



EASY VR WITH REACT AND A-FRAME

James Porter @complexview

WHY VR?



- Massive potential in education, collaboration and games
- Something amazing about creating 3D worlds which can be explored... VR heightens this.
- And becoming available: £5 for smartphone version...

ABOUT ME (AND VR)

- Not 3D specialist
 - Many years ago did a single CS assignment with OpenGL (your phone will be way more powerful than the tech I was using then)
 - Played around a little with Processing in 3D
 - Built VR experience at Hackathon earlier this year (with the stuff I'm going to introduce)
- I keep ending up as Javascript developer
 - Starting to really appreciate js: ES 2017+, tooling, React, RxJS, Redux, React Native etc
- Looking for a way into 3D/VR; but has to be simple and efficient

WHY NOT VR (IN THIS SESSION)

- Want to focus on ideas... let's not get distracted with setup etc





DEMO

amimetic.co.uk/elegy/index.html or goo.gl/jSnmea

THE PLAN

- Motivation or is there a way to get into VR for the non-specialist?
- Introduction to A-Frame
- Using React with A-Frame
- Three mini sessions:
 - 1. Hello (Virtual) World
 - 2. Interactive App
 - 3. Data Visualisation

Alex St. John: Shut up and be grateful for your 80 hour weeks

By Dan Pearson

 Recommend 191  Tweet  Share { 23 }

THU 21 APR 2016 7:15PM GMT / 3:15PM EDT / 12:15PM PDT
[#PEOPLES](#) [#JOBS](#)

"Wage slaves" should "shake off mental shackles" says multi-millionaire. [UPDATE: St. John's daughter blasts his "toddler meltdown"]

[Update] By now you've probably all read Alex St. John's awful post about crunch in the games industry, but you may not be aware that Amilia St. John (his daughter who also got into tech) has replied with a



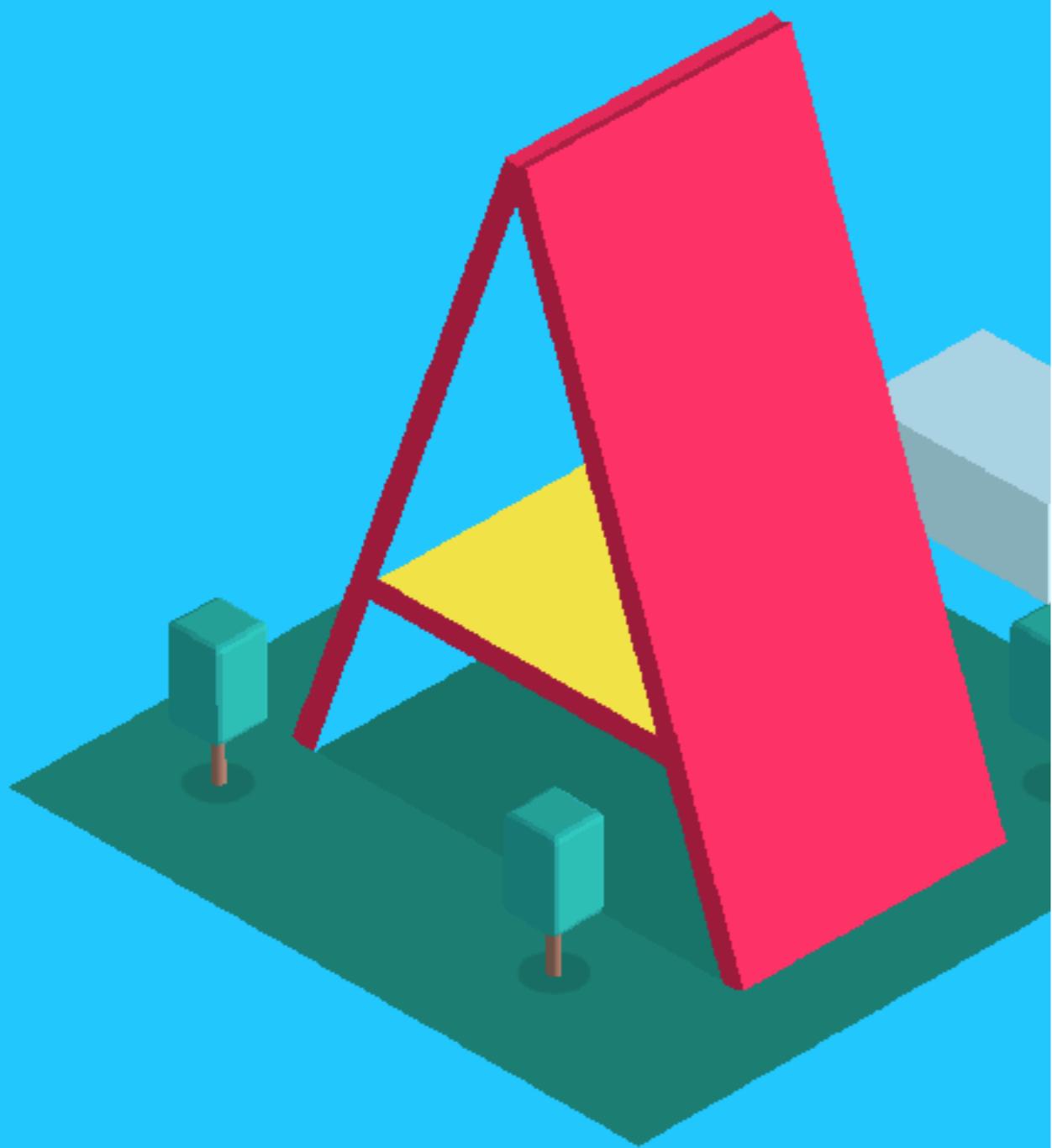
Don't be in the game industry if you can't love all 80 hours/week of it — you're taking a job from somebody who would really value it.

