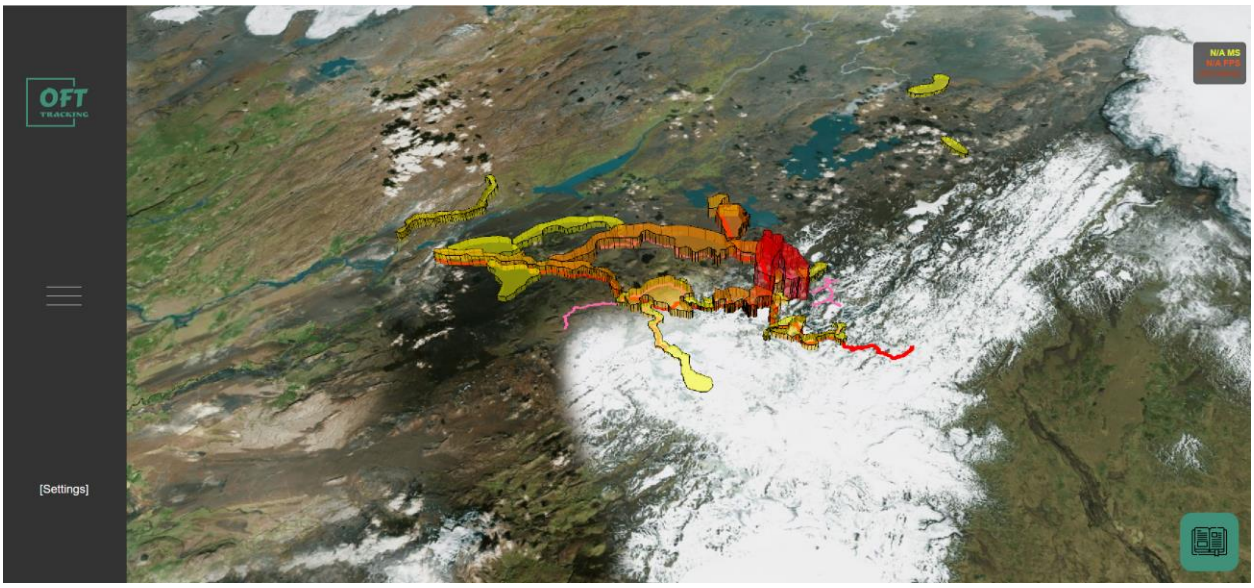
Application Development in Cartography (103-0227-00L)

Autumn semester 2023

Course lead: Prof. Dr. Lorenz Hurni

OfftheRoadTracker

Final report



Code repository Frontend : https://gitlab.ethz.ch/ikgcartoapps-hs23/offtheroadtracker

Code repository Backend : https://github.com/Stimmikex/OfftheRoadTracker-API

Web application: https://ikgcartoapps.ethz.ch/project/off-the-road-tracker/

API: http://185.112.144.101:8000/

Author: Styrmir Óli Þorsteinsson

Study program: M.Sc. Computer Science

sthorsteinss@student.ethz.ch

Reykjavík, 04.01.2024

Introduction

The main purpose of the project is to visualize recorded off-road tracks in the Icelandic highlands. With that, it solves the problem that there is no spotlight or talks about the impact of mass tourism in that area. Showing is better than saying, therefore the project shows the impact of more people and the resulting need for more monitoring via rangers.

The project contains unique data, and this is nowhere to be found other than here and a goal of the project is to make this public and usable for everyone.

The main target group: The users can be government agencies and rangers at the moment but in the future, this could be a platform for more information about protected areas and then it becomes useful for general tourists that come to Iceland.

Thematic content: Tracks and Zone a naturally protected area. Spatial and temporal: Polygons for the zones or the bounding boxes, Points and Lines for the Tracks and the same goes for point of interest and the roads are polylines.

The project tells the store of the need of usable data to see the impact these places, this is recorded but never used and now this data can be used and inserted (future) to that rangers and general public is informed on the tourism impact on the area.

Data

When it comes to the data I'm not using any Which ArcGIS/QGIS/FME functions and all the data I'm getting is from provided excel file from the Environmental agency of Iceland. This excel file that was provided was not in good hands and had to be reformatted and from that refactored to be geoJson and that took a Python script and some time. After that refactoring the geoJson could be stored and used on the Cesium Ion platform.

There is a RESTful API setup with this data now (Zones, areas and Tracks) that is working but not used at the moment with the frontend of the project. (There will be a link)

Design

Maps

The Base map comes from Cesium and that's the real work map. Satellite images work well for rangers to get a idea of where the track is located.

Visualization: I applied the extrusion method as it was the best visualization of the zones and the amount of track within the zones.

