

Projektbericht

Anwendung von Airborne GPX Tracking

Abgabe bei
BSc. Klaus Förster
Mag. Bernhard Öggl

Maximilian Kranabetter – 01417549
Laura Mainetti – 01317471
Emanuel Pixner – 01418232



Leopold Franzens Universität Innsbruck
Institut für Geographie
Innsbruck, am 13.6.2018

Inhaltsverzeichnis

1.	<u>THEMENAUSWAHL</u>	1
2.	<u>VORBEREITUNG</u>	1
3.	<u>IMPLEMENTIERUNG</u>	2
3.1.	<u>INDEX.HTML</u>	2
3.2.	<u>INFO.HTML</u>	3
3.3.	<u>MAP.HTML</u>	4
3.4.	<u>CSS STYLESHEET</u>	5
3.5.	<u>DIE VERWENDETEN DATEN</u>	5
3.6.	<u>JAVASCRIPT</u>	6
3.7.	<u>PLUGINS</u>	9
4.	<u>ERGEBNIS</u>	11
5.	<u>PROBLEME</u>	13
6.	<u>LITERATURVERZEICHNIS</u>	15

1. Themenauswahl

Um das Projekt durchführen zu können, wurde zunächst die Verfügbarkeit von Daten zu den gewünschten Ideen überprüft. Zur Debatte stand die Anwendung von GNSS-Tracking zur Ortung von Haustieren wie Katzen oder Wildtieren um aus den Bewegungsdaten, Naturkatastrophen Vorher zu sagen (ICARUS 2018). Das zweite Thema beinhaltete die Anwendung von Airborne GPX Tracking am Beispiel von Paragleit-Flügen in Tirol. Da hierfür mehr Daten gefunden werden konnten und diese frei verfügbar sind, wurde dieses Thema gewählt.

2. Vorbereitung

In einem ersten Schritt wurde eine Organisation auf dem Versionsportal Github angelegt mit dem Namen lauraMaxPxi. Es wurde ein repository (Repo) mit Namen lauraMaxPxi.github.io erstellt, auf das alle Gruppenmitglieder Bearbeitungsrechte hatten. Da bisher noch nie von mehreren Leuten gleichzeitig an einem Repo gearbeitet wurde, musste eine Lösung für etwaige Änderungen des Inhalts gefunden werden.

Dabei wurde eine Anleitung (vgl. Abb. 1) auf der Website git (git o.J.) gefunden, nach der in weiterer Folge vorgegangen wurde. Allerdings ergaben sich dabei ein paar Schwierigkeiten, beispielsweise konnten trotz der Ausführung der Befehle `git fetch` und `git merge` die bearbeiteten Dateien nicht auf das Repo gepusht werden. Durch den Befehl `allow unrelated histories` konnte das Problem großteils behoben werden. Allerdings scheint diese Lösung nicht ideal.



Abb. 1: Arbeitsablauf beim gemeinsamen Arbeiten in einem Repo (git o.J.).

Anschließend wurde nach opensource Material gesucht, um die Projektwebsite zu befüllen. Dabei wurden Flugdaten von XContest (2018a) herangezogen sowie Bilddaten und Textdaten von diversen Quellen, die in Kapitel 3 in der chronologischen Reihenfolge ihres Auftretens genannt werden.

3. Implementierung

In einem ersten Schritt wurden die drei Html-Seiten erstellt: Index.html, Info.html und Map.html. Dabei wurde parallel ein CSS Stylesheet erstellt. Anschließend wurden die Positionsdateien der Flüge in das benötigte Format gebracht und über ein Javascript in die Kartenapplikation geladen. Abschließend wurden die drei Plugins installiert.

3.1. Index.html

Zunächst wurde das index.html-File für unser Projekt erstellt und als Grundlage hierfür diente die biketirol-Index.html-Seite von Emanuel Pixner. Als erstes wurden der Titel und die wichtigsten Überschriften hinzugefügt und alle Teile des Skriptes auskommentiert, die vorerst nicht gebraucht wurden.

Als nächstes wurde ein grober Entwurf gezeichnet, wie wir uns die Karte vorstellten, auf ein Blatt Papier gezeichnet. Die Texte in das html-File einzutragen und die Links (auf tirol.at und die anderen beiden Webseiten) gestaltete sich als unkompliziert.

Die Bilder wurde von Pixis Facebook-Account heruntergeladen, danach wurden die Bilder von ursprünglich > 2000 Pixel auf die halbe Größe verkleinert, um Speicherplatz zu sparen, und hochgeladen. Um die Webseite noch ansprechender zu machen, entschlossen wir uns, die Breite des Bildes, wie es im Browser angezeigt wird auf width="500" zu reduzieren.

Das Header-Bild wurde von vornherein auf 1280x480 Pixel verkleinert, was der Größe der Header-Bilder in den Biketirol-Webseiten entspricht.

In den bisherigen Workloads wurde meistens der Link auf das Github-Repo und der Link zur E-Mail-Adresse unter dem Author-Image angelegt. Da wir drei Teammitglieder sind, haben wir uns dazu entschlossen, nur den Link auf das Github-Repo oben anzuzeigen und am unteren Ende der Homepage, im *Footer*, ein Impressum anzulegen. Als Style verwendeten wir hier dieselbe Vorlage wie im „Author“ (in diesem Fall verschachtelt in den „Footer“). Um den Text (unsere Namen) direkt rechts neben den Bildern anzuzeigen verwendeten wir „display:inline“. Diesen Tip haben wir der Seite „stackoverflow“ zu verdanken.

Davor hatten wir bereits mehrere Befehle ausprobiert. Als „Versuchsobjekt“ diente dabei die erste Webseite von Max (die Webseite über das EGEA-French Weekend im OLAT-Ordner WL1 Upload).

Zum Schluss wurden noch einige Fehler ausgebessert, die auskommentierten Programmzeilen, die noch vom biketirol-File übrig sind, müssen noch gelöscht werden.

3.2.Info.html

Feature Lufträume:

Ein zusätzliches Feature wäre noch die Einbindung von Lufträumen in die Kartendarstellung der Website. Die Daten für die Lufträume sind auf der Website von AustroControl erhältlich in mehreren Formaten (kml, OpenAir, txt). Die Luftraumstruktur von Österreich ist auf der Infoseite im PDF-Format eingebunden. Die Einbindung auf die Website erfolgte gleich wie mit den Bildern auf der Info-Seite.

