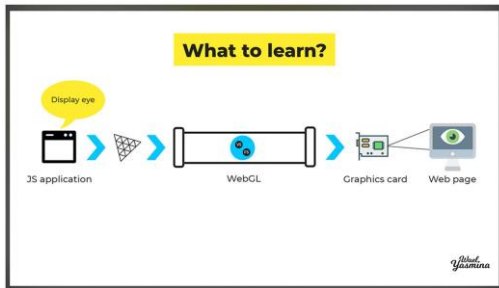


# Simpel guide til Three.js

## Hvad er THREE.js



Ny udvikling i moderne browsere og bærbare computere (smartphones) gør det nu endnu mere håndgribeligt at bruge 3D i frontend og markedsføring. Three.js er et forsøg på at skabe et nemt, letvægtigt, generelt 3D bibliotek til anvendelse på tværs af browsere.

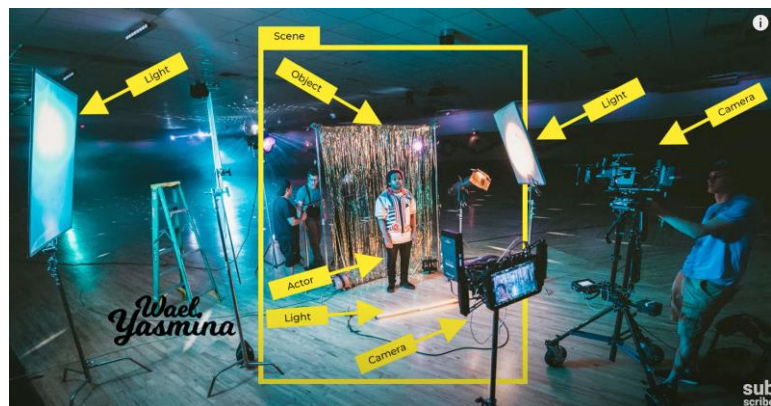
WebGL renderer derfor scenen, kamera og alt andet indhold som vi opsætter med code. Til forskel fra 2D hvor brugeren har det endelige perspektiv.

## The scene

Med 3D arbejdes der med de klassiske 3 akser med samt negative værdier/akser. Som skrevet tidligere skal scenen opstilles med alt som et skuespil behøver og manuskriptet er:

```
function animate();
```

(ovenstående billeder er fra Wael Yasmīnas videoer, og alle rettigheder henvises dertil)



## How to

1. Som start på et Three.js projekt skal der installeres en module bundler. Jeg valgte webpack med succes og henviser til Carstens videoer under Automatisering 1;

Automatisering 13 – webpack config, Automatisering 17 – webpack billeder (samt 19+20)

Følgende skrives i terminalen jævnfører ovenstående vidoer:

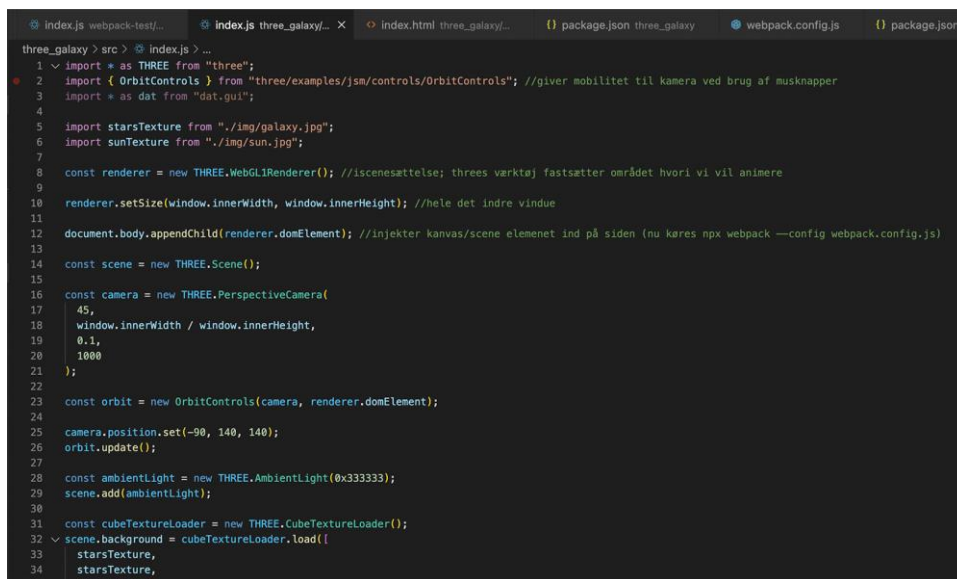
```
npm init -> npm i webpack webpack-cli -> npm i webpack-dev-server -D
```

OBS! Køres først når der er indhold at renderer; npx webpack --config webpack.config.js (skal renderer en main.js i build)

Andre bundlere kan bruges dog kan Parcel give problemer.

2. Install THREE: npm i three

3. Indhold codes med henvisning til Wael Yasminas begynder tutorials, her er udgangspunkt i solsystems øvelsen:



```
three_galaxy > src > index.js > ...
1 import * as THREE from "three";
2 import { OrbitControls } from "three/examples/jsm/controls/OrbitControls"; //giver mobilitet til kamera ved brug af musknapper
3 import * as dat from "dat.gui";
4
5 import starsTexture from "../img/galaxy.jpg";
6 import sunTexture from "../img/sun.jpg";
7
8 const renderer = new THREE.WebGLRenderer(); //iscenesattelse; threes værktøj fastsætter området hvori vi vil animere
9
10 renderer.setSize(window.innerWidth, window.innerHeight); //hele det indre vindue
11
12 document.body.appendChild(renderer.domElement); //injekter kanvas/scene elementet ind på siden (nu køres npx webpack --config webpack.config.js)
13
14 const scene = new THREE.Scene();
15
16 const camera = new THREE.PerspectiveCamera(
17   45,
18   window.innerWidth / window.innerHeight,
19   0.1,
20   1000
21 );
22
23 const orbit = new OrbitControls(camera, renderer.domElement);
24
25 camera.position.set(-90, 140, 140);
26 orbit.update();
27
28 const ambientLight = new THREE.AmbientLight(0x333333);
29 scene.add(ambientLight);
30
31 const cubeTextureLoader = new THREE.CubeTextureLoader();
32 scene.background = cubeTextureLoader.load([
33   starsTexture,
34   starsTexture,
```

Vi instancierer scenes sider som en kasse med givet 6 sider af klassen CubeTextureLoader(). Dernæst instancieres solen som en 'instance' af TextureLoader() klassen, hvor det draperes i den instancierede Mesh klasse, for så at animerer denne under function animate(). Dette kan gøres da animation er en sekvens af transformationer, der sker over tid:

```

three_galaxy > src > index.js > ...
34   starsTexture,
35   starsTexture,
36   starsTexture,
37   starsTexture,
38   starsTexture,
39   });
40
41   const textureLoader = new THREE.TextureLoader();
42
43   //sun center of solar system
44   const sunGeo = new THREE.SphereGeometry(16, 30, 30);
45   const sunMat = new THREE.MeshBasicMaterial({
46     map: textureLoader.load(sunTexture),
47   });
48   const sun = new THREE.Mesh(sunGeo, sunMat);
49   scene.add(sun);
50
51   function animate() {
52     sun.rotateY(0.004);
53
54     renderer.render(scene, camera); //linker scenen med kameraet, sendes med som argument
55   }
56
57   renderer.setAnimationLoop(animate); //sender animate med som argument til setAnimationLoop
58
59   window.addEventListener("resize", function () {
60     camera.aspect = window.innerWidth / window.innerHeight;
61     camera.updateProjectionMatrix(); //matrix which contains the projection
62     renderer.setSize(window.innerWidth, window.innerHeight);
63   });

```

## Perspektiver:

Jeg tror ikke det er et understatement at sige at 3D vil være alle steder indenfor de næste 5 år. Skal du købe et spisebord og se om det passer i stuen eller se om kjolen fra zalando passer til de nye sko? – 3D og filtre hvis du spørger mig. Som med filtre på SoMe/Markedsføring websites/spil eller portefolie websites så giver 3D en følelse gamificering og customisation. Personligt tror jeg ikke at jeg har en præcis ide om hvilket omfang det vil få, men vi ser med alt andet teknologi at desto tættere på den menneskelige virkelighed de kan komme, desto mere begejstring kan det medføre. Men ligesom LP'en så er 2D nok kommet for at blive.

## Kilder/sources:

<https://threejs.org/>

<https://threejs.org/manual/#en/fundamentals>

<https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene>

[Docs - MeshBasicMaterials](#)

<https://www.npmjs.com/package/dat.gui>

[Youtube - Beginners tutorial Wael Yasmina](#)

Leslie WU08

[LesM007/repo\\_solsystem](#)

[LesM007/repo test](#)