

Projekt HydroTirol

Fließgewässer in Tirol

Visualisierung hydrologischer und raumplanerischer Daten



Projektbericht

Maximilian Ender & Anna Keuschnigg

Sommersemester 2021

716409-0 / VU Geoinformatik: Web mapping

Klaus Förster BSc & Mag. Bernd Öggl

INHALTSVERZEICHNIS

<u>1. WAS WIRD DARGESTELLT?</u>	<u>3</u>
<u>2. WIE SOLL ES DARGESTELLT WERDEN?</u>	<u>4</u>
2.1. PROJEKTPHASE UND AUFBAU	4
2.2. ARBEITSAUFTEILUNG/ ZEITMANAGMENT.....	14
2.3. PROBLEME UND KOMPLIKATIONEN.....	14
<u>3. ERGEBNIS/FAZIT.....</u>	<u>15</u>
QUELLEN	16

1. Was wird dargestellt?

Um unsere neu erlernten Kenntnisse bei der Websiteerstellung in einem eigenen Projekt darzustellen, wurde sich darauf geeinigt hydrologische Daten von Tirol zu bearbeiten. Dieser Themenbereich liefert, aufgrund der zentralalpinen Lage Tirols und den damit verbundenen Naturgefahren einen umfangreichen Datenschatz.

Hierbei handelt es sich primär um eine Darstellung von hydrologischen in Kombination mit raumplanerischen Daten (z.B. Gefahrenzonenplan). Weiters soll eine genauere hydrologische Übersicht der großen Talflüsse eingebunden werden, um einen Überblick über deren Abflussverhalten (Vergleich Abflussspende, usw.) porträtieren zu können.

Für die Projektvisualisierung werden vier HTML-Seiten erstellt, deren Thematiken wie folgt lauten:

1) Einführung mit Erläuterungen: könnte als eine Art „Startseite“ verstanden werden, welche grundlegende Informationen zur Thematik (z.B. kurze Erklärung zu hydrologischen und raumplanerischen Daten) beinhaltet; mehrheitlich text- und bildbasiert.

2) Übersicht über hydrologische Daten: Fokus liegt auf einer Übersichtskarte, die alle hydrologischen Messstationen (Wasserstandsdaten) und auch Wetterstationsdaten (z.B. Schneehöhe, Temperatur) des Bundeslandes Tirol mit ihren Momentanwerten darstellt.

3) Übersicht über raumplanerische Daten: Fokus liegt auf einer Übersichtskarte, die ausgewählte themenbezogene raumplanerische Daten (z.B. gelbe/rote Zone der WLV; Hochwasserüberflutungsflächen der Bundeswasserbauverwaltung, usw.) beinhaltet.

4) Flussportraits: hierbei sollen einige ausgewählte große Talflüsse im Bundesland Tirol überblicksmäßig in Kartenform dargestellt werden, in Frage kommen z.B. Inn, Lech, Öztaler Ache, Ziller; der Fokus liegt wiederum auf einer Übersichtskarte, in der die einzelnen Fließgewässer mittels Popup-Fenster vorgestellt werden.

Da es sich möglichst um OpenData handeln soll, werden vorwiegend offizielle Internetseiten für die Datenbeschaffung herangezogen, beispielsweise *data.gv.at* oder *ehyd.gv.at*. Für die Darstellung und richtige Interpretation der Flussportraits wird tiefergehende Literatur herangezogen (vgl. Muhar, et. al.: 2019; Fohrer, et. al.: 2016).

2. Wie wird dargestellt?

Organisation und Repository erstellen: Um an diesem Projekt von beiden PC's arbeiten zu können, wird eine Organisation mit dem Namen *hydrotirol* auf Github erstellt. In dieser wird ein öffentliches Repository namens *hydrotirol.github.io* angelegt, um hierin alle erstellten Dateien dieses Projekts einbinden zu können, auch das Konzept des Projekts liegt hier als PDF-File. [für das Github-Repository: <https://github.com/hydrotirol/hydrotirol.github.io>]

2.1. Projektphase und Aufbau

Repository wird in *Visual Studio Code* durch *Clone Repository* und kopieren des https-Code aus Github eingebunden und lokal abgespeichert.

index.html: Neues File *index.html* wird erstellt und mit den standardgemäßen HTML-Einstellungen (siehe Abb. 1) versehen.



```
... @@ -0,0 +1,12 @@
1 + <!DOCTYPE html>
2 + <html lang="en">
3 + <head>
4 +   <meta charset="UTF-8">
5 +   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 +   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 +   <title>Document</title>
8 + </head>
9 + <body>
10 +
11 + </body>
12 + </html>  
```

Abbildung 1: Screenshot aus *index.html* - Code bei Erstellung.