Die Einbindung der Lufträume in die Karte hat nicht wie gewünscht funktioniert, weshalb das Feature „Lufträume“ weggelassen wurde.

Infotext:

Die Infos über den Innsbrucker Gleitschirmverein und hilfreiche Informationen zum Umgang mit den Lufträumen als nichtmotorisierte Pilot sind auf der Homepage der IGV entnommen.

<http://www.innsbruckergleitschirmfliegerverein.org/index.php/fluggebiete/ctr>

Feature Bilder:

Die Bilder auf der Infoseite wurden von XContest eingebunden. Im Quellcode der XContest-Homepage wurde die Adresse der Bilder gesucht. Die Adresse wurde kopiert und in die eigene Seite mit der Image-Funktion eingebunden. Ein erweitertes Feature für die erstellten Websites wäre die Darstellung von Fotos der ausgewählten Flugtracks. Diese sind aber nicht immer vorhanden, weshalb diese Variante gewählt wurde.

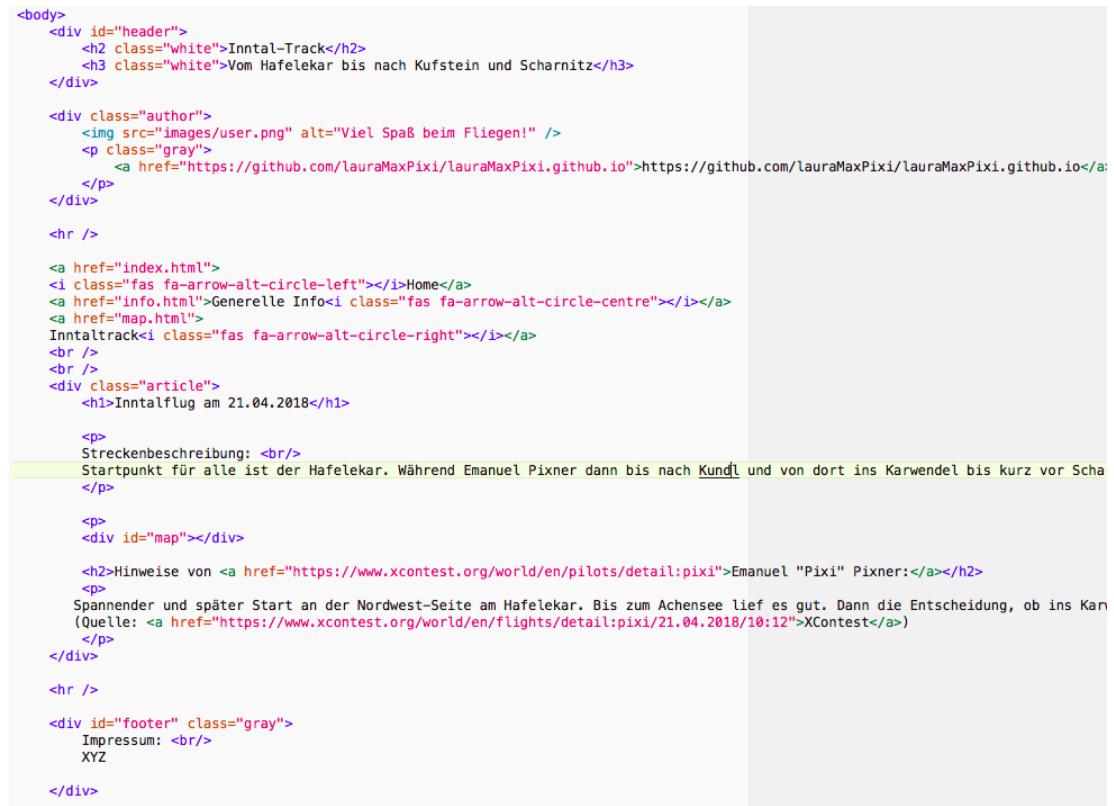
```

```

Abb. 2: Beispiel für die Einbindung eines Fotos aus XContest (eigene Darstellung).

3.3.Map.html

Als Grundlage wurde auch hier das index.html des biketirol Beispiels herangezogen. Als Textquelle diente die Beschreibung des Fluges von Emanuel Pixner auf XContest (2018b) herangezogen. Die Gesamtbeschreibung wurde selbst verfasst anhand der vorhandenen Flugdaten von Emanuel Pixner, Reinhold Korak, Siegfried Steixner und Armin Eder (Daten und Beschreibung ebenfalls erhältlich auf XContest 2018b) (vgl. Abb. 3).



```
<body>
  <div id="header">
    <h2 class="white">Inntal-Track</h2>
    <h3 class="white">Vom Hafelekar bis nach Kufstein und Scharnitz</h3>
  </div>

  <div class="author">
    
    <p class="gray">
      <a href="https://github.com/lauraMaxPixel/lauraMaxPixel.github.io">https://github.com/lauraMaxPixel/lauraMaxPixel.github.io</a>
    </p>
  </div>

  <hr />

  <a href="index.html">
    <i class="fas fa-arrow-alt-circle-left"></i>Home</a>
  <a href="info.html">Generelle Info<i class="fas fa-arrow-alt-circle-centre"></i></a>
  <a href="map.html">
    Inntaltrack<i class="fas fa-arrow-alt-circle-right"></i></a>
  <br />
  <br />
  <div class="article">
    <h1>Inntalflug am 21.04.2018</h1>

    <p>
      Streckenbeschreibung: <br/>
      Startpunkt für alle ist der Hafelekar. Während Emanuel Pixner dann bis nach Kufstein und von dort ins Karwendel bis kurz vor Schachl
    </p>

    <p>
      <div id="map"></div>
      <h2>Hinweise von <a href="https://www.xcontest.org/world/en/pilots/detail:pixel">Emanuel "Pixel" Pixner:</a></h2>
      <p>
        Spannender und später Start an der Nordwest-Seite am Hafelekar. Bis zum Achensee lief es gut. Dann die Entscheidung, ob ins Karwendel oder direkt nach Kufstein. (Quelle: <a href="https://www.xcontest.org/world/en/flights/detail:pixel/21.04.2018/10:12">XContest</a>)
      </p>
    </div>
  <hr />
  <div id="footer" class="gray">
    Impressum: <br/>
    XYZ
  </div>
  </div>
```

Abb. 3: Text-Editor Ansicht des map.html Files (eigene Darstellung).

