Intro to Three.js

Before:
In 90s browser can at most display 2000texts.
C++ is the common tool to do 3D programming.

Now:
We have WebGL
We can do 3D programing on the browser.

Since it is kind of painful to do 3D programming with WebGL API, some nice people built some nice frameworks on top of the WebGL.

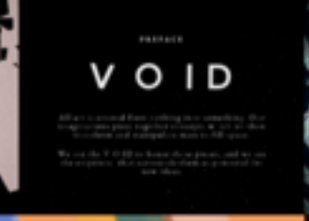
And Three.js is one of the best.



RÉSONANCE
L'ODYSSÉE SYMPHONIQUE

WE ARE ACTIVATION NODEPLUS


OMM



VOID

ENOUGH


DENNIS

THE
Fallen
OF
World War II

THE BOAT





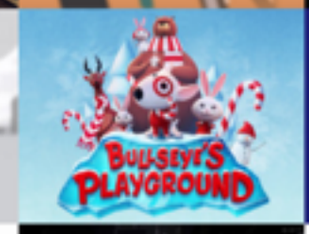
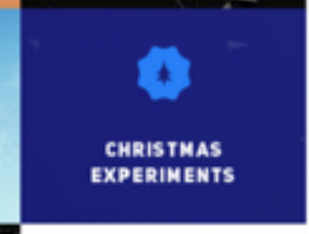
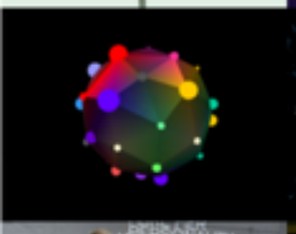
GIVE
SHAPE


THE DILLA
DIMENSION

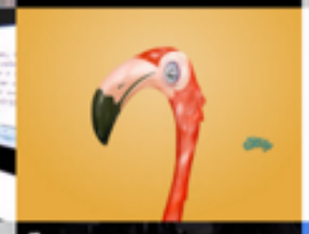

WAY TO GO

URBAN GALAXY ONLINE

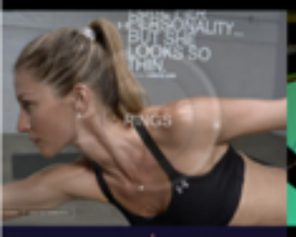
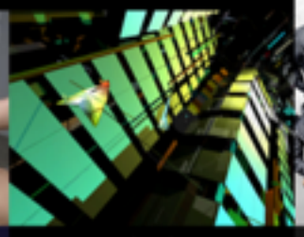
LEIA


