Visualizações 3D para a web com WebGL

Introdução ao Three.js

Óscar Oliveira (oao@estg.ipp.pt)

P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Conteúdos

- Introdução
- *Renderer*
- *Mesh*
  - Geometrias
  - Materiais
- Luzes
- Animação
- Conclusão

Introdução

Imagem[1]

WebGL (Web Graphics Library) é uma **API** de JavaScript para a **renderização de gráficos 2D e 3D interativos** dentro de qualquer **navegador da Web** compatível **sem a utilização de plug-ins.**[1]

three.js

Three.js é um **motor WebGL** baseado em **JavaScript** que pode executar jogos movidos com GPU ou outros aplicativos gráficos diretamente do navegador. A biblioteca three.js fornece várias **funções e APIs** para desenhar cenas 3D em seu navegador.[1]

three.js

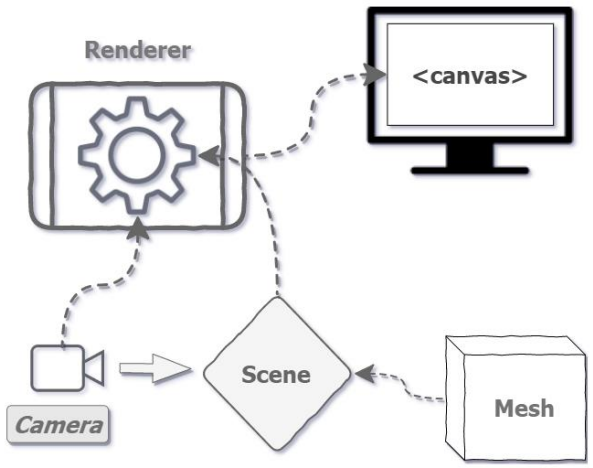
https://threejs.org/

https://github.com/mrdoob/three.js

/build

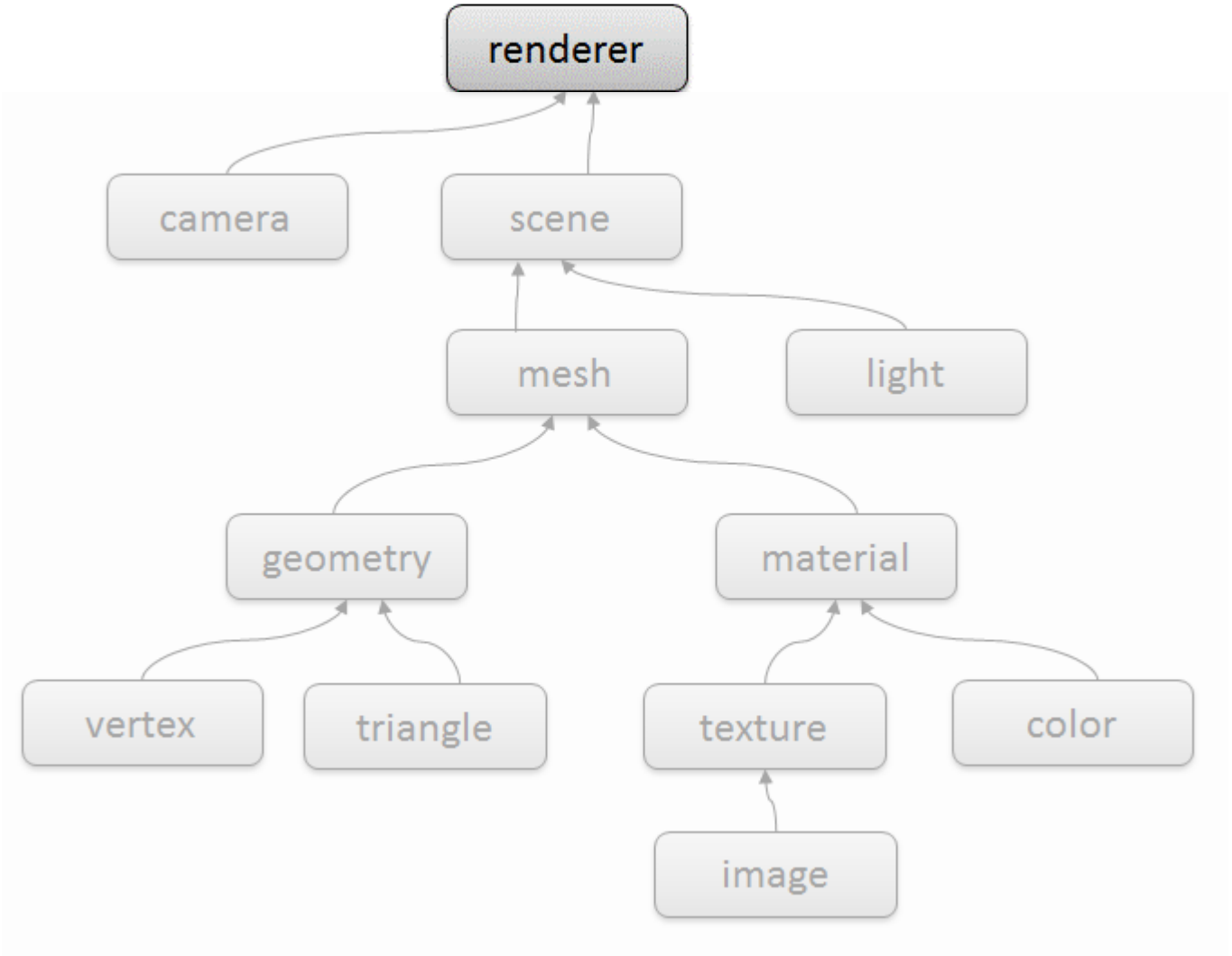
\exemplos\1-introducao

Renderer



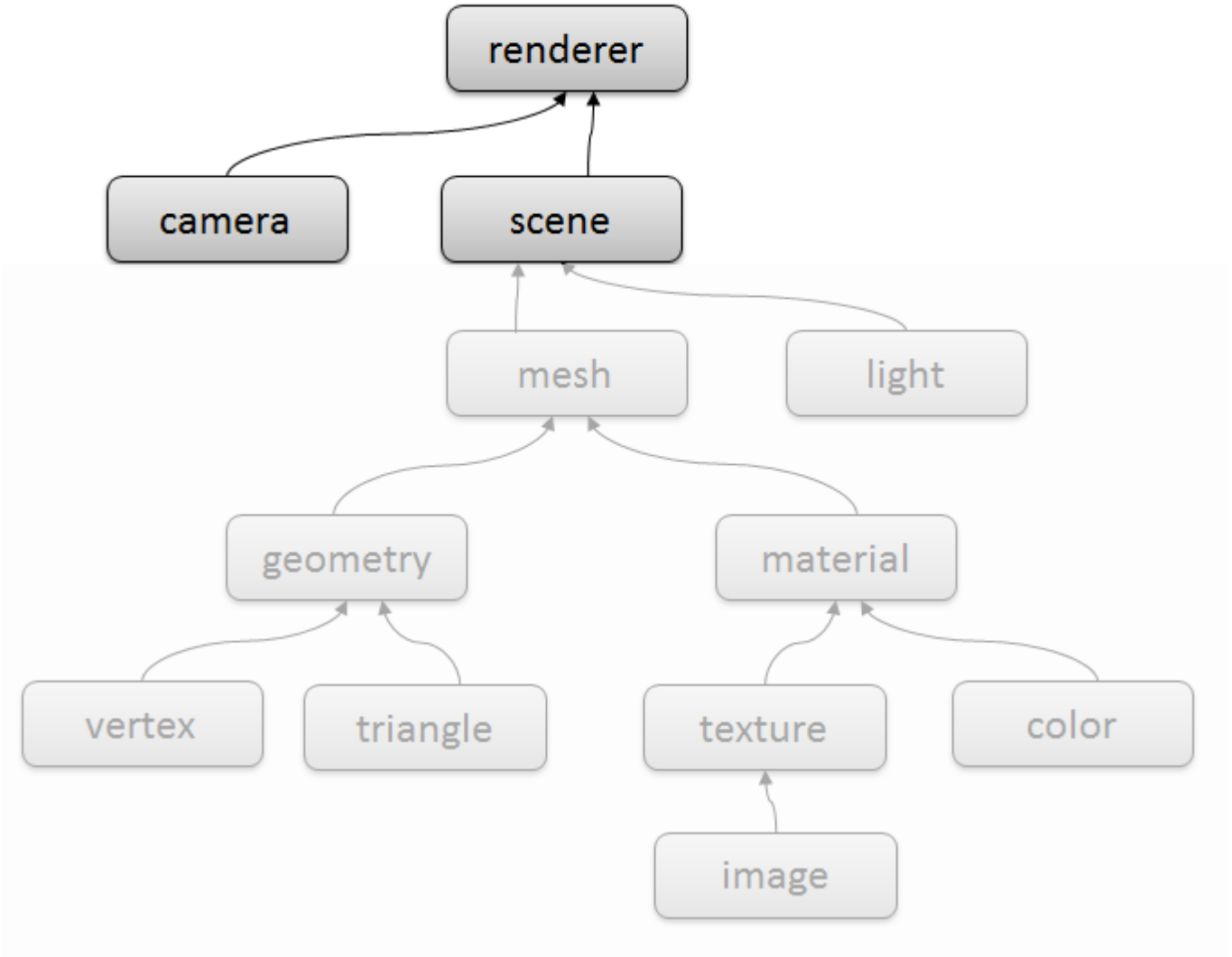
Imagem[1]

