



Institut für Geographie

Geoinformatik: Web mapping
LV-Nr.: 716409

Leitung:
Förster Klaus, BSc
Mag. Öggl Bernhard

Abschlussbericht Web Mapping Themenwanderwege des Nationalparks Hohe Tauern

Vorgelegt von:
Hofmann Janina (01423056)
Köck Johannes (01216907)
Merlot Roman (11745595)

SS 2019

Abgabedatum:
18.06.2019

Inhaltsverzeichnis

<u>ABBILDUNGSVERZEICHNIS.....</u>	<u>2</u>
<u>1. EINLEITUNG.....</u>	<u>3</u>
<u>2. DATENGRUNDLAGE</u>	<u>4</u>
<u>3. DARSTELLUNG UND UMSETZUNG DER INTERNETSEITE</u>	<u>5</u>
<u>3.1. HEAD DER HTML-SEITEN</u>	<u>6</u>
<u>3.2. HEADER.....</u>	<u>6</u>
<u>3.2. KARTEN-APPLIKATIONEN</u>	<u>7</u>
<u>3.3. UMSETZUNG DER KARTEN-APPLIKATIONEN</u>	<u>11</u>
<u>4. PROBLEME UND HERAUSFORDERUNGEN</u>	<u>17</u>
<u>5. FAZIT</u>	<u>18</u>
<u>7. QUELLEN.....</u>	<u>18</u>

Abbildungsverzeichnis

Abbildung 1 : Erstellte GPX Daten der Themenwege	5
Abbildung 2 : Header des Krumpern Projektes	6
Abbildung 3 : HTML-Code der Ungeordneten Liste des Navigationsmenüs	7
Abbildung 4 : Kartenplugin mit der Suchfunktion, der Auswahl der Kartenhintergründe und den Overlays auf der Startseite.	8
Abbildung 5 : Zuschaltbare Minimap	8
Abbildung 6 : Suchfunktion am Beispiel der Gehzeit.	9
Abbildung 7 : Popup mit kurzer Wegbeschreibung und Link zum Weg.	9
Abbildung 8 : Routenstatistik des ausgewählten Themenweges.	9
Abbildung 9 : Zeigt das Höhenprofil des ausgewählten Themenweges.	10
Abbildung 10 : Farbliche Codierung der Steigung.	10
Abbildung 11 : Popups des Themenweges (oben) Popup eines POI (unten).	11
Abbildung 12 : Verlinkung der Leaflet bibliothek im Header der HTML Dateien.	11
Abbildung 13 : Einbinden der verschiedenen Kartenhintergründe.	12
Abbildung 14 : Verlinkung des Vollbildmodus der leaflet Karten.	12
Abbildung 15 : Hinzufügen der "Vollbildsteuerung".	12
Abbildung 16 : Verlinkung des Markercluster Plugins	12
Abbildung 17 : Erstellen der Markercluster für die POI-Popups	13
Abbildung 18 : Verlinkung des Minimap Plugins im HEADER.	13
Abbildung 19 : Hinzufügen der Minimap zur Karte.	13
Abbildung 20 : Verlinkung des Such Plugins.	14
Abbildung 21 : Abbildung 21: Erstellen und definieren des Suchfeldes.	14
Abbildung 22 : Abbildung 22: Styling des Suchfeldes in der CSS-Datei.	14
Abbildung 23 : Abbildung 23: Verlinkung des GPX-Track Plugins und setzen sogenannter Anker in dem HTML-Datei.	15
Abbildung 24 : Abbildung 24: Funktion welche aus den GPX-Daten die Routenstatistik ausliest.	15
Abbildung 25 : Verlinkung des Höhenprofil Plugins in der HTML-Seite.	15
Abbildung 26 : Erstellen und Hinzufügen des Höhenprofils	16
Abbildung 27 : Verlinkung des d3- Plugins.	16
Abbildung 28 : Verlinkung zu den Themenwegseiten	17

1. Einleitung

Der Nationalpark Hohe Tauern bietet seinen Besuchern eine vielseitige Auswahl von Themenwegen an. Von leichte bis schwere, lange bis kurze, Rundweg bis Zielweg - die Entscheidung welchen Weg man gehen möchte fällt häufig sehr schwer und somit war unser Ziel die einzelnen Themenwege zu beschreiben und durch eine Auswahlfunktion bei einer Übersichtskarte die Entscheidung, welchen Weg man gehen möchte zu erleichtern. Im Rahmen der Lehrveranstaltung Geoinformatik: Web Mapping des Masterstudiums Geographie der Universität Innsbruck werden die Grundlagen der modernen Web Mapping Methoden erlernt. Als Abschlussprojekt sollen die Studenten die gelernten Methoden anwenden können und anhand einer eigenen Webseite mit verschiedenen Kartenapplikationen das Erlernte anwenden. Ziel der Lehrveranstaltung Geoinformatik: Web mapping ist es eine mit den im Rahmen der Lehrveranstaltung erlernten Techniken eine Kartenapplikation mit HTML, CSS und Javascript zu einem selbst gewählten Projekt zu erstellen.

Das Projekt "Themenwege im Nationalpark Hohe Tauern" wurde von dem Kollektiv "Krumpern" bestehend aus Janina Hofmann, Johannes Köck und Roman Merlot bearbeitet und beinhaltet zum Teil erlernte Methoden und Anwendungen, welche noch weiter ausgearbeitet wurden und für die Web-Karten von Leaflet angewandt wurden.

