Implementing a networked VR experience on OpenACS

Antonio Pisano

What the talk is about

- Briefly...
  - How my toy project came to be
  - Main technologies I used for VR on the web
- A little longer...
  - OpenACS/Naviserver features I used
  - Platform-specific challenges and improvements
- Possibly...
  - A live demo!

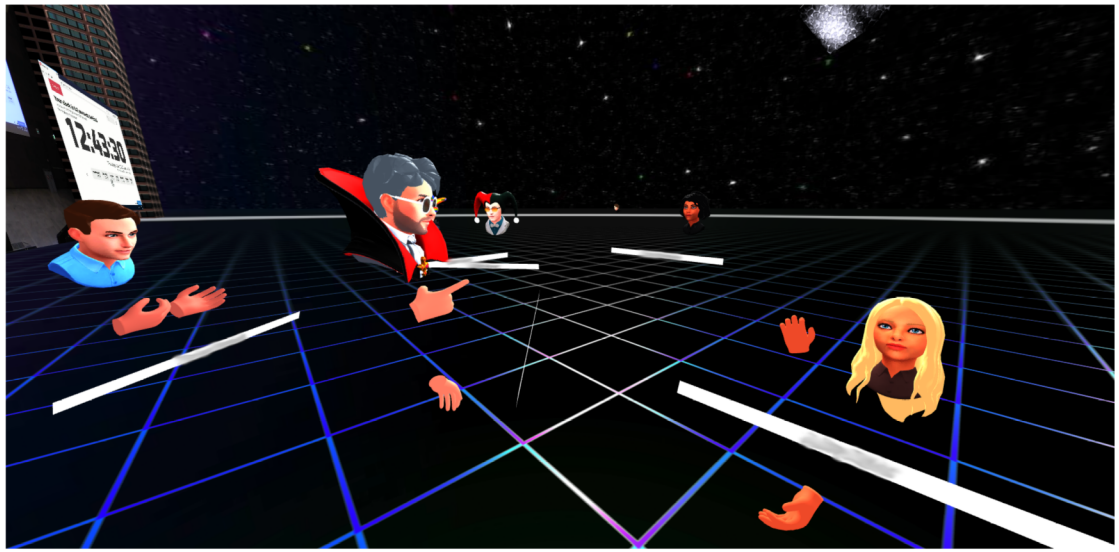
Timeline

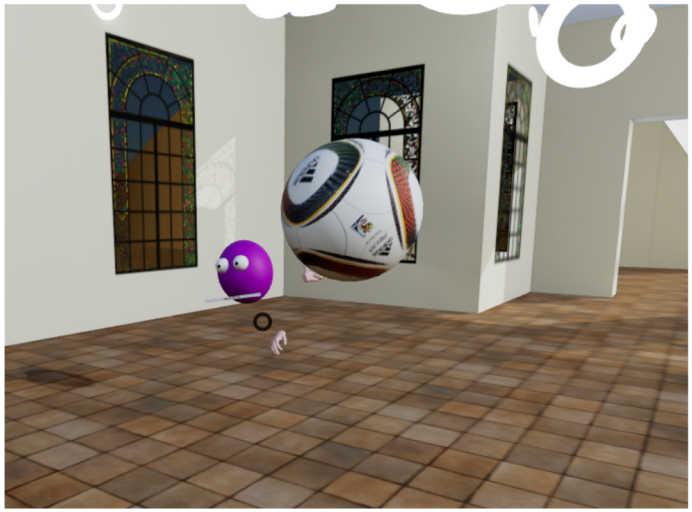
- September 2019
  - I buy myself an Oculus Quest
- March 2020
  - We get to stay at home a lot
- Spring-Summer 2020
  - Bored out of my mind
  - Doing stuff virtually is very trendy
- October 2020
  - First commit in the repo
- Since then
  - Continuous casual improvements

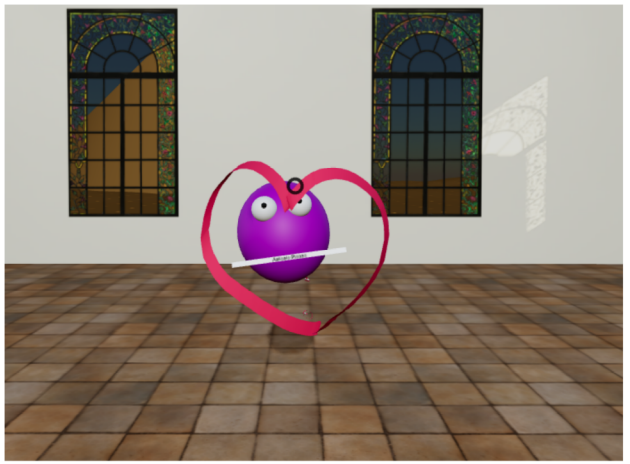
The idea

- A social VR experience in the style of Mozilla Hubs
- “Why not using Mozilla Hubs then?”
  - No self hosting at the time
  - Too complex for my (non-existent) requirements
  - Confident I could get far enough with a smaller footprint
  - Sustainability concerns

The outcome (1)





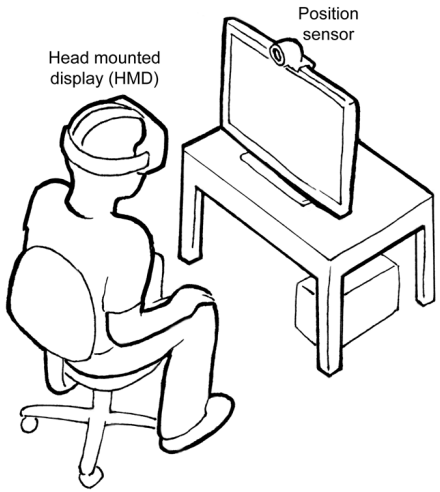


The outcome (2)

