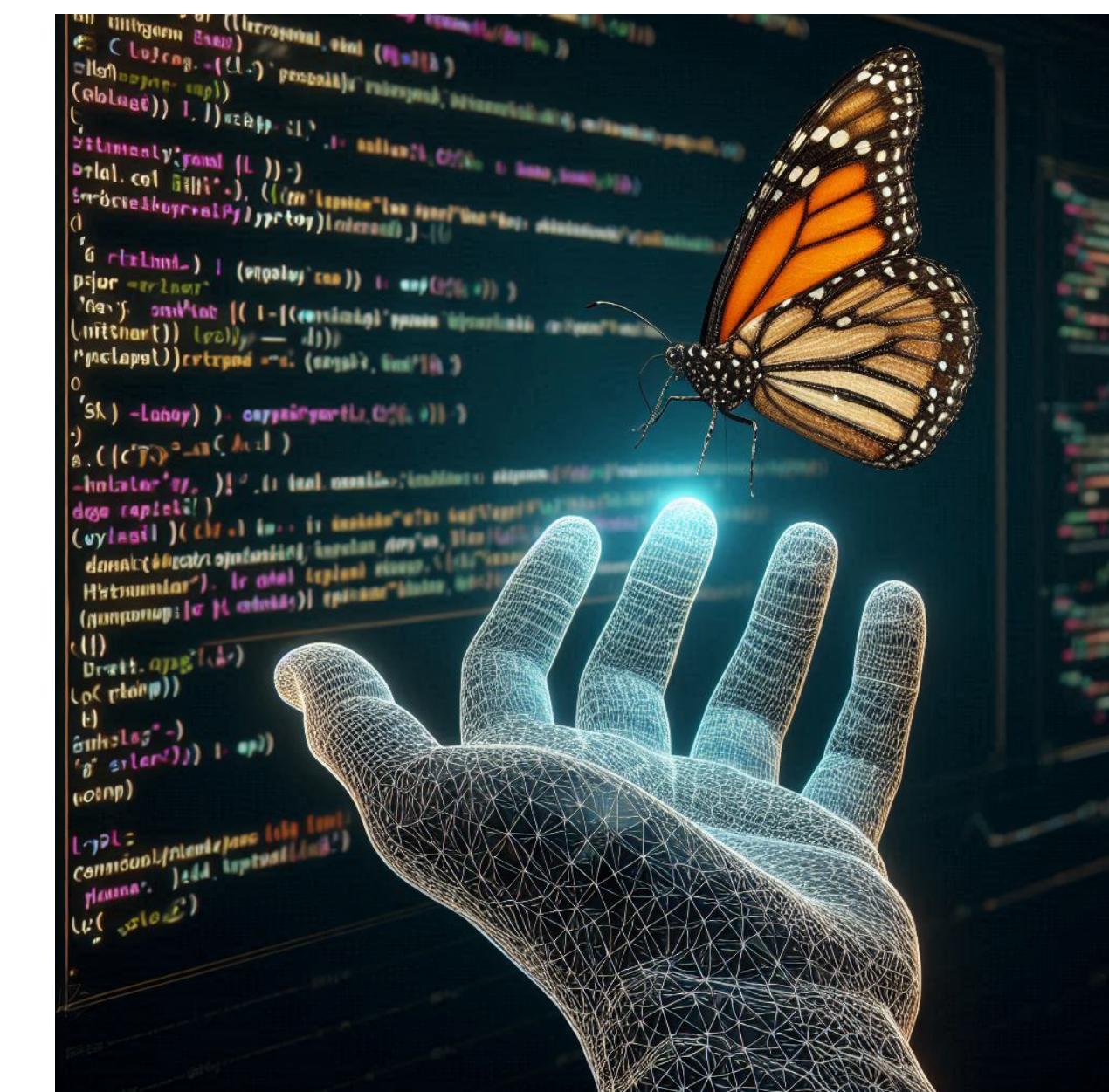


My wish

Butterfly

- program a moving 3d butterfly in javascript
- spawn it in vr trough a spatial volume overlay
- attach it to a coordinate and worldid in the base layer app
- someone else comes along into that world later, sees the moving butterfly, and is able to look at the original source code
- improve the code of the butterfly together with someone, in vr



Picking up from aardvark



- no base app integration
- networking came late in development
- used a custom rendering system based on aarvark-react
not sure if was possible to use three, pmndrs etc

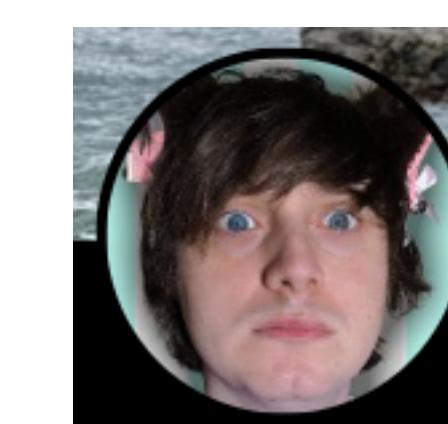


-OSC integration worldspace coordinate

(already a go for VRChat Resonite, probably chillout and others!)

- designing from the ground up with decentralized networking
p2p webworker webrtc actors