Nachvollziehen kann man die Arbeitsschritte anhand des dafür erstellen Repositories in Github: <https://github.com/krumpern/krumpern.github.io>

Für die gemeinsame Bearbeitung des Projekts wurde eine Organisation erstellt, in welcher die drei Bearbeiter kollaborieren konnten.

Der nachfolgende Bericht soll die geleistete Arbeit darstellen und Arbeitsschritte aufzeigen, dies dient in erster Linie dem Zweck einen Überblick über die eigene Arbeit und eine Grundlage für zukünftige Arbeiten im Bereich des Web Mappings zu schaffen. Zunächst geben wir einen Einblick in die verwendeten Daten und die Bearbeitung dieser. Außerdem wird der Aufbau der Homepage dargestellt und kurz auf die verwendeten Programmiersprachen eingegangen. Das Ergebnis wird präsentiert und aufgekommene Probleme und Verbesserungsvorschläge erläutert. Ein kurzes Fazit zeigt die erlernten Fähigkeiten und zukünftige Anwendung im eigenen Bereich auf.

2. Datengrundlage

Die verwendeten Daten und Bilder sind aus Open Source Quellen bezogen, was eine Voraussetzung der Lehrveranstaltung war. Als Ideensammlung wurde die Internetseite <https://www.data.gv.at/> herangezogen, von welcher auch die Datengrundlagen vom Nationalpark Hohe Tauern stammen. Es wurden folgende Datenquellen verwendet:

- Themenwege Nationalpark Hohe Tauern
(OPEN DATA ÖSTERREICH(2017B))
- Points of Interest (OPEN DATA ÖSTERREICH(2017A))
- Außengrenze (OPEN DATA ÖSTERREICH(2018))

Gegenläufig zum Konzept wurde die Zonierung des Nationalparks nicht verwendet, da dies eine für Themenwege eher irrelevante Zusatzinformation ist, welche die Lesbarkeit und Übersichtlichkeit eingeschränkt hätte.

Die Daten liegen als .JSON Datei vor und wurden wie folgt bearbeitet:

Die Daten der Points of Interest liegen in der Projektion EPSG:31258 also dem MGI/ Austria GKM 31 vor. Jedoch bezieht sich die Karte von Leaflet auf die Projektion EPSG: 4326 also WGS 84 und somit war eine Umwandlung nötig. Diese wurde mit QGIS durchgeführt und wieder als .JSON Datei abgespeichert. Im nächsten Schritt wurde eine .js Datei erstellt (POI_WGS84.js) und eine Konstante mit dem Namen POI erstellt, auf welche für den Aufruf der Daten verwiesen wurde. Die Dateiorganisation inklusive der .js Dateinamen und der Namen der Konstanten sind in Tabelle 1 zu erkennen.

Die Daten der Außengrenze lag schon in der richtigen Projektion (WGS 84) vor und wurde somit nur in der Border.js Datei als Konstante definiert: const Border.

Die Daten der Themenwege musste wie die Points of Interest mittels QGIS in die richtige Projektion umgewandelt werden (NPHTT_Themenwege2017.json) und wurden im nächsten Schritt als wege.js gespeichert. Diese Datei ist die Grundlage für die Übersichtskarte und die Suchfunktion, was in den nachfolgenden Kapiteln genauer dargestellt wird. Fehlende Daten wurden ergänzt und zusätzlich für die zu durchsuchenden Kategorien (Gehzeit, Schwierigkeit, Jahreszeit) jeweils die Namen des Themenweges hinter die eigentliche Information geschrieben. Auf die Gründe dafür wird bei der Beschreibung der Suche genauer eingegangen. Diese Datei wurde kopiert und eine .js Datei mit dem Namen wegePopup.js erstellt. Diese beinhaltet die gleichen Informationen wie wege.js nur ohne den Zusatz der Namen bei den Kategorien Gehzeit, Schwierigkeit und Jahreszeit, wodurch die Popups für die einzelnen Themenwege schöner dargestellt werden können, ohne gedoppelte Informationen über den Namen zu enthalten.

Für die Höhenprofile, welche nachfolgend genauer erläutert werden, mussten .gpx Files erstellt werden. Hierfür wurde das JSON File mit Projektion WGS 84 mithilfe der Website <https://mygeodata.cloud/converter/json-to-gpx> in ein gpx umgewandelt (NPHTT_Themenwege2017.gpx). Dieser Datei wurden dann mit Hilfe der Internetseite <https://www.gpsvisualizer.com/elevation> Höhenwerte für jeden Punkt zugeordnet. Von dieser Datei ausgehend wurden jeweils die einzelnen Themenwege selektiert und eine neue Datei mit dem Namen der Themenwege erstellt, wie in Abbildung 1 zu sehen.

Tabelle 1: Datenorganisation, inklusive der zugewiesenen Konstanten.

