

An Introduction to WebVR

or

How to make your user sick in 60 seconds.

You should **care** about **WebVR**.



Hardware & Concepts



Hardware & Concepts

```
scene = new THREE.Scene();
let camera = new THREE.PerspectiveCamera( 75, window.innerWidth / window.innerHeight, 0.1, 1000 );
let renderer = new THREE.WebGLRenderer();
renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );
let geometry = new THREE.SphereGeometry( 500, 60, 40 );
geometry.scale( -1, 1, 1 );
let video = document.createElement( 'video' );
videoTexture = new THREE.VideoTexture( video );
videoTexture.wrapS = videoTexture.wrapT = THREE.RepeatWrapping;
videoTexture.repeat.set( 10, 10 );
videoMaterial = new THREE.MeshBasicMaterial( { map: videoTexture } );
mesh = new THREE.Mesh( geometry, videoMaterial );
function render() {
    requestAnimationFrame( render );
    renderer.render( scene, camera );
    renderer.render( scene, camera );
}
```

WebVR API



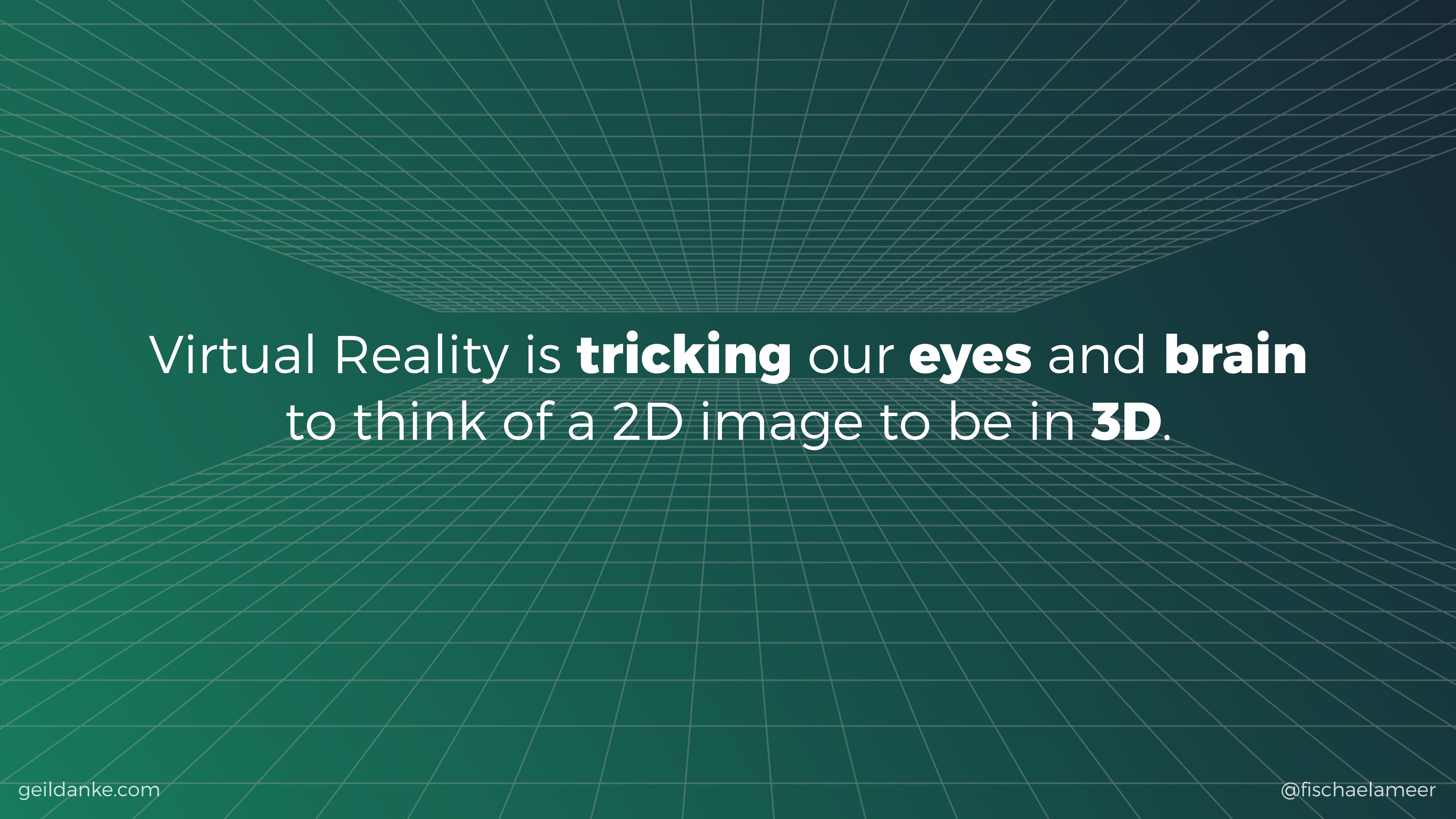
Hardware & Concepts

```
scene = new THREE.Scene();
let camera = new THREE.PerspectiveCamera( 75, window.innerWidth / window.innerHeight, 0.1, 1000 );
let renderer = new THREE.WebGLRenderer();
renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );
let geometry = new THREE.SphereGeometry( 500, 60, 40 );
geometry.scale( -1, 1, 1 );
let video = document.createElement( 'video' );
videoTexture = new THREE.VideoTexture( video );
videoMaterial = new THREE.MeshBasicMaterial( { map: videoTexture } );
let mesh = new THREE.Mesh( geometry, videoMaterial );
function render() {
    requestAnimationFrame( render );
    renderer.render( scene, camera );
}
render();
```

WebVR API



UX Design & VR



Virtual Reality is **tricking** our **eyes** and **brain**
to think of a 2D image to be in **3D**.

A landscape photograph of a coastal area with mountains in the background. A large, semi-transparent white grid is overlaid on the image, extending from the foreground towards the horizon. The grid consists of numerous thin, light-colored lines forming a perspective view.

Virtual Reality **changes** the
way we **relate** to **technology**.

Google Cardboard



Samsung Gear VR



Google Daydream View



Mobile VR

Desktop VR

Standalone VR

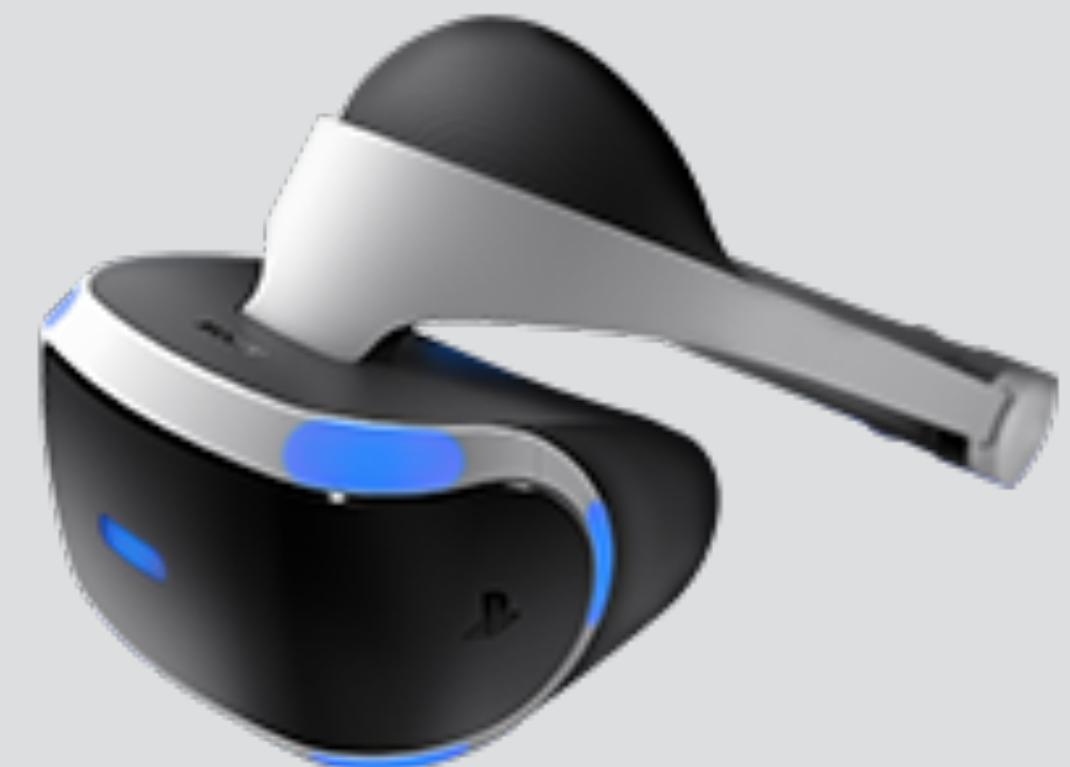
Oculus Rift



HTC Vive



Playstation VR



Mobile VR

Desktop VR

Standalone VR



Mobile VR

Desktop VR

Standalone VR

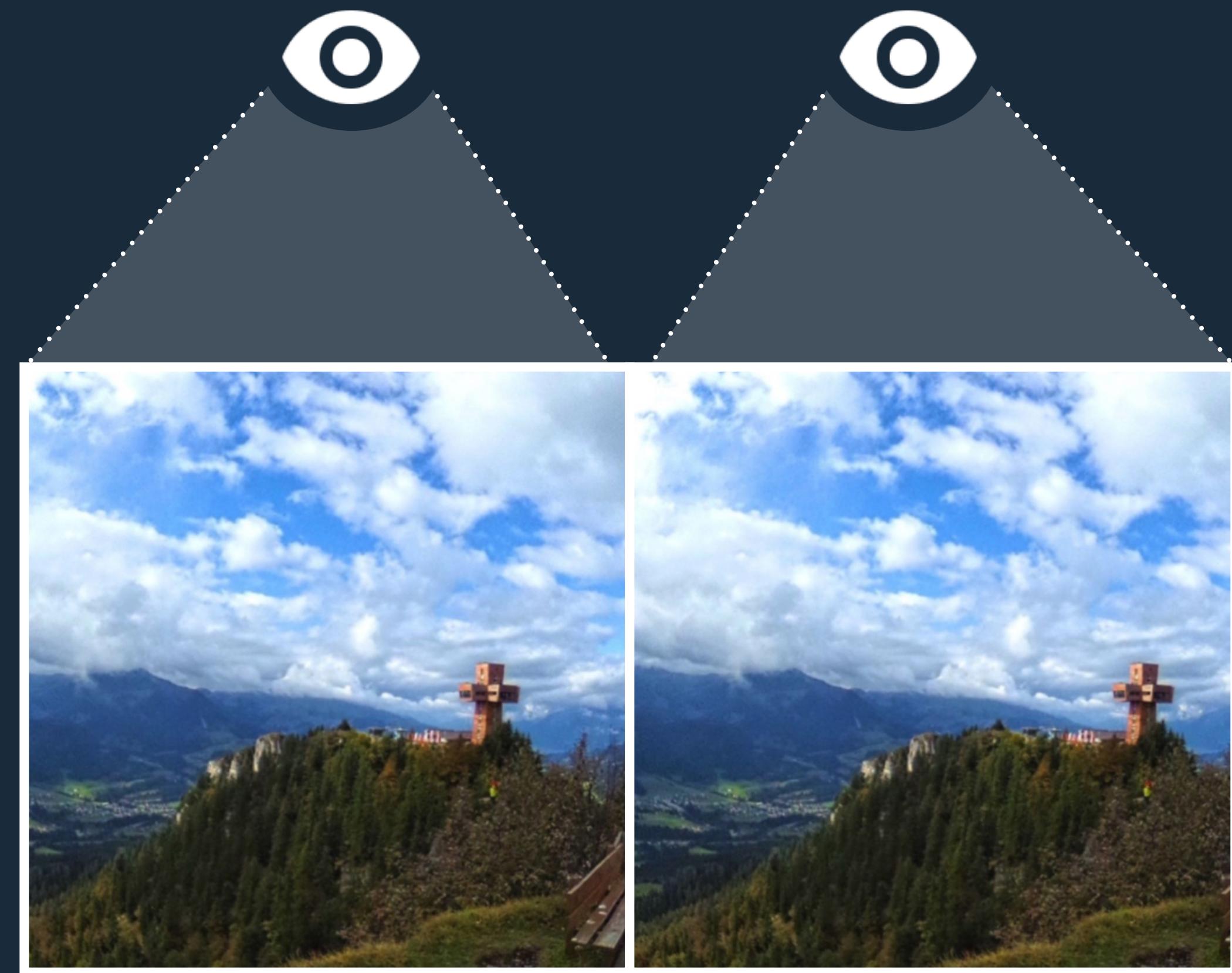
Virtual Reality Concepts

A landscape photograph of a coastal area with mountains and a grid overlay.