BULLSEYE'S
PLAYGROUND

CHRISTMAS
EXPERIMENTS






A360

BUT SHE
LOOKS SO
THIN


O Green

verold

963

photosynth

The
CHRISTMAS
EXPERIMENTS
One Day, One Creation

THE HOBBIT
A JOURNEY
THROUGH
MIDDLE-EARTH

Clara.io

iAd

BLAST

London Olympics

Google

https://github.com/mrdoob/three.js/

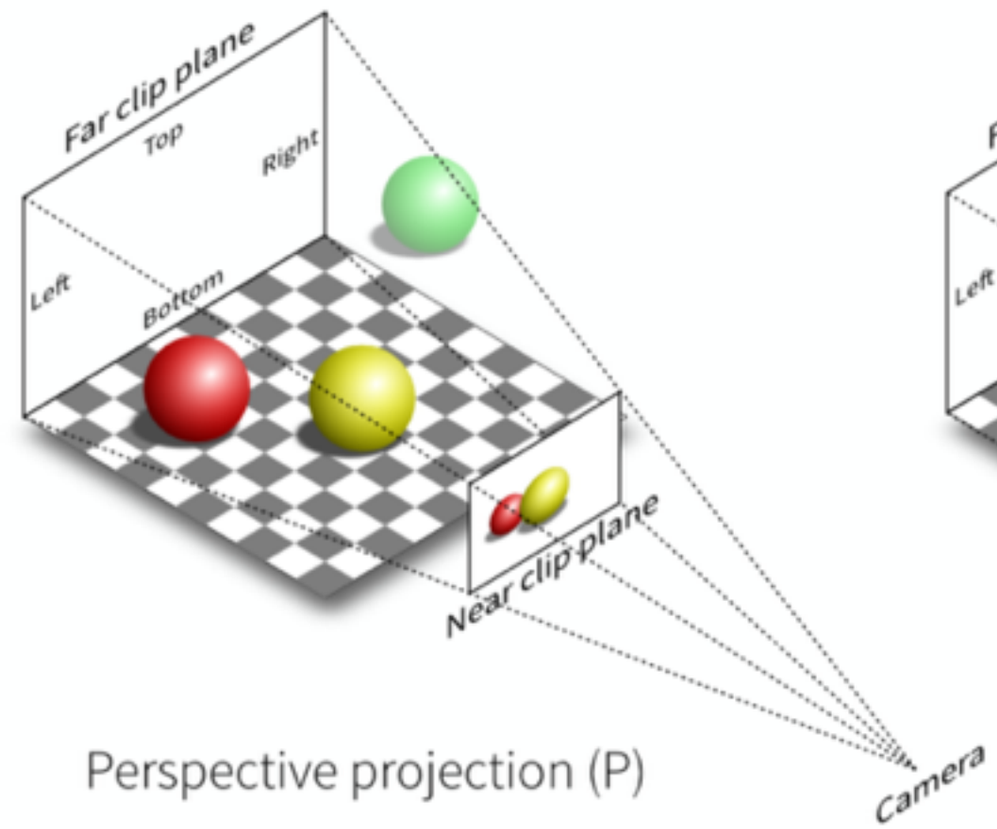
Create Your First Scene

Scene
Camera
Renderer
Object

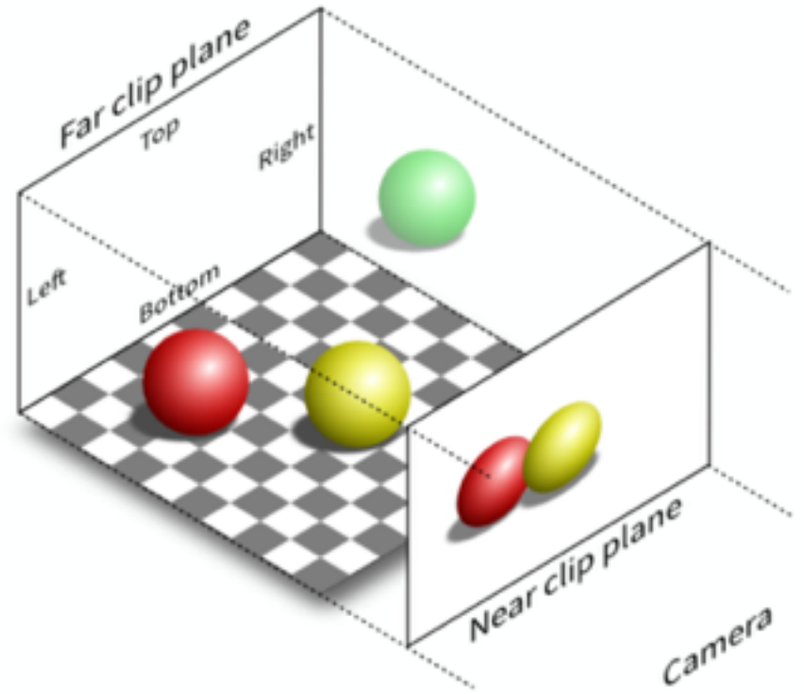
```
var scene = new THREE.Scene();
var camera = new THREE.PerspectiveCamera(75, window.innerWidth/window.innerHeight, 0.1, 1000);

var renderer = new THREE.WebGLRenderer();
```

```
var scene = new THREE.Scene();
scene.add();
scene.fog = new THREE.Fog( hex, near, far );
scene.background = new THREE.Color();
```



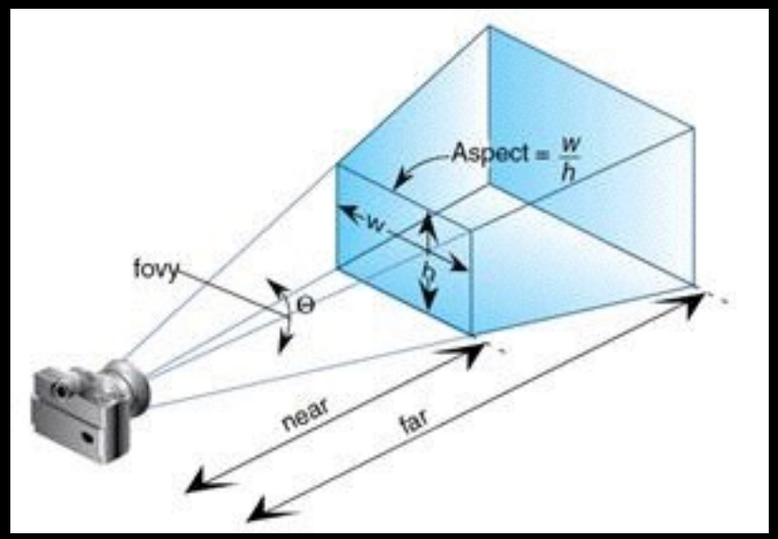
Perspective projection (P)



Orthographic projection (O)

PerspectiveCamera(fov, aspect, near, far)

fov — Camera frustum vertical field of view.
aspect — Camera frustum aspect ratio.
near — Camera frustum near plane.
far — Camera frustum far plane.



FOV: Degree of your eyes opening. If its 0 that means your eyes are closed, you cant see nuthin. If its 180, your eyes are wide open and you can see everything but everything become smaller.

Near: The closest position where you can see, any number less than that you can’t see. Don't put a negative value in this parameter, Three.js cant get your joke.

Far: The furthest position where you can see, anything beyond that you can't see.

Aspect ratio: The ratio between width and height. For example, big screen might have a small aspect ratio since its super wide. Anything that has a vertical screen might have a larger aspect ratio, such as your phone screen.

```
renderer = new THREE.WebGLRenderer();
renderer.setPixelRatio( window.devicePixelRatio );
renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );
```

```
var geometry = new THREE.BoxGeometry();
var material = new THREE.MeshBasicMaterial();
var cube = new THREE.Mesh(geometry, material);
scene.add(cube);
```

MeshBasicMaterial - A material for drawing geometries in a simple shaded (flat or wireframe) way.

MeshLambertMaterial - A material for non-shiny (Lambertian) surfaces.

MeshPhongMaterial - A material for shiny surfaces.

MeshStandardMaterial - A material that can stimulates metal.

ShaderMaterial - Material rendered with custom shaders. A shader is a small program written in GLSL to run on the GPU. You may want to use a custom shader if you need to:
1.implement an effect not included with any of the built-in materials
2.combine many objects into a single Geometry or BufferGeometry in order to improve performance