Nachfolgend wurden die Verlinkungen zu den *stylesheets* eingebunden (unter anderem *cdnjs.cloudflare* und *fonts*) sowie zur *style.css*.

Der Website wurde vorerst *Hydrotirol* benannt.

Im *<header>* wurde ein banner-Foto eingefügt, sowie die Verlinkung zu unseren Email-Adressen und unserem Github-Repo, diese wurden jeweils mit einem ausgewählten Icon abgespeichert.

Anschließend folgt der *<main>* für eine erste textliche Darstellung.

Sind diese ersten Einstellungen der Website erfolgt, kann der **style.css** mit den unterschiedlichen Regeln für Bilder und Textformate erstellt werden (siehe Abb. 2).

```

1  body {
2      max-width: 1440px;
3      margin: auto;
4      font-family: "Open Sans", sans-serif;
5  }
6
7  header {
8      max-width: 1440px;
9      text-align: center;
10 }
11
12 header p {
13     text-align: right;
14     font-weight: bold;
15     font-size: 0.8em;
16     padding-right: 1em;
17     margin-top: -2em;
18 }
19
20 .white,
21 .white a:link,
22 .white a:visited {
23     color: white !important
24 }
25

```

Abbildung 2: Ausschnitt aus style.css, zu sehen ist ein Ausschnitt der Voreinstellungen, welche die Formatierung der Webseiten definiert.

Als Nächstes werden mehrere **.html** erstellt und durch eine Liste ins index.html eingebunden: **flussportraits.html; hydro.html & rp.html**. Anschließend wurden Begriffserklärungen und weiterer Text in das index.html eingebunden und der style.css optisch angepasst. Die Verlinkungen im index.html zu den anderen HTML-Seiten sieht wie auf Abbildung 3 und 4 zu erkennen aus.

```

62 <ul>
63     <li>
64         <a href="hydro.html" target="_blank"><strong>Darstellung hydrologischer Daten</strong></a>
65         <br>Übersichtskarte, auf der
66         hydrologische Daten (z.B. Abfluss- und Niederschlagsdaten) des Bundeslandes Tirol visualisiert werden.
67     </li>
68     <br>
69     <li>
70         <a href="rp.html" target="_blank"><strong>Darstellung raumplanerischer Daten: </strong></a> <br>
71         Übersichtskarte,
72         auf der raumplanerische Daten im Kontext alpiner Naturgefahren (z.B. Gefahrenzonenplan) visualisiert
73         werden.
74     </li>
75     <br>
76     <li>
77         <a href="flussportraits.html" target="_blank"><strong>Flussportraits</strong></a> <br> Übersichtskarte,
78         auf der
79         ausgewählte größere Talflüsse des Bundeslandes Tirols hinsichtlich ihrer hydrologischer Eigenschaften
80         präsentiert werden.
81     </li>
82 </ul>
83

```

Abbildung 3: Code für Beschreibung des Projektinhaltes

Projekthinhalte

- Darstellung hydrologischer Daten

Übersichtskarte, auf der hydrologische Daten (z.B. Abfluss- und Niederschlagsdaten) des Bundeslandes Tirol visualisiert werden.

- Darstellung raumplanerischer Daten:

Übersichtskarte, auf der raumplanerische Daten im Kontext alpiner Naturgefahren (z.B. Gefahrenzonenplan) visualisiert werden.

- Flussportraits

Übersichtskarte, auf der ausgewählte größere Talflüsse des Bundeslandes Tirols hinsichtlich ihrer hydrologischen Eigenschaften präsentiert werden.

Abbildung 4: Darstellung und Beschreibung der anderen Websites im index.html mit eingebundener Verlinkung durch die Unterüberschriften; zeigt, was der geschriebene Code siehe Abb.3 darstellt.

Die HTML-Seiten wurden durch eine Verlinkung über die Unterüberschriften in den Bereich *Projekthinhalte* eingebunden. Daraufhin folgen einige Begriffserklärungen (siehe Abb. 5) und Definitionen, welche auf den verlinkten Websites thematisiert werden. Am Ende der index.html lassen sich alle verwendeten Quellen für Text und Abbildungen finden. All diese unterschiedlichen Informationen wurden durch das Erstellen einer ungeordneten Liste `` und Einfügen eigener Listenelemente `` eingefügt (siehe Abb. 5).