Animation: For the animation I have the zones extrusion based on the time of the tracks so the user can go through the days and see the number of tracks per day.

Guidelines: The guidelines on usability and functionality. This is something that was a constant problem in the project as the 3D map that was used was high on computing power and made the project look slow.

User interface

The UI is very minimal and that was by design as the main future user group would be using the project in a app or on mobile webpage. The theme is also designed to work well with the cesium world map so that was a focus with the earthen color theme.

The layout of the side and top menu is to keep the focus on the map and the menu as a complement to the map more than an eye sore to the user. The current state of the functionality has very low benefits as the submission as they mainly just show the extruded zones by date and for all the tracks, the menu is generative so that was a focus to make the website more like a template to ease of use and future implementation.

The project follows the guidelines of keeping the design minimal (google minimal design), flexibility with the template feature and the screen size flexibility. Accessibility and only display of generate based on the user's needs (this is for performance).

Implementation

The code is split into Frontend components [Header, Legend, Timeline, Sidebar and Cesium] and scripts [layers.js, main.js, points.js, roads.js, tracks.js, zones.js. and many more], The Backend is structured the same way but with API routes and dataOut for the SQL calls.

Cesium was used in the project and I'm using the basic Cesium map and from that I create my objects and stick them to the map. But the cesium viewer was the groundwork for all cesium related functions.

Frontend UI implementation, I used the Vue components. [Sidebar, Legend, Header and Timeline] these are all UI elements that are created for the main user interaction with the cesium map.

Implementation of the advanced functionality, they are all implemented and run on the client-side but in the future with more work on the API I want the functionality to be all ran on the server-side. The current version of the advanced functionality Is run as scripts to get data, sort data and extrude data.

When it comes to clean code practices, I went with Eslint and other linters for the sass and I set it to be required at the beginning of the projects, so everything was clean coded from the start.

For the implementation I was set in my ways of how I wanted to conduct this project so there are little changes that I would like to make to the project other than starting with API backend from the beginning and maybe having more options with other frameworks to ease implementation.

Conclusion

In conclusion I'm satisfied with the data and the structure of the project to take in more of this kind of data in the future but further implementation is need to polish the project better.

Challenges: The project was the data and performance and getting that in a usable state. The data was harder as it was not as easy as importing it from as RSS, API or other service as I had to do it and completed the service on the backend for future use.

Limitations: UI and more functionality is something that I found have likes to work on more and that was the plan but the API took most of the time to complete at the end of the projects lifetime.

Learning: Learned a lot about time management and working with new frameworks and components, for Cesium and map frameworks this was the first project ever done and the same goes for Vue.

Appendix

Contents

Requirement	Compliance
Topographic raster layer	No maps, Future implementation
Topographic vector layer	Zones extrusion.
Thematic univariate visualization I	Points of interest,
Thematic univariate visualization II	
Thematic multivariate visualization I	Tracks, Roads and Extrusion for Zones
Advanced animation	Timeline player for the zones.
Cartographic UI element I	Legend.
Cartographic UI element II	Timeline
General UI element I	Layer switch, Data toggle, Timeline player.
General UI element II	
Advanced functionality	API routing and Timeline player
Imprint	Styrmir Þorsteinsson, 103-0227-00L, OfftheRoadTracker

Table 1: Record of fulfilled requirements.

Data

Description	URL
Environmental Agency of Iceland	This is not Provided
OpenStreetMap	https://www.openstreetmap.org/#map=11/64.0176/-19.0176
uMap	https://umap.openstreetmap.fr/en/map/fjallabak_988006#15/63.9973/-19.0569

Table 2: Data sources used in this project.

Software

Name	Description	URL	License
CesiumJS	Virtual globe, 3D maps	https://github.com/CesiumGS/cesium	Apache 2.0
Vue.js	User interface	https://github.com/vuejs/vue	MIT
Express	API, Backend	https://expressjs.com/	MIT
Postgres	Database	https://www.postgresql.org/	MiT

Table 3: Web frameworks and libraries used in this project.