Anschließend wurde der Bereich unterhalb des *headers* angepasst, damit die Verlinkung zur Hauptseite (index.html) und Information (info.html) gegeben war und zwar im gleichen Stil wie bei den anderen beiden Html-Seiten (vgl. Abb. 4).

```
47      <div id="footer" class="gray">
48        <a class="back" href="index.html">
49          <i class="fas fa-arrow-alt-circle-left"></i>Hauptseite</a>
50        <a class="next" href="info.html">Zusatzinfo
51          <i class="fas fa-arrow-alt-circle-right"></i>
52        </a>
53      </div>
```

Abb. 4: Verlinkung des map.html Files mit den anderen beiden Html-Seiten (eigene Darstellung).

Das Impressum wurde ebenfalls vom index.html File übernommen (vgl. Abb. 5).

```

78 <div id="footer" class="gray">
79   <h2>Impressum</h2>
80   <br/><br/>
81   <div class="author">
82     <img src='./images/laura.jpg' style='display:inline;'/>
83     <p style='display:inline;'> Laura Mainetti – E-Mail: <a href="mailto:laura.mainetti@student.uibk.ac.at">laura.mainetti@st
84     <br/><br/>
85     <img src='./images/pixi.jpg' style='display:inline;'/>
86     <p style='display:inline;'> Emanuel "Pixi" Pixner – E-Mail: <a href="mailto:emanuel.pixner@student.uibk.ac.at">emanuel.pi
87     <br/><br/>
88     <img src='./images/max.jpg' style='display:inline;'/>
89     <p style='display:inline;'> Max Kranabetter – E-Mail: <a href="mailto:maximilian.kranabetter@student.uibk.ac.at">maximili
90   </div>
91 </div>

```

Abb. 5: Impressum, übernommen aus dem index.html File (eigene Darstellung).

Für das Einbinden der Kartenapplikation wurde ein div mit der id „map“ angelegt. Ebenso wurde das zugehörige Javascript im head-Bereich des html-Files als Script hineingeladen (vgl. Abb. 6).

```

66       <p>
67         <div id="map"></div>
68
24       <script defer src="igcView.js"></script>

```

Abb. 6: Einbinden der Kartenapplikation und des zugehörigen Javascripts (eigene Darstellung).

3.4.CSS Stylesheet

Für das CSS Stylesheet wurde als Vorlage das des biketirol Beispiels verwendet.

3.5.Die verwendeten Daten

Die Trackdaten der vier verschiedenen Flüge sind auf XContest (abrufbar, entweder als .igc oder als .kmz Datei. Über den Converter GPS Visualizer (2018) können die .kmz files in .gpx Format gebracht werden (vgl. Abb. 7). Mit dem Onlinetool von Mapbox (o.J.) können die Daten dann in GeoJSON Format gebracht werden (vgl. Abb. 8).

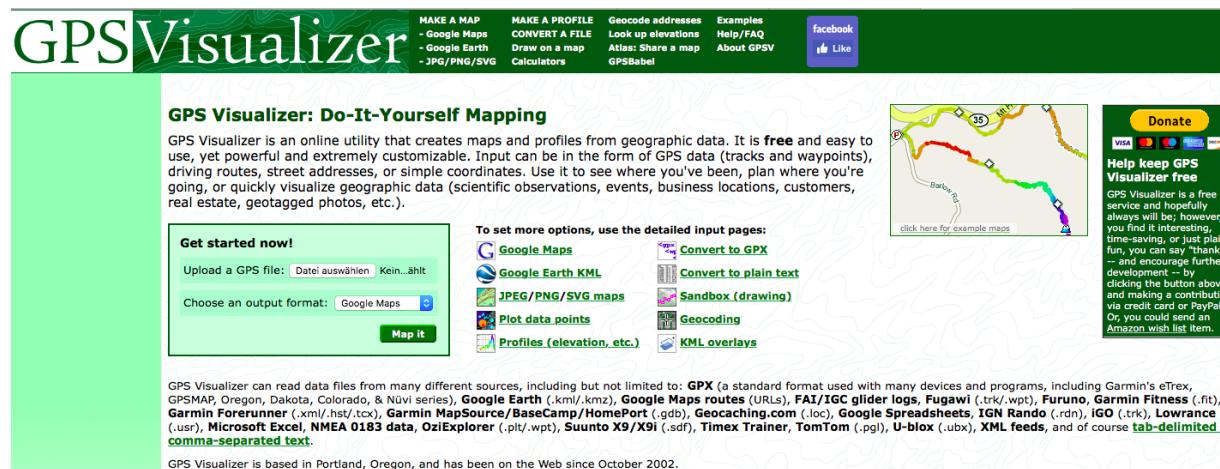


Abb. 7: Upload-Seite des GPS Visualizer Converters (GPS Visualizer 2018).

toGeoJSON turns KML & GPX into GeoJSON

The screenshot shows the 'toGeoJSON' converter tool. On the left, under the 'GPX' tab, there is a text area containing GPX XML code. On the right, under the 'GeoJSON' tab, there is a text area showing the converted GeoJSON object. Below the tabs is a blue button with the text '↓ togeojson.js less than 10 kB minified'.

Abb. 8: toGeoJSON Converter von Mapbox (o.J.).

Anschließend in den Texteditor der Wahl kopieren und als .json speichern. Von da an können die Daten weiter verwendet werden.

3.6.Javascript

Anschließend wurde versucht, die Grundkarte für map.html, nämlich im igcView.js herzustellen. Als Grundkartenlayer wurden Layer von Openstreetmap (o.J.) und Basemap.at (2016) herangezogen (vgl. Abb. 9).

```

11 const myLayers = {
12   osm: L.tileLayer("https://s.tile.openstreetmap.org/{z}/{x}/{y}.png", {
13     subdomains: ["a", "b", "c"],
14     attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
15   })
16 },
17 geolandbasemap: L.tileLayer(
18   "https://s.wien.gv.at/basemap/geolandbasemap/normal/google3857/{z}/{y}/{x}.png", {
19     subdomains: ["maps", "maps1", "maps2", "maps3", "maps4"],
20     attribution: "Datenquelle: <a href='https://www.basemap.at'>basemap.at</a>"
21   }
22 ),
23 bmaporthofoto30cm: L.tileLayer(
24   "https://s.wien.gv.at/basemap/bmaporthofoto30cm/normal/google3857/{z}/{y}/{x}.jpeg", {
25     subdomains: ["maps", "maps1", "maps2", "maps3", "maps4"],
26     attribution: "Datenquelle: <a href='https://www.basemap.at'>basemap.at</a>"
27   }
28 ),
29 );
30
31 myMap.addLayer(myLayers.geolandbasemap);

```

Abb. 9: Einrichtung der Grundkartenlayer (eigene Darstellung).