Objetos



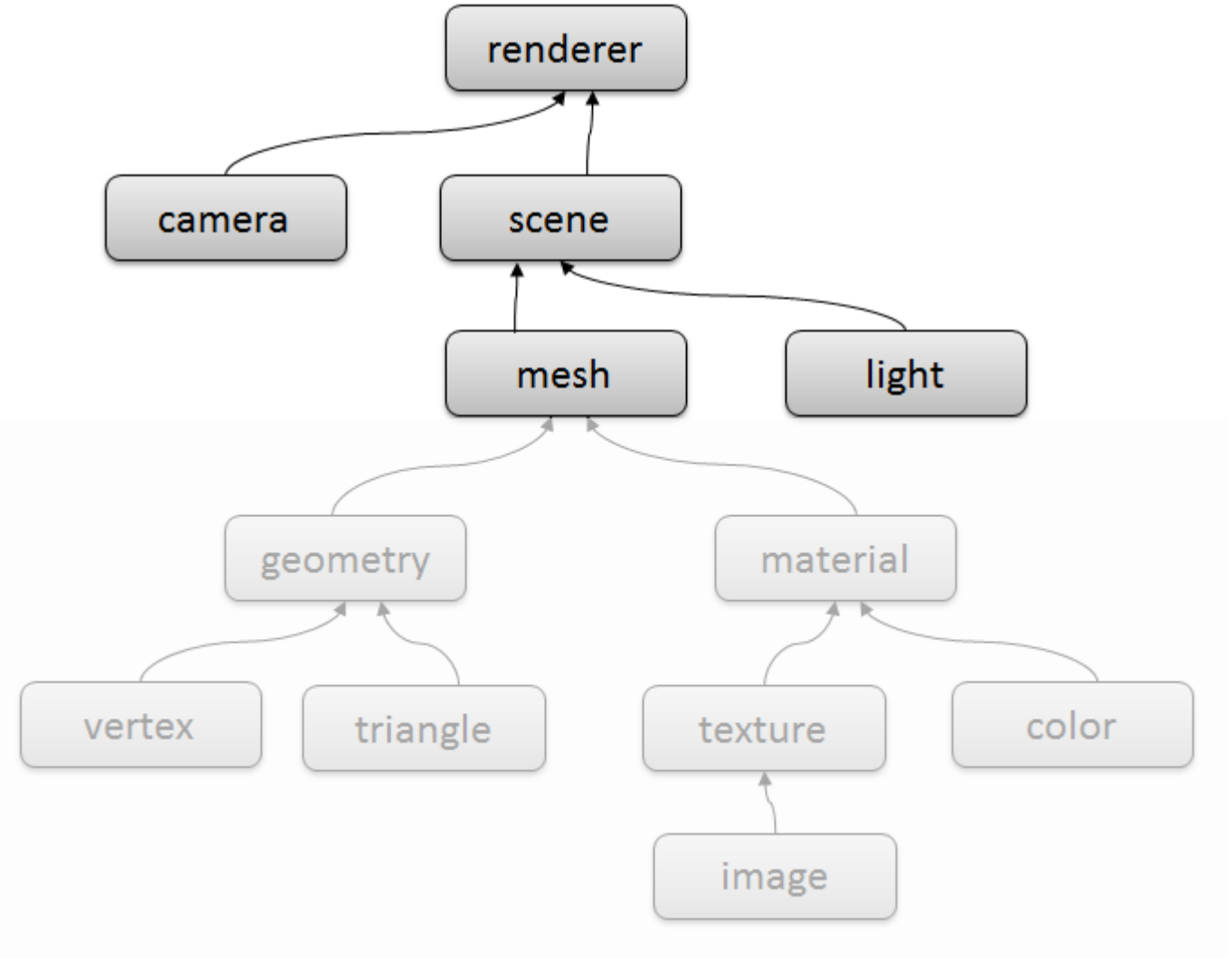
Imagem[1]

Objetos



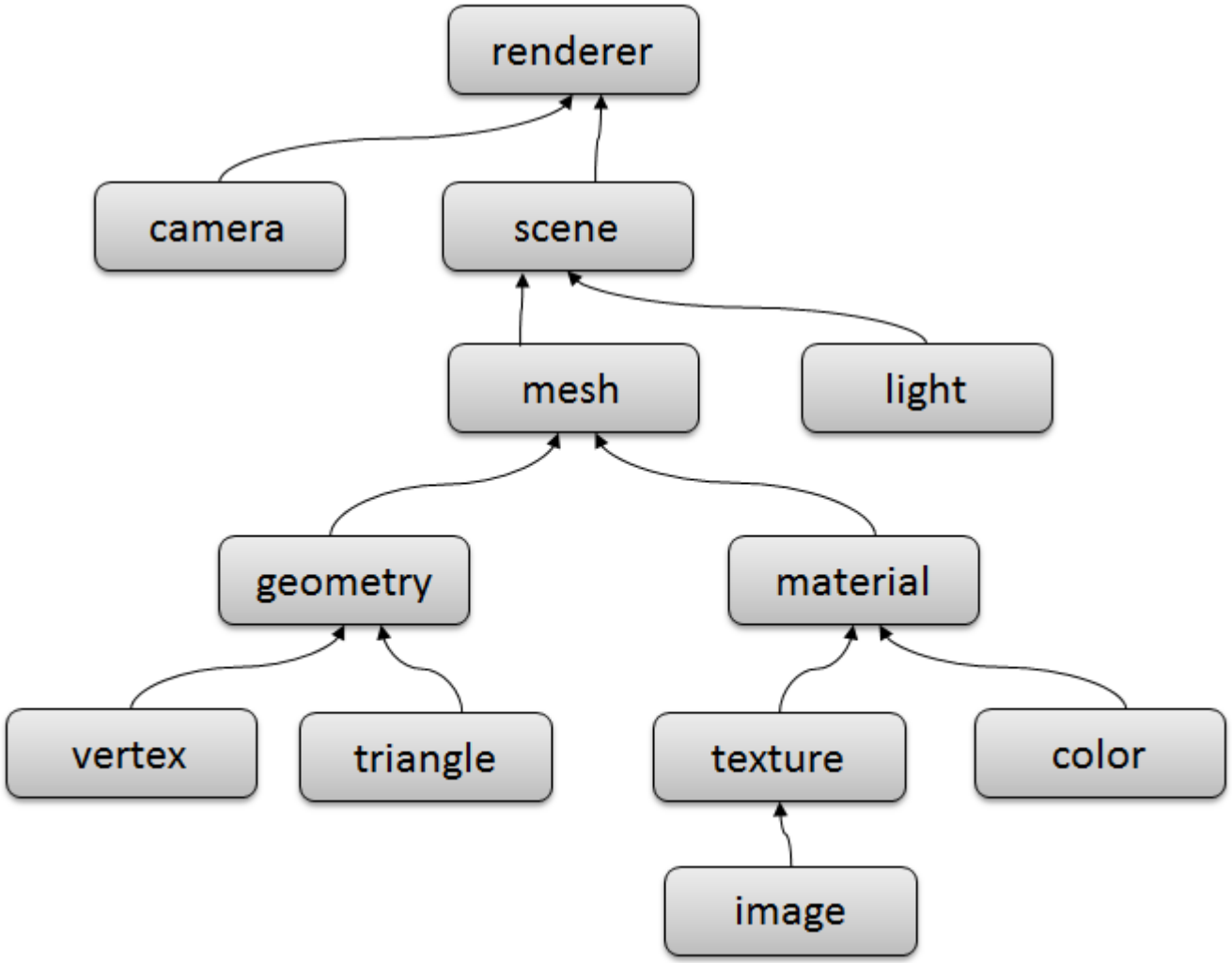
Imagem[1]

