Escuela Profesional de
Ciencia de la Computación

Computer graphics

Three.js

MSc. Vicente Machaca Arceda

Universidad Nacional de San Agustín de Arequipa

April 27, 2021

Overview

vmachacaa@unsa.edu.pe

Table of Contents

vmachacaa@unsa.edu.pe

WebGL

WebGL (Web Graphics Library) is a JavaScript API for rendering high-performance interactive 3D and 2D graphics within any compatible web browser without the use of plug-ins.

vmachacaa@unsa.edu.pe

Table of Contents

vmachacaa@unsa.edu.pe

Three.js

Three.js is a cross-browser JavaScript library used to create and display animated 3D computer graphics in a web browser using **WebGL**.

vmachacaa@unsa.edu.pe

Three.js

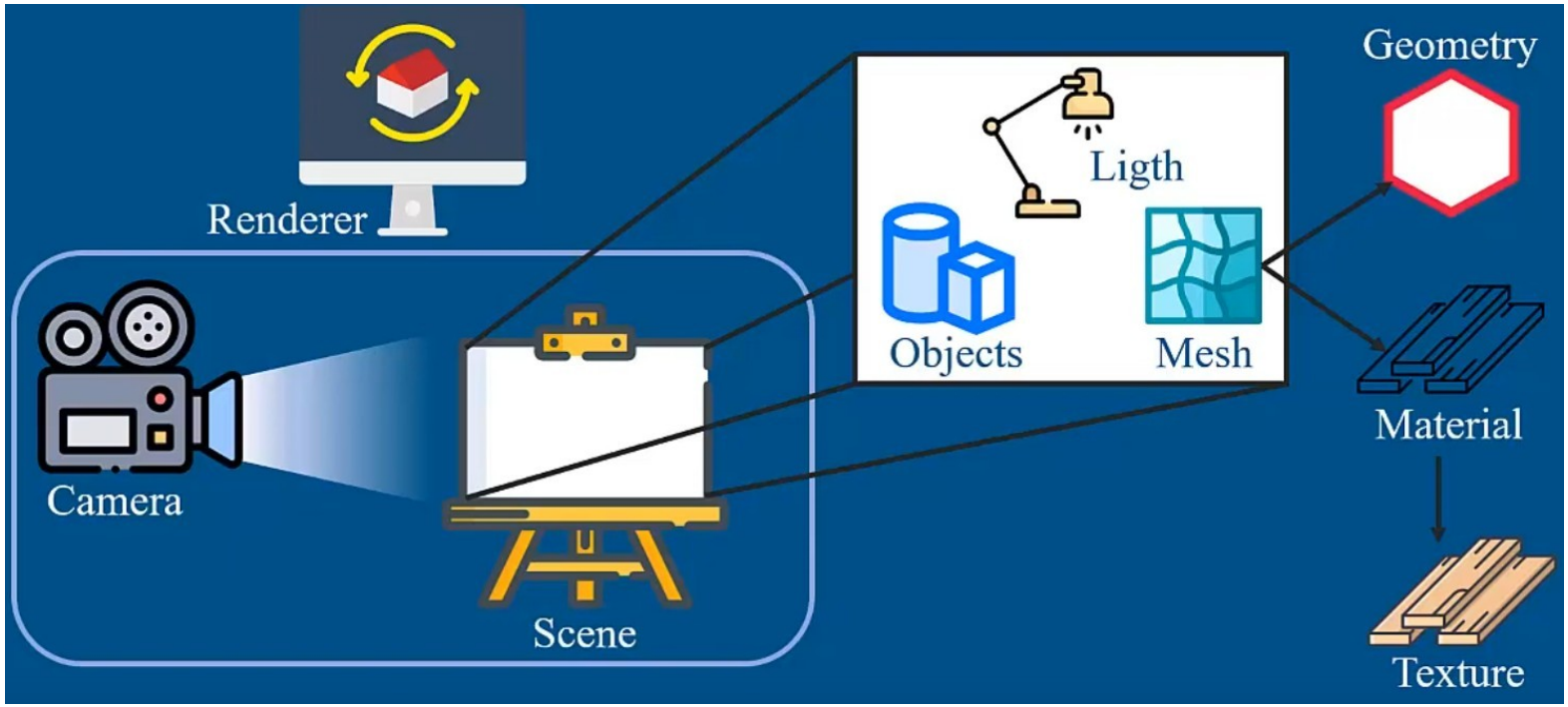

Figure: Three.js overview.