Anschließend wurde eine entsprechende Layercontrol gebaut sowie ein metrischer Maßstab (Abb. 10).

```

35  let myMapControl = L.control.layers({
36
37     "Openstreetmap": myLayers.osm,
38     "Basemap": myLayers.geolandbasemap,
39     "basemap.at Orthofoto": myLayers.bmaporthofoto30cm,
40   },
41   {
42     "Tracks": trackGroup,
43     "Start und Ziel": markerGroup,
44     "Steigung": steigung,
45   });
46
47   myMap.addControl(myMapControl);

```

```

89   myMapControl.expand();
90
91   L.control.scale({
92     maxWidth: 200,
93     metric: true,
94     imperial: false,
95     position: "bottomleft"
96
97 }).addTo(myMap);

```

Abb. 10: Einrichten der Layercontrol und des metrischen Maßstabes (eigene Darstellung).

Anschließend müssen die Trackdaten eingefügt werden, die zuvor über Mapbox (o.J.) in .json Format konvertiert wurden:

In einem nächsten Schritt werden die so erzeugten GeoJSON Daten als Javascript Variable definiert. Dadurch können sie im map.html eingebunden werden (vgl. Abb. 11).

```

16
17   <script src="data/ebner.json"></script>
18   <script src="data/steixner.json"></script>
19   <script src="data/korak.json"></script>
20   <!!--<script src="data/pixner.json"></script>
21

```

Abb. 11: Einbinden der Trackdaten in das map.html File (eigene Darstellung).

Dann können sie im igcView.js als Tracks eingebunden werden (vgl. Abb. 12).

```

82
83   let ebnerTrack = L.geoJSON(ebnerData, {
84     color: "blue"
85   }).addTo(ebnerGroup);
86
87   let korakTrack = L.geoJSON(korakData, {
88     color: "red"
89   }).addTo(korakGroup);
90
91   let steixnerTrack = L.geoJSON(steixnerData, {
92     color: "yellow"
93   }).addTo(steixnerGroup);
94
95   /*
96   let pixnerTrack = L.geoJSON(pixnerData, {
97     color: "green"
98   }).addTo(pixnerGroup);
99   */
100
101  myMap.fitBounds(ebnerGroup.getBounds());
102

```

Abb. 12: Einbinden der Tracks in das igcView.js File (eigene Darstellung).

Dadurch können die Tracks nun in der Kartenapplikation eingesehen werden (vgl. Abb. 13).

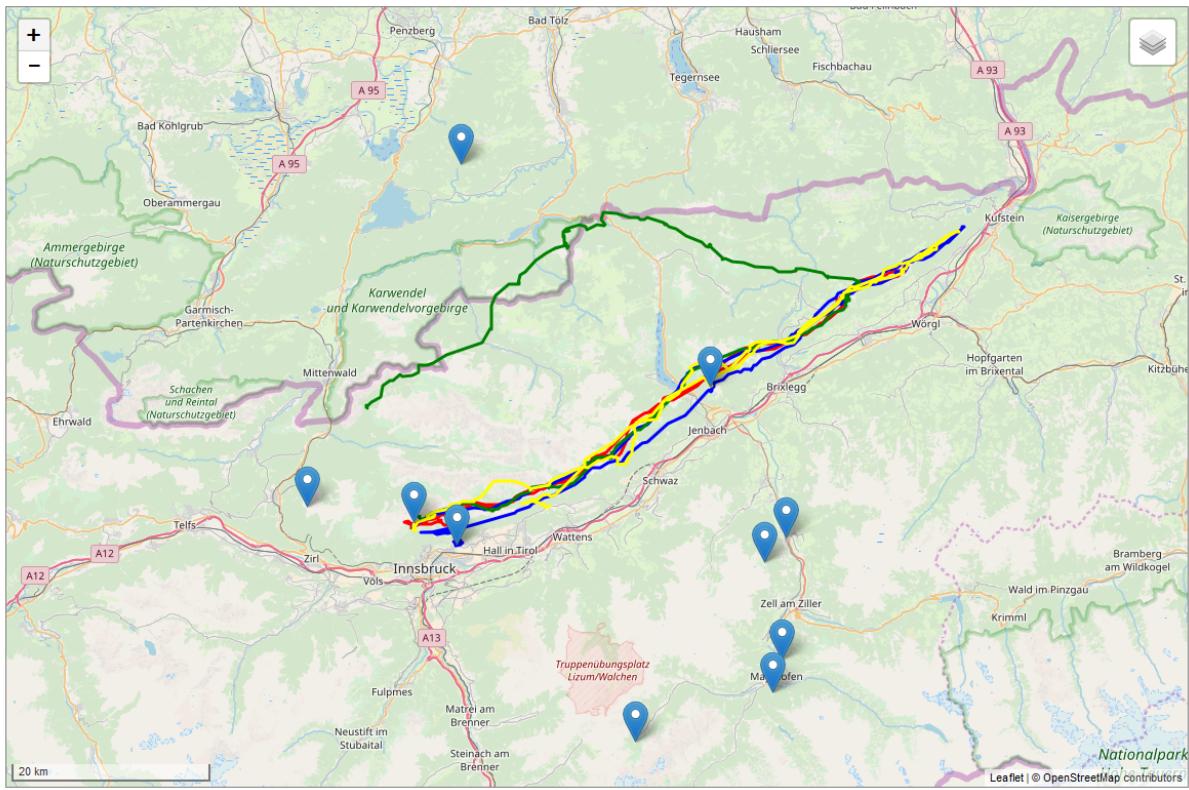


Abb. 13: Darstellung der Tracks in der Kartenapplikation (eigene Darstellung).

Thermik-Hotspots:

Die Thermik-Hotspots der Website Thermalkk7 (2018) wurden heruntergeladen und wie bereits zuvor beschrieben von gpx-Format in Geojson-Format umgewandelt. Diese Datei sollte als zusätzlicher Layer in der Layercontrol auswählbar sein (Abb. 14). Dies hat leider nicht wie gewünscht funktioniert, Wehalb das Feature weggelassen wurde.