Objetos



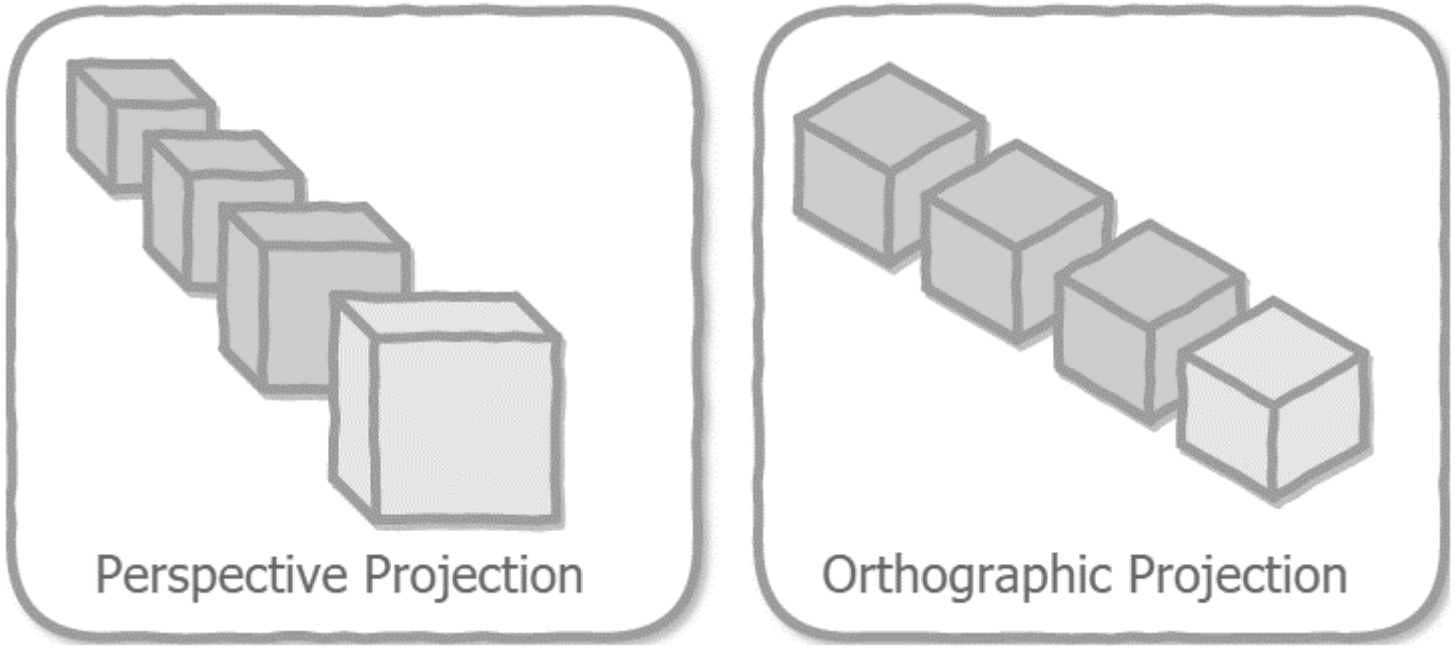
Imagem[1]

Objetos



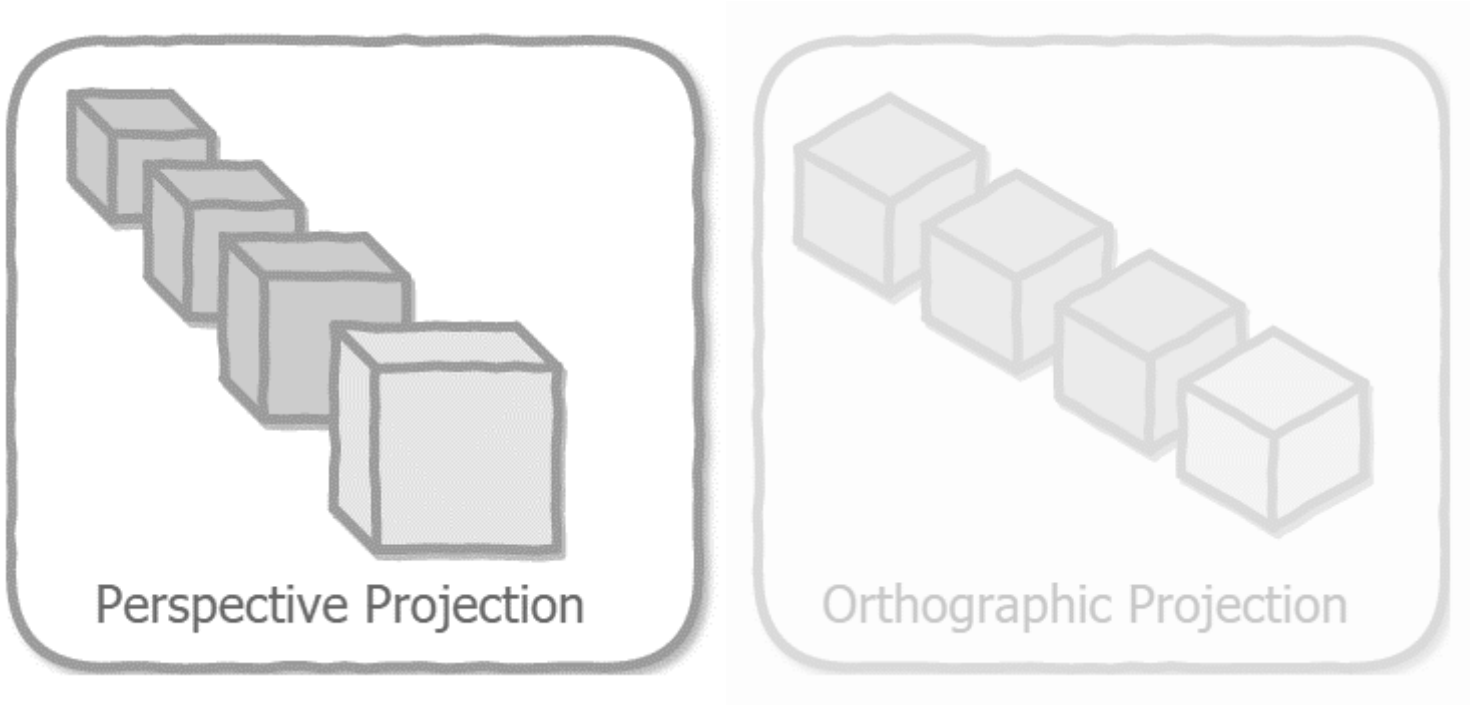
Imagem[1]

Camera



Imagem[1]

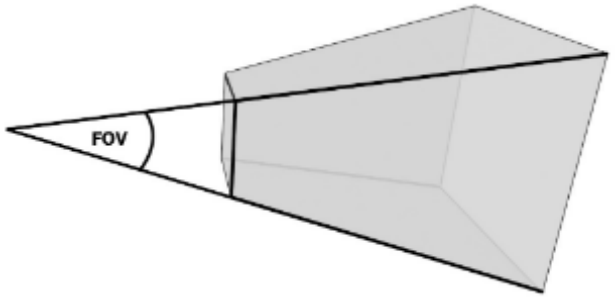
Camera



Imagem[1]

