

**Aareschwümmenid**

**Sunne oder Schatteplätzli?**

**Technische Informationen für die Jury**

**Technische Informationen für die Jury**

Aktueller Stand des Sourcecodes

* https://github.com/cblaettl/sol-umbra

Ausgangslage

* Worauf habt ihr euch fokussiert?
* Welche technischen Grundsatzentscheide habt ihr gefällt?

Für die gegebenen Problemstellung entschieden wir uns für eine reine Frontend-Lösung. So wird die Rechenleistung für das Rechnen des Schattenwurfs auf die Client-Maschine, respektive auf dessen Browser, bereitgestellt.

Technischer Aufbau

* Welche Komponenten und Frameworks habt ihr verwendet?

Folgende Bibliotheken wurden verwendet:  
vue.js: rendern der web-app  
ifc.js: darstellen eines 3d-Models (ifc-File), wird in webassembly gerendert.   
mapbox/maplibre: verknüpfung 3d-Modell zu einem Koordinaten-System  
vanilla-js: glue-code

* Wozu und wie werden diese eingesetzt?  
  Siehe oben

Implementation

* Gibt es etwas Spezielles, was ihr zur Implementation erwähnen wollt?
* Was ist aus technischer Sicht besonders cool an eurer Lösung?

Das live rendering der Schatten/Sonnen – Simulation finden wir besonders ansprechend. Die technischen Feinheiten

Abgrenzung / Offene Punkte

* Welche Abgrenzungen habt ihr bewusst vorgenommen und damit nicht implementiert? Weshalb?  
  Es wurde kein Backend-Implementiert, weil dies für ein POC nicht notwendig ist. Selbst für einen ersten Release, um zu sehen ob die Anwendung Verwendung findet, wäre dies nicht notwendig. Falls dann weiterführende Funktionen wie z.B. updaten der Meta-Informationen via Kunden-Feedback oder Hervorheben von Events / Lokalen / Firmen oder ähnlichem, müsste dieser Entscheid nochmals überdenkt werden.