EFFICIENCY?

John Maynard Keynes,
Economic Possibilities for our
Grandchildren
(1930)*

shared as possible. Three-hour shifts or a fifteen-hour week may put off the problem for a great while. For three hours a day is quite enough to satisfy the old Adam in most of us!

- Looking for something that...
 - If it works for side project/hackathon *a fortiori*...
 - Modern javascript experience: HMR (or at least Live Reloading), ES6+, easy use of 3rd party code/NPM



A-FRAME

A-FRAME

- “Entity-Component-System”
- Attach geometry, material, behaviours etc
- Really easy to get something built
- Just write HTML, converted to 3D scene

HELLO WORLD

```
<meta charset="utf-8">  
<title>Hello, World! • A-Frame</title>  
<meta name="description" content="Hello, World! • A-Frame">  
<script src="../../dist/aframe.js"></script>  
</head>  
<body>  
  <a-scene>  
    <a-sphere position="0 1.25 -1" radius="1.25" color="#EF2D5E"></a-sphere>  
    <a-box position="-1 0.5 1" rotation="0 45 0" width="1" height="1" depth="1" color="#4CC3D9"></a-box>  
    <a-cylinder position="1 0.75 1" radius="0.5" height="1.5" color="#FFC65D"></a-cylinder>  
    <a-plane rotation="-90 0 0" width="4" height="4" color="#7BC8A4"></a-plane>  
    <a-sky color="#ECECEC"></a-sky>  
    <a-entity position="0 0 3.8">  
      <a-camera></a-camera>  
    </a-entity>  
  </a-scene>
```

