

[illegible]

ENV SETUP:

Prva i najvažnija stvar jeste da imate instaliran node.js na vašem kompjuteru.

WINDOWS // MAC // LINUX

Koristimo biblioteke i skripte koje nisu dostupne bez manuelne instalacije i koje nažalost neće raditi, ako probamo da ih otvorimo na običnom live-serveru. Zbog toga koristimo npm i npx pakete i komande. (Deluje vrlo teško i naporno, ali ako ste uspešno instalirali biće vrlo jednostavno)

U VSCodeu u gornjem meniju idite na terminal i otvorite novi terminal. Proverite da li je file path terminala isti kao folder u kom želite da radite.

Primer: Ako radite u folderu FMK1 koji je na vašem desktopu, terminal bi trebalo da prikaže C:\Users\%USER\Desktop\FMK1>)

Pokrenućemo par instalacionih komandi:

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

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

Napravite `index.html` i `main.js` fajlove. `Index.html` treba da ima `script` tag koji vodi na `main.js`, `canvas` sa `id-om` i `<main>`.

```

1  { } package.json
2
3  <? index.html • JS main.js
4
5  <? index.html > html > body > main
6
7  <!DOCTYPE html>
8
9  <html lang="en">
10
11  <head>
12
13    <meta charset="UTF-8">
14
15    <meta http-equiv="X-UA-Compatible" content="IE=edge">
16
17    <meta name="viewport" content="width=device-width, initial-scale=1.0">
18
19    <title>Document</title>
20
21  </head>
22
23  <body>
24
25    <canvas id="bg"></canvas>
26
27    <main>
28
29    </main>
30
31    <script type="module" src=".main.js"></script>
32
33  </body>
34
35  </html>

```

Sada možete da pokrenete svoj live server uz sledeću komandu:

```
npx vite
```

Ovim je pokrenut live-server koristeći vite.js framework.

Svaki put kada sačuvate svoj kod, dokle god je ova komanda aktivna, automatski će se refrešovati vaš live-server.

Trebalo bi da dobijete local-host link ka svom serveru u terminalu sličnom ovome:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    COMMENTS
```

```
VITE v4.2.1  ready in 156 ms
```

```
→ Local:  http://localhost:5173/
```

```
→ Network: use --host to expose
```

```
→ press h to show help
```

```
(!) Could not auto-determine entry point from rollupOptions  
dependency pre-bundling.
```

[illegible]

JAVASCRIPT:

Imamo HTML i imamo live-server. Možemo početi kodovanje scene u main.js fajlu.

Importi su malo drugačiji od klasičnog rada sa three.js-om.

```
// imports
import * as THREE from "three";
import { GLTFLoader } from 'three/addons/loaders/GLTFLoader.js';
import { OrbitControls } from 'three/examples/jsm/controls/OrbitControls.js';

JS main.js
JS main.js > ...
1 // imports
2 import * as THREE from "three";
3 import { GLTFLoader } from 'three/addons/loaders/GLTFLoader.js';
4 import { OrbitControls } from 'three/examples/jsm/controls/OrbitControls.js';
5
6 // 3d file preload
7 const blenderFajl = new URL('Vasilije.glb', import.meta.url);
8
9 // renderer
10 const renderer = new THREE.WebGLRenderer({
11     canvas: document.querySelector('#bg'), alpha: false
12 });
13 renderer.setSize(window.innerWidth, window.innerHeight);
14 document.body.appendChild(renderer.domElement);
15
16 // scene
17 const scene = new THREE.Scene();
18
19 //camera
20 const camera = new THREE.PerspectiveCamera(45, window.innerWidth / window.innerHeight, 0.01, 1000);
21 camera.position.setZ(5);
22 camera.position.setY(5);
23 camera.position.setX(0);
24
25 // orbit controls
26 const controls = new OrbitControls(camera, renderer.domElement);
27 controls.enabled = false;
28
```

Preskočicu objašnjavanje postavljanja kamere, scene i renderera (linije 9-23). O tome možete pogledati rezime Three.js-a.

Linija 7 je važna, sa njom procesujemo naš blender fajl. U blenderu sačuvajte svoj fajl kao .glb i stavite ga u isti direktorijum u kom pišete kod. Varijabla može da se zove kako god vi želite, važno je samo da umesto 'Vasilije.glb' bude tačan naziv vašeg fajla. Nakon toga dodajete `import.meta.url` i zatvarate `new URL()` funkciju.

Linija 25 nam omogućava da pomeramo kameru mišem na browseru da bi mogli lakše da osmislimo kadar koji posle želimo da zamrznemo, osim ako želite da napravite igricu ili sajt kroz koji korisnik može da se kreće orbit kontrolama.

Linija 26 prima boolean type, što znači da `true` dozvoljava orbiting, dok `false` ih isključuje.

```
29 // Light
30 const pointLight = new THREE.PointLight(0xff00ff, 2.0);
31 pointLight.position.set = (300,500,0)
32 const Light2 = new THREE.DirectionalLight(0xffffffff, 1.0);
33 scene.add(pointLight, Light2);
34
35 // Load assets
36 const assetloader = new GLTFLoader();
37 assetloader.load(blenderFajl.href, function(gltf) {
38     const model = gltf.scene;
39     scene.add(model);
40 }, undefined, function(error){
41     console.log(error);
42 });
43
```

SAVET: Kada radite sa .glb modelima koristite `directional` osvetljenje – osvetliće vam ceo model.

Linije 36 – 42 prekopirajte, one omogućavaju procesovanje i loadovanje .glb fajlova u browseru.

```
45 // draw
46 function animate() {
47     renderer.render(scene, camera);
48 }
49 renderer.setAnimationLoop(animate);
```


Za kraj samo podesite animatenu funkciju kao i kada radite sa three.js geometrijom i to je to. Uspešno ste učitali svoj blender model na browser. Slobodno unesite CSS fajl iz prošle vežbe u ovaj folder i alterujete HTML strukturu.

HAPPY DESIGNING !

[illegible]

CAMERA SCROLL:

```
44 // scroll Animation
45
46 function moveCamera() {
47     const t = document.body.getBoundingClientRect().top;
48
49     camera.rotation.x = t * +0.000007;
50     camera.position.x = t * +0.0007;
51     camera.position.y = t * -0.001;
52     camera.rotation.z = t * -0.0002;
53 }
54
55 document.body.onscroll = moveCamera;
56 moveCamera();
```

Pripremio: Andrej Alfirević

[illegible]

WEBGL **17.04.2023.**
FMK_SUMMER_SEMESTER_2K23//GIT