# Shay Keinan

@Shay_Keinan

VR
history
& basic

React VR

Three.js

# Lucid Dreaming

# Mandukya Upanishad

500 BC

**Saint Augustine**

415 BC

**Dreams and the Means of directing them.**

Marquis d'Hervey de Saint-Denys

1867

# The Stereoscope

Charles Wheatstone

1838

PYGMALION'S SPECTACLES

By STANLEY G. WEINBAUM

*Author of "The Black Flame," "A Martian Odyssey," etc.*

By Continental Publications, Inc.

# Pygmalion's Spectacles

Stanley Weinbaum

1935

**The Sensorama**

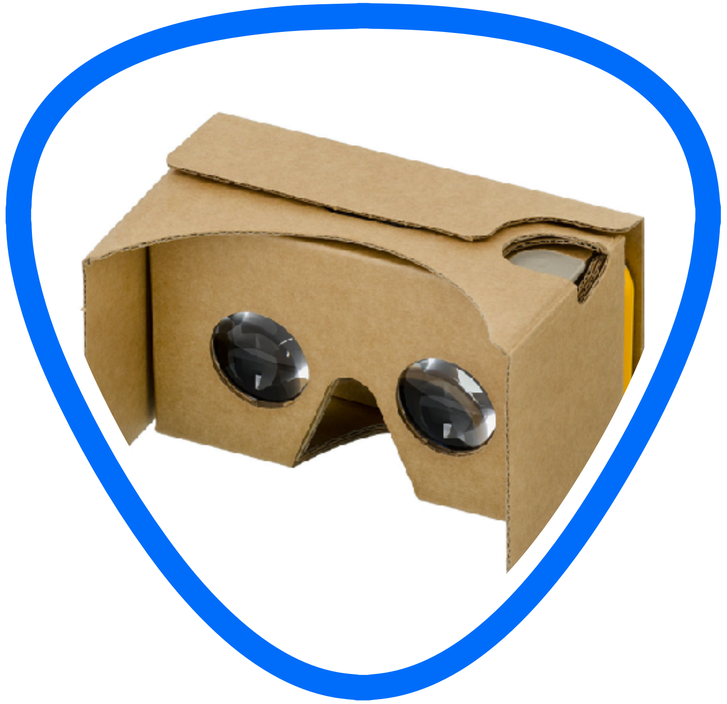
**Sega VR**

**Nintendo Virtual Boy**

# Virtual Reality

Jaron Lanier

1987



Google cardboard

Gear VR

**HTC Vive**

**Playstation VR**

**Oculus Rift**

oculus Go

oculus

from facebook

VIVE

# VIRTUAL REALITY

# VIRTUAL REALITY

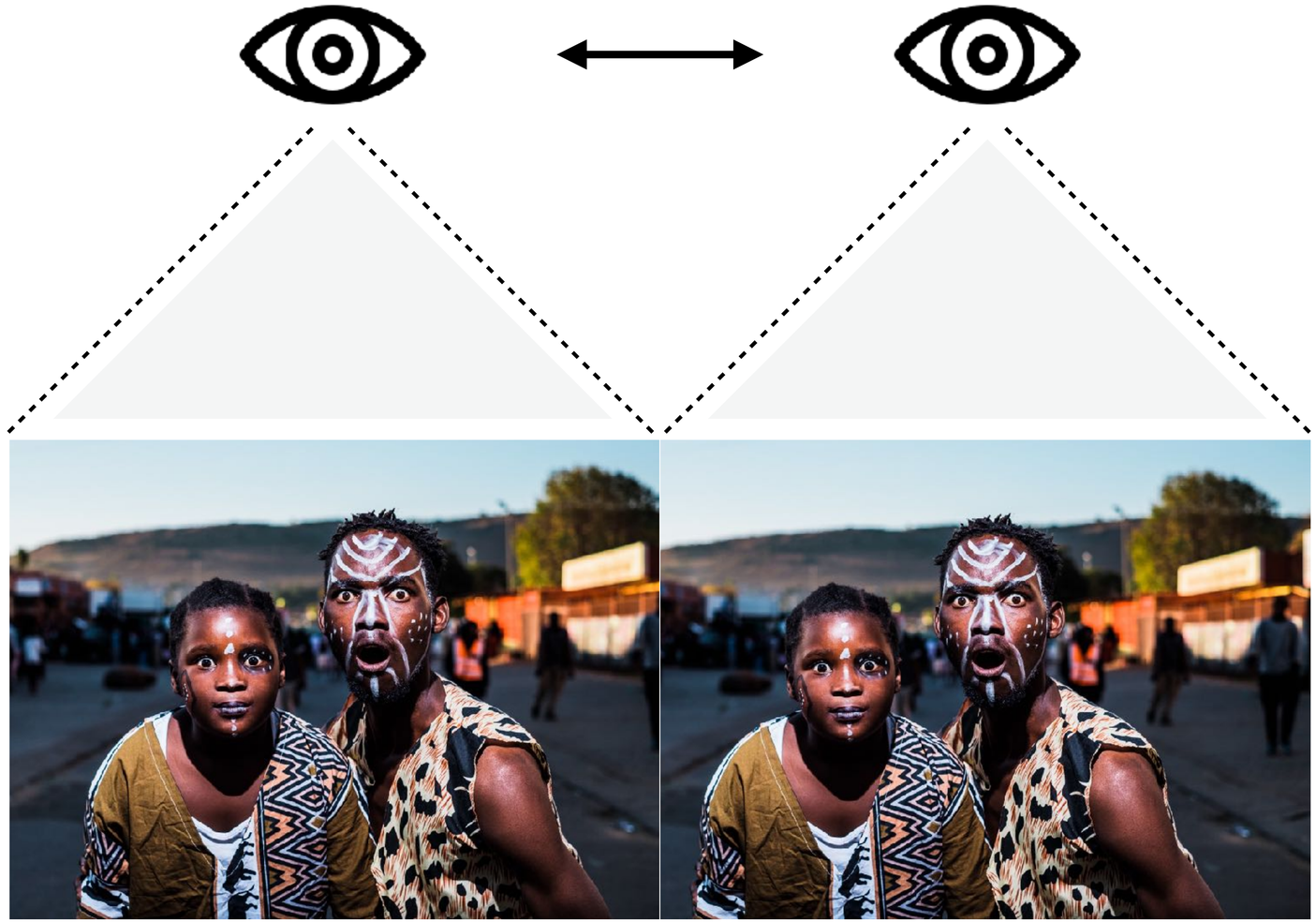
## CONCEPTS

# Stereoscopic Imaging

# Stereoscopic Imaging

# Stereoscopic Imaging

# Stereoscopic Imaging



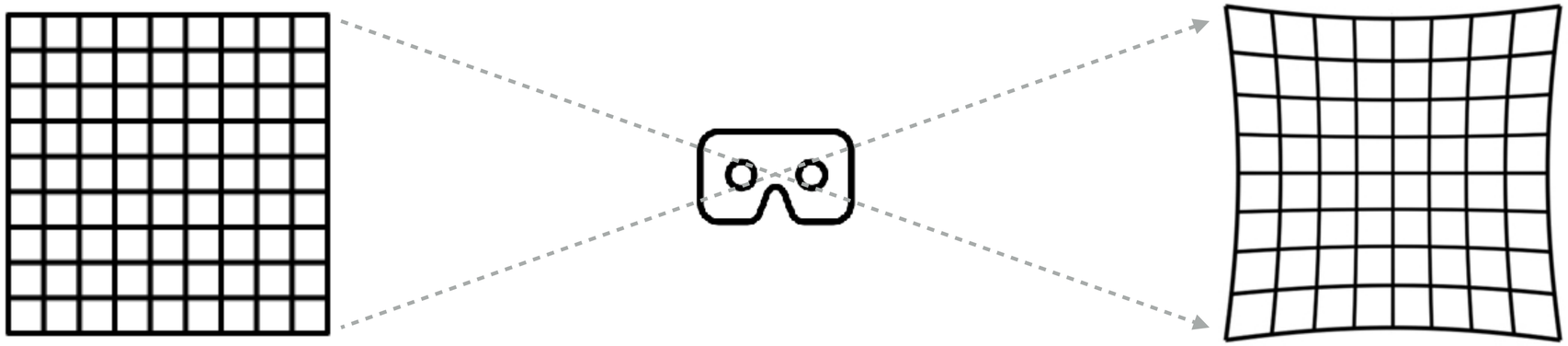
# Headset Lenses

Oculus
Rift

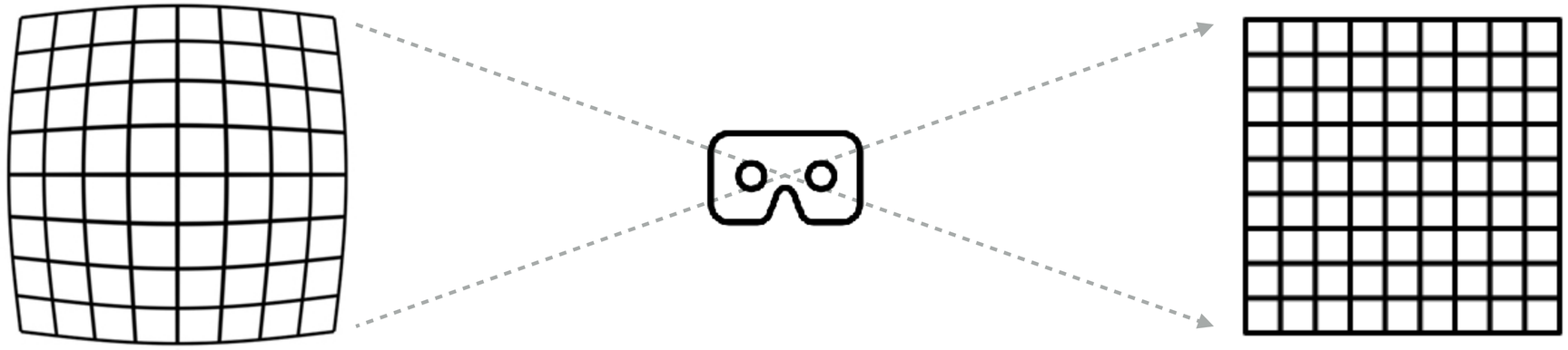
HTV
Vive

Google
Cardboard

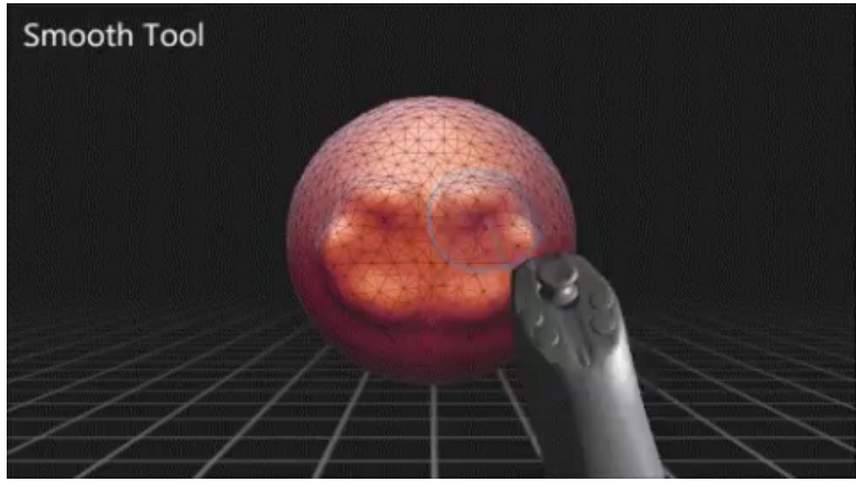
