

download

WEB 3D WITH **ANGULAR** AND THREE.JS

Luca Morello
Senior Full Stack Developer

In collaborazione con:



COMUNE DI BERGAMO



ABOUT ME



Web Developer
since 2010



download



SORINT_{lab}

WHAT IS THREE.JS

- It is a **library**, it isn't a framework.
- It is based on **Canvas and WebGL** (GPU power).
- It is composed by **Javascript Objects**.

download



 **SORINT**_{lab}

WHY ANGULAR?

- It provides an **organized structure** for growing apps.
- It simplifies the UI design with the **data-binding**.
- It is based on **TypeScript**, a solid JS superset.

[download](#)



 **SORINT**_{lab}

LET'S START!

Download the project from github.com/zosma180/web3d-talk

```
> cd web-3d-talk
```

```
> npm i
```

```
"@angular/*": "~11.1.0",  
"three": "0.124.0",
```

```
> npm start
```

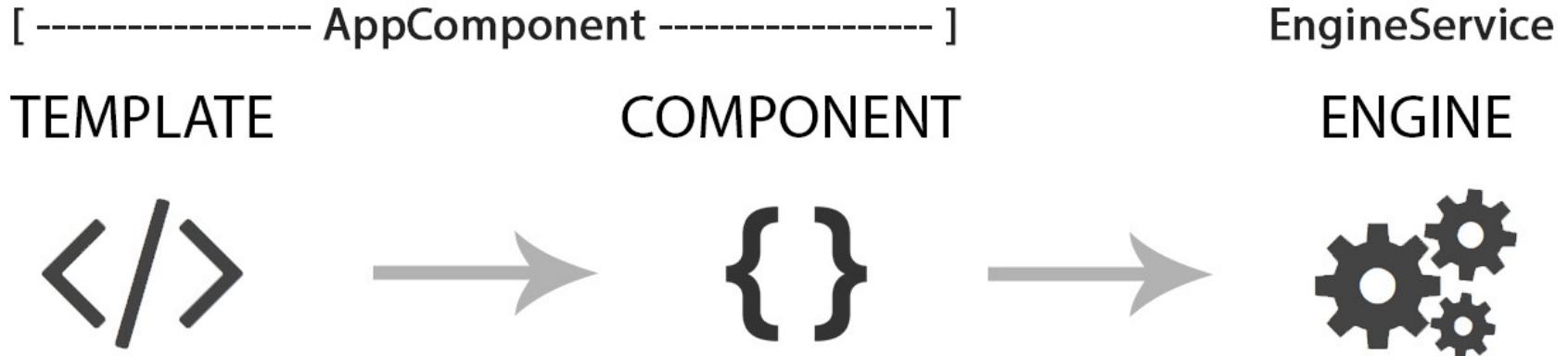
Take a quick look on the scaffolding.

[download](#)



 SORINT_{lab}

APPLICATION ENTRY POINT



- **TEMPLATE**: it has only a <div> element that will contain the generated canvas by ENGINE.
- **COMPONENT**: it loads the 3D models and call the ENGINE passing the <div> element and the models.
- **ENGINE** It does the hard work (see the next slide).

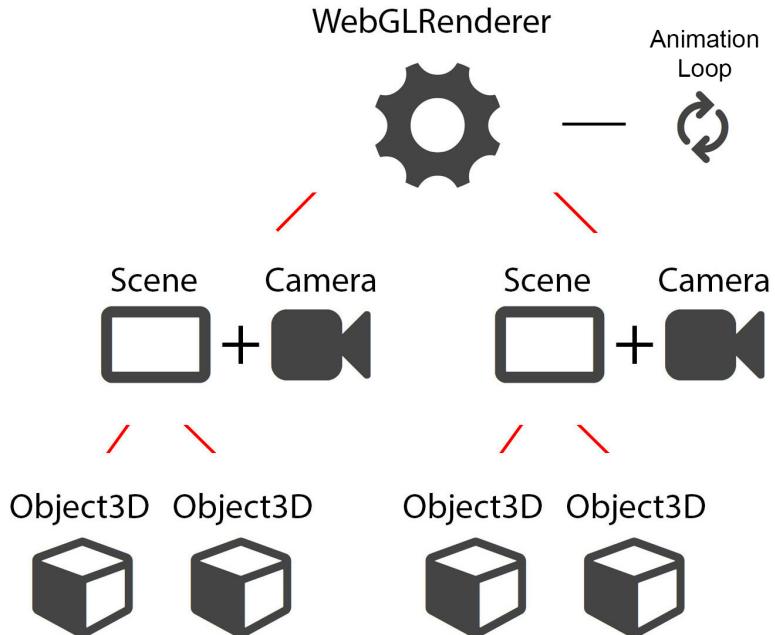
[download](#)



 SORINT_{lab}

TURN ON THE ENGINE

How it works?