Begriffserklärungen

- **Abfluss:** Jenes Wasservolumen V , das in einem Fließgewässer in einer bestimmten Zeiteinheit t die Querschnittsfläche A durchströmt.
- **Hochwasser:** Eine zeitlich begrenzte Überschreitung von bestimmten Abflussschwellenwerten als Folgeerscheinung meteorologischer Ereignisse, geht zumeist mit der Überschwemmung von Binnenland einher.
- **Niederwasser:** Eine zeitlich begrenzte Unterschreitung von bestimmten Abflussschwellenwerten als Folgeerscheinung meteorologischer Ereignisse, wie z.B. Dürre.
- **HQ100:** Als HQ100 wird ein sog. "hunderjährliches" Hochwasser verstanden (auch als Jahrhunderthochwasser bezeichnet). Dies definiert ein Abflussereignis, welches statistisch gesehen genau einmal in 100 Jahren auftritt.
- **Hochwasserbemessung:** Ingeniertechnische ermittelte und modellierte Hochwasserscheitelabflüsse innerhalb einer Zeitperiode t , d.h. wie oft tritt ein bestimmter Abflusswert in einer bestimmten Zeitspanne auf (z.B. 100-jährliches Hochwasser hat eine exakte Eintrittswahrscheinlichkeit von $1/100$).

Abbildung 5: Benötigte Begriffserklärungen und wie diese im index.html dargestellt wurden.

Auf den unterschiedlichen eingebundenen HTML-Seiten wurde folgendes dargestellt:

Hydro.html: Wird die erste Unterüberschrift (Darstellung hydrologischer Daten) des Projekts angetippt, gelangt man auf die hydro.html. Hier wurde, wie auf allen anderen Seiten, nach dem Banner-Bild eine Adresszeile mit passenden Icons eingefügt.

Am Anfang des HTMLs finden sich die unterschiedlichen Code-Referenzen, welche für die unterschiedlichen Styles auf der Website nötig sind; hier lassen sich unter anderem mehrere Leaflet-Links und weiters eine fonts-Verlinkung für andere Schriftarten finden. Durch die Leaflet-Links wurde zum Beispiel die *Minimap*, der Fullscreen und der Rainviewer sowie die Vordefinitionen der Color.js, style.css und der paparse.min.js eingebunden (siehe Abb.6).

```

4  <head>
5    <meta charset="UTF-8">
6    <meta http-equiv="X-UA-Compatible" content="IE=edge">
7    <meta name="viewport" content="width=device-width, initial-scale=1.0">
8    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css" />
9    <link rel="preconnect" href="https://fonts.gstatic.com">
10   <link
11     href="https://fonts.googleapis.com/css2?family=Open+Sans:ital@0;1&family=Roboto+Condensed:ital@0;1&display=swap"
12     rel="stylesheet">
13   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.7.1/leaflet.css">
14   <script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.7.1/leaflet-src.min.js"></script>
15   <link rel="stylesheet" href="https://cdn.jsdelivr.net/gh/mwasil/Leaflet.Rainviewer/leaflet.rainviewer.css" />
16   <script src="https://cdn.jsdelivr.net/gh/mwasil/Leaflet.Rainviewer/leaflet.rainviewer.js"></script>
17   <script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet-providers/1.12.0/leaflet-providers.min.js"
18     integrity="sha512-LixflAm9c0/qONbz9F1Ept+QJ6QBpb7wU1Puyv1EHl0TVgwSK8j3yMV3elnE1GQcv7Y5QTF1F/FqyeE/N4LnKQ=="
19     crossorigin="anonymous"></script>
20   <!-- Leaflet Fullscreen-->
21   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/leaflet.fullscreen/2.0.0/Control.FullScreen.css"
22     integrity="sha512-KQk/GTCcAywe4iEOhnZ6ZmWdNv/3NvOE6f0iUvrNVCX/oD7+sPXAeDP90aOX3EXHmerC6gDYU1UBa5d4hxjWE=="
23     crossorigin="anonymous" />
24   <script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet.fullscreen/2.0.0/Control.FullScreen.js"
25     integrity="sha512-TaNrKSd5T0m4PfgJMFYkDpQ1X8GmqXBlh+Kmk83Mfrcx9sJtZqI1zcZd0n0xpdJevfPL7voAcbb1rUaYx+LM1g=="
26     crossorigin="anonymous"></script>
27   <!-- Leaflet Minimap-->
28   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/leaflet-minimap/3.6.1/Control.Minimap.css"
29     integrity="sha512-efbAfGnnrjA+hLw0Lu91W034fBGPsMwZMVCTwLUI2PDX/m7r0iuhYZ+D2mZ8RkCpC/I/7pdgoL8T4eYvMHNoQg=="
30     crossorigin="anonymous" />
31   <script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet-minimap/3.6.1/Control.Minimap.js"
32     integrity="sha512-ceQPs2CHke3gSINL/JV37W1rF3JOM64yuh999hnRhTP7tNtcS8p5h1TKhn8CEIhsFweS8rZMPVotAKjoyxGwNg=="
33     crossorigin="anonymous"></script>
34   <!-- Papa parse -->
35   <script src="papaparse.min.js"></script>
36
37   <script src="colors.js"></script>
38
39   <link rel="stylesheet" href="style.css">