# Stereoscopic Imaging

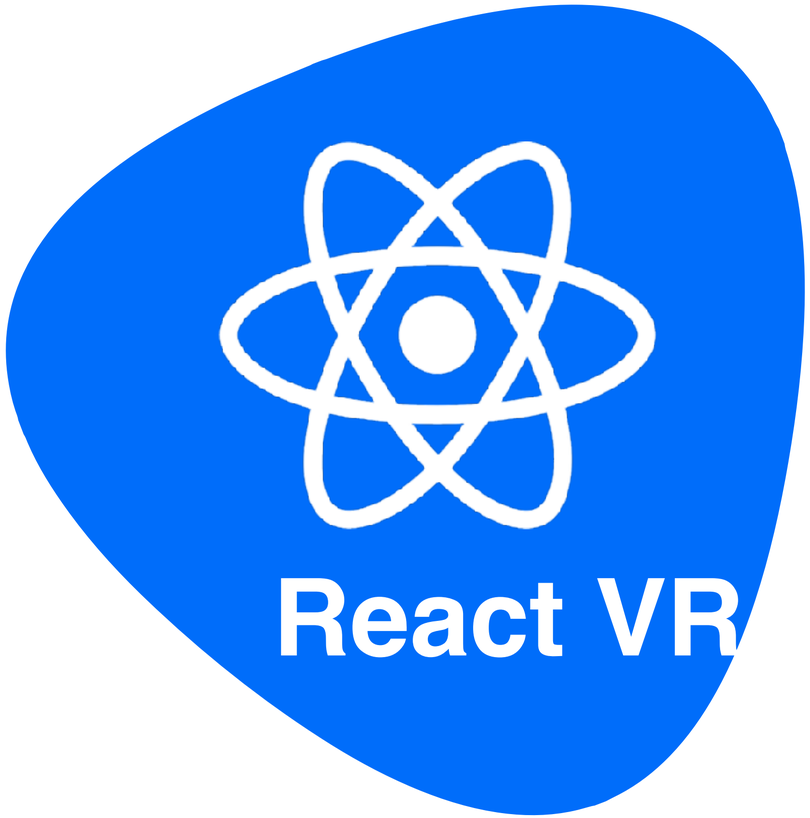


# Stereoscopic Imaging



# Body Movement




React VR

# React VR

## The Rendering Engines

Three.js

React Native

CUBE SLAM
LEVEL 1
YOU
BOB
BOB
GHOST BALL

# Getting started

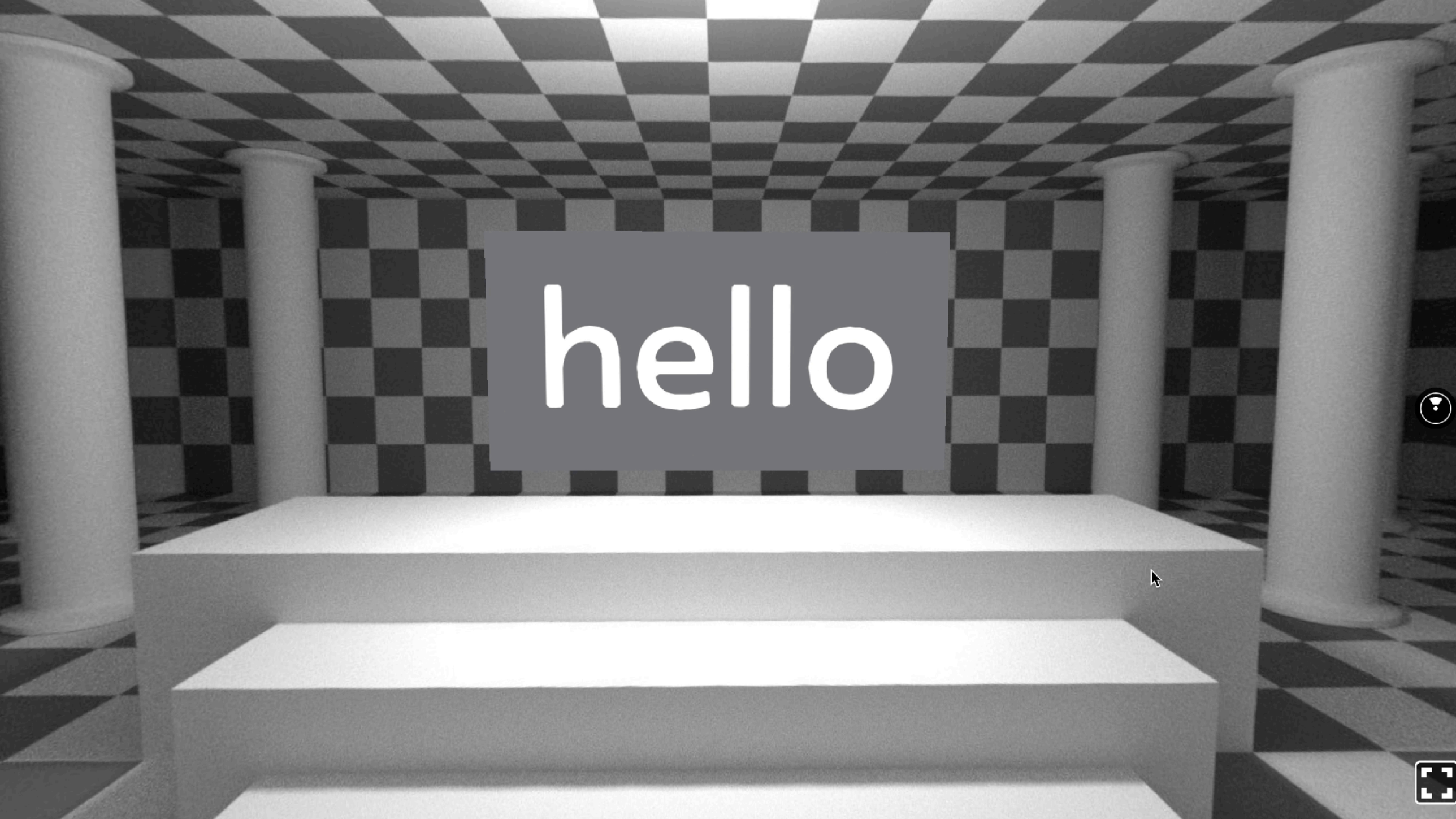
```
npm install -g react-vr-cli
```

```
react-vr init WelcomeToVR
```

```
cd WelcomeToVR
```

```
npm start
```

```
http://localhost:8081/vr/index.html
```

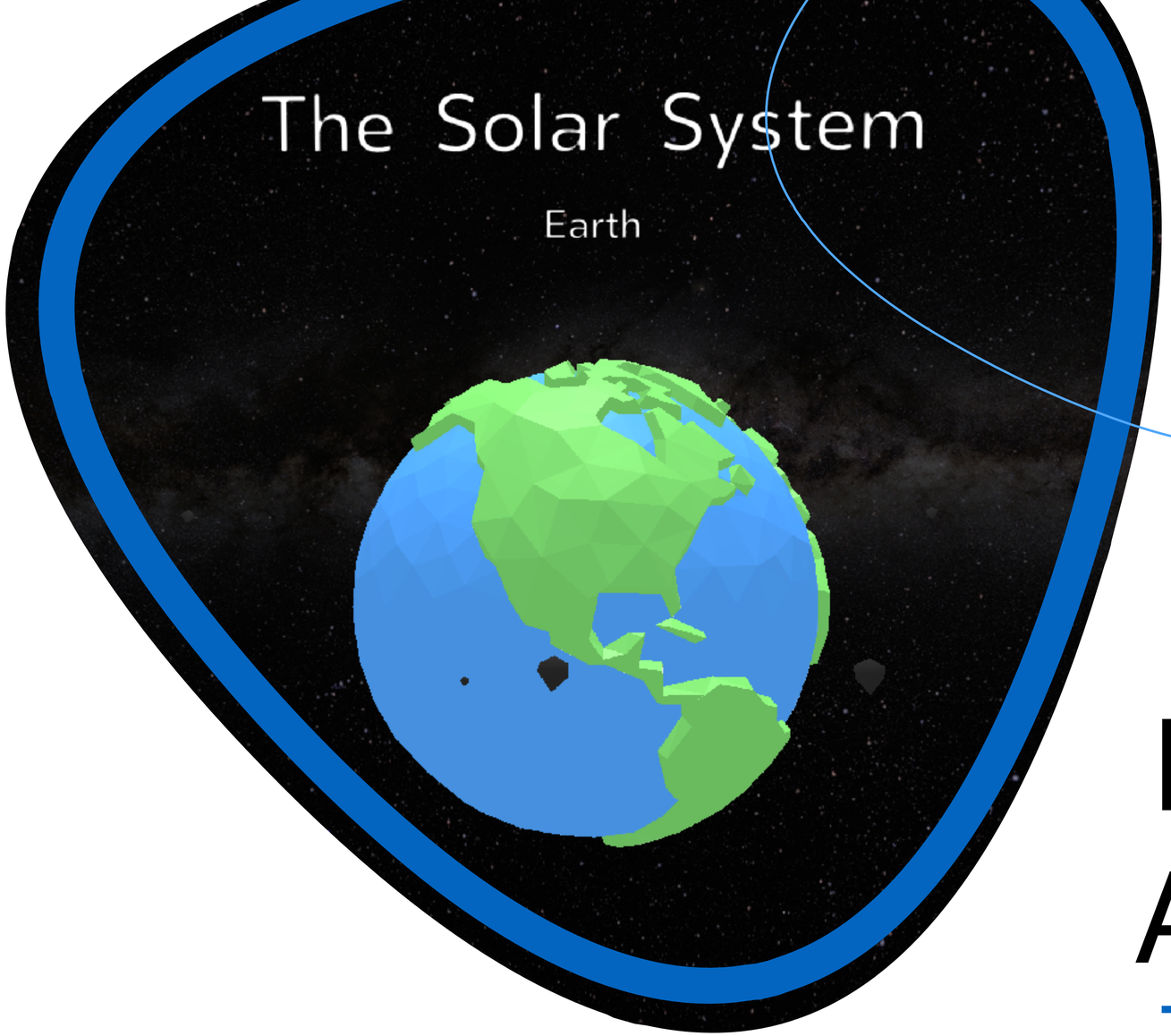

hello

# Folder Structure

- __tests__
- node_modules
- static_assets
- vr
  - Index.html
  - Client.js
- Index.vr.js

# index.vr.js

```
import React from 'react';
import { AppRegistry, asset, Pano, Text, View } from 'react-vr';

class WelcomeToVR extends React.Component {
  render() {
    return (
      <View>
        <Pano source={asset('chess-world.jpg')}/>
        <Text
          style={{
            fontSize: 0.8,
            textAlign: 'center',
            transform: [{translate: [0, 0, -3]}],
          }}>
          hello
        </Text>
      </View>
    );
  }
};

AppRegistry.registerComponent('WelcomeToVR', () => WelcomeToVR);
```



