CS 641, Haik Sahakian

Mobile Web Development

Three.JS

Three.JS

Three.JS

- *Crowd Simulator.* By Branislav Ulicny, one of the writers of three.js. Real-time 3D graphics recorded as a web video.
  https://www.youtube.com/watch?v=5T9j9d9H7co

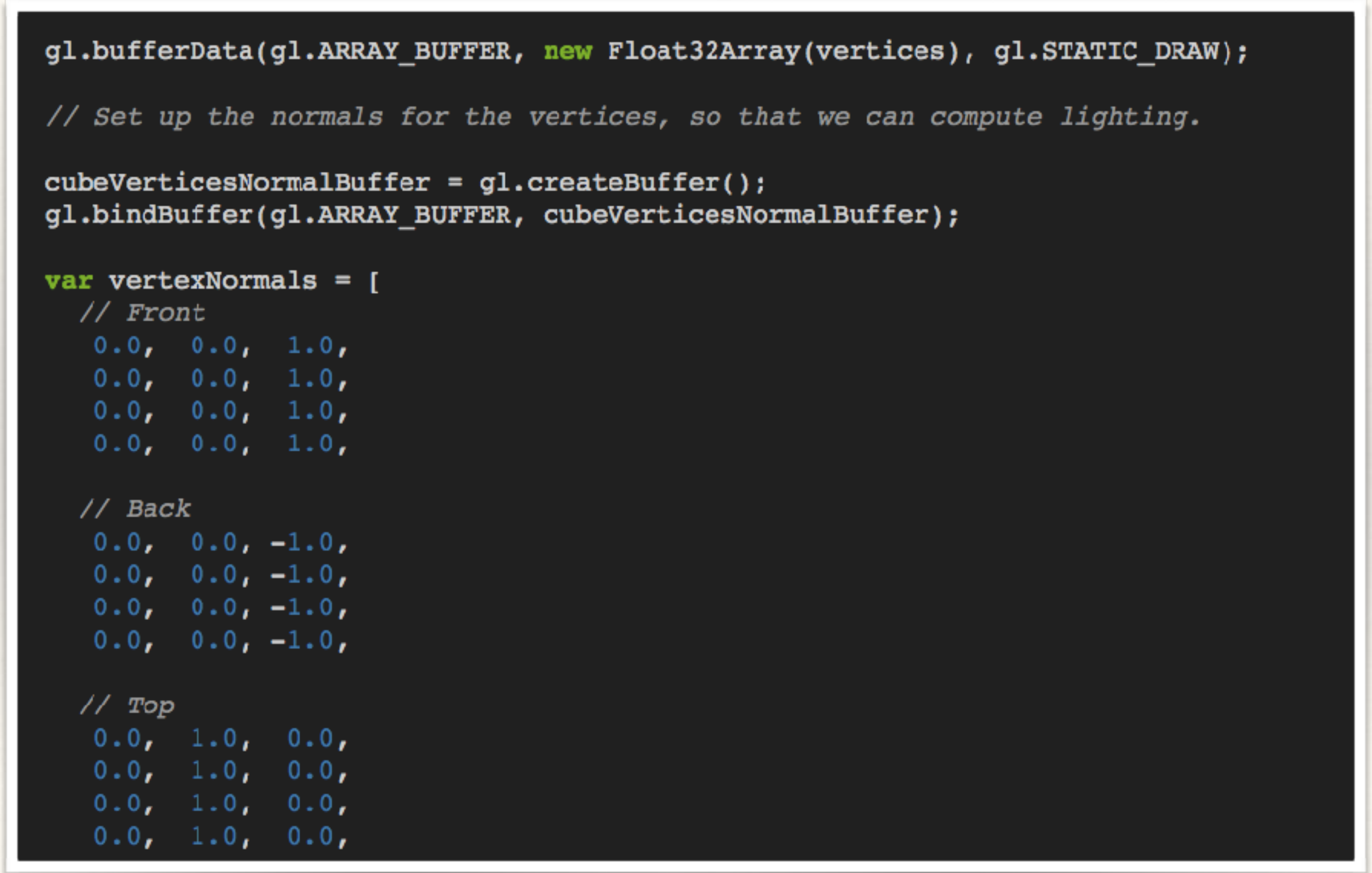
What Is It?

