Setup

Jeg importerer THREE.js, PointerLockControls og GLTFLoader.

Three.js er et bibliotek der bruger WebGL til at render 3D grafik i browser

PointLockControls er en API der er designet til at arbejde i ”First Person”

GLTFLoader bliver brugt til at indlæse .gltf filer (3D modeller)

```
import * as THREE from 'three';
import { PointerLockControls } from 'three/examples/jsm/controls/PointerLockControls';
import { GLTFLoader } from 'three/examples/jsm/loaders/GLTFLoader';
```

Jeg har lavet en variabel ”scene” der THREE.JS metode ”Scene()” der bruges til at definere hvad der skal renders på siden.

Nedenunder bruger jeg ”PerspectiveCamera()” der tager imod forskellige argumenter. 1. parameter er vertikal vinkel, altså hvor tæt kameraet er på scene. 2.parameter er hvor meget kameraet skal opfange (som regel bruger man browserens width/height). 3. og 4. er hvor tæt og hvor langt kameraet skal opfange.

”WebGLRenderer()” er metoden der håndterer rendering på siden.

```
var mouse = { x: 0, y: 0 };
var flashlight;
const scene = new THREE.Scene();

const camera = new THREE.PerspectiveCamera(100, window.innerWidth / window.innerHeight, 0.1, 1600);

const renderer = new THREE.WebGLRenderer({
  canvas: document.querySelector('#bg'),
});

renderer.setPixelRatio(window.devicePixelRatio);
renderer.setSize(window.innerWidth, window.innerHeight);
camera.position.setZ(-100);

const controls = new PointerLockControls(camera, renderer.domElement);
```

getGrounds laver mit rum ud af kasser. Den bruger 4 parametre: x position, y, position, x position og imageI, som bliver brugt til at definere hvilket billede fra mit Images array der skal indlæses som texture på hver kasse(eller mur)

Jeg kører et for-loop og for hver iteration henter jeg x, y, z positioner fra mit positions array og randomizer imageI. Derefter bliver getGround funktionen kaldt og sætter muren ud fra positionerne og indlæser den tilfældige texture.

```
const images = ["images/1.jpg", "images/2.jpg", "images/3.png", "images/4.png", "images/5.jpg", "images/6.jpg"]
let imageI = 0;


const positions = [
    [-600, 1200, 0], [600, 1200, 0], [0, 1200, 0],
    [-600, 1200, 600], [600, 1200, 600], [0, 1200, 600],
    [-600, 1200, -600], [600, 1200, -600], [0, 1200, -600],
    [0, 0, -1200], [600, 0, -1200], [-600, 0, -1200],
    [0, 600, -1200], [600, 600, -1200], [-600, 600, -1200],
    [0, 0, 1200], [600, 0, 1200], [-600, 0, 1200],
    [0, 600, 1200], [600, 600, 1200], [-600, 600, 1200],
    [-1200, 600, 0], [-1200, 600, 600], [-1200, 600, -600],
    [-1200, 0, -600], [-1200, 0, 0], [-1200, 0, 600],
    [1200, 600, 0], [1200, 600, 600], [1200, 600, -600],
    [1200, 0, -600], [1200, 0, 600], [1200, 0, ],
    [600, -600, 0], [600, -600, 600], [600, -600, -600],
    [0, -600, 0], [0, -600, 600], [0, -600, -600],
    [-600, -600, 0], [-600, -600, 600], [-600, -600, -600]
];

for (let i = 0; i < positions.length; i++) {
    imageI = Math.floor(Math.random() * images.length);
    const [x, y, z] = positions[i];

    scene.add(getGround(x, y, z, imageI));
}
```

```
function getGround(x, y, z, i) {
    let texture = new THREE.TextureLoader().load(images[imageI])
    const groundGeometry = new THREE.BoxGeometry(600, 600, 600);
    const groundMaterial = new THREE.MeshStandardMaterial({ map:texture});
    const ground = new THREE.Mesh(groundGeometry, groundMaterial);
    ground.position.set(x, y, z);
    return ground;
}
```

Jeg har defineret funktionen ”animate” som render scene og kamera, når den bliver kaldt.
Jeg har så lavet et animations loop ved brug af requestAnimationFrame, som kalder animate 60 gange i sekundet om og om igen.

```
function animate() {
  requestAnimationFrame(animate);
  renderer.render(scene, camera);
}
```

Jeg bruger GLTFLoader til at indlæse min lommelygte.
Bagefter har jeg pillet ved lygtens position og rotation, så dens start position matcher mit kamera og tilføjet spotlight (lyset) som child af flashlight.

```
loader.load('models/scene.gltf', function (gltf) {
    flashlight = gltf.scene;
    scene.add(flashlight);
    flashlight.add(spotlight);
    flashlight.position.z = -90;
    flashlight.position.x = -25;
    flashlight.position.y = -40;
      camera.rotation.y = Math.PI;
}, undefined, function (error) {
    console.error(error);
  });
```

Jeg lavet et spotlight ved brugen af SpotLight klassen, som tager i mod et parameter der definerer lysets farve i hexadecimal.

Nedenunder styrer jeg spotlights properties: angle, penumbra og decay.
Angle håndterer hvor tæt strålen er, hvilket resulterer i et større eller minde område der bliver oplyst.

Penumbra er hvor ”bløde” kanterne er. 0 er helt skarpe kanter. Jeg synes personligt at 1 får lyset til at se mere realistisk ud.

Decay er hvor meget strålens kraft aftager ved længere distance.

```
const spotlight = new THREE.SpotLight(0xffffff);
spotlight.angle = 1;
spotlight.penumbra = 1;
spotlight.decay = 2;
```

Jeg har lavet en eventlistener der kører min ”onMouseMove” funktion når musen bevæger sig.

mouse.x og mouse.y variablerne bliver opdateret ud fra musens position i browser vinduet.

Disse variabler bliver brugt til at lave sort magi som får lyset og lommelygten til at følge musens position (nogenlunde)

```
function onMouseMove(event) {
  controls.lock();
  mouse.x = (event.movementX / window.innerWidth);
  mouse.y = (event.movementY / window.innerHeight);

  var vector = new THREE.Vector3(-mouse.x, -mouse.y, 0.5);
  vector.unproject(camera);
  var dir = vector.sub(camera.position).normalize();
  var target = new THREE.Vector3().addVectors(camera.position, dir.multiplyScalar(100));

  var distance = camera.position.z / dir.z;
  var pos = camera.position.clone().add(dir.multiplyScalar(distance));

  spotlight.position.copy(pos);

  flashlight.lookAt(target);

}

document.addEventListener('mousemove', onMouseMove, false);
```

Jeg har lavet funktionen flicker, som justerer lysets styrke tilfældigt fra 0.5 til 1.

flicker bliver kaldt ved ”id” som er en setInterval der kører 10 gange i sekundet. Det giver lyset en effekt af lavt batteri/defekt lommelygte

Hver 30 sekundt kører ”stop”, som også er en setInterval.
Stop sætter spotlights styrke til 0 og stopper min flicker setIntervals, så der er helt mørkt.
Nedenunder har jeg lavet en setTimeout som så starter min flicker igen efter 0.75 sekunder, så der er lys igen.

```
let TurnOFF = 30000;
let id = setInterval(flicker, 100);

function flicker() {
  spotlight.intensity = Math.floor((Math.random() * 5) + 3) / 10;
}

let stop = setInterval(() => {
  spotlight.intensity = 0;
  clearInterval(id);
  setTimeout(() => {
    id = setInterval(flicker, 100);
  }, 750);
}, TurnOFF);
```