A-FRAME

Examples

Hello World

Hello Metaverse

360° Image

360° Image Gallery

360° Video

Anime UI

Audio Visualization

ShopifyVR

Spheres & Fog

A-Painter 

Tracked Controllers 



LEARN MORE

- Great examples (with source code)
- Really friendly slack

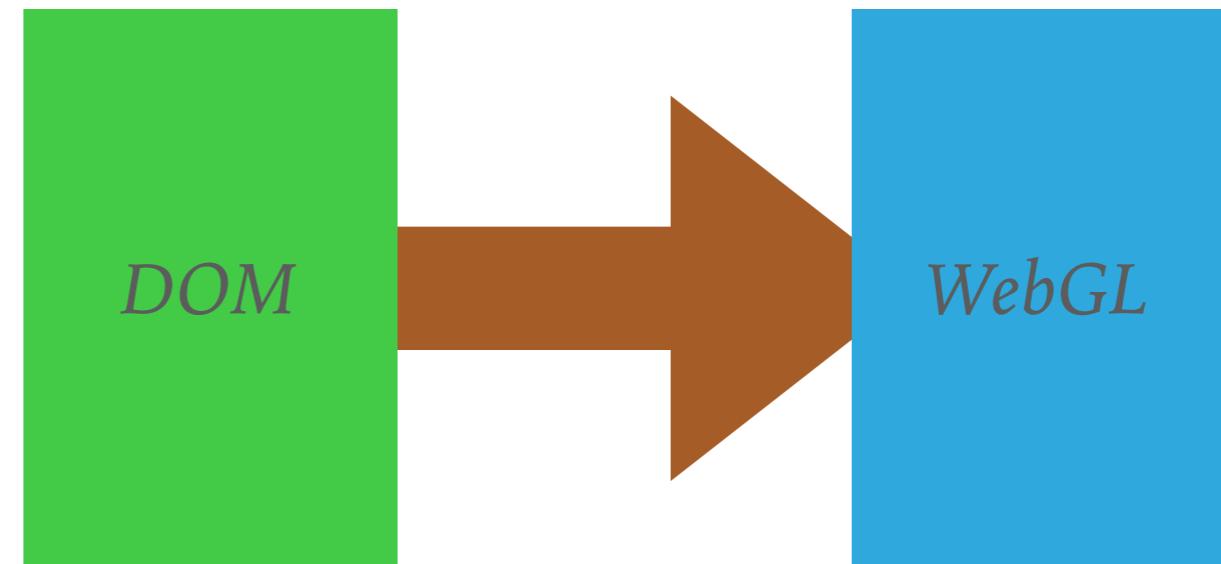
A-FRAME...