- A JavaScript library to make 3D graphics easier.
- Has many renderers. Can use WebGL, 3D CSS, or other renderers. WebGL is the key one.
- Just one library JS file to include.

WebGL

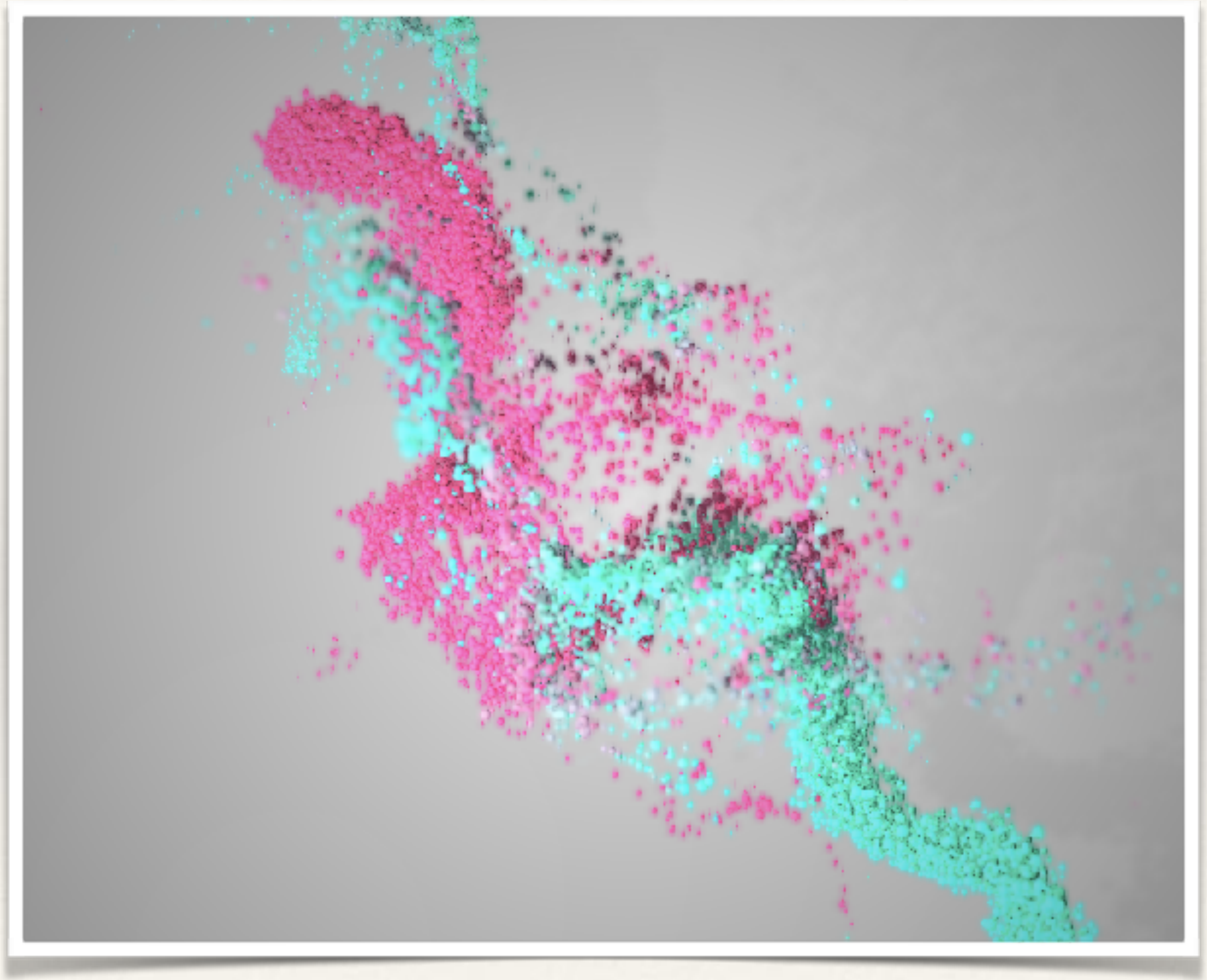
- A slimmed down version of OpenGL for the Web.
- Uses the canvas tag.
- Rather than calling getContext(“2d”), you use getContext(“webgl”).
- A completely new API. Low-level and verbose.
- Well suited to a library that provides an easier to use, higher-level, API.