Camera

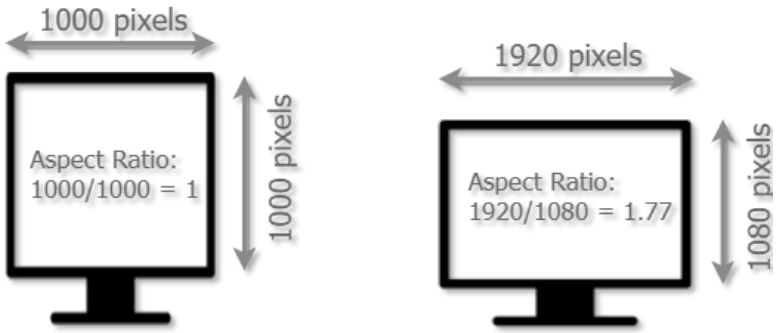
```
new THREE.PerspectiveCamera(fov, aspect, near, far);
```



Imagem[1]

Camera

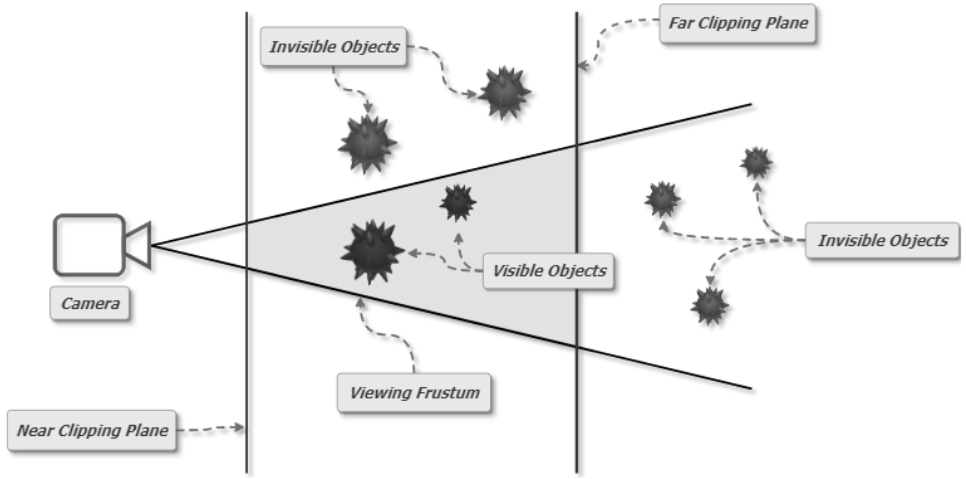
```
new THREE.PerspectiveCamera(fov, aspect, near, far);
```



Imagem[1]

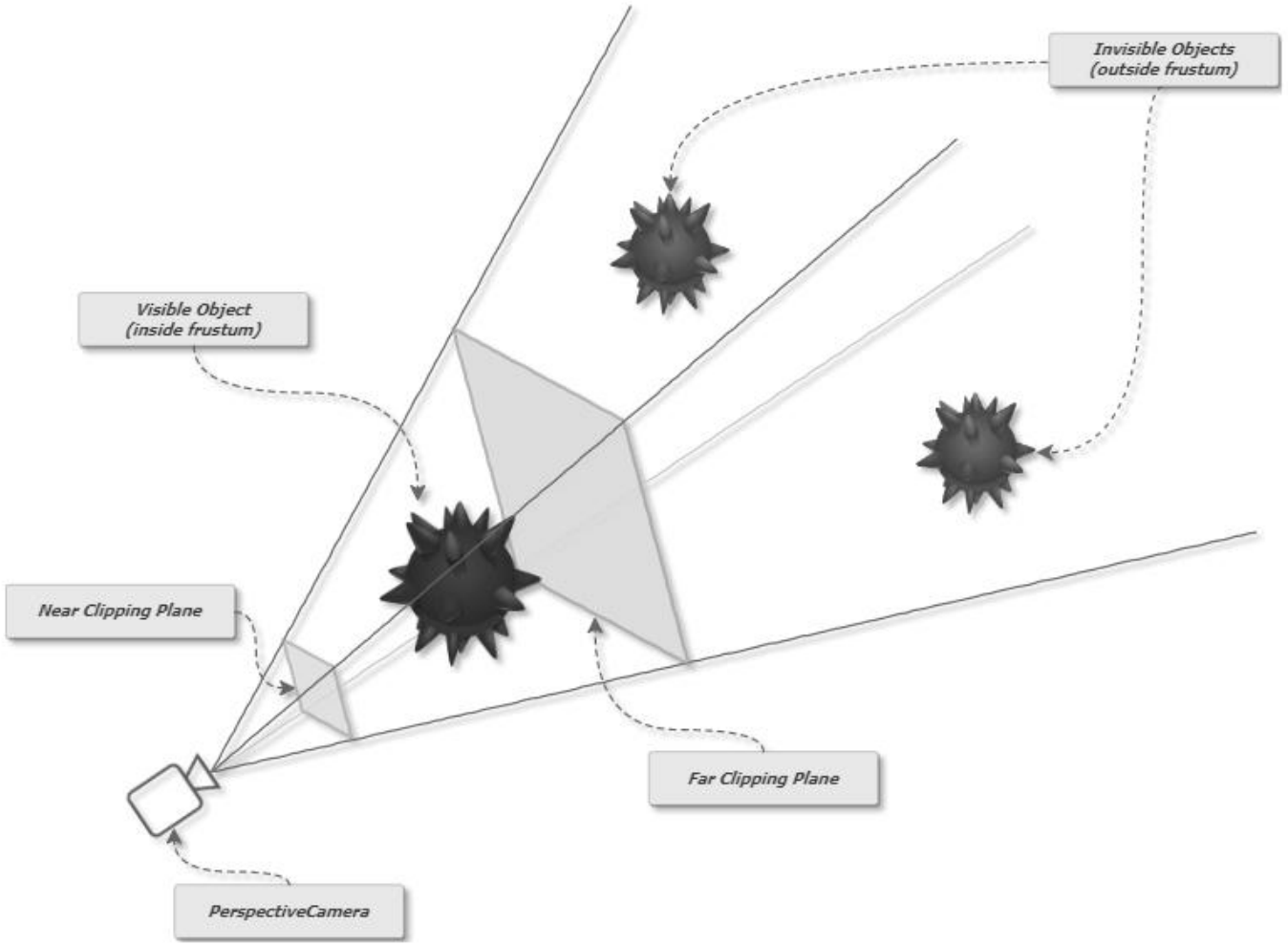
Camera

```
new THREE.PerspectiveCamera(fov, aspect, near, far);
```



