

«Mont»

Augmented-Reality-Webapp für iOS in Vanilla Javascript

Fachhochschule Graubünden (FHGR)
Departement Angewandte Zukunftstechnologien
Institut für Multimedia Production
Januar 2023

Prof. lic. phil. Thomas Weibel
Realpstrasse 55
4054 Basel
+41 79 301 99 87
thomas.weibel@fhgr.ch

Abstract

«Mont» ist eine Augmented-Reality-Webapp für iOS-/iPadOS-Geräte, die sämtliche Hügelkuppen und Berggipfel der Schweiz einschliesslich Höhe und Distanz angibt. Das Projekt beruht auf Open Data und ist in Vanilla JS geschrieben.

Stichwörter: HTML5, Javascript, VanillaJS, Webapp, Mobile, Open Source, Open Data, Open Government Data, Geodata, Geoinformatics, Topography, Switzerland

Abstract	2
Einleitung	4
Die Webapp «Mont»	6
Daten	6
Digitales Höhenmodell der Schweiz	7
Sammlung geografischer Namen der Schweiz	8
Speicherung und Abruf	8
Programmierung	9
Funktionsweise	11
Kamerastream	11
Geolokalisierung	12
Lagesensoren	14
Magnetsensor/Kompass	17
Höheninterpolation	18
Gipfeltriangulation	19
Gipfelsichtbarkeit	20
Darstellung	21
Animation	22
Fehlerbehandlung	23
Smartphone-Ausrichtung	23
Andere	24
Betriebssystem	25
Geolocation	25
Geografie	26
Kamera	27
Magnetkompass, Gyroskop	28
Eigenschaften	28
Beschränkungen	29
Weiterentwicklungsmöglichkeiten	31
Schlussfolgerungen	33
Quellenverzeichnis	34
Illustrationsverzeichnis	38
Abkürzungsverzeichnis	39

Einleitung

Die Geschichte des Web, 1989 ursprünglich als Konzept für eine zentrale Datenablage des «Cern» in Genf vorgeschlagen von Tim Berners-Lee¹, lässt sich in folgende Entwicklungsschritte gliedern:

- Strukturierte Dokumente (HTML, 1990)
- Interaktivität, Web 2.0 (serverseitige Programmierung, Perl/PHP, 1995/1995)
- Grafikdesign (CSS, 1996)
- Onlinedienste und -applikationen (clientseitige Programmierung, JS, 1996)
- Mobiloptimierung (CSS Media Queries, 2000)
- Social Media (2003)
- Erweiterte Realität (VR, AR, 2015)

Der Sammelbegriff «Erweiterte Realität» (engl. «Extended Reality», ER) steht für «Virtual Reality» (VR), «Augmented Reality» (AR) sowie «Mixed Reality» (MR) und umfasst eine Reihe von Ansätzen, deren Ziel es ist, die erlebte Realität auf geeigneten Geräten mit Daten zu überlagern und dadurch einen informativen Mehrwert zu bieten. Heute ist es möglich, mittels Javascript (JS) auf die Kamera und die Bewegungssensoren des Endgeräts zuzugreifen und so – zumindest theoretisch – handprogrammierte AR-Webapps zu erstellen. Die meisten Onlineprojekte in erweiterter Realität (VR, AR, MR) setzen bisher jedoch 3D-Software (Closed Source), Grafikengines bzw. die Nutzung einschlägiger JS-Frameworks voraus.

Die Fragestellung, die zu meinem Projekt «Mont» geführt hat, lautete daher: Ist es möglich, eine Augmented-Reality-Webapp für Smartphones zu entwickeln

- als handgeschriebene Website (HTML, Javascript),
- *ohne* Verwendung externer Libraries oder Frameworks,
- *ohne* den Einsatz von Closed-Source-Software,
- auf der Basis offener Daten (Open Government Data)?

¹ Berners-Lee, Tim (1989): Information Management: A Proposal. Genf, <https://www.w3.org/History/1989/proposal.html>, 10.11.2022.

Mein Versuch bestand darin, eine AR-Webapp in reinem (sog. «Vanilla») Javascript² zu programmieren. Im technischen Sinn ist «Mont» eine für Smartphones konzipierte Webseite (*mobile only*) mit lediglich 10 kb reinem Programmcode, die eine Kameraansicht der umgebenden Landschaft bietet, die lokale Topografie berechnet und innerhalb einer gegebenen Reichweite die Namen aller sichtbaren Hügelkuppen und Berggipfel (einschliesslich ihrer Gipfelhöhe und Entfernung) anzeigt. Zusätzlich gibt ein Kompass die aktuelle Blickrichtung an, ein Höhenmesser die Meereshöhe des aktuellen Standorts.

«Mont» ist über über einen via QR-Code³ einfach kommunizierbaren URL⁴ zugänglich und basiert auf der Sammlung geografischer Namen der schweizerischen Landesvermessung und dem digitalen Höhenmodell der Schweiz, die vom Bundesamt für Landestopografie auf der Plattform opendata.swiss im Rahmen der im November 2018 vom Bundesrat verabschiedeten «Strategie für offene Verwaltungsdaten in der Schweiz 2019-2023» als Open Government Data⁵ publiziert wurden. Alle erforderlichen Daten (räumliche Informationen des 200-m-Gitternetzes des digitalen Höhenmodells der Schweiz; räumliche Informationen aller von Swisstopo erfassten Hügelkuppen und Berggipfel der Schweiz) sind in der Form von Reintextdateien gespeichert. Als client-seitige Anwendung speichert «Mont» keinerlei Nutzerdaten.

² Diese Art der Programmierung wird oft auch «Vanilla JS» genannt, in Anlehnung an die weite Verbreitung des Vanillegeschmacks in Lebensmitteln.

³ Weibel, Thomas (2022): «Mont». <https://www.thomasweibel.ch/mont2/qr.png>, 27.12.2022

⁴ Weibel, Thomas (2022): «Mont». <https://www.thomasweibel.ch/mont2/>, 27.12.2022

⁵ Swisstopo (2021): Digitale Geodaten von swisstopo sind neu kostenlos und können frei genutzt werden. https://www.swisstopo.admin.ch/de/home/meta/medieninformationen.detail.news.html/swisstopo-internet/news2021/news_release/20210301.html, Bern, 1.3.2021

Die Webapp «Mont»

Augmented Reality setzt im Wesentlichen dreierlei voraus:

1. Kameraansicht der User-Umgebung
2. Geolokalisierung
3. Zugriff auf die Gyroskop- und Beschleunigungssensoren des Endgeräts
4. Zugriff auf den Kompass-/Magnetsensor des Endgeräts
5. Daten und Inhalte zwecks Erweiterung des Nutzererlebnisses

Augmented-Reality-Anwendungen setzen in aller Regel spezifische Hardware voraus – dedizierte, mit Kameras und Sensorik versehene VR- oder AR-Brillen, Smartphone-Erweiterungen wie das mit optischen Linsen bestückte Google Cardboard⁶ oder gar aus komplexen Audio- und Beamersystemen bestehende Visualisierungsumgebungen im grossen Massstab. Daraus ergeben sich auch ihre Beschränkungen: Weil diese Geräte im Nutzeralltag nicht verfügbar sind, ist das einzige potenziell AR-fähige Gerät, das Userinnen, User jederzeit bei sich tragen, das Smartphone. Die Webapp «Mont» ist daher konsequent für Smartphones konzipiert, soll so niederschwellig nutzbar sein wie möglich (beim Spazieren oder Wandern etwa) und darf keinerlei Zusatzhardware voraussetzen.

Aus Gründen, die im Verlauf dieses Papers ausführlicher erläutert werden sollen, beziehen sich alle OS- und gerätespezifischen Angaben auf Mobilgeräte der Marke Apple und das Betriebssystem iOS bzw. iPadOS.

Daten

Am 30. November 2018 verabschiedete der Bundesrat die «Open Government Data»-Strategie für die Jahre 2019–2023. Diese sieht vor, dass die Bundesverwaltung ihre Daten zur freien Nutzung veröffentlicht; ausgenommen sind lediglich Bestände, bei denen ein

⁶Google (2014): Google Cardboard. <https://arvr.google.com/cardboard/>, 10.11.2022

«legitimes Schutzinteresse»⁷ überwiegt, etwa in Bereichen der öffentlichen Sicherheit oder beim Datenschutz.

Seit der Publikation der bundesrätlichen Strategie am 30. November 2018 wurden in mehreren Schritten grosse Datenbestände publiziert – seit dem 1. März 2021 zum Beispiel sind sämtliche Geodaten des Bundesamtes für Landestopografie frei erhältlich. Diese Daten liegen in gut dokumentierter Form auf der Plattform opendata.swiss vor und können frei heruntergeladen und genutzt werden.

Digitales Höhenmodell der Schweiz

Die Grundvoraussetzung für eine Anwendung im Bereich der Topografie ist das Vorhandensein räumlicher Daten, welche die Geländestruktur der Schweiz möglichst genau abbilden. Das digitale Höhenmodell DHM 25/200 mit einer Maschenweite von 200 Metern⁸ beschreibt die Geländeoberfläche der Schweiz dreidimensional⁹ und bietet ein abstrahiertes Geländemodell in der Form von rund 1,5 Mio. Datenpunkten mit ihren Koordinaten in allen drei Raumachsen. Die x-Achse entspricht dabei der Longitude (östliche Länge) sowie der Latitude (nördliche Breite) in Metern der Schweizerischen Landesvermessung von 1903 (LV03) oder von 1995 (LV95). Die z-Achse (Raumachse) wird in Metern über Meer angegeben. Das detailliert dokumentierte¹⁰ Höhenmodell liegt in verschiedenen Tabellenformaten sowie einem vollständigen 3D-Modell der Schweiz im Drawing Interchange Format (DXF) vor, das für den Import in die meisten 3D-Programme verwendet werden kann.

⁷ Schweizerische Eidgenossenschaft (2018): Strategie für offene Verwaltungsdaten in der Schweiz 2019–2023 (Open-Government-Data-Strategie, OGD-Strategie).

<https://www.news.admin.ch/news/message/attachments/55083.pdf>, S. 2

⁸ Swisstopo (2010): DHM 25/200, das digitale Höhenmodell der Schweiz mit einer Maschenweite von 200 m.

<https://opendata.swiss/de/dataset/das-digitale-hohenmodell-der-schweiz-mit-einer-maschenweite-von-200-m>, 3.1.2023

⁹ Das Modell beschreibt die Topografie der Schweiz ohne Bewuchs und Bebauung; Seen werden mit der Höhe der Seeoberflächen dargestellt.

¹⁰ Swisstopo (2023): DHM 25/200, das digitale Höhenmodell der Schweiz mit einer Maschenweite von 200 m. Weiterführende Informationen, Dokumente.

<https://www.swisstopo.admin.ch/de/geodata/height/dhm25.html#dokumente>, 10.1.2023

Sammlung geografischer Namen der Schweiz

Die Namen aller Geländeerhebungen der Schweiz stammen aus der Sammlung geografischer Namen der Landesvermessung swissNAMES3D¹¹. Dieser Datensatz stellt die vollständigste toponymische Sammlung der Schweiz (und des Fürstentums Liechtenstein) dar und bildet die Basis der Beschriftungen aller schweizerischen Landeskarten. Für die vorliegende Anwendung «Mont» wurden der Datenbank alle Einträge der Kategorien «Alpine Gipfel», «Hauptgipfel», «Gipfel», «Haupt Hügel» und «Hügel» entnommen. Damit ist die Webapp «Mont» in der Lage, über 9000 Erhebungen auf dem Gebiet der Schweiz zu erkennen und darzustellen.

Speicherung und Abruf

Um eine Ablage aller Daten im Browser-Cache bei Aufruf der Webapp (und damit eine Funktionalität auch in Gebieten ohne hinreichenden Datenempfang) sicherzustellen, speichert «Mont» die Höhendaten des Geländemodells und der Geländeerhebungen nicht in einer Datenbank ab, sondern in Reintextform als JS-Arrays. Diese Arrays mit den Variablen `a`, `b` und `c` (Longitude, Latitude und Meereshöhe) für das Höhenmodell sowie den Variablen `nm`, `lo`, `la`, und `he` (Gipfelnamen, Longitude, Latitude und Gipfelhöhe) werden am Anfang des HTML-Dokuments als externe Skripte eingebunden – die geografische Länge aller Geländepunkte des Höhenmodells beispielsweise als `<script src="data/a.js"></script>` – und werden im eigentlichen Programmskript `script.js` mittels `var` als Variablen deklariert. Durch das «hoisting» genannte Verfahren¹², mit dem Javascript Deklarationen von Funktionen, Variablen oder Klassen per Default an den Kopf des auszuführenden Programmcodes setzt, können im weiteren Verlauf des Skripts sämtliche Rauminformationen in der Syntax `Variablenname[Indexnummer]` abgefragt und verarbeitet werden. Geländepunkte des Höhenmodells liegen so entsprechend den Raumachsen als `a[n1]`, `b[n1]` und `c[n1]` vor; Gipfelnamen als `nm[n2]`, `lo[n2]`, `la[n2]` und `he[n2]`.

¹¹ Swisstopo (2014): swissNAMES3D, geografische Namen der Landesvermessung. <https://opendata.swiss/de/dataset/swissnames3d-geografische-namen-der-landesvermessung>, 3.1.2023

¹² Mozilla Foundation (2022): Resources for Developers, MDN Web Docs Glossary: Definition of Web-related Terms, Hoisting. <https://developer.mozilla.org/en-US/docs/Glossary/Hoisting>, 6.1.2023

Programmierung

Weil es sich bei «Mont» im technischen Sinn um eine einfache Webseite handelt, setzt die Nutzung keinerlei App-Installation voraus. Nach dem Zugriff werden (in dieser Reihenfolge) die nötigen Metadaten, die Anwendungsdaten, das Styling (CSS), das Markup (HTML) und schliesslich das Programm (JS) geladen und ausgeführt.

Je nach vorgenommenen Konfiguration ersucht das Smartphone nach einer Zustimmung für

- die Geolokalisierung,
- den Zugriff auf die Kamera sowie
- den Zugriff auf die Gyroskopsensoren.

Um die Funktionalität zu gewährleisten, ist eine Zustimmung in allen drei Punkten erforderlich; im Fall einer Ablehnung gibt «Mont» eine entsprechende Fehlermeldung aus. Mit Ausnahme der Lagesensoren bzw. des Gyroskop- und Beschleunigungssensors lässt sich eine Zustimmungen in den Browsereinstellungen speichern, so dass die Abfrage beim nächsten Laden der Webapp unterbleibt.

Sobald «Mont» auf dem Bildschirm angezeigt wird, ist die Webapp betriebsbereit. Weil alle Daten im Cache des Browsers zwischengespeichert werden, bleibt die Applikation auch in unwegsamem Gelände ohne hinreichenden Datenempfang funktionsfähig und muss zur erneuten Verwendung nicht neu geladen werden.

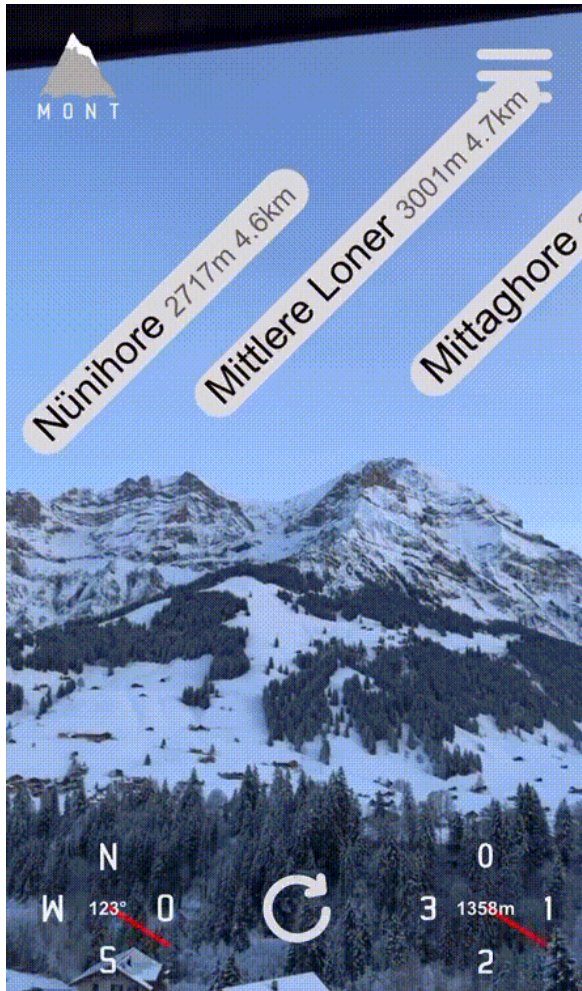


Abbildung 1: Screenshot der Webapp MONT: Blick von Adelboden aus in Richtung Ost-Südost. (Eigene Darstellung)

Beim App-Start und nach Vorliegen aller erforderlichen Zustimmungen lokalisiert die App das Gerät und rechnet dann die vom GPS-Sensor empfangenen Koordinaten in Schweizer Landeskoordinaten um. Anschliessend werden auf der Basis des digitalen Höhenmodells die Meereshöhen der vier nächstgelegenen Raumpunkte ermittelt und mittels bilinearer Interpolation die approximative Meereshöhe des aktuellen Userstandorts berechnet. Anschliessend durchsucht die App die Koordinaten und Meereshöhen sämtlicher Berggipfel und Hügelkuppen der Schweiz, filtert innerhalb der Reichweite von 20 km liegende Erhebungen heraus und berechnet mittels horizontaler und vertikaler Triangulation alle, die vom aktuellen Standort aus tatsächlich

sichtbar sind. Oberhalb jedes Gipfels wird der Name, die Höhe sowie Distanz eingefügt. Der Schriftgrad der dargestellten Gipfelnamen ist dabei abhängig von der jeweiligen Luftliniendistanz. Alle Beschriftungen auf dem Bildschirm folgen den jeweiligen Gipfeln entsprechend den Bewegungen des Smartphones.

Ein Kompass unten links zeigt die Ausrichtung des Smartphones auf einer analogen Kompassrose sowie in Kompassgraden an; ein Altimeter unten rechts stellt die interpolierte Meereshöhe numerisch sowie analog auf einer 1000-m-Skala dar. Der Refresh-Button ermöglicht eine Neuberechnung der Topografie nach einem Standortwechsel.

Funktionsweise

In technischer Hinsicht ist «Mont» eine Webpage in HTML 5 mit externem Stylesheet in CSS 3, externen Daten-Arrays sowie dem externen Programmcode in Javascript ES 6. Für eine Nutzung der Webapp reicht ein vollständiges Laden der Seite.

Kamerastream

Während herkömmliche Webseiten Text und Medieninhalte auf einem neutralem Hintergrund darstellen, ist für eine AR-Webapp ein Zugriff auf den Videostream der Gerätekamera notwendig, der anschliessend mit sinnstiftenden Informationen versehen werden kann. Weil die meisten Mobilgeräte über eine oder mehrere Kameras verfügen, schlug das World-Wide-Web-Konsortium (W3C) erstmals 2012 die JS-Methode `getUserMedia` des `navigator`-Objekts vor¹³, deren einfachste Form

```
const stream = await navigator.mediaDevices.getUserMedia({
  video: {
    width: 1280,
    height: 720,
    aspectRatio: 16/9
  }
});
```

¹³ World Wide Web Consortium (2012): Media Capture and Streams. W3C Working Draft 28 Juni 2012. Cambridge MA, <https://www.w3.org/TR/2012/WD-mediacapture-streams-20120628/>, 10.11.2022

bereits programmatisch auf den Videostream der Front- oder Rück-Kamera zugreifen und diesen in eine Webseite einbetten kann. Anschliessend lassen sich Text- und Medienhalte in der Form von Overlays über einer dynamischen Ansicht der realen Userumgebung darstellen.

Geolokalisierung

Voraussetzung für Applikationen, die standortabhängige Daten ausspielen, ist eine möglichst präzise Lokalisierung des aktuellen Userstandorts. Seit 2008¹⁴ ist die Geolocation API Teil des HTML5-Standards. Zusätzlich zur Position lassen sich auch die Genauigkeit der Koordinaten, die Fortbewegungsgeschwindigkeit und, zumindest theoretisch, die Bewegungsrichtung abfragen. Da die Ermittlung des Standorts einen starken Eingriff in die Privatsphäre der Userinnen und User bedeutet¹⁵, sind Abfragen nur in sicheren Kontexten, also nur über das HTTPS-Protokoll möglich. Dazu erfordert jede Standortabfrage (zumindest erstmalig) das ausdrückliche Einverständnis der Nutzerinnen und Nutzer.

Die Geolokalisierung ermittelt die Koordinaten des aktuellen Breiten- und Längengrads mit Hilfe des GPS-Sensors (falls verfügbar) oder des Mobilfunksignals, erreichbarer WLAN-Netzwerke sowie der IP-Adresse des Geräts (falls GPS nicht verfügbar ist)¹⁶. Im Fall eines Rückgriffs auf die IP-Adresse werden die Daten des Provider-Vermittlungsknotens zur Positionsbestimmung herangezogen.

Abfragen lassen sich die Eigenschaften `longitude` und `latitude` mittels der Javascript-Methode `getCurrentPosition()` des `navigator`-Objekts. Im Freien und je nach Verfügbarkeit von GPS auf dem Endgerät bzw. der Qualität der empfangenen Mobilfunk-/WIFI-Signale erzielt die HTML5-Geolokalisierung Genauigkeiten von 5-20 m¹⁷. Die Ausgabe erfolgt in Dezimalgraden gemäss dem geodätischen Referenzsystem WGS84. Für das

¹⁴ World Wide Web Consortium (2008): Geolocation API Specification. Geopriv Suggestion Draft [DATE: 3 December 2008]. Cambridge MA,

<https://www.w3.org/2008/geolocation/drafts/API/spec-source-CDT.html>, 10.11.2022

¹⁵ World Wide Web Consortium (2022): Geolocation API. W3C Recommendation 01 September 2022.

<https://www.w3.org/TR/geolocation/#security>, 4.1.2023

¹⁶ World Wide Web Consortium (2022): Geolocation API. W3C Recommendation 01 September 2022.

<https://www.w3.org/TR/geolocation/#introduction>, 4.1.2023

¹⁷ Im Gebäude- oder Wageninneren bzw. in der Nähe starker Magnetfelder, wie sie insbesondere im öffentlichen Verkehr auftreten, kann der GPS-Empfang gestört werden, so dass die Genauigkeit der Positionsbestimmung entweder stark reduziert ist oder eine Lokalisierung gänzlich verunmöglicht wird.

Medienhaus Chur beispielsweise lauten diese Angaben 46.84793 Grad nördliche Breite und 9.50186 Grad östliche Länge. Weil die Geodaten des Bundesamtes für Landestopografie jedoch in den Formaten der schweizerischen Landesvermessung von 1903 oder 1995 vorliegen, müssen die Dezimalgrade gemäss WGS84 in m gemäss LV03 bzw. LV95 umgerechnet werden¹⁸. «Mont» nutzt die Landeskoordinaten (Longitude als Variable e , Latitude als Variable n) im Format LV03.

```
navigator.geolocation.getCurrentPosition((position)=> {  
  let b1=position.coords.latitude*3600;  
  let l1=position.coords.longitude*3600;  
  let b2=(b1-169028.66)/10000;  
  let l2=(l1-26782.5)/10000;  
  let n=200147.07+308807.95*b2+3745.25*Math.pow(l2,2)+76.63  
*Math.pow(b2,2)+119.79*Math.pow(b2,3)-194.56*Math.pow(l2,2)*b2;  
  let e=600072.37+211455.93*l2-10938.51*l2*b2-0.36*l2  
*Math.pow(b2,2)-44.54*Math.pow(l2,3);  
}
```

Für den genannten Standort Medienhaus Chur lauten die entsprechenden Koordinaten 757'348 m östliche Länge und 190'601 m nördliche Breite.

¹⁸ Swisstopo (2016): Näherungsformeln für die Transformation zwischen Schweizer Projektionskoordinaten und WGS84. Bern, https://www.swisstopo.admin.ch/content/swisstopo-internet/de/swisstopo/documents.download/swisstopo-internet/de/documents/geo-documents/ch1903wgs84_d.pdf, 10.11.2022, S. 3

Lagesensoren



Abbildung 2: Smartphone mit den drei Raumachsen x , y und z . (Illustration: W3C)

Lage-/Beschleunigungssensoren und Gyroskop ermöglichen gemäss W3C-Spezifikation¹⁹ das Auslesen der räumlichen Orientierung des Geräts entlang der drei Raumachsen x , y und z mithilfe des Eventlisteners `deviceorientation` und in den drei Lagewinkeln `alpha`, `beta` und `gamma`.

¹⁹ World Wide Web Consortium (2022): DeviceOrientation Event Specification, W3C Working Draft, 19 September 2022. Cambridge MA, <https://www.w3.org/TR/orientation-event/>, 10.11.2022

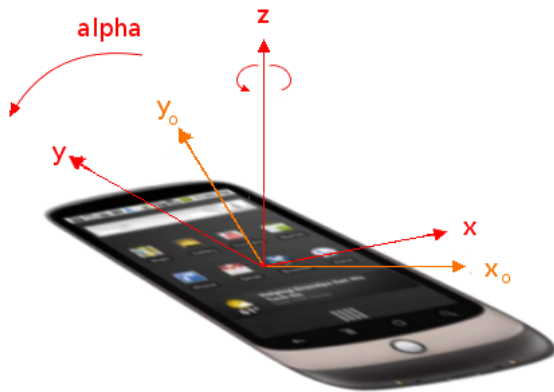


Abbildung 3: Smartphone mit dem Lagewinkel α (Hochachse). (Illustration: W3C)



Abbildung 4: Smartphone mit dem Lagewinkel β (Querachse). (Illustration: W3C)



Abbildung 5: Smartphone mit dem Lagewinkel γ (Längsachse). (Illustration: W3C)

Mittels der Abfrage

```
if (typeof DeviceMotionEvent.requestPermission=== "function") {
  DeviceMotionEvent.requestPermission()
    .then((state)=> {
      if (state=== "granted") {
        window.addEventListener("deviceorientation", direction);
      }
      else {
        fail();
      }
    })
    .catch(console.error);
}
else {
  window.addEventListener("deviceorientation", direction);
}
```


lassen sich anschliessend die Lagewinkel mittels der JS-Eigenschaften `event.alpha`, `event.beta` und `event.gamma` unabhängig von der verwendeten iOS- bzw. iPadOS-Version ermitteln und die Funktion `direction` auslösen, deren Vorgehensweise nachstehend erläutert wird.

Magnetsensor/Kompass

Für die horizontale Triangulation kommt die (experimentelle) Eigenschaft `event.webkitCompassHeading` zum Zug. Im Gegensatz zum relativen Lagewinkel der Hochachse `alpha`, die den Nullpunkt («Norden») auf die Ausrichtung des Smartphones beim Start der App kalibriert (sog. «*game-based calibration*»), greift die Eigenschaft `webkitCompassHeading` auf das im Smartphone oder Tablet verbaute Magnetkompassmodul zurück und liefert absolute Kompassdaten (sog. «*world-based calibration*»)²⁰, die für eine korrekte horizontale Triangulation notwendig sind. Zusätzlich wird zum Zweck einer vertikalen Triangulation auch der Lagewinkel der Längsachse, `event.beta`, ermittelt. Die beiden Lagewinkel werden in den Variablen `heading` und `tilt` gespeichert.

```
const direction={()=> {
  heading=event.webkitCompassHeading;
  tilt=event.beta;
  document.querySelector("#needle1").style.transform="rotate("+heading+"deg)";
  document.querySelector("#degrees").innerText=Math.round(heading)+"°";
};
```

Jedesmal, wenn die Smartphone-Sensorik eine Lageveränderung feststellt, greift `webkitCompassHeading` auf den vom Magnetsensor gelieferten Kompasswinkel zu und gibt diesen zurück. Schliesslich wird die Kompassnadel `needle1` um den neuen Wert rotiert und die gerundete Gradzahl ausgegeben.

Obwohl die Eigenschaft `webkitCompassHeading` absolute Ausrichtungsdaten liefert, erweisen sich die ausgegebenen Werte als in der Regel als wenig präzise. Technische Vergleichsstudien zeigen, dass die Abweichungen zwischen der Kompassanzeige eines

²⁰ Tibbett, Richard (2014): Device Orientation «Alpha» Calibration. Implementation Status and Challenges. Cambridge MA, https://www.w3.org/2008/geolocation/wiki/images/e/e0/Device_Orientation_%27alpha%27_Calibration-Implementation_Status_and_Challenges.pdf, 10.11.2022, S. 3

Smartphones oder Tablets einerseits und einem in Geodäsie und Kartografie gebräuchlichen Magnetkompass teils beträchtlich sind²¹. Namentlich bei grosser Fernsicht aber wären für eine korrekte Passung von Gipfelansicht und Datendarstellung Kompassgenauigkeiten nötig, wie sie lediglich wissenschaftliche Bussolen zu liefern vermögen. Um allzu störende Fehlanzeigen zu vermeiden, wurde in der Webapp «Mont» der Erkennungsradius als globale Variable `reach` auf 20 km begrenzt.

Höheninterpolation

Die korrekte vertikale Triangulation erfordert eine möglichst präzise Angabe der aktuellen Standorthöhe über Meer. Für deren Berechnung wird die Funktion `altimeter` mit einer in drei Schritten erfolgenden bilinearen Höheninterpolation genutzt.

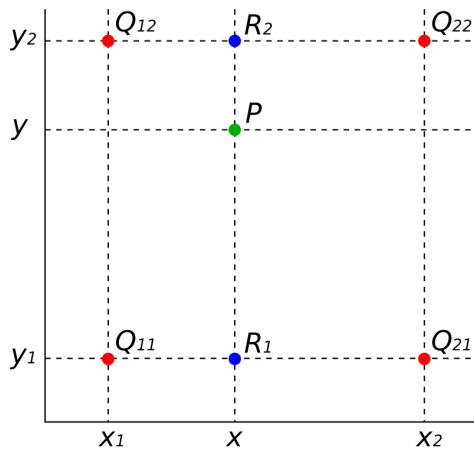


Abbildung 5: Gegeben sind die Meereshöhen der Eckpunkte Q_{11} , Q_{12} , Q_{21} und Q_{22} ; gesucht ist der interpolierte Wert an der Position P . (Illustration: Wikimedia Commons)

²¹ Novakova, Lucie/Pavlis, Terry L. (2017): Assessment of the precision of smart phones and tablets for measurement of planar orientations: A case study. Journal of Structural Geology, Bd. 97, S. 93ff.

Den Positionen Q_{11} , Q_{21} , Q_{12} und Q_{22} entsprechen die Variablen h_1 , h_2 , h_3 und h_4 . In den Arrays a und b sind die geografischen Längen und Breiten aller Geländepunkte des digitalen Höhenmodells gespeichert, der Array c enthält die jeweiligen Meereshöhen. Die Variablen fx und fy bezeichnen die relative Distanz zum nächsttieferen Messpunkt auf der x- (Longitude) bzw. y-Achse (Latitude).

Die lineare Interpolation wird insgesamt dreimal angewendet, zuerst zweimal für die x-Achse (die jeweiligen Rundungen mittels `Math.floor` beziehen sich auf die DHM-Rasterweite von 200 m) und anschliessend einmal für die y-Achse.

```
const altimeter=(()=> {
  let h1, h2, h3, h4;
  for (i=0; i<a.length; i++) {
    if (a[i]==Math.floor(locX/200)*200 && b[i]==Math.floor(locY/200)*200)
h1=c[i];
    if (a[i]==Math.ceil(locX/200)*200 && b[i]==Math.floor(locY/200)*200)
h2=c[i];
    if (a[i]==Math.floor(locX/200)*200 && b[i]==Math.ceil(locY/200)*200)
h3=c[i];
    if (a[i]==Math.ceil(locX/200)*200 && b[i]==Math.ceil(locY/200)*200) h4=c[i];
  }
  let fx=(locX-Math.floor(locX/200)*200)/200;
  let fy=(locY-Math.floor(locY/200)*200)/200;
  let ht1=fx*(h2-h1)+h1;
  let ht2=fx*(h4-h3)+h3;
  let ht=fy*(ht2-ht1)+ht1;
});
```

Aus h_1 und h_2 wird mittels linearer 1D-Interpolation ht_1 ermittelt. Anschliessend wird ht_2 auf gleiche Weise aus h_3 und h_4 bestimmt. Die auf diese Weise bilinear ermittelte Standorthöhe ht ergibt sich schliesslich aus einer linearen 1D-Interpolation auf der y-Achse zwischen ht_1 und ht_2 .

Gipfeltriangulation

Nach der Lokalisierung des Geräts, der Ermittlung seiner horizontalen (Kompassausrichtung) und vertikalen Lagewinkel sowie der interpolierten Meereshöhe liegen nun – zusammen mit den Höheninformationen des digitalen Höhenmodells und der Sammlung

geografischer Namen der Landesvermessung – sämtliche Informationen vor, die für eine Triangulation aller sichtbaren Berggipfel und Hügelkuppen nötig sind.

Mithilfe von `Math.sqrt(Math.pow(nodeX,2)+Math.pow(nodeY,2))` werden die Luftliniendistanzen zwischen Userposition und sämtlichen DHM-Rasterpunkten ermittelt (wobei die Variablen `nodeX` und `nodeY` die Differenz in geografischer Länge bzw. Breite und ergo die Länge der beiden Katheten des entsprechenden rechtwinkligen Dreiecks darstellen) und im Hilfsarray `nodeDists` zwischengespeichert. Der Arkussinus aus dem Verhältnis von Breitendifferenz `nodeY` und Rasterpunktdistanz `nodeDist`, multipliziert mit $180/\pi$, ergibt bei positiven Breitendifferenzen als absolute Summe mit 90 Grad den horizontalen Lagewinkel `nodeHAng` Ost, bei negativen Breitendifferenzen als absolute Differenz zu 270 Grad den horizontalen Lagewinkel `nodeHAng` West.

```
if (nodeX>0) nodeDistAng=90+Math.asin(nodeY/nodeDist)*180/Math.PI;
else nodeDistAng=270-Math.asin(nodeY/nodeDist)*180/Math.PI;
let nodeHAng=Math.abs(nodeDistAng);
```

Der absolute vertikale Lagewinkel lässt sich analog als Arkustangens aus dem Verhältnis von Höhendifferenz `c[n]-ht` und Rasterpunktdistanz `nodeDist`, multipliziert mit $180/\pi$, bestimmen:

```
Math.abs(Math.atan((c[i]-ht)/nodeDist)*180/Math.PI);
```

Danach werden auf dieselbe Weise die horizontalen und vertikalen Lagewinkel aller topografischen Erhebungen ermittelt. Schliesslich werden die Indexnummern aller Rasterpunkte des Höhenmodells sowie aller Hügelkuppen und Berggipfel, die innerhalb der Erkennungsdistanz von 20 km liegen, in den Hilfsarrays `nodes` bzw. `peaks` zwischengespeichert. Für jeden Rasterpunkt des digitalen Höhenmodells und für jeden Gipfel innerhalb des Erkennungsradius liegt nun, relativ zur Position der Userin, des Users, die exakte Kompassrichtung und Neigung vor.

Gipfelsichtbarkeit

Eine Ausgabe sämtlicher innerhalb einer Sichtweite von 20 km verfügbaren Gipfelinformationen wäre nicht sinnvoll, weil nur wenige davon vom Userstandpunkt aus

tatsächlich sichtbar sind. Aus diesem Grund erfolgt als nächstes eine Auswertung aller verfügbaren räumlichen Informationen aus dem DHM und der Sammlung geografischer Namen.

Die mathematische Beurteilung der Sichtbarkeit eines innerhalb der Erkennungsdistanz liegenden Gipfels lässt sich mit einer geschachtelten Schleife (*nested loop*) erreichen, die sämtliche Gipfelhöhen der Indizes aus `peaks` mit den Meereshöhen aller zwischen User- und Bergposition liegenden DHM-Rasterpunkte der Indizes aus `nodes` (innerhalb einer horizontalen und vertikalen Toleranz) vergleicht. Um der horizontalen Ausdehnung von Hügeln und Bergen Rechnung zu tragen, wurde dabei die horizontale Toleranz (als globale Variable `toleranceX`) auf 20 Grad festgesetzt; die vertikale Toleranz (als globale Variable `toleranceY`) beträgt 1 Grad.

```
for (i=0; i<peaks.length; i++) {
  for (j=0; j<nodes.length; j++) {
    if (
      Math.abs(nodeAngX[nodes[j]]-peakAngX[peaks[i]])<toleranceX &&
      nodeDists[nodes[j]]<peakDists[peaks[i]] &&
      nodeAngY[nodes[j]]>peakAngY[peaks[i]]+toleranceY
    ) peaks[i]=null;
  }
}
```

Weisen innerhalb des horizontalen Sektors von 20 Grad vor einem Berggipfel liegende Geländepunkte höhere vertikale Lagewinkel auf, verdecken sie die Sicht, und der fragliche Gipfel wird als Nullobjekt `null` gekennzeichnet und damit bei der Darstellung herausgefiltert. Um alle verfügbaren Rauminformationen zu nutzen (und ergo nicht nur die Meereshöhen aller DHM-Rasterpunkte, sondern auch die Gipfelhöhen aller topografischen Erhebungen zu berücksichtigen), wird die geschachtelte Schleife sinngemäss mit den Indizes aller innerhalb des Erkennungsradius liegenden Gipfel aus `peaks` wiederholt. Von Algorithmus-bedingten Unwägbarkeiten abgesehen (Kompassabweichungen, Ungenauigkeit der bilinearen Höheninterpolation etc.) filtert dieses Verfahren nicht sichtbare Gipfelnamen zuverlässig heraus.

Darstellung

Die Darstellung des Ergebnisses auf dem Smartphone-Bildschirm erfolgt mittels einer Schleife, die sämtliche im Array `peaks` indizierten Berggipfel aus dem Gipfel-Array `nm` als `<p>`-Tag der CSS-Klasse `output` dem DOM (bzw. dem Elternelement `canvas`) hinzufügt und mit

Zusatzinformationen (Luftliniendistanz aus `peakDists` und Gipfelhöhe aus `he`) versieht. Der Schriftgrad verhält sich dabei, bis auf eine Mindestgrösse von $2em$, umgekehrt proportional zur Distanz; ein `z-index` lässt die Beschriftungen nähergelegener Gipfel weiter entfernte überlagern. Das Styling der Beschriftungen wiederum richtet sich nach dem ermittelten Schriftgrad.

```
for (i=0; i<peaks.length; i++) {
  const para=document.createElement("p");
  canvas.appendChild(para);
  para.id="o"+i;
  para.className="output";
  para.innerHTML=nm[peaks[i]]+" <span class=\"small\"> "+he[peaks[i]]+"m
"+Math.round(peakDists[peaks[i]]/100)/10+"km</span>";
  para.style.fontSize=5-3*peakDists[peaks[i]]/reach+"em";
  para.style.zIndex=Math.round(reach-peakDists[peaks[i]])+2;
  para.style.borderRadius=para.offsetHeight/2+"px";
}
```

Animation

Als letztes bedarf die Darstellung der Animation, damit sich sämtliche Beschriftungen bei horizontalen oder vertikalen Bewegungen des Endgeräts im Einklang mit dem Horizont über den Bildschirm bewegen. Positionsbestimmung und Triangulation erfolgen durchgehend dreidimensional, die Bildschirmanimation der Gipfelinformationen dagegen wird, entsprechend der Objektivbrennweite, zweidimensional und mittels linearer CSS-Transitionen vorgenommen.

Um die einzelnen Gipfelnamen am Horizont zu registrieren, wird der genannten, bei jeder horizontalen Lageänderung aufgerufenen Kompassfunktion `direction` eine Schleife hinzugefügt, die alle in Erkennungsreichweite befindlichen Gipfel durchläuft und neu auf dem Bildschirm positioniert. Dabei bestimmt der Faktor `fov` das Verhältnis der (horizontalen und vertikalen) Schriftanimation zu allfälligen Schwenkbewegungen durch die Userin, den User. Dieser Faktor wird von der Brennweite der Rückkamera bestimmt, die bei Smartphones und Tablets von Apple 26 mm beträgt²², was einem *field of view* (FOV) von 69.39 Grad²³ im Querformat (*landscape mode*) entspricht. Da die Webapp «Mont» aber in für Smartphones typischer vertikaler Lage

²² Apple (2022): iPhone 14, iPhone 14 Plus, Tech Specs. <https://www.apple.com/iphone-14/specs/>, 8.1.2023

²³ Fulton, Wayne (2014): Field of View Calculator (FoV) of a Camera and Lens. <https://www.scantips.com/lights/fieldofview.html>, 8.1.2023

(*portrait mode*) benutzt wird und der Kamerasensor ein Seitenverhältnis von 16:9 aufweist, ergibt sich proportional ein vertikaler FOV von 39.09 Grad, der seinerseits der vollen Bildschirmbreite von $100v_w$ entspricht und so einen Animationsfaktor von $2.562_{v_w}/\text{Grad}$ ergibt. Eine Multiplikation der Kompassausrichtung `heading` mit dem Faktor `fov` plus 50 setzt die Beschriftungen also entsprechend ihrer Richtung horizontal in die Mitte des Bildschirms; eine Multiplikation des Neigungswinkels `tilt` mit `fov` plus 89 (als proportionales Ergebnis aus der halben Bildschirmbreite $50/9*16$) mittet die Beschriftungen vertikal ein.

```
let fov=2.562; // 26mm focal length > 69.39° fov horizontal, 16:9 ratio > 39.03
fov vertical = 100vw
for (i=0; i<peaks.length; i++) {
  let label=document.querySelector("#o"+i);
  label.style.left=((peakAngX[peaks[i]]-heading)*fov+50)+"vw";
  label.style.bottom=((peakAngY[peaks[i]]+90-tilt)*fov+89)+"vw";
}
```

Weitere kleinere Routinen beheben Inkonsistenzen der Darstellung wie den (sich aus der Periodisierung der Anzeige ergebenden) toten Winkel beim Nulldurchgang des Kompasses, von der linearen CSS-Transition herrührende Fehldarstellungen der Beschriftungen beim Kompassdurchgang von 270, 0 und 90 Grad oder eine numerische Kompassanzeige von 360 statt 0 Grad.

Fehlerbehandlung

Weil bei der Nutzung von «Mont» Fehler auftreten können (fehlende Berechtigungen des Webbrowsers, Nutzung ausserhalb des Datenbereichs o.ä.), ist ein App-weites Error-Management mit der Ausgabe einschlägiger Fehlermeldungen nötig. Dabei werden zweierlei Fehler unterschieden: eine falsche Ausrichtung des Smartphones sowie Fehler, die ein neues Laden der Webapp erfordern.

Smartphone-Ausrichtung

Um die Verwendung von «Mont» so intuitiv wie möglich zu halten, ist eine Nutzung im Smartphone-typischen Hochformat (*portrait mode*) vorgesehen. Weil die Haltung im Querformat

(*landscape mode*) die Kalibrierung des Magnetkompassmoduls beeinflusst – ein Drehen des iPhones führt zu einer Abweichung von 90 bzw. -90 Grad –, wird mittels einer einfachen CSS-media query eine das User Interface `<main>` verdeckende Fehlermeldung `<mark>` ausgegeben.

```
@media (orientation:landscape) {
  main {
    visibility: hidden;
  }
  mark {
    visibility: visible;
  }
}
@media (orientation:portrait) {
  main {
    visibility: visible;
  }
  mark {
    visibility: hidden;
  }
}
```

Diese Grafik fordert die Userin, den User dazu auf, das Smartphone im Hochformat zu halten. Nach dem Drehen in die korrekte Lage wird `<mark>` ausgeblendet und die App angezeigt.

Andere

Die nachfolgenden Fehler verhindern das korrekte Funktionieren von «Mont» und erfordern ggf. eine Änderung der Geräteeinstellungen. Fehler dieses Typs werden dem Mitteilungs-Array `msgs` hinzugefügt, der beim App-Start durch die Funktion `fail` überprüft wird. Liegen Meldungen vor (bzw. ist die Länge von `msgs >0`), werden die Meldungen aufgelistet, was eine Anzeige des UI verhindert.

```
const fail={()=> {
  if (msgs.length>0) {
    let msg=msgs.join("\n");
    document.querySelector("#errorType").innerText=msg;
    document.querySelector("#error").style.visibility="visible";
  }
}
```



```
};
```

Lässt sich der Fehler beheben (etwa im Fall fehlender Zustimmungen durch die Userin, den User), muss «Mont» anschliessend neu geladen werden.

Betriebssystem

Weil die zur Bestimmung der horizontalen Ausrichtung genutzte Eigenschaft `webkitcompassheading` gegenwärtig nur von Mobilgeräten von Apple unterstützt wird²⁴, prüft «Mont» beim Start das vorliegende Betriebssystem und gibt auf Geräten mit Android oder Windows Phone die entsprechende Meldung aus. Nur auf Apple-Geräten (iPhone, iPad, iPod) wird mittels der Funktion `query` die topografische Berechnung gestartet.

```
const getMobileOS=( )=> {
  let userAgent=navigator.userAgent || navigator.vendor || window.opera;
  if (/windows phone/i.test(userAgent)) {
    msgs.push("Kein iOS-Gerät");
    console.error("Kein iOS-Gerät");
    fail();
  }
  if (/android/i.test(userAgent)) {
    msgs.push("Kein iOS-Gerät");
    console.error("Kein iOS-Gerät");
    fail();
  }
  if (/iPad|iPhone|iPod/.test(userAgent) && !window.MSStream) {
    query();
  }
};
```

Geolocation

Um ein mögliches Tracking durch Websites zu verhindern, verzichten viele Userinnen und User darauf, dem Webbrowser die Berechtigung zur Lokalisierung zu erteilen. Entweder ist der Zugriff auf Lokalisierungsdaten via GPS (oder erreichbare Mobilfunksignale,

²⁴ Mozilla Foundation (2022): Resources for Developers, MDN Web Docs Glossary: Definition of Web-related Terms, DeviceOrientationEvent.
<https://developer.mozilla.org/en-US/docs/Web/API/DeviceOrientationEvent>, 6.1.2023

WLAN-Netzwerke sowie die IP-Adresse des Geräts) geräteweit abgeschaltet, oder aber die Abfrage der entsprechenden Berechtigung beim App-Start wird abgelehnt. In diesem Fall ist eine Lokalisierung unmöglich und damit die Voraussetzung für eine Gipfeltriangulation nicht gegeben, was mit der Ausgabe der entsprechenden Meldung quittiert wird.

```

if (navigator.geolocation) {
  navigator.geolocation.getCurrentPosition((position)=> {
    // geolocation success
  }
  altimeter();
}, ()=> { // geolocation fail
  msgs.push("Ortung nicht möglich");
  fail();
}, { // geolocation options
  enableHighAccuracy: true,
  timeout: 5000,
  maximumAge: 0
});
}
else {
  msgs.push("Ortung nicht möglich");
  fail();
}

```

Geografie

«Mont» beinhaltet lediglich Topografie-Daten des schweizerischen Landesgebiets. Eine Nutzung im Ausland wäre mit der Festsetzung eines Default-Standorts (wie etwa die alte Sternwarte Bern, der «Nullpunkt» des Schweizer Koordinatensystems²⁵) zwar grundsätzlich möglich, ist bei einer auf der realen Userumgebung basierenden AR-Anwendung aber nicht sinnvoll. Aus diesem Grund erfolgt eine Abfrage der Grenzkoordinaten zur Verhinderung der UI-Anzeige im Fall eines Zugriffs von einem Standort ausserhalb der Schweiz.

```

if (e>=480000 && e<=865000 && n>=74000 && n<=302000) {
  locX=e;
  locY=n;
}

```

²⁵ Swisstopo (2023): Schweizer Koordinatensystem.

<https://www.swisstopo.admin.ch/de/wissen-fakten/geodaesie-vermessung/koordinaten/schweizer-koordinaten.html>, 10.1.2023

```
    init();
  }
  else {
    msgs.push("Ausserhalb der Schweiz");
    fail();
    init();
  }
}
```

Im Fall Aufrufs ausserhalb dieser Grenzkordinaten wird ebenfalls eine entsprechende Fehlermeldung ausgegeben.

Kamera

Wie jede AR-Anwendung setzt auch «Mont» einen Zugriff auf den Kamerastream voraus. Wird dieser durch eine fehlende Zustimmung durch die Userin, den User unterbunden, sind die Voraussetzung für ein Funktionieren der Webapp nicht gegeben, und das UI wird nicht angezeigt.

```
if (navigator.mediaDevices.getUserMedia) {
  navigator.mediaDevices.getUserMedia({video:{facingMode: "environment", height:
2560, width: 1440, aspectRatio: 16/9}})
  .then ((stream)=> {
    if (initial) document.querySelector("#camera").srcObject=stream;
    initial=false;
  })
  .catch ((err)=> {
    msgs.push("Kein Zugriff auf Kamera");
    fail();
  });
}
else {
  msgs.push("Kein Zugriff auf Kamera");
  fail();
}
```

Ist der Zugriff auf die Kamera dagegen erfolgreich, wird die Ausgabe auf dem Bildschirm entsprechend den Gerätespezifikationen in Höhe, Breite und Seitenverhältnis formatiert und der Stream initialisiert.

Magnetkompass, Gyroskop

Eine korrekte Registrierung der Gipfelbeschriftungen am Horizont und eine Animation entsprechend den Gerätebewegungen erfordert einen Zugriff auf das im Gerät verbaute Magnetkompassmodul sowie auf die Lage- und Beschleunigungssensoren. Im Gegensatz zu Lokalisierung und Kamerazugriff lässt sich hier eine Zustimmung seit der Veröffentlichung von iOS 13 am 1. September 2020 nicht länger über die Systemeinstellungen realisieren, sondern erfordert laut Apple-Designrichtlinien²⁶ zwingend eine Nutzer-Interaktion. Die Startfunktion `activate` ist daher Teil der Funktion `menu`, die von der Userin, dem User mit einem Klick-Event betätigt werden muss, um das App-Intro zu schliessen.

```
const menu=()=>> {  
  const instr=document.querySelector("#info");  
  if (instr.style.visibility=="hidden") instr.style.visibility="visible";  
  else {  
    instr.style.visibility="hidden";  
    activate();  
  }  
};
```

Wird die anschliessende Anfrage um die Berechtigung auf die Bewegungssensoren abgelehnt, ist eine korrekte Registrierung der Beschriftungen am Horizont nicht möglich, und das UI wird ausgeblendet.

Eigenschaften

Wie alle Webapps ist «Mont» einfach zugänglich: Um die Anwendung zu nutzen, reicht der Klick auf einen Weblink, die Nutzung eines darauf verweisenden QR-Codes oder die Eingabe des URL. Die Installation einer Anwendung aus dem Apple App Store ist nicht nötig. Weil die Webapp ausschliesslich aus Client-seitigen Webtechnologien besteht, ist der Quelltext von «Mont» offen zugänglich und beruht zudem auf Open (Government) Data, die ebenfalls

²⁶ Apple (2023): Gyroscope and accelerometer. On-device gyroscopes and accelerometers can supply data about a device's movement in the physical world.
<https://developer.apple.com/design/human-interface-guidelines/inputs/gyro-and-accelerometer/>, 10.1.2023

unter einer offenen Lizenz²⁷ zur Verfügung stehen. Die einzige Bedingung für eine Nutzung und Publikation der Geodaten ist laut Nutzungsbedingungen des Bundesamtes für Landestopografie eine Quellenangabe.

Als reines Webprojekt ist «Mont» ausgesprochen datensparsam: Die gesamte Datenmenge beträgt nur 31 MB; die Grösse des eigentlichen Programmcodes beträgt nur 11,1 kb. Weil darüber hinaus keinerlei externen Libraries oder Frameworks verwendet werden, ist die Webapp autark und daher leicht portier- und anpassbar. Die Programmierung ist effizient und dank Inline-Kommentierung für Entwickler transparent.

Weil «Mont» für Programmierung und Daten lediglich beliebig editierbare Reintextdateien nutzt und darüber hinaus durch den Verzicht auf Libraries und Frameworks über keinerlei externen *Third-Party*-Abhängigkeiten verfügt, lässt sich die App mit wenig Aufwand an beliebige Aufgabestellungen anpassen. Die einzige Voraussetzung ist das Vorhandensein georeferenzierter Daten, aus denen sich die für «Mont» erforderlichen Datenarrays erzeugen lassen.

Beschränkungen

Die grösste Einschränkung bei der Nutzung von «Mont» ist die fehlende Kompatibilität mit Android- oder anderen Betriebssystemen. Die Webapp beruht zwar durchgängig auf den offenen Webtechnologien HTML 5, CSS 3 und JS ES 6 2016ff. Das W3C schlägt seit 2021 einen Standard zum Auslesen des in Smartphones verbauten Magnetkompassmoduls vor²⁸, doch ist das entsprechende JS-Objekt `Magnetometer` mit den meisten Webbrowsern nach wie vor nicht kompatibel. Da also eine robuste, d.h. für alle Plattformen geeignete JS-Methode zum Auslesen der Kompassorientierung bis heute fehlt und die für «Mont» genutzte Eigenschaft `webkitcompassheading` ebenfalls (noch) nicht zum Webstandard zählt, bleibt «Mont» experimentell. Daher ist zur Zeit eine Nutzung nur in Umgebungen sinnvoll, in denen die

²⁷ Swisstopo (2021): Nutzungsbedingungen für kostenlose Geodaten und Geodienste (OGD) von swisstopo. <https://www.swisstopo.admin.ch/de/home/meta/konditionen/geodaten/ogd.html>, 10.1.2023

²⁸ World Wide Web Consortium (2021): Magnetometer. W3C Working Draft 7 December 2021. Cambridge MA, <https://www.w3.org/TR/magnetometer/>, 18.1.2023

Hardwareumgebung kontrolliert werden kann (etwa durch eine Geräteabgabe durch Unternehmen oder Institutionen).

Dazu ist das in Smartphones verbaute Magnetkompassmodul nicht sehr zuverlässig. Selbst abseits magnetischer Störquellen (innerhalb von Gebäuden, in Faradayschen Käfigen wie dem Auto, in der Nähe elektrischer Ober- und Fahrleitungen etc.) lassen sich Abweichungen zu einem konventionellen Magnetkompass feststellen, was die Genauigkeit der Registrierung der topografischen Informationen am Horizont beeinträchtigt. Im Zuge der laufend vorgenommenen Hardwareoptimierungen ist bei aktuellen Smartphonemodellen aber mit einem Leistungszuwachs auch im Bereich der Sensorik (Magnetkompass, Gyroskop, Beschleunigungssensoren) zu rechnen.

Das Handling von «Mont» als Webseite, die weitgehende Berechtigungen erfordert (Lokalisierung, Kamera- und Sensorzugriff), entspricht nicht den von nativen Apps geprägten UX-Erwartungen. Selbst wenn sich der Zugriff auf Kamerastream und Gyroskopdaten systemweit in den Einstellungen regeln lässt, bleibt das Handling der verschiedenen Rechteabfragen durch die Userin, den User umständlich. Diese Einschränkung ergibt sich aus den Systemvorgaben der Betriebssystem- und Browserhersteller und lässt sich von Seiten der Webapp nicht beheben. Weil aber immer mehr leistungsfähige Webapps auch im Geoinformatik-Bereich auf den Markt gebracht werden, ist aber damit zu rechnen, dass künftige OS- und Browserversionen das einschlägige Rechtemanagement vereinfachen werden.

Die Nutzung von «Mont» beschränkt sich schliesslich auf die Schweizer Landesgrenzen, weil nur OGD der Schweizer Behörden genutzt wurden. Eine Anpassung der Webapp auf Zwecke im Ausland ist – bei Vorliegen georeferenzierter Daten im Format WGS 84 – jedoch ebenfalls ohne weiteres möglich.

Weiterentwicklungsmöglichkeiten

Die Entwicklung der Webapp «Mont» brachte eine Reihe von Erkenntnissen über die Leistungsfähigkeit der aktuellen Webtechnologien an den Tag, die bisher nur schlecht dokumentiert waren. Aufgrund der genannten Einschränkungen (fehlende Plattformunabhängigkeit, eingeschränkte UX) ist ein kommerzieller Einsatz von «Mont» zur Zeit dennoch nicht sinnvoll. Eine Ausnahme könnte der Einsatz in einer *Closed-Hardware-Umgebung* sein, in der die benötigten iOS-Endgeräte zur Benutzung an die Userinnen und User abgegeben werden. In Frage kämen hier Anwendungen in Tourismus, Kultur oder der öffentlichen Hand, die ortsgebundene Informationen vermitteln wollen (historische Informationen zu Gebäuden und Plätzen innerhalb eines Stadtgebiets, touristische Infrastruktur oder botanische/zoologische Informationen in einem Nationalpark, Wander- oder Skigebiet, Anbieter- oder Programminformationen auf einen grösseren Festival- oder Messegelände etc.). Die Webapp würde sinnvollerweise in einer *Kiosk-Mode*-Browserapplikation angeboten, die ein Schliessen und ein Navigieren auf andere Webseiten nicht zulässt.

Bestehende, als native Apps für beide Plattformen umgesetzte Lösungen im Bereich von Geografie (wie z.B. swisstopo des Bundesamtes für Landestopografie²⁹) oder Topografie (wie z.B. PeakFinder der PeakFinder GmbH, Zürich³⁰) haben sich auf dem Softwaremarkt gut etabliert. Die Stärke der Webapp «Mont» liegt darin, dass sich die Art der vermittelten Informationen mit wenig Aufwand an beliebige Zwecke anpassen lässt: Jede Art georeferenzierter Inhalte kann durch einen einfachen Austausch der entsprechenden Datenarrays (und, je nach Inhaltstyp, durch Erzeugung des entsprechenden Markups) angezeigt werden. Als Inhalt kommt dabei jede Art von Information (in Text, Audio, Grafik, Bild und Video) in Frage. Auf diese Weise lassen sich, als prototypische Anwendung oder als Angebot in einer *Closed-Hardware-Umgebung*, kostengünstig und innert kurzer Frist multimediale AR-Guides realisieren.

Eine grundsätzlich andere Möglichkeit besteht in der Umsetzung als native Anwendung für die Plattformen iOS und Android: Die schlanke Art der Programmierung der vorliegenden

²⁹ Swisstopo (2020): swisstopo-App. <https://www.swisstopo.admin.ch/de/karten-daten-online/karten-geodaten-online/swisstopo-app.html>, 16.1.2023

³⁰ PeakFinder (2010): PeakFinder App. <https://www.peakfinder.org/de/mobile/>, 16.1.2023

Webapp «Mont» lässt eine einfache Anpassung an beliebige Plattformen zu. Eine native Programmierung würde die Mehrzahl der genannten Einschränkungen beheben, hätte aber einen grösseren Projektumfang und damit zeitliche und finanzielle Mehraufwände zur Folge.

Schlussfolgerungen

Die Programmierung der Webapp «Mont» lässt sich als erfolgreichen Fehlschlag bezeichnen: Das Fehlen robuster JS-Methoden zur Ermittlung absoluter Ausrichtungsdaten und der dadurch erzwungene Rückgriff auf die experimentelle Eigenschaft `webkitCompassHeading` verhindert die Erstellung für alle gängigen Plattformen optimierter Geodaten- und AR-Webapplikationen. Dennoch hat der vorliegende Versuch gezeigt, dass heutige Webtechnologien weit leistungsfähiger sind als Webseiten gemeinhin vermuten lassen. Augmented Reality, also Anwendungen im Bereich der Erweiterten Realität, lassen sich selbst ohne Zuhilfenahme externer JS-Libraries oder -Frameworks realisieren.

Das W3-Konsortium hat die Bedeutung einer Kompassfunktion zwecks Erstellung von Geodaten-basierten Anwendungen schon früh erkannt. Bereits im Jahr 2009 schlugen Autoren des W3C eine Programmierschnittstelle vor³¹, mit der – zusätzlich zur Geolokalisierung – auch die Geräteausrichtung in absoluten Kompassgraden hätte möglich sein sollen. Die aktuelle Version der Geolocation API des W3C³² sieht zwar verschiedene Möglichkeiten einer Abfrage der Kompassorientierung vor, die im Fall des durch die W3C vorgesehenen `heading`-Attributs aber bisher nicht umgesetzt wurden³³.

Bedingt durch die experimentelle Methode zur Ermittlung der Kompassorientierung und die genannten UX-Defizite hat die Webapp «Mont» lediglich experimentellen Status. Künftige Entwicklungen im Bereich der Webtechnologien und der Sensorik könnten Abhilfe schaffen, so dass eine robuste, Plattform-unabhängige Programmierung von Extended-Reality-Anwendungen in der Form von Webapps in Vanilla JS möglich wird.

³¹ World Wide Web Consortium (2009): The Compass API. W3C Editor's Draft 29 October 2009. <https://dev.w3.org/2009/dap/system-info/compass.html>, 10.1.2023

³² World Wide Web Consortium (2022): Geolocation API. W3C Recommendation 01 September 2022. <https://www.w3.org/TR/geolocation/>, 4.1.2023

³³ Das `position`-Objekt der Methode `getCurrentPosition` beinhaltet laut W3C-Empfehlung ein `heading`-Attribut, das aber statt einer Kompassausrichtung lediglich den Wert `NaN` (*not a number*) zurückgibt.

Quellenverzeichnis

- Apple (2022): iPhone 14, iPhone 14 Plus, Tech Specs.
<https://www.apple.com/iphone-14/specs/>, 8.1.2023
- Apple (2023): Gyroscope and accelerometer. On-device gyroscopes and accelerometers can supply data about a device's movement in the physical world.
<https://developer.apple.com/design/human-interface-guidelines/inputs/gyro-and-accelerometer/>, 10.1.2023
- Berners-Lee, Tim (1989): Information Management: A Proposal. Genf,
<https://www.w3.org/History/1989/proposal.html>, 10.11.2022.
- Fulton, Wayne (2014): Field of View Calculator (FoV) of a Camera and Lens.
<https://www.scantips.com/lights/fieldofview.html>, 8.1.2023
- Google (2014): Google Cardboard. <https://arvr.google.com/cardboard/>, 10.11.2022
- Mozilla Foundation (2022): Resources for Developers, MDN Web Docs Glossary: Definition of Web-related Terms, Hoisting.
<https://developer.mozilla.org/en-US/docs/Glossary/Hoisting>, 6.1.2023
- Mozilla Foundation (2022): Resources for Developers, MDN Web Docs Glossary: Definition of Web-related Terms, DeviceOrientationEvent.
<https://developer.mozilla.org/en-US/docs/Web/API/DeviceOrientationEvent>, 10.1.2023
- Novakova, Lucie/Pavlis, Terry L. (2017): Assessment of the precision of smart phones and tablets for measurement of planar orientations: A case study. *Journal of Structural Geology*, Bd. 97, S. 93ff.
- PeakFinder (2010): PeakFinder App. <https://www.peakfinder.org/de/mobile/>, 16.1.2023
- Schweizerische Eidgenossenschaft (2018): Strategie für offene Verwaltungsdaten in der Schweiz 2019–2023 (Open-Government-Data-Strategie, OGD-Strategie).
<https://www.newsd.admin.ch/newsd/message/attachments/55083.pdf>, S. 2

Swisstopo (2021): Digitale Geodaten von swisstopo sind neu kostenlos und können frei genutzt werden.

https://www.swisstopo.admin.ch/de/home/meta/medieninformationen.detail.news.html/swisstopo-internet/news2021/news_release/20210301.html, Bern, 1.3.2021

Swisstopo (2010): DHM 25/200, das digitale Höhenmodell der Schweiz mit einer Maschenweite von 200 m.

<https://opendata.swiss/de/dataset/das-digitale-hohenmodell-der-schweiz-mit-einer-maschenweite-von-200-m>, 3.1.2023

Swisstopo (2023): DHM 25/200, das digitale Höhenmodell der Schweiz mit einer Maschenweite von 200 m. Weiterführende Informationen, Dokumente.

<https://www.swisstopo.admin.ch/de/geodata/height/dhm25.html#dokumente>, 10.1.2023

Swisstopo (2014): swissNAMES3D, geografische Namen der Landesvermessung.
<https://opendata.swiss/de/dataset/swissnames3d-geografische-namen-der-landesvermessung>, 3.1.2023

Swisstopo (2016): Näherungsformeln für die Transformation zwischen Schweizer Projektionskoordinaten und WGS84. Bern,

https://www.swisstopo.admin.ch/content/swisstopo-internet/de/swisstopo/documents.download/swisstopo-internet/de/documents/geo-documents/ch1903wgs84_d.pdf, 10.11.2022, S. 3

Swisstopo (2021): Nutzungsbedingungen für kostenlose Geodaten und Geodienste (OGD) von swisstopo.

<https://www.swisstopo.admin.ch/de/home/meta/konditionen/geodaten/ogd.html>, 10.1.2023

Swisstopo (2023): Schweizer Koordinatensystem.

<https://www.swisstopo.admin.ch/de/wissen-fakten/geodaesie-vermessung/koordinaten/schweizer-koordinaten.html>, 10.1.2023

Swisstopo (2020): swisstopo-App.

<https://www.swisstopo.admin.ch/de/karten-daten-online/karten-geodaten-online/swisstopo-app.html>, 16.1.2023

Tibbett, Richard (2014): Device Orientation «Alpha» Calibration. Implementation Status and Challenges. Cambridge MA, https://www.w3.org/2008/geolocation/wiki/images/e/e0/Device_Orientation_%27alpha%27_Calibration-_Implementation_Status_and_Challenges.pdf, 10.11.2022, S. 3

Weibel, Thomas (2022): «Mont». <https://www.thomasweibel.ch/mont2/>, 10.11.2022

Weibel, Thomas (2022): «Mont». <https://www.thomasweibel.ch/mont2/qr.png>, 27.12.2022

World Wide Web Consortium (2008): Geolocation API Specification. Geopriv Suggestion Draft [DATE: 3 December 2008]. Cambridge MA, <https://www.w3.org/2008/geolocation/drafts/API/spec-source-CDT.html>, 10.11.2022

World Wide Web Consortium (2009): The Compass API. W3C Editor's Draft 29 October 2009. <https://dev.w3.org/2009/dap/system-info/compass.html>, 10.1.2023

World Wide Web Consortium (2012): Media Capture and Streams. W3C Working Draft 28 Juni 2012. Cambridge MA, <https://www.w3.org/TR/2012/WD-mediacapture-streams-20120628/>, 10.11.2022

World Wide Web Consortium (2021): Magnetometer. W3C Working Draft 7 December 2021. Cambridge MA, <https://www.w3.org/TR/magnetometer/>, 18.1.2023

World Wide Web Consortium (2022): DeviceOrientation Event Specification, W3C Working Draft, 19 September 2022. Cambridge MA, <https://www.w3.org/TR/orientation-event/>, 10.11.2022

World Wide Web Consortium (2022): Geolocation API. W3C Recommendation 01 September 2022. <https://www.w3.org/TR/geolocation/>, 4.1.2023

World Wide Web Consortium (2022): Geolocation API. W3C Recommendation 01 September 2022. <https://www.w3.org/TR/geolocation/#security>, 4.1.2023

World Wide Web Consortium (2022): Geolocation API. W3C Recommendation 01 September 2022. <https://www.w3.org/TR/geolocation/#introduction>, 4.1.2023

Illustrationsverzeichnis

Abb. 1: Weibel, Thomas (2022): Mont, Screenshot.

<https://www.thomasweibel.ch/mont2/>, 17.12.2022

Abb. 2: World Wide Web Consortium (2022): DeviceOrientation Event Specification, W3C Working Draft, 19 September 2022. Cambridge MA, <https://www.w3.org/TR/orientation-event/>, 10.11.2022

Abb. 3: World Wide Web Consortium (2022): DeviceOrientation Event Specification, W3C Working Draft, 19 September 2022. Cambridge MA, <https://www.w3.org/TR/orientation-event/>, 10.11.2022

Abb. 4: World Wide Web Consortium (2022): DeviceOrientation Event Specification, W3C Working Draft, 19 September 2022. Cambridge MA, <https://www.w3.org/TR/orientation-event/>, 10.11.2022

Abb. 5: World Wide Web Consortium (2022): DeviceOrientation Event Specification, W3C Working Draft, 19 September 2022. Cambridge MA, <https://www.w3.org/TR/orientation-event/>, 10.11.2022

Abb. 6: National Institutes of Health (2012): Interpolation methods used in MIPAV, Medical Image Processing, Analysis, and Visualization.

https://mipav.cit.nih.gov/pubwiki/index.php/Interpolation_methods_used_in_MIPAV, 17.1.2023

Abkürzungsverzeichnis

AR	Augmented Reality
CSS	Cascading Style Sheets
DHM	Digital Height Model
ER	Enhanced Reality, Extended Reality
ES	ECMAScript
FOV	Field of View
HTML	Hypertext Markup Language
JS	Javascript
LV03	Landesvermessung 1903
LV95	Landesvermessung 1995
OGD	Open Government Data
UI	User Interface
UX	User Experience
VR	Virtual Reality
W3C	World Wide Web Consortium
WGS 84	World Geodetic System 1984