See the engine logic implementation in the `engine.service.ts`

download

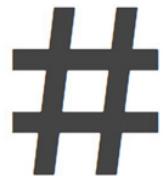


 SORINT_{lab}

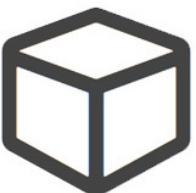
IT'S TIME TO MODEL

Basic Model

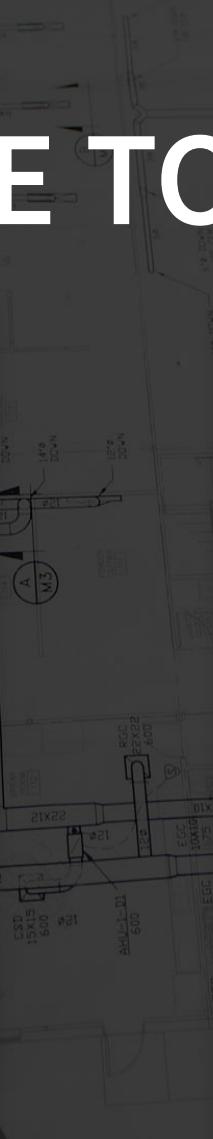
Material



Geometry



Mesh



Nesting

Object3D



Mesh



Mesh



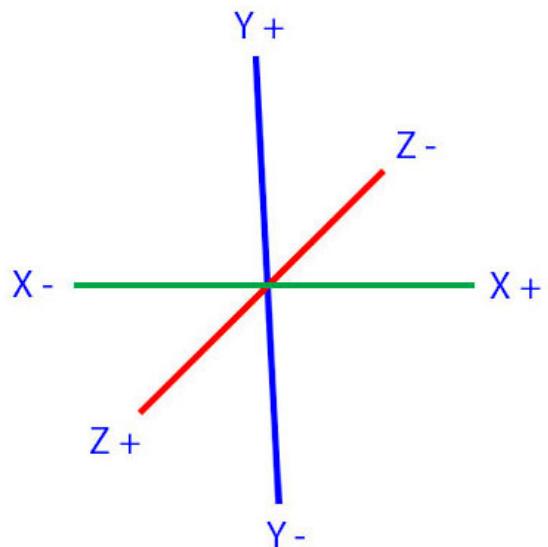
download



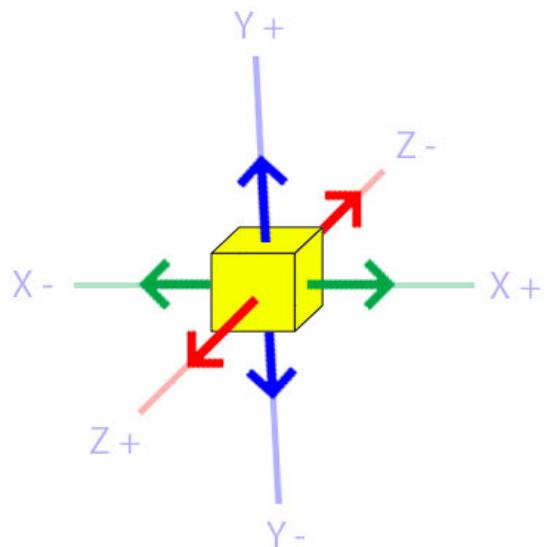
 SORINT_{lab}

SIMPLE ACTIONS

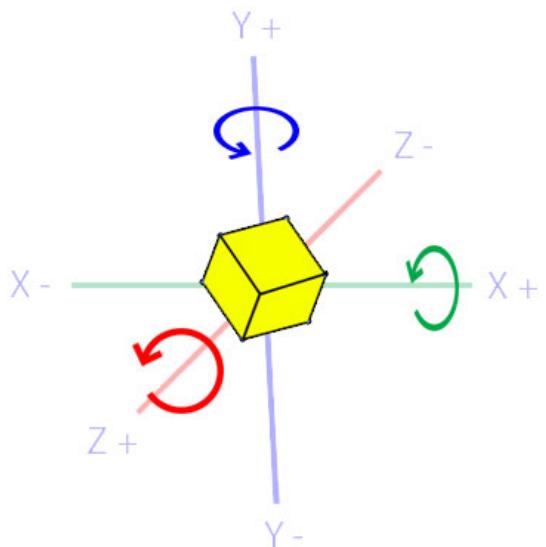
Orientation



Position



Rotation



[download](#)



 SORINT_{lab}

FLY BY A PLANET 1/3

Create a basic model in **planet.service.ts** snippet-1

```
private addPlanet(): Promise<void> {
  const geometry = new SphereBufferGeometry(50, 100, 100);
  const material = new MeshBasicMaterial({ color: '#FF0' });
  this.planet = new Mesh(geometry, material);

  this.model.add(this.planet);
  return Promise.resolve();
}
```

[download](#)



