

Welcome!

Agenda

- THREE.js
- Hosting with Vercel
- Wrapping Up
- Surveys

THREE.js

Documentation

Examples

What is THREE.js?

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 fellow named Ricardo Cabello (but everyone calls him Mr. Doob). Here is his:

- [Website](#)
- [GitHub](#)
- [Twitter](#)

What is it built with?

It's built on top of WebGL, which is a JavaScript API

It uses a renderer to show the 3D environment

How do we install it?

There are many ways to install it, but we will be using NPM (as well as Parcel):



```
npm install --save three
```

We will also be using a few addons. Orbit Controls will allow us to change the perspective, and Dat GUI will allow us to change the speed of animations:



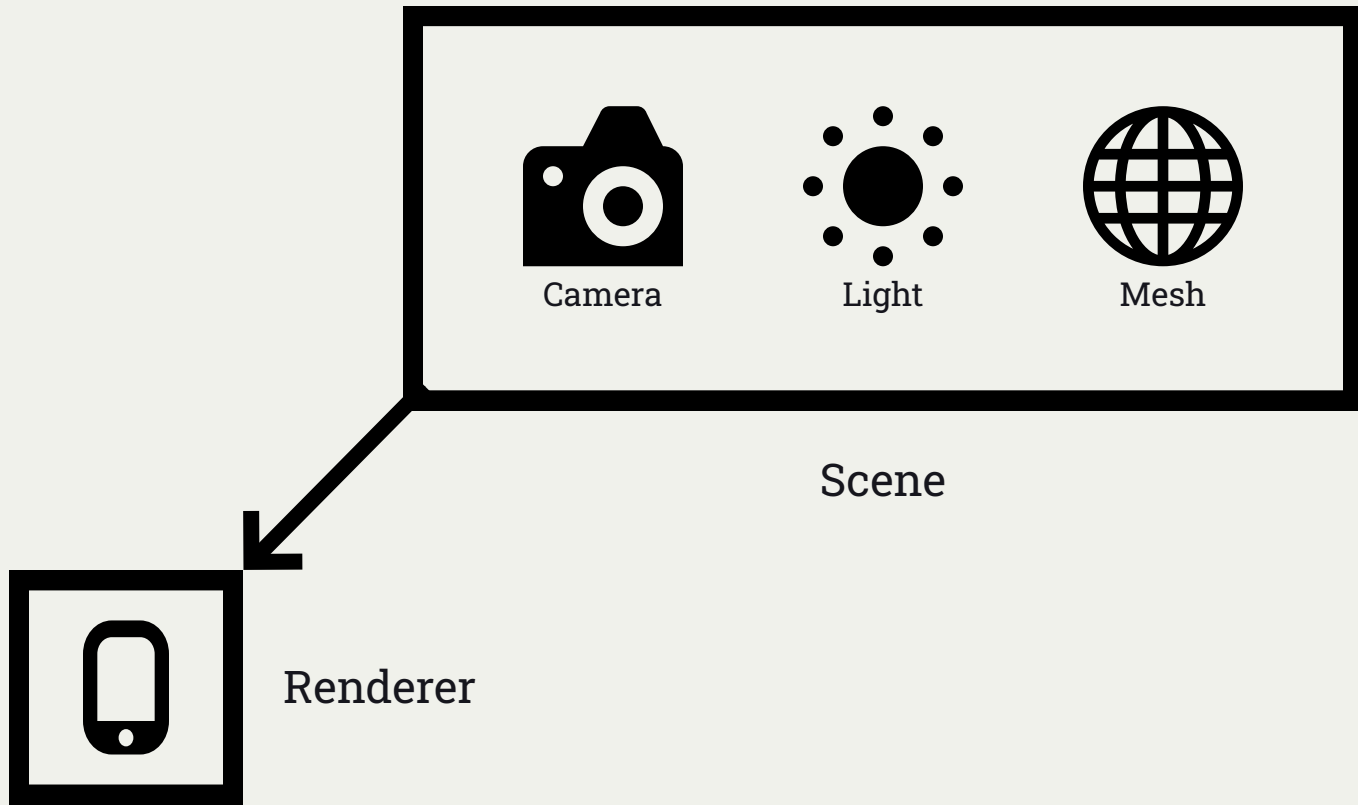
```
npm install --save three-orbitcontrols dat.gui
```

Key Components

Key Components

We extend things that THREE.js provides to create:

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



What is a mesh?

Material



What is a mesh?

Material



Geometry



What is a mesh?

Material



Geometry



Mesh



What will we be using?