Datei	Dateiname .js	Konstante
Points of Interest	POI_WGS84.js	POI
Außengrenze	Border.js	Border
Themenwege	wege.js	wege
	wegePopup.js	wegePopup

- ≡ 1_Natur-und_Kulturlehrweg_Debanttal.gpx
- ≡ 2_Alpenblumenweg_Oberseite_kurz.gpx
- ≡ 3_Alpenblumenweg_Oberseite_lang.gpx
- ≡ 4_Wasserschaupfad_Umbalfälle.gpx
- ≡ 5_Nussdorfer_Berg_Erlebnisweg.gpx
- ≡ 6_Wasserlehrweg_St_Jakob.gpx
- ≡ 7_Geschichtenweg_Dorfertal.gpx
- ≡ 8_Kinderlehrweg_Kals.gpx
- ≡ 9_Waldlehrpfad_Zedlacher_Paradies.gpx
- ≡ 10_Glorerhütte_Geolehrpfad.gpx
- ≡ 11_Gletscherlehrweg_Innergschloess.gpx
- ≡ 12_Natur-und_Kulturlehrweg_Oberhauser...
- ≡ 13_Wassererlebnisweg_St_Jakob_kleine_R...
- ≡ 14_Glocknerstrasse_BergeDenken.gpx

Abbildung 1: Erstellte GPX Daten der Themenwege

Die Bilder für die Webseite wurden von der Seite PIXABAY (<https://pixabay.com/de/>) verwendet, da diese eine große Auswahl an Bildern frei zur Verfügung stellt. Auch die Kartenhintergründe sind frei zugänglich und können mittels URL Codes in die Leaflet Karten eingebunden werden.

3. Darstellung und Umsetzung der Internetseite

Dieses Kapitel beschreibt den Aufbau der Internetseite mit den jeweiligen Kartenapplikationen und interaktiven Elementen. Die gesamt Internetseite besteht aus einer Startseite, welche einen Überblick über die Themenwege, den Nationalpark Hohe Tauern und den Points of Interest in einer Karte gibt. Auf dieser Seite kann man die Wege, anhand von vier Kriterien auswählen. Außerdem wird in der Seite "Über Uns" die Gruppe und die Links zu den Github-Repositories der Mitglieder dargestellt. Die Seite Wege wird untergliedert in eine allgemeine Wege Seite, auf der die Links zu den jeweiligen Themenwege aufgelistet sind und die Themenwege, auf diesen Seiten stehen Beschreibungen zu den jeweiligen Wegen mit einer

Routenstatistik und es wird eine Karte angezeigt, welche nur den einen Weg zeigt. Hier ist die Steigung des Weges erkennbar und Popups zeigen die wichtigsten Informationen zum Weg. Ein Bild gibt einen ersten visuellen Eindruck über die Route.

3.1. Head der HTML-Seiten

Der Head in den HTML-Seiten gibt an, welche Seiten, Plugins, Stylsheets und Daten verlinkt werden sollen. Außerdem sind hier die Metainformationen über die Webseite, wie der Titel und das charset enthalten.

3.2. Header

Der Header ist ein Element, welches für alle Unterseiten der Webseite von der Gestaltung her gleich bleibt und sich als Identifikationsmerkmal durchzieht. Lediglich die Titel und Untertitel der Seiten passen sich an. Der Header besteht, wie in Abbildung 2 erkennbar, aus einem Hintergrundbild, aus dem Nationalpark Hohe Tauern, einer Überschrift und Unterüberschrift und einer Navigationsleiste. In dieser Navigationsleiste können die einzelnen Themenwege ausgewählt werden. Die Gestaltung des Headers, bzw der gesamten Homepage orientiert sich an den Farben im Header und symbolisiert durch das Blau den Himmel und das Wasser im Nationalpark und das Grüne beschreibt die Vegetation.

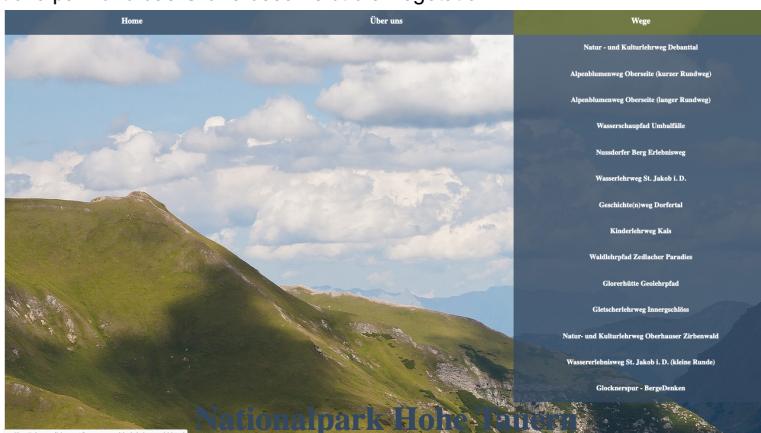


Abbildung 2: Header des Krumpen Projektes

Der Header setzt sich aus dem Body und der Navigationsleiste zusammen, welche im HTML als ungeordnete Liste gegliedert ist siehe Abbildung 3 und im CSS gestylt wird. So wird erreicht, dass beim Überfahren des Kästchens Wege die einzelnen Themenwege erscheinen und sich die Kästchen farblich absetzen, wenn die Maus darüber fährt.

```

<ul>
  <li class="topmenu"><a href="index.html">Home</a></li>
  <li class="topmenu"><a href="ueberuns.html">Über uns</a></li>
  <li class="topmenu"><a href="auswahl.html">Wege</a>
    <ul>
      <li class="submenu"><a href="kulturlehrwegDebanttal.html">Natur - und Kulturlehrweg Debanttal</a></li>
      <!--Verlinkung der Minimap -->
      <li class="submenu"><a href="alpenblumenweg.html"> Alpenblumenweg Oberseite (kurzer Rundweg)</a></li>
      <li class="submenu"><a href="alpenblumenweg_oberseite.html">Alpenblumenweg Oberseite (langer Rundweg)</a></li>
      <li class="submenu"><a href="wasserschaupfad.html">Wasserschaupfad Umbalfälle</a></li>
      <li class="submenu"><a href="bergerlehrweg.html">Nussdorfer Berg Erlebnisweg</a></li>
      <li class="submenu"><a href="wasserlehrweg.html">Wasserlehrweg St. Jakob i. D.</a></li>
      <li class="submenu"><a href="geschichtenweg.html">Geschichte(n)weg Dorfertal</a></li>
      <li class="submenu"><a href="kinderlehrweg.html">Kinderlehrweg Kals</a></li>
      <li class="submenu"><a href="waldlehrpfad.html">Waldlehrpfad Zedlacher Paradies</a></li>
      <li class="submenu"><a href="geolehrpfad.html"> Glorerhütte Geolehrpfad</a></li>
      <li class="submenu"><a href="gletscherlehrpfad.html">Gletscherlehrweg Innergletscher</a></li>
      <li class="submenu"><a href="zirbenwald.html"> Natur- und Kulturlehrweg Oberhauser Zirbenwald</a></li>
      <li class="submenu"><a href="wassererlebnisweg.html">Wassererlebnisweg St. Jakob i. D. (kleine Runde)</a></li>
      <li class="submenu"><a href="glocknerspur.html">Glocknerspur - BergeDenken</a></li>
    </ul>
  </li>
</ul>