A-FRAME

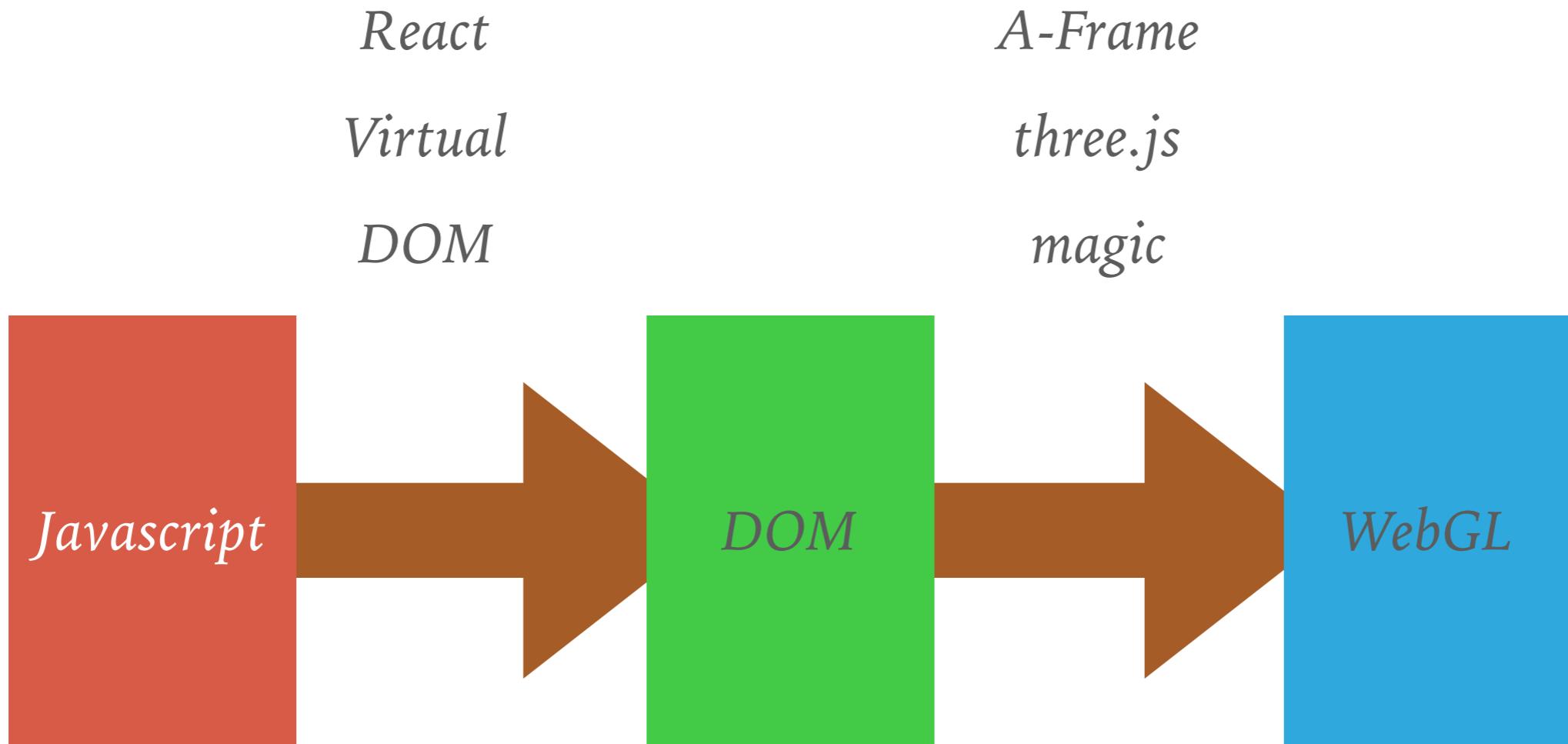
A-Frame

three.js

magic



+ REACT?



KEY BENEFITS

- Encapsulate
- Reuse
- Modern tooling (Webpack, Babel etc)
- State(React itself but when we need more e.g. Redux)

TENSIONS

- ‘Components’
- A-Frame mixin system vs React composition
- Global-ish asset loading

AFRAME-REACT

.....

- Library for combining React and A-Frame
- <https://github.com/ngokevin/aframe-react>
- Real, substantial stuff done with it!



BASICS

<https://github.com/ngokevin/aframe-react>

API

`aframe-react` ships with `Scene` and `Entity` React components, which are all we really need.

`<Scene {...components}>`

The `Scene` React component wraps `<a-scene>`:

```
<Scene>
  <Entity/>
</Scene>
```

`<Entity {...components}>`

The `Entity` React component wraps `<a-entity>`.

```
<Entity geometry={{primitive: 'box'}} material='color: red'/>
```

Primitives

To render A-Frame primitives with all of the `aframe-react` magic, pass the `primitive` prop with the name of the primitive:

```
<Entity primitive='a-box' onClick={() => { console.log('Clicked!'); }}/>
```

GETTING STARTED

- Go to: <https://github.com/jamesporter/React-AFrame-Intro>
- or <https://goo.gl/bA5cpO>
- Links to 3 main repos + other resources

**HELLO
(VIRTUAL)
WORLD**

one



THE COMPLETE APP (SORT OF)

```
class Starter extends React.Component {
  render() {
    return <Scene>
      <Camera>
        {/*<Cursor/>*/}
      </Camera>
      <BaseLights/>

      <Translation x={-2} y={-1} z={-2}>
        <Cube size={2}/>
        {/*<Sphere size={1}/>*/}
      </Translation>

      <Translation x={2} y={-1} z={-2}>
        <Cube size={2} colour="#336699"/>
      </Translation>
      <Sky/>
    </Scene>;
  }
}

ReactDOM.render(<Starter/>, document.querySelector('.scene-container'));
```