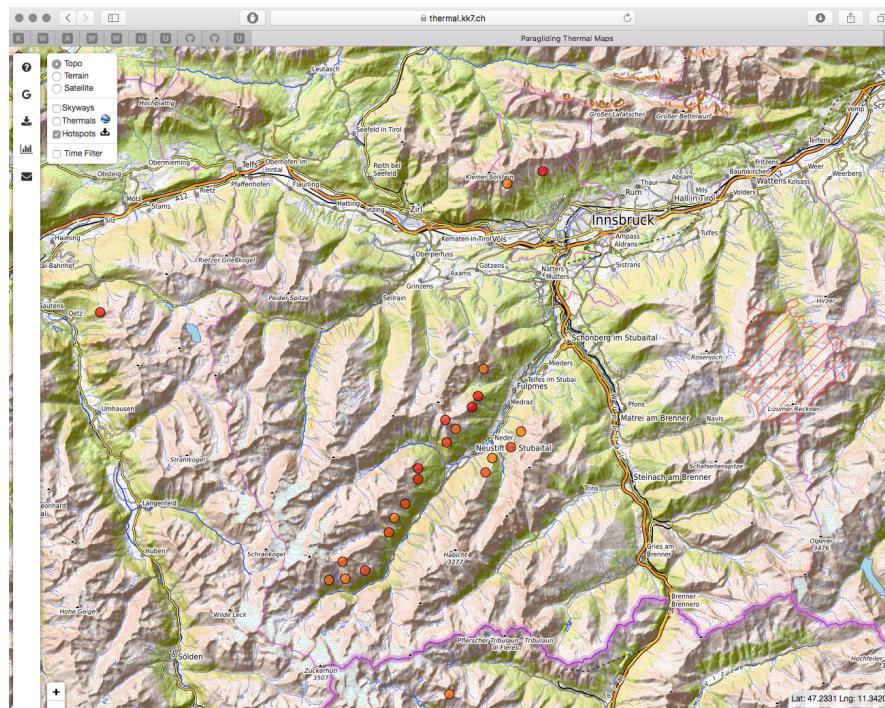


Abb. 14: Beispiel einer Darstellung der Thermik Hotspots (thermalkk7.ch).

3.7.Plugins

Es wurden drei Plugins ausgewählt, die in die Kartenapplikation integriert werden sollten. Es wurden ein *Fullscreen* Plugin (Leafletjs 2016) sowie ein Höhenprofil Plugin (MrMufflon 2016) und ein Hash Plugin (mlevans 2013) ausgewählt.

Einige der Plugins waren noch aus dem Biketirol Beispiel vorhanden, die konnten 1:1 übernommen werden (*Fullscreen Control* als Beispiel). Dafür wurde das gesamte Unterverzeichnis js/ in das Organisationsrepo hochgeladen. Von dort aus konnte es verwendet werden:

Zunächst wurde das **Fullscreen** Plugin eingebaut. Dazu muss lediglich der Code aus Abbildung 15 in das map.html File kopiert werden und die Einstellung fullscreenControl im igcView.js auf true gestellt werden (vgl. Abb. 16).

```
26      <script src="js/leaflet.fullscreen/Leaflet.fullscreen.js"></script>
27      <link href="js/leaflet.fullscreen/leaflet.fullscreen.css" rel='stylesheet' />
```

Abb. 15: Einbinden des Leaflet.fullscreen.js und .css Files im map.html (eigene Darstellung).

```
1  let myMap = L.map("map", {
2    fullscreenControl: true,
3  });
```

Abb. 16: Einbinden der Fullscreen Control über die Einstellung true im igcView.js (eigene Darstellung).

Für das Einbinden des **Höhenprofil** Plugins müssen die Trackdaten schon eingebunden sein, da sie sich auf die Tracks beziehen. Dann müssen die zugehörigen Scripts und Stylesheets im map.html File eingebunden werden (vgl. Abb. 17). Abschließend wird noch eine entsprechende Control gebaut, um das Höhenprofil ein- und ausschalten zu können (vgl. Abb. 18).

```
29      <!--Stylesheet und plugin-script für elevation anzeigen in karte -->
30      <script src="js/d3/d3.v3.min.js" charset="utf-8"></script>
31      <script src="js/leaflet.elevation/leaflet.elevation-0.0.4.min.js" charset="utf-8"></script>
32      <link rel="stylesheet" href="js/leaflet.elevation/leaflet.elevation-0.0.4.css" />
```

Abb. 17: Einbinden des Höhenprofil Plugins im map.html File (eigene Darstellung).

```

134 //Pixi's solution
135 let heightProfile = L.control.elevation({
136   position: "bottomright",
137   theme: "steelblue-theme", //default: lime-theme
138   width: 600,
139   height: 140,
140   margins: {
141     top: 10,
142     right: 20,
143     bottom: 30,
144     left: 50
145   },
146   useHeightIndicator: true, //if false a marker is drawn at map position
147   interpolation: "linear", //see https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_interpolate
148   hoverNumber: {
149     decimalsX: 3, //decimals on distance (always in km)
150     decimalsY: 0, //decimals on heighthttps://www.npmjs.com/package/leaflet.coordinatesight (always in m)
151     formatter: undefined //custom formatter function may be injected
152   },
153   xTicks: undefined, //number of ticks in x axis, calculated by default according to width
154   yTicks: undefined, //number of ticks on y axis, calculated by default according to height
155   collapsed: true, //collapsed mode, show chart on click or mouseover
156   imperial: false //display imperial units instead of metric
157 });
158 heightProfile.addTo(myMap);

```

Abb. 18: Einbauen einer Höhenprofil Control (eigene Darstellung).

Weiter wird das Plugin im Javascript folgendermaßen eingebaut: Es muss eine neue Variable angelegt werden, die sich auf die Daten des Tracks bezieht, also auf das zugehörige Javascript, und dann an das Höhenprofil gebunden wird (vgl. Abb. 19).

```

161 let pixnerEle = L.geoJSON(pixnerData, {
162   onEachFeature: heightProfile.addData.bind(heightProfile)
163 }).addTo(myMap);

```

Abb. 19: Anlegen einer neuen Variable und anbinden an das Höhenprofil (eigene Darstellung).

Abschließend wurde noch das Hash Plugin in das Projekt eingebaut. Dafür mussten die zugehörigen Scripts und Stylesheets im map.html eingebunden werden (vgl. Abb. 20) und eine neue Variable im Javascript angelegt werden (vgl. Abb. 21).

```

51
52 <script src="js/leaflet.hash/leaflet-hash.js"></script>
--
```

Abb. 20: Einbinden des Hash Plugins in das map.html File (eigene Darstellung).

```

--  
7 let hash = new L.Hash(myMap);
```

Abb. 21: Anlegen einer neuen Variable im Javascript igcView.js (eigene Darstellung).