 SORINT_{lab}

FLY BY A PLANET 2/3

Add a new look with a texture snippet-2

```
private getTexture(name: string): Promise<Texture> {
  return new Promise((res, rej) => {
    new TextureLoader().load(`assets/${name}`, res, undefined, rej);
  });
}

private addPlanet(): Promise<void> {
  return this.getTexture('planet.jpg').then(texture => {
    ...
    const material = new MeshBasicMaterial({ map: texture });
    ...
  });
}
```

[download](#)



 SORINT_{lab}

FLY BY A PLANET 3/3

Add the clouds to **planet.service.ts** snippet-3

```
private addClouds(): Promise<void> {
  return this.getTexture('clouds.png').then(texture => {
    const geometry = new SphereBufferGeometry(50.1, 100, 100);

    const material = new MeshBasicMaterial({
      map: texture,
      transparent: true,
      opacity: 0.9
    });

    this.clouds = new Mesh(geometry, material);
    this.model.add(this.clouds);
  });
}
```

[download](#)



 SORINT_{lab}

AND THERE WAS LIGHT 1/3

- Lights are not lights.
- Lights are effects defined by two things:
the **direction** of beam;
the **brightness effect** on the materials.
- You can have multiple lights in the scene.



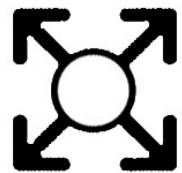
download



 SORINT_{lab}

AND THERE WAS LIGHT 2/3

Directions



PointLight



AmbientLight

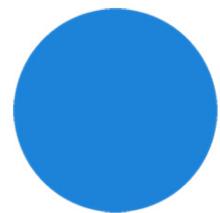


HemisphereLight



SpotLight

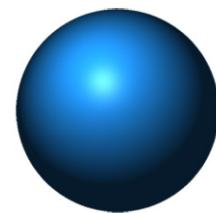
Materials



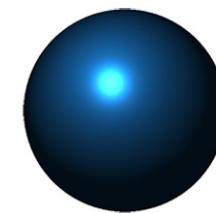
MeshBasicMaterial



MeshLambertMaterial



MeshPhongMaterial



MeshPhysicalMaterial

download



 SORINT_{lab}

AND THERE WAS LIGHT 3/3

Add the light to `engine.service.ts` snippet-4

```
init(host: HTMLElement, models: Object3D[]): void {  
    ...  
    const light = new PointLight('#FFF', 1.2);  
    light.position.set(100, 0, 130);  
    this.scene.add(light);  
    ...  
}
```

Replace all the materials in `planet.service.ts` snippet-5

```
...  
const material = new MeshLambertMaterial({ map: texture });  
...
```

[download](#)



 SORINT_{lab}

IT'S ANIMATED!

Animate all the things snippet-6

```
constructor(private engineService: EngineService) {
  this.engineService.animation$.subscribe(() => this.animate());
}

private animate(): void {
  this.planet.rotation.y += 0.1 / 60;

  this.clouds.rotation.y = - this.planet.rotation.y;
  this.clouds.rotation.z = this.planet.rotation.y;
}
```

[download](#)



 SORINT_{lab}

ADD AN EXTERNAL MODEL 1/2

Add the function to load external models in
planet.service.ts snippet-7

```
private getGLTF(name: string): Promise<Object3D> {
  new GLTFLoader().load(
    `assets/${name}`,
    gltf => {
      const result = new Object3D();
      result.add(...gltf.scene.children);
      res(result);
    },
    undefined,
    rej
  );
}
```

[download](#)



 SORINT_{lab}

ADD AN EXTERNAL MODEL 2/2

Add a satellite to **planet.service.ts** snippet-8

```
private addSatellite(): Promise<void> {
  return this.getGLTF('satellite.gltf').then(object => {
    object.position.set(-20, 5, 80);
    object.rotation.set(-0.2, 1.4, 0);
    this.model.add(object);
  });
}
```

[download](#)



 SORINT_{lab}

BROWSER SUPPORT

Image by caniuse.com

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android *	Blackberry	Opera Mobile *	Chrome Android
6	12	56	63	9	50	9.2		4			
7	13	57	64	9.1	51	9.3		4.1			
8	14	58	65	10	52	10.2		4.3			
9	15	59	66	10.1	53	10.3		4.4		12	
10	16	60	67	11	54	11.2		4.4.4	7	12.1	
11	17	61	68	11.1	55	11.4	all	67	10	46	67
	18	62	69	12		12					
		63	70	TP							
			71								

download



 SORINT lab

ADVANCED TOPICS TO EXPLORE

- Shaders
- Custom geometries
- Animation system (AnimationMixer, AnimationClip, etc...)

[download](#)



 SORINT_{lab}

download

GO AND EXPAND THE UNIVERSE

THANKS!

In collaborazione con:



COMUNE DI BERGAMO

 SORINT *lab*