```

Abbildung 6: Eingefügte Verlinkungen im hydro.html.

Die in der Karte eingebundenen hydrologischen Daten, sind Online durch eine CSV-Datei eingebunden worden. Dazu kommt ein sogenannter Parser (*Papaparse*): Durch das Erstellen der **papaparse.min.js** können die Online-abgerufenen CSV-Werte in Echtzeit-JSON-Werte umgewandelt und unkompliziert dargestellt werden (sog. *Parsen*). Diese **papaparse.min.js** (siehe Abb. 7) ist notwendig, damit das Popup überhaupt eingebunden werden kann.


```

1  /* @license
2  Papa Parse
3  v5.3.1
4  https://github.com/mholt/PapaParse
5  License: MIT
6  */
7  !function (e, t) {
8    "function" == typeof define && define.amd ? define([], t) : "object" == typeof module && "undefined" != typeof exports ? module.exports = t() : e.Papa = t()
9  }(this, function s() {
10    "use strict";
11    var f = "undefined" != typeof self ? self : "undefined" != typeof window ? window : void 0 != f ? f : {};
12    var n = !f.document && !f.postMessage,
13        o = n && /blob:\/i.test((f.location || {}).protocol),
14        a = {},
15        h = 0,
16        b = {
17          parse: function (e, t) {
18            var i = (t = t || {}).dynamicTyping || !1;
19            M(1) && (t.dynamicTypingFunction = i, i = {});
20            if (t.dynamicTyping = i, t.transform = !M(t.transform) && t.transform, t.worker && b.WORKERS_SUPPORTED) {
21              var r = function () {
22                if (!b.WORKERS_SUPPORTED) return !1;
23                var e = (i = f.URL || f.webkitURL || null, r = s.toString(), b.BLOB_URL || (b.BLOB_URL = i.createObjectURL(new Blob(["(", r, ")();"]), {
24                  type: "text/javascript"
25                }))),
26                    t = new f.Worker(e);
27                var i, r;
28                return t.onmessage = _, t.id = h++, a[t.id] = t;
29              }();
29              return r.userStep = t.step, r.userChunk = t.chunk, r.userComplete = t.complete, r.userError = t.error, t.step = M(t.step), t.chunk = M(t.chunk), t.complete = M(t.complete),
30                input: e,
31                config: t,
32                workerId: r.id
33              })
34            }
35          }
36        }

```

Abbildung 7: Ausschnitt aus der papaparse.min.js; Code für Umwandlung von csv zu json.

Die im Code verlinkte **hydro.js** beinhaltet alle Informationen für die Darstellung der Karte sowie den aufscheinenden Pop-ups (*.bindPopup*).

Für das Pop-up der hydrologischen Messstationen wird der Stationsname, der Gewässername und der aktuelle Wasserstand und weiters Datum und Uhrzeit (mit *.toLocaleString*, auf die hierzulande benützte Datumsschreibweise („de“) umgewandelt) sowie die Stationsnummer (siehe Abb. 8) angezeigt.



Abbildung 8: Dargestellte Karte im hydro.html, mit Popup. Rechts Open: Auswahl für Baselayers und Dateneinstellungen. Rechts unten Rainviewer und Minimap.

Analog zu dieser Vorgangsweise wurden einige Wetterstationsdaten (Temperatur und Schneehöhe) als Popup in einen zweiten Overlay („Wetterdaten“) eingebunden, der Code gleicht jenem des AWS-Tirol-Beispiels. Der Vorteil dieser Daten liegt darin, dass diese bereits im JSON-Format verfügbar sind.

Folgend lässt sich rechts unten in der Karte die *Minimap* und der *Rainviewer* bedienen. Die von leaflet eingebundene Minimap dient einer räumlichen Übersicht. Bei Betätigen des Rainviewers, kann die aktuelle und bis zu 2h vergangene Niederschlagssituation vom im Kartenbereich visualisiert werden.