4. Ergebnis

Ergebnis sind drei Html-Seiten, die miteinander verlinkt sind und über die URL <https://laura-maxpixi.github.io/index.html> eingesehen werden können. Im folgenden sind die drei Seiten Index.html (Abb. 22), Info.html (Abb. 23) und Map.html (Abb. 24) dargestellt.

Das Index.html dient dabei als Hauptseite, auf der das Paragleiten in Tirol thematisiert wird sowie auch ein Überblick über den Sport des Paragleitens an sich geliefert wird. Es wird ein Beispiel genannt. Der untere Seitenbereich zeigt das Impressum, im oberen Bereich unter dem Titelbild sind die anderen beiden Seiten verlinkt.

Index.html

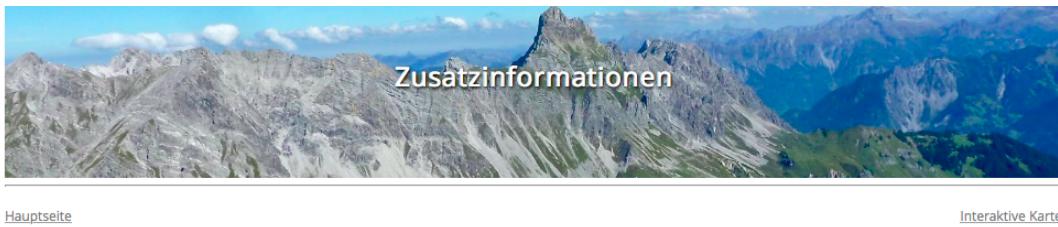


Abb. 22: Die Hauptseite des Projekts „Anwendung vom Airborne GPX Tracking“ ist das Index.html. Von hier gelangt man auf die Infoseite und die interaktive Karte (eigene Darstellung).

Im Info.html wird das Fluggebiet Innsbruck Umgebung vorgestellt. Dabei wird zunächst genauer auf das Gebiet Seegrube und Nordkette eingegangen. Es folgt eine Beschreibung der Lufträume in Innsbruck sowie eine Erklärung des Begriffs „Streckenfliegen“. Abschließend zeigt eine Fotoserie Aufnahmen von einem Flug.

Das Map.html zeigt abschließend vier Tracks von Piloten, die am 21.4.2018 vom Hafelekar aus gestartet sind und entweder bis nach Kufstein und wieder retour, oder über das Karwendel Richtung Scharnitz geflogen sind. Zudem wird eine Streckenbeschreibung geliefert sowie ein Kommentar zu diesem Flug vorgestellt.

Info.html



[Hauptseite](#)

[Interaktive Karte](#)

Fluggebiet Innsbruck Umgebung

Fluggebiet Seegrube an der Nordkette

Die Seegrube ist das Hauptfluggebiet der Innsbrucker Gleitschirmflieger. Es lockt nicht nur mit traumhaften Panoramablicken weit über viele Täler hinweg, sondern hat zudem noch ein großes Streckenflugpotential, da die Bergflanken größtenteils OST-WEST ausgerichtet sind. Ob der Flug Richtung Achensee und dann weiter ins Zillertal gehen soll oder Richtung Westen ins Oberland, ist egal. Die fliegerischen Möglichkeiten sind groß und die Ausblicke wirklich beeindruckend. Wer schon einmal in diesem Gebiet einen Blick aus luftigen Höhen erhaschen konnte, der fliegt hier öfters. Mittlerweile wurde die neue Nordkettenbahn auch mit der neu gebauten Hungerburgbahn komplettiert, was jedem Piloten seit Dezember 2007 die Möglichkeit gibt, wieder wie gewohnt in nur wenigen Minuten direkt zur Seegrube zu kommen. Weitere Informationen zur Seegrube als Startplatz findest du [hier](#).

(Quelle: [IGV - Innsbrucker Gleitschirmfliegerverein](#))

Lufträume in Innsbruck

Die auf AustroControl bereitgestellten Luftfahrtdata unterliegen dem AIRAC System. Das bedeutet, dass die Daten im Zyklus von 28 Tagen berichtigt und somit auf der Website von AustroControl (siehe unten) aktualisiert werden können. Für den Flugbetrieb wichtige Informationen, die ad hoc verlautbart werden müssen, werden mittels NOTAM ausgegeben. Weitere Informationen zu der Regelung der Lufträume findest du [hier](#). Um die aktuellste Version der Struktur Lufträume in Österreich im Format "OpenAir", "cup" oder "kml" zu erhalten, besuche die Download-Page der [AustroControl](#). Zusätzlich gibt es eine schematische Abbildung für die Übersicht der vorhandenen Luftraumstruktur. (Quelle: [AustroControl](#))



Abb. 23: Das Info.html ist die Infoseite des Projekts. Hier wird ein Überblick über das Gebiet, die Lufträume sowie das Streckenfliegen an sich geliefert (eigene Darstellung).