- webxr volumes? probably somewhat unsupported.
OpenVR render volumes are a go

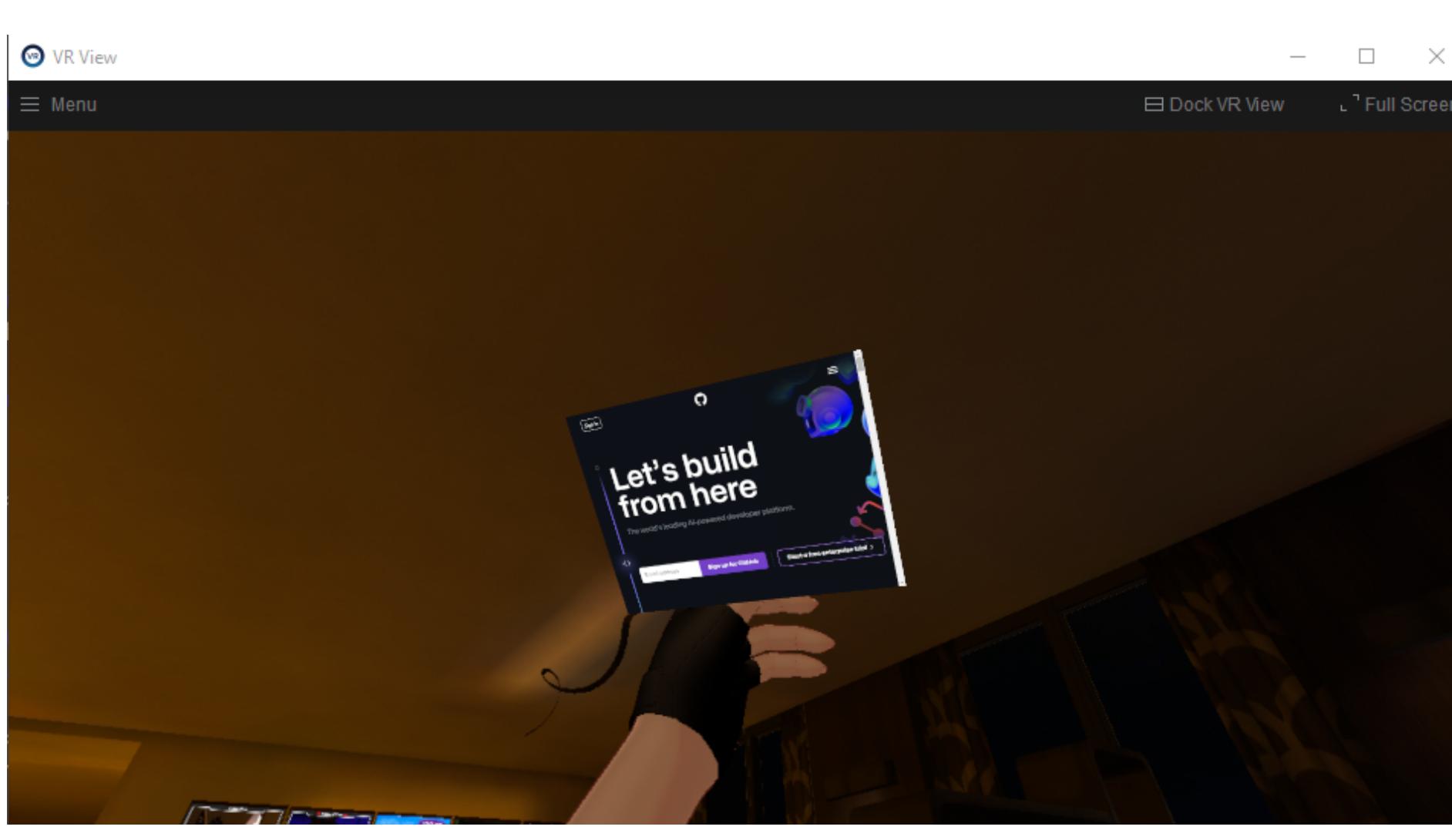


What needs to be improved

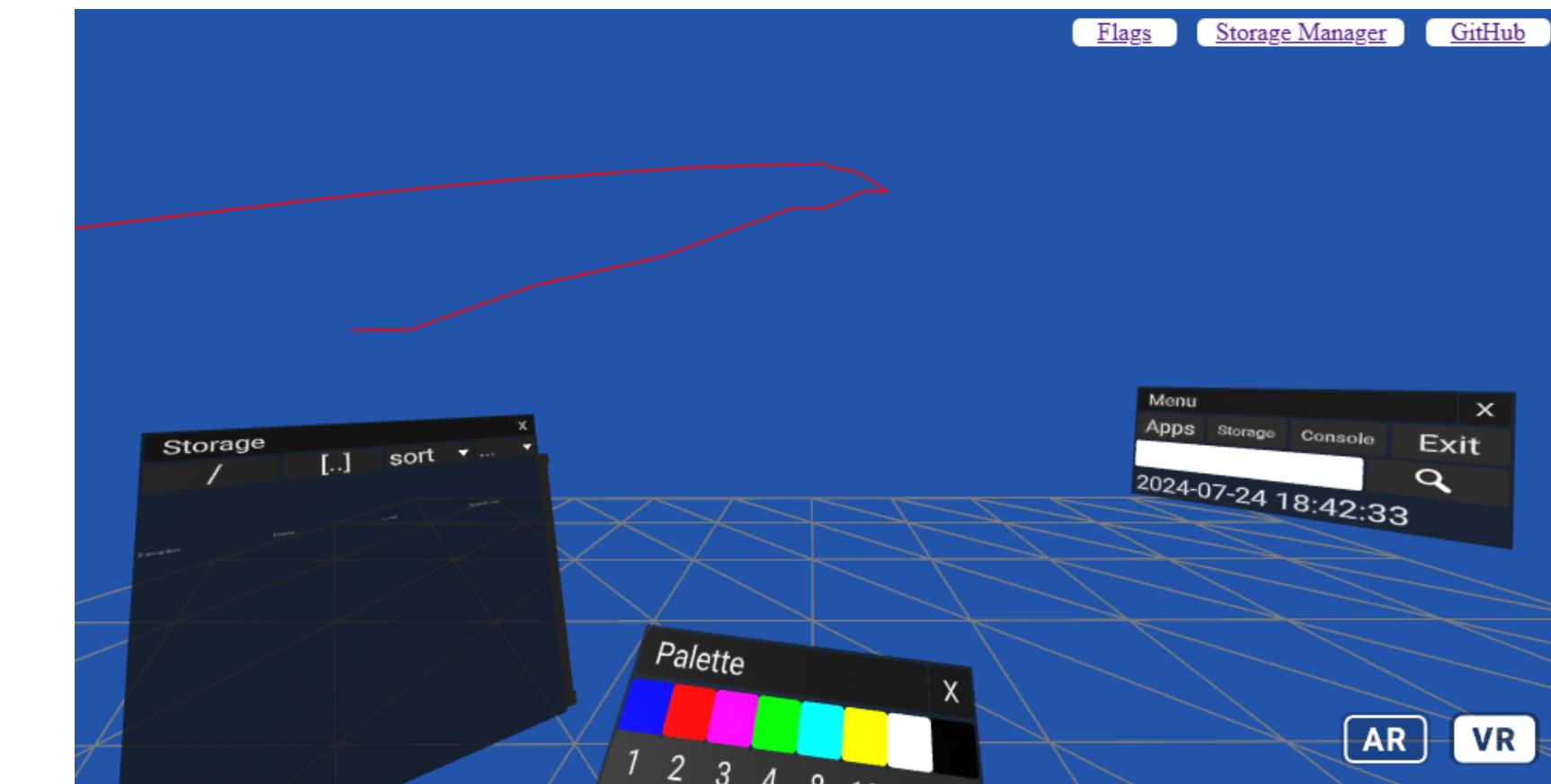
openxr overlays?

openvr overlays need more work
depth info, volumes

webxr iframe? nesting
webxr pc support
webxr entry

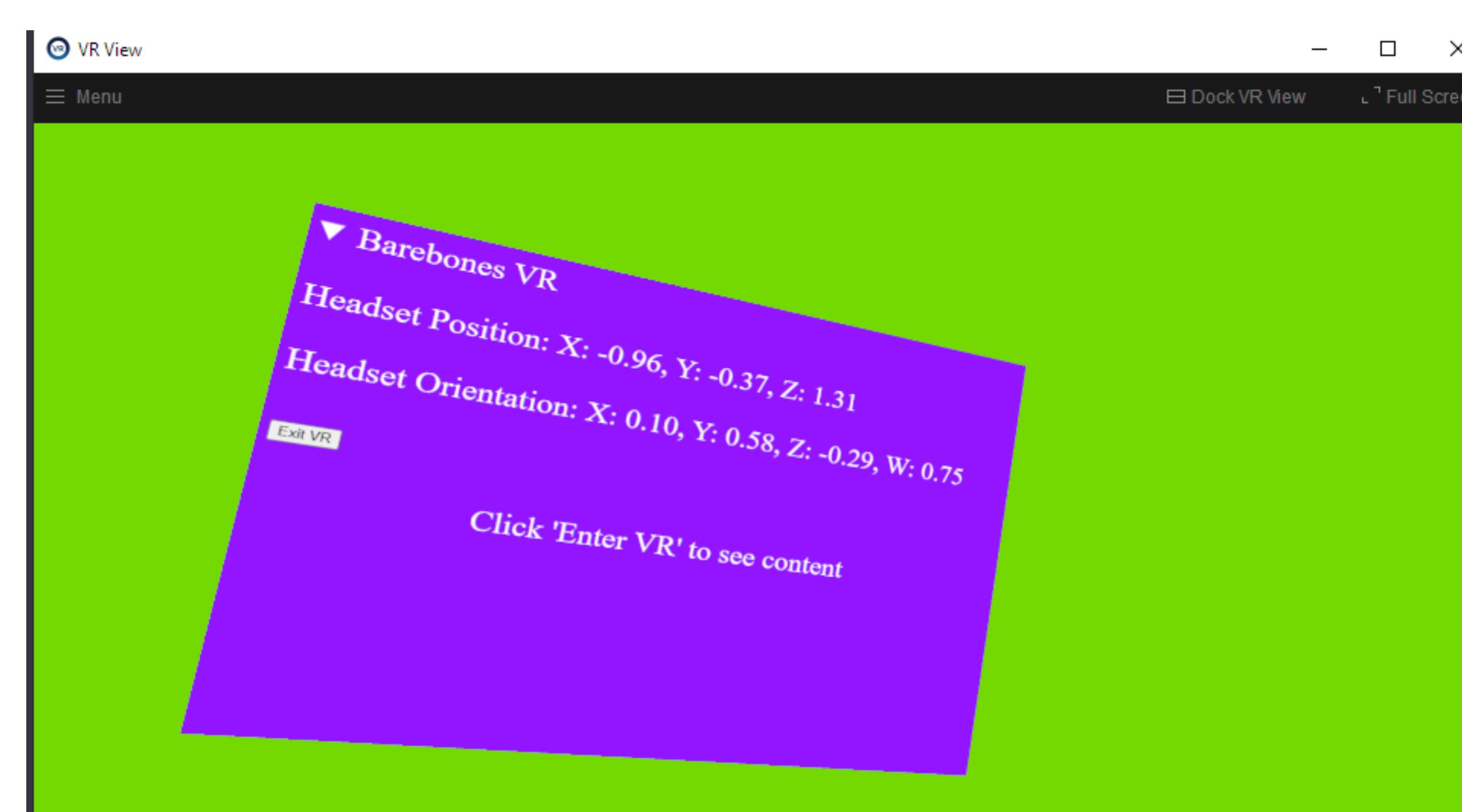
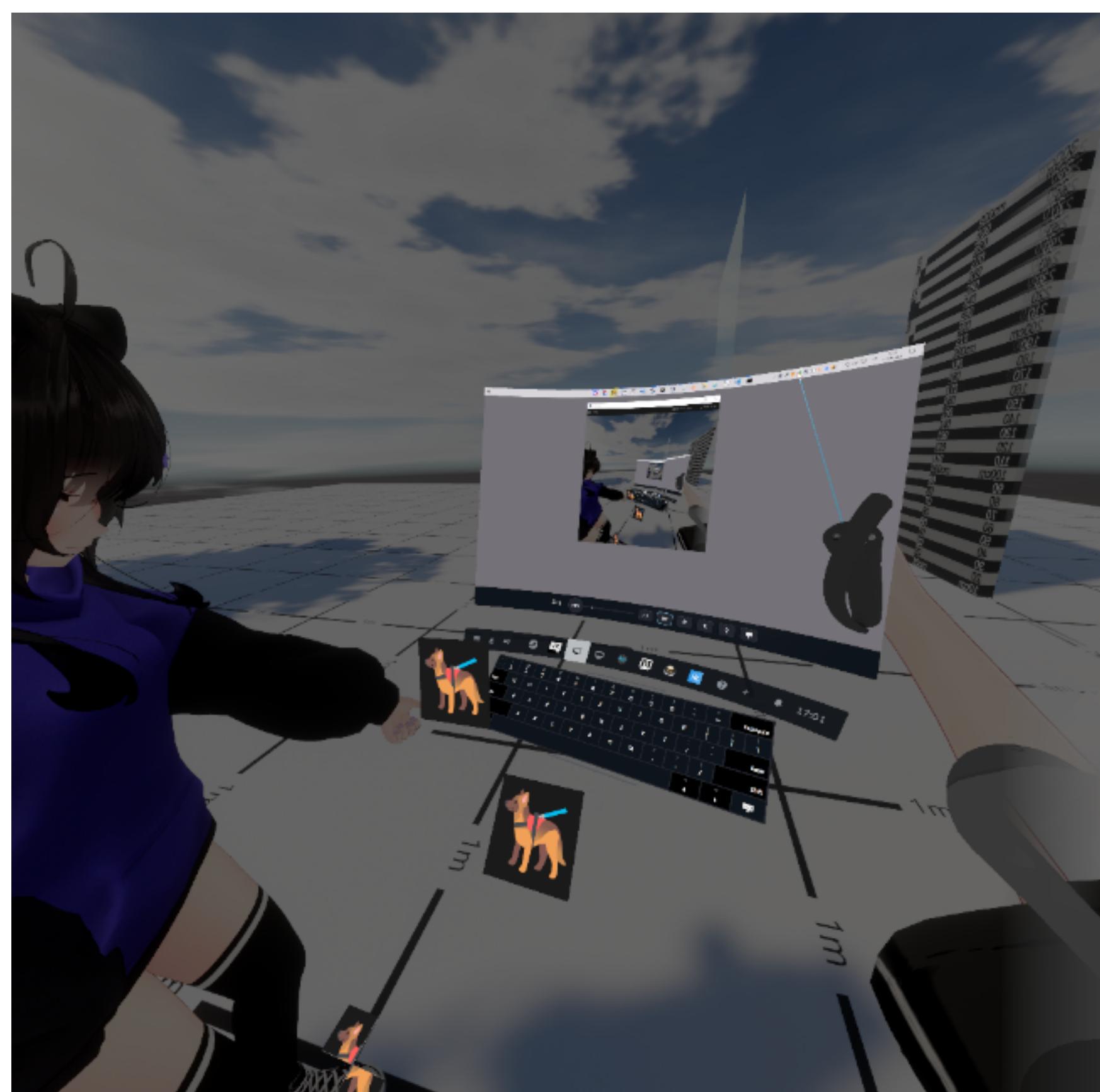


btw, probably also stardust



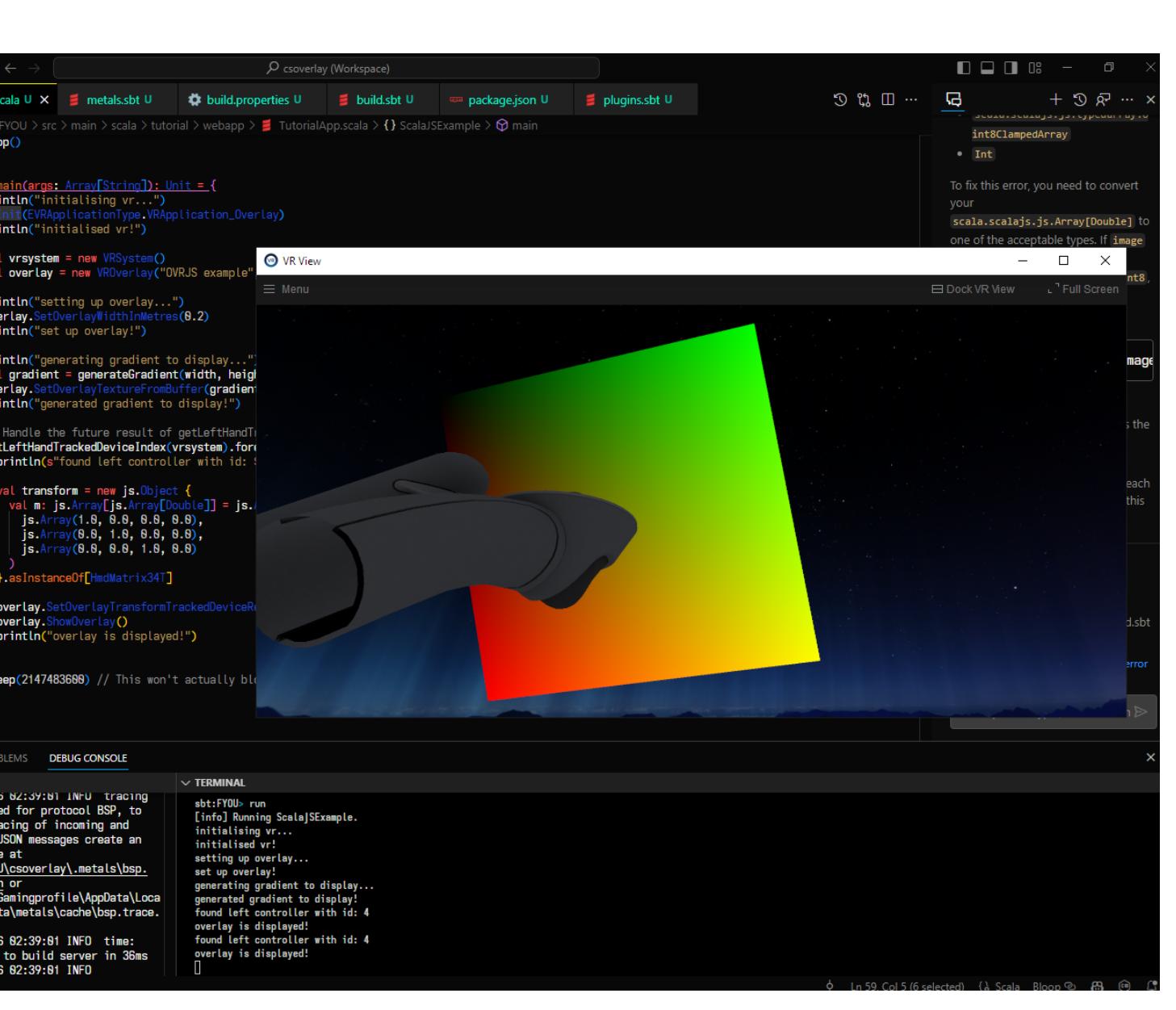
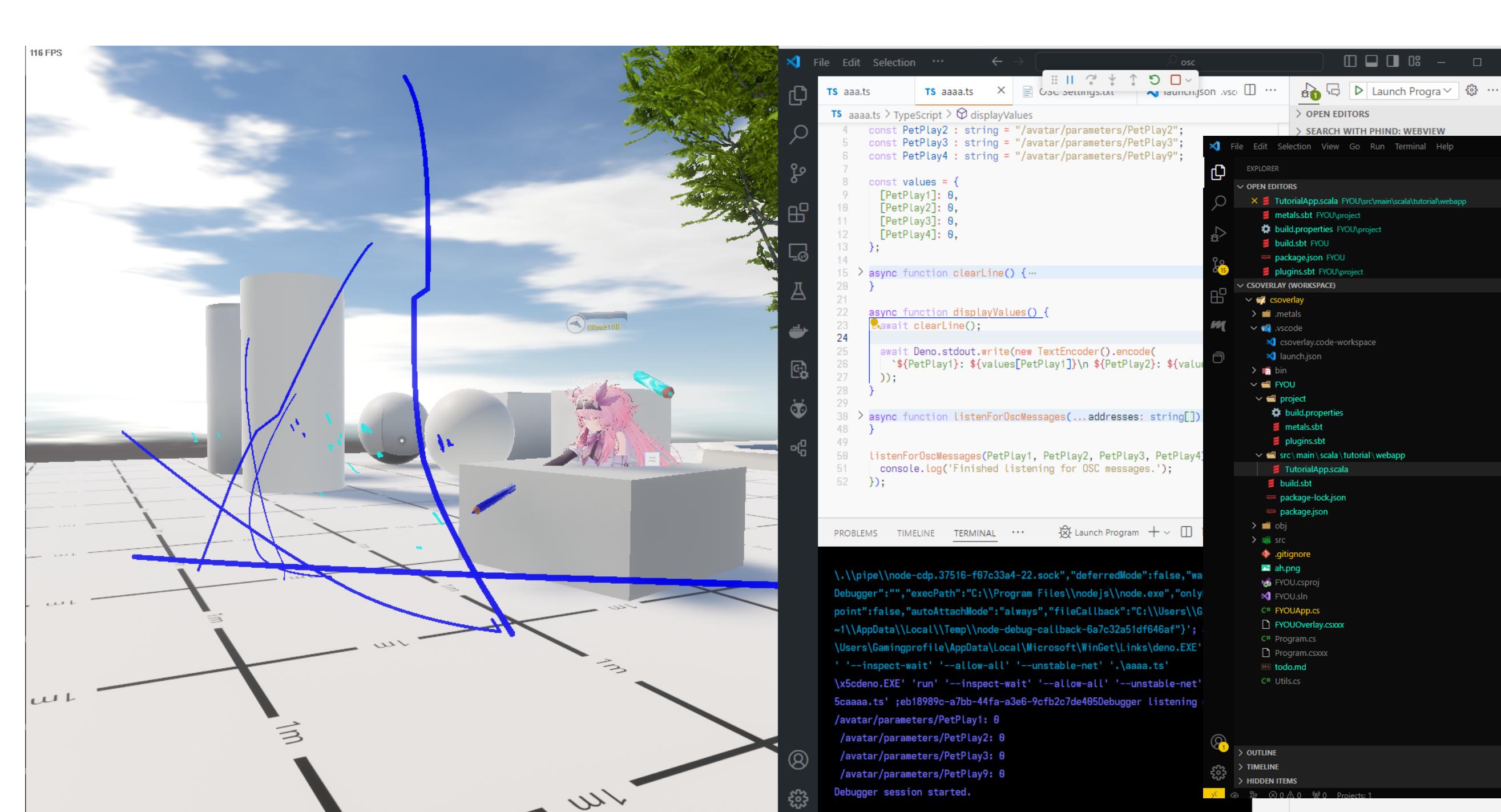
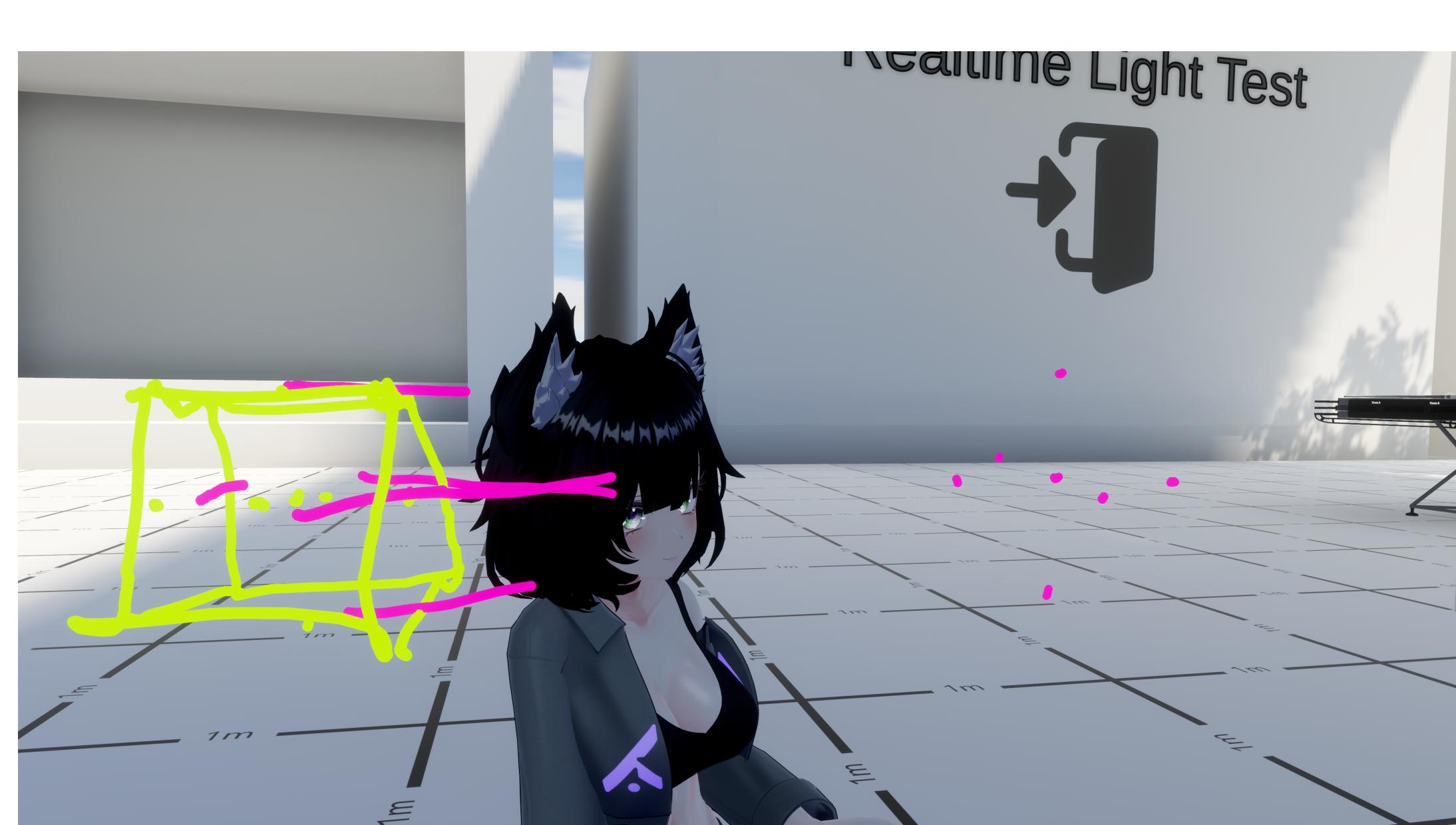
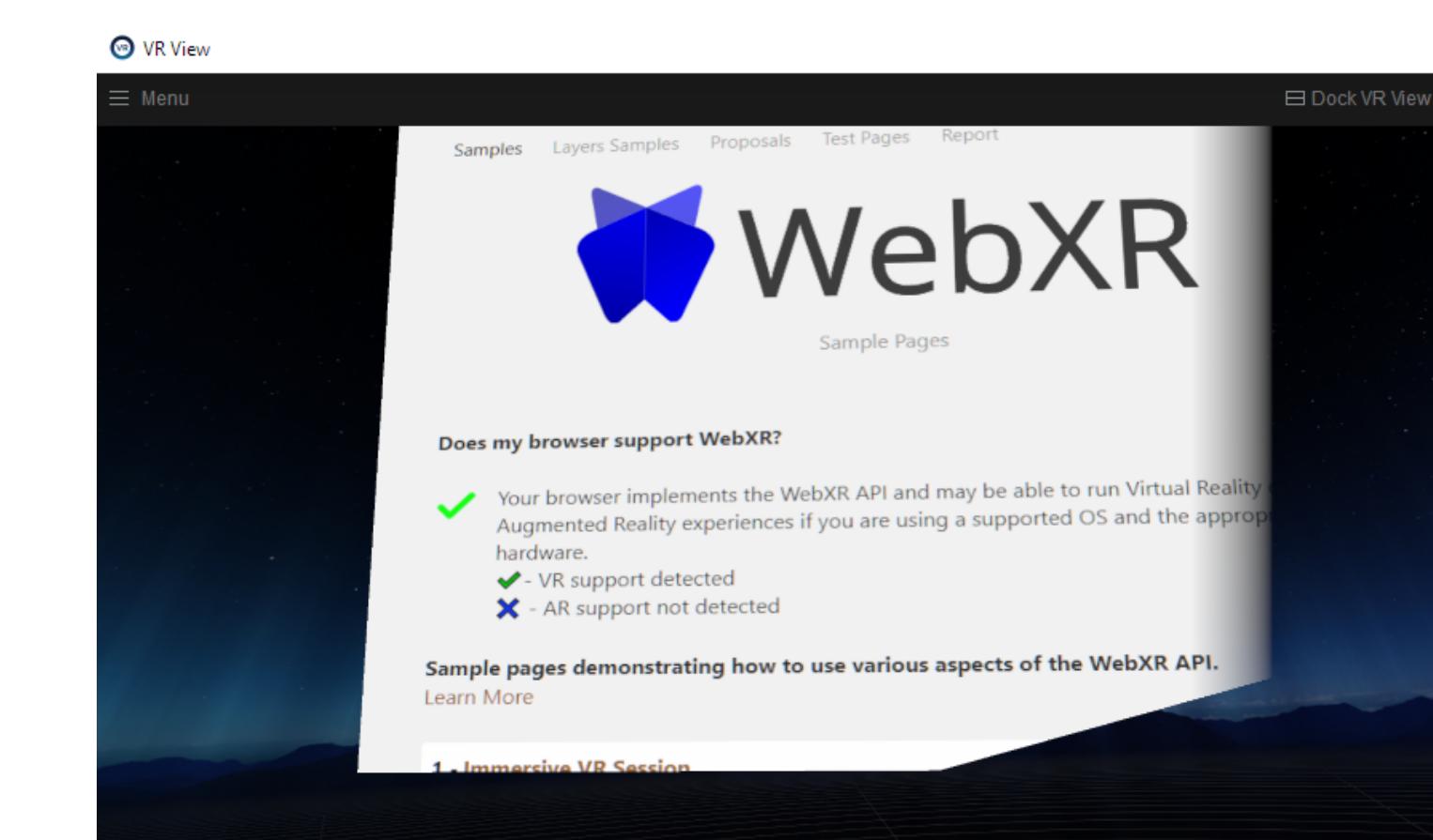
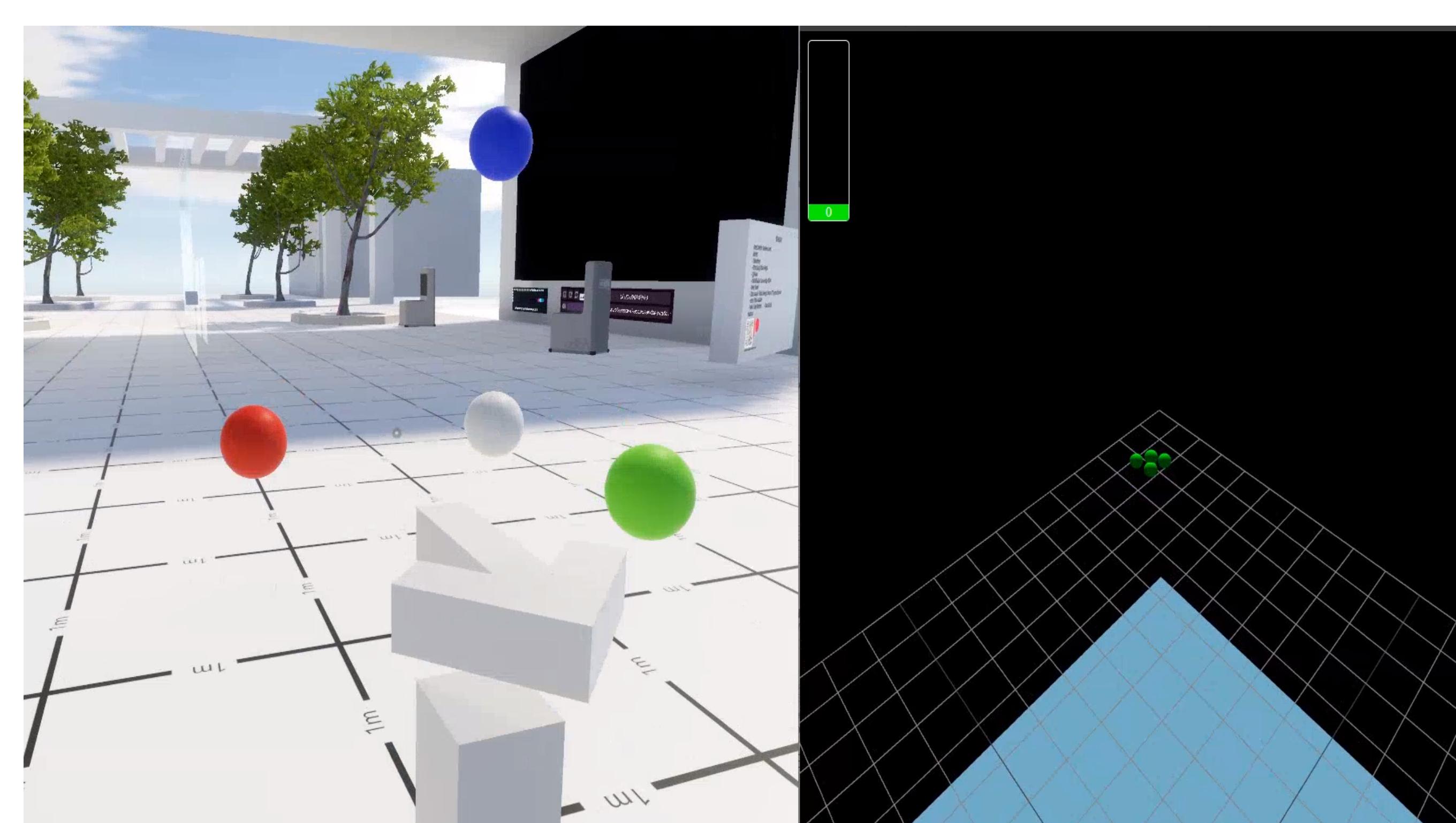
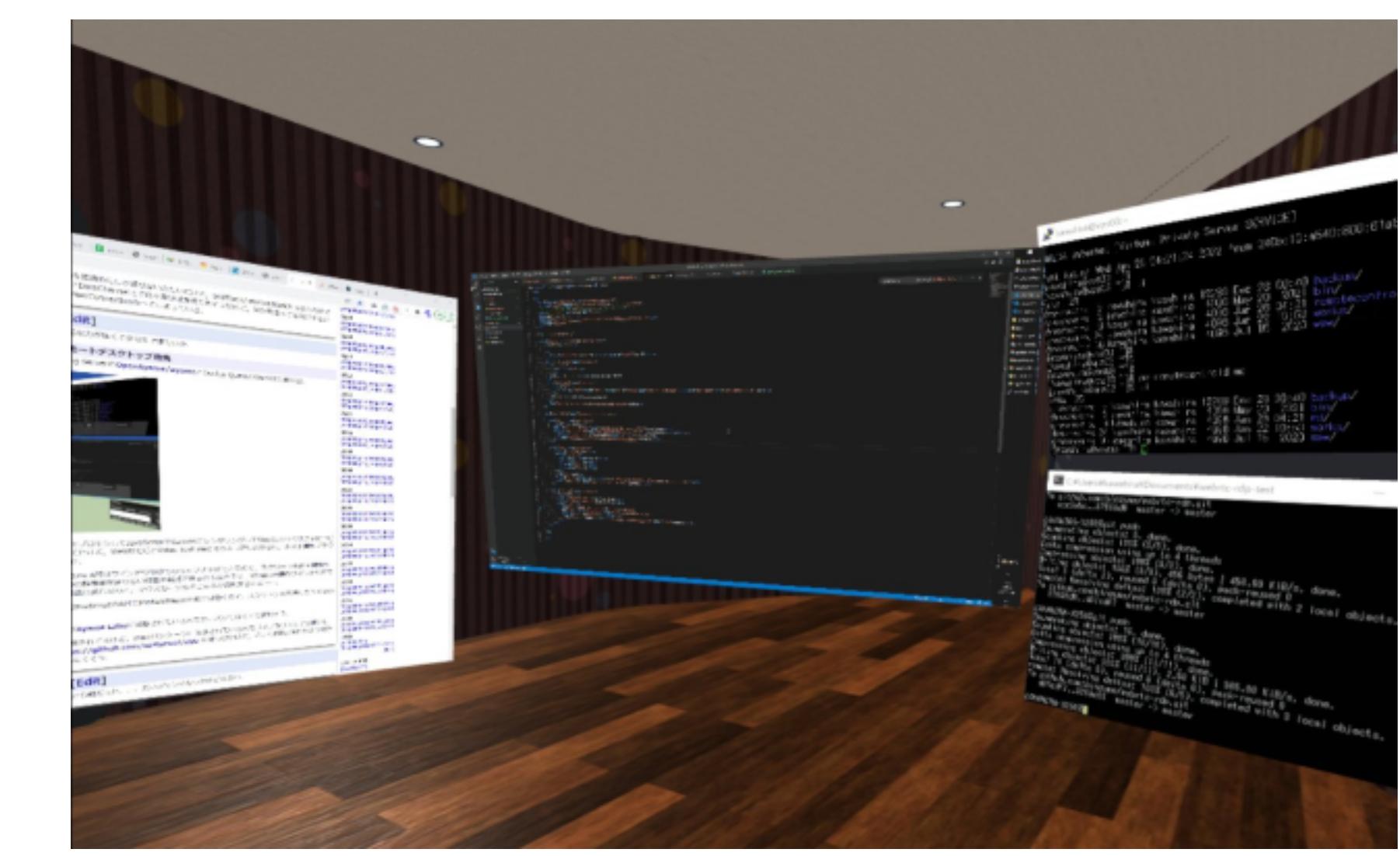
vrchat apis and osc already offer

- world position, shoutout
- what world you're in
- who you're friends with

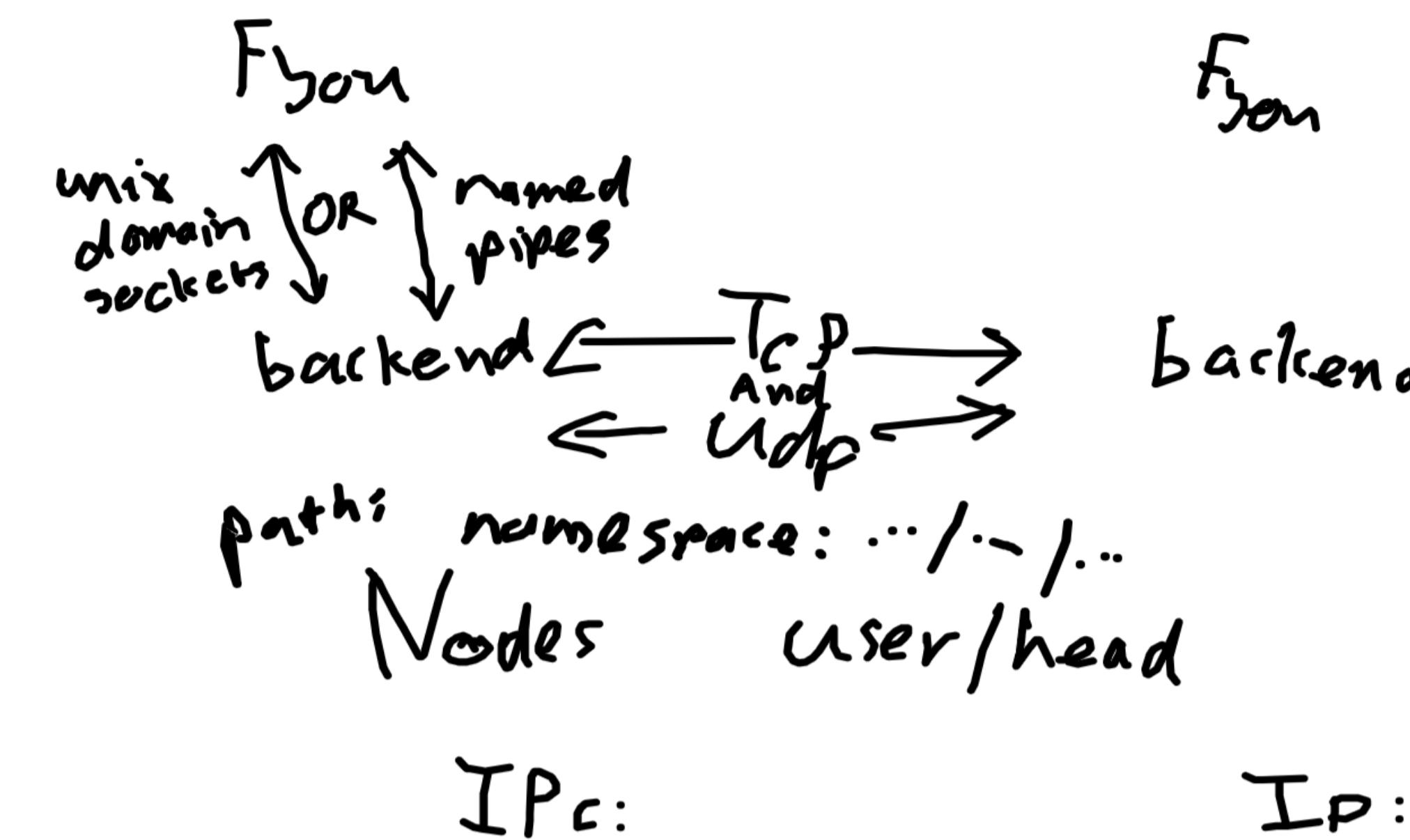


stuff I wanna work on

- modular networking system, vrchat integration
- desktop sharing overlay rdp
- overlay volumes like aardvark/exokit
- porting openvr api to typescript, Deno(node) FFI



Le fyou Protocol ~



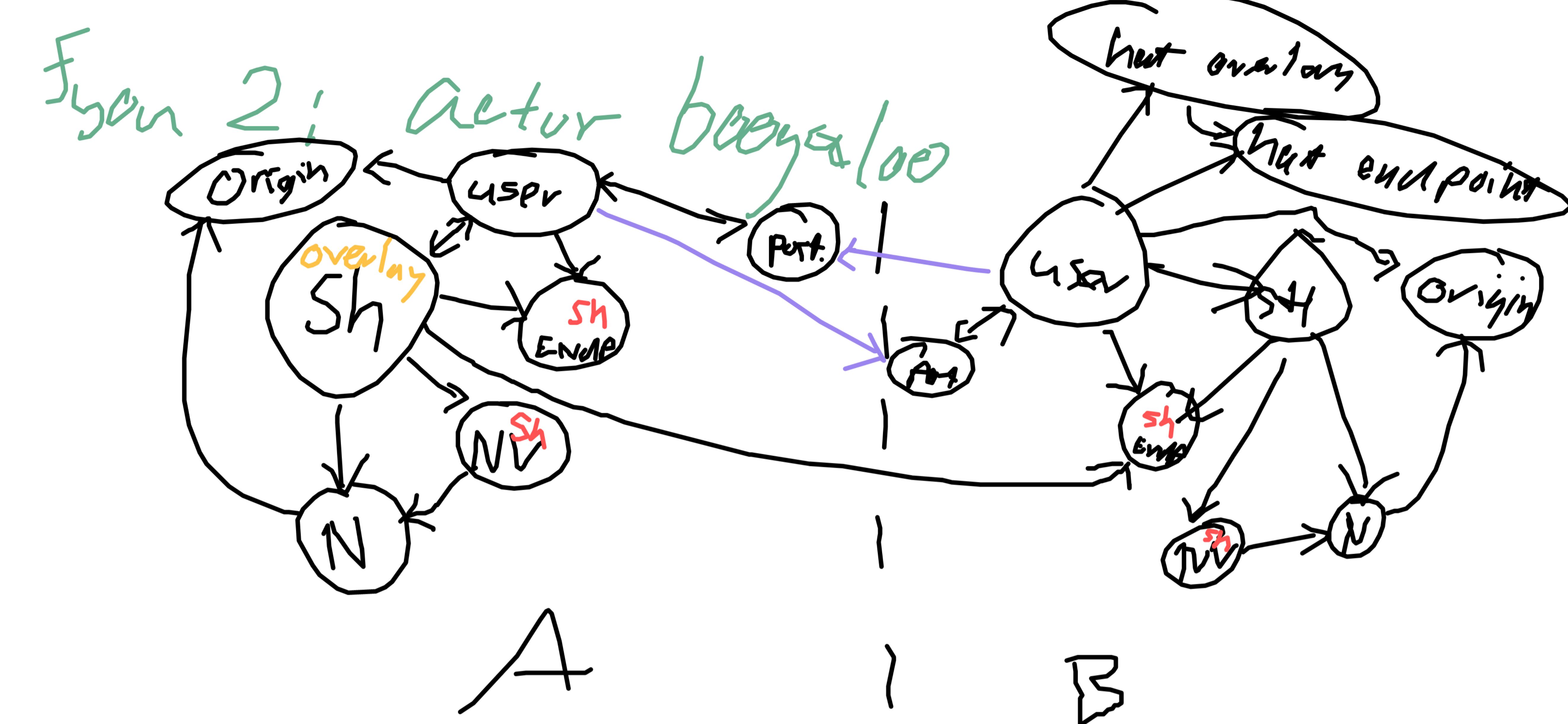
Icp → Connect "window"
← handshake "user"
→ create "user/window/..."

IP → create "user/window/..."
Icp ← Create "user/window/..."

addressnamespace/useridnamespace/fyounamespace/appnamespace
localhost:ip/potatouser/fyounamespace/window

IP:
- Connect a → b * namespace
- handshake b → c * namespace
- announce b → t
- welcome t → a
- handshake a → t

- Subscribe * path
- Node position data
 - * path
 - * position
 - * rotation
- Unsubscribe
 - * path
 - * t
- Ping
- Node Created
- Node destroyed



Node

- k/11
- Set position endpoint
- List ALL nodes → List<NV> addr

Sh endpoint

- List windows → List<ShNV> addr

NV - subscribe addr

- get position/rotation/size addr

Sh NV

- Subscribe addr

Portal

- provide Overlays addr
- List<Endpoint>