Am Ende dieser Seite befinden sich die Verlinkungen mit kurzer Beschreibung zu den anderen zum Projekt gehörigen HTML-Seiten.

rp.html: Wenn Darstellung raumplanerischer Daten aufgerufen wird, wird man zur rp.html weitergeleitet. Auch hier sieht man ein Foto (eigene Aufnahme) am Anfang der Seite mit den Adresszeilen-Verlinkungen von Emails und *Repository* direkt darunter.

Am Anfang des Codes wurden wiederum die unterschiedlichen Verlinkungen der *styles* eingefügt.

Darunter folgt wiederum eine Karte, welche wiederum durch die **rp.js** definiert wurde. Hierin werden die Überflutungszonen, kategorisiert durch die nachfolgenden Begriffsdefinitionen in unterschiedlicher Farbgebung, dargestellt. Zudem wurde wieder ein Popup eingebaut, in welcher die jeweilige Kategorie-Bezeichnung angeführt wird, eingebaut.

Da die zur Verfügung stehenden Überflutungszonen zu viel Speicherplatz einnahmen, wurden diese in QGIS vereinfacht, ein geographisches Koordinatensystem eingestellt und konnten somit wiederum in das Repository eingefügt werden.

Abgesehen von den Überflutungsgebieten in Tirol wurde hierbei wieder eine *Minimap* eingebunden (Siehe Abb. 9).

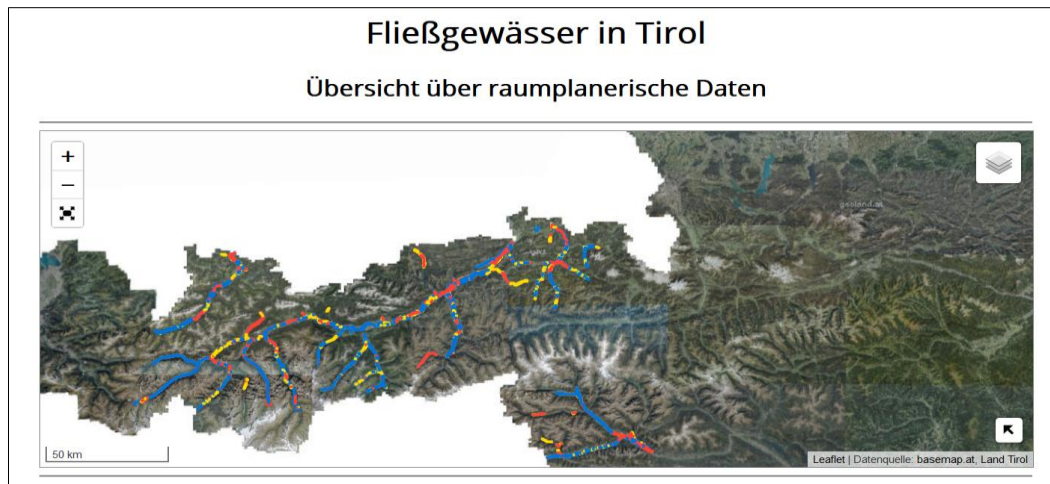


Abbildung 9: Dargestellte Karte im *rp.html*

Nachfolgend wurden Begriffsdefinitionen der unterschiedlichen in der Karte dargestellten Gefahrenzonen gegeben. Diese wurden größtenteils nach dem Bericht des *Bundesministerium für Land- und Forstwirtschaft, Umwelt- und Wasserwirtschaft (2016)* übernommen.

flussportraits.html: Diese Seite erscheint, wenn die letzte Unterschrift des Projektinhaltes angetippt wird. In diese HTML-Seite, wurde wiederum ein selbstgemachtes Foto (Header) und eine Karte mit fünf ausgewählten Flüssen in Tirol eingebunden. Unter dem Banner wurden wiederum die Email-Adressen und die Github-Organisation mit ausgewählten ICONs dargestellt (Adresszeile).

Die Informationen für die dargestellte Karte wurden, in diesem Fall in das sog. **fp.js** eingebunden. In dieser wurden, wie in den vorherigen Seiten, die unterschiedlichen *Basemaps* eingebunden und die Darstellung der Karte definiert.

Das *Shapefiles* für das Gewässernetz Tirol wurde aus *opendata.gv.at* entnommen und in *QGIS* auf fünf ausgewählte Flüsse (Inn, Isel, Lech, Ötztaler Ache und Ziller) reduziert und an die Grenzen des Bundeslandes Tirol angepasst. Diese wurden in die Karte eingebunden und durch eine unterschiedliche Farbwahl abgegrenzt. Zusätzlich wurde die *Leaflet-Minimap* eingebunden.