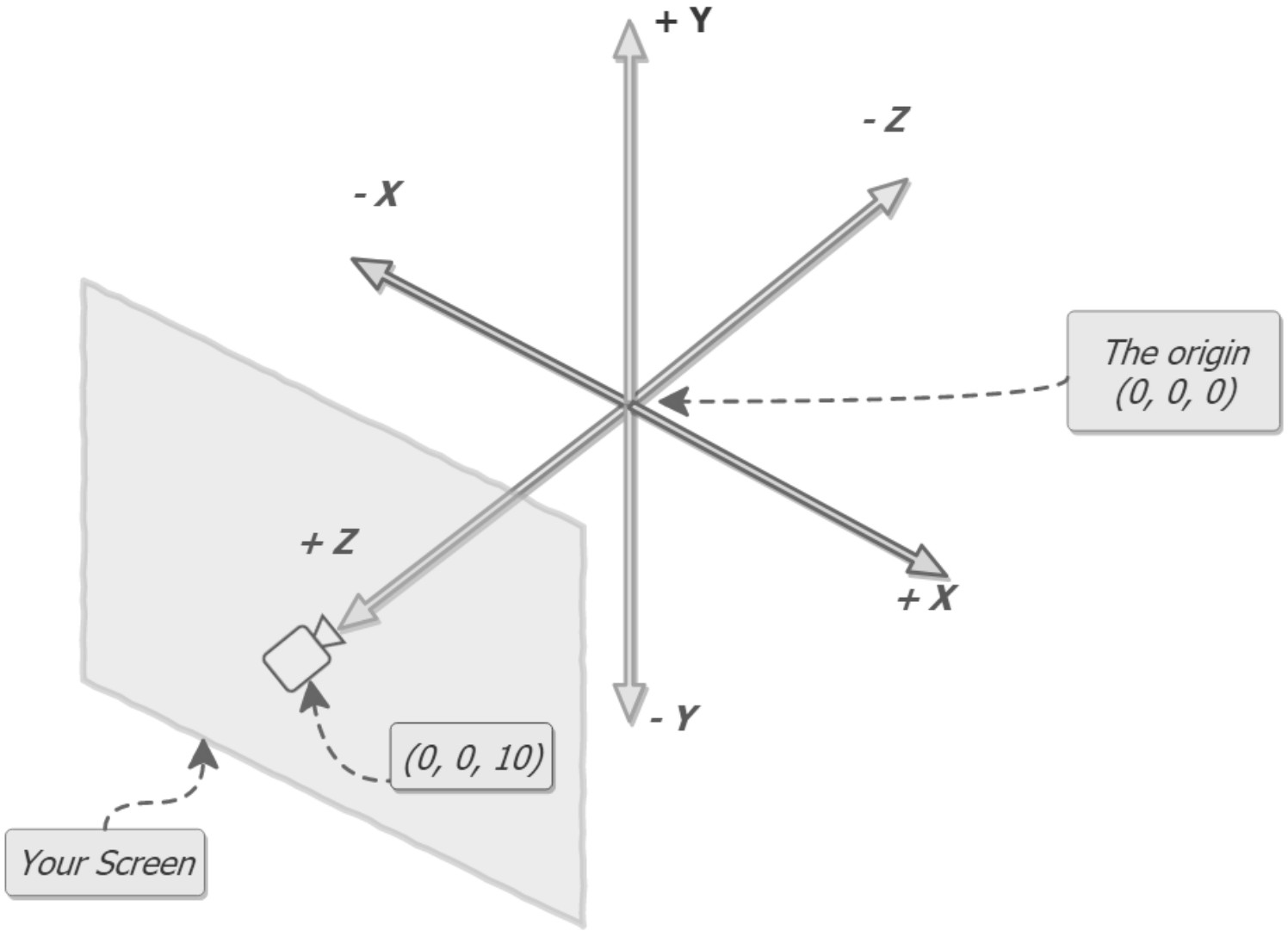
Imagem[1]

Camera



Imagem[1]

Sistema de coordenadas



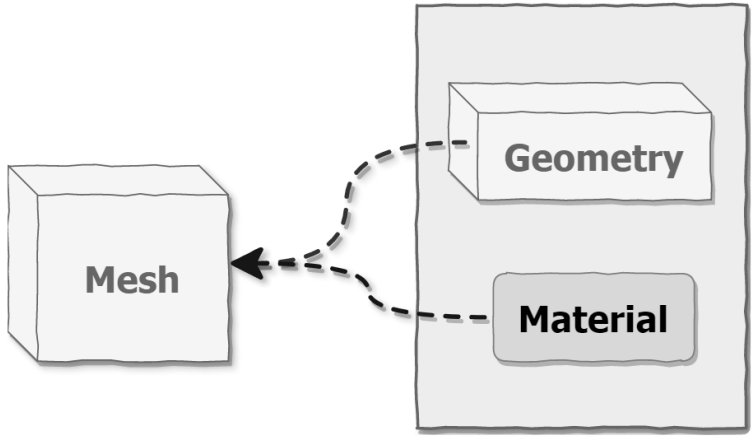
Imagem[1]

Renderer

```
camera = new THREE.PerspectiveCamera(fov, ratio, near, far);

scene = new THREE.Scene();

renderer = new THREE.WebGLRenderer({ antialias: true });
renderer.setSize(width, height);

renderer.render(scene, camera);
```

\exemplos\2-renderer

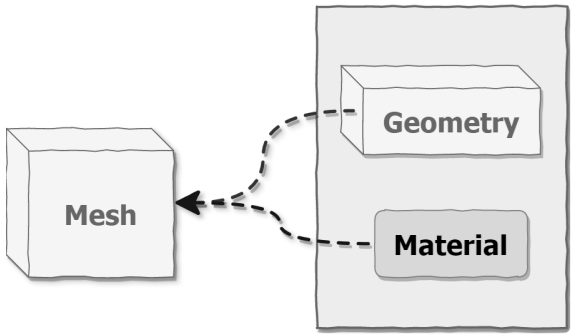
Mesh



Imagem[1]

Mesh

```
const geometry = new THREE.BoxBufferGeometry(2, 2, 2);
const material = new THREE.MeshNormalMaterial();
const mesh = new THREE.Mesh(geometry, material);
```

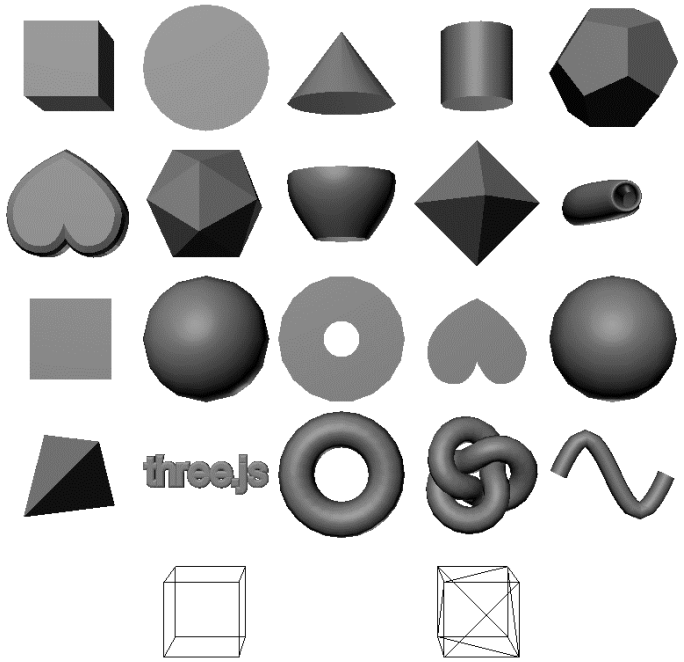


Imagem[1]

Geometrias

Primitivas

- **BoxBufferGeometry**, BoxGeometry
- **CircleBufferGeometry**, CircleGeometry
- **ConeBufferGeometry**, ConeGeometry
- **CylinderBufferGeometry**, CylinderGeometry
- **DodecahedronBufferGeometry**, DodecahedronGeometry
- ExtrudeBufferGeometry, ExtrudeGeometry
- **IcosahedronBufferGeometry**, IcosahedronGeometry
- LatheBufferGeometry, LatheGeometry
- **OctahedronBufferGeometry**, OctahedronGeometry
- ParametricBufferGeometry, ParametricGeometry
- PlaneBufferGeometry, PlaneGeometry
- PolyhedronBufferGeometry, PolyhedronGeometry
- RingBufferGeometry, RingGeometry
- ShapeBufferGeometry, ShapeGeometry
- **SphereBufferGeometry**, SphereGeometry
- **TetrahedronBufferGeometry**, TetrahedronGeometry
- TextBufferGeometry, TextGeometry
- **TorusBufferGeometry**, TorusGeometry
- **TorusKnotBufferGeometry**, TorusKnotGeometry
- TubeBufferGeometry, TubeGeometry
- EdgesGeometry
- WireframeGeometry