Maps

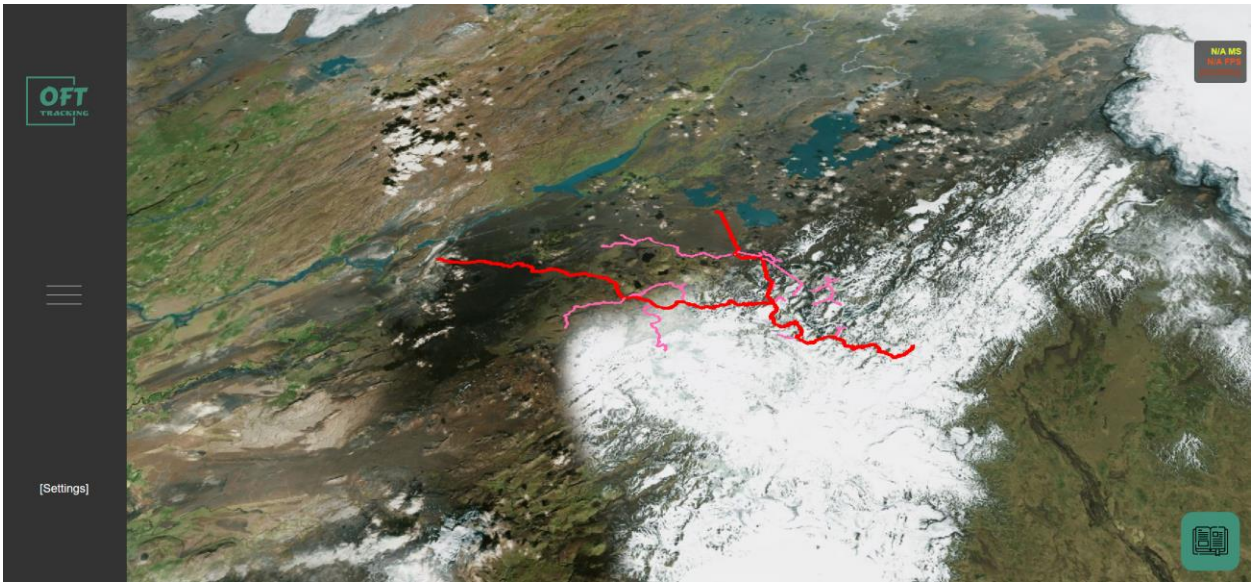


Figure 1: Image of the main page with the cesium vector layer

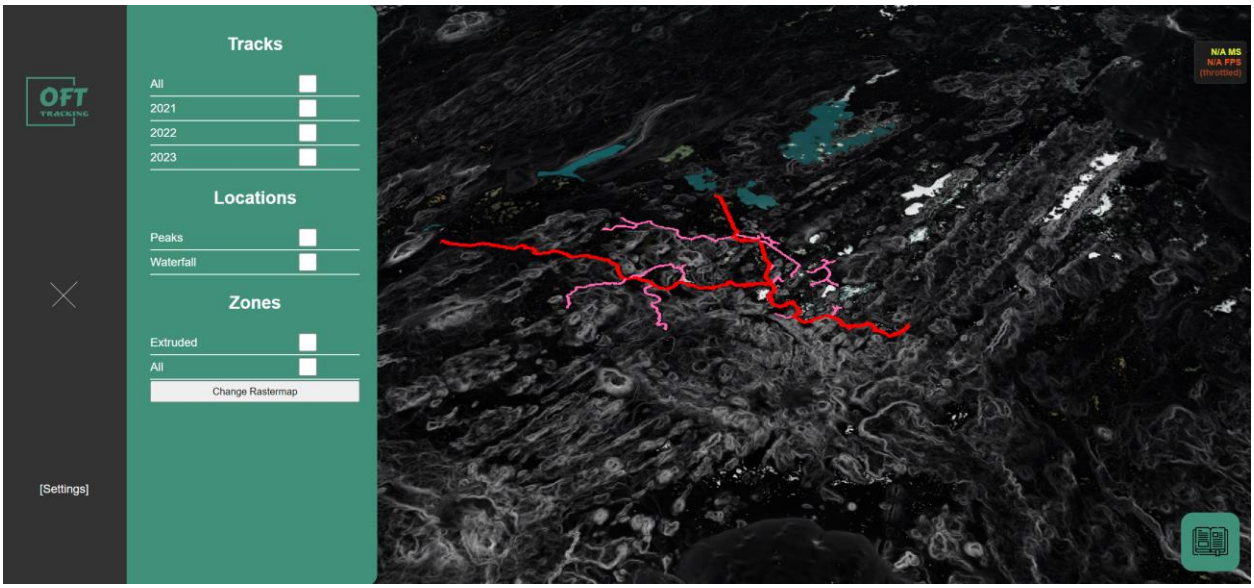


Figure 2: Image of the raster layer (Only one found)