- THREE
- OrbitControls
- dat.gui

Renderer



```
function createRenderer() {  
  let renderer = new THREE.WebGLRenderer({  
    antialias: true,  
  });  
  
  renderer.setSize(window.innerWidth, window.innerHeight);  
  renderer.setClearColor("#16161d");  
  renderer.setPixelRatio(window.devicePixelRatio);  
  
  let output = document.querySelector("#output");  
  output.appendChild(renderer.domElement);  
  
  return renderer;  
}
```

Scene



```
function createScene() {  
    return new THREE.Scene();  
}
```

Camera



```
function createCamera() {  
  let camera = new THREE.PerspectiveCamera(  
    45,  
    window.innerWidth / window.innerHeight,  
    0.1,  
    1000,  
  );  
  camera.position.x = -30;  
  camera.position.y = 40;  
  camera.position.z = 30;  
  camera.lookAt(0, 0, 0);  
  return camera;  
}
```

AxesHelper



```
function createAxesHelper() {  
  let axesHelper = new THREE.AxesHelper(40);  
  return axesHelper;  
}
```

Cube



```
function createCube() {  
  let geometry = new THREE.BoxGeometry(4, 4, 4);  
  let material = new THREE.MeshLambertMaterial({  
    color: "dodgerblue",  
  });  
  let mesh = new THREE.Mesh(geometry, material);  
  return mesh;  
}
```

Add everything to the Scene



```
1 let renderer = createRenderer();
2 let scene = createScene();
3 let camera = createCamera();
4 let axesHelper = createAxesHelper();
5 let cube = createCube();
6
7 scene.add(axesHelper, cube);
8 renderer.render(scene, camera);
```

Add everything to the Scene



```
1 let renderer = createRenderer();  
2 let scene = createScene();  
3 let camera = createCamera();  
4 let axesHelper = createAxesHelper();  
5 let cube = createCube();  
6  
7 scene.add(axesHelper, cube);  
8 renderer.render(scene, camera);
```

Add everything to the Scene



```
1 let renderer = createRenderer();  
2 let scene = createScene();  
3 let camera = createCamera();  
4 let axesHelper = createAxesHelper();  
5 let cube = createCube();  
6  
7 scene.add(axesHelper, cube);  
8 renderer.render(scene, camera);
```


Light



```
function createLight() {  
  let light = new THREE.PointLight("#ffffff", 2);  
  light.position.set(10, 18, 10);  
  return light;  
}
```

LightHelper



```
function createLightHelper(light) {  
    return new THREE.PointLightHelper(light);  
}
```

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 we will be using `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 the calls for optimum performance
- It is often hardware-booster

requestAnimationFrame



```
function animate() {  
  // Make sure to render your THREE.js scene here!  
  requestAnimationFrame(animate);  
}
```

Hosting with Vercel

Remove main and add build



```
1 {
2   "name": "threejs-demo",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "build": "parcel build src/index.html",
8     "start": "parcel src/index.html"
9   },
10  "author": "",
11  "license": "ISC",
12  "devDependencies": {
13    "parcel": "^2.0.0-rc.0"
14  },
15  "dependencies": {
16    "dat.gui": "^0.7.7",
17    "three": "^0.132.2",
18    "three-orbit-controls": "^82.1.0"
19  }
20 }
```

Remove main and add build



```
1 {
2   "name": "threejs-demo",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "build": "parcel build src/index.html",
8     "start": "parcel src/index.html"
9   },
10  "author": "",
11  "license": "ISC",
12  "devDependencies": {
13    "parcel": "^2.0.0-rc.0"
14  },
15  "dependencies": {
16    "dat.gui": "^0.7.7",
17    "three": "^0.132.2",
18    "three-orbit-controls": "^82.1.0"
19  }
20 }
```

Steps for Deploying

- Upload all of your code to GitHub
- Sign up for Vercel using your GitHub account
- Import the repository to Vercel
- Customize your build settings:
 - First box: npm run build
 - Second box: dist
 - Third box: npm install

What's Next?

Moving Forward

- The projects will be due on the 13th of September
- I'll post my availability so that we can catch up and I can help you out
 - I'll also still be available to message

It's all over*

But not really!

Quick Rewind

How did we design this course?

It was all about:

Quick Rewind

How did we design this course?

It was all about:

- Skills that will last

Quick Rewind

How did we design this course?

It was all about:

- Skills that will last
- New approaches to learning

Quick Rewind

How did we design this course?

It was all about:

- Skills that will last
- New approaches to learning
- Preparing you for jobs

Quick Rewind

How did we design this course?

It was all about:

- Skills that will last
- New approaches to learning
- Preparing you for jobs
- Putting you in a room with driven and intelligent people

Progress over Perfection

Another Quick Rewind

What did we cover on the first night?

- What JavaScript is and does
- Variables
- Data Types

Think about how far you have come in just 60 hours.