Map.html

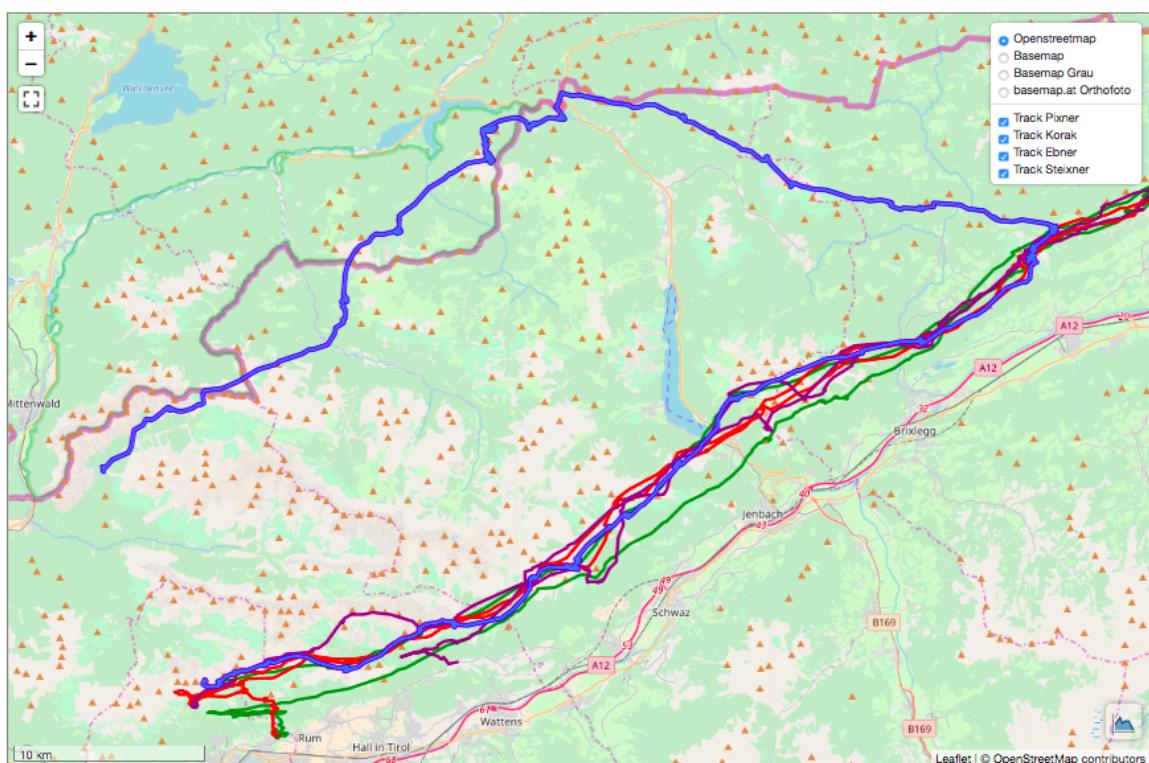


Abb. 24: Das Map.html zeigt die interaktive Kartenapplikation mit Fullscreen Plugin, Höhenprofil und verschiedenen Tracks. Hier finden sich auch eine Streckenbeschreibung und ein Kommentar zum Flug (eigene Darstellung).

5. Probleme

Trackdaten einbinden:

Beim Einbinden der Trackdaten in die Kartenapplikation sind zunächst Fehler aufgetreten: Das Verwenden der .kmz Dateien führte in weiterer Folge dazu, dass anstatt nur einer Linie pro Track mehrere Linien inklusive Marker angezeigt wurden. Dabei waren diese Linien nicht einmal deckungsgleich (vgl. Abb. 25).

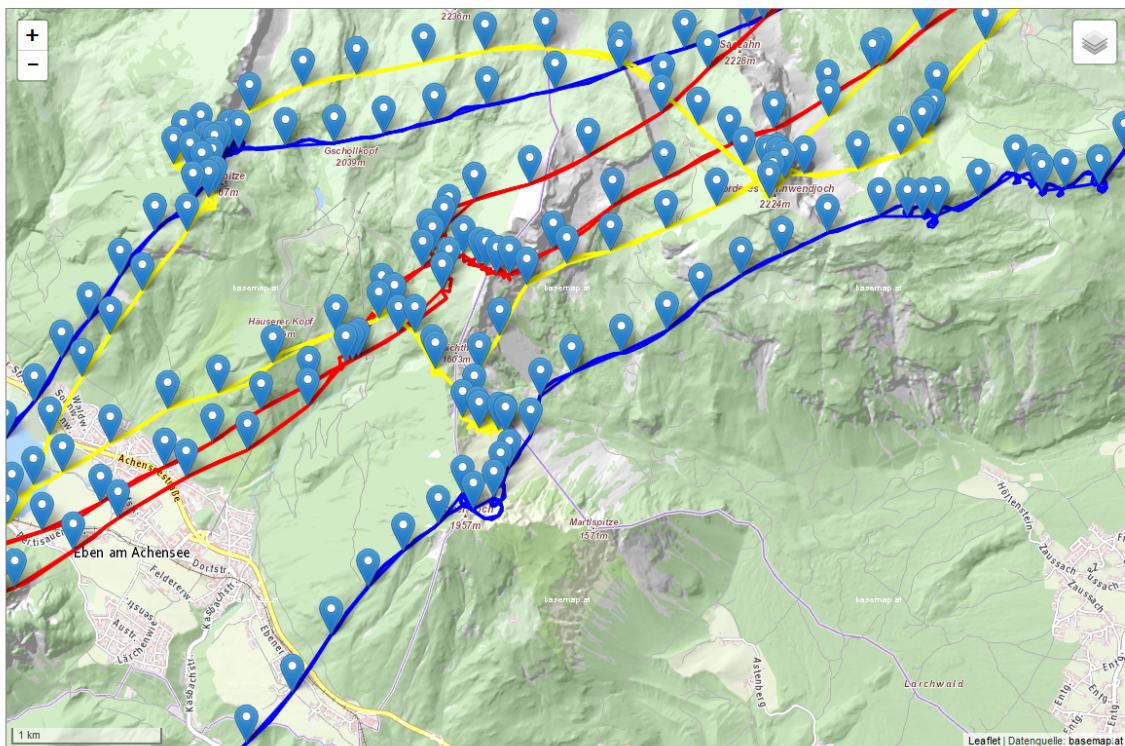


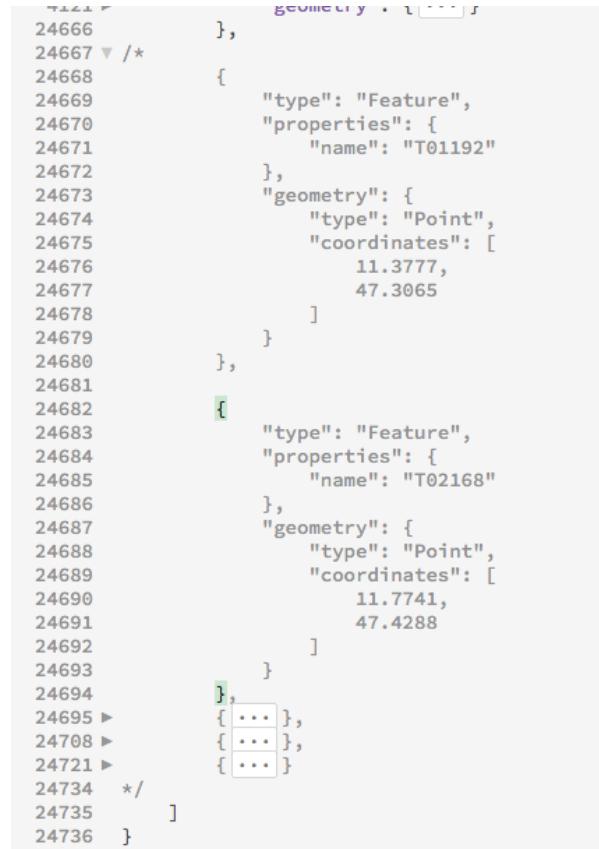
Abb. 25: Mehrfache Anzeige der Tracks mit mehreren Linien und Markern (eigene Darstellung).

