Before we begin...

- Open up these slides:
  - https://bit.ly/2LhoIJh
- Open up the documentation:
  - THREE.js Docs
- Have a look through some examples:
  - THREE.js Examples

THREE.js

Learning Objectives

- **Understand** WebGL and requestAnimationFrame
- **Understand** X, Y and Z planes
- **Use** THREE.js effectively

Agenda

- THREE.js

Review

- Finish off React app
- Deploy the app with Now
- P5.js

Project Time!

THREE.js

Examples

What are we building?

Background

What is THREE.js?

The aim of the project is to create a lightweight 3D library with the lowest level of complexity

More or less, it is an attempt to make 3D stuff easier in browsers

Who made THREE.js?

- A guy named Ricardo Cabello (but everyone calls him Mr. Doob)
  - Website
  - Github
  - Twitter

What is it created with?

What is it built with?

- It's built on top of WebGL, a JavaScript API
- It uses a renderer to show the 3D environment
  - Renderers for Canvas, SVG, CSS3D

How do we include it?

- Just like any other JavaScript library
- We reference the Three.js script before our own (or use NPM)
- There are also a bunch of helper files that you can use too, like:
  - Orbit Controls
  - datGui

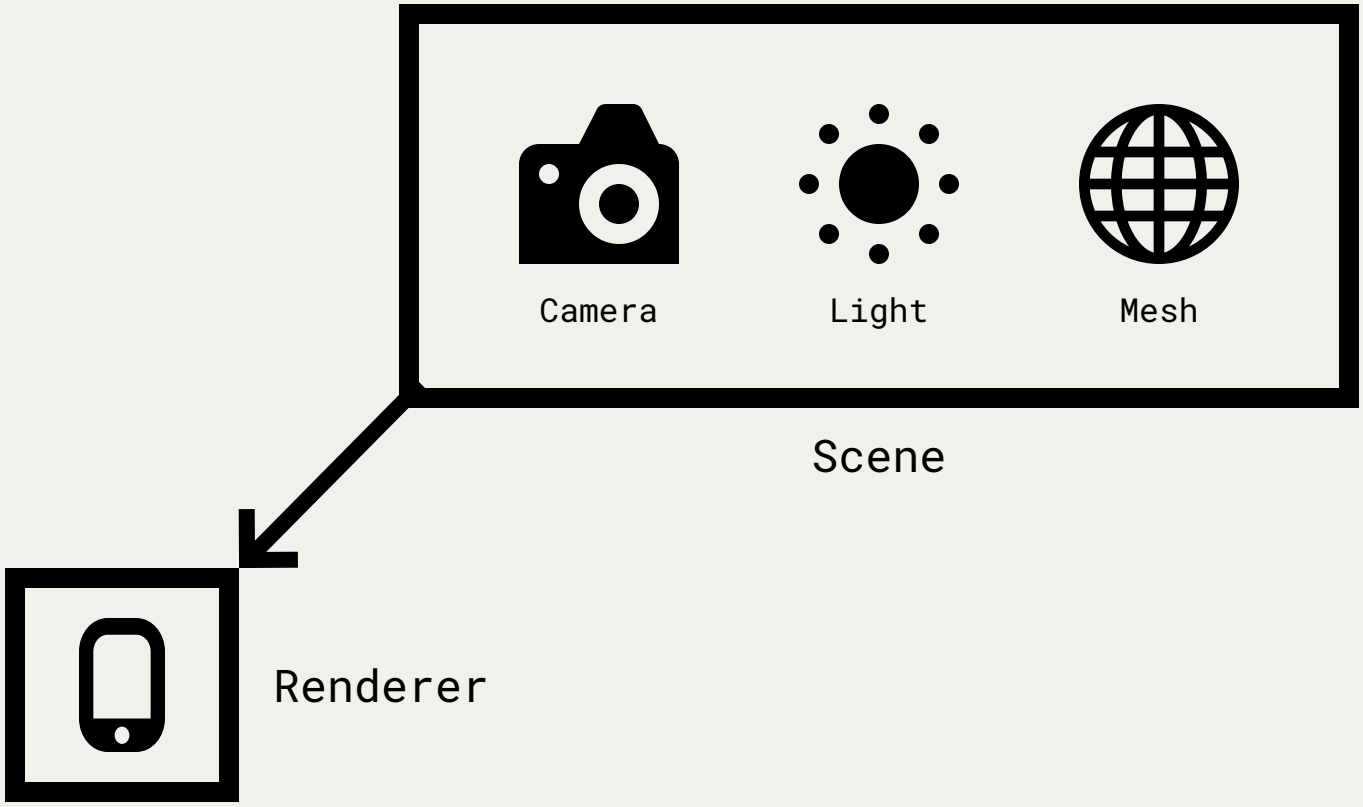
Key Components

Key Components

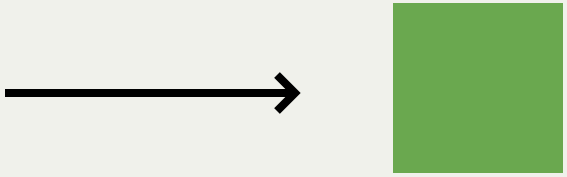
- Camera
- Scene
- Renderer
- Light(s)
- Mesh(es)

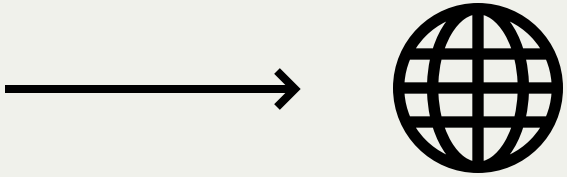
We extend the stuff that Three.js provides

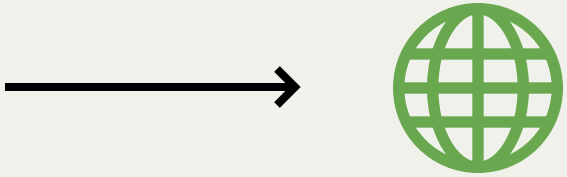
Key Components



What is a mesh?

Material 

Geometry 

Mesh 

What will we be using?

What will we be using?

- Three.js
- OrbitControls.js ?
- Stats.js ?
- datGui.js ?

requestAnimationFrame