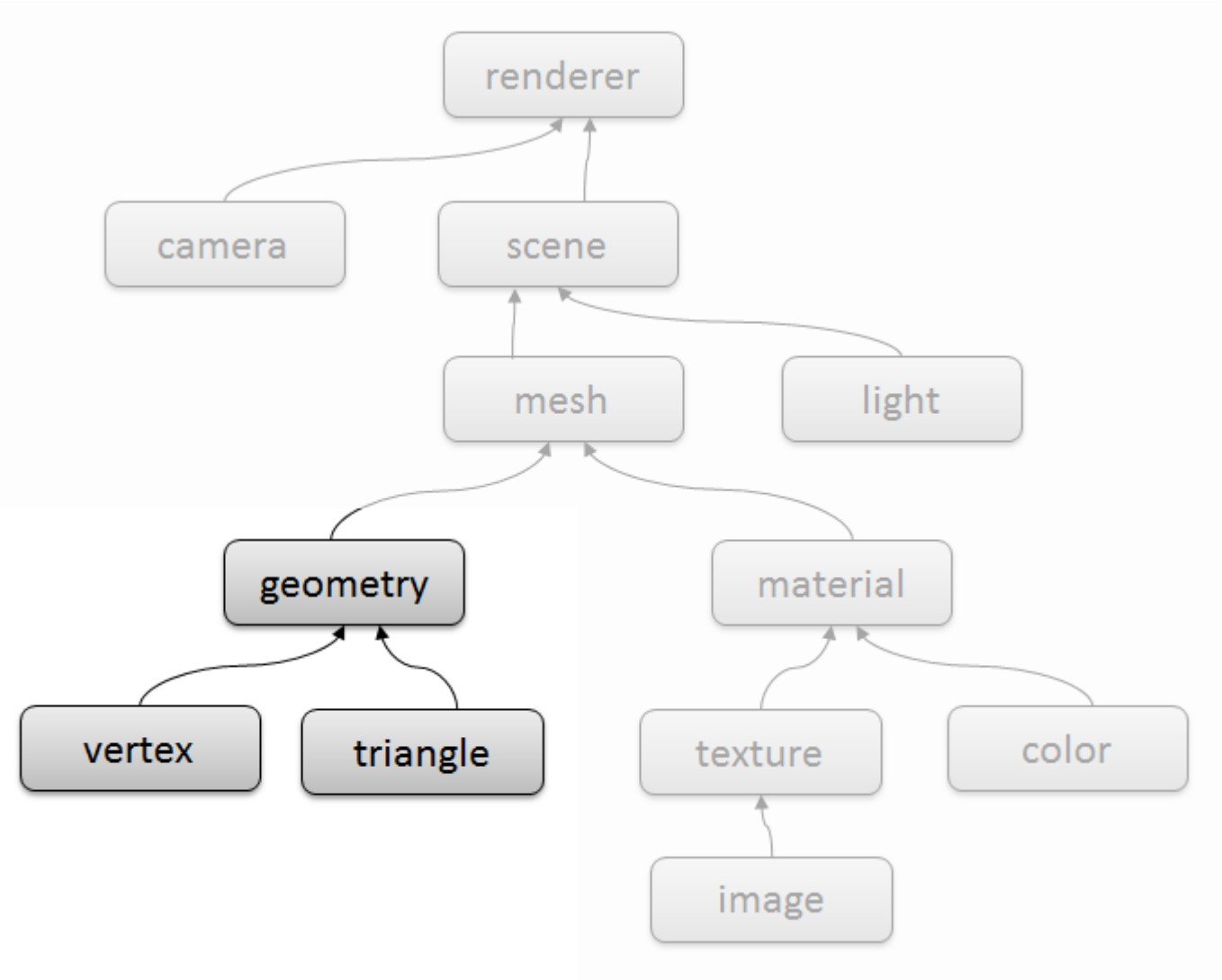


Imagem[1]

\exemplos\3-geometrias [1-6]

Geometrias

Personalizadas

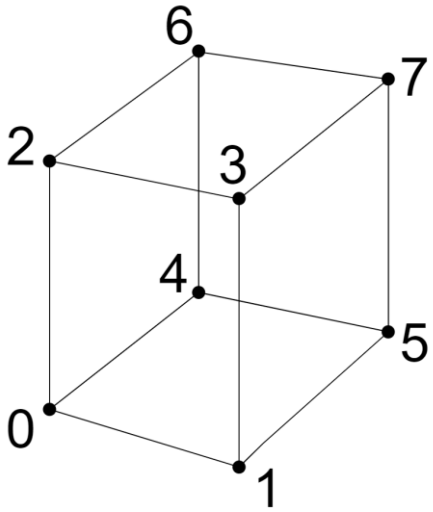


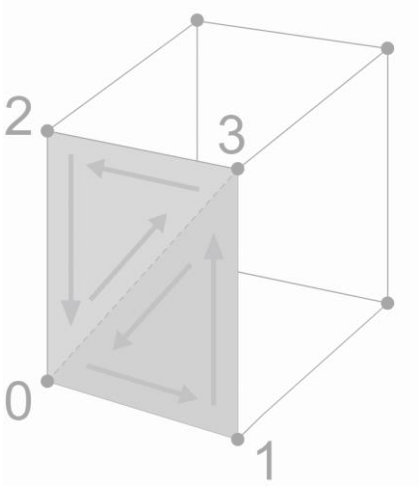
Imagem[1]

Geometrias

Personalizadas

```
const geometry = new THREE.Geometry();
geometry.vertices.push(
    new THREE.Vector3(-1, -1,  1),  // 0
    new THREE.Vector3( 1, -1,  1),  // 1
    new THREE.Vector3(-1,  1,  1),  // 2
    new THREE.Vector3( 1,  1,  1),  // 3
    new THREE.Vector3(-1, -1, -1),  // 4
    new THREE.Vector3( 1, -1, -1),  // 5
    new THREE.Vector3(-1,  1, -1),  // 6
    new THREE.Vector3( 1,  1, -1),  // 7
);
geometry.faces.push(
    // front
    new THREE.Face3(0, 3, 2), new THREE.Face3(0, 1, 3),
    // right
    new THREE.Face3(1, 7, 3), new THREE.Face3(1, 5, 7),
    // back
    new THREE.Face3(5, 6, 7), new THREE.Face3(5, 4, 6),
    // left
    new THREE.Face3(4, 2, 6), new THREE.Face3(4, 0, 2),
    // top
    new THREE.Face3(2, 7, 6), new THREE.Face3(2, 3, 7),
    // bottom
    new THREE.Face3(4, 1, 0), new THREE.Face3(4, 5, 1),
);
```



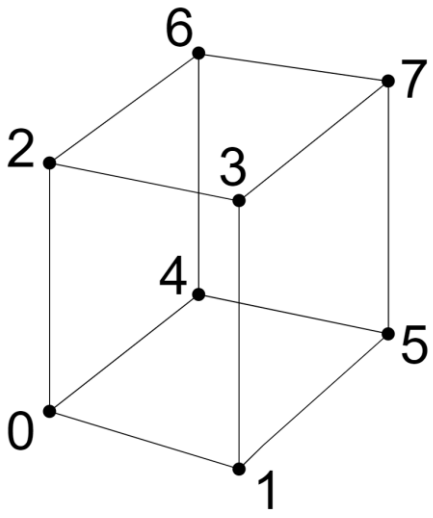


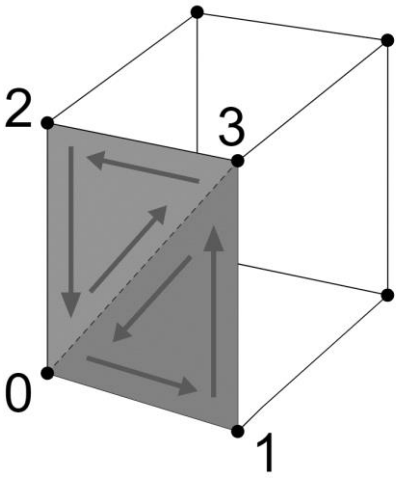
Imagem[1]

Geometrias

Personalizadas

```
const geometry = new THREE.Geometry();
geometry.vertices.push(
    new THREE.Vector3(-1, -1,  1),  // 0
    new THREE.Vector3( 1, -1,  1),  // 1
    new THREE.Vector3(-1,  1,  1),  // 2
    new THREE.Vector3( 1,  1,  1),  // 3
    new THREE.Vector3(-1, -1, -1),  // 4
    new THREE.Vector3( 1, -1, -1),  // 5
    new THREE.Vector3(-1,  1, -1),  // 6
    new THREE.Vector3( 1,  1, -1),  // 7
);
geometry.faces.push(
    // front
    new THREE.Face3(0, 3, 2), new THREE.Face3(0, 1, 3),
    // right
    new THREE.Face3(1, 7, 3), new THREE.Face3(1, 5, 7),
    // back
    new THREE.Face3(5, 6, 7), new THREE.Face3(5, 4, 6),
    // left
    new THREE.Face3(4, 2, 6), new THREE.Face3(4, 0, 2),
    // top
    new THREE.Face3(2, 7, 6), new THREE.Face3(2, 3, 7),
    // bottom
    new THREE.Face3(4, 1, 0), new THREE.Face3(4, 5, 1),
);
```