Maybe, in this course, you have doubted yourself?

You all know enough!

- You will be useful JavaScript developers in the workplace

You all know enough!

- You will be useful JavaScript developers in the workplace
- BUT:

You all know enough!

- You will be useful JavaScript developers in the workplace
- BUT:
 - You do have so much more to learn

You all know enough!

- You will be useful JavaScript developers in the workplace
- BUT:
 - You do have so much more to learn
 - Forever...

You all know enough!

- You will be useful JavaScript developers in the workplace
- BUT:
 - You do have so much more to learn
 - Forever...
 - We all do

You all know enough!

- You will be useful JavaScript developers in the workplace
- BUT:
 - You do have so much more to learn
 - Forever...
 - We all do
 - That is a great thing

James Clear:

"If you get one percent better each day for one year, you'll end up 37 times better by the time you're done"

Most important things we taught

How to think like a programmer

- The concepts are, by far, the most important thing

Most important things we taught

How to debug

- Everything comes with practice, but practice is easier when you can debug:
 - Well
 - and patiently,
 - and calmly,
 - and cheerfully (enjoy those little wins!)
- Fixing other people's code is incredibly rewarding

What now?

You (kind of) never have to see me again!

What now?

You (kind of) never have to see me again!

- Don't stop coding: practice

What now?

You (kind of) never have to see me again!

- Don't stop coding: practice
- Learning new languages and frameworks makes you better at other ones

What now?

You (kind of) never have to see me again!

- Don't stop coding: practice
- Learning new languages and frameworks makes you better at other ones
- Participate! Meetups, conferences, open source, blogs

What now?

You (kind of) never have to see me again!

- Don't stop coding: practice
- Learning new languages and frameworks makes you better at other ones
- Participate! Meetups, conferences, open source, blogs
- Stay in touch with each other

What now?

You (kind of) never have to see me again!

- Don't stop coding: practice
- Learning new languages and frameworks makes you better at other ones
- Participate! Meetups, conferences, open source, blogs
- Stay in touch with each other
- Enjoy: it's going to be easier that way

What now?

You (kind of) never have to see me again!

- Don't stop coding: practice
- Learning new languages and frameworks makes you better at other ones
- Participate! Meetups, conferences, open source, blogs
- Stay in touch with each other
- Enjoy: it's going to be easier that way
- **Get great: blow my mind!**

Practice

- Exercism
- Project Euler
- Rosetta Code
- CodeWars
- CoderByte
- Hacker Rank
- Code Signal
- Leet Code
- CodeKata

Passive Consumption

- Twitter
- Newsletters
- GitHub
- Web Inspiration
- Podcasts

Go to

- Meetups
- Conferences (many are remote and many post videos of talks)
- Hackathons
- Digital Meetups
- Join Slack channels and Discords

Books

- Eloquent JavaScript
- Speaking JavaScript
- JavaScript: The Definitive Guide
- JavaScript: The Good Parts
- The You Don't Know JS Series
- Understanding ES6
- Exploring ES6
- Functional Light
- Essential JS Design Patterns

Getting Great at Programming

- Keep Learning
- Read books about code and other people's code
- Look for parallels between everything
- Have a plan (such as [this roadmap](#)).
- Teach and help others with their code
- Don't get too caught up in the code, as well
 - Focus on "real life" and the "user"
- Don't be afraid to take a break
- Focus on the language itself
- Don't forget to enjoy it!

Getting Great at Programming

- Watch videos about coding
 - FrontEnd Masters
 - Egghead
 - Dash
 - Codecademy.
 - Plural Sight
 - Udacity.
 - Udemy.

For all of you

For all of you

- You have time and skills

For all of you

- You have time and skills
- Keep pushing

For all of you

- You have time and skills
- Keep pushing
- Keep celebrating the little wins

For all of you

- You have time and skills
- Keep pushing
- Keep celebrating the little wins
- Remember it is a marathon, not a sprint

For all of you

- You have time and skills
- Keep pushing
- Keep celebrating the little wins
- Remember it is a marathon, not a sprint
- Figure out what suits you

For all of you

- You have time and skills
- Keep pushing
- Keep celebrating the little wins
- Remember it is a marathon, not a sprint
- Figure out what suits you
- Be proud - you have come an exceptionally long way in a short time

Now, a little bit about me

Why do I teach?

- It's the most rewarding thing I can ever imagine
- Every day is inspiring
 - The effort
 - The projects
 - The people

Please, please, please stay in touch!

jack.jeffress@generalassembly.ly

[Instagram](#), [Twitter](#)

Thank You!

Thank You!

Really, for everything

Survey Time

One last bit of feedback

That is all!