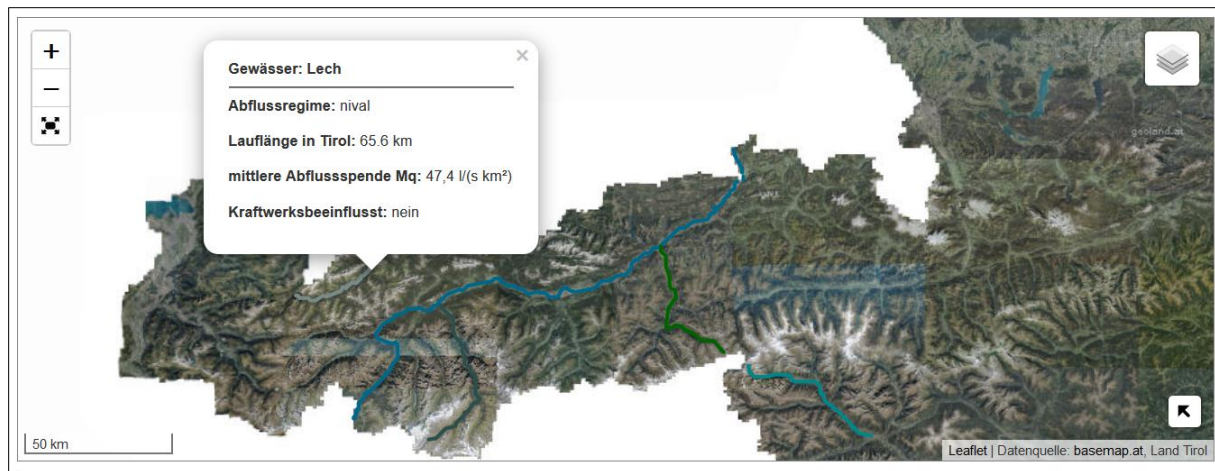


Abbildung 10: Dargestellte Karte im flussportraits.html.

Bei Klicken auf die abgebildeten Flüsse öffnet sich ein *Popup* (Visualisierung durch den Befehl `.bindPopup`). Hierin wird der Gewässername, das Abflussregime (ermittelt aus Betrachtung des Jahreabflussganges in *Hydro Online* des Bundeslandes Tirols), die Lauflänge in Tirol, die mittlere Abflusssspende und eine Kraftwerksbeeinflussung (ja/nein, vgl. TRUE/FALSE) dargestellt (siehe Abb. 10). Diese Informationen wurden (bei der Auswahl der fünf Flüsse) mit Hilfe von *QGIS* berechnet oder anderwärtig recherchiert und zusätzlich der Raumbezug eingebunden und durch eine Umwandlung des Koordinatensystems und des Dateiformats (Shapefile → JSON) eingebunden (siehe Abb. 11).

	A	B	C	D	E	F	G
1	Fluss	Abflussregim	Länge	Mq	MQ	kraftwerksbeeinflusst	
2	Inn	nivo-glazial	207.08	28,9 l/(s km ²)	165 m ³ /s (Inr ja		
3	Isel	nivo-glazial	58.9	32,4 l/(s km ²)	38,8 m ³ /s (Li	nein	
4	Ötztaler Ache	glazial	66.53	37,4 l/(s km ²)	31,3 m ³ /s (B	nein	
5	Lech	nival	65.62	47,4 l/(s km ²)	44,1 m ³ /s (L	nein	
6	Ziller	nivo-glazial	55.74	39,5 l/(s km ²)	44,5 m ³ /s (H	ja	

Abbildung 11: Erstellte Tabelle (Inhalt des Popups), bevor diese in ein GeoJson umgewandelt wurde.

Darunter lässt sich eine kurze Beschreibung der unterschiedlichen Flussportraits finden. Am Ende der Seite lässt sich wiederum ein Abschnitt für die alle anderen Projektinhalte verlinkt.

Beim Erstellen der unterschiedlichen HTML-Seiten, kommt es zur Bearbeitung der verschiedenen, eingebundenen *Files*.