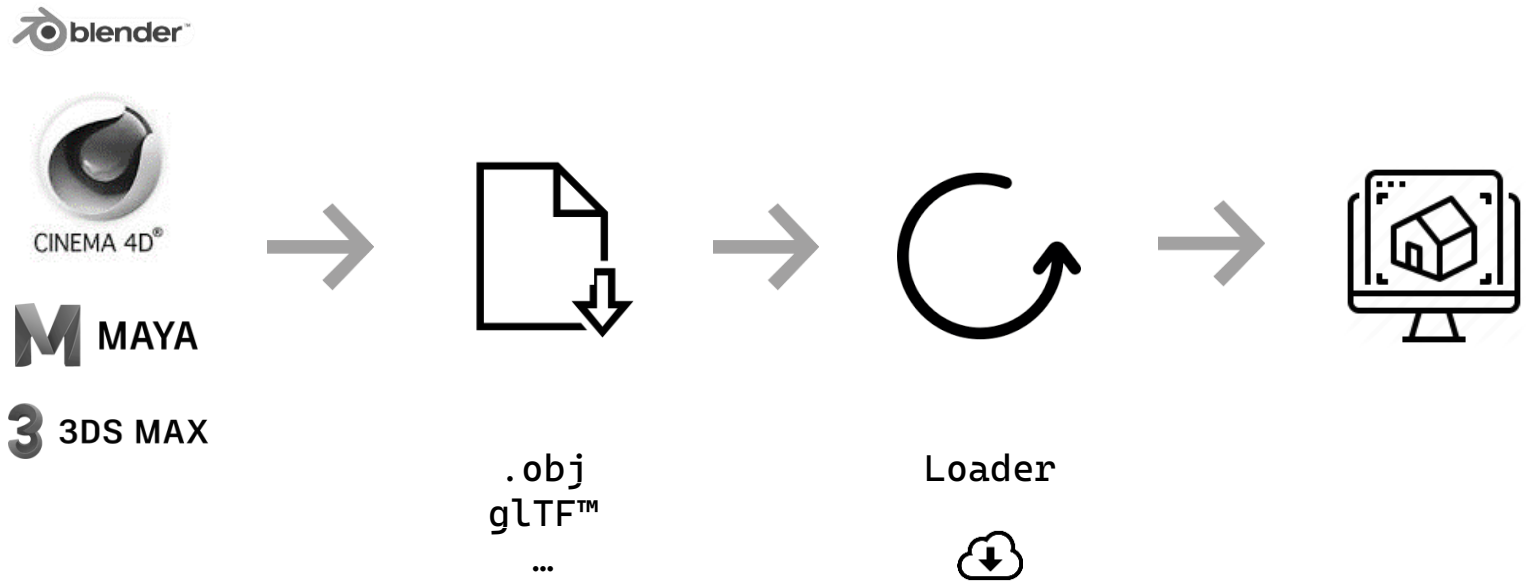


\exemplos\3-geometrias [7]

Imagem[1]

Geometrias

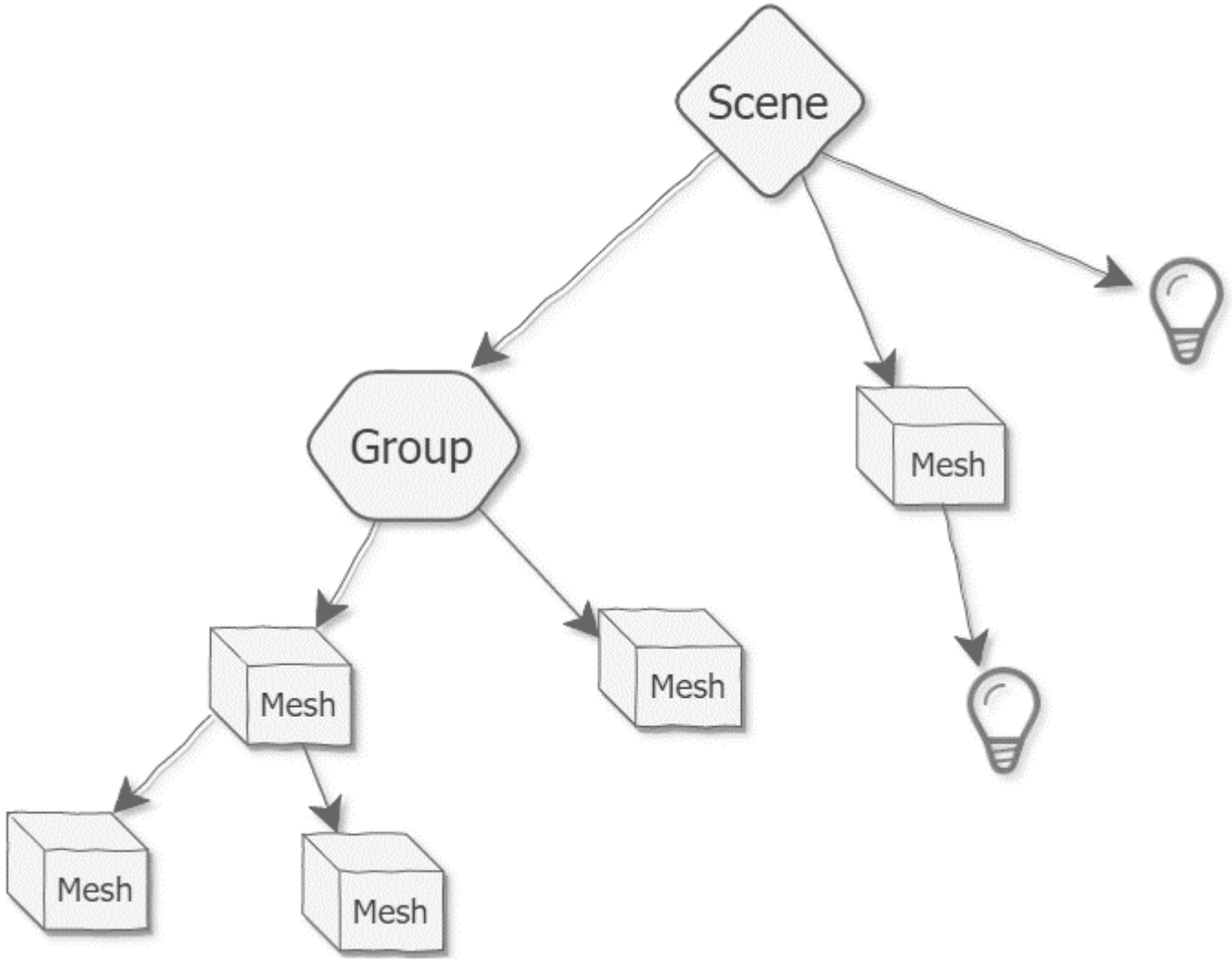
Importar modelos



\exemplos\3-geometrias [8]

Geometrias

Grupos



\exemplos\3-geometrias [9]

Imagem [1]

Material

```
const material = new THREE.MeshPhongMaterial({
  color: 0xFF0000,
  flatShading: true,
});

const material = new THREE.MeshPhongMaterial();
material.color.setHSL(0, 1, .5);
material.flatShading = true;
```

Imagem[1]

THREE.Color

```
m1 = new THREE.MeshBasicMaterial({color: 0xFF0000});
m2 = new THREE.MeshBasicMaterial({color: 'red'});
m3 = new THREE.MeshBasicMaterial({color: '#F00'});
m4 = new THREE.MeshBasicMaterial({color: 'rgb(255,0,0)'});
m5 = new THREE.MeshBasicMaterial({color: 'hsl(0,100%,50%)’});

material.color.set(0x00FFFF);   // same as CSS's #RRGGBB style
material.color.set(cssString);  // any CSS color, e.g.,
                                // 'purple’, '#F32',
                                // 'rgb(255, 127, 64)',
                                // 'hsl(180, 50%, 25%)'
material.color.set(someColor);  // some other THREE.Color
material.color.setHSL(h, s, l); // where h, s, and l are 0 to 1
material.color.setRGB(r, g, b); // where r, g, and b are 0 to 1
```

Color names

Material

+ RAPIDO

- REALISTICO

MeshBasicMaterial(*)

MeshLambertMaterial

MeshPhongMaterial

MeshStandardMaterial

MeshPhysicalMaterial

- RAPIDO

+ REALISTICO

Outros materiais (especiais):
MeshNormalMaterial, MeshDepthMaterial e ShadowMaterial.

