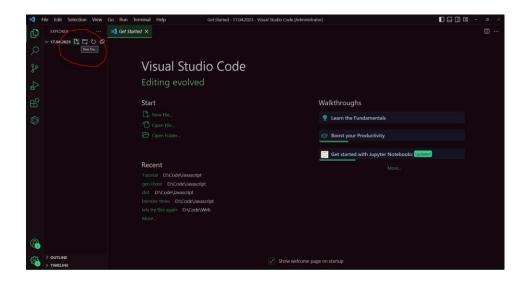
THREE.JS 17.04.2023. FMK_SUMMER_SEMESTER_2K23//GIT

PROJECT SETUP:

> Prvo je važno da napravimo naše html i javascript files.



- > HTML file nazivamo index.html, dok javascript file nazivamo main.js
- > U HTML file ispod < body> tag-a dodajemo sledeće dve skripte. (NAPOMENA main.js skripta treba da se zove onako kako ste nazvali vaš javascript file, dok druga skripta mora da bude identična).

```
<script src='https://cdnjs.cloudflare.com/ajax/libs/three.js/r83/three.min.js'></script>
 ♦ index.html > ♠ html > ♠ script
       <!DOCTYPE html>
       <html Lang="en">
       <head>
           <meta charset="UTF-8">
           <meta http-equiv="X-UA-Compatible" content="IE=edge">
           <meta name="viewport" content="width=device-width, initial-scale=1.0">
           <title>Document</title>
       </head>
       <body>
       </body>
           <script src="main.js"></script>
           <script src='https://cdnjs.cloudflare.com/ajax/libs/three.js/r83/three.min.js'></script>
       </html>
```

> Nakon toga u *<body>* tag-u stavljamo <canvas> tag sa id-om. U ovom primeru id će imati ime "bg".

```
♦ index.html
              JS main.js
<!DOCTYPE html>
     <html Lang="en">
      <head>
         <meta charset="UTF-8">
         <meta http-equiv="X-UA-Compatible" content="IE=edge">
         <meta name="viewport" content="width=device-width, initial-scale=1.0">
          <title>Document</title>
      </head>
      <body>
 10
         <canvas id="bg"></canvas>
     </body>
         <script src="main.js"></script>
          <script src='https://cdnjs.cloudflare.com/ajax/libs/three.js/r83/three.min.js'></script>
      </html>
```

Ovde je gotov setup projekta u HTML-u.

> Posto koristimo javascript "biblioteku" moramo da je importujemo u js file.

import * as THREE from "https://unpkg.com/three@0.127.0/build/three.module.js";

SCENE SETUP:

- > Prvo moramo napraviti *scenu* u kojoj ćemo praviti i manipulisati 3D grafikom.
- >> To radimo tako sto napravimo novu varijablu i pridajemo joj funkciju new THREE.Scene();
- >> Možete promeniti pozadinu scene tako sto vasoj varijabli dodate atribut .background. (Boja mora biti u hexadecimalnoj vrednosti [primer sa slike je crna boja pozadine "0x000000"])
- >> $\underline{\text{Ovde}}$ možete da birate boje i da odmah dobijete njenu hexadecimalnu vrednost (samo zamenite # sa 0x i ctrl+V ostatak [primer #FF5633 \rightarrow 0xff5633])



CAMERA:

- > Da bi smo videli sta se desava u nasoj sceni treba nam kamera. Nju dodajemo kroz kod na slici.
- >> Za one koje zanima: Kao i u blenderu postoje Perspective i Orthographic kamera. Perspective kamera uzima FOV, aspect ratio i near-far argumente, dok orthographic kamera uzima shader i procesing argumente. Mozete pročitati tačnu sintaksu <u>ovde</u>.

```
Js main.js > ...

1    // imports
2    import * as THREE from "https://unpkg.com/three@0.127.0/build/three.module.js";

4    // setup
5    const scene = new THREE.Scene();
6    scene.background = new THREE.Color(0x000000);
7    const camera = new THREE.PerspectiveCamera(75, window.innerWidth / window.innerHeight, 0.1, 1000);
8    scene.add(camera);
9
```

NAPOMENA! Morate dodati kameru na scenu uz scene.add(camera); [linija 8 na slici].

> Menjanje pozicije kamere može se postići ili kroz korišćenje dimension-specific atributa, to jest da stavite za svaku osu novu liniju ili možete da uradite prosto camera.position = new THREE.Vector3(x,y,z); Mada, ako koristite starije verzije biblioteke može da se desi da interpreter nije u mogućnosti da renderuje kod. U tom slučaju koristite dimension-specific verziju.

```
8  // camera
9  const camera = new THREE.PerspectiveCamera(75, window.innerWidth / window.innerHeight, 0.1, 1000);
10  scene.add(camera);
11  camera.position.setZ(0);
12  camera.position.setY(0);
13  camera.position.setX(0);
14
```

RENDERER:

```
// renderer
const renderer = new THREE.WebGLRenderer({
   canvas: document.querySelector('#bg'), alpha: true
});
renderer.setPixelRatio(window.devicePixelRatio);
renderer.setSize(window.innerWidth, window.innerHeight);
renderer.render(scene, camera);
```

> Renderer renderuje scenu. Prekopirajte ovaj kod, ali pazite da u .querySelector() pravilno napišete ID vašeg canvasa (HTML canvas id iz project setup-a)

LIGHT:

> Na našoj sceni je mrkli mrak. Kada bismo dodali neku geometriju ništa ne bismo videli. Postoje bezbroj različitih "reflektora" koje možete da koristite. Možete eksperimentisati i mešati razna svetla, ali nama je trenutno najvažnije da imamo ambijentalno svetlo koje će jednako osvetliti celu scenu.

```
// Lights
const ambientLight = new THREE.AmbientLight(0xffffff);
scene.add(ambientLight);
```

> Boja svetla se može promeniti u vidu hexkoda (linija 23). Bela će naravno predstavljati boje kakve jesu, dok će crvena naravno naglasiti crvene tonove geometrije u sceni.

POINTLIGHT-u može da se manuelno podesi pozicija.

```
// lights
const ambientLight = new THREE.AmbientLight(0xffffff);
const pointLight = new THREE.PointLight(0xffaaff);
pointLight.position.set = (300,5,-500);
scene.add(ambientLight, pointLight);
```

GEOMETRY:

> Najzabavniji deo 3.js-a jeste pravljenje i manipulisanje 3D geometrijom. Jedan objekat se sastoji od određene vrste geometrije i materijala. Dokumentacija three.js-a ima listu

geometrija i sve parametre koji vam trebaju za rad. GEOMETRIJA, MATERIJALI.

```
// geometries
const geometry = new THREE.BoxGeometry(20,20,20);
const material = new THREE.MeshStandardMaterial({ color: 0xde7a3f,wireframe:false });
const pyro = new THREE.Mesh(geometry, material);
scene.add(pyro);
```

Za geometriju i materijal morate da napravite posebne varijable (slika iznad).

- >> Line 30. je primer četvorougaone geometrije gde imamo X,Y i Z parametre koje možemo menjati da dobijemo kocku ili pravougaonik.
- >> Line 31. je primer standardnog materijala objekta. Uzima parametre boje (hexadecimalna vrednost) i wireframe-a (boolean vrednost (true ili false).
- >> Line 32. je primer pravljenja 3D objekta uz pomoć geometrije i materijala. Ime varijable PYRO je samo primer. Varijabla može da se zove kako god vi želite (to se odnosi na većinu varijabli, ali radi konvencije geometry i material uglavnom ostaju isti osim, ako radite sa velikim brojem različitih objekata i geometrija)
- >> Line 33. je najvažniji deo ovog snippeta, zato što ako ne napišemo ovu liniju koda nećemo dodati naš objekat na scenu. (Kao kada je ilustracija van artboarda u ilustratoru, a želimo da je eksportujemo u jpg/png/pdf etc.)

Kao što smo podesili poziciju kamere uz pomoć .position atributa, tako možemo da ga iskoristimo i za objekte. (primer dole)

```
pyro.position.x = 0;
pyro.position.y = 0;
pyro.position.z = -50;
```

Primer više oblika sa različitim geometrijama/materijalima:

```
const geometry = new THREE.BoxGeometry(20,20,20);
const material = new THREE.MeshStandardMaterial({ color: 0xde7a3f,wireframe:false });
const pyro = new THREE.Mesh(geometry, material);
scene.add(pyro);

pyro.position.x = 0;
pyro.position.y = 0;
pyro.position.z = -50;

const geometry2 = new THREE.TorusGeometry(30,5,5,80);
const material2 = new THREE.MeshStandardMaterial({ color: 0xde7a3f,wireframe:false });
const pyro2 = new THREE.Mesh(geometry2, material2);
scene.add(pyro2);

pyro2.position.x = 0;
pyro2.position.y = 0;
pyro2.position.z = -50;
```

ANIMACIJA I "TRUE" RE<u>NDERING:</u>

"True" rendering je jednostavno rečeno proces kojim govorimo browseru da procesuje sve što smo do sad uradili. Ovaj kod je najpraktičnije prekopirati. Razmišljajte o ovoj funkciji kao draw funckiji u p5.js-u.

Kada smo napravili scenu sa svim objektima koje želimo, možemo da ih animiramo dodajući kod u animate funkciju.

```
47
     // draw
48
     function animate() {
49
         requestAnimationFrame(animate);
50
         pyro.rotation.z += 0.01;
51
         pyro.rotation.y += 0.02;
52
53
54
         pyro2.rotation.x -= 0.01;
         pyro2.rotation.y -= 0.02;
55
         renderer.render(scene, camera);
56
57
58
     animate();
59
```

Primer: Line 51. na objekat koji želimo da dodamo (varijabla se zove pyro , pa zbog toga koristimo pyro) dodajemo atribut rotation (može da bude i position), na taj atribut dodajemo još jedan atribut koji specificira dimenziju u kojoj želimo da se nešto promeni. Kao iteratori u for/while petljama možemo da koristimo += ili -= da manipulišemo rotaciju ili poziciju objekta. **SAVET:** koristite decimale u stotinkama ili hiljadarkama (sa 2 ili 3 nule) za bolje efekte, pošto three.js animira u 60FPS po default-u. Zato, ako stavite += ili -= 1; animacija će biti previše brza.

THREE.JS WEBSITE:

Sada kada imamo završenu animaciju ili grafik koji želimo da se renderuje na sajtu, poželjno je da možemo da dodajemo tekst, slike i drugu html arhitekturu.

Prva stvar koju moramo da uradimo jeste da u html-u dodamo div van canvasa u kome ćemo izgraditi ostatak našeg sajta.

Napravite CSS file ili u style tag-u html-a podesite margin na 0; Canvasu stavite fixed poziciju, a glavnom divu (u primeru na slici *main*) apsolutnu poziciju sa z-indexom većim od 1;

```
main {
    width: 100vw;
    color: ■white;
    z-index: 99;
    position: absolute;
    width: 100%;
    margin: 0px auto;
    padding: 120px 0px;
    padding-bottom: 0px;
    display: grid;
    grid-template-columns: repeat(12, 1fr);
}

canvas {
    position:fixed;
```

Sada možete da dizajnirate i dodajete content, dok će vaš 3D grafik ostati u pozadini.

Happy designing!

CAMERA SCROLL:

```
65
66  // scroll Animation
67
68  function moveCamera() {
69     const t = document.body.getBoundingClientRect().top;
70
71     camera.position.z = t * -0.01;
72     camera.position.x = t * -0.1;
73     // camera.rotation.y = t * -0.0001;
74     }
75
76     document.body.onscroll = moveCamera;
77     moveCamera();
```

Pomoću ovog koda možete da naterate kameru da se pomera uz scroll. Zabavite se sa menjanjem promenjivih!

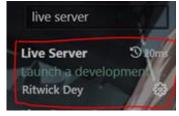
FAQ:

1 // Ceo kod i fajlove možete preuzeti na mom github repo-u.

2 // Ako koristite VSCode predlažem da instalirate LiveServer extension. Jednim klikom možete da pokrenete lokalni server i možete da proveravate greške i promene u kodu, bez potrebe da otvarate, zatvarate i refrešujete code file bezbroj puta.

Da bi ste instalirali extension uđite u VSCode i na tastaturi iskoristite shortcut CTRL+SHIFT+X.

U search baru koji se pojavi ukucajte LiveServer i instalirajte prvi koji vam izađe (autor je Ritwick Dey).



Pripremio: Andrej Alfirević

THREE.JS 17.04.2023. FMK_SUMMER_SEMESTER_2K23//GIT