- A few cool things work including...
  - Spawn 3D models on the scene
  - WebRTC streaming (using Janus WebRTC Server)
  - Painting
  - Physics
  - Mozilla Hubs scene support
- Mostly untested on a scale
  - Testing on multiple VR headsets is a challenge in itself
  - No real customers so far

The client side - WebXR

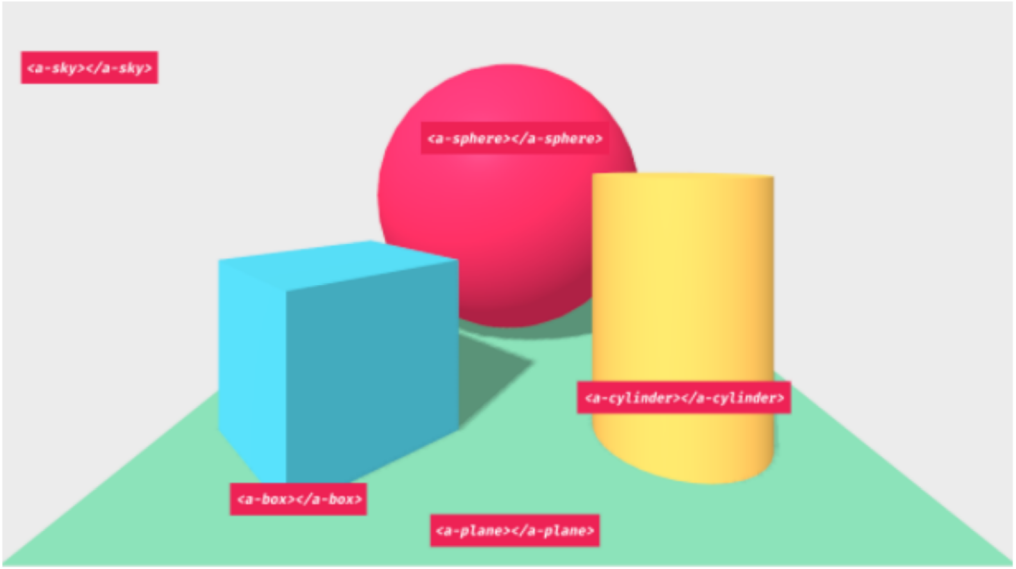
- A set of standards for the web, finalized in 2018
- Supports interacting with devices designed for “eXtended Reality”, including VR headsets
- eXtended Reality = Virtual Reality + Augmented Reality + Mixed Reality…
- Successor of WebVR, conceived in 2014
- Combined with WebGL, enables 3D immersive experiences on the web



The client side - A-Frame

- Abstraction on top of three.js and WebXR
- 3D objects are described using HTML tags and attributes
- Supports both desktop and headset devices

```
<html>
  <head>
    <script src="https://aframe.io/re
  </head>
  <body>
    <a-scene>
      <a-box position="-1 0.5 -3" rot
      <a-sphere position="0 1.25 -5"
      <a-cylinder position="1 0.75 -3
      <a-plane position="0 0 -4" rota
      <a-sky color="#ECECEC"></a-sky>
    </a-scene>
  </body>
</html>
```



The server side – Websockets (1)

- In a nutshell
  - Changes originating locally are broadcast to the other peers
    - Position, rotation, hand gestures…
  - Updates from the peers update our local representation
    - Rotate, translate the peer avatar/hands…
- Websockets on NaviServer
  - Supported via the websocket module since 2015 (4.99.7)
  - Based on the ns_connchan command
  - Initially, building the websocket message happened in Tcl

The server side – Websockets (2)

- Network traffic is generated whenever somebody moves
  - On desktop → not touching the keyboard = no traffic
  - On a headset → you always move… and you also have hands!
  - One headset >= 1 websocket message per screen refresh at all times
- Browser will start showing different behavior in such conditions
  - Message segmentation: 1 channel read → partial message
  - Buffering: 1 channel read → multiple messages
- General network problems add to the mix
  - Partial channel read/write, unexpected close
  - TLS

The server side – Websockets (3)

- NaviServer websocket implementation in 2020 needed some love
  - No support for segmented messages at the protocol level
  - No support for partial read and write operations at the channel level
  - Partial reads + appending binary content at the Tcl level → prone to unwanted UTF-8 conversions → corrupted messages
- Gustaf Neumann rewrites most of the module December 2020
  - C-Level interface to handle websocket messages
  - Handling of partial read/write operations
  - Handling of segmented messages
  - Faster performances

The server side – OpenACS features

- Site node system
  - Every package instance can be configured as a separate VR experience
  - Instances can be mounted in the context of different subsites targeting different cohorts
- Permissions and parameters
  - Some features have been connected with the user’s level of privilege
  - Most features can be enabled/disabled/configured via parameters
- Includes and Virtual URL Handlers
  - The include mechanism and vuh files enable self-contained environments
  - Clear separation between global and environment-specific resources

Thanks for watching!

- My contacts
  - antonio@elettrotecnica.it
  - https://github.com/Elettrotecnica
- Project link
  - https://github.com/Elettrotecnica/aframe-vr

Time for a demo?