CUBE.JS

```
import {Entity} from 'aframe-react';
import React from 'react';

export default ({size = 1, colour = "#DD8800"}) => <Entity
geometry={{`primitive: box; width: ${size}, height: ${size}, depth: ${size}`}}
material={{color: colour, shader: 'flat'}}>;
```

TRANSLATE.JS

```
.....import {Entity} from 'aframe-react';
import React from 'react';

export default ({x = 0,y = 0,z = 0, children}) => <Entity
position={`${x} ${y} ${z}`}>
  {children}
</Entity>;
```

INTERACTIVE APP

two



MINECRAFT IN 30 LINES OF CODE*

```
class Starter extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      locations: generateLocations(30)
    };
  }

  render() {
    return <Scene>
      <Camera>
        <Cursor/>
      </Camera>
      <BaseLights/>

      {this.state.locations.map((loc, idx) => <Translation {...loc}>
        <Cube size={2} colour="#336699" onClick={() => this.removeCube(idx)} />
      </Translation>)}

      <Sky/>
    </Scene>;
  }

  removeCube(idx){
    this.setState({
      locations: [...this.state.locations.slice(0, idx), ...this.state.locations.slice(idx + 1)]
    })
  }
}

ReactDOM.render(<Starter/>, document.querySelector('.scene-container'));
```

INTERACTIVECUBE.JS

```
export default ({size = 1, colour = "#DD8800", onClick}) => <Entity  
  geometry={`primitive: box; width: ${size}; height: ${size}; depth: ${size}`}  
  material={{color: colour, shader: 'flat'}}  
  onClick={onClick}  
/>;
```

See also onMouseEnter, onMouseLeave for e.g. hover effects

```
onMouseEnter={() => this.setState({hover: true})}  
onMouseLeave={() => this.setState({hover: false})}
```

DATAVIS

three



DATA VISUALISATION

```
<Translation z={-3}>
  {this.state.data.map((value, idx) => {
    const loc = {
      x: idx - 13,
      y: value/4 - 5,
      z: -8 * Math.cos(idx/26 * 3.14 - 3.14/2)
    };

    //never do this!
    const i = idx + 1.0;
    const pad = n => n < 10 ? "0" + Math.round(n) : Math.round(n);
    const colour = "#" + pad(33 * i/13) + pad(66 * i/26) + pad(99 * i/26);

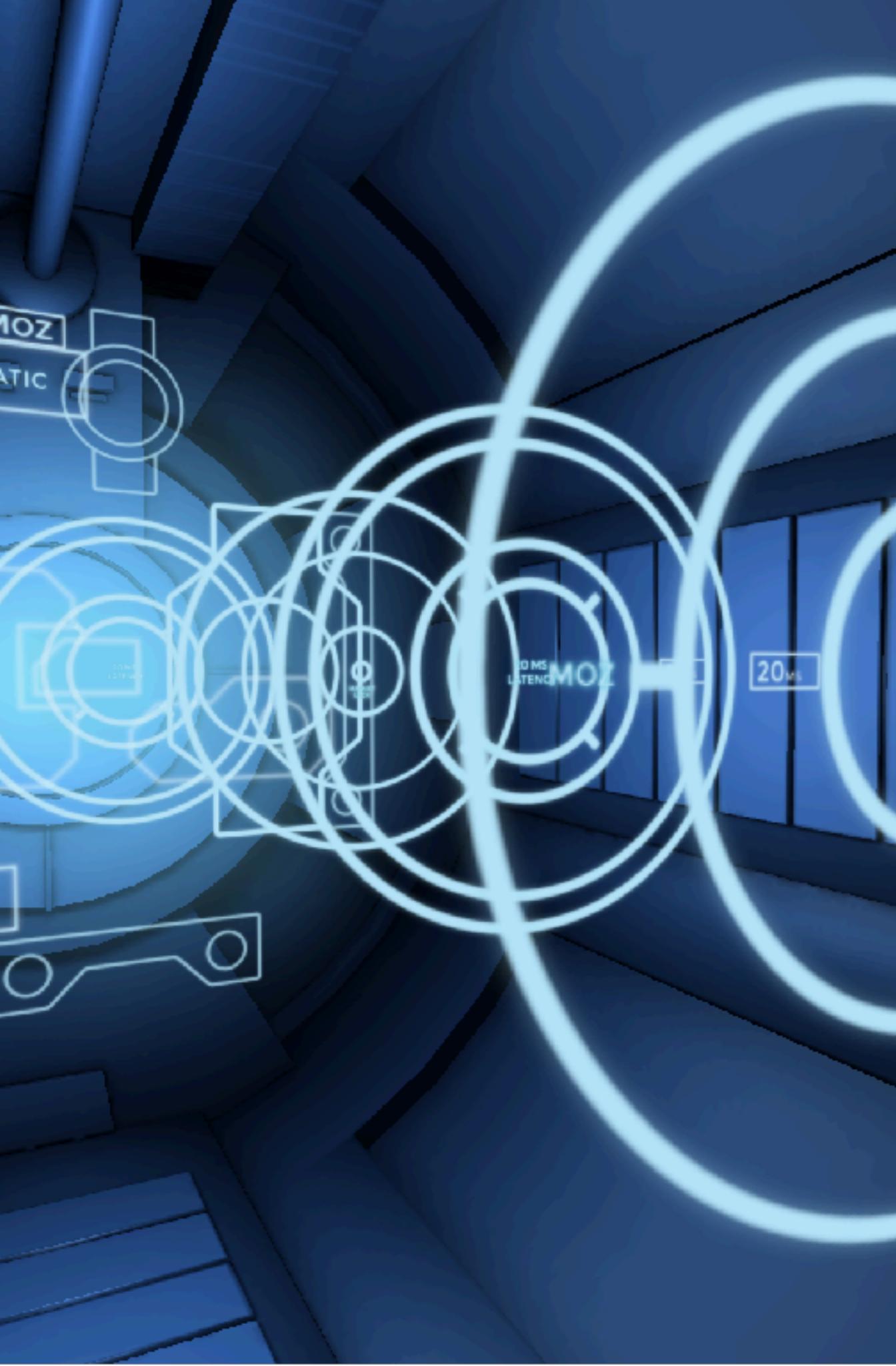
    return <Translation {...loc}>
      <Box height={value/2} width={0.8} depth={0.2} colour={colour} />
    </Translation>
  )})
</Translation>
```

END

Questions/Thoughts?

FURTHER IDEAS

bonus



2.5D

- 3D is hard and really slow
(even if you are good at it)
- So just do 2D
- With heavy use of transparency
- In Layers

LIGHTING

```
<Entity light={{type: 'point', intensity: 0.4, distance: 2, color: "#FFCC22"}}
position={[0, 0, 0.2]}>
```

IMAGE ASSETS/TEXTURES

```
<Entity geometry={{primitive: 'box', width: 10, height: 10, depth: 0.1}}
        material={{src: "url(img/map.png)"}}
        position={"0 0 -1"}
    />
```

- Can be really simple
- (Though in practice probably better to pre-load assets and refer to them by ids. But can do this with A-Frame.)

A-FRAME-REACT IS REALLY THIN WRAPPING

- 2 files
- Main one 144 lines of code
- Main things:
 - Attach events
 - Convert props to A-Frames more primitive serialisation format
- <https://github.com/ngokevin/aframe-react/blob/master/src/index.js>