vmachacaa@unsa.edu.pe

Three.js

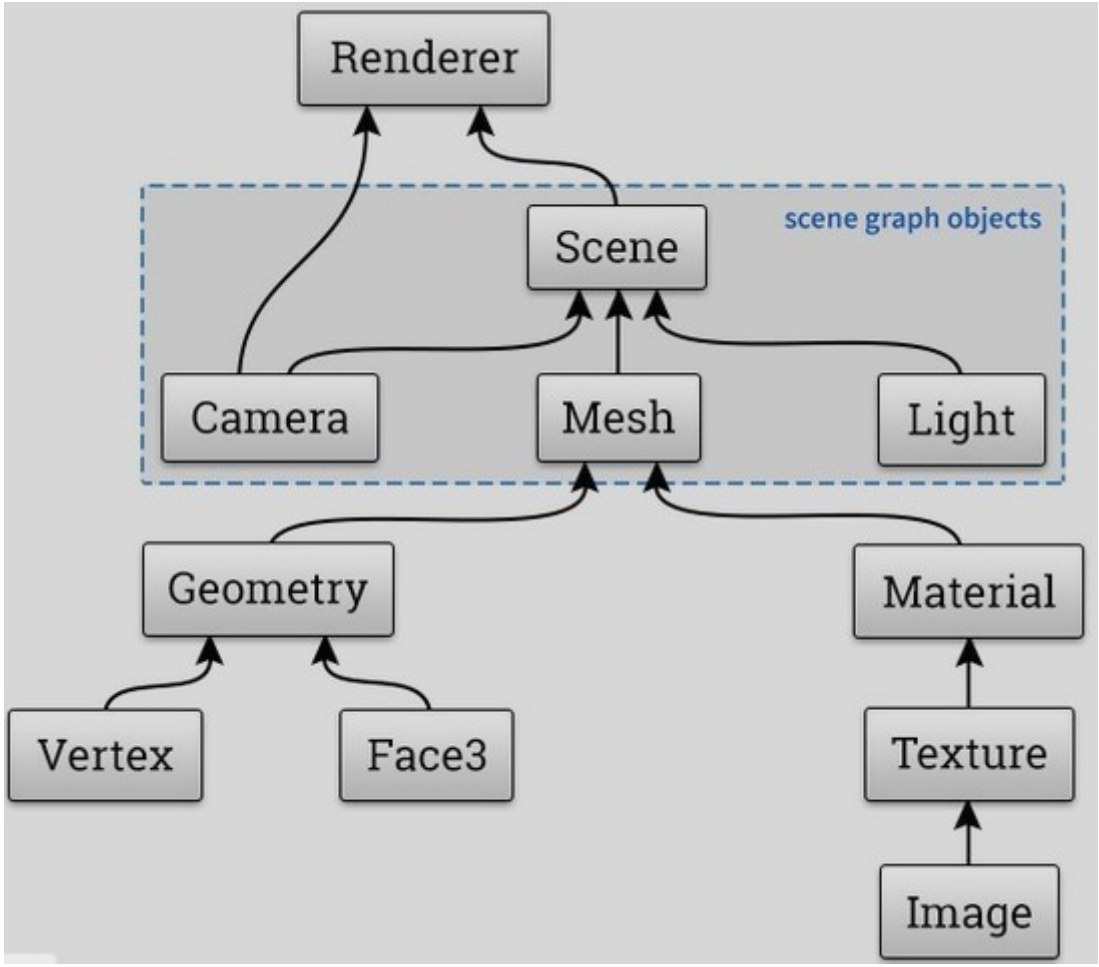

Figure: Three.js overview.