OGD_HYDRO.js: Hierin wurden die Verlinkungen (siehe Abb. 12.1) zu den Dateien, für die Darstellung und nachfolgender Einbindung in die Karten niedergeschrieben.

```
1  const OGDHYDRO = [{
2    title: "Überflutungsflächen",
3    source: "https://data-tiris.opendata.arcgis.com/datasets/gefahrenzonen-wasser/explore",
4    data: "data/Gefahrenzonen_Wasser.geojson"
5  }, {
6    title: "Flussportraits",
7    data: "data/fp.geojson"
8  }, {
9    title: "HydroDaten",
10   data: "https://wiski.tirol.gv.at/hydro/ogd/OGD_W.csv"
11  }]
```

Abbildung 12.1: Inhalt der OGD_HYDRO.js

```
62  for (let config of OGDHYDRO) {
63    console.log("Config: ", config.data);
64    fetch(config.data)
65      .then(response => response.json())
66      .then(geoJsonData => {
67        if (config.title == "Flussportraits") {
68          drawportraits(geoJsonData);
69        }
70      })
71  }
```

Abbildung 12.2: Beispiel der Verlinkung, in diesem Fall aus dem fp.js.

color.js: In diesem File sind die unterschiedlichen Farbwerte für die unterschiedlichen Überflutungsflächen, die hydrologischen Daten, die Flussportraits und die Temperaturanzeigen erläutert. Eine Verlinkung erfolgt den Befehl **const COLORS** erfolgen.

style.css: Hierin werden alle Formatierungen für Schriftarten, dargestellte Bilder und Karten festgelegt.

Main.js: (bei Bereinigung des *Repositorys* gelöscht, da nicht benötigt für die erstellten HTML-Seiten.). Diese wurde erstellt, da zu Beginn nicht klar war, ob auf der index.html auch eine Karteneinbindung erfolgt. Hierin erfolgte die Verlinkung und Benennung der verschiedenen *Baselayers*, welche dann jeweils in die unterschiedlichen .js für die unterschiedlichen HTML-Seiten eingebunden. Dies wurde jeweils am Anfang der Codes eingefügt und sieht wie folgend aus:

```
1  let basemapGr = L.tileLayer.provider('BasemapAT.grau');
2
3  const map = L.map("map", {
4    fullscreenControl: true,
5    center: [47.25, 11.5],
6    zoom: 7.5,
7    layers: [
8      basemapGr
9    ]
10 });
11
12 let layerControl = L.control.layers({
13   "BasemapAT Grau": basemapGr,
14   "BasemapAT Orthofoto": L.tileLayer.provider("BasemapAT.orthofoto"),
15   "BasemapAT Oberfläche": L.tileLayer.provider("BasemapAT.surface"),
16   "BasemapAT Overlay": L.tileLayer.provider("BasemapAT.overlay"),
17   "Basemap Overlay+Ortho": L.layerGroup([
18     L.tileLayer.provider("BasemapAT.orthofoto"),
19     L.tileLayer.provider("BasemapAT.overlay")
20   ])
21 }).addTo(map);
```

Abbildung 13: Einstellungen für die unterschiedlichen Basemaps, welche in jeder Karte dargestellt werden.

Csvtest.html & csv_to_json.js: (bei Bereinigung des Repositorys gelöscht, da nicht benötigt für die erstellten .html Seiten.) Versuchseiten, ob die Umwandlung von csv zu json (Papa-parse) funktioniert, die Codes wurden in die hydro.js integriert.

2.2. Arbeitsaufteilung/ Zeitmanagment

30.05. Maxi (Grundgerüst aller HTMLs erstellt) – 7 Stunden

04.06. Anna (Dokumentation) – 5 Stunden

07.06.; 08.06. & 09.06. Maxi (Versuch, alle benötigten Daten einzubinden, erste Erstellung der JS-Codes) & Anna (umwandeln der Daten in richtiges Dateiformat; Hilfestellung für Maxi beim Code schreiben) – 12 bzw. 8 Stunden jeweils

10.06. Maxi & Anna (Fertigstellung und richtige Einbindung der Daten; Ausbau der Texte auf den Websites) – min. 8 Stunden

14.06. Anna (Vereinfachung der Überflutungsflächen, damit sich die Datei hinsichtlich ihres Speicherplatzes in das Github-Repo pushen lässt) – 2 Stunden

14.06. Maxi (Einbinden der vereinfachten Überflutungsflächen und der Wetterstationsdaten aus dem aws-tirol Beispiel – pushen der fertigen Codes und Dateien) – 5 Stunden