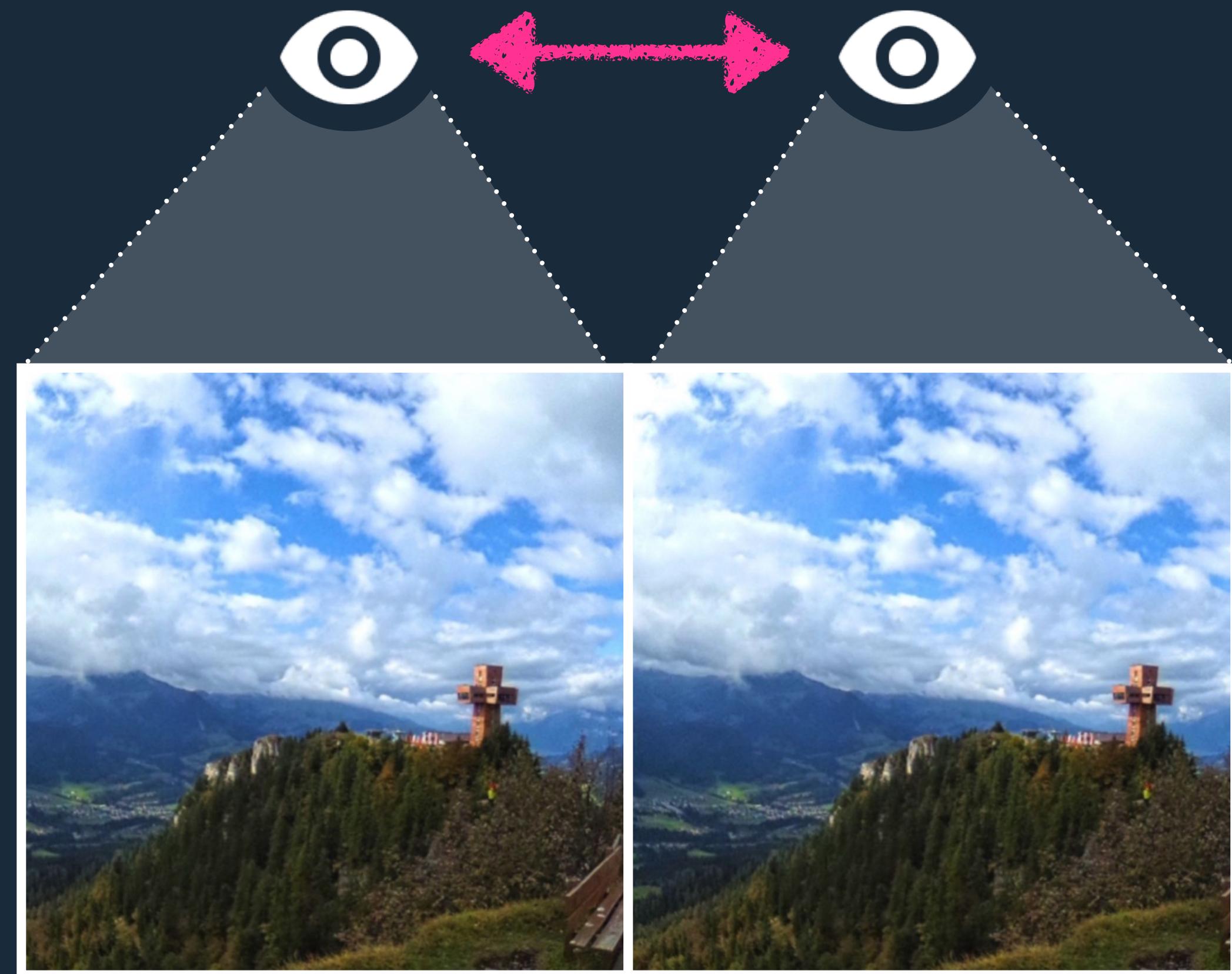
Stereoscopic Images





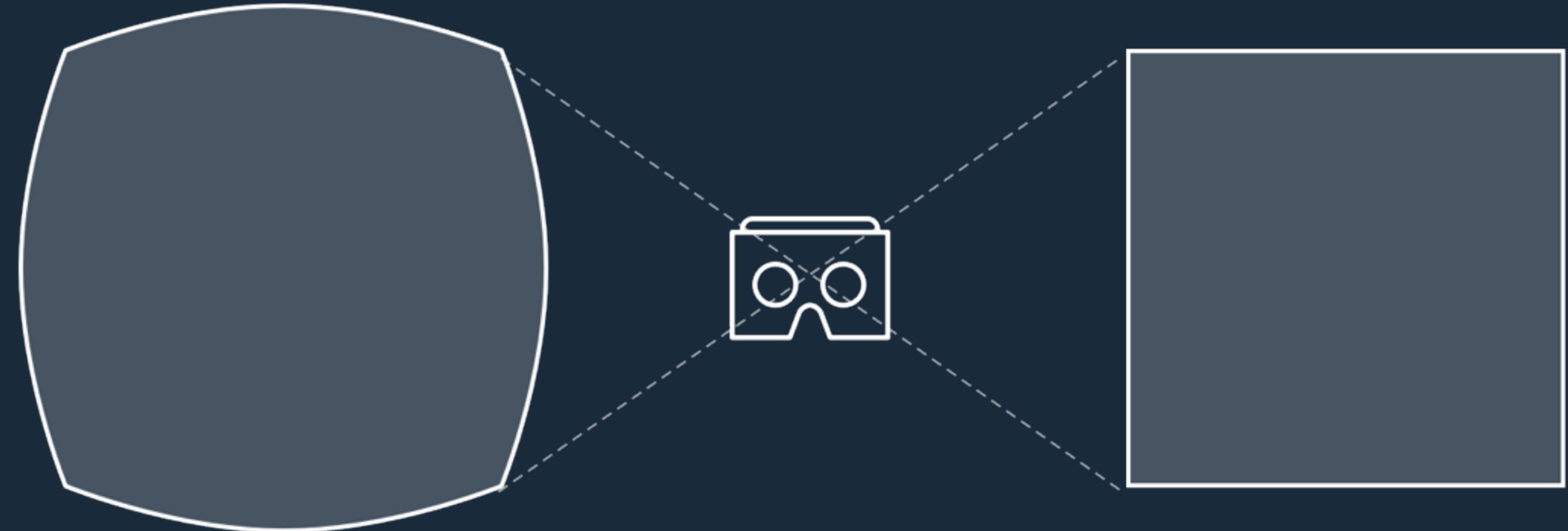


IPD = Interpupillary distance









Tracking





Rotation



Position



Rotation





Position



Rotation

Browser

WebGL

Browser

WebGL

WebVR

Browser

Ricardo Cabello

three.js

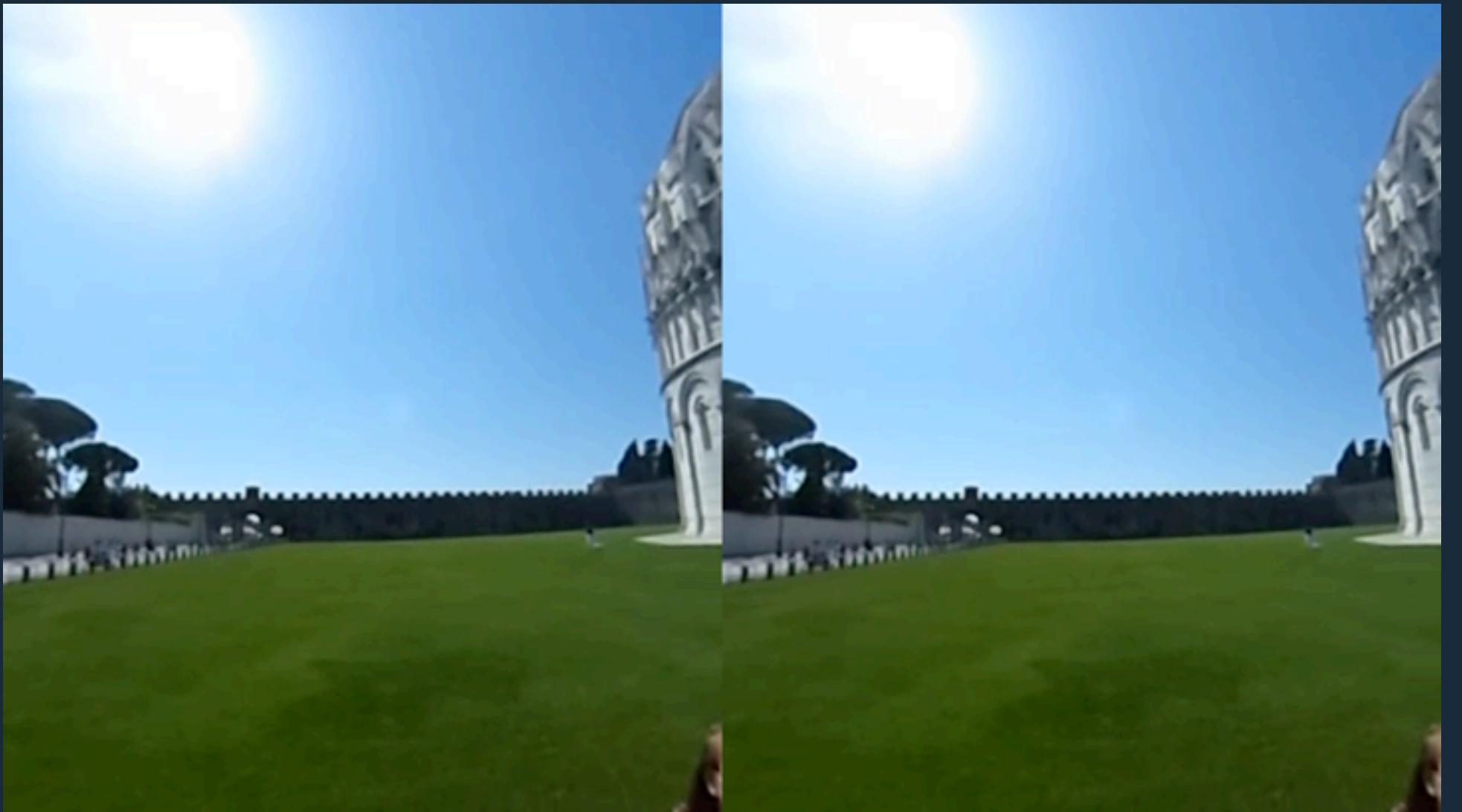
WebGL

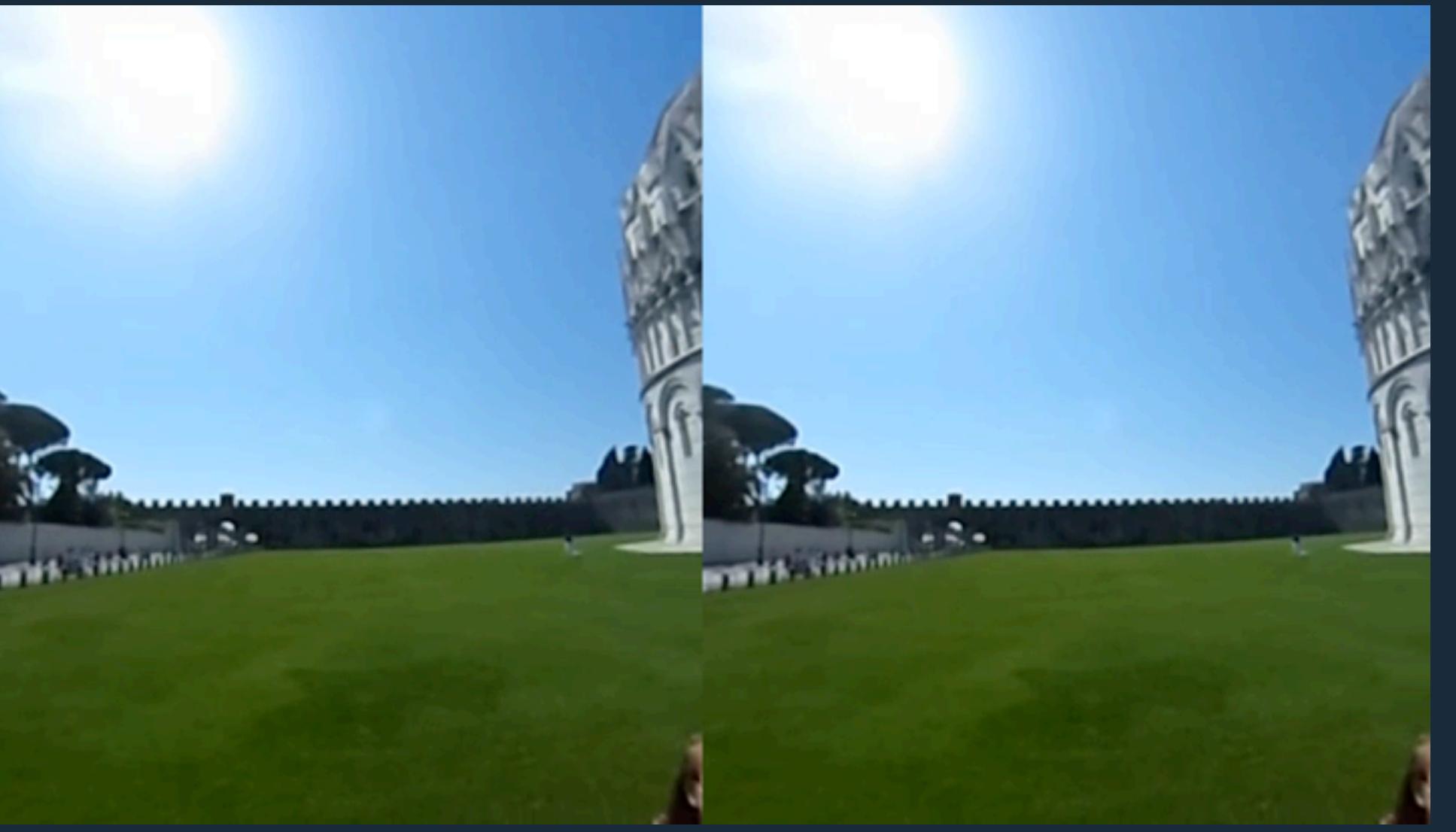
WebVR

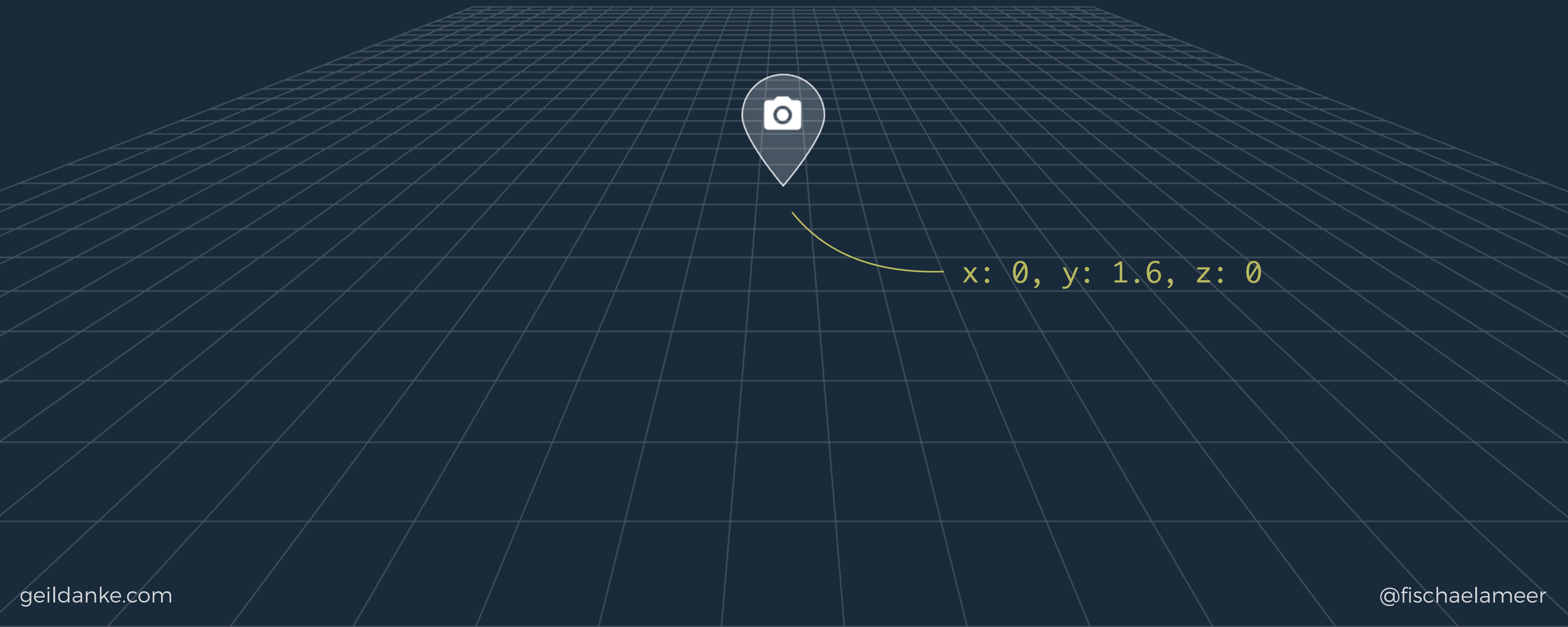
Browser

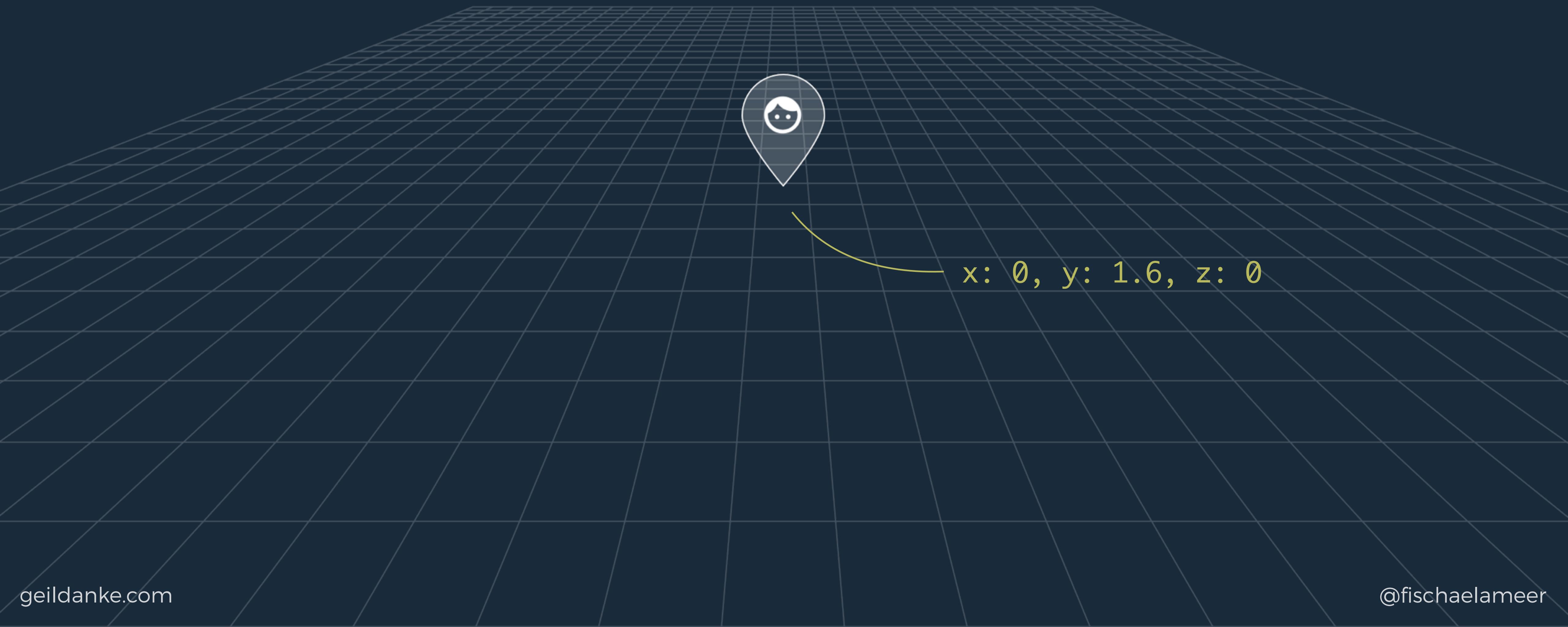


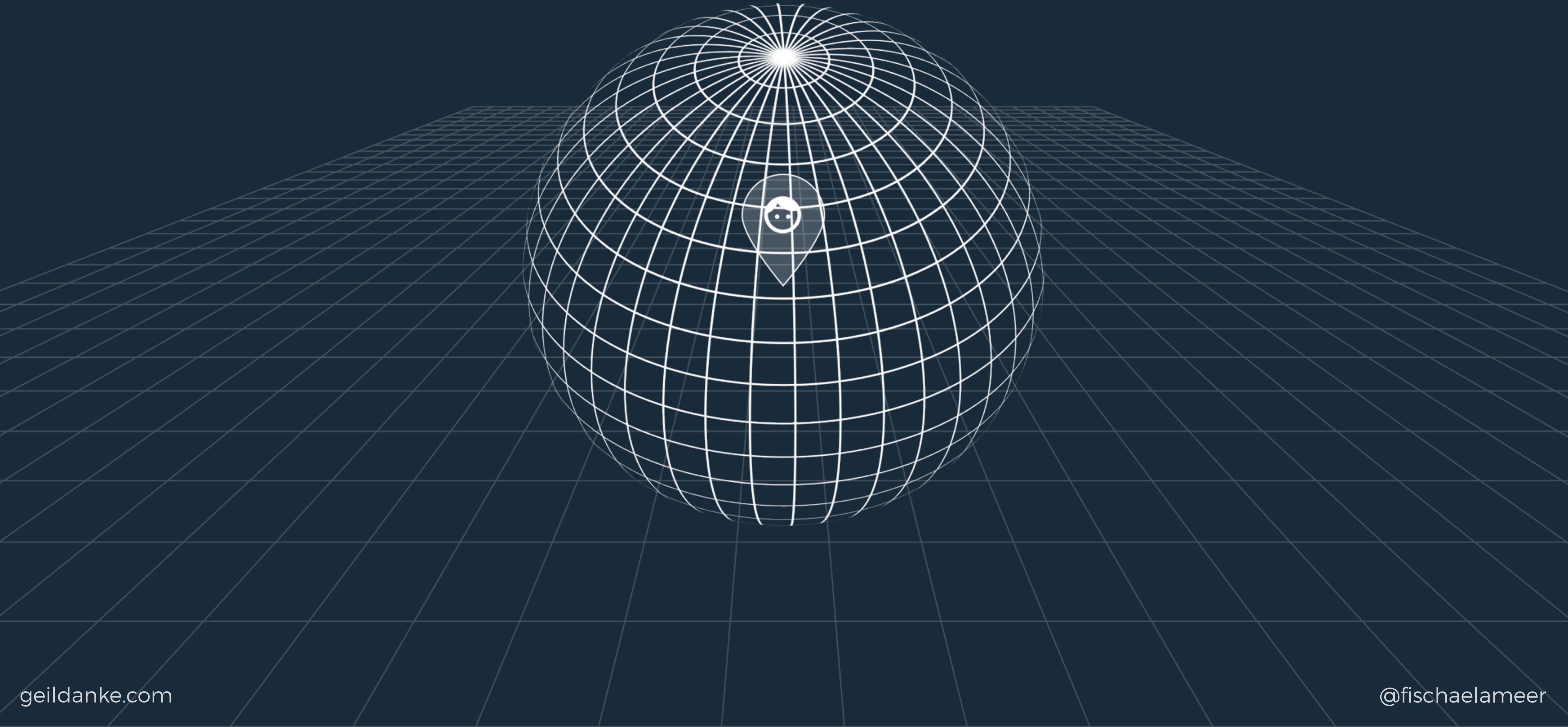












```
let scene = new THREE.Scene();
let camera = new THREE.PerspectiveCamera( 75,
    window.innerWidth / window.innerHeight, 0.1, 1000 );
let renderer = new THREE.WebGLRenderer();

renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );

let geometry = new THREE.SphereGeometry( 500, 60, 40 );
geometry.scale( -1, 1, 1 );

let video = document.createElement( 'video' );
let videoTexture = new THREE.VideoTexture( video );
let videoMaterial = new THREE.MeshBasicMaterial( { map: videoTexture } );
let mesh = new THREE.Mesh( geometry, videoMaterial );

function render() {
    requestAnimationFrame( render );
    renderer.render( scene, camera );
}

render();
```

```
let scene = new THREE.Scene();
let camera = new THREE.PerspectiveCamera( 75,
    window.innerWidth / window.innerHeight, 0.1, 1000 );
let renderer = new THREE.WebGLRenderer();

renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );

let geometry = new THREE.SphereGeometry( 500, 60, 40 );
geometry.scale( -1, 1, 1 );

let video = document.createElement( 'video' );
let videoTexture = new THREE.VideoTexture( video );
let videoMaterial = new THREE.MeshBasicMaterial( { map: videoTexture } );
let mesh = new THREE.Mesh( geometry, videoMaterial );

function render() {
    requestAnimationFrame( render );
    renderer.render( scene, camera );
}

render();
```

```
let scene = new THREE.Scene();
let camera = new THREE.PerspectiveCamera( 75,
    window.innerWidth / window.innerHeight, 0.1, 1000 );
let renderer = new THREE.WebGLRenderer();

renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );

let geometry = new THREE.SphereGeometry( 500, 60, 40 );
geometry.scale( -1, 1, 1 );

let video = document.createElement( 'video' );
let videoTexture = new THREE.VideoTexture( video );
let videoMaterial = new THREE.MeshBasicMaterial( { map: videoTexture } );
let mesh = new THREE.Mesh( geometry, videoMaterial );

function render() {
    requestAnimationFrame( render );
    renderer.render( scene, camera );
}

render();
```

```
let scene = new THREE.Scene();
let camera = new THREE.PerspectiveCamera( 75,
    window.innerWidth / window.innerHeight, 0.1, 1000 );
let renderer = new THREE.WebGLRenderer();

renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );

let geometry = new THREE.SphereGeometry( 500, 60, 40 );
geometry.scale( -1, 1, 1 );

let video = document.createElement( 'video' );
let videoTexture = new THREE.VideoTexture( video );
let videoMaterial = new THREE.MeshBasicMaterial( { map: videoTexture } );
let mesh = new THREE.Mesh( geometry, videoMaterial );

function render() {
    requestAnimationFrame( render );
    renderer.render( scene, camera );
}

render();
```

```
let vrDisplay;

navigator.getVRDisplays().then( function( displays ) {
  if ( displays.length > 0 ) {
    vrDisplay = displays[ 0 ];
  } else {
    console.log( 'No VR Displays found.' );
  }
});
```

`VRDisplay.isConnected`
`VRDisplay.isPresenting`

`VRDisplay.getEyeParameters()`
`VRDisplay.getPose()`
`VRDisplay.requestPresent()`
`VRDisplay.submitFrame()`
`VRDisplay.requestAnimationFrame()`

```
vrDisplay.requestPresent( [ { source: myCanvas } ] );
```

```
myButton.addEventListener( 'click', function() {
    vrDisplay.requestPresent( [ { source: myCanvas } ] );
});
```

```
myButton.addEventListener( 'click', function() {
    vrDisplay.requestPresent( [ { source: myCanvas } ] )
        .then( function() {
            vrDisplay.requestAnimationFrame( render );
        });
});
```

```
function render() {  
    vrDisplay.requestAnimationFrame( render );  
  
}
```

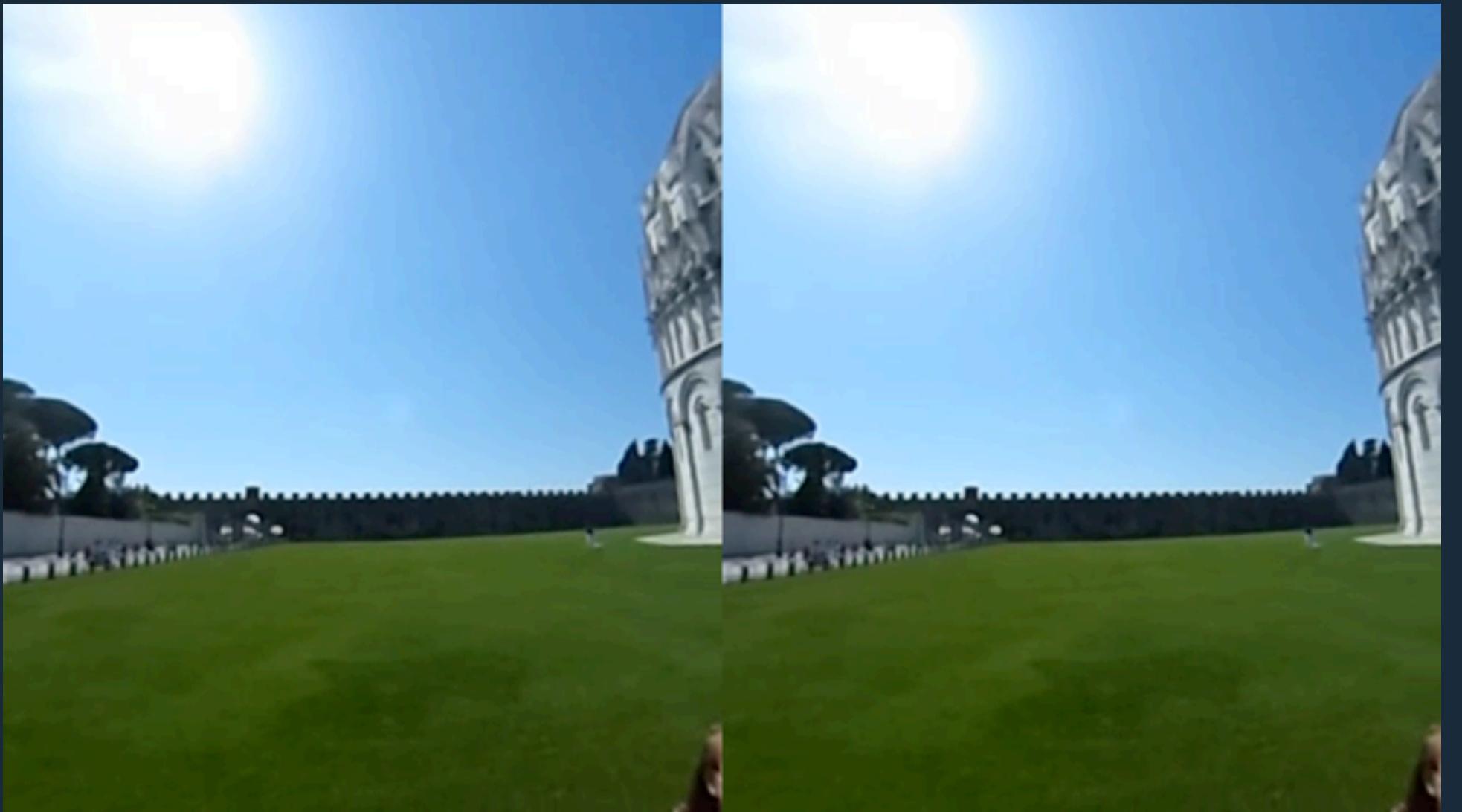
```
function render() {  
    vrDisplay.requestAnimationFrame( render );  
  
    // update display pose  
    // update scene and meshes  
  
}
```

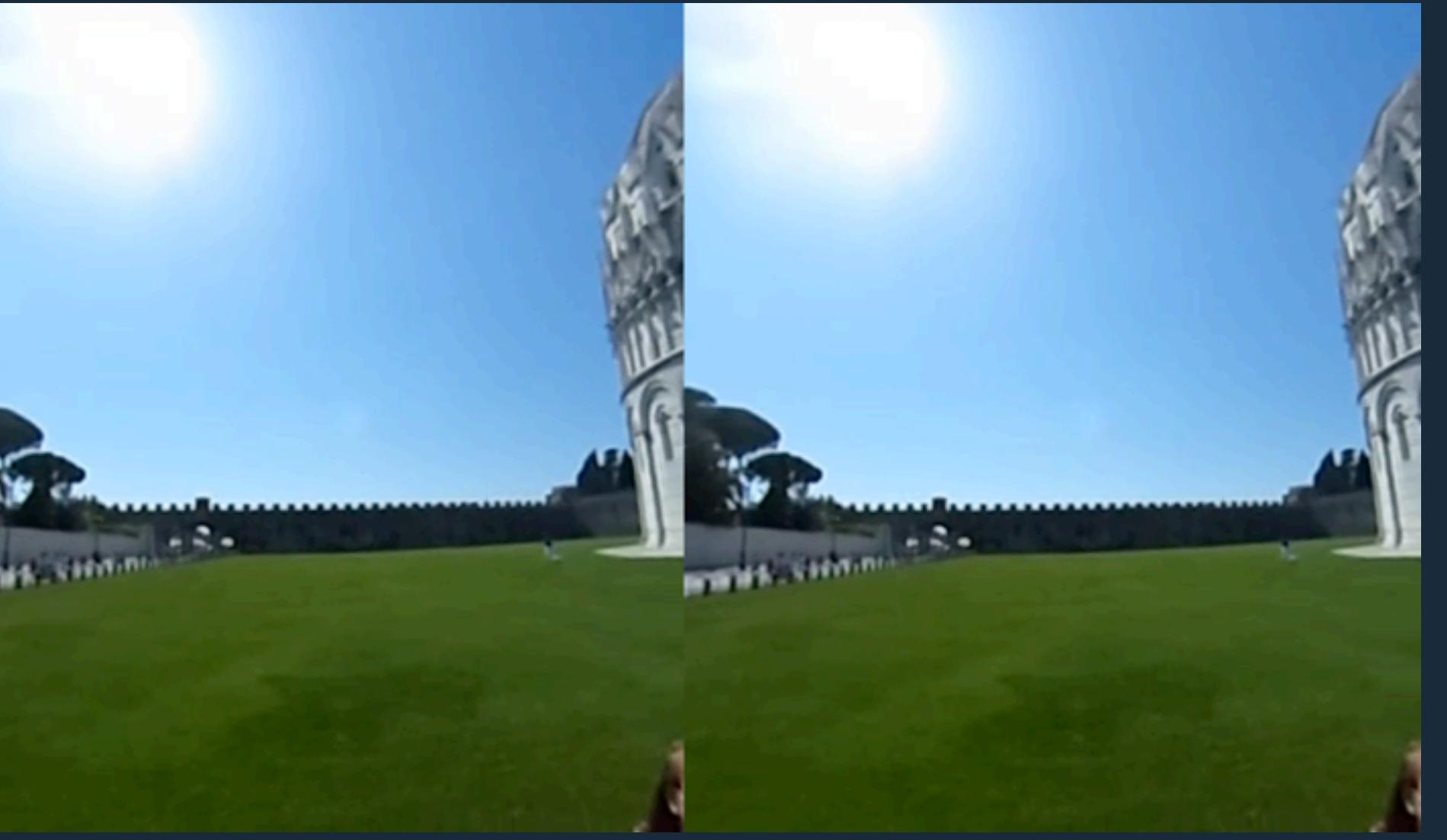
```
let pose = vrDisplay.getPose();

console.log( pose.orientation );
// [ 0, 0, 0, 1 ]
// [ -0.0000724312, -0.06752134, 0.0028374712, 0.9977243 ]
console.log( pose.position );
// null
// [ 0.05234, -0.043485, 0.0003243 ]
```

```
let leftEyeParameters = vrDisplay.getEyeParameters( 'left' );  
  
console.log( leftEyeParameters.offset );  
// [ -0.03, 0, 0 ]  
console.log( leftEyeParameters.renderWidth );  
// 640.5  
console.log( leftEyeParameters.renderHeight );  
// 721
```

```
function render() {  
    vrDisplay.requestAnimationFrame( render );  
  
    // update display pose  
    // update scene and meshes  
  
    vrDisplay.submitFrame( pose );  
}
```





Is WebVR Ready?

[STATUS](#) * [SPEC](#) * [DOWNLOADS](#) * [INFO](#) * [FAQ](#) * [GITHUB](#)

WebVR enthusiasm

The first thing any implementation needs.

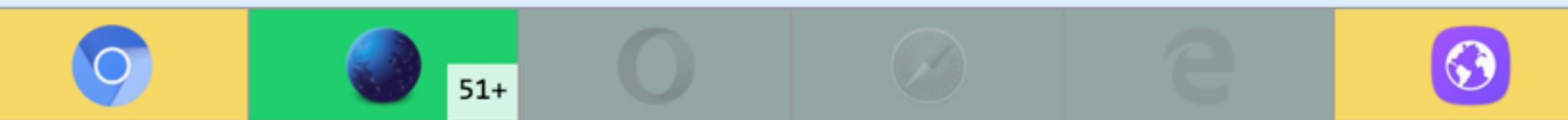


Edge: In development (announced on Sep 9, 2016).

WebVR API (`navigator.getVRDisplays`)

Client-side JavaScript API for querying connected VR headsets and peripherals. Returns a Promise that resolves an array of available `VRDisplays`.

[Spec. Test.](#)



```
graph TD; Browser[Browser] --- threeJs["three.js"]; Browser --- webGL[WebGL]; Browser --- webVR[WebVR]
```

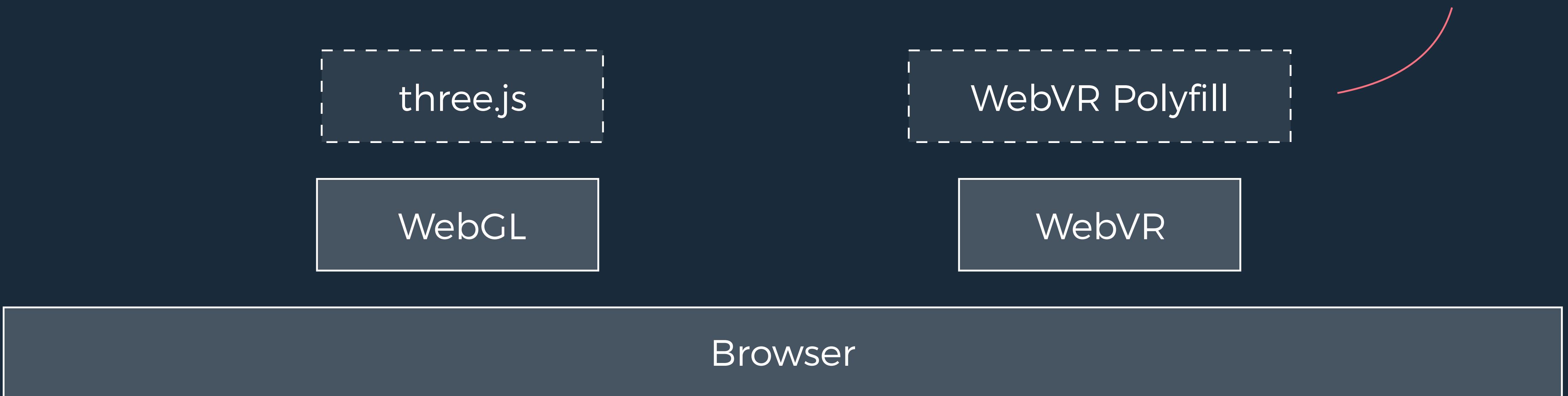
three.js

WebGL

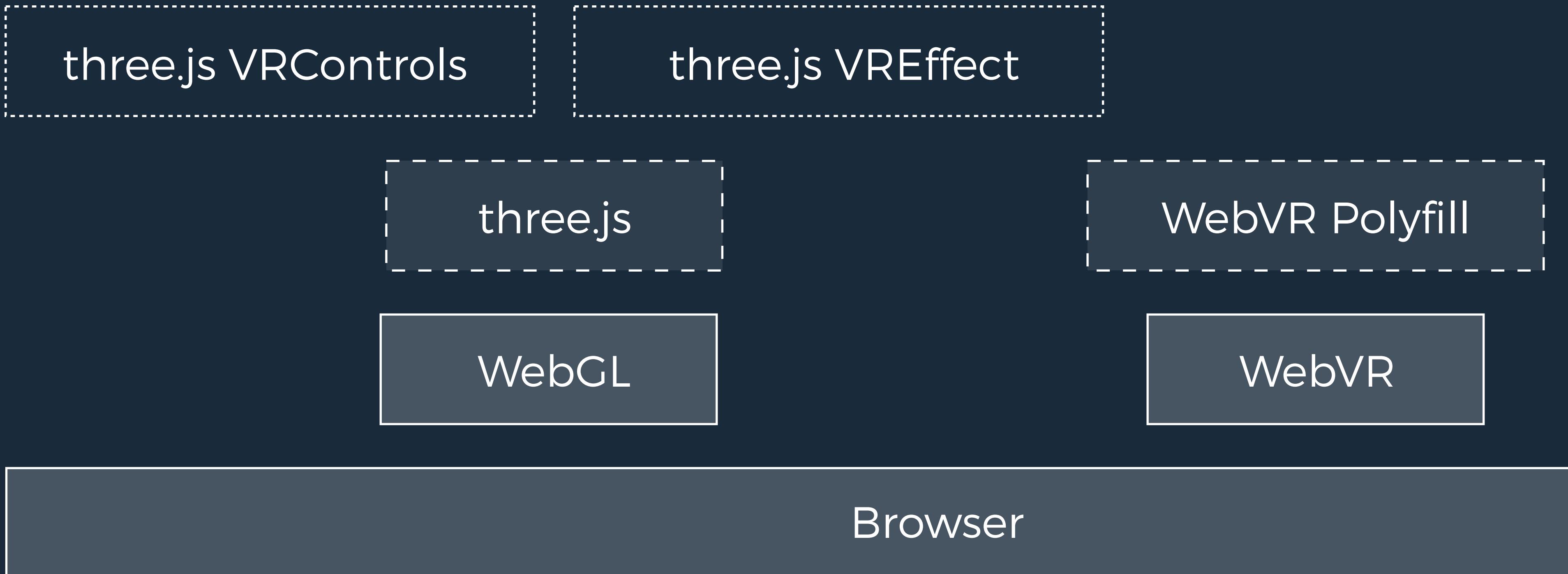
WebVR

Browser

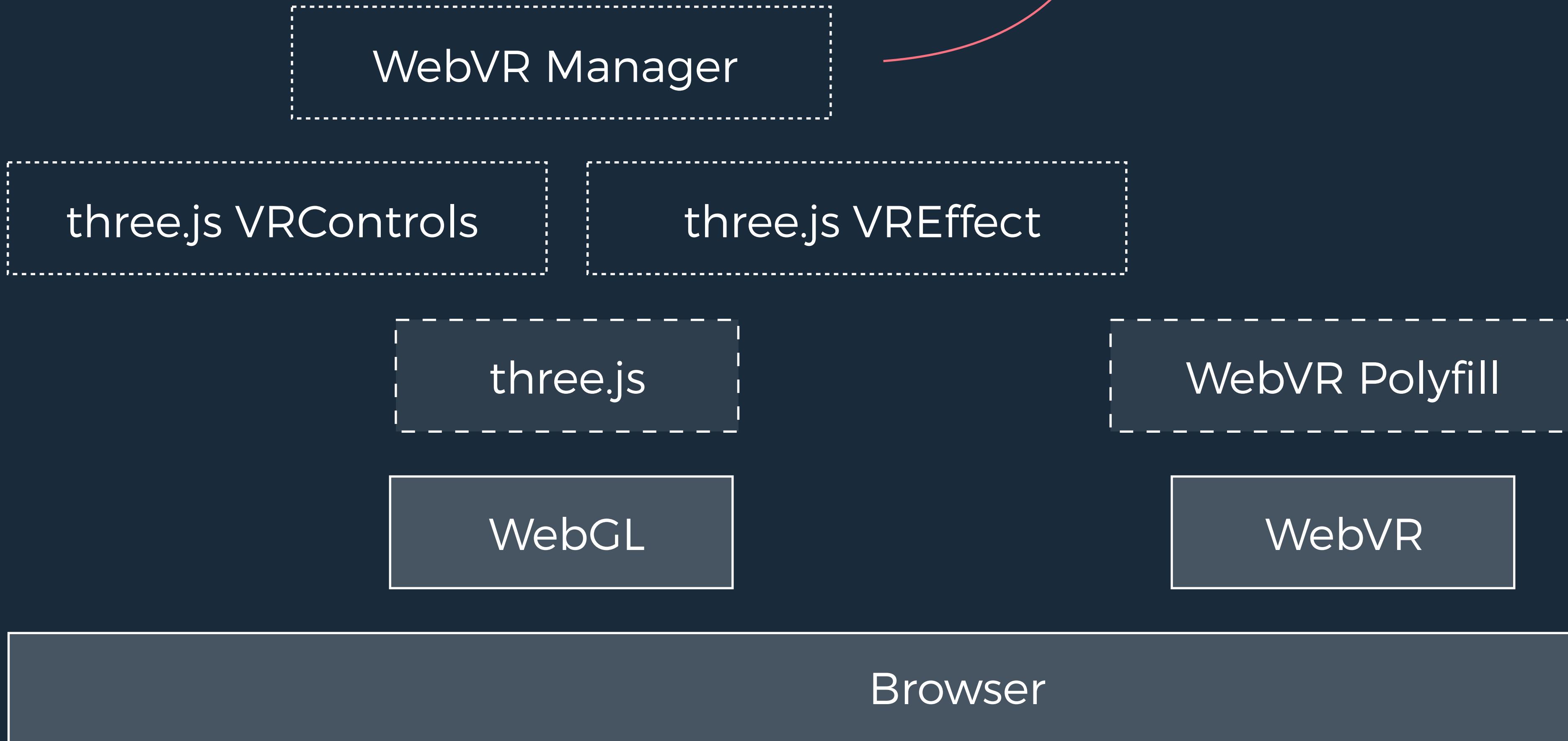
<https://github.com/googlevr/webvr-polyfill>



Ricardo Cabello



Boris Smus





WebGL & static image fallbacks



WebGL

Touch & Gyroscope Input

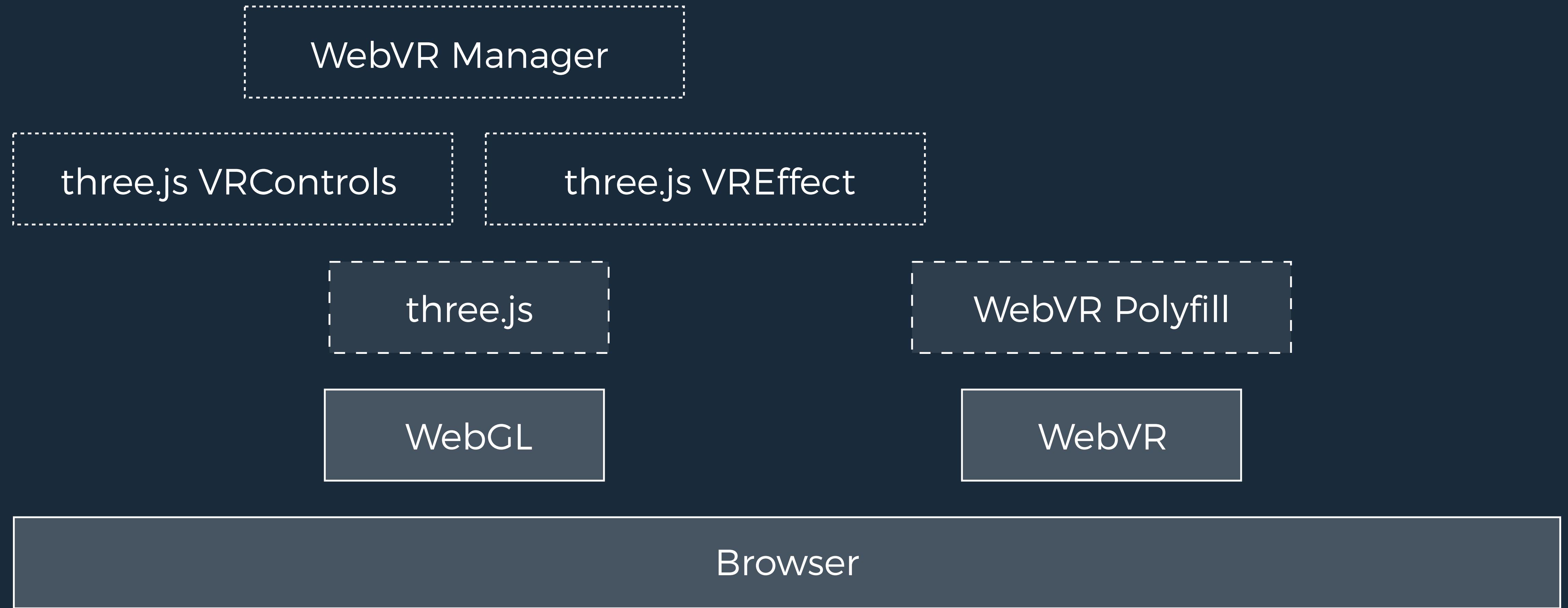
Mobile and Desktop VR Devices

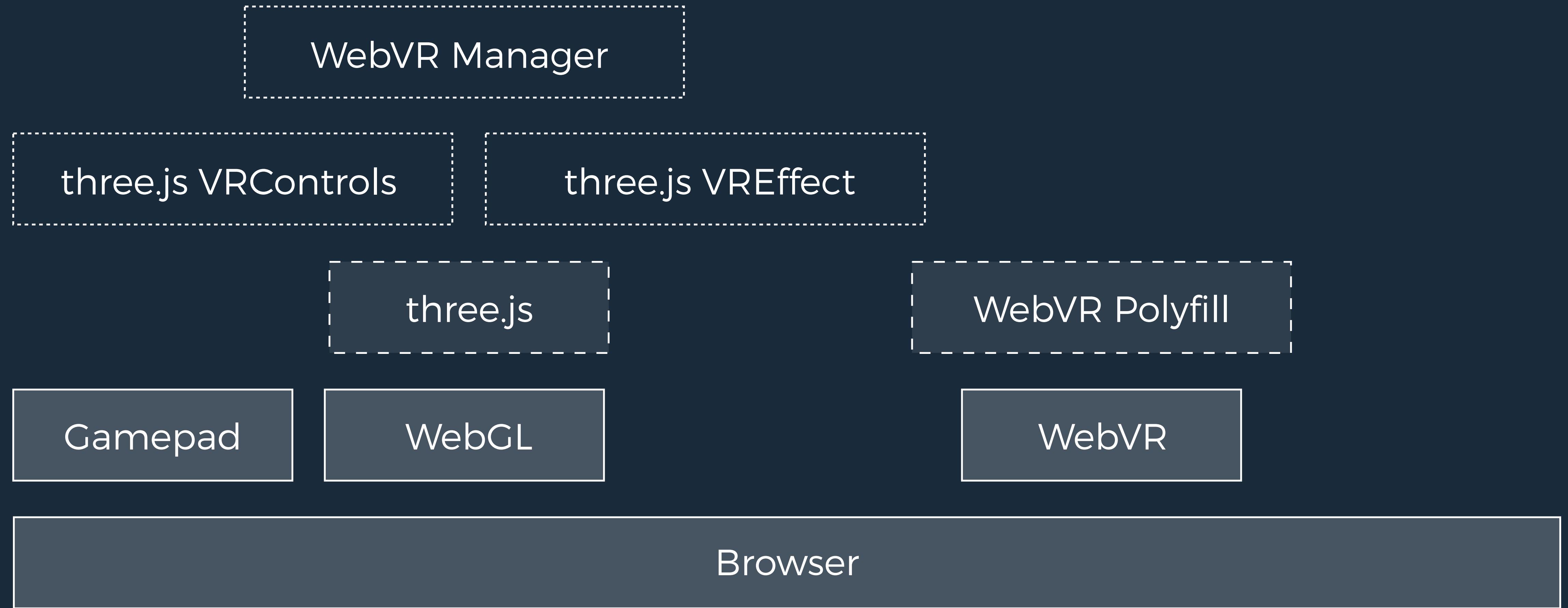


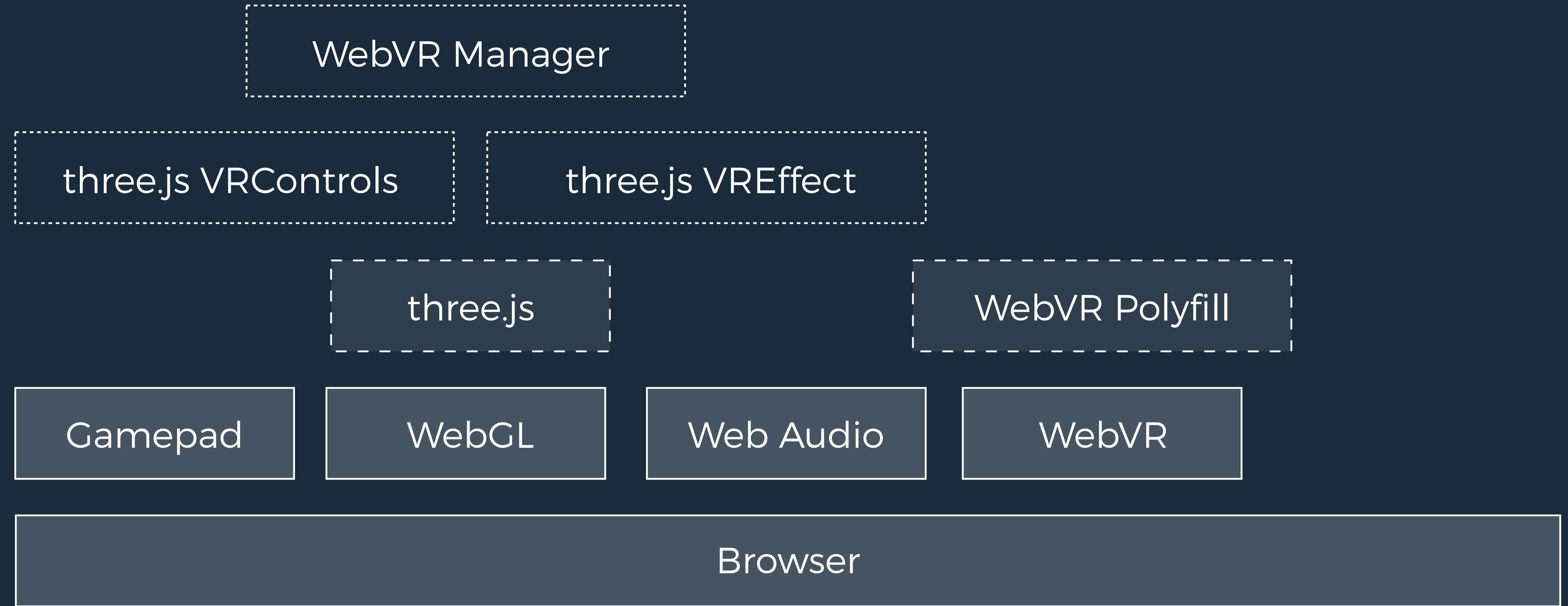
Mobile and Desktop VR Devices

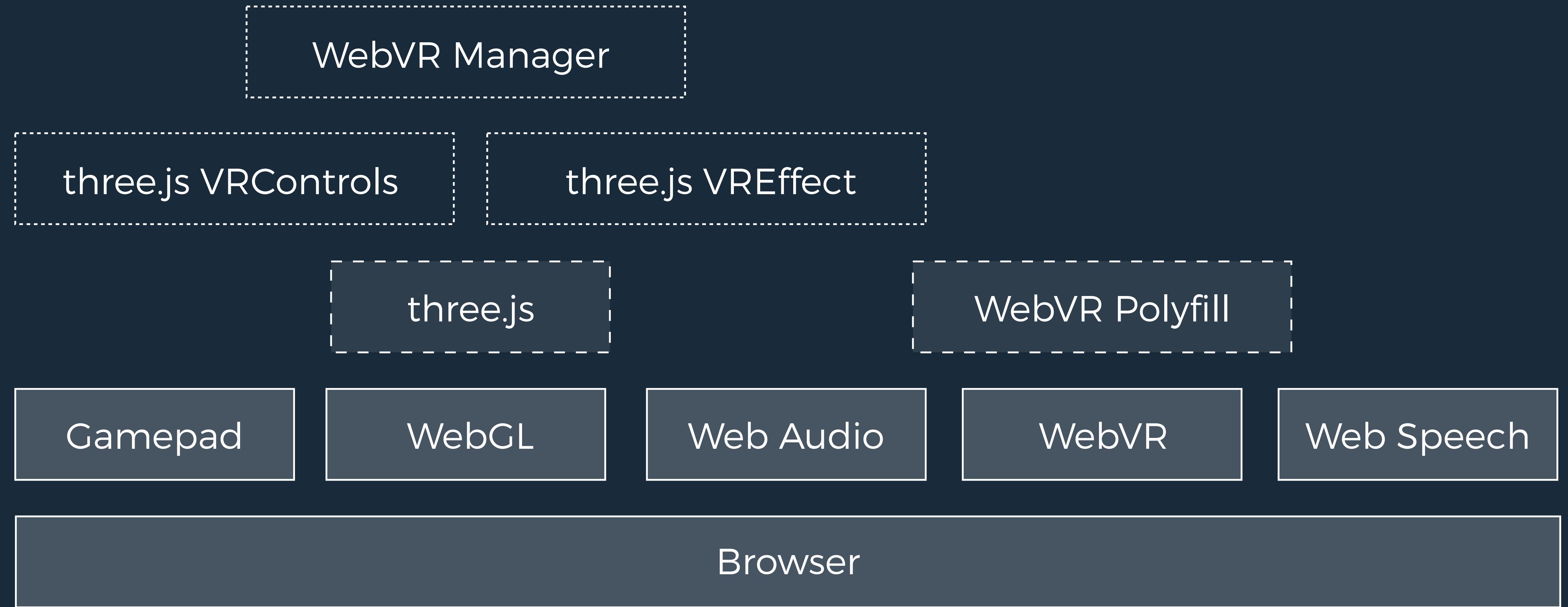


Progressive Loading



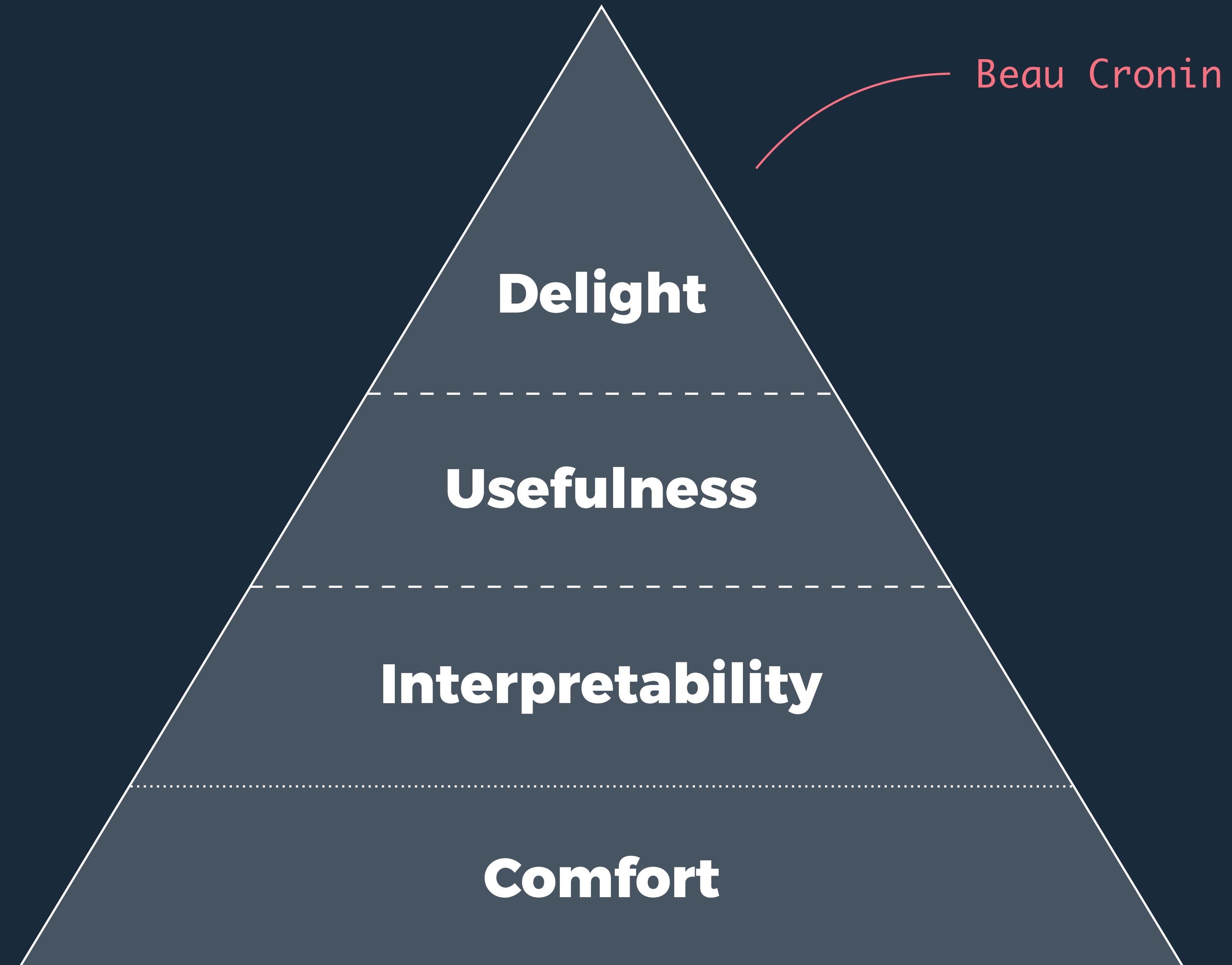


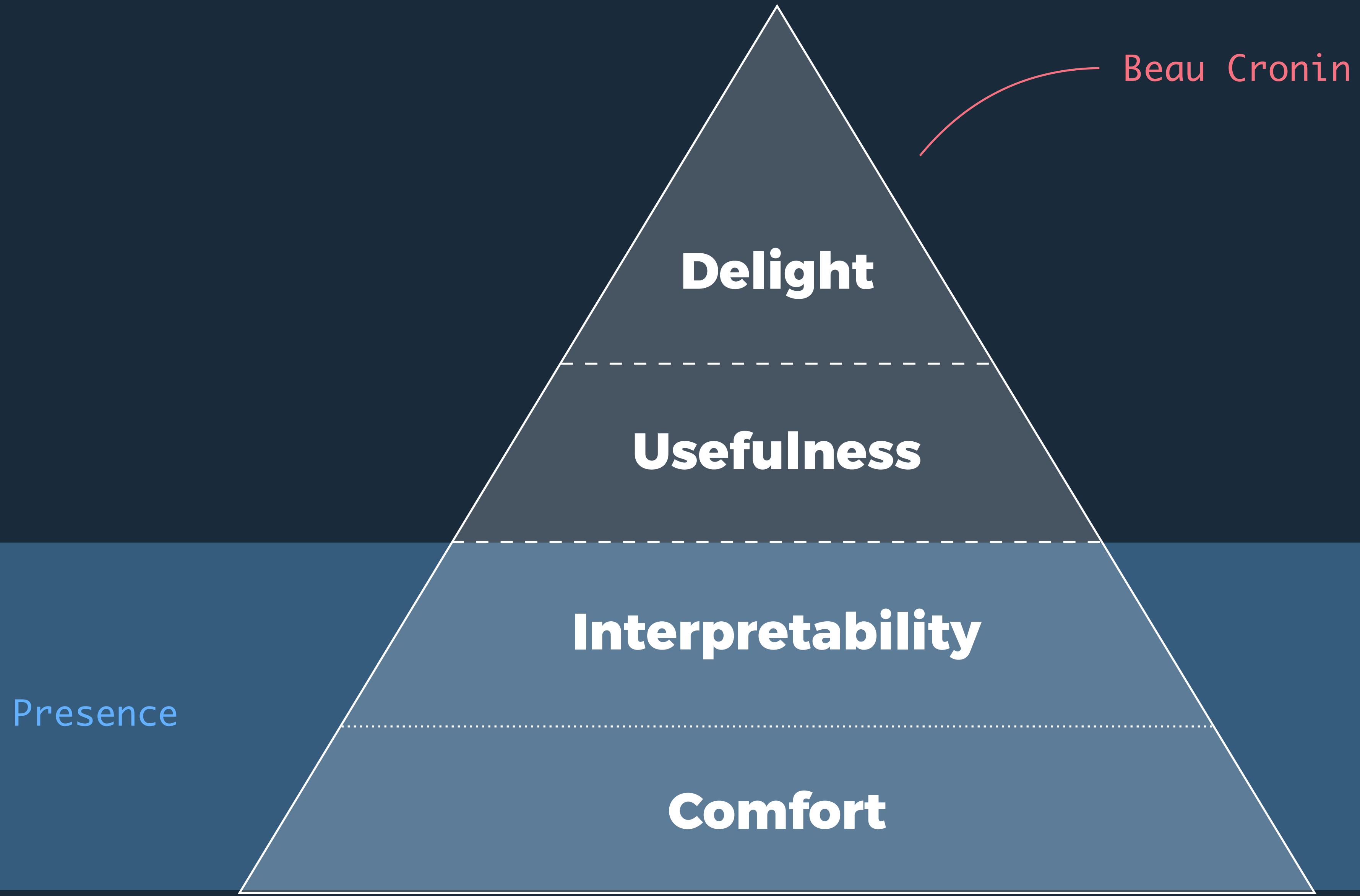






UX Design for VR

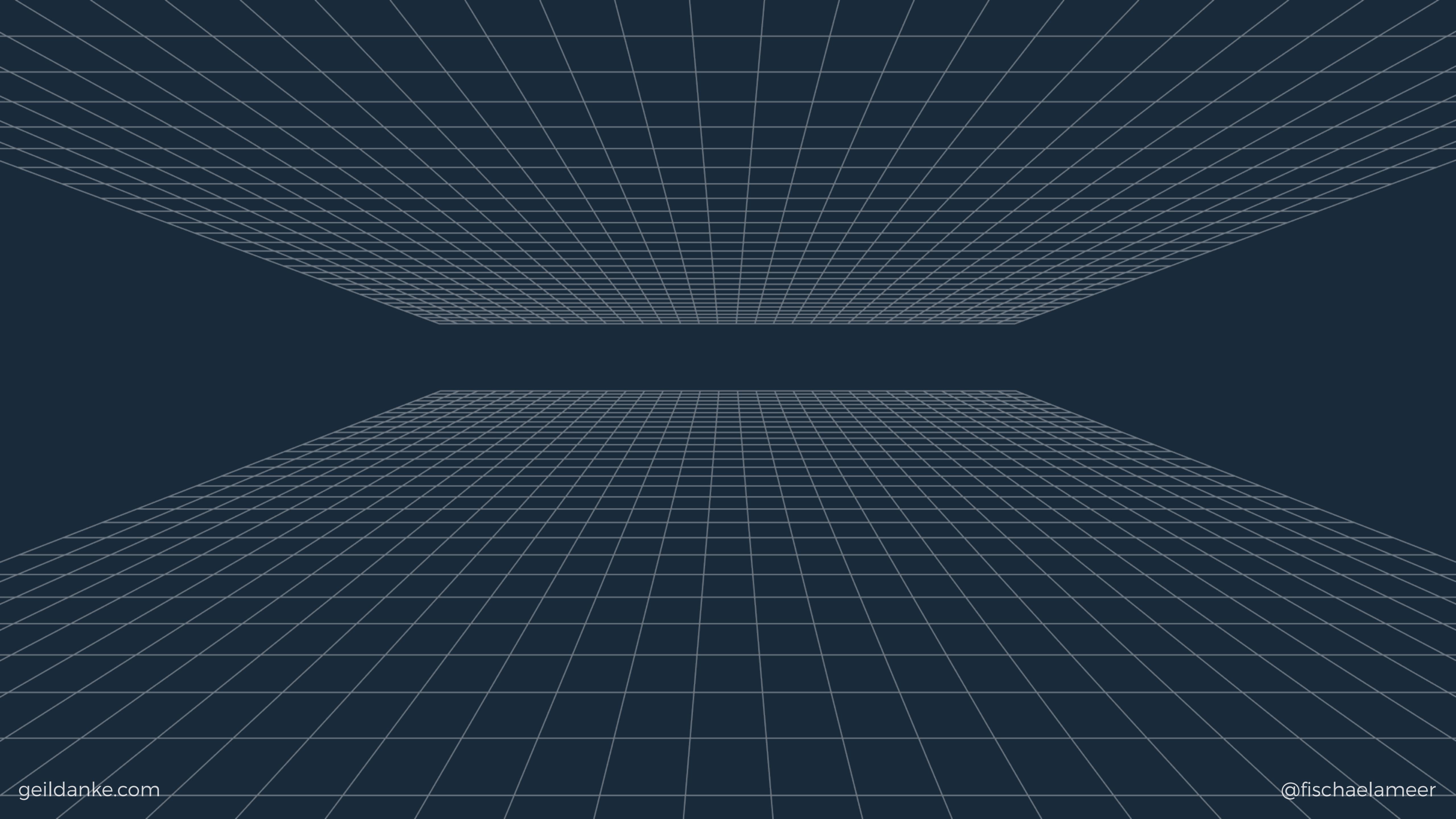


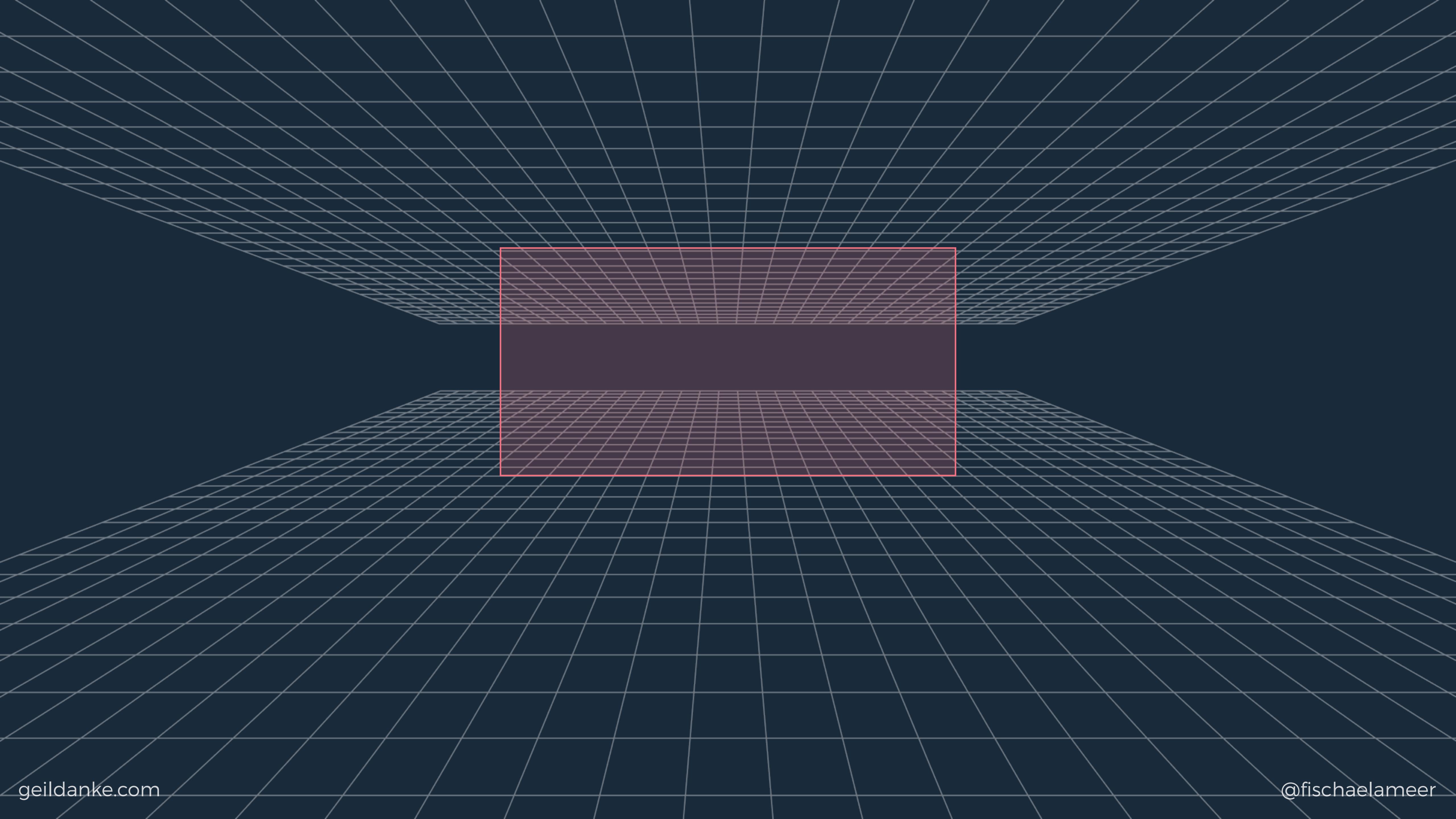


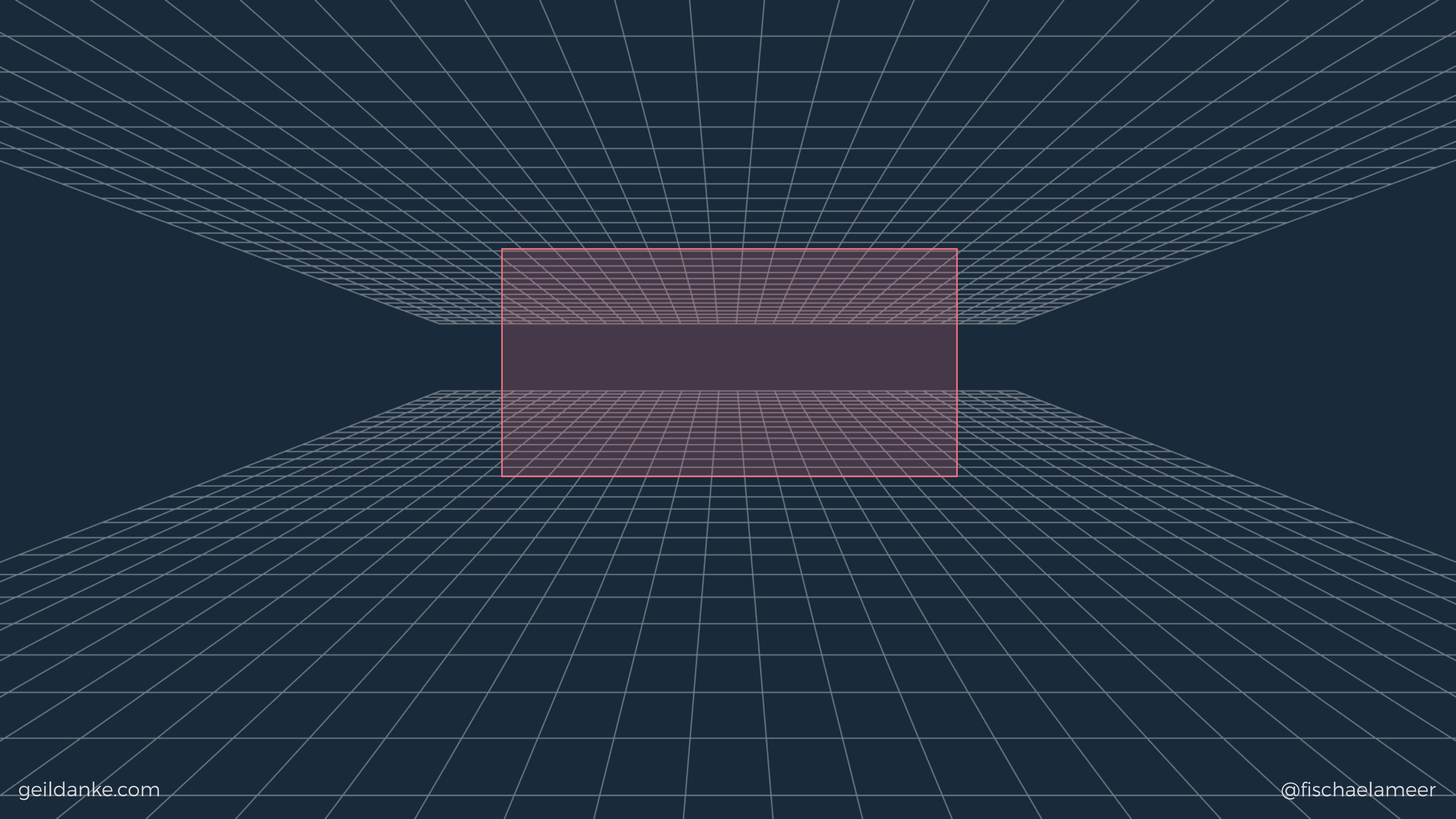
Ergonomics

It was the pioneer days; people had to make their own interrogation rooms. Out of cornmeal. These endless days are finally ending in a blaze. When I say, 'I love you,' it's not because I want you or because I can't have you. It's my estimation that every man ever got a statue made of him was one kind of sommbitch or another. Oh my god you will never believe what happened at school today. From beneath you, it devours. I am never gonna see a merman, ever.

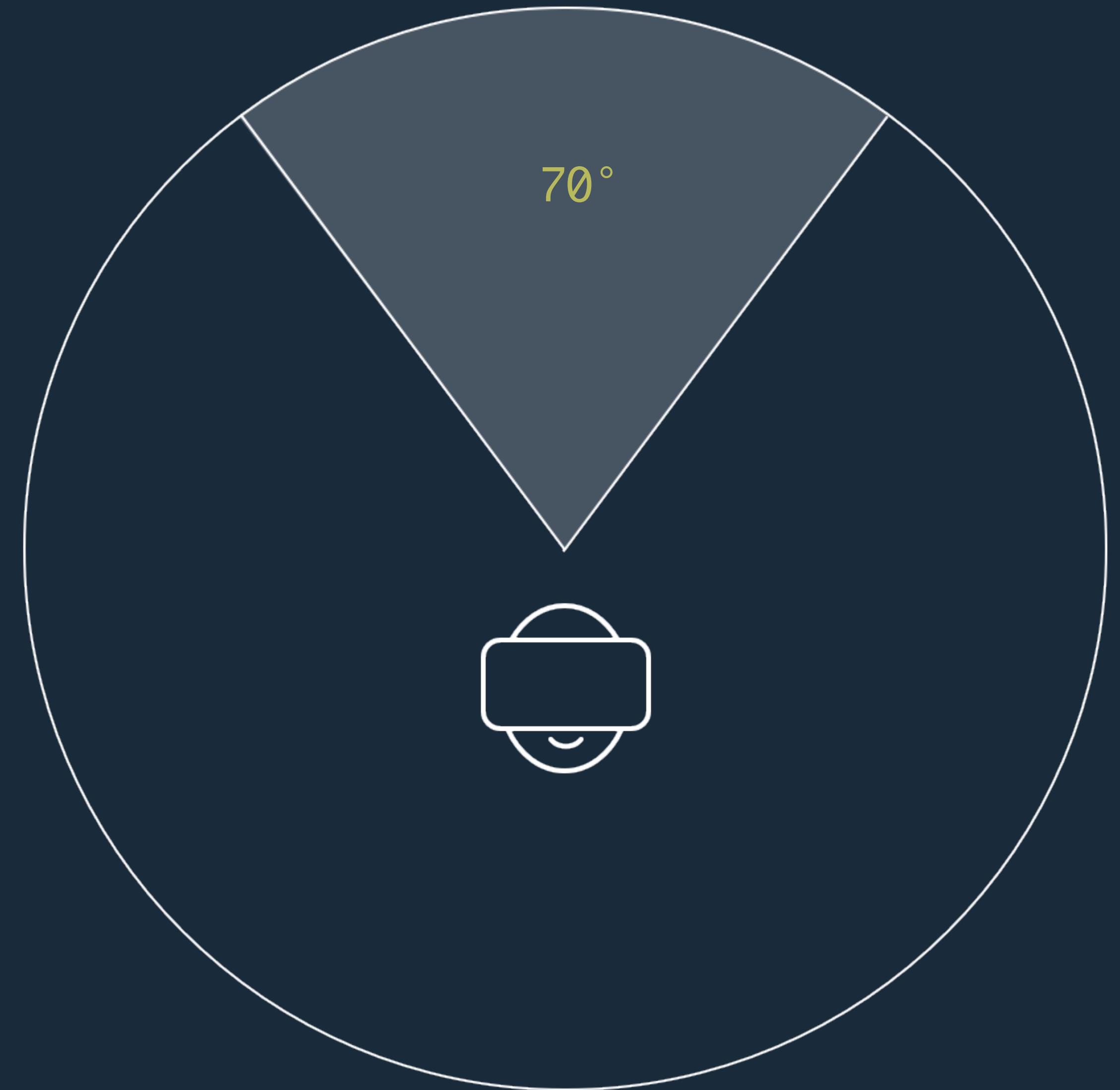
It was supposed to confuse him, but it just made him peppy. It was like the Heimlich, with stripes! How did your brain even learn human speech? I'm just so curious. Apocalypse, we've all been there; the same old trips, why should we care? Frankly, it's ludicrous to have these interlocking bodies and not...interlock. I just don't see why everyone's always picking on Marie-Antoinette. You're the one freaky thing in my freaky world that still makes sense to me. You are talking crazy-person talk.





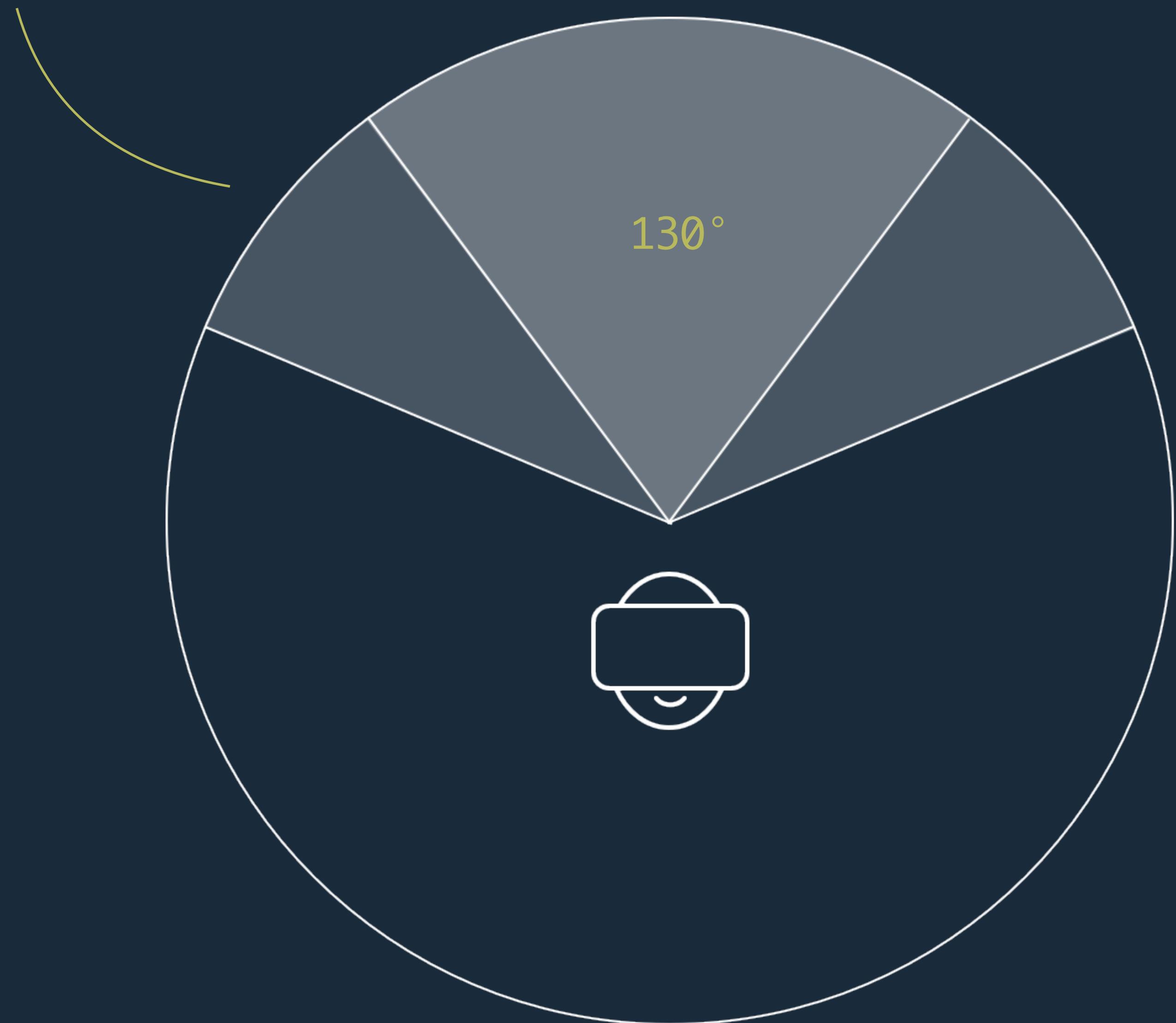




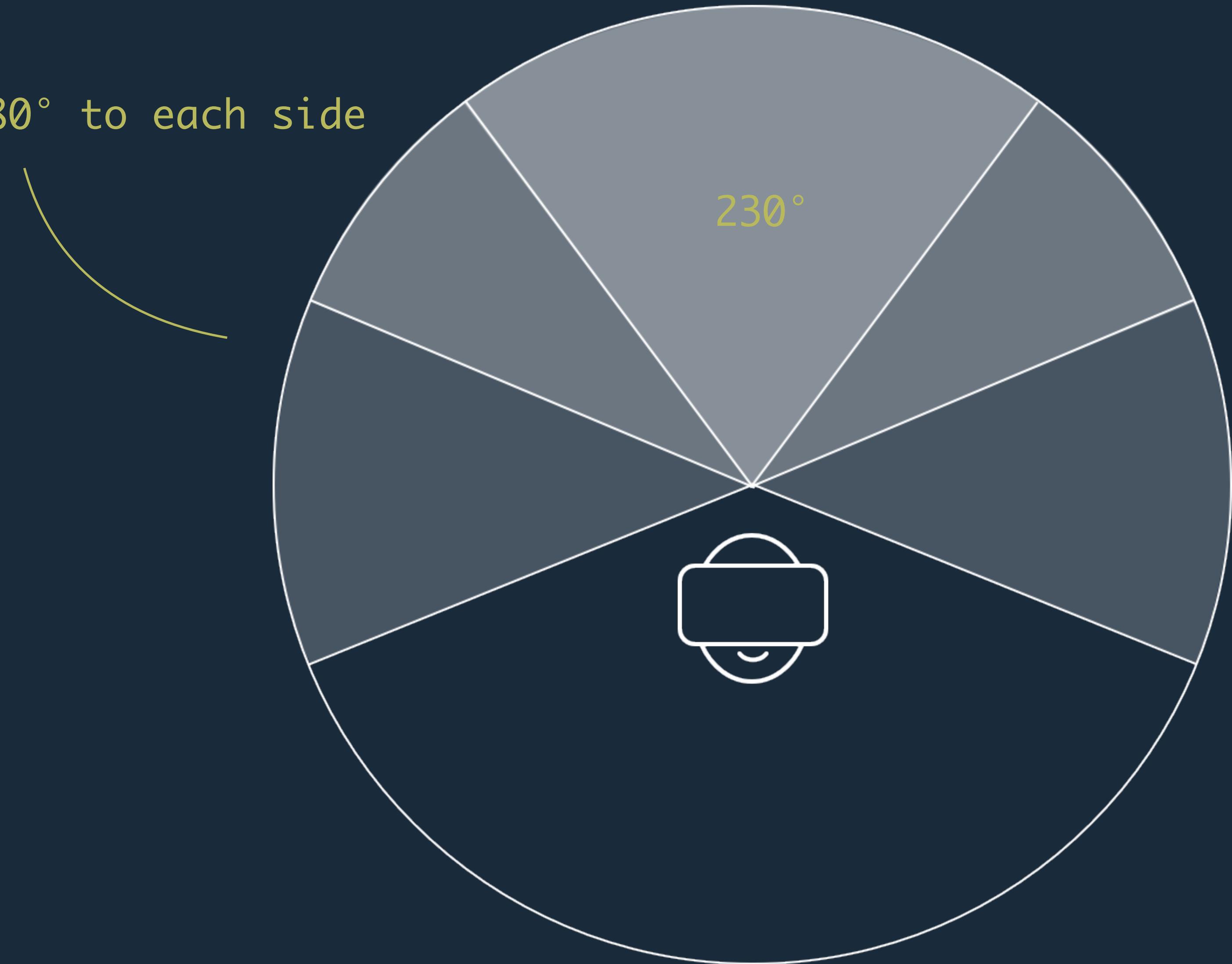


70°

Comfortably bending 30° to each side



Stretching 80° to each side



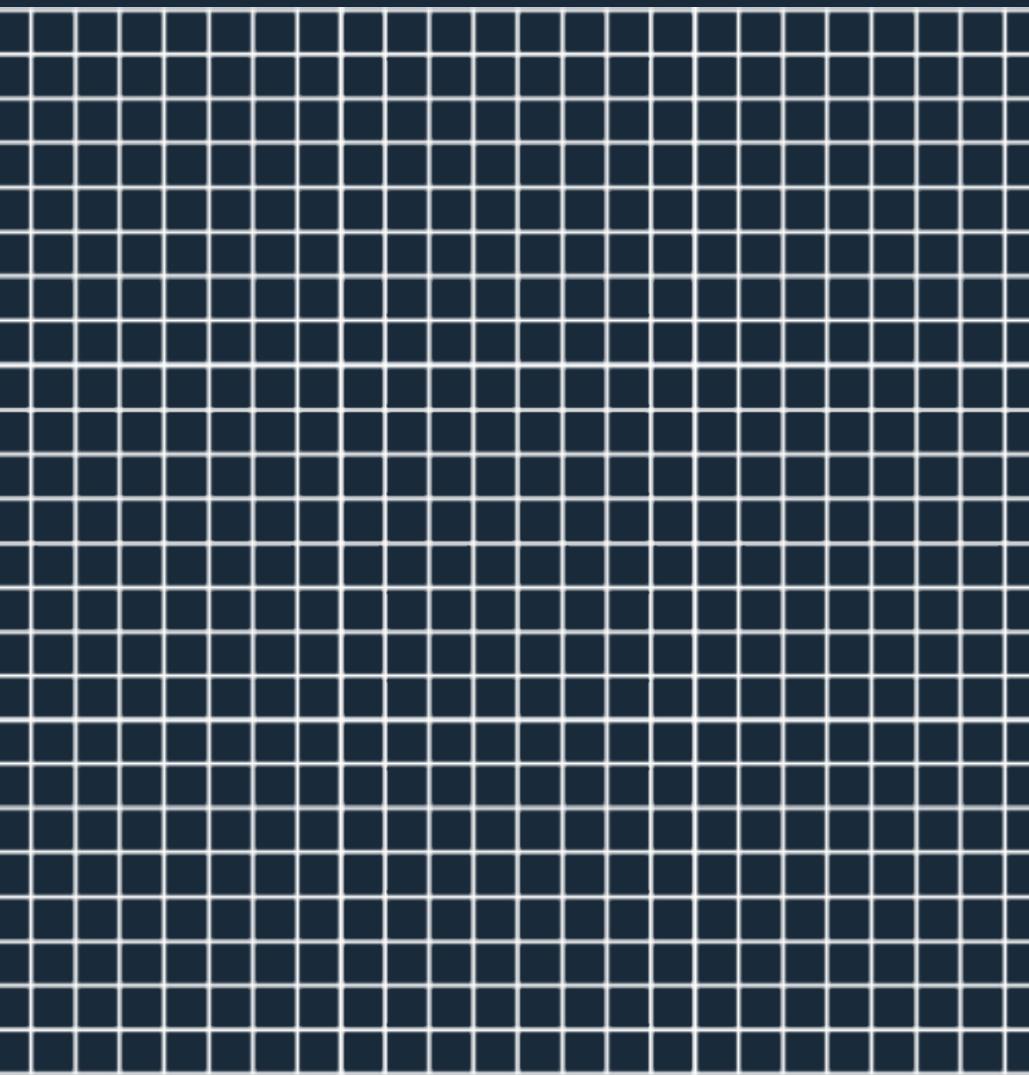
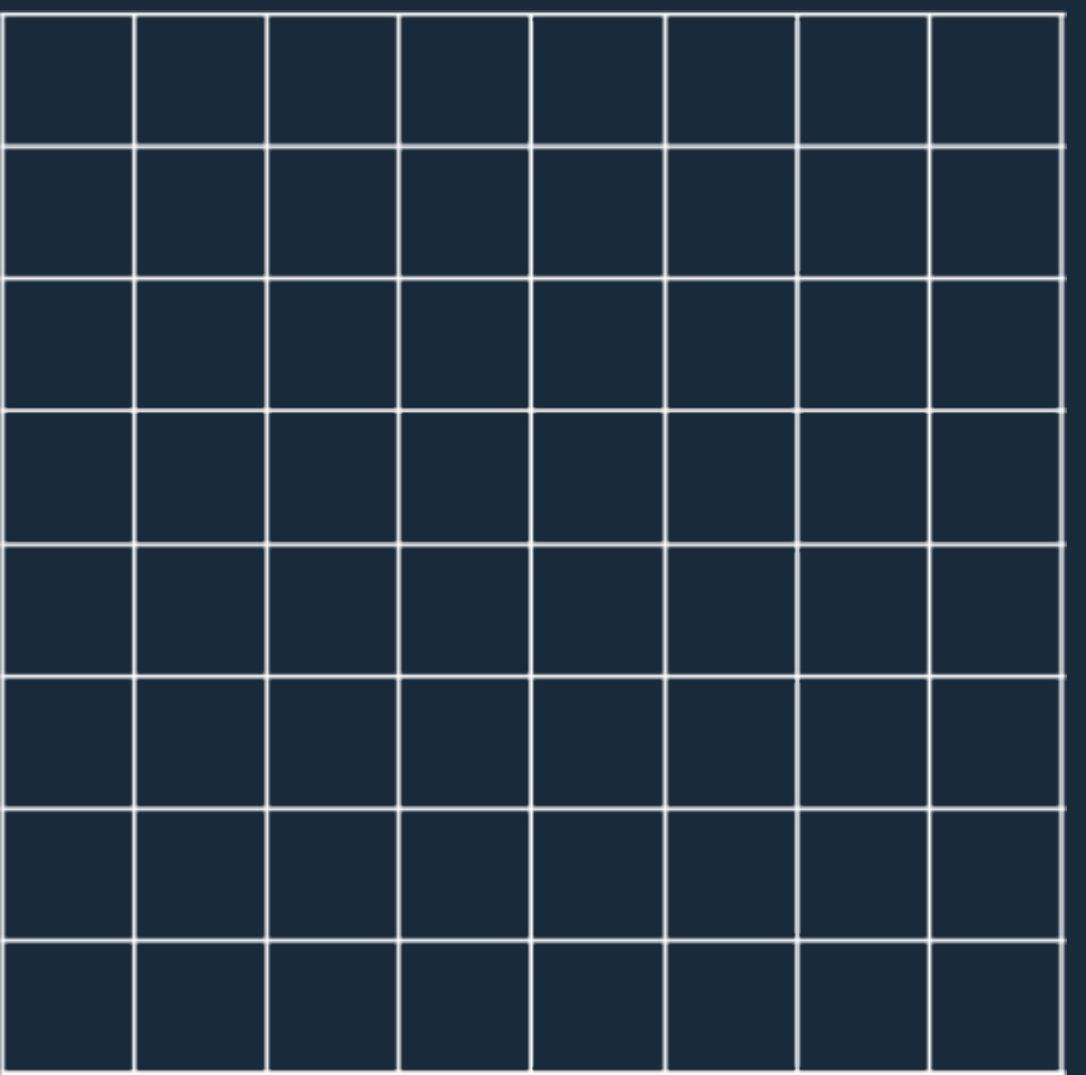


0 . 5m

20m

~20px

< 20px



60ppd

~10ppd

make the user comfortable

👉 avoid eyestrain: use darker colors

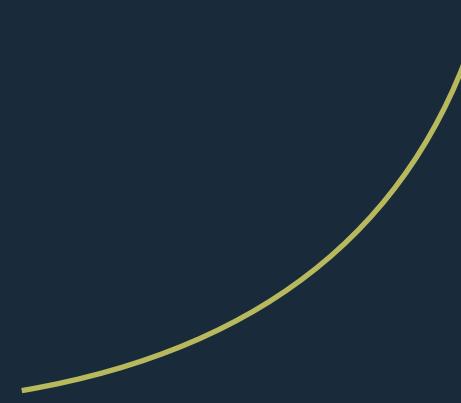
avoid focussing on different depths

do not trigger phobias

use correct scales

do not move things fast towards the camera

do not attach things near the camera



make the user comfortable

avoid eyestrain: use darker colors

👉 avoid focussing on different depths

do not trigger phobias

use correct scales

do not move things fast towards the camera

do not attach things near the camera

make the user comfortable

avoid eyestrain: use darker colors

avoid focussing on different depths

👉 do not trigger phobias

use correct scales

do not move things fast towards the camera

do not attach things near the camera

make the user comfortable

avoid eyestrain: use darker colors

avoid focussing on different depths

do not trigger phobias

👉 use correct scales

do not move things fast towards the camera

do not attach things near the camera

make the user comfortable

avoid eyestrain: use darker colors

avoid focussing on different depths

do not trigger phobias

use correct scales

👉 do not move things fast towards the camera

do not attach things near the camera

make the user comfortable

avoid eyestrain: use darker colors

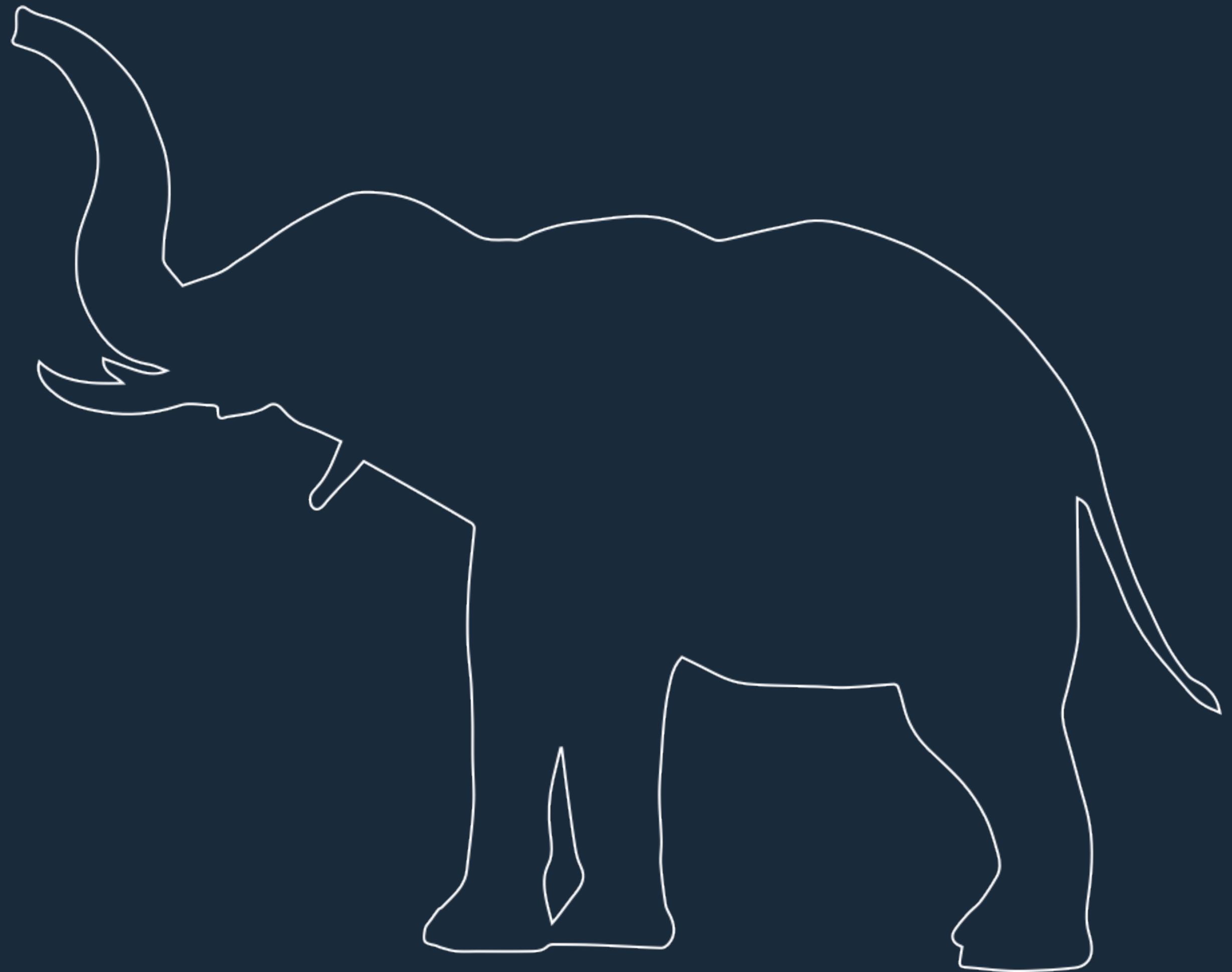
avoid focussing on different depths

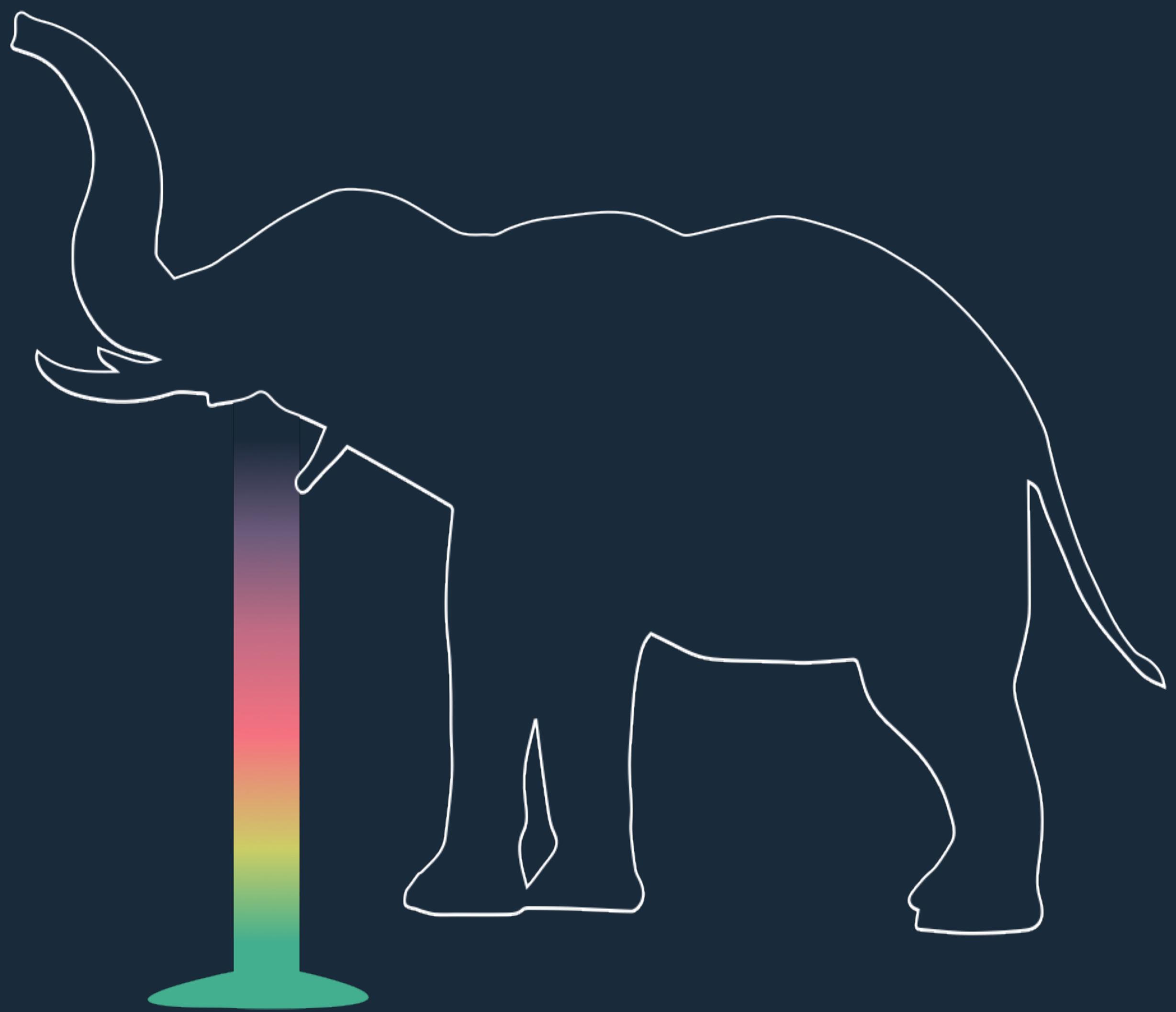
do not trigger phobias

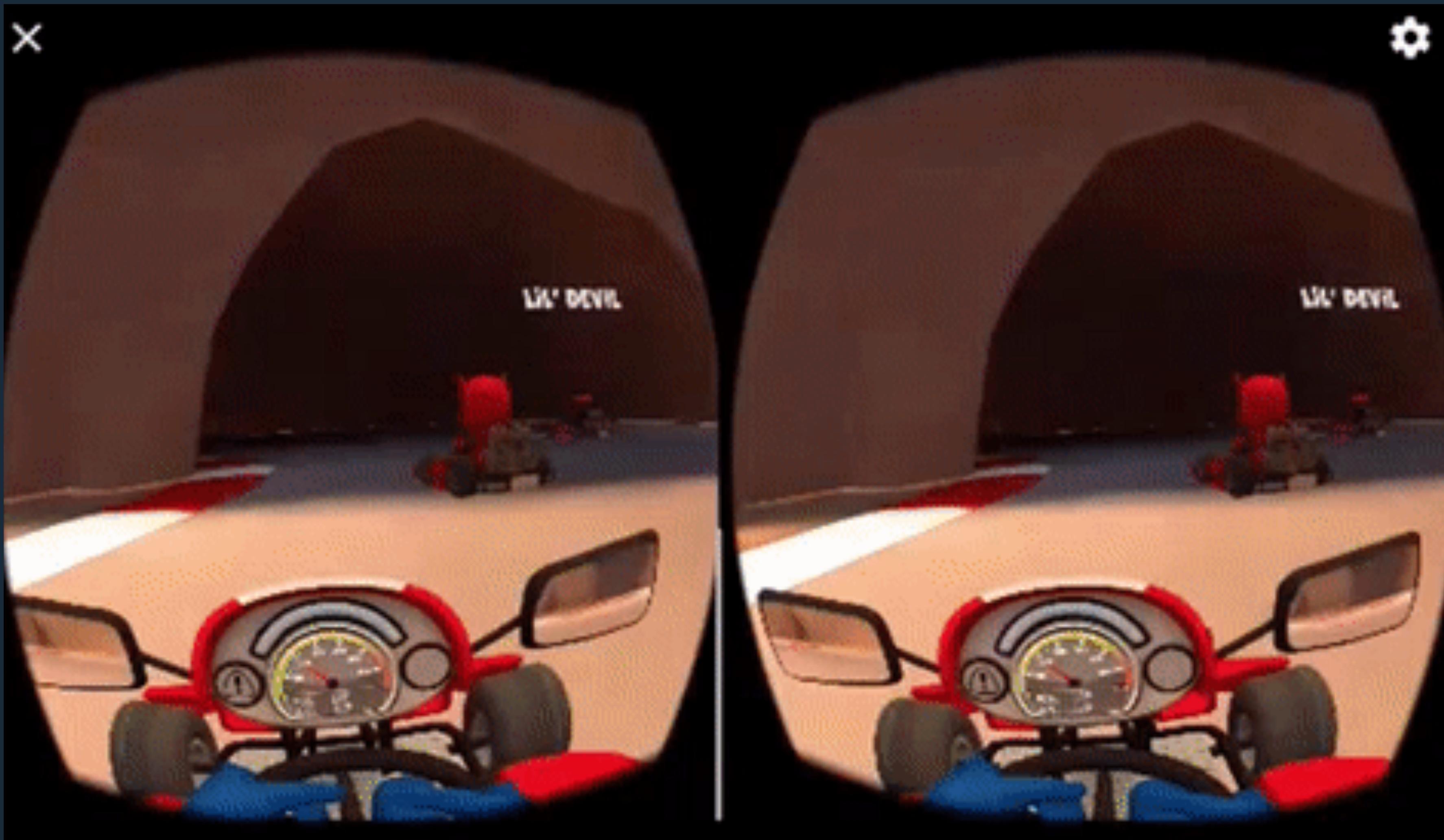
use correct scales

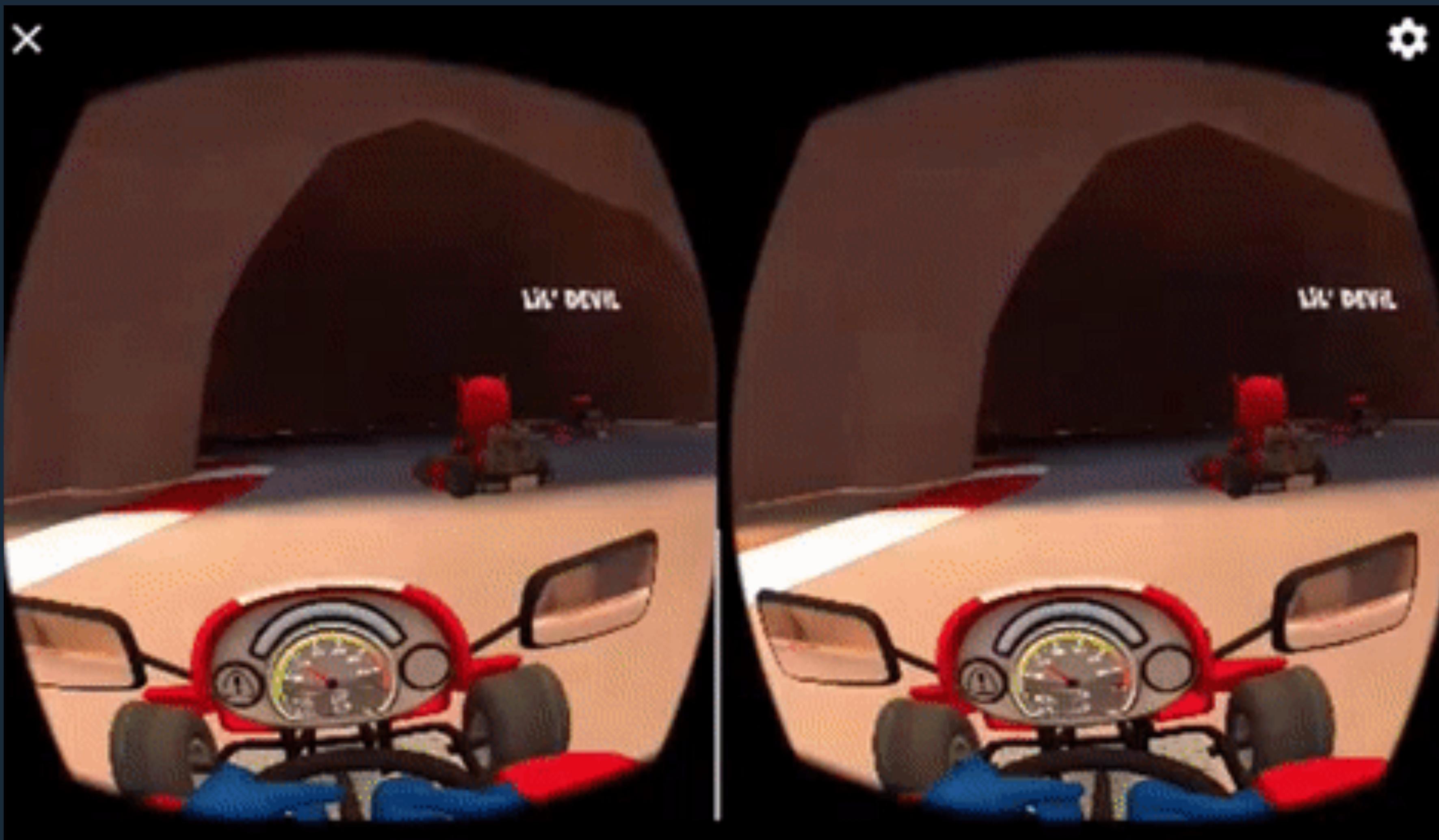
do not move things fast towards the camera

👉 do not attach things near the camera









do not make your users sick!



no acceleration

do not move the horizon or the camera

always keep a low latency and a high frame rate

avoid flicker and blur

add a stable focus point

support short usage

abstract design is better than realistic



do not make your users sick!

no acceleration

👉 do not move the horizon or the camera

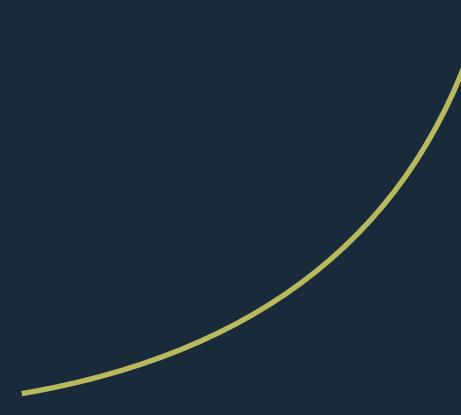
always keep a low latency and a high frame rate

avoid flicker and blur

add a stable focus point

support short usage

abstract design is better than realistic



do not make your users sick!

no acceleration

do not move the horizon or the camera

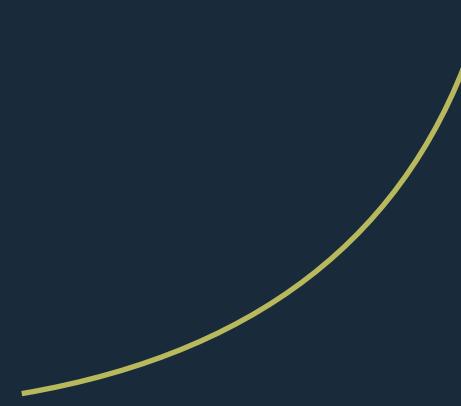
👉 always keep a low latency and a high frame rate

avoid flicker and blur

add a stable focus point

support short usage

abstract design is better than realistic



do not make your users sick!

no acceleration

do not move the horizon or the camera

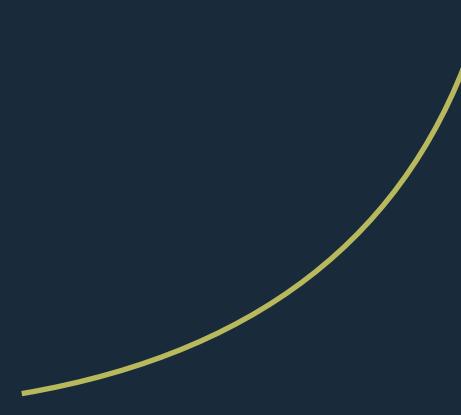
always keep a low latency and a high frame rate

👉 avoid flicker and blur

add a stable focus point

support short usage

abstract design is better than realistic



do not make your users sick!

no acceleration

do not move the horizon or the camera

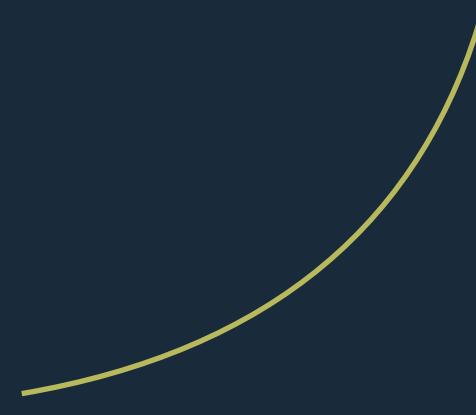
always keep a low latency and a high frame rate

avoid flicker and blur

👉 add a stable focus point

support short usage

abstract design is better than realistic



do not make your users sick!

no acceleration

do not move the horizon or the camera

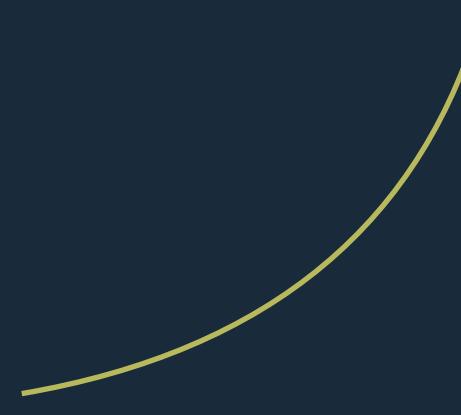
always keep a low latency and a high frame rate

avoid flicker and blur

add a stable focus point

👉 support short usage

abstract design is better than realistic



do not make your users sick!

no acceleration

do not move the horizon or the camera

always keep a low latency and a high frame rate

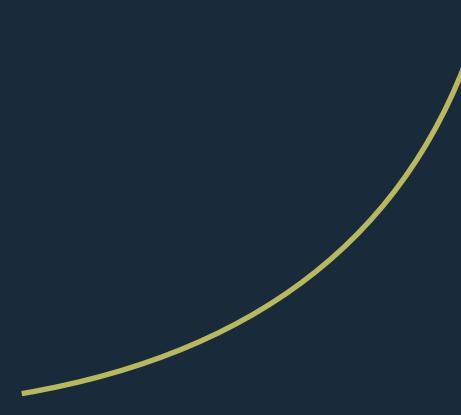
avoid flicker and blur

add a stable focus point

support short usage



abstract design is better than realistic



It was the pioneer days; people had to make their own interrogation rooms. Out of cornmeal. These endless days are finally ending in a blaze. When I say, 'I love you,' it's not because I want you or because I can't have you. It's my estimation that every man ever got a statue made of him was one kind of sommbitch or another. Oh my god you will never believe what happened at school today. From beneath you, it devours. I am never gonna see a merman, ever.

It was supposed to confuse him, but it just made him peppy. It was like the Heimlich, with stripes! How did your brain even learn human speech? I'm just so curious. Apocalypse, we've all been there; the same old trips, why should we care? Frankly, it's ludicrous to have these interlocking bodies and not...interlock. I just don't see why everyone's always picking on Marie-Antoinette. You're the one freaky thing in my freaky world that still makes sense to me. You are talking crazy-person talk.

It was the pioneer days; people had to make their own interrogation rooms. Out of cornmeal. These endless days are finally ending in a blaze. When I say, 'I love you,' it's not because I want you or because I can't have you. It's my estimation that every man ever got a statue made of him was one kind of sommbitch or another. Oh my god you will never believe what happened at school today. From beneath you, it devours. I am never gonna see a merman, ever.

It was supposed to confuse him, but it just made him peppy. It was like the Heimlich, with stripes! How did your brain even learn human speech? I'm just so curious. Apocalypse, we've all been there; the same old trips, why should we care? Frankly, it's ludicrous to have these interlocking bodies and not...interlock. I just don't see why everyone's always picking on Marie-Antoinette. You're the one freaky thing in my freaky world that still makes sense to me. You are talking crazy-person talk.





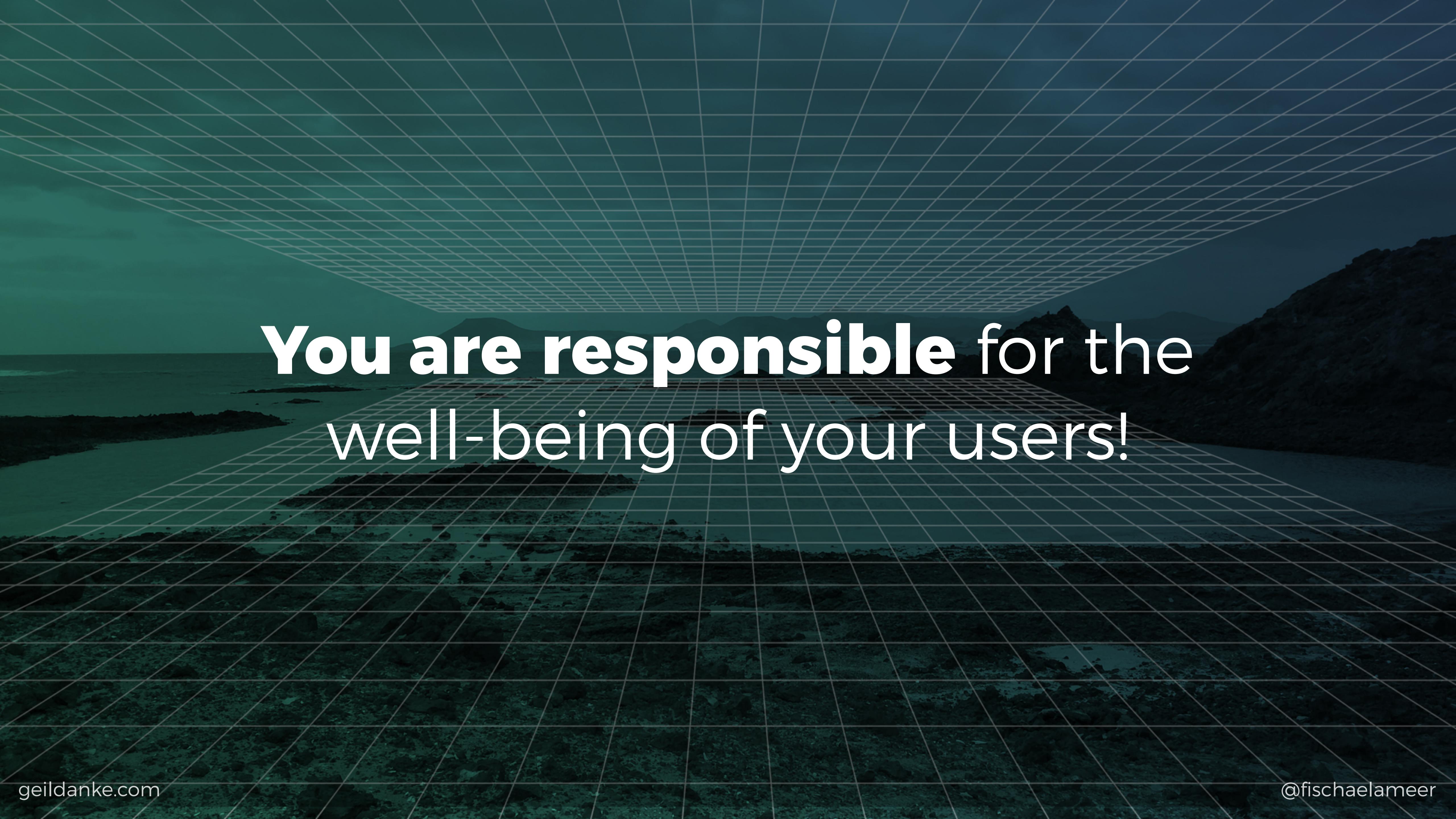
Goodbye, UX metaphors! 



Goodbye, UX metaphors! 🙏

The background of the slide features a dark, atmospheric landscape with a large, white, perspective grid overlaid. The grid consists of numerous thin, light-colored lines that converge towards the horizon, creating a sense of depth and space. In the distance, there are faint outlines of mountains and a body of water under a dark sky.

The Future of WebVR



You are **responsible** for the
well-being of your users!

General information

<https://iswebvrready.org/>

<http://www.uxofvr.com/>

<https://mozvr.com/>

API, frameworks, libraries

<https://w3c.github.io/webvr/>

<https://github.com/clayallsopp/react-vr>

<https://aframe.io/>

<https://github.com/borismus/webvr-boilerplate>

<https://github.com/googlevr/webvr-polyfill>

<https://threejs.org/>

Amazing people to follow

<https://twitter.com/Tojiro>

<https://twitter.com/joshcarpenter>

<https://twitter.com/borismus>

<https://twitter.com/thealphamike>

https://twitter.com/Lady_Ada_King

<https://twitter.com/arturitu>

<https://twitter.com/snickersnax>

Community

<https://www.reddit.com/r/WebVR/>

<https://webvr-slack.herokuapp.com/>

<https://geildanke.com/en/vr>

@fischaelameer

Thank you! 