WebGL Code Example



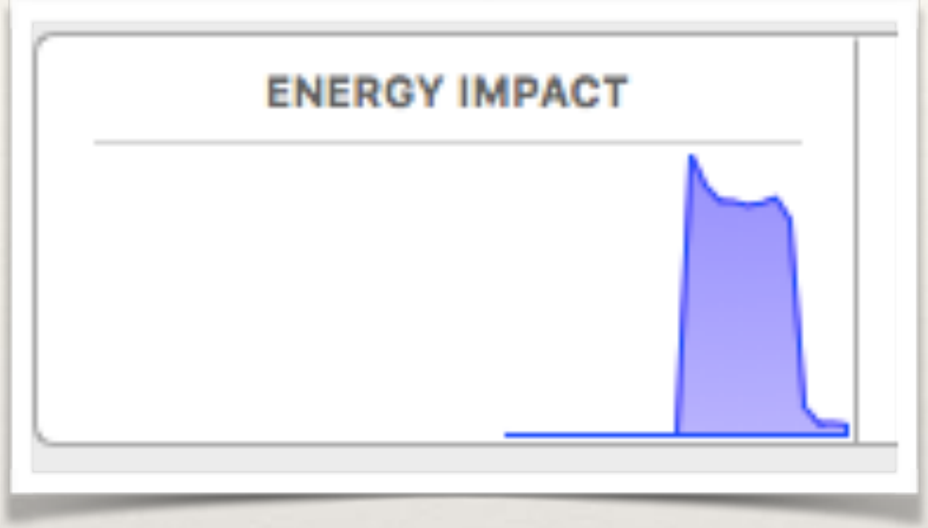
```
gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(vertices), gl.STATIC_DRAW);

// Set up the normals for the vertices, so that we can compute lighting.

cubeVerticesNormalBuffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, cubeVerticesNormalBuffer);

var vertexNormals = [
  // Front
   0.0,  0.0,  1.0,
   0.0,  0.0,  1.0,
   0.0,  0.0,  1.0,
   0.0,  0.0,  1.0,

  // Back
   0.0,  0.0, -1.0,
   0.0,  0.0, -1.0,
   0.0,  0.0, -1.0,
   0.0,  0.0, -1.0,

  // Top
   0.0,  1.0,  0.0,
   0.0,  1.0,  0.0,
   0.0,  1.0,  0.0,
   0.0,  1.0,  0.0,
```

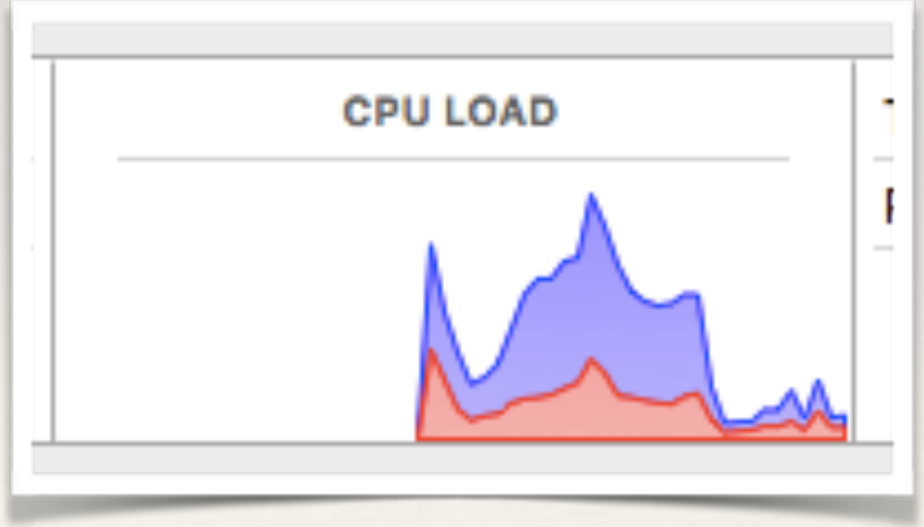
WebGL Examples



https://www.chromeexperiments.com/webgl

WebGL Examples





WebGL is one of the few things on the web that easily maxes out the CPU.

To stay fast, keep the number of objects small, and their materials simple.

Remember phones are at least 5 times slower than desktops.

Three.JS Basics

The Minimum You Need

- A Scene Object
- A Camera Object
- A Mesh Object
- A Material Object
- A Geometry Object
- A Renderer Object
- A call to requestAnimationFrame() to draw it all

Start with a Blank HTML File

```
<html>
    <body>
        <script src="three.min.js"></script>
        <script>

        // Your code goes here

        </script>
    </body>
</html>
```

Creating a Scene (1 of 3)

- First, create a Scene object.
- To create an object in Three.JS, we must define its shape and what its skin looks like.
- To do this we create a Mesh object, and attach a Geometry and a Material object to it.

Creating a Scene (1 of 3)

```
var mesh, camera, scene, renderer;

camera = new THREE.PerspectiveCamera( 50,
window.innerWidth / window.innerHeight, 1, 100 );

camera.position.z = 100;

scene = new THREE.Scene();

var geometry = new THREE.BoxGeometry( 33,33,33 );

var material = new THREE.MeshBasicMaterial( { color:
0x447744 } );
```

Creating a Scene (2 of 3)

- Then we add the Mesh object to our Scene.
- There are 4 cameras, 6 kinds of light object, 13 kinds of material, and so on. You choose one, customize it, and add it to your object or scene.

Creating a Scene (2 of 3)

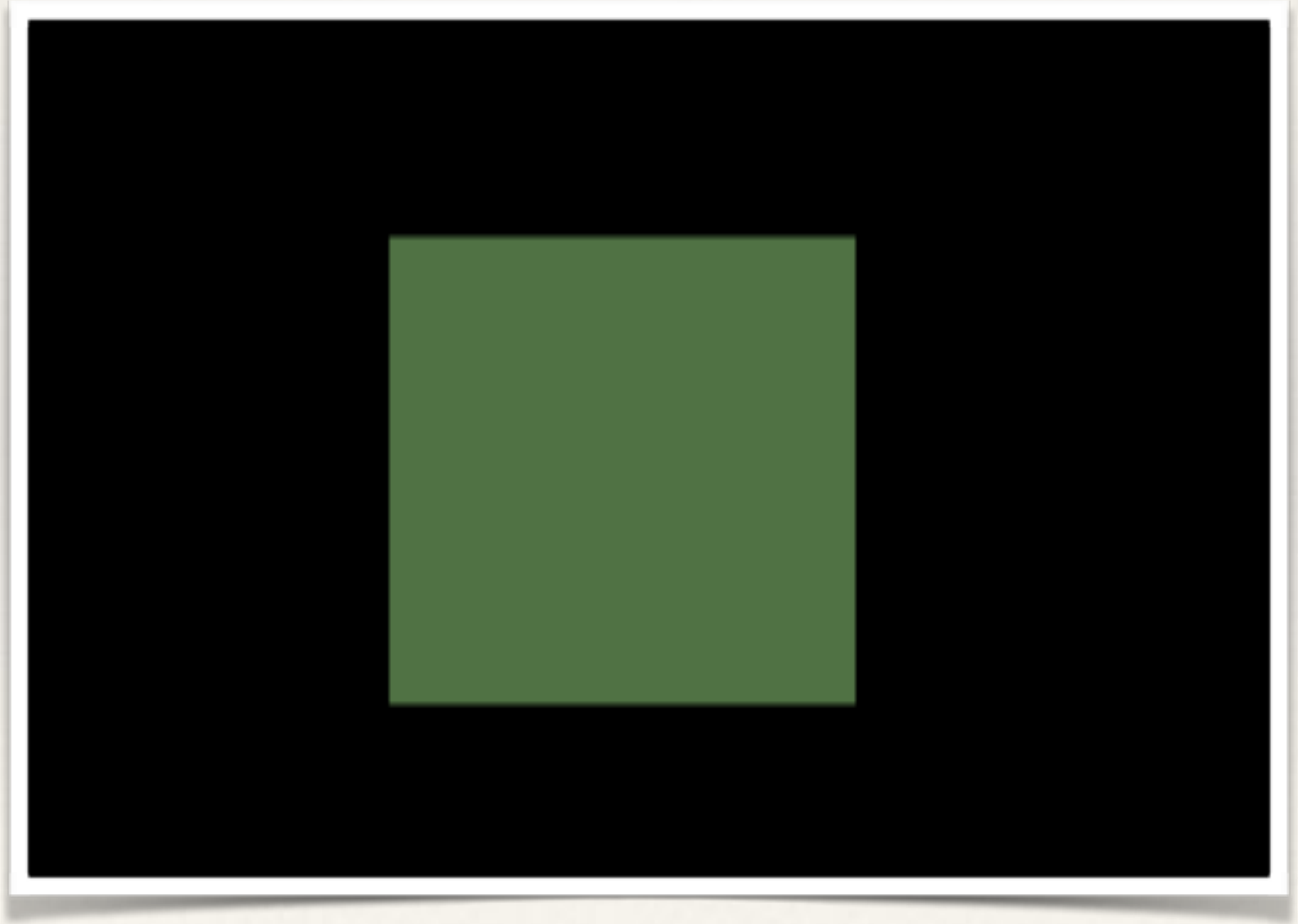
```
var mesh, camera, scene, renderer;

camera = new THREE.PerspectiveCamera( 50,
window.innerWidth / window.innerHeight, 1, 100 );

camera.position.z = 100;

scene = new THREE.Scene();

var geometry = new THREE.BoxGeometry( 33,33,33 );

var material = new
THREE.MeshBasicMaterial( { color: 0x447744 } );

mesh = new THREE.Mesh(geometry, material);

scene.add(mesh);
```

Creating a Scene (3 of 3)

```
renderer = new THREE.WebGLRenderer( { antialias:
true } );

renderer.setSize( window.innerWidth,
window.innerHeight );

document.body.appendChild( renderer.domElement );

animate();


function animate() {

   requestAnimationFrame( animate );

   renderer.render( scene, camera );

}
```

That’s It

- Test your 3D scene in your browser.



Textures

Textures

- Let’s add an image to the cube as a texture.

Textures

```
var geometry = new
THREE.BoxGeometry( 33,33,33 );

var texture = THREE.ImageUtils.loadTexture(
"./betteroffdead-512.jpg" );

var material = new THREE.MeshBasicMaterial(
{ map: texture } );

mesh = new THREE.Mesh(geometry, material);

scene.add(mesh);
```

Textures

Rotation

Rotation

- To be *sure* that's a cube, let's rotate it.
- Now that we've set up the 3D environment, this is easy.

-
  ```
  mesh.rotation.x = 150;
  ```

Rotation

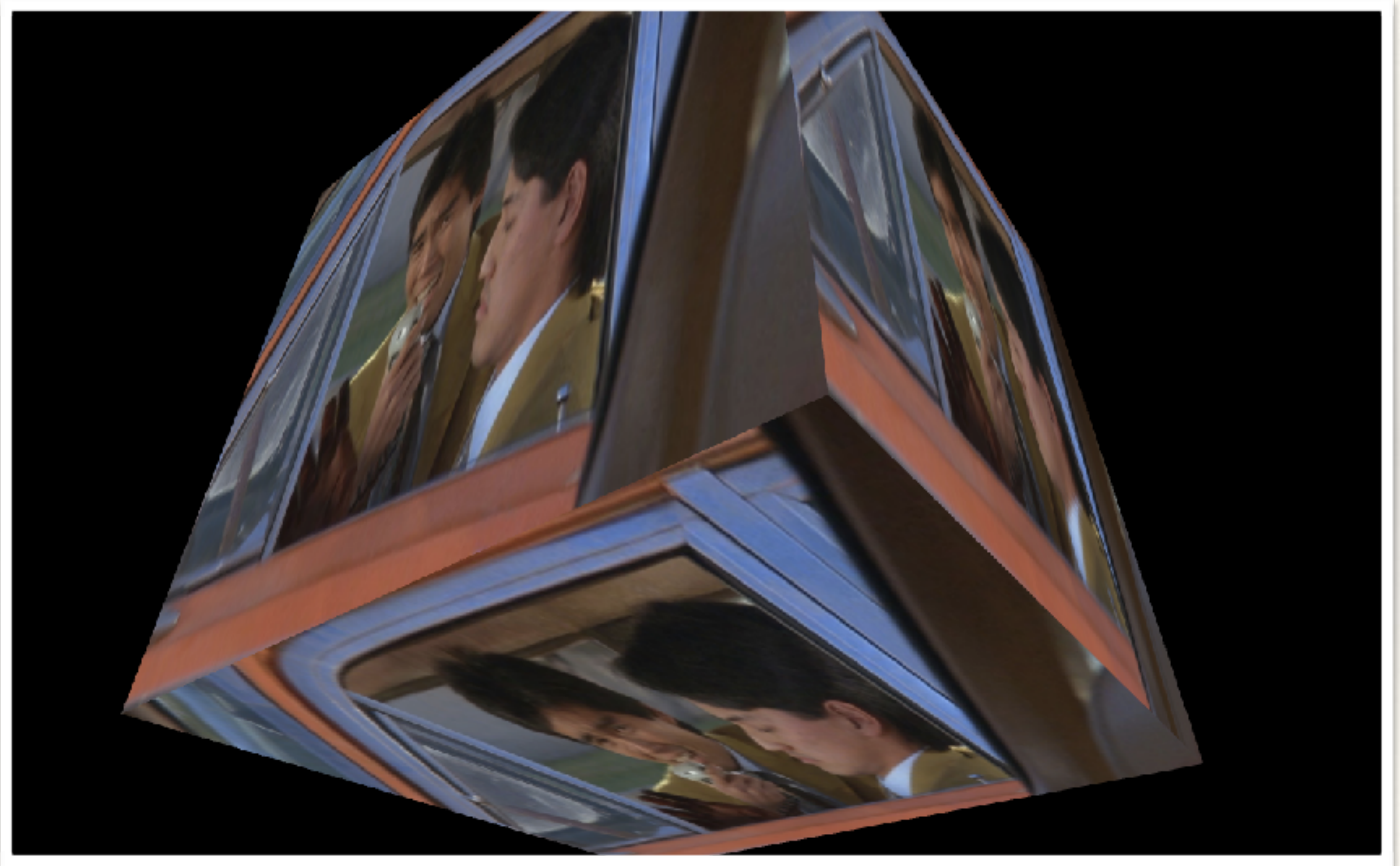
Rotation

- Or we could spin it.
- We'll do this in animate().

Rotation

```
function animate() {
    requestAnimationFrame( animate );
    mesh.rotation.x += 0.01;
    mesh.rotation.y += 0.01;
    renderer.render( scene, camera );
}
```

Rotation



But That’s a Picture of Someone Else

No Problem,
We’ll Use the Camera

Textures from the Camera

- Three.JS is a 3D animation library, and does not have camera support built-in.
- To capture a screen shot from the camera, and use it as a texture in the Three.JS animation, we'll have to extract an image from the video and pass it to our Three.JS code.

Steps

1. Call getUserMedia() to get a video stream.
2. Assign this stream to a <video> tag.
3. Copy a single frame of this video to a <canvas> tag.
4. In our Three.JS scene, create a texture from this canvas tag, rather than from an image.

Step 1: Call getUserMedia()

```
var videoParams = { "video": true };
var errorCallback = function(error) {
   console.log("Video capture error: ", error.code);
};

navigator.getUserMedia(videoParams, function(stream) {
   // do something with the stream
}, errorCallback);
```

Step 2: Assign this to a Video tag

```
var videoParams = { "video": true };
var errorCallback = function(error) {
   console.log("Video capture error: ", error.code);
};

if(navigator.getUserMedia) {
   navigator.getUserMedia(videoParams, function(stream) {
      myVideoTag.src = stream;
      myVideoTag.play();
   }, errorCallback);
}
```

Step 3: Copy a Video Frame to a Canvas tag

```
var canvas = document.getElementById("canvas");

var context = canvas.getContext("2d");

context.drawImage(myVideoTag, 0, 0, 1024,1024);
```

Step 4: Create a Texture from the Canvas

```
var texture = new THREE.Texture(canvas);

texture.needsUpdate = true;

var geometry = new THREE.BoxGeometry( 33,33,33 );

var material = new
THREE.MeshBasicMaterial( { map: texture } );

mesh = new THREE.Mesh(geometry, material);
```

