Project: Open Social Spaces (“Rüteliwiese, Horw”)

Project members: Gino Cattelan (gino.cattelan@stud.hslu.ch), Gianni Pinelli(, Benjamin Auer ([auer.benjamin@gmail.com](mailto:auer.benjamin@gmail.com)) , Andy Gubser (Conceptual Coaching), Marco Hassan und Raphael Tholl (Technical Coaching)  
Please add your contact detail / github links

Link to opendata.ch: <https://hack.opendata.ch/project/605>

**Executive Summary**

* + Why ?

Regarding spatial planning local governments often face different challenges. Town hall meetings are often attended by a very homogeneous group consisting of mostly elderly people, while the groups actually using the open social spaces, in particular the youth and minorities, are often underrepresented.

* + How?

We quickly came to the conclusion that a solution is required to invite locals to get involved with shaping their open social spaces. We evaluated different ideas from using BIM objects to creating an App. With the available resources we decided on creating a very easily accessible, 2D design tool that also includes the possibility to share, read and vote on different ideas.

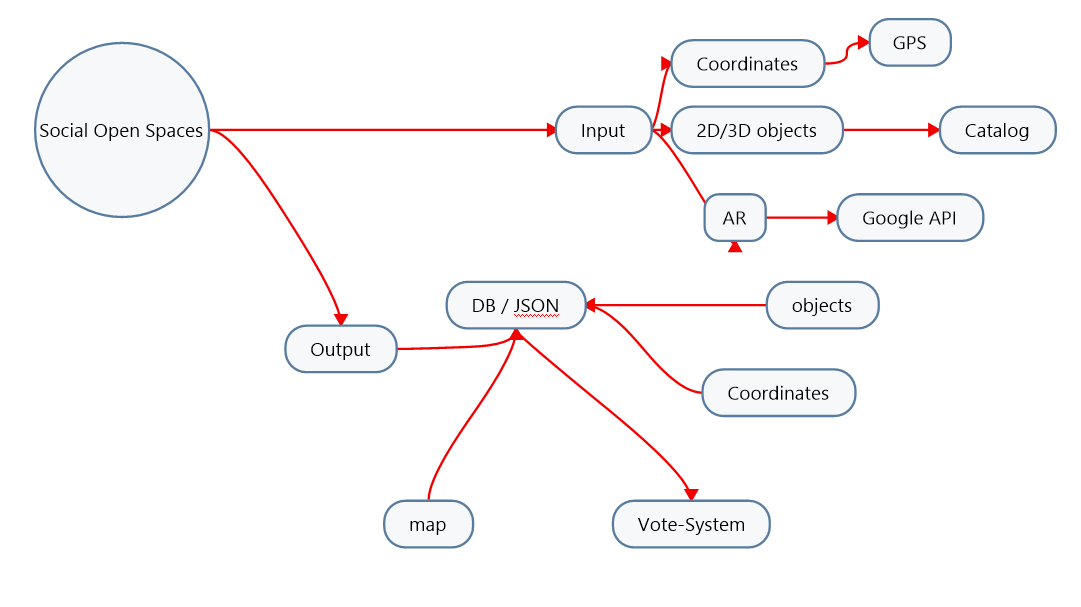
* + What?

We developed a web application accessible via on site QR code. On the application you’re then able to place certain items like trees, benches ect. on a map with AR. It’s then possible for the individual to vote on the ideas of other participants. This data then can be exported into a JSON Database for the government to use.