```

Abbildung 3: HTML-Code der Ungeordneten Liste des Navigationsmenüs

3.2. Karten-Applikationen

Neben den bereits im Header verarbeiteten Informationen besitzen die Hauptseite und die Seiten der Themenwege eine **Leaflet Karte** mit wechselnden Input. Jedoch sind gewisse Einstellung fest angelegt und können an jeder Karte verwendet werden.

Dazu zählen zum einen, mithilfe von Plus- und Minus-Buttons eine **Zoom- Funktion** (oben links), diese Funktion ist als Standardfunktion von Leaflet Karten enthalten (LEAFLET.JS(2019)).

Sollten in der Karte verarbeitet Icons durch die Veränderung der Zoomgrößen überlappen oder zu wenig Informationen vorhanden sein werden sogenannte **Markercluster** aus der Leaflet-Bibliothek verwendet. Dies ermöglicht jederzeit eine übersichtliche Karte. Innerhalb der Icons sind mithilfe eines Popups weitere nützliche Daten über den Nationalpark eingebaut (GITHUB DANZEL(2019)).

Neben der Zoomfunktion gibt es darunter noch die Möglichkeit, sich die Karte im **Vollbildmodus** anzeigen zu lassen. Sollte die Bearbeitung im Vollbildmodus abgeschlossen sein, lässt sich dieser jederzeit über die ESC-Taste schließen (GITHUB GIULY90 & JFIREBAUGH(2019)).

Das wichtigste Feature jeder Karte befindet sich jedoch auf der rechten oberen Seite. Dort besteht neben verschiedenen Kartentypen die Möglichkeit die gewünschten Daten als **Overlays** in der Karte zuzuschalten. Hierzu zählen die Grenzen des Nationalparks, sowie die Themenwege und den bereits angesprochenen Points of Interest, siehe Abbildung 4.

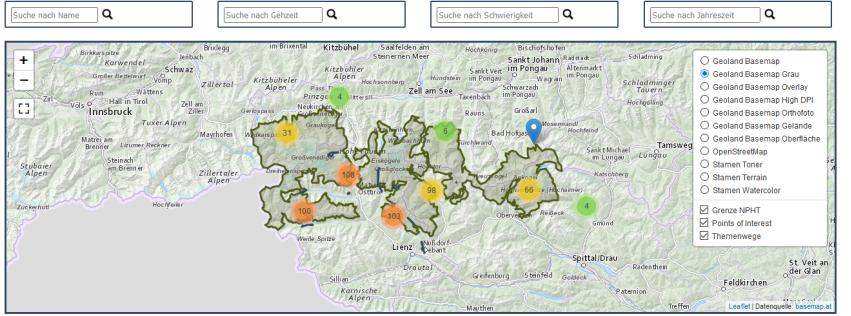


Abbildung 4: Kartenplugin mit der Suchfunktion, der Auswahl der Kartenhintergründe und den Overlays auf der Startseite.

Nur bei den Karten auf den Seiten der Themenwege befindet sich eine **Minimap** um den Überblick behalten zu können, wo sich der Weg genau befindet, siehe Abbildung 5 (GITHUB NORKART(2019)).



Abbildung 5: Zuschaltbare Minimap

Die **Suchfunktion** oberhalb der Karte ermöglicht dem Nutzer nach den vier Kriterien dem Namen der Wege, der Gehzeit, dem Schwierigkeitsgrad und der Jahreszeit einen geeigneten Weg auszuwählen. Der Nutzer kann in die jeweilige Suche das gewünschte Kriterium eintippen und dann in den Vorschlägen auswählen. Dies hat den Vorteil, dass alle Wege mit einem bestimmten Kriterium, also zum Beispiel einer Gehzeit von zwei Stunden beachtet werden können, siehe Abbildung 6. Wenn eine Auswahl getroffen wurde, wird auf den Weg in der Hauptkarte gezoomt. Hier kann man ein Popup öffnen, welches einen Link zur Seite mit dem Themenweg beinhaltet, wie in Abbildung 7 dargestellt (GITHUB STEFANOCUDINI(2019)).