Schließlich fiel auf, dass die .kmz Dateien ca. viermal so groß waren wie die .igc Dateien. Daraufhin wurden die .igc Dateien konvertiert und in die Kartenapplikation eingebunden. Dadurch wurde jeder Track nur noch als eine Linie ohne Marker dargestellt. Allerdings tauchten nun einzelne Marker in der Karte verteilt auf (vgl. Abb. 13).

Unbekannte Marker in der Karte:

In der Karte sind Marker sichtbar, welche von leaflet.css als Gruppe im Hintergrund erkannt wurden und automatisch ein Marker erstellt wurde (vgl. Abb. 26). Diese „marker pane“ sind erst im Web-Entwickler tool ersichtlich, wobei unklar war, woher die Koordinaten stammen. Erst durch das Ausblenden der Tracks ist aufgefallen, dass diese Marker in den Tracks gespeichert waren.

Durch löschen bzw. ausblenden dieser Punktkoordinaten wurde das Problem behoben. Es ist dabei nicht eindeutig klar, woher diese Punkte stammen. Vermutlich wird beim Export der Dateien aus XContest ein Bereich angegeben, der für eine uns unklare Verwendung benutzt wird.



```
24666     },
24667     /*
24668     {
24669         "type": "Feature",
24670         "properties": {
24671             "name": "T01192"
24672         },
24673         "geometry": {
24674             "type": "Point",
24675             "coordinates": [
24676                 11.3777,
24677                 47.3065
24678             ]
24679         }
24680     },
24681
24682     [
24683         {
24684             "type": "Feature",
24685             "properties": {
24686                 "name": "T02168"
24687             },
24688             "geometry": {
24689                 "type": "Point",
24690                 "coordinates": [
24691                     11.7741,
24692                     47.4288
24693                 ]
24694             }
24695         },
24696         [ ... ],
24697         [ ... ],
24698         [ ... ]
24734     */
24735   ]
24736 }
```

Plugins:

Zusätzlich war ein großes Problem, dass wir lange Zeit kein einziges Plugin aktivieren konnten, obwohl wir es vermeintlich richtig decoded haben. Dabei besagte bei einem Plugin die Anleitung, dass ein Teil des codes in den body kopiert werden sollte, um ein .js und ein .css zu laden. Dadurch wurde alles, was zuvor im head Bereich decoded wurde, überschrieben. Folglich „wusste“ die Internetseite nicht, dass andere Scripts und Links geladen werden sollten.

Durch Ausklammern des besagten Codes konnte das Fullscreen Plugin dann aktiviert werden – voll funktionsfähig.

Generell war das Einbinden der Plugins neben der korrekten Darstellung der Tracks das Hauptproblem. Die Konsole zeigte an, dass ein Problem mit dem leaflet.js vorliegt. Was unwahrscheinlich erscheint, da dieses Script auch im Unterricht erfolgreich verwendet wurde.

6. Literaturverzeichnis

AustroControl (2018): Luftraumstruktur in Österreich. https://www.austrocontrol.at/piloten/vor_dem_flug/aim_produkte/luftraumstruktur (Zugriff am 8.6.2018).

Basemap.at (2016): Home. <https://www.basemap.at> (Zugriff am 10.6.2018).

git (o.J.): Distribuierte Arbeit mit Git (xxx) – An einem Projekt mitarbeiten. <https://git-scm.com/book/de/v1/Distribuierte-Arbeit-mit-Git-xxx-An-einem-Projekt-mitarbeiten> (Zugriff am 10.6.2018).

GPS Visualizer (2018): GPS Visualizer: Do-It-Yourself-Mapping. <http://www.gpsvisualizer.com> (Zugriff am 10.6.2018).

ICARUS (2018): Frühwarnsystem auf vier Beinen. <https://www.animal-sensors.mpg.de/5181/fruehwarnsystem-auf-vier-beinen> (Zugriff am 10.6.2018).

IGV (?): Innsbrucker Gleitschirmverein. <http://www.innsbruckergleitschirmfliegerverein.org> (Zugriff am 10.6.2018).

Leafletjs (2016): Plugins. <https://leafletjs.com/plugins.html#fullscreen-controls> (Zugriff am 10.6.2018).

Mapbox (o.J.): toGeoJSON turns KML & GPX into GeoJSON. <https://mapbox.github.io/to-geojson/> (Zugriff am 10.6.2018).

mlevans (2013): Leaflet-hash. <https://github.com/mlevans/leaflet-hash> (Zugriff am 12.6.2018).

MrMufflon (2016): Leaflet.Elevation. <https://github.com/MrMufflon/Leaflet.Elevation> (Zugriff am 10.6.2018).

Openstreetmap (o.J.): Urheberrecht und Lizenz. <https://www.openstreetmap.org/copyright> (Zugriff am 10.6.2018).

Thermalkk7 (2018): Paragliding Thermal Maps. <https://thermal.kk7.ch> (Zugriff am 12.6.2018).

XContest (2018a): World XContest 2018 [home]. <https://www.xcontest.org/world/en/> (Zugriff am 10.6.2018).

XContest (2018b): Flight detail Emanuel Pixner – 21.4.2018 – 96,13km. <https://www.xcontest.org/world/en/flights/detail:pixi/21.04.2018/10:12> (Zugriff am 9.6.2018).

XContest (2018c): Flight detail Reinhold Korak – 21.4.2018 – 114,21km. <https://www.xcontest.org/world/en/flights/detail:RK8/21.04.2018/09:38> (Zugriff am 11.6.2018).

XContest (2018d): Flight detail Siegfried Steixner – 21.4.2018 – 115,32km. <https://www.xcontest.org/world/en/flights/detail:Siggi/21.04.2018/10:07> (Zugriff am 11.6.2018).

XContest (2018e): Flight detail Armin Eder – 21.4.2018 – 126,78km. <https://www.xcontest.org/world/en/flights/detail:vvinnetou/21.04.2018/10:39> (Zugriff am 11.6.2018).