User interface

Figure 4: overall layout

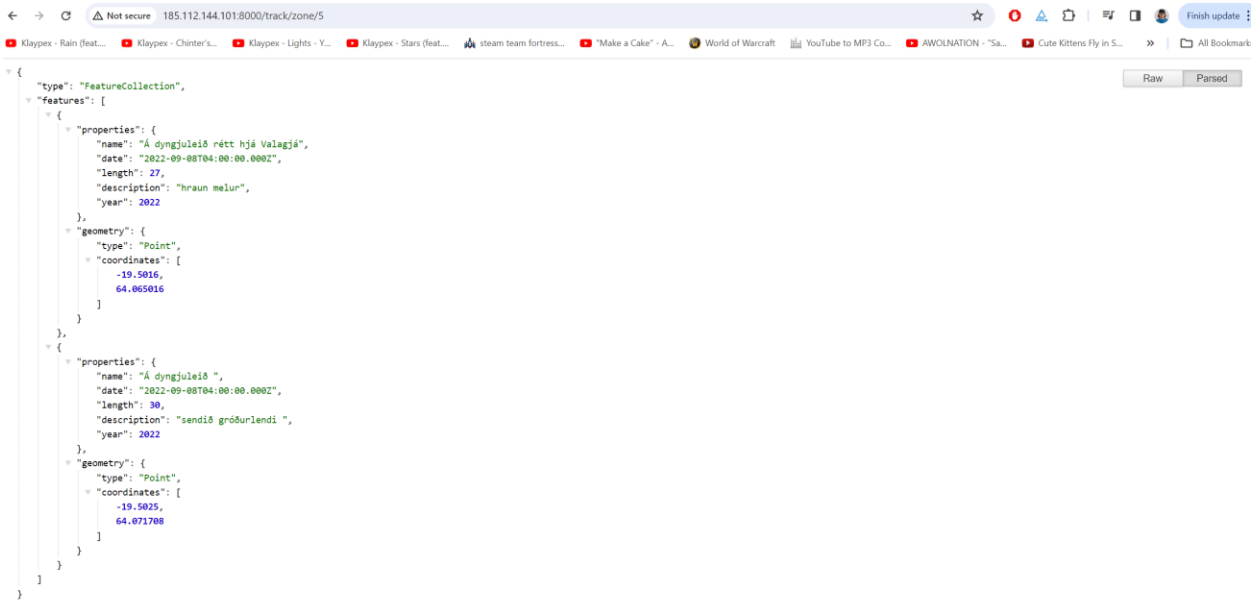


Figure 6: API setup

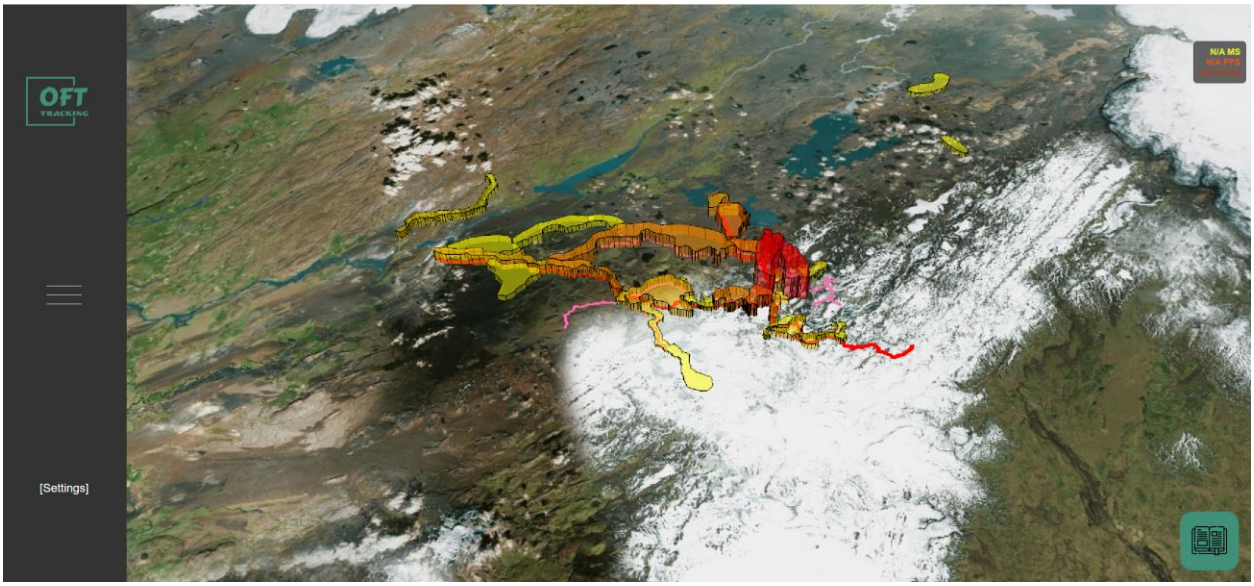


Figure 7: Zones extruded.