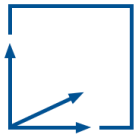
Working in VR: A web-based approach combining 2D and 3D interfaces

Lorenzo Russo da Costa Auer

24.09.2020



Final: Bachelor in Informatics: Games Engineering

Supervisor: Prof. Gudrun Johanna Klinker, Ph.D.

Advisor: Sandro Weber

Motivation

- Increasing accessibility of VR
- Improved perception and learning via immersion
- Web makes it mobile
- Powerful web technologies

Problem Description: Issues

- Render websites in VR
- Interact with websites in VR
- Make interaction enjoyable
- Synchronize accross multiple browsers

Existing Solutions / Related Work

WebVR / WebXR Device API

- Connection between browser and VR headset
- WebVR obsolete
- WebXR standards not finished yet
- Browser only support experimental features

Three.js

- Framework on top of WebGL
- Used for 3D rendering in the browser
- Implements WebXR
- Capable of rendering in VR

A-Frame

- Framework on top of three.js
- Entity-component based
- Integrated Scene Inspector
- Focused on VR content
- No capabilities of rendering websites

Mozilla Hubs

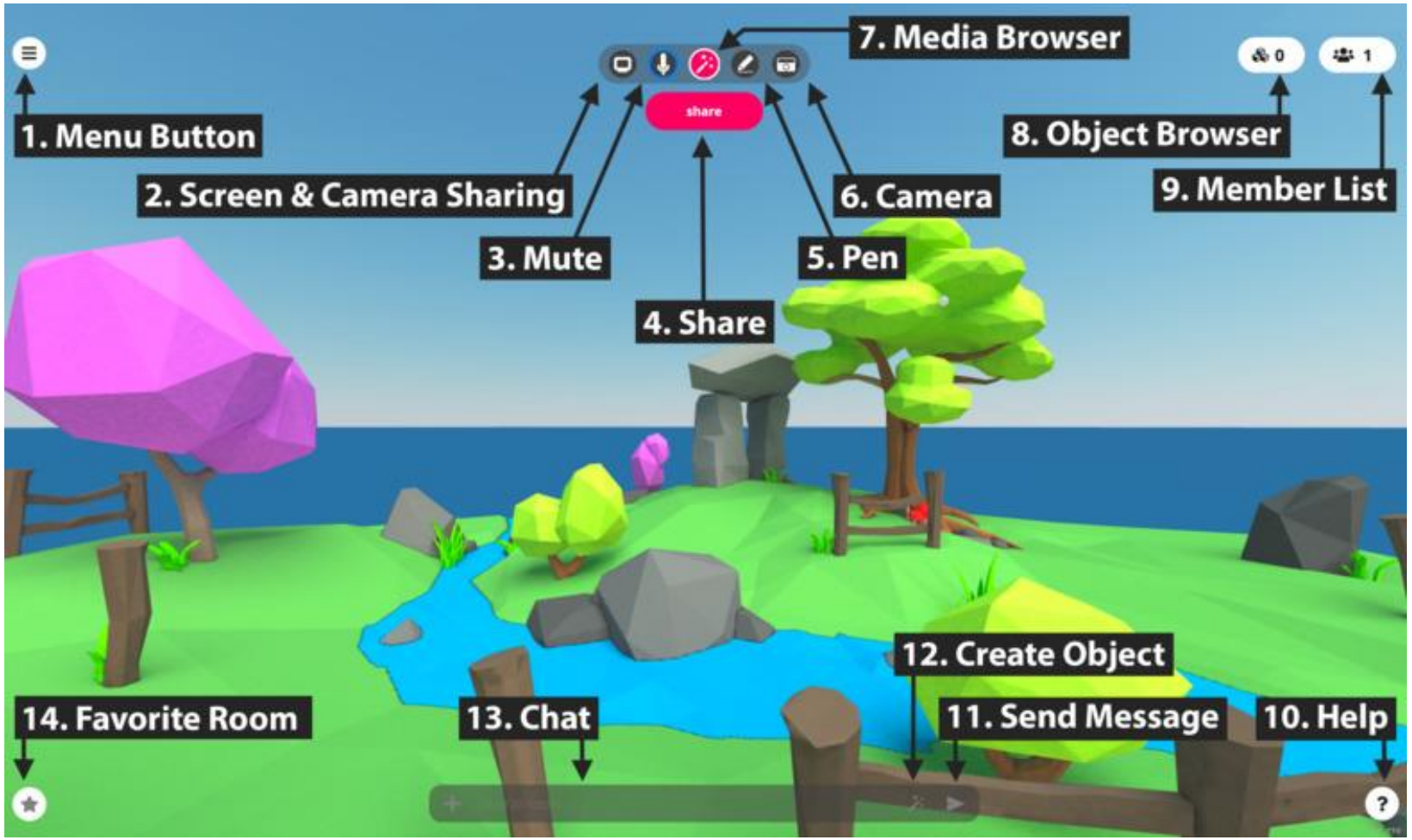


Figure 1: Demonstration of Mozilla Hubs features

Ubi - Interact

- Frontend:
  - Built with Vue.js
  - Contains multiple projects (e.g. VR smartphone keyboard)
  - Contains XRHub which already implements a three.js room
  - Multiple device connectors implemented

Ubi - Interact

- Backend
  - Node.js server
  - Implements websocket connection with Subscription feature
  - Implements storage
  - Message type based processing

Goals of this Thesis

- VR environment containing websites
- Make websites interactive in VR
- Design a good 2D interaction in VR
- Connect multiple VR users to one „XRHub“ and synchronize it

Critical Research Issues

- Create User friendly website interaction in VR
- WebVR/WebXR compatibility between three.js and browsers
- Browser security limiting the possibilities (SOP, CORS)
- Embedding policies of websites

Proposed Work / Approach

- Implement basic VR room
- Add website rendering
- Add website interaction
- Add room sharing and synchronization

Implementation

Implementing the website

- Use of two different renderers (WebGL, CSS3D)
- Placing WebGL plane on top of website
- Make plane transparent
- Enable NoBlending option for the plane

Interacting with the website

- Interaction toggle switching between website and WebGL scene
- M&K interaction with the WebGL scene kept as close to the VR interaction as possible
- No direct DOM access due to SOP

Connecting the clients

- Each „XRHub“ has a unique id
- „XRHubs“ can be entered by URL containing their id
- Changes are published via websocket
- Published information contains the full three.js object
- Website interaction difficult to synchronize due to SOP

Possible Solutions

- Bypass SOP
  - Modifiy DOM of embedded websites and directly insert it
  - Modify headers with a reverse proxy and use CORS
- Only use websites of which you can access the source code
- Wait for stable WebXR version and browser support

Discussion / Suggested Future Work

Implement VR interaction with websites

- Decide which option works best
- Develop a good UX Design for the interaction
- Implement it

Implement more features

- Implement the website implementation in mozilla hubs
- Implement mozilla hubs features in XRHub
- Multiple features realizable by using other websites
- Combine existing Ubii features with the XRHub

Testing

- Counteract regression while project gets bigger
- Can serve as documentation for developers
- Develop solution for automated E2E test in VR

Conclusion

- Using websites in VR is possible
- Different ways to bypass browser security limitations
- Big potential by combining web technologies and VR
- Long way ahead

List of References

1. Figure 1: https://hubs.mozilla.com/docs/hubs-features.html