15.06. Anna (Projektbericht fertig stellen, Repository bereinigen) – 7 Stunden

2.3. Probleme und Komplikationen

- Datei der Wasserstandsdaten nur im CSV-Format verfügbar und musste in JSON umgewandelt werden. Dies stellte uns Programmieranfänger vor eine riesige Herausforderung, welche nach unzähligen Stunden des Probierens gelöst werden konnte. In diesem Zuge wurden die jetzt gelöschten `csvtest.html` und `csv_to_json.js` Dateien erstellt und ausgearbeitet. Das Problem konnte durch einen Papaparse, also ein Umschreiben der Live-Daten (sog. Parsen), behoben werden. So konnte der aktuelle Wasserstand in Echtzeit eingebunden werden.
- Vereinfachung (*Simplify*) der Überflutungsflächen in QGIS, um weniger Speicherplatz zu verbrauchen, da die Dateien nicht mehr *gepusht* werden konnte.
- Da uns nicht bewusst war, wie sich die aufgetretenen Befehle beheben lassen, wurde zwischenzeitlich das gesamte Repository gelöscht, da sich darauf auch einige sehr fehlerhafte Dateien befunden haben. Dadurch sind die einzelnen *Commits* verloren gegangen, was ein komplett strukturiertes Nachvollziehen der einzelnen Arbeitsschritte im Nachhinein leider nicht mehr zulässt. Weiters konnte zwischenzeitlich, wie obig schon erwähnt, nicht *gepusht* werden, was vermutlich auch einiges *Commits* „verschwinden“ ließ.

- Geographisches Koordinatensystem wählen, da die Fließgewässerdaten und Überflutungsflächen jeweils in projizierten Koordinatensystemen dargestellt waren. Zuerst konnten jene nicht in den Karten dargestellt werden. Um sie dennoch darstellen zu können wurde der Raumbezug dieser in QGIS auf WGS84, ein geographisches Koordinatensystem überschrieben.

3. Ergebnis/Fazit

Es konnten alle von uns im Konzept ausgearbeiteten Punkte eingebaut und visualisiert werden. Es standen hierfür genügend Open-Source Daten zur Verfügung, um dieses Ziel zu erreichen. Somit wurden insgesamt vier HTML-Seiten mit den dazugehörigen JavaScripts und den benötigten style.css und color.js für die Formatierung der Websites erstellt. Was in den jeweiligen Codes zu finden ist, wurde in Kapitel 2 ausgiebig und strukturiert erläutert. Alle im vergangenen Semester erlernten Programmier-/Ausführungssprachen konnten verfestigt und nochmals selber angewandt werden.

Hierbei entstanden folgende 4 Websites – welche unter den jeweiligen Links aufgerufen werden können:

index.html - <https://hydrotirol.github.io/index.html>

rp.html - <https://hydrotirol.github.io/rp.html>

hydro.html - <https://hydrotirol.github.io/hydro.html>

flussportraits.html - <https://hydrotirol.github.io/flussportraits.html>

Quellen

Bundesministerium für Landwirtschaft, Regionen und Tourismus: eHyd. <https://ehyd.gv.at/> (zuletzt abgerufen am 21.05.2021).

Bundesministerium für Landwirtschaft, Regionen und Tourismus: Leben mit Naturgefahren. <https://www.naturgefahren.at/karten.html> (zuletzt abgerufen am 21.05.2021).

Bundesministerium für Land- und Forstwirtschaft, Umwelt- und Wasserwirtschaft (2016): Technische Richtlinien für die Gefahrenzonenplanung gemäß § 42a WRG (Fassung 1. Juli 2016). BMLFUW, Wien.

Fohrer, N. (Hrsg.); Bormann, H.; Miegel, K.; Casper, M.; Bronster, A.; Schumann, A. & Weiler, M. (2016): Hydrologie. Haupt, Bern.

Muhar, S.; Muhar, A.; Egger, G. & Siegrist, D. (Hrsg.) (2019): Flüsse der Alpen. Vielfalt in Natur und Kultur. Haupt, Bern.

Wikipedia: wikipedia.org (zuletzt abgerufen am 10.06.2021).

Bildquellen:

Pixabay: pixabay.com (zuletzt abgerufen am 10.06.2021).

Eigene Bilder

Datenquellen:

Überflutungsflächen: <https://data-tiris.opendata.arcgis.com/datasets/gefahrenzonen-wasser/explore>

Flussportraits – ausgeschnitten/vereinfacht aus: <https://data-tiris.opendata.arcgis.com/datasets/fliessgewaesser-1>

Hydrologische Daten: https://wiski.tirol.gv.at/hydro/ogd/OGD_W.csv

Landesgrenzen Tirol: <https://nextcloud.bev.gv.at/nextcloud/index.php/s/4rpARdQJQZZKc5m>

Wetterstationsdaten: <https://wiski.tirol.gv.at/lawine/produkte/ogd.geojson>