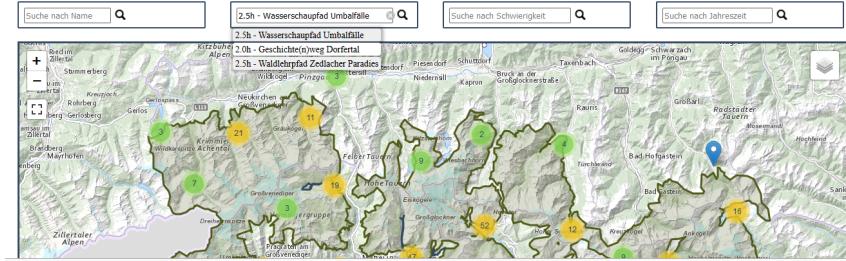


Abbildung 6: Suchfunktion am Beispiel der Gehzeit.

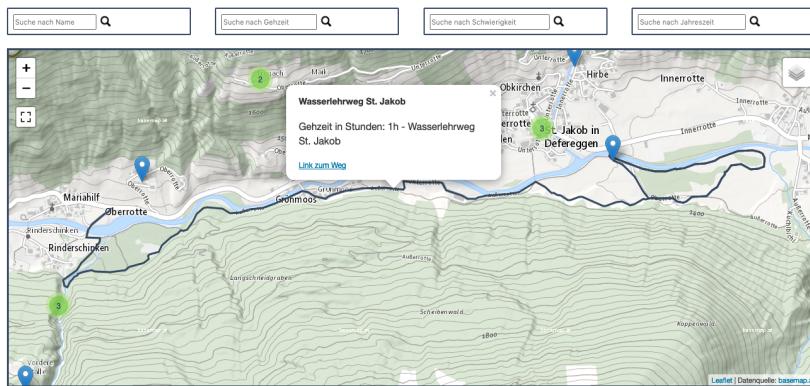


Abbildung 7: Popup mit kurzer Wegbeschreibung und Link zum Weg.

Durch das Hineinladen der gpx Tracks können weitere Informationen über die Wege in einer Liste oberhalb der Karte als **Routenstatistik** wie in Abbildung 8 abgebildet, eingesehen werden ((GITHUB MPETAZZONI(2019)).



Abbildung 8: Routenstatistik, Symbolbild und Kurzinformation des ausgewählten Themenweges.

Zusätzlich kann man auf der Übersichtskarte ein **Höhenprofil** des Weges einblenden. Durch das Bewegen der Maus über den Weg erkennt man, auf welcher Höhe man sich befindet, siehe Abbildung 9 (GITHUB MRMUFFLON(2019)).

Übersicht des Themenwegs



Abbildung 9: Zeigt das Höhenprofil des ausgewählten Themenweges.

Außerdem können bei den Karten der jeweiligen Themenwege die **Steigung** der Route angeschalten werden. Diese ist farblich unterteilt, je nach Steigung.

Übersicht des Themenwegs

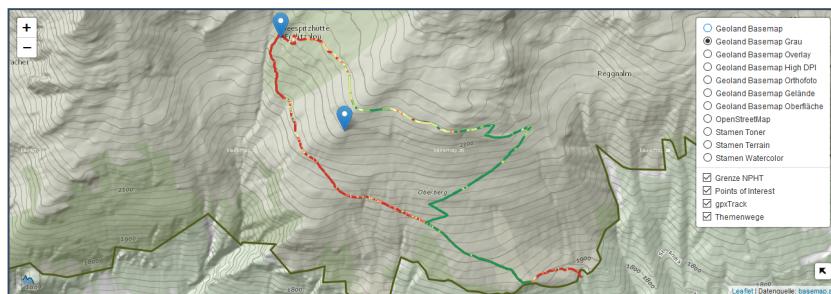


Abbildung 10: Farbliche Codierung der Steigung, durch das Overlay gpxTrack zuschaltbar.

Pop-ups wurden zum einen für die Points of Interest in beiden Karten und zum anderen für die Themenwege in den jeweiligen Übersichtskarten erstellt. Die Daten wurden aus den jeweiligen .js entnommen. Die Pop-ups liefern wichtige Informationen zu den einzelnen Features.



Abbildung 11: Popups des Themenweges (oben) Popup eines POI (unten).

Um eine gute Navigation gewährleisten zu können, wurden auf der Seite und den Popups **Links** eingefügt, welche auf die internen Seiten der Homepage zugreifen lassen, siehe Abbildung 11. Jedoch sind nicht für alle POIs Links vorhanden, deswegen wurde eine neue HTML Seite erstellt, welche auf das Fehlen des Links hinweist.

3.3. Umsetzung der Karten-Applikationen

Die Umsetzung der einzelnen interaktiven Elemente werden im folgenden kurz beschrieben. Das Einbauen der **Leaflet Karte** funktioniert durch die Verlinkung zum .css und .js über die HTML Seite mittels folgenden Links in Abbildung 12. Diese werden im head des HTML Dokuments eingebunden.

```
<!--leaflet verlinken -->
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.5.1/dist/leaflet.css" />
<script src="https://unpkg.com/leaflet@1.5.1/dist/leaflet.js"></script>
```

Abbildung 12: Verlinkung der Leaflet Bibliothek im Header der HTML Dateien.

Im zugehörigen .js werden die Kartenapplikationen bearbeitet. Dies muss auch im HEAD des HTML Dokuments verbunden werden. Eine Implikation einer Kartengrundlage kann wie

in Abbildung 13 aussehen. Als Kartenhintergründe wurden elf Hintergründe verlinkt und diese als Overlay hinzugefügt.

```
let karte = L.map("map");

const kartenLayer = {
  osm: L.tileLayer("https://s.tile.osm.org/{z}/{x}/{y}.png", {
    subdomains: ["a", "b", "c"],
    attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a>  
contributors, <a href="https://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>',
  }),
  geolandbasemap: L.tileLayer("https://s.wien.gv.at/basemap/geolandbasemap/normal/  
google3857/{z}/{y}/{x}.png", {
    subdomains: ["maps", "maps1", "maps2", "maps3", "maps4"],
    attribution: 'Datenquelle: <a href="https://www.basemap.at">basemap.at</a>',
  }),
};
```

Abbildung 13: Einbinden der verschiedenen Kartenhintergründe.