vmachacaa@unsa.edu.pe

Table of Contents

vmachacaa@unsa.edu.pe

Hello world

Include three.js from: Enlace

```
1  <script src="three.min.js"> </script>
```

vmachacaa@unsa.edu.pe

Hello cube

Mesh and Camera

A *Mesh* object is compose by a *Geometry* and *Material*.

```
var geometry = new THREE.BoxGeometry();
var material = new THREE.MeshBasicMaterial(
                            {color:0x780000}
                        );
var cube = new THREE.Mesh( geometry, material );
```

Camera

```
var camera = new THREE.PerspectiveCamera(
75,
window.innerWidth/window.innerHeight
);
camera.position.z = 5;
```

vmachacaa@unsa.edu.pe

Hello cube

Scene and Renderer

The *Scene* just takes the *cube*.

```
1 var scene = new THREE.Scene();
scene.backgroundColor = new THREE.Color(0x000000);
3 scene.add(cube);
```

Renderer takes the *scene* and the *camera*

```
1 var renderer = new THREE.WebGLRenderer();
renderer.setSize( window.innerWidth, window.innerHeight );
3 document.body.appendChild( renderer.domElement );
renderer.render( scene, camera );
```

vmachacaa@unsa.edu.pe

Hello cube

Animation

Animation

```
var animate = function(){
  requestAnimationFrame(animate);
  cube.rotation.x += 0.01;
  cube.rotation.y += 0.01;
  renderer.render( scene, camera );
}
animate();

```

vmachacaa@unsa.edu.pe

Hello cube

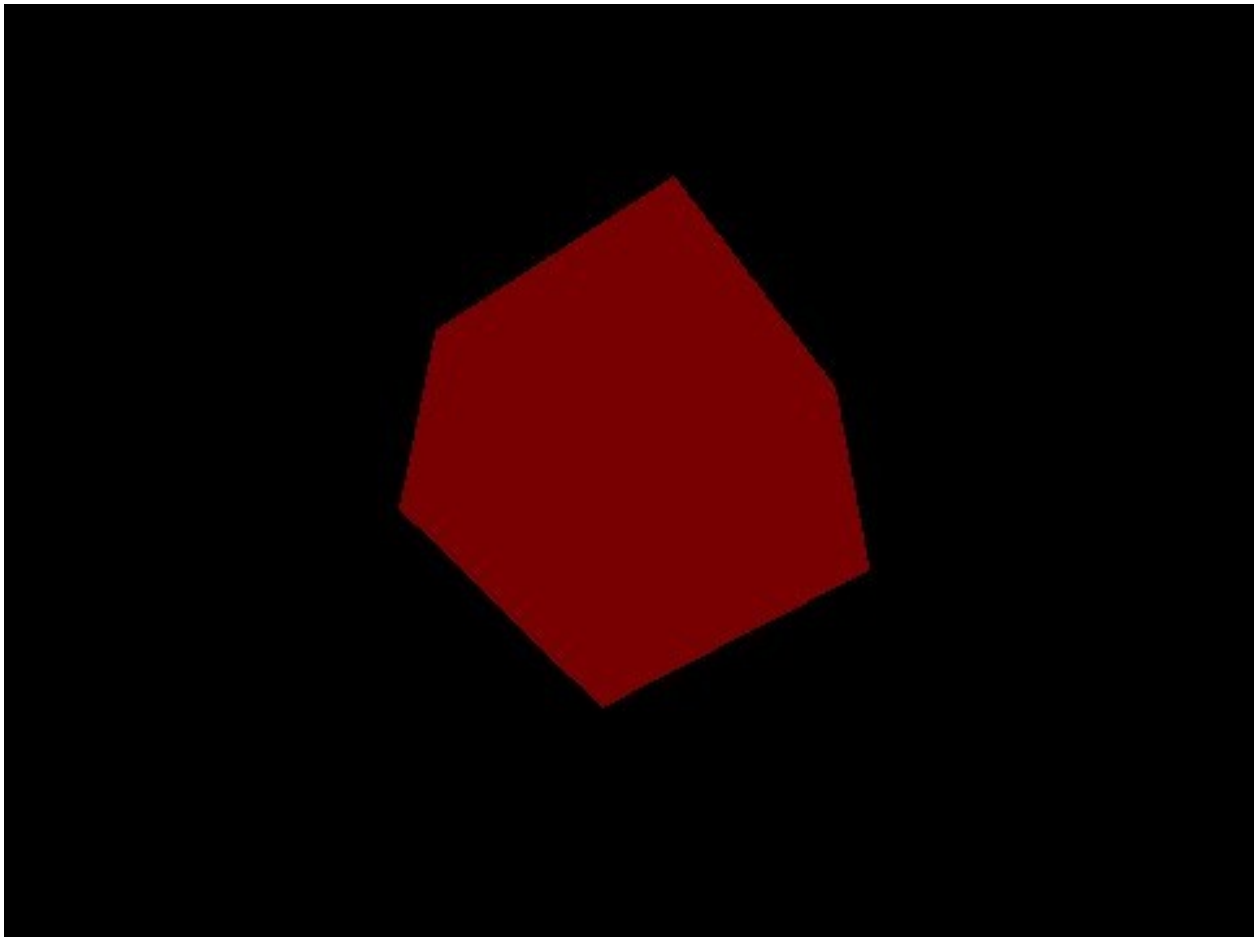

Figure: Three.js cube example.

vmachacaa@unsa.edu.pe

Table of Contents

vmachacaa@unsa.edu.pe

Camera

Example: Enlace

Perspective

More realistic and similar to real world.

Orthographic

Old, it was used in old games.

vmachacaa@unsa.edu.pe

Camera

Perspective camera

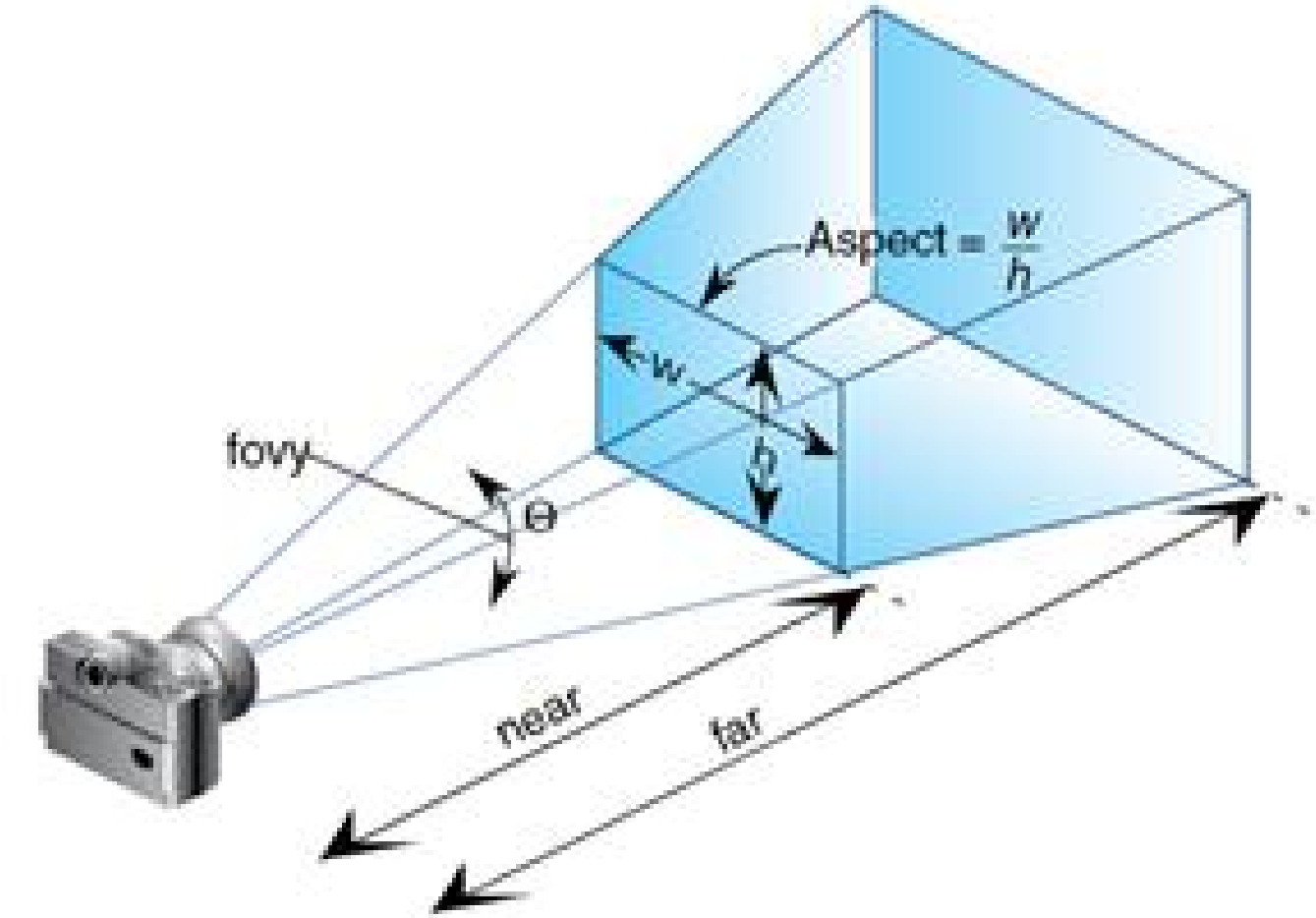

Figure: Perspective camera.

vmachacaa@unsa.edu.pe

Camera

Perspective camera

```
var camera_1 = new THREE.PerspectiveCamera(
75, // fov
window.innerWidth/window.innerHeight, // aspect
0.1, // near
2000 // far
);
var camera = camera_1;
```

vmachacaa@unsa.edu.pe

Camera

Orthographic camera

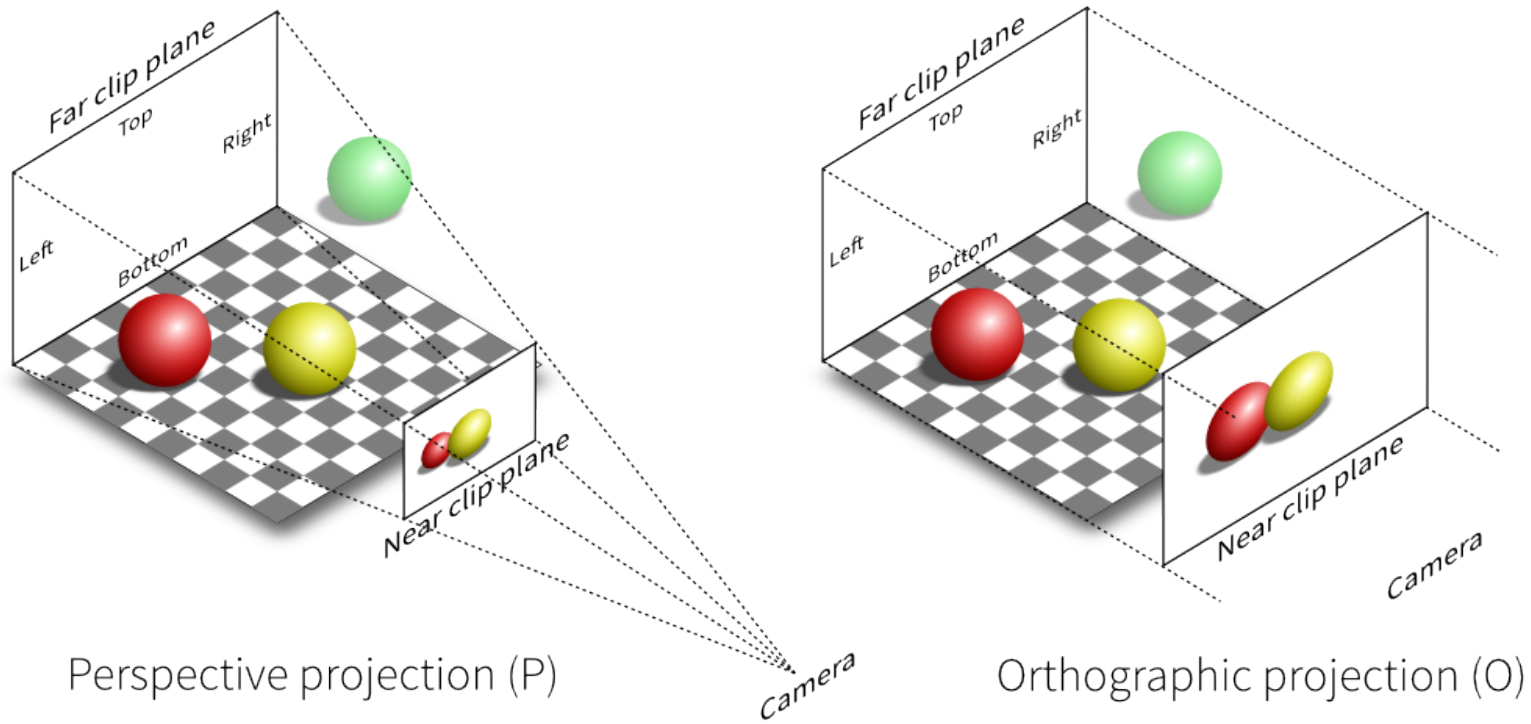

Figure: Perspective camera.

vmachacaa@unsa.edu.pe

Camera

Orthographic camera

```
var camera_2 = new THREE.OrthographicCamera(
5, //left
-5, //right
5, //top
-5, //bottom
3, // near
10 // far
);
var camera = camera_2;
```

vmachacaa@unsa.edu.pe

Table of Contents

vmachacaa@unsa.edu.pe

Mouse events

OrbitControls

Include OrbitControls from: Enlace

```
<script src="OrbitControls.js"> </script>
```

Add the following:

```
var controls = new THREE.OrbitControls(
    camera,
    renderer.domElement
);

```

vmachacaa@unsa.edu.pe

Table of Contents

vmachacaa@unsa.edu.pe

Geometrics

Similar to BoxGeometry,

```
var geometry = new THREE.BoxGeometry();

```

There are other Geometrics:

```
var geometry = new THREE.CircleGeometry(...);
var geometry = new THREE.ConeGeometry(...);
var geometry = new THREE.CylinderGeometry(...);
var geometry = new THREE.PlaneGeometry(...);
var geometry = new THREE.TetrahedronGeometry(...);
var geometry = new THREE.SphereGeometry(...);
var geometry = new THREE.TorusGeometry(...);
```

See example: *4_geometrics.html*

vmachacaa@unsa.edu.pe

Geometrics

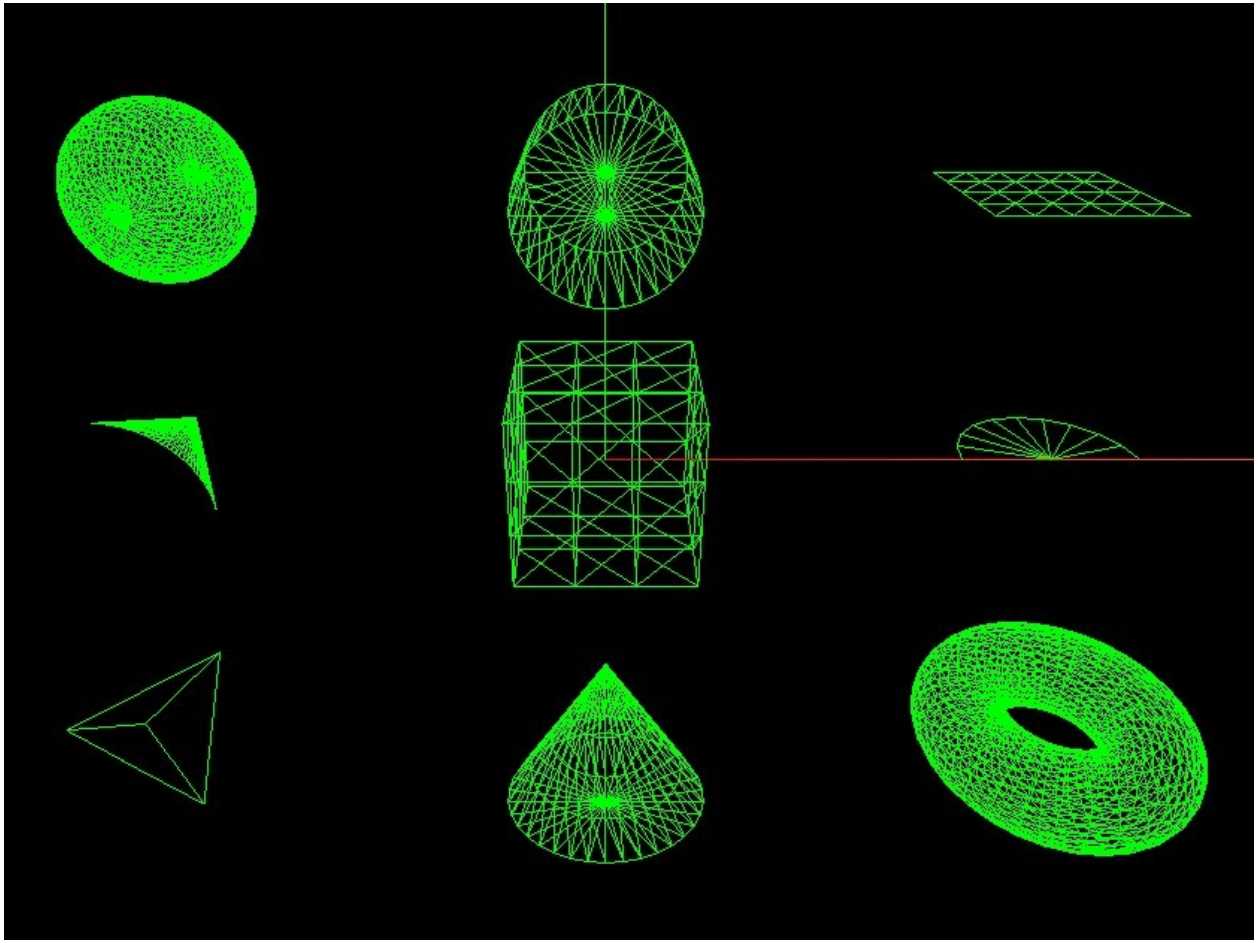

Figure: Geometrics example.

vmachacaa@unsa.edu.pe

Table of Contents

vmachacaa@unsa.edu.pe

Lights

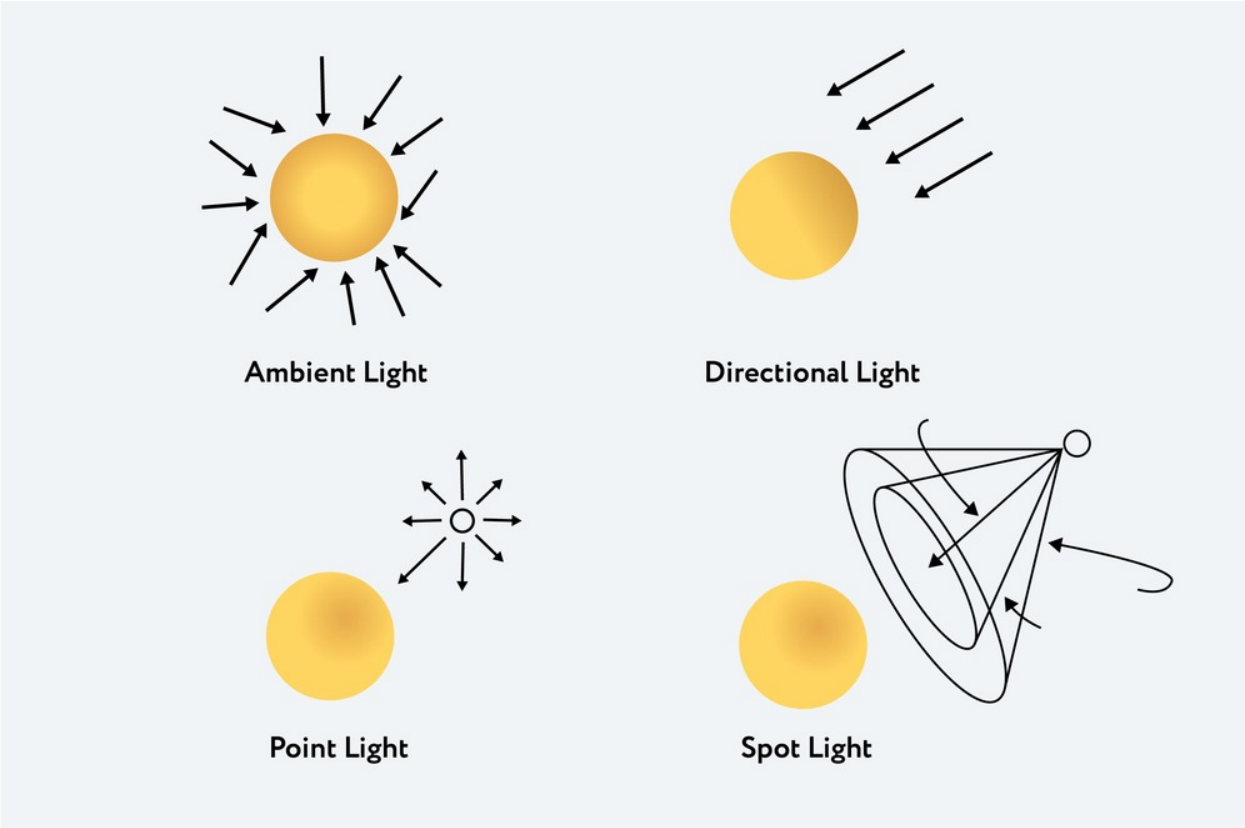

Figure: Lights in Three.js.

vmachacaa@unsa.edu.pe

Lights

```
// Ambient light
const color = 0xFFFFFF;
const intensity = 0;
const light = new THREE.AmbientLight(color, intensity);
scene.add(light);
```

```
// Directional light
const color = 0xFFFFFF;
const intensity = 1;
const light = new THREE.DirectionalLight(color, intensity
  );
light.position.set(0, 10, 0);
light.target.position.set(-5, 0, 0);
scene.add(light);
scene.add(light.target);
```

See example: *5_lights.html*

vmachacaa@unsa.edu.pe

Lights

Realistic light

In order to get a more realistic light, you could try two lights, for example you could use hemispehre light with directional light. See example: here.

vmachacaa@unsa.edu.pe

Questions?

vmachacaa@unsa.edu.pe