**Input**

**What:**

* + We created an online AR-Tool to generate Data.
  + On their phone, users can select from different objects (park bench, park bin, tree) and place them in AR into the park.
  + Geodata from phone is added to placed objects
* **How:** 
  + JavaScript, HTML and CSS were used to build the Web-Application
  + WebXR (ARCore based) was implemented to access AR
  + Example code was used and customized for our problem ([Download](https://github.com/googlecodelabs/ar-with-webxr/archive/master.zip))
  + 3D low-poly models (park bench, park bin, tree) were made using Blender
* **Obstacles:** 
  + No previous knowledge on web AR applications
  + Web AR application only runs on very recent phones ([List](https://developers.google.com/ar/discover/supported-devices))
  + Adding geodata to AR objects needs some additional calculations and code (not done yet)
  + The used ARCore library only works with HTTPS, not on the local network. Therefore we had to use a temporary website (<https://star.cyon.site>)
* **Impression:** 
  + Some impressions of the AR application:



Survey-Access-Ideas:

. OnSite QR-Code

. OnSite Tablet / digital Station

Survey-Ideas:

. 3D (oder 2D) Objekte auf Platz platzieren und verschieben (à la Sims haus bauen)

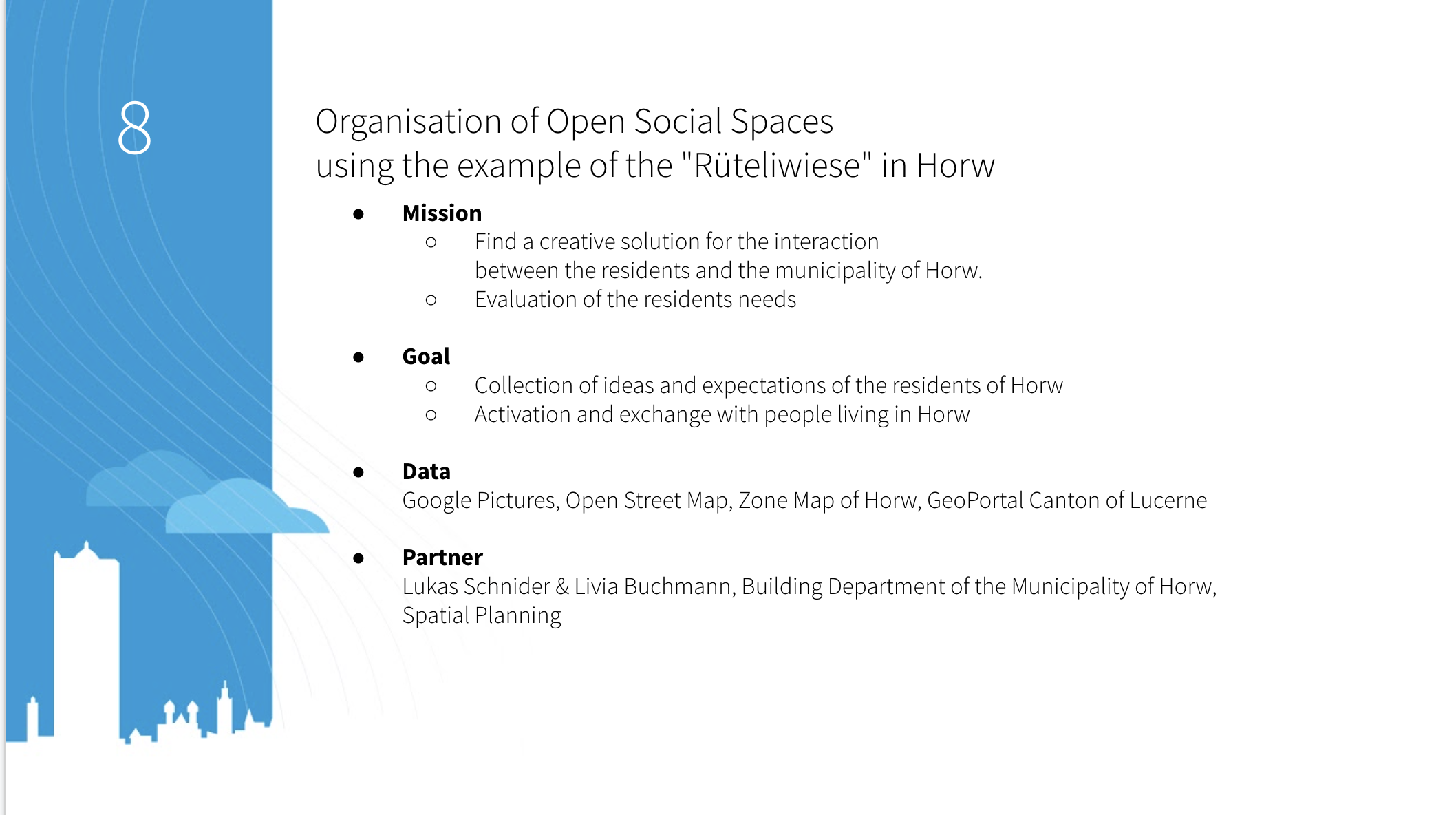
. Wettbewerb / andere Ideen anschauen und bewerten (AR-Event?)