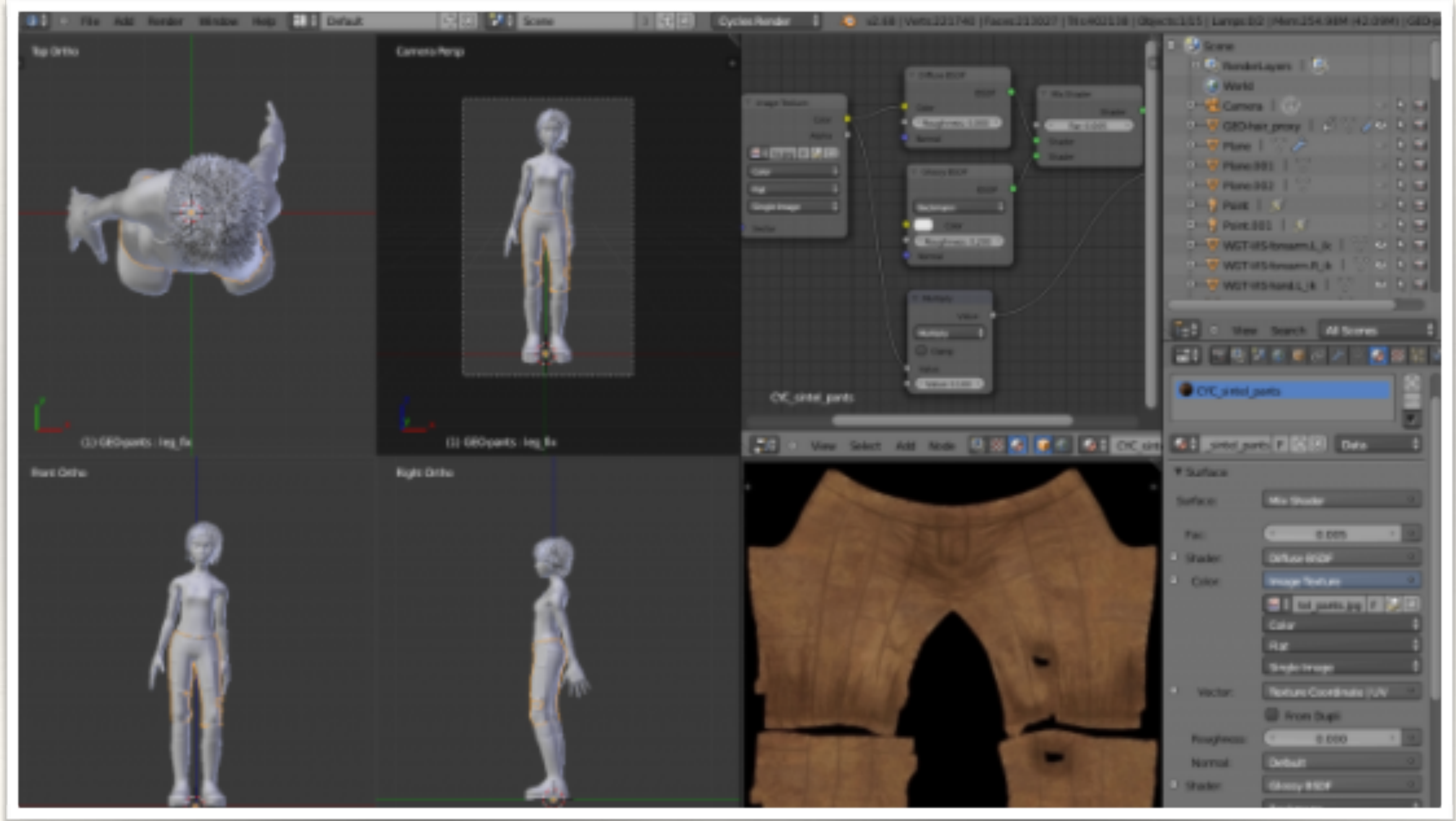
Textures from the Camera

Blender

Blender

- An app to create 3D objects and export them to JavaScript that can be loaded by Three JS.
- Makes creating objects easy.
- Download: https://www.blender.org/
- Tutorial: http://blog.teamtreehouse.com/the-beginners-guide-to-three-js

Blender



File
Add
Render
Window
Help
Default
Scene
Cycles Render
Top Ortho
Camera Persp
Front Ortho
Right Ortho
CYC_sintel_pants
Scene
RenderLayers
World
Camera
Plane
View
Select
Add
Node
Surface
Mix Shader
Diffuse BSDF
Image Texture
Glossy BSDF

Reference

Reference

- An excellent set of slides to learn Three JS are at: http://davidscottlyons.com/threejs/presentations/frontporch14/#slide-0