In order to show changes in a Three.js scene, we need to re-render it. **setInterval** used to be the way we do that, but now we use **requestAnimationFrame**. Why?

- It runs as quickly as possible (higher FPS)
- It stops when the tab or window is no longer active (better battery life)
- It times it correctly for optimum performance
- Is often hardware-boosted

Coding time!

Scene & Camera

```
const scene = new THREE.Scene();

let width = window.innerWidth;
let height = window.innerHeight;

let camera = new THREE.PerspectiveCamera(45, width / height, 0.1, 1

camera.position.x = -30;
camera.position.y = 40;
camera.position.z = 30;

camera.lookAt(scene.position);
```

Renderer

```
let renderer = new THREE.WebGLRenderer();
renderer.setClearColor("#ECEFF1");
renderer.setSize(window.innerWidth, window.innerHeig

document.querySelector("#output")
        .appendChild(renderer.domElement);

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

Our first shape

```
let cubeGeometry = new THREE.BoxGeometry(4, 4, 4);
let cubeMaterial = new THREE.MeshBasicMaterial({
  color: "#FF8F00",
  wireframe: true
});

let cube = new THREE.Mesh(cubeGeometry, cubeMateria
```

Homework

- It's Project Time!
- Read up on ES2015
  - Translate some of your previous code into it!
- Finish all exercises from class
- Upload your homework to GitHub
- Prepare for next lesson

Homework (Extra)

- Go through some tasks in Exercism
- Get into JavaScript30
- Go through The Modern JavaScript Tutorial
- Read Exploring ES6
- Read Eloquent JavaScript
- Read Speaking JavaScript

What's next?

- Project Presentations!

Questions?

Feedback time!

Lesson 18: ***THREE.js***

https://ga.co/js05syd

Thanks!