Tools:

. bimObjects

( <https://www.bimobject.com/en-us/product> / <https://www.polantis.com/objects/types>)

. Web- / App-Developement-Tools



Abklärungen:

. Web-Apps vorhanden **(**[**https://github.com/ignacio-chiazzo/ARKit**](https://github.com/ignacio-chiazzo/ARKit)**, WebGLStudio.js, )**

. Sind schon

APIs (js):

* Geodaten (Gino, Andy)  
  <https://wiki.openstreetmap.org/wiki/API>
* 3D Objekte downloaden (Stefan)

<https://archive3d.net/> (Easy Access)

<https://3dsky.org/> (Quality Content)

bim-objects

turbosquid

<https://grabcad.com/>

CGTrader

* AR (Gianni, Stefan)

<https://developers.google.com/ar/discover/concepts>

<https://ar-js-org.github.io/studio/> <https://github.com/AR-js-org>

<https://bitstars.github.io/droidar/>

<http://www.mixare.org>

<http://www.arreverie.com/artoolkit.html>

<https://www.adobe.com/ch_de/products/aero.html>

https://in.pycon.org/cfp/2019/proposals/exploring-augmented-reality-with-python~e0rKb/

<https://bitesofcode.wordpress.com/2017/09/12/augmented-reality-with-python-and-opencv-part-1/>

https://bitesofcode.wordpress.com/2018/09/16/augmented-reality-with-python-and-opencv-part-2/

<https://www.smashingmagazine.com/2016/02/simple-augmented-reality-with-opencv-a-three-js/>

AR-Demo: <https://cdn.soft8soft.com/demo/applications/augmented_reality/augmented_reality.html>

Anleitung:

<https://developers.google.com/web/updates/2018/06/ar-for-the-web>

<https://codelabs.developers.google.com/codelabs/ar-with-webxr#0>

Library:

<https://immersive-web.github.io/webxr/>

* AR-Copy (Andy)  
  Github: https://github.com/cyrildiagne/ar-cptext  
  it’s already implemented into an app: <https://play.google.com/store/apps/details?id=app.arcopypaste&hl=en&gl=US>
* **Outlook**
  + Next steps:
    - What would you need (data, skills, tools …)?
    - Will you keep on working on the project?
  + How to get in touch with you?
  + Link to project in opendata.ch

**Input IBMler**

* Python erzeugt Daten, welche später via Node-RED auf einer Weltkarte angezeigt werden
* Python schickt Daten an Node-RED via API (http POST request)
* Node-RED bietet Usern die Möglichkeit verschiedene Objekte auf der Weltkarte anzuzeigen e.g. Bäume, Bänke usw.
* User wählen anzuzeigende Objekte via Dropdown Menu Node (Dashboard Nodes) aus. Ein Switch Case und Array.filter filtern dann die von Python kommenden Daten basierend auf der User Auswahl.

Hier zwei Links die helfen in JavaScript ein Switch Case und Array.filter zu nutzen:

Switch Case: https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Statements/switch

Array.filter: https://javascript.info/array-methods#filter