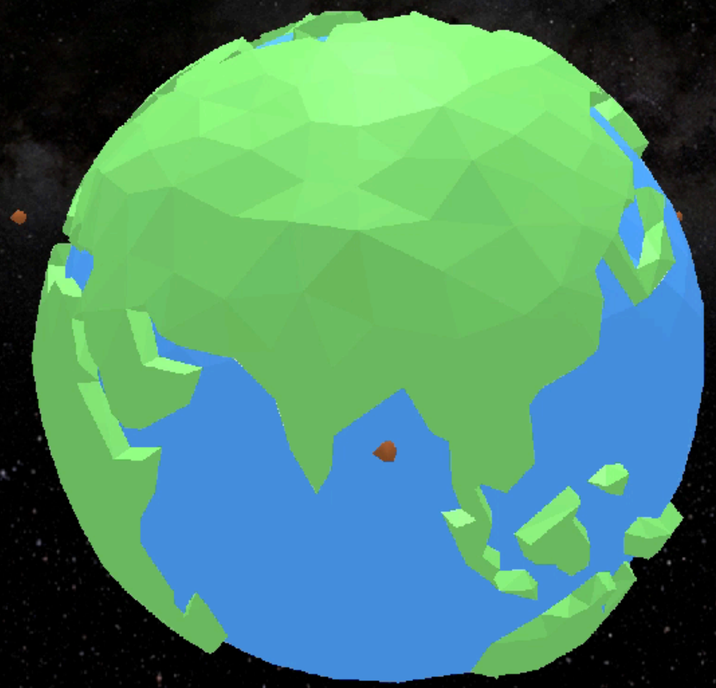

# My Solar System App

First steps with React VR

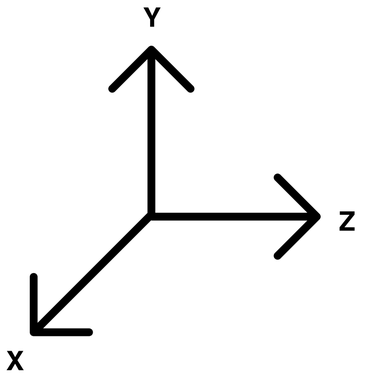
**bit.ly/solar-demo**

# The Solar System

Earth

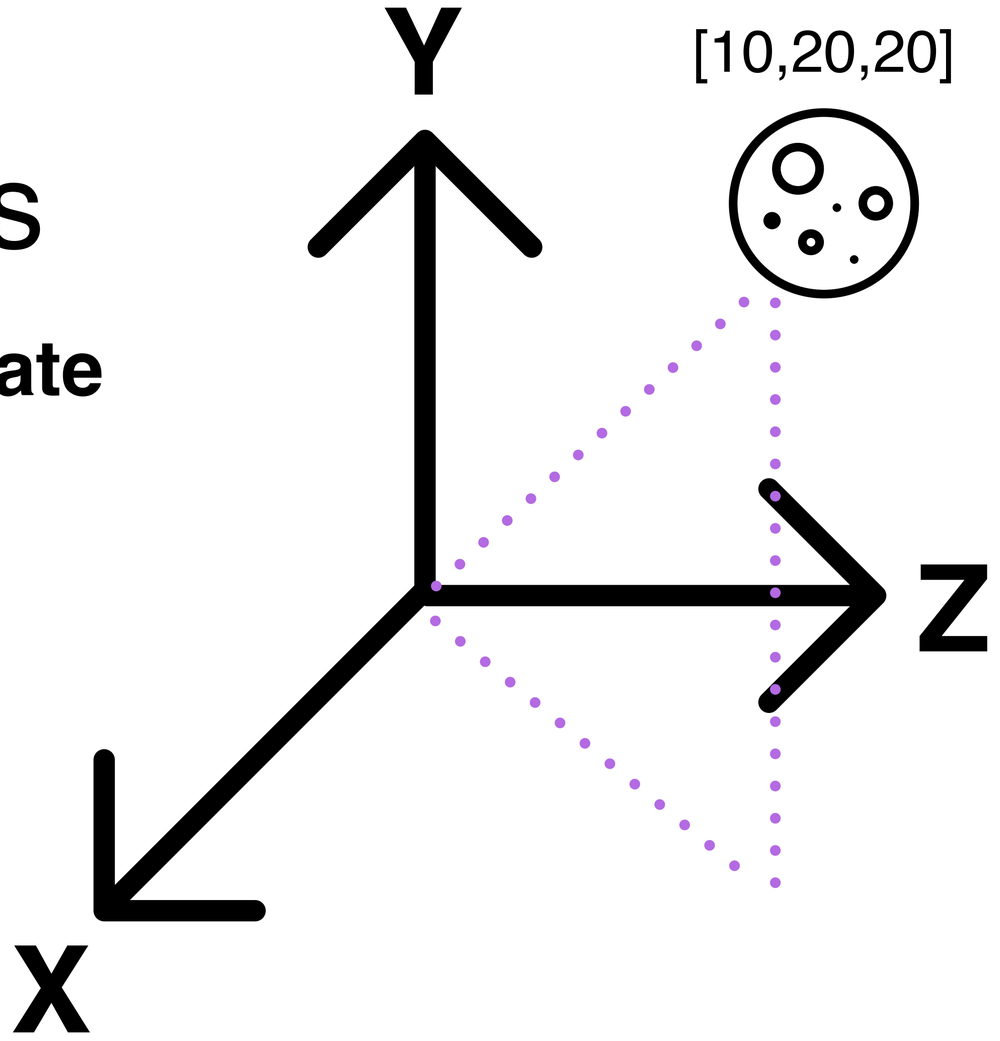


# 3D Coordinates



# Transformations

**Translate, Scale & Rotate**




transform: [{translate: [10,20,20]}]

# Core Components

**<View />**

**<Text />**

**<Image />**

# Core Components

<View />

<Text />

<Image />

# Core Components

<View />

<Text />

<Image />

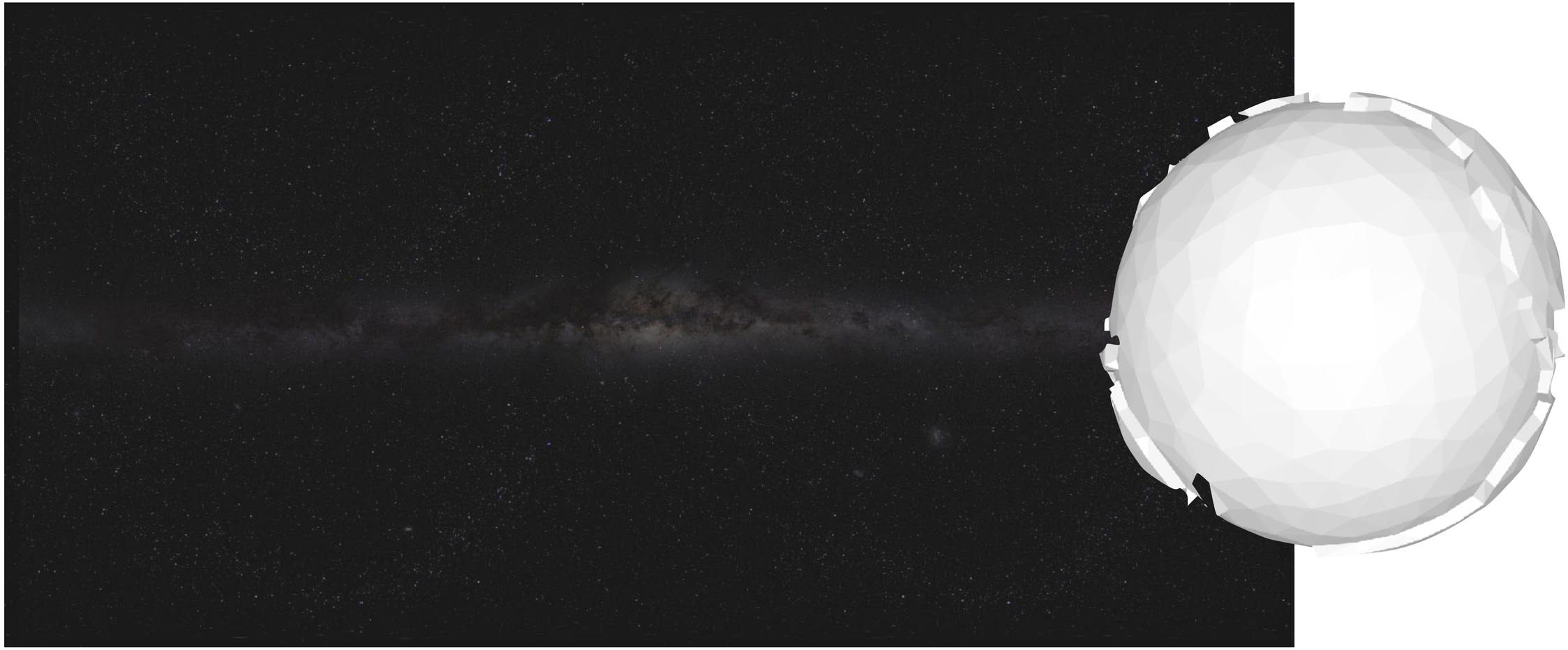
# Core Components

<View />

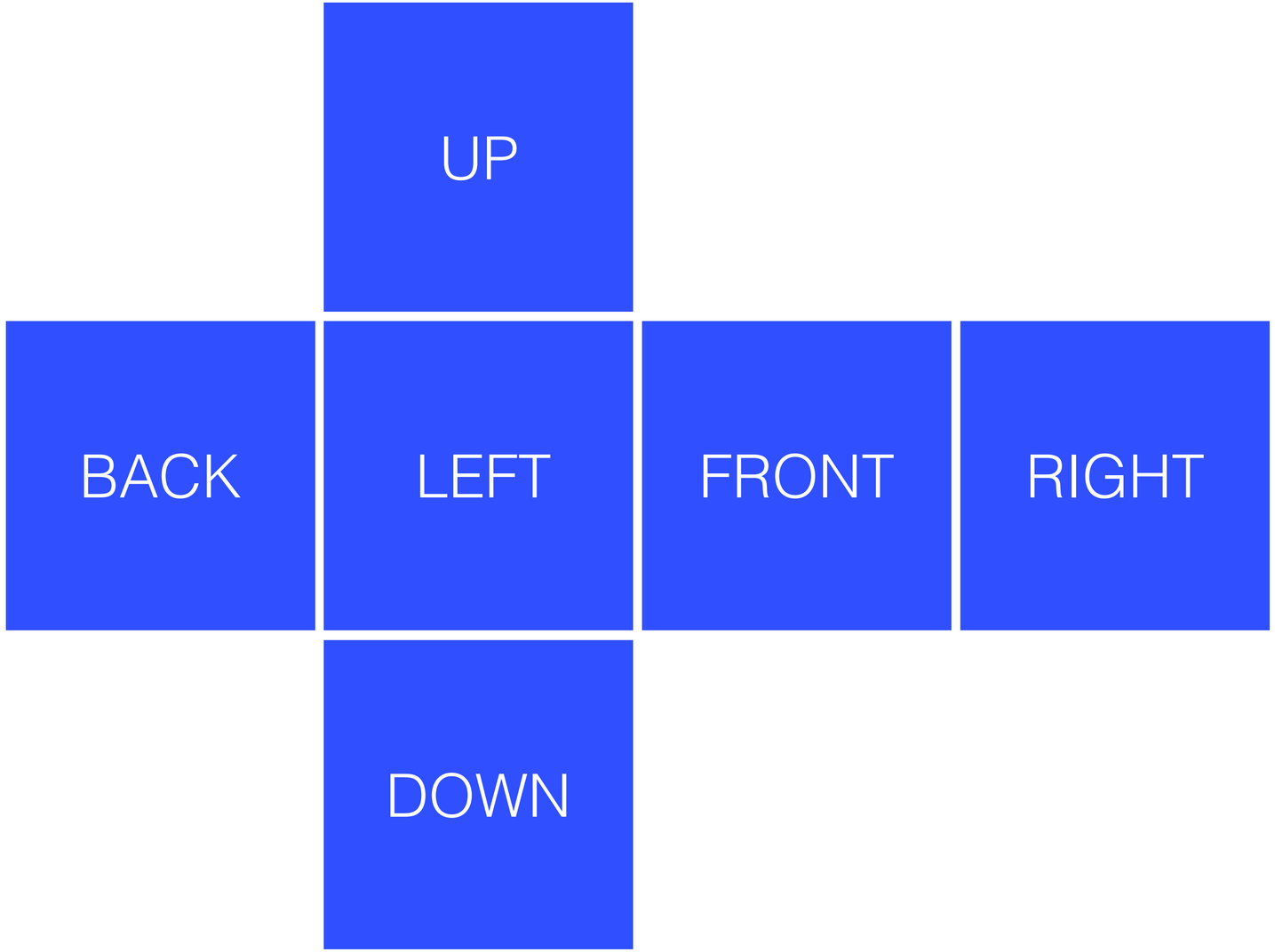
<Text />

<Image />

# Pano



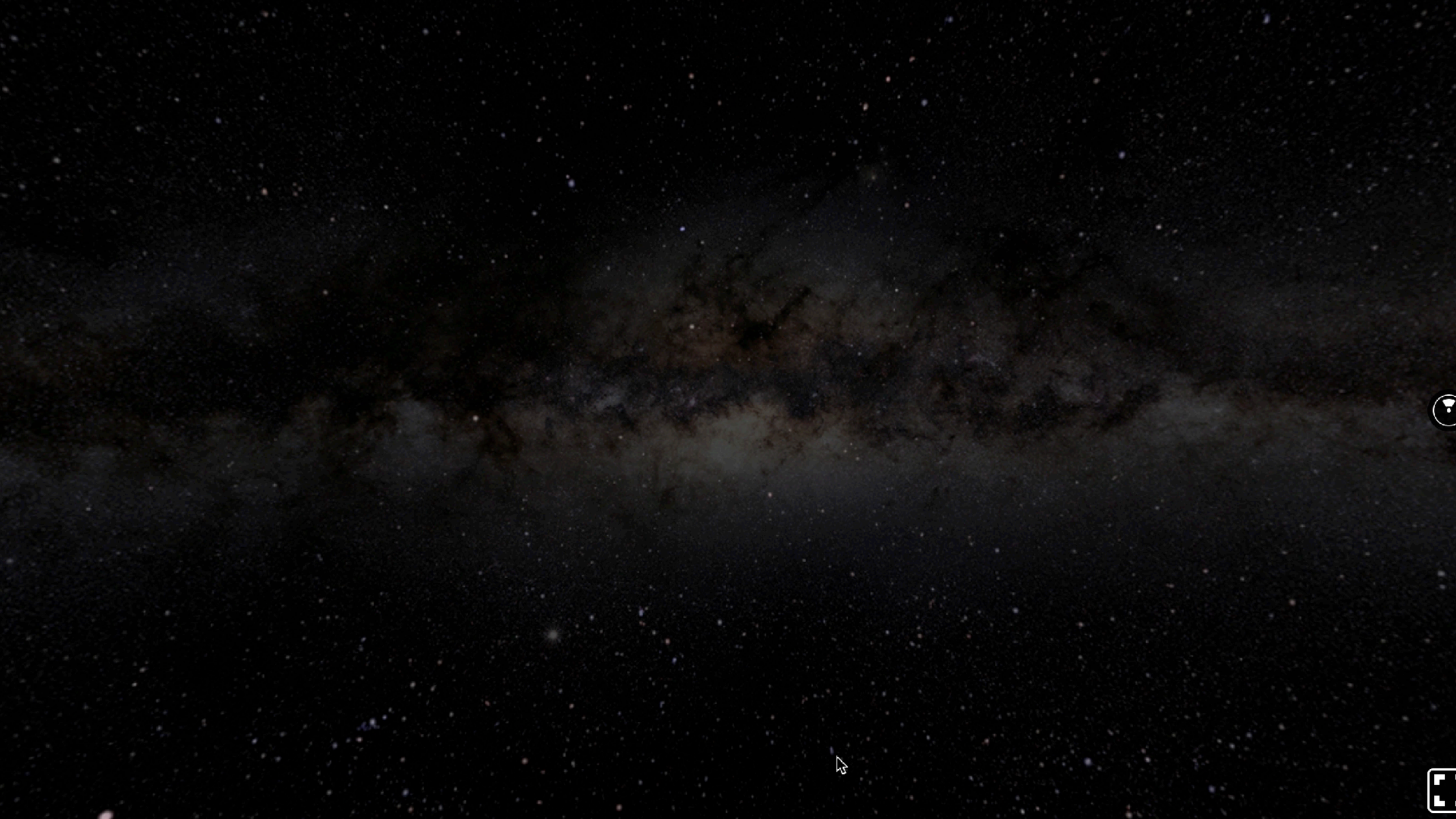
# Pano



# Pano

index.vr.js

```
render() {
  return (
    <View>
      <Pano source={asset('stars.jpg')}/>
    </View>
  );
}
```

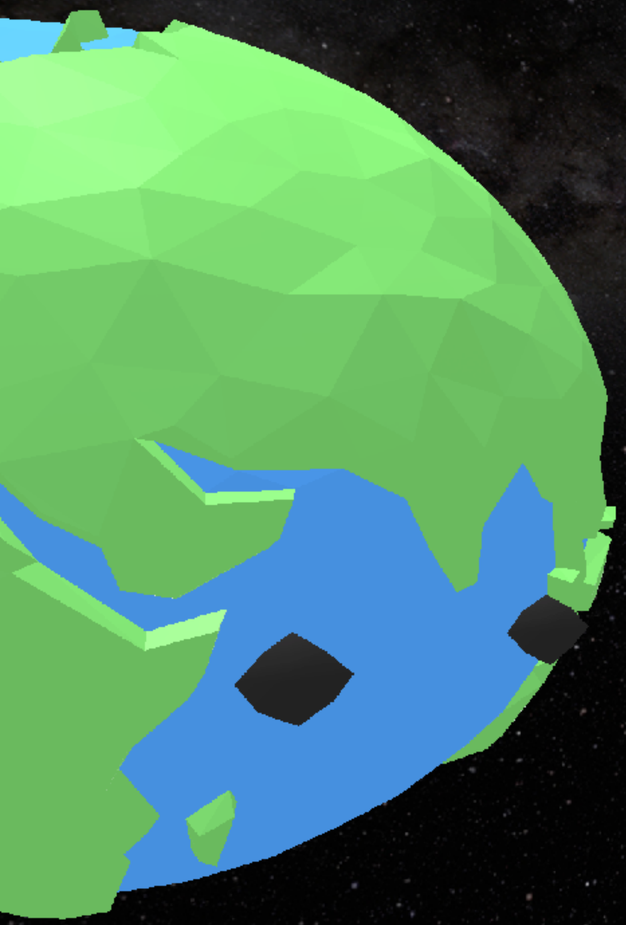


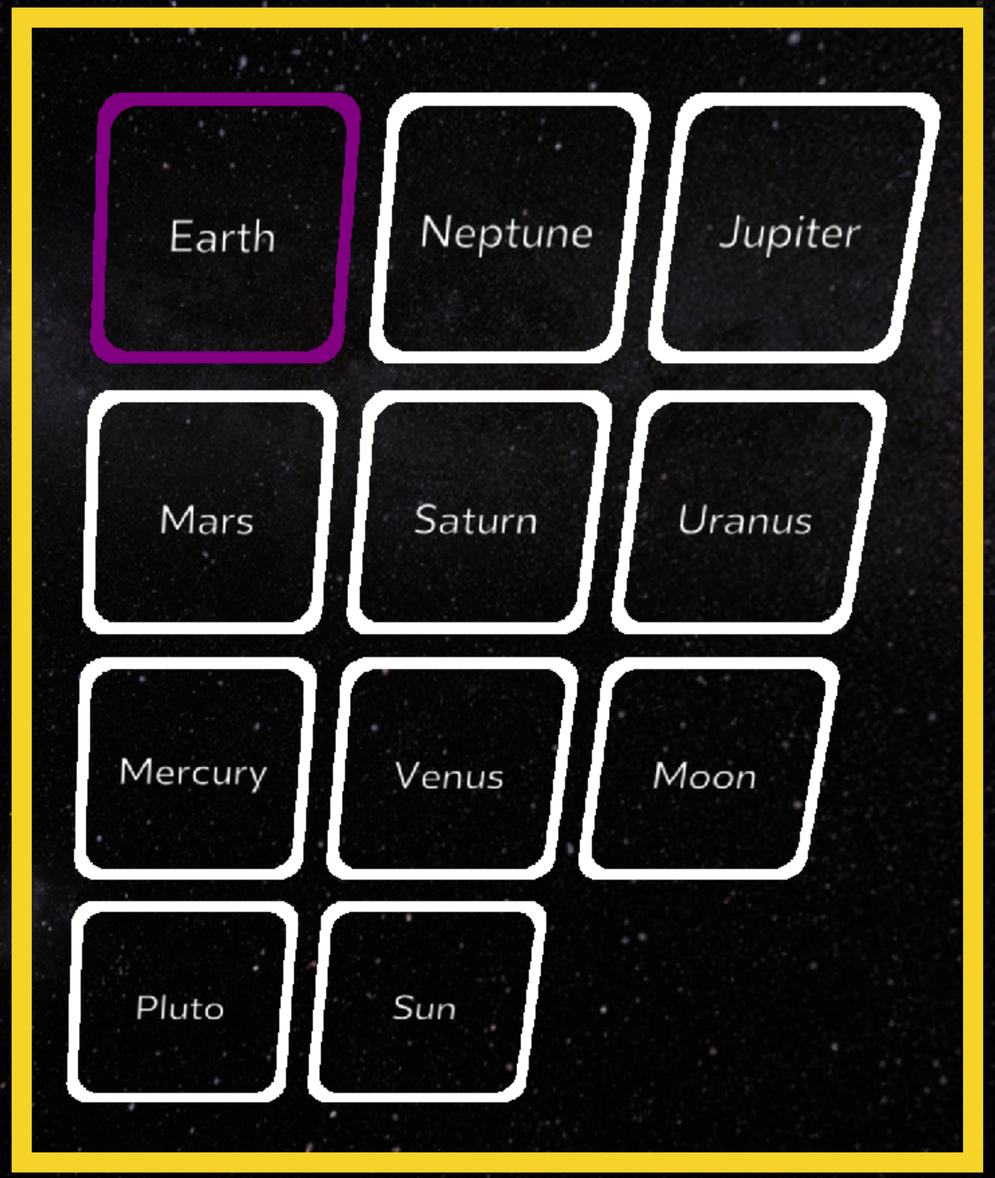
# Styling with Flex box

The same styling and layout system across **web**, **React Native**, and **VR**

01

02




Earth
Neptune
Jupiter
Mars
Saturn
Uranus
Mercury
Venus
Moon
Pluto
Sun

<Menu />

# <Menu />

```
menu() {
  return (
    <View
      billboarding={'on'}
      style={styles.menu}>
      { Object.keys(planets).map((planet) =>(
        <View style={styles.planetBtn}>
          <Text style={styles.planetBtnLabel}>{planet}</Text>
        </View>
      )) }
    </View>
  );
}
```

# <Menu />

```
menu() {
  return (
    <View
      billboarding={'on'}
      style={styles.menu}>
      { Object.keys(planets).map((planet) =>(
        <View style={styles.planetBtn}>
          <Text style={styles.planetBtnLabel}>{planet}</Text>
        </View>
      )) }
    </View>
  );
}
```

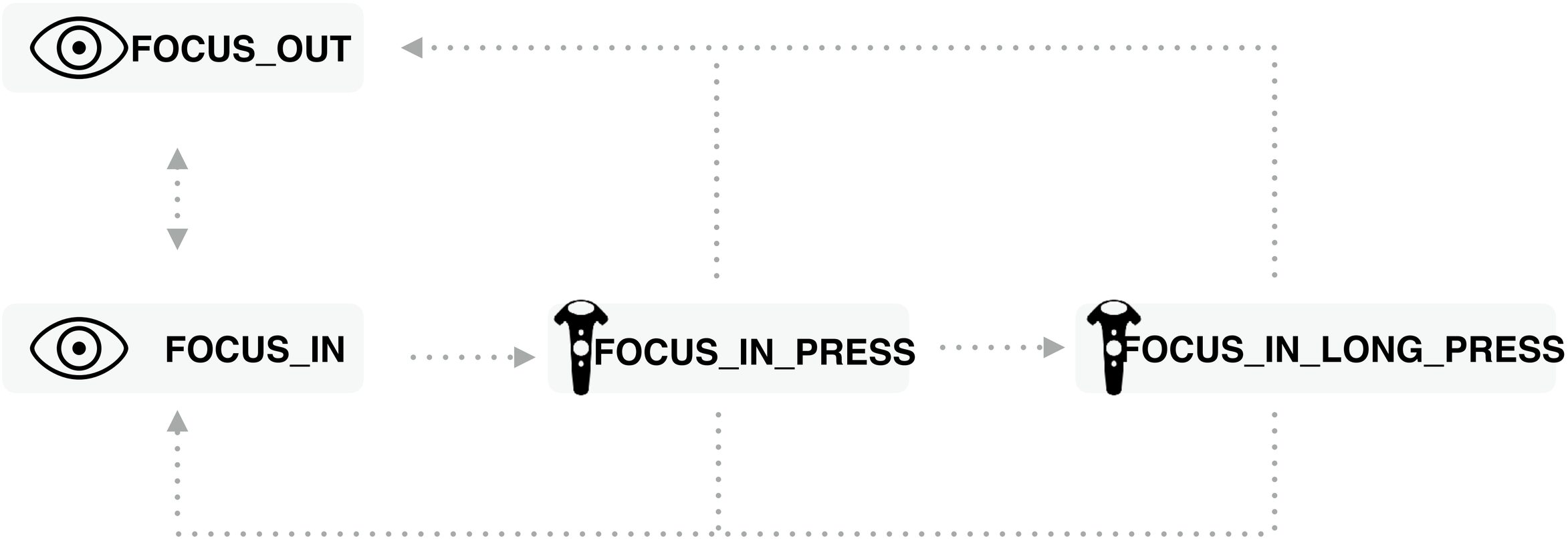
Billboard

# The Solar System

Earth

# Interactions

# Interactions

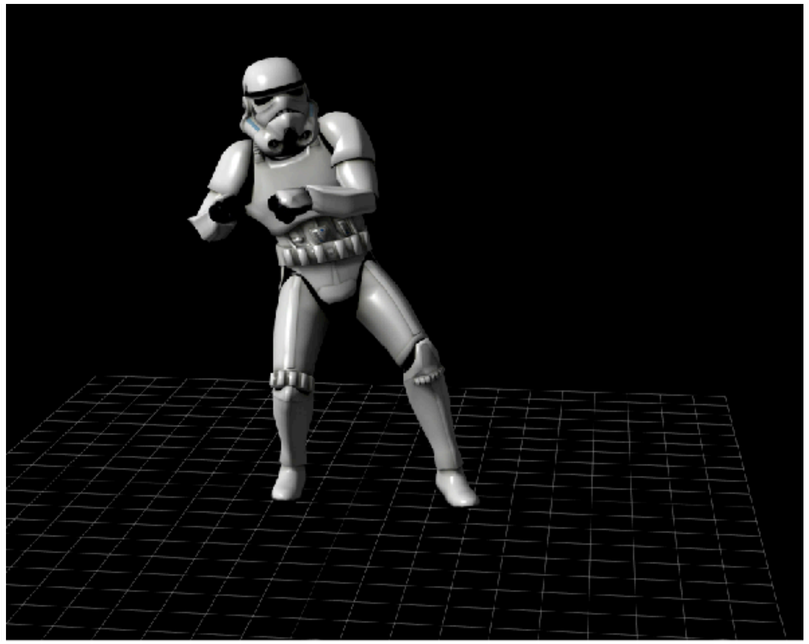


# Interactions

```
menu() {
  return (
    <View
      billboarding={'on'}
      style={styles.menu}>
      { Object.keys(planets).map((planet) => (
        <VrButton key={`button-${planet}`}
                  onClick={() => this.handleClick(planet)}>
          <View style={styles.planetBtn}>
            <Text style={styles.planetBtnLabel}>{planet}</Text>
          </View>
        </VrButton>
      )) }
    </View>
  );
}
```

# The Solar System

Earth



SS

# Models and Textures

- __tests__
- node_modules
- static_assets
  - models
    - Mars.obj
  - textures
    - Mars.png

# The Solar System

Earth

<Planet />

# Loading Models

<Planet />

```
<Model
  source={{obj: asset(`models/Earth.obj`)}}
  texture={asset(`textures/Earth.png`)}
  lit={true}
  style={styles.planet}
/>
```

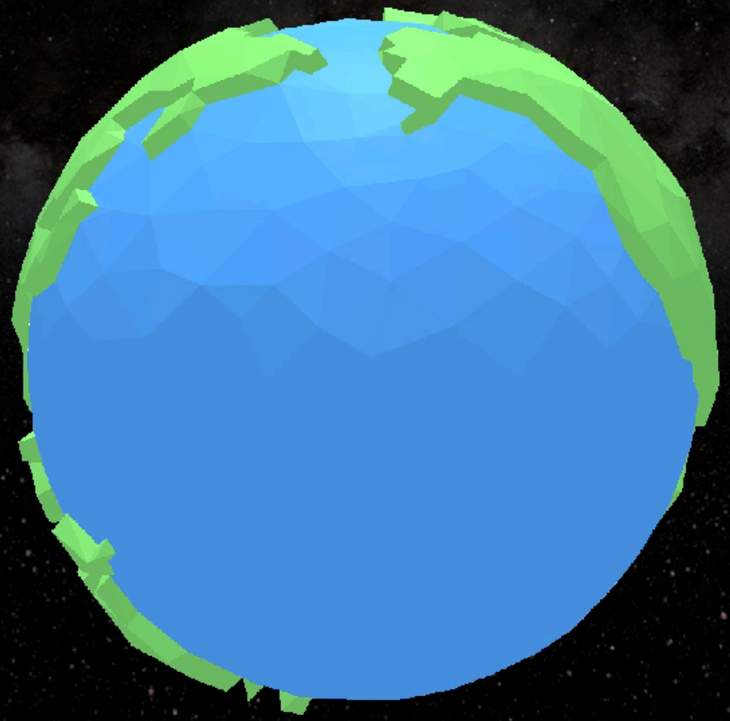
# Loading Models

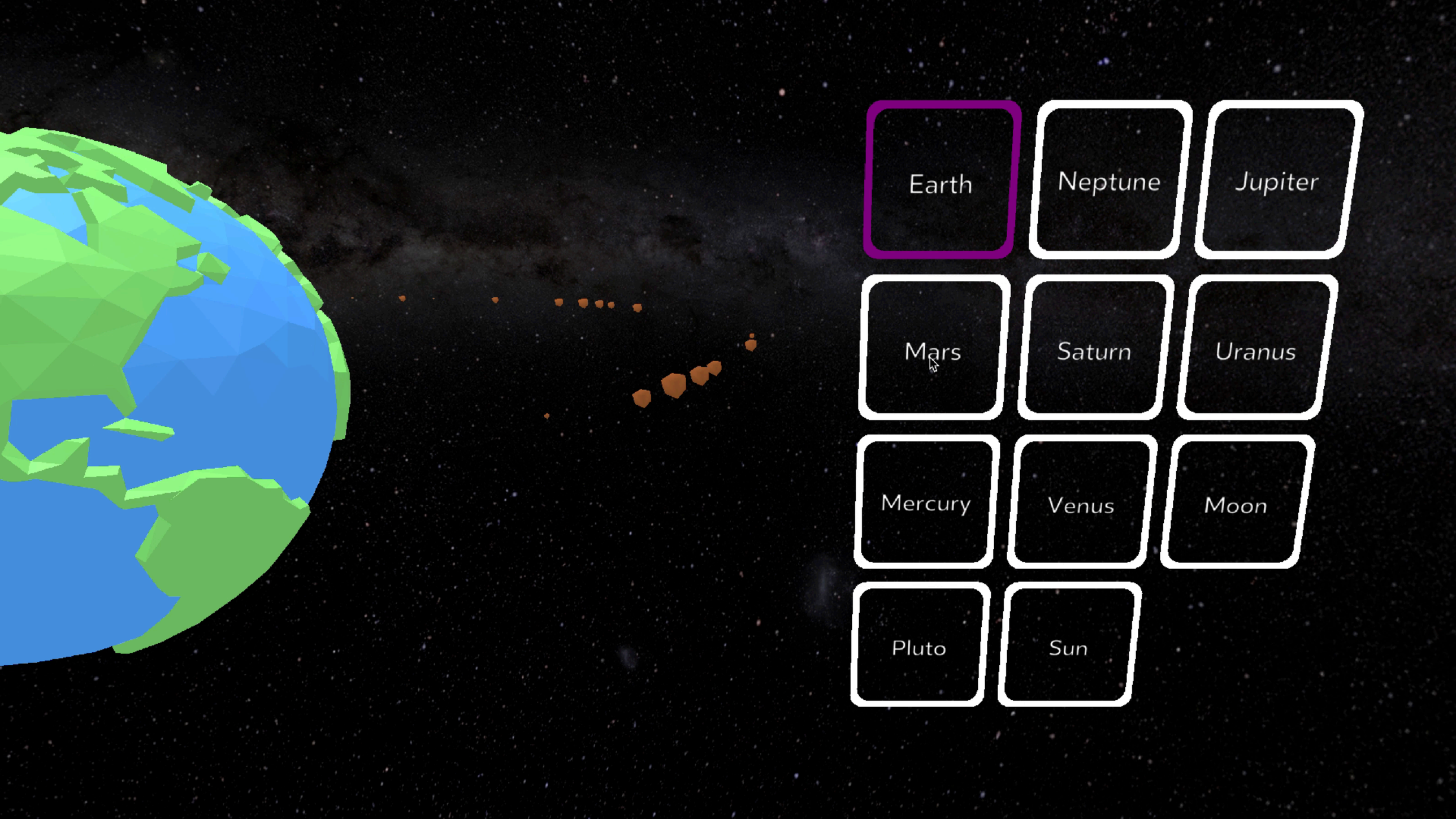
<Planet />

```
<Model
  source={{obj: asset(`models/${currentPlanet}.obj`)}}
  texture={asset(`textures/${currentPlanet}.png`)}
  lit={true}
  style={styles.planet}
/>
```

# The Solar System

Earth




Earth
Neptune
Jupiter
Mars
Saturn
Uranus
Mercury
Venus
Moon
Pluto
Sun

```
constructor() {
  super();

  this.state = {
    rotation: 0
  };
}

componentDidMount() { this.rotate(); }

// planet rotate animation
rotate() {
  const now = Date.now();
  const delta = now - this.lastUpdate;

  this.time++;

  this.lastUpdate = now;
  this.setState({
    rotation: this.state.rotation + delta / 150
  });
  this.frameHandle = requestAnimationFrame(this.rotate);
}
```

# The Solar System

Earth

```
constructor(props) {
  super();

  this.state = {
    bounceValue: new Animated.Value(1)
  };
}

// bounce animation
bounce({initial, toValue, friction = 1.5}) {
  value.setValue(initial);

  Animated.spring(
    this.state.bounceValue,
    {
      toValue,
      friction,
    }
  ).start();
}
```

# The Solar System

Earth

# Using Native Modules

# Three.js

| | | |
|---|---|---|
| **Materials** | **Cameras** | **Geometries** |
| **Loaders** | **Helpers** | **Lights** |
| **Physics** | **Renderers** | **Plugins** |

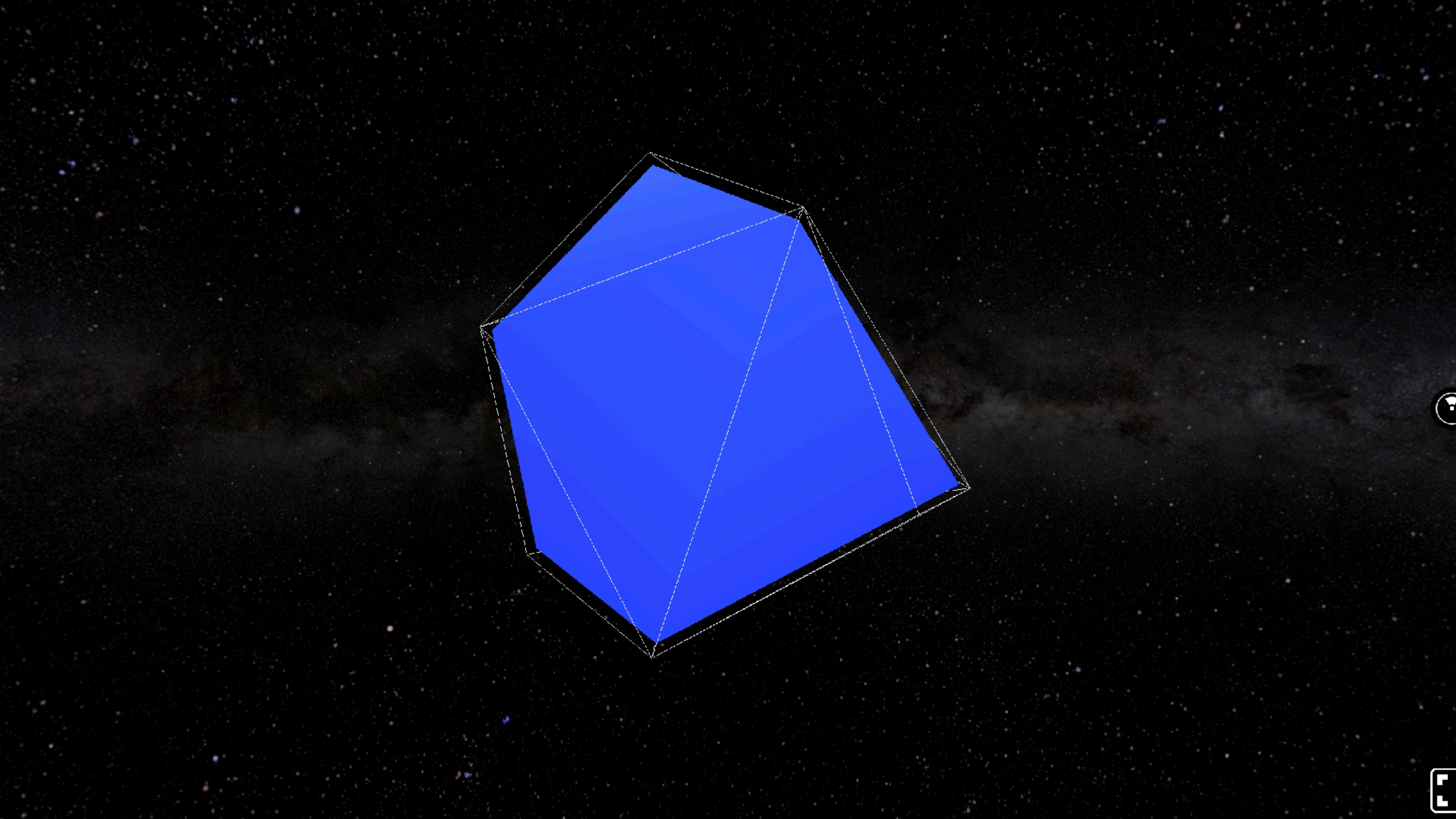
TetrahedronGeometry

# The Module

```
import {Module} from 'react-vr-web';
import * as THREE from 'three';

export default class AsteroidsModule extends Module {
  constructor(scene) {
    // The name of the module in NativeModules
    super('Asteroids');

    this.scene = scene;
  }

  add() {
    const geometry = new THREE.TetrahedronGeometry(10, 1);
    const material = new THREE.MeshLambertMaterial({color: 0x7F492A});
    const mesh = new THREE.Mesh(geometry, material);

    this.scene.add(mesh);
  }
}
```

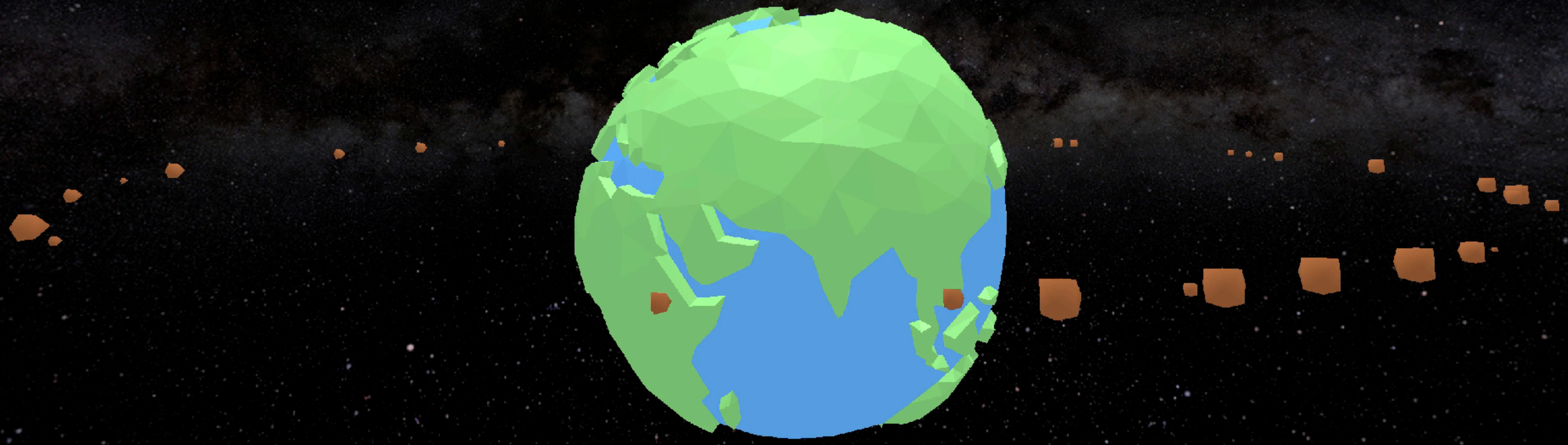
# Client.js

```
import {VRInstance} from 'react-vr-web';
import AsteroidsModule from '../components/nativeModules/asteroidsModule';
import * as THREE from 'three';

function init(bundle, parent, options) {
  const scene = new THREE.Scene();
  const Asteroids = new AsteroidsModule(scene);
  const vr = new VRInstance(bundle, 'solarSystem', parent, {
    // Add custom options here
    nativeModules: [ Asteroids ],
    scene,
    ...options,
  });
  vr.render = function() {
    // Any custom behavior you want to perform on each frame goes here
    Asteroids.render();

  };
  // Begin the animation loop
  vr.start();
  return vr;
}
```



# The Solar System

Earth



# Virtual Reality Tour

# VideoPano

```
<VideoPano source={{ uri: 'assets/my-video.webm' }} />
```

<Pano source={asset(room.jpeg)} />

<Pano source={asset(kitchen.jpeg)} />

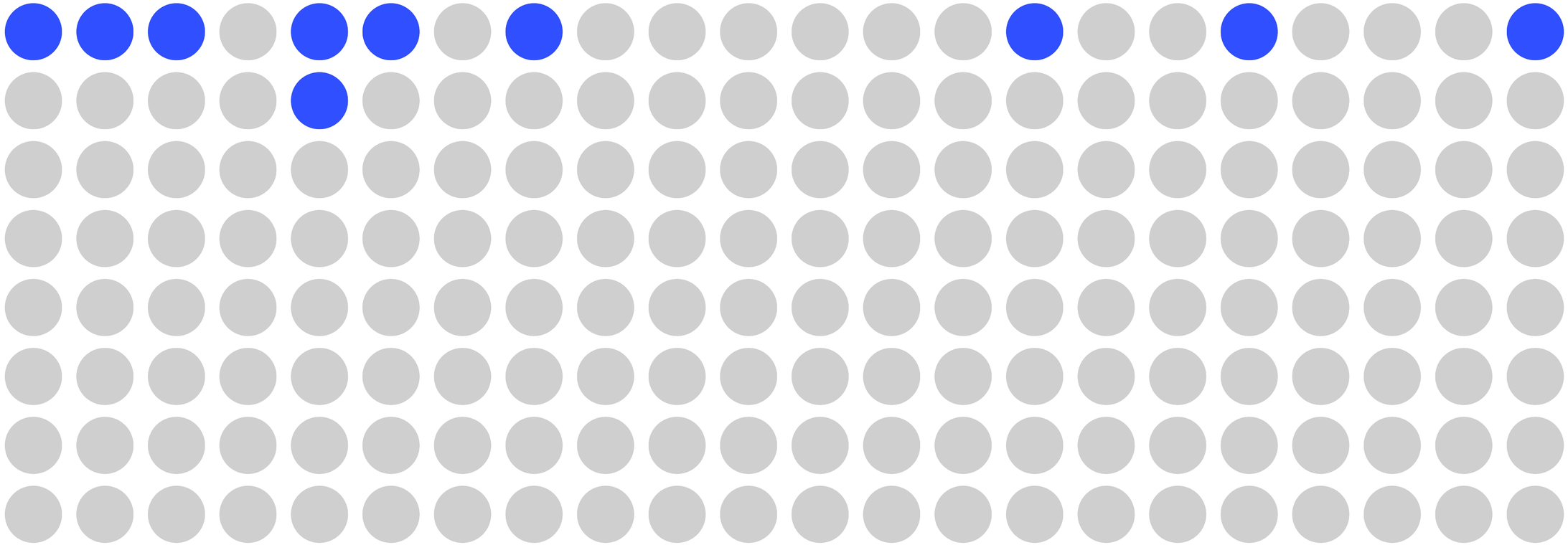
Room
Hotspot

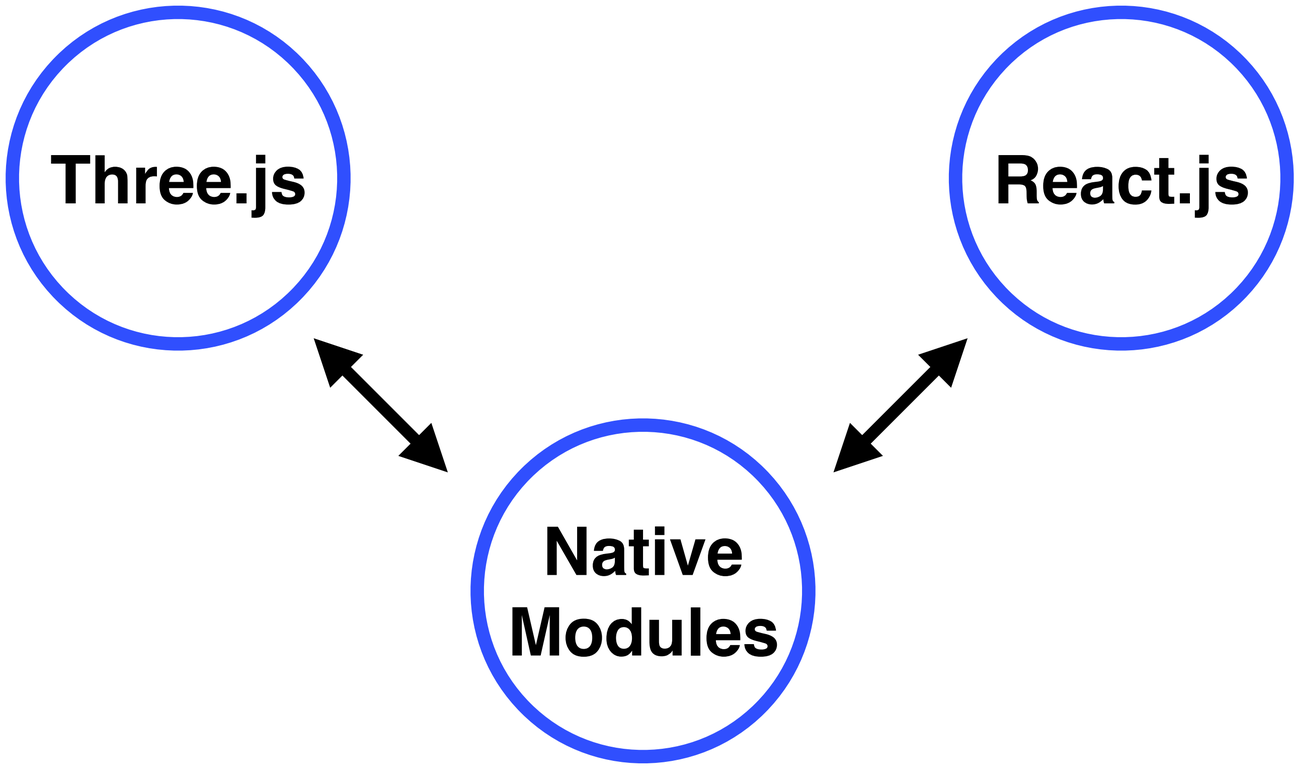
# Three.js

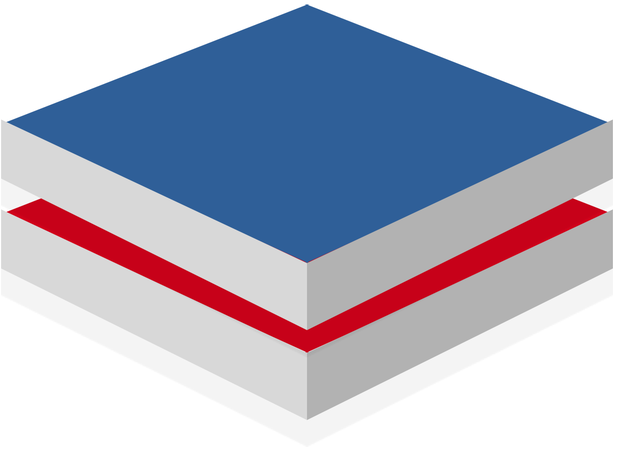
Geometries Materials Loaders Helpers Cameras Lights Controls

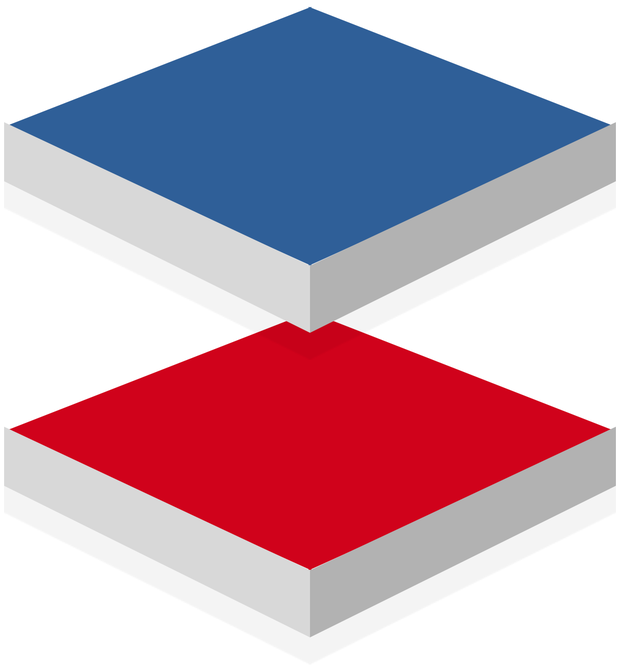
# React VR

Geometries Materials Loaders Helpers Cameras Lights Controls



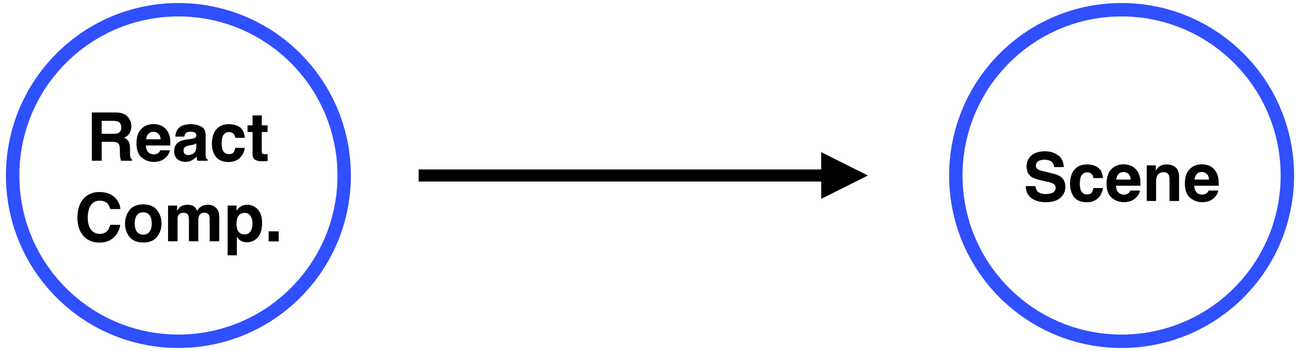

Three.js
React.js
Native
Modules





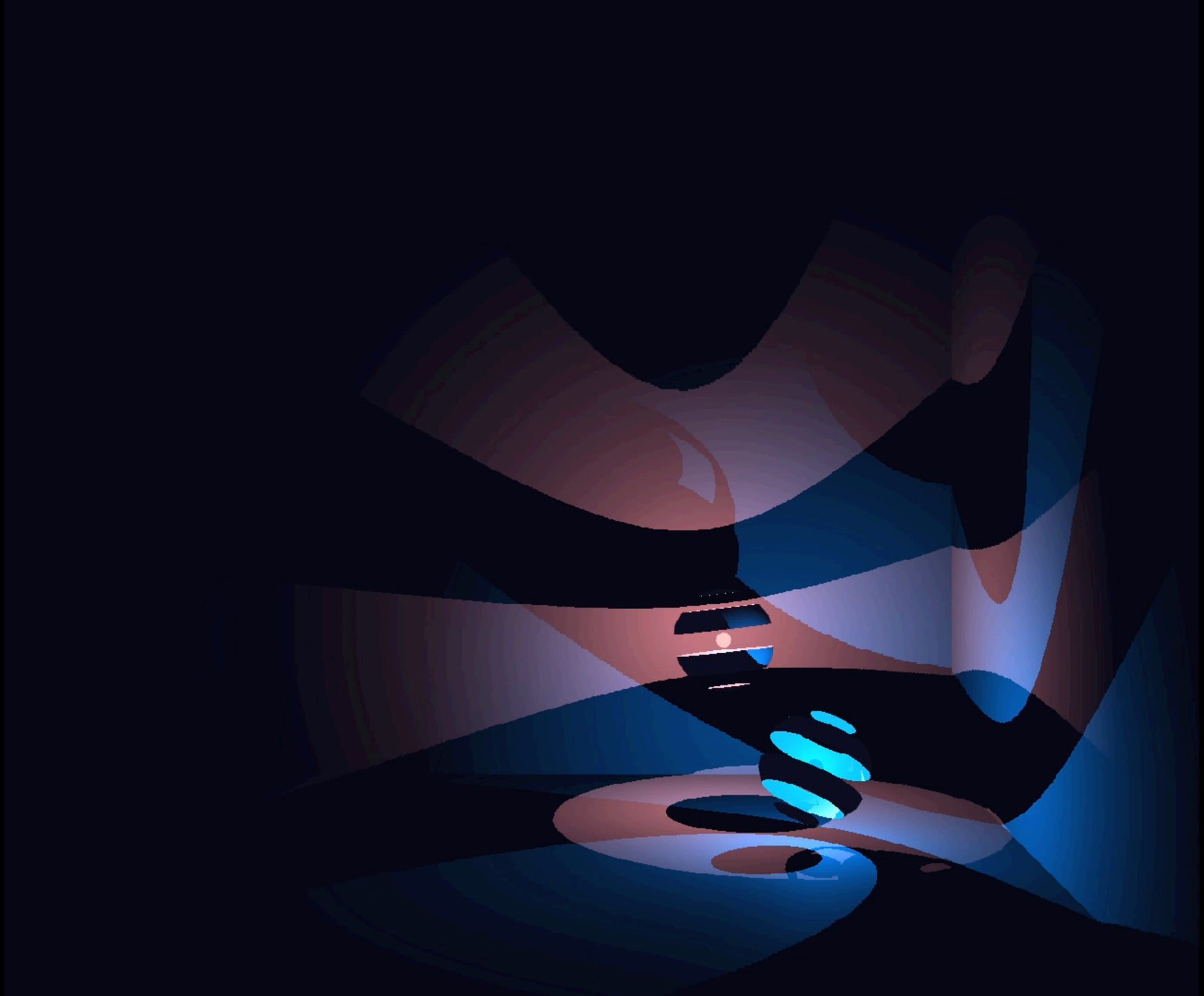
React.js

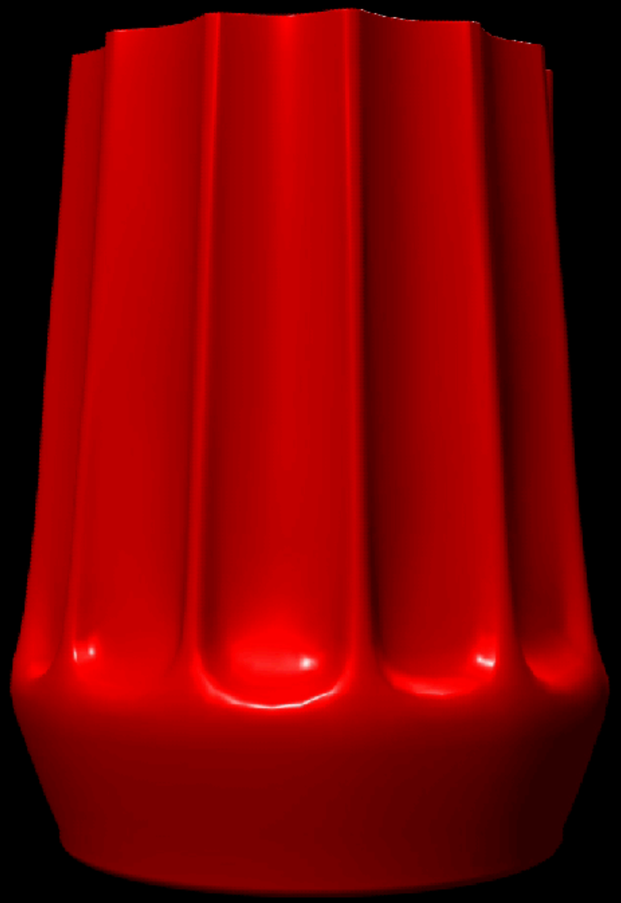
Three.js


React
Comp.
Scene

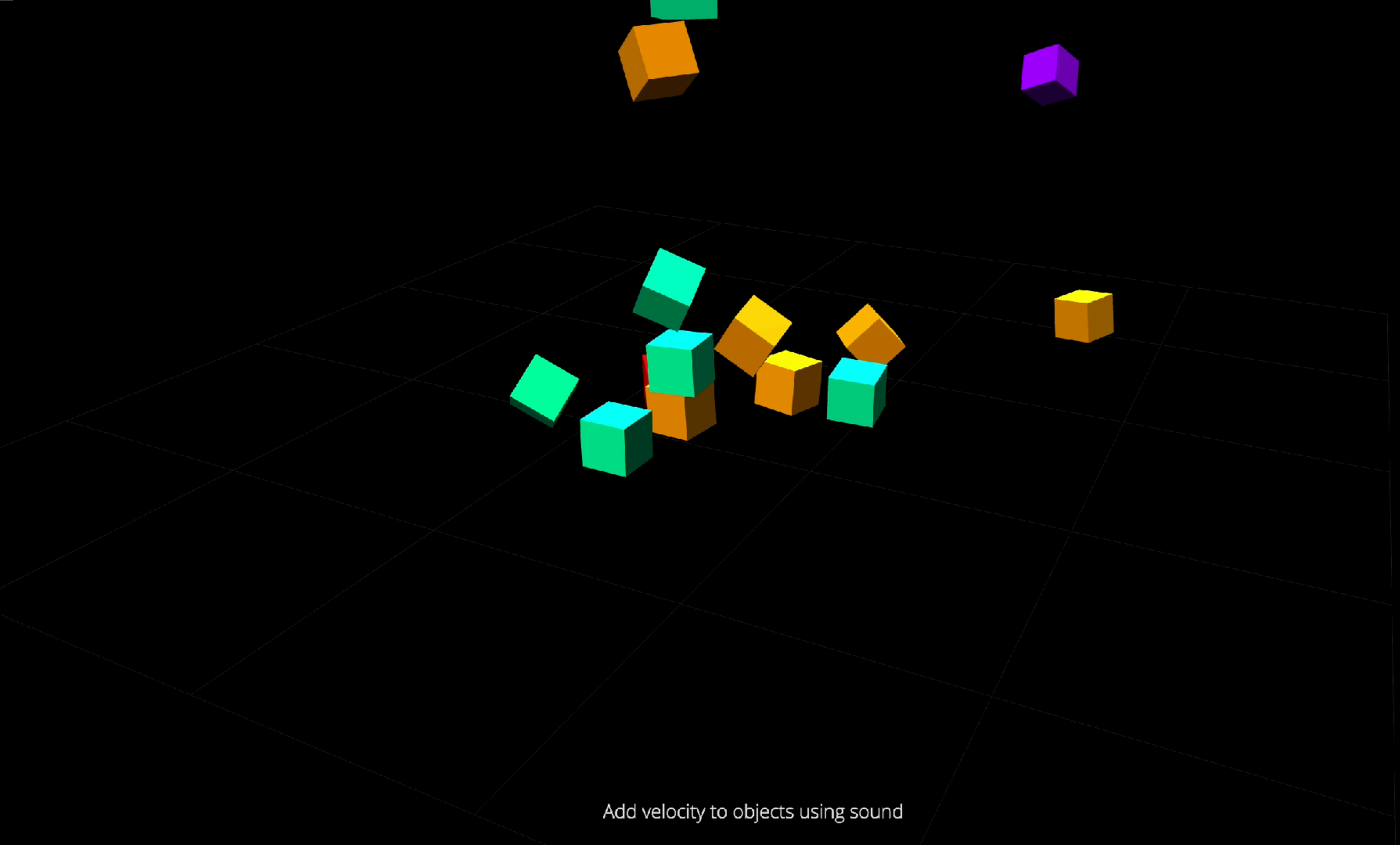
```
init(element) {
  this.scene = new THREE.Scene();

  // width & height of the placeholder
  const width = element.offsetWidth;
  const height = element.offsetHeight;

  this.camera = new THREE.PerspectiveCamera(45, width / height, 0.1, 1000);

  this.renderer = new THREE.WebGLRenderer();
  this.renderer.setSize(width, height);

  this.controls = new OrbitControls(this.camera, this.renderer.domElement);

  element.appendChild(this.renderer.domElement);

  this.animate();
}

animate = () => {
  this.controls.update();

  requestAnimationFrame(this.animate);

  this.renderer.render(this.scene, this.camera);
};
```

**bit.ly/three-vr**

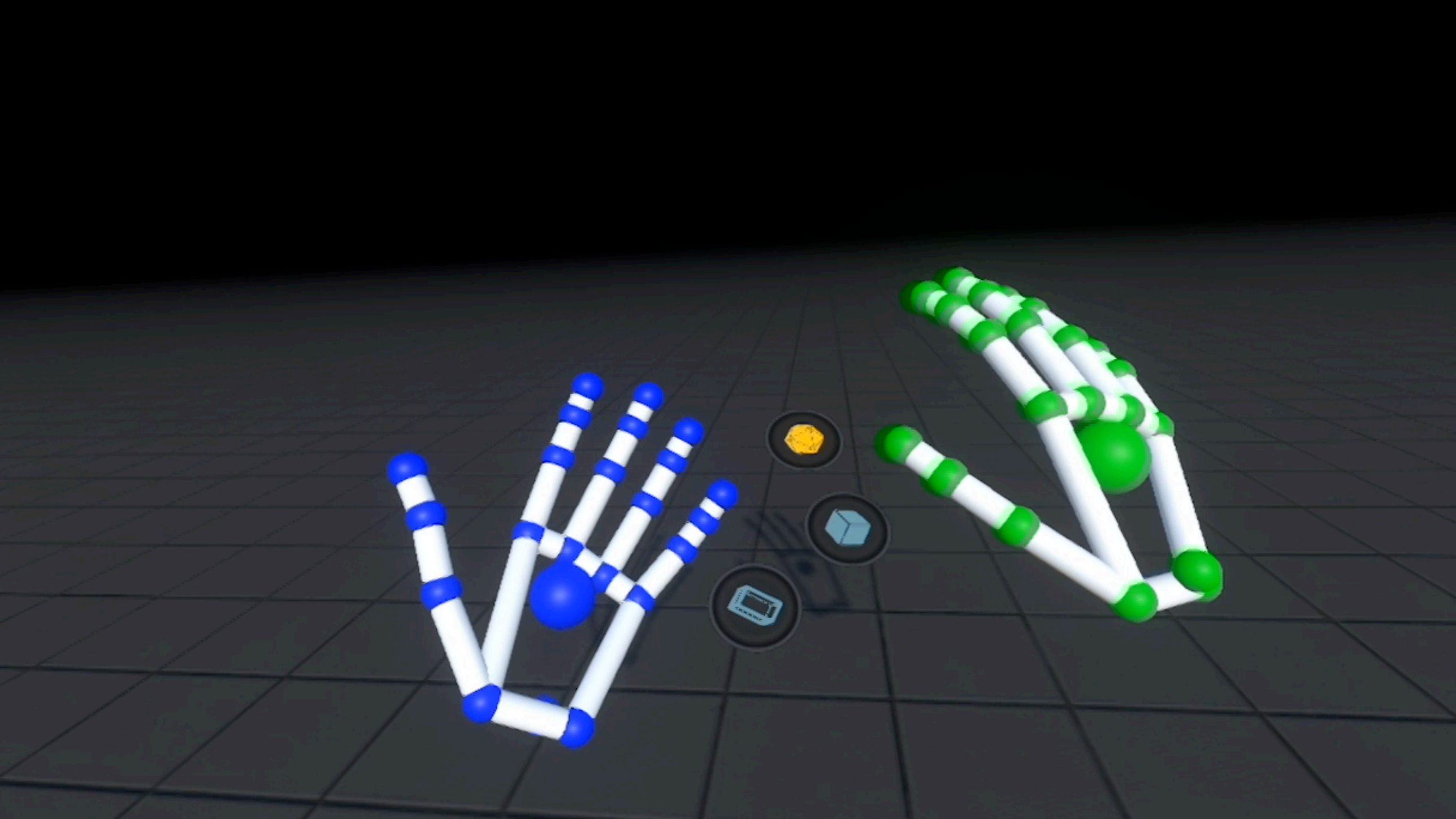




Demo of a physics engine with Three.js (cannon.js)



Add velocity to objects using sound



WebVR

```
navigator.getVRDisplays().then(displays => {
  if (displays.length) {
    this.controls = new THREE.VRControls(this.camera);
    this.controls.standing = true;
  }
});
```

```
this.effect = new THREE.VREffect(this.renderer);
this.effect.setSize(this.width, this.height);
```

**React VR**

focus on Layout

**React**
**+**
**3D Engine**

focus on the
3D scene

# Thank You!