Der **Vollbildmodus** kann durch ein Plug-in hinzugefügt werden. Hierfür benötigt man die Verlinkung im HTML, wie in Abbildung 14 erkenntlich und den Befehl im .js, wie in Abbildung 15 zu sehen ist.

```
<!--Fullscreen Plugin -->
<script src="https://api.mapbox.com/mapbox.js/plugins/leaflet-fullscreen/v1.0.1/
leaflet.fullscreen.min.js"></script>
<link href="https://api.mapbox.com/mapbox.js/plugins/leaflet-fullscreen/v1.0.1/
leaflet.fullscreen.css"
rel="stylesheet" />
```

Abbildung 14: Verlinkung des Vollbildmodus der leaflet Karten.

```
karte.addControl(new L.Control.Fullscreen());
```

Abbildung 15: Hinzufügen der "Vollbildsteuerung".

Für die Funktion der **Markercluster** müssen zunächst die Daten eingelesen werden und Marker erstellt werden, welche dann Mittels des Clusters gebündelt werden können. Dies funktioniert durch die Einbindung des Plugins im HTML und mit einer Marker Cluster Group im .js definiert wie in Abbildung 16 und 17 erkenntlich.

```
<!--Verlinkung MarkerCluster-->
<script src="https://unpkg.com/leaflet.markercluster@1.4.1/dist/
leaflet.markercluster.js"></script>
<link href="https://unpkg.com/leaflet.markercluster@1.4.1/dist/MarkerCluster.css"
rel="stylesheet" />
<link href="https://unpkg.com/leaflet.markercluster@1.4.1/dist/MarkerCluster.Default.css"
rel="stylesheet" />
```

Abbildung 16: Verlinkung des Markercluster Plugins.

```

//-----POI POPUP-----//
let PointsofInterest = L.markerClusterGroup();
const poi_json = L.geoJson(POI)

PointsofInterest.addLayer(poi_json);
PointsofInterest.bindPopup(function(layer) {
    const props = layer.feature.properties;
    const NAME = (props.NAME)
    const popupText = `<h3>${props.NAME}</h3>
<p>Seehöhe: ${props.SEEHOEHE}</p>
<p>Link: ${props.URL_INTERN}</p>`;
    return popupText;
});
karte.addLayer(PointsofInterest);
layerControl.addOverlay(PointsofInterest, "Points of Interest");

```

Abbildung 17: Erstellen der Markercluster für die POI-Popups

Die Implementierung der **Minimap** geschieht ebenfalls mit einem Open-Source Plugin. Die Verlinkung im HTML funktioniert mit dem Code in Abbildung 18 und dem .js Befehl in Abbildung 19.

```

<!--Verlinkung der Minimap -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet-minimap/3.6.1/
Control.Minimap.min.js"></script>
<link href="https://cdnjs.cloudflare.com/ajax/libs/leaflet-minimap/3.6.1/
Control.Minimap.min.css" rel="stylesheet" />

```

Abbildung 18: Verlinkung des Minimap Plugins im HEADER.

```

// Minimap

new L.Control.Minimap(
  L.tileLayer("https://(s).wien.gv.at/basemap/geolandbasemap/normal/google3857/{z}/{y}/{x}.png", {
    subdomains: ["maps", "maps1", "maps2", "maps3", "maps4"],
  }), {
    zoomLevelOffset: -4,
    toggleDisplay: true,
    minimized: true
  })
.addTo(karte);

```

Abbildung 19: Hinzufügen der Minimap zur Karte.

Die **Overlays** der Daten funktioniert ähnlich wie die Einbindung der jeweiligen Kartengrundlagen. Beispielsweise funktioniert die Einbindung der Points of Interest als Overlay wie in Abbildung 17

Die **Suchfunktion** verwendet als Grundlage die Leaflet Control Search von stefanocudini (GITHUB STEFANOCUDINI (2019)). Dieses Plugin ist zunächst für eine einfache Suchanfrage

angepasst. Durch die Mehrfacheinbindung und dem Styling in css und dem mitgelieferten css kann dies unseren Bedürfnissen angepasst werden. Zunächst werden die auf dem Github-Repository verfügbaren Dateien heruntergeladen und in dem Ordner plugins gespeichert. Für die Darstellung der Lupensymbole in der Suchleiste werden die mitgelieferten Such-Icons im Ordner images gespeichert. Die .js und .css werden im HTML, wie in Abbildung 20 erkenntlich, verlinkt.

```
<!--leaflet search -->
<script src="plugins/leaflet-search.min.js"></script>
<link href="plugins/leaflet-search.min.css" rel="stylesheet" />
```

Abbildung 20: Verlinkung des Such Plugins.

Die einzelnen Suchleisten werden mittels einer Konstante erstellt und über die Funktion L.Control.Search definiert. Hierzu wird das jeweilige Kriterium, welches durchsucht werden soll angegeben, wie in Abbildung 21 dargestellt. Der Container 'findbox' wird jeweils für die vier Suchboxen verändert und anhand von .css gestylt, wie in Abbildung 22 zu sehen ist. Zusätzlich wurden noch Kleinigkeiten im, vom Entwickler mitgegeben .css verändert, um dies unseren Vorstellungen nach anzupassen. Wie im Kapitel zur Datengrundlage erwähnt wurden die Daten vom wege.js verwendet und durch die Namensinformation ergänzt. Die Texte wurden noch angepasst und der Platzhalter durch das jeweilige Kriterium verändert.

```
// Suchleiste einfügen
const suchFeld_Name = new L.Control.Search({
  layer: Themenwege_json,
  propertyName: 'NAME_DE',
  marker: false,
  initial: false,
  collapsed: false,
  textPlaceholder: "Suche nach Name",
  position: 'topright',
  container: 'findbox',
  autoCollapse:true,
});
```

Abbildung 21: Erstellen und definieren des Suchfeldes.

```
#findbox {
  border-radius:.125em;
  border:2px solid #rgba(52, 73, 97,1);
  margin-bottom: 20px;
  padding: 4px 4px 4px 4px;
  width: 20em;
  height:30px;
  position: topleft;
}
```

Abbildung 22: Styling des Suchfeldes in der CSS-Datei.

Die **Routenstatistik** wird durch eine Verlinkung des gpx track Plugins von mpetazzoni (GITHUB MPETAZZONI (2019)) erreicht. Neben dieser Verlinkung im HTML müssen "Anker" im HTML geschaffen werden (Abbildung 23), welche dann im .js durch eine Abfrage des .gpx tracks gefüllt werden (Abbildung 24).

```
<!-- verlinken des GPX plugins -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet-gpx/1.4.0/gpx.min.js"></script>

<h1>Routen Statistik:</h1>
<p>
    Niedrigster Punkt: <b><span id="min"></span> m</b> <br>
    Höchster Punkt: <b><span id="max"></span> m</b> <br>
    Höhenunterschied: <b><span id="diff"></span> m</b>

```

Abbildung 23: Verlinkung des GPX-Track Plugins und setzen sogenannter Anker in dem HTML-Datei.

```
)}.on('loaded', function(e) {
    karte.fitBounds(e.target.getBounds());

    const minSpan = document.getElementById("min");
    const maxSpan = document.getElementById("max");
    const diffSpan = document.getElementById("diff");

    minSpan.innerHTML = e.target.get_elevation_min();
    maxSpan.innerHTML = e.target.get_elevation_max();
    diffSpan.innerHTML = Math.round(e.target.get_elevation_gain());

}).on('addline', function(e) {
```

Abbildung 24: Funktion welche aus den GPX-Daten die Routenstatistik ausliest.

Das **Höhenprofil** wird mit einem Plugin des Github Users Raruto (GITHUB RARUTO (2019)) umgesetzt. Dafür müssen die Links von Abbildung 25 in die HTML Seite eingebunden werden und das Skript von Abbildung 26 in das .js file.

```
<!-- D3.js -->
<script src="https://unpkg.com/d3@4.13.0/build/d3.min.js" charset="utf-8"></script>

<!-- leaflet-elevation -->
<link rel="stylesheet" href="https://unpkg.com/@raruto/leaflet-elevation@latest/leaflet-elevation.css"
/>
<script src="https://unpkg.com/@raruto/leaflet-elevation@latest/leaflet-elevation.js"></script>
```

Abbildung 25: Verlinkung des Höhenprofil Plugins in der HTML-Seite.

```

const controlElevation = L.control.elevation({
  collapsed: true,           //für in Karte Implementierte Höhenprofile.
  position: "bottomleft",
});

controlElevation.addTo(karte);
controlElevation.addData(e.line);

```

Abbildung 26: Erstellen und Hinzufügen des Höhenprofils

Die **Steigung** der Routen wird durch eine Schleife erstellt, welche je nach Steigung auf eine bestimmte Farbe zugreift, welcher ein Wert der Steigung zugewiesen wurde. Als Verlinkung zu den Höheninformationen des gpx muss, wie in der Verlinkung des Höhenprofils das d3.js verlinkt werden, welches in Abbildung 27 dargestellt wird.

```

<!-- D3.js -->
<script src="https://unpkg.com/d3@4.13.0/build/d3.min.js" charset="utf-8"></script>

```

Abbildung 27: Verlinkung des d3- Plugins.

Popups sind keine externen Plugins sondern werden in der Leaflet Bibliothek mitgeliefert und benötigen deshalb keine Verlinkung im HTML. Wie Popups erstellt werden können ist in Abbildung 17 (Markercluster) erkenntlich.

Die **Links** zur Navigation auf der Seite wurden im HTML definiert, wie in Abbildung 28 zu sehen.

```

<div id="menu">
  <ul>
    <li class="topmenu"><a href="index.html">Home</a></li>
    <li class="topmenu"><a href="uberuns.html">Über uns</a></li>
    <li class="topmenu"><a href="auswahl.html">Wege</a>
      <ul>
        <li class="submenu"><a href="kulturlehrwegDebant.html">Natur - und Kulturlehrweg Debanttal</a>
        <a></li>
        <li class="submenu"><a href="alpenblumenweg.html"> Alpenblumenweg Oberseite (kurzer Rundweg)</a>
        <a></li>
        <li class="submenu"><a href="alpenblumenweg_oberseite.html">Alpenblumenweg Oberseite (langer Rundweg)</a></li>
        <li class="submenu"><a href="wasserschaupfad.html">Wasserschaupfad Umbalfälle</a></li>
        <li class="submenu"><a href="bergerlebnisweg.html">Nussdorfer Berg Erlebnisweg</a></li>
        <li class="submenu"><a href="wasserlehrweg.html">Wasserlehrweg St. Jakob i. D.</a></li>
        <li class="submenu"><a href="geschichtengweg.html">Geschichte(n)weg Dorfertal</a></li>
        <li class="submenu"><a href="kinderlehrweg.html">Kinderlehrweg Kals</a></li>
        <li class="s" </li>
        <li class="s" The li element represents a list item. If its parent element is an ol, ul, or menu
        <li class="s" element, then the element is an item of the parent element's list, as defined for
        </a></li> those elements. Otherwise, the list item has no defined list-related relationship
        <li class="s" to any other li element.
        <li class="s" Runde)</a></li>
        <li class="submenu"><a href="glocknerspur.html">Glocknerspur - BergeDenken</a></li>
      </ul>
    </li>
  </ul>

```

Abbildung 28:: Verlinkung zu den Themenwegseiten

4. Probleme und Herausforderungen

Bei einem Projekt im Rahmen einer universitären Lehrveranstaltung stößt man immer wieder auf Probleme und Herausforderungen. Dies gilt ganz besonders auch für dieses Projekt. So ist die Suche wie sie jetzt auf der Homepage umgesetzt wurde eine Kompromisslösung. Zu Beginn des Projektes wäre geplant gewesen, dass es eine "Do-it-all" Suche geben soll. Also lediglich ein Suchfeld, welches vier verschiedenen Properties (Gehzeit, Name, Jahreszeit, Schwierigkeit) des JSON abfragen kann. Da die Umsetzung den Rahmen, sowie unser Know-How gesprengt hätte, haben wir uns für die nun vorhanden Lösung entschieden.

Ein weiteres Problem stellten die teilweise lückenhaften Daten dar. So sind zum Beispiel bei vielen POIs keine Höhenwerte im JSON hinterlegt, sodass im PopUp bei der Seehöhe lediglich "NULL" dargestellt wird. Diesen Umstand haben wir mit dem Ersetzen des Eintrages "NULL" durch "Nicht vorhanden" in der JSON-Datei bereinigt.

Da alle Bilder natürlich frei verfügbar sein müssen war das Finden geeigneter Bilder für diverse Themenwege sehr schwer bis unmöglich. Somit musste teilweise Symbolbilder verwendet werden. Mit genügend Zeit und bei einem größer angelegten Projekt könnten benötigte Bilder selbst erstellt bzw. vom Auftraggeber gestellt werden, womit man dieses Problem umgeht.

5. Fazit

Die Erstellung unserer Website hat uns als Kollektiv vor einige Herausforderungen gestellt, wovon die meisten aber im Team und mit den erlernten Fähigkeiten bzw. mit dem andenken neuer Lösungsansätze sehr gut gelöst werden konnten.

Die Lehrveranstaltung ist eine gute Einführung in die Welt der Programmiersprachen und damit verbundenen neuen Denkansätze und Denkanstöße, welche wir im Zuge unseres Projektes noch erweitern konnten und einen guten ersten Einstieg in die Welt des Programmierens geboten hat. Auf die Grundkenntnisse, welche der Kurs vermittelte kann nun selbstständig weiter aufgebaut werden. Die Fähigkeit selbstständig Webpages zu erstellen ist eine wertvolle Zusatzqualifikation. Diese Fähigkeit mit dem Erstellen von Webkarten zu verknüpfen bietet sich bei Geographen an und ist der logische nächste Schritt.

7. Quellen

Kommentiert [1]: Die anderen Plugins sollten a nu als Quelle drin sein?

OPEN DATA ÖSTERREICH(2017B): Themenwege NPHT Tirol 2017. URL.:
<https://www.data.gv.at/katalog/dataset/A7FB6495-8FB3-5241-C0A1-AE7F0B9EF892>.
[02.06.19]

OPEN DATA ÖSTERREICH(2017A): Points of Interest NPHT.URL.:
<https://www.data.gv.at/katalog/dataset/2804DB77-4DC5-5480-5379-4C2AA4472EB5>.
[02.06.19]

OPEN DATA ÖSTERREICH(2018): Nationalpark Hohe Tauern Aussengrenze (11/2018). URL.:
<https://www.data.gv.at/katalog/dataset/9AEFA112-D22B-5FFA-E996-9A221C2E4FD3>.
[02.06.19]

GITHUB STEFANOCUDINI (2019): Search stuff in a Leaflet map. URL.:
<https://github.com/stefanocudini/leaflet-search#leaflet-control-search>. [05.06.19]

GITHUB MPETAZZONI (2019): A GPX track plugin for Leaflet.js.
URL.:<https://github.com/mpetazzoni/leaflet-gpx>. [05.06.19]

GITHUB GIULY90 & JFIREBAUGH (2019): A fullscreen control for Leaflet
URL.:<https://github.com/Leaflet/Leaflet.fullscreen>. [05.06.19]

GITHUB NORKART (2019): A minimap control plugin for Leaflet
URL.:<https://github.com/Norkart/Leaflet-MiniMap>. [05.06.19]

GITHUB DANZEL (2019): Marker Clustering plugin for Leaflet
URL.:<https://github.com/Leaflet/Leaflet.markercluster>. [05.06.19]

LEAFLET.JS (2019): Leaflet an open-source JavaScript library
URL.:<https://leafletjs.com> [05.06.19]

PIXABAY (2019) URL.:<https://pixabay.com> [05.06.19]