\exemplos\4-material [1]

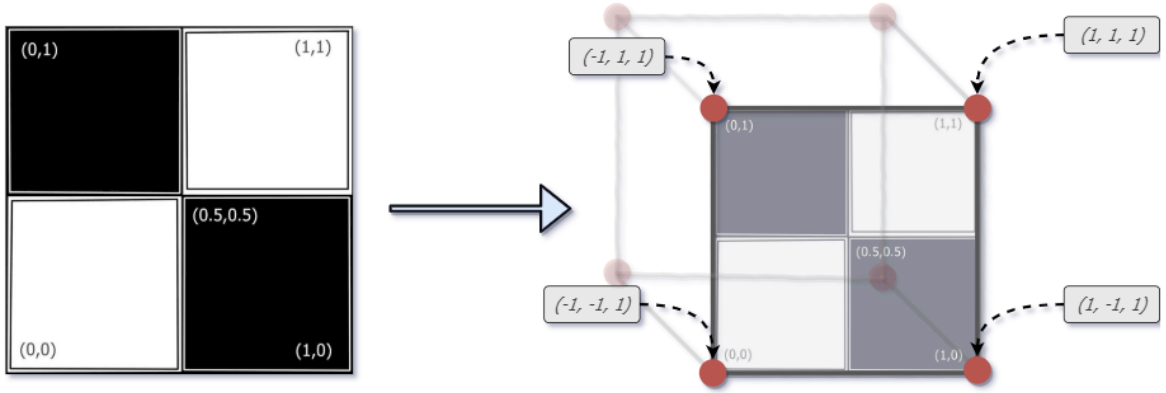
Material Feature Table

(*) Não é afetado pela luz.

Texturas

UV Mapping mapeia mapas de coordenadas 2D (u, v) coordinates em coordenadas 3D (x, y, z) :

$$(u, v) \longrightarrow (x, y, z)$$



\exemplos\4-material [2-4]

Imagem [1]

Luz

Imagem [1]

Luzes

- AmbientLight

  Luz que ilumina globalmente todos os objetos de forma igual.

- PointLight

  Luz emitida em todas as direções por um único ponto.

- DirectionalLight

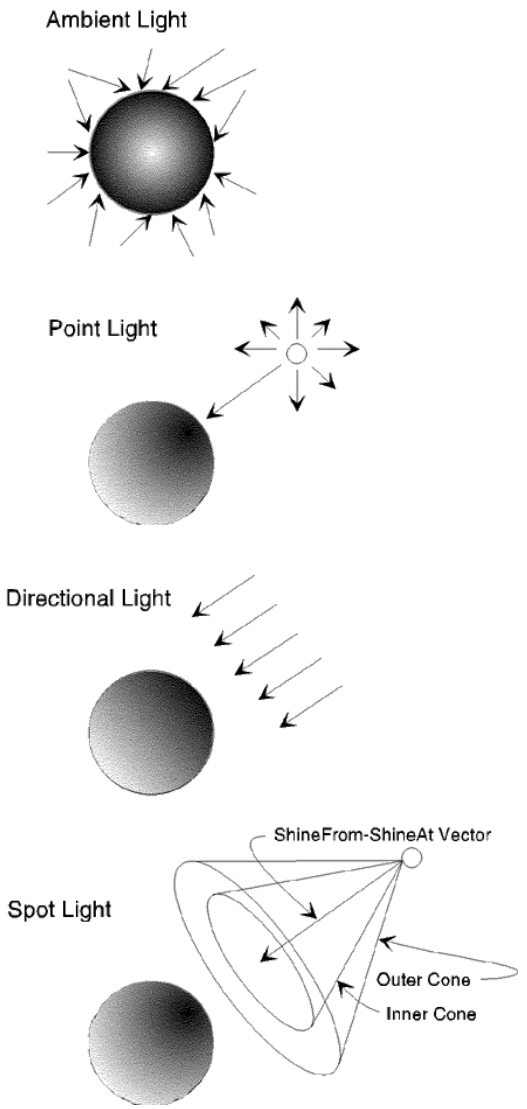
  Luz emitida numa direção específica.

- SpotLight

  Luz emitida a partir de um único ponto numa direção ao longo de um cone.

- HemisphereLight

  Fonte de luz posicionada diretamente acima da cena, com a cor desbotando da cor do céu para a cor do solo.



Imagem[1]

\exemplos\5-luz [1-2]

Sombras

- Configurar o renderizador

```
renderer.shadowMap.enabled = true;
// melhorar a qualidade das sombras
// renderer.shadowMap.type = THREE.PCFSoftShadowMap;
```

- Configurar os objetos

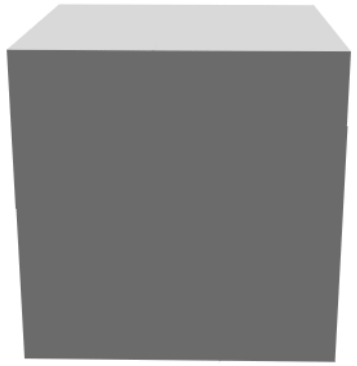
```
object3d.castShadow = true;
object3d.receiveShadow = false;
```

- Configurar as luzes

```
light.castShadow = true;
```

\exemplos\5-luz [3]

Animação



Imagem[1]

animate() {…}

```
function animate() {
    requestAnimationFrame(animate);

    mesh.rotation.x += 0.01;

    renderer.render(scene, camera);
}
```

The window.requestAnimationFrame method tells the browser that you wish to perform an animation and requests that the browser call a specified function to update an animation before the next repaint. The method takes a callback as an argument to be invoked before the repaint. [1]

\exemplos\6-animacao [1]

Animação

Exemplo – Labirinto

\exemplos\6-animacao [2]

Animações

Complexas

- O sistema de aninação (Animation System) do three.js é muito versátil e é muito semelhante ao sistema utilizado pelo Unreal Engine.
  - AnimationMixer – Controlador principal usado para controlar a aninação de um objeto.
    - Se manipularmos vários objetos serão necessários vários AnimationMixer.
  - AnimationAction – Determina como e quando as AnimationClip são executadas.
  - AnimationClip – Conjunto reutilizável de KeyframeTrack que representam uma animação.
  - KeyframeTrack – Sequência temporizada de keyframes usadas para animar uma propriedade específica de um objeto.
- Saber mais:

https://threejs.org/docs/#manual/en/introduction/Animation-system

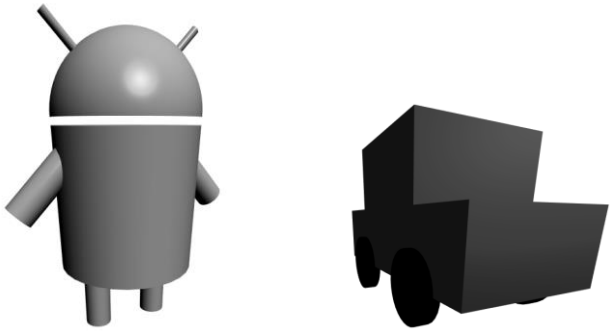
https://www.youtube.com/watch?v=YjJuniW0ktk

\exemplos\6-animacao [mixer]

Conclusão

"The only way to learn a new programming language is by writing programs in it."

Dennis Ritchie



Recursos

https://threejsfundamentals.org/

https://discoverthreejs.com/

https://adolfoguimaraes.github.io/threejs/

https://www.youtube.com/watch?v=uzkedMF-I4Q

https://www.youtube.com/watch?v=8jP4xpga6yY

https://www.youtube.com/watch?v=6oFvqLfRnsU

Visualizações 3D para a web com WebGL

Introdução ao Three.js

Óscar Oliveira (oao@estg.ipp.pt)

P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO