

CONSTRUA MUNDOS 3D DO ZERO USANDO JAVASCRIPTO

Vítor Norton
Dev Advocate @ SuperViz
@vt_norton

Beto Costa
Software Engineer @ Suzano
@robertheory





THREEJS

Para fazer algo em 3D com three.js, nós precisamos de três coisas: cena, camera e um renderizador. Para que possamos renderizar a cena com a camera

Tá falando grego?



DICIONÁRIO 3D

Bora conhecer algumas palavras chaves que serão importantes para conseguir construir o seu mundo



DICIONÁRIO 3D

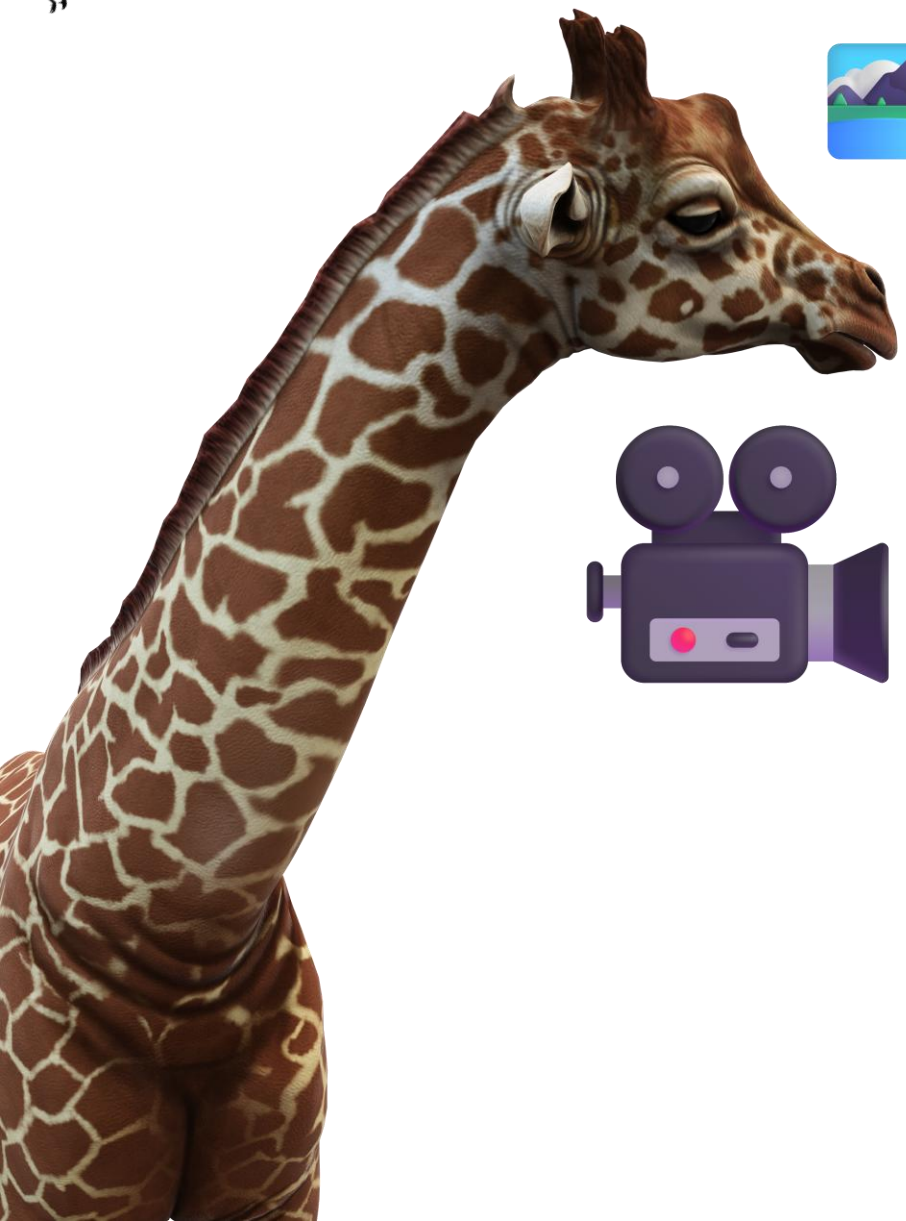
Bora conhecer algumas palavras chaves que serão importantes para conseguir construir o seu mundo



SCENE

/sēn/ *substantivo.*

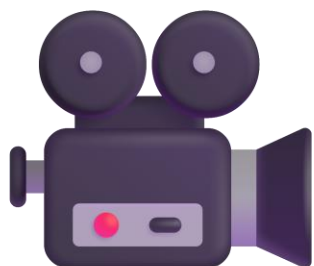
cena. cenário. palco. panorama. local. arena.



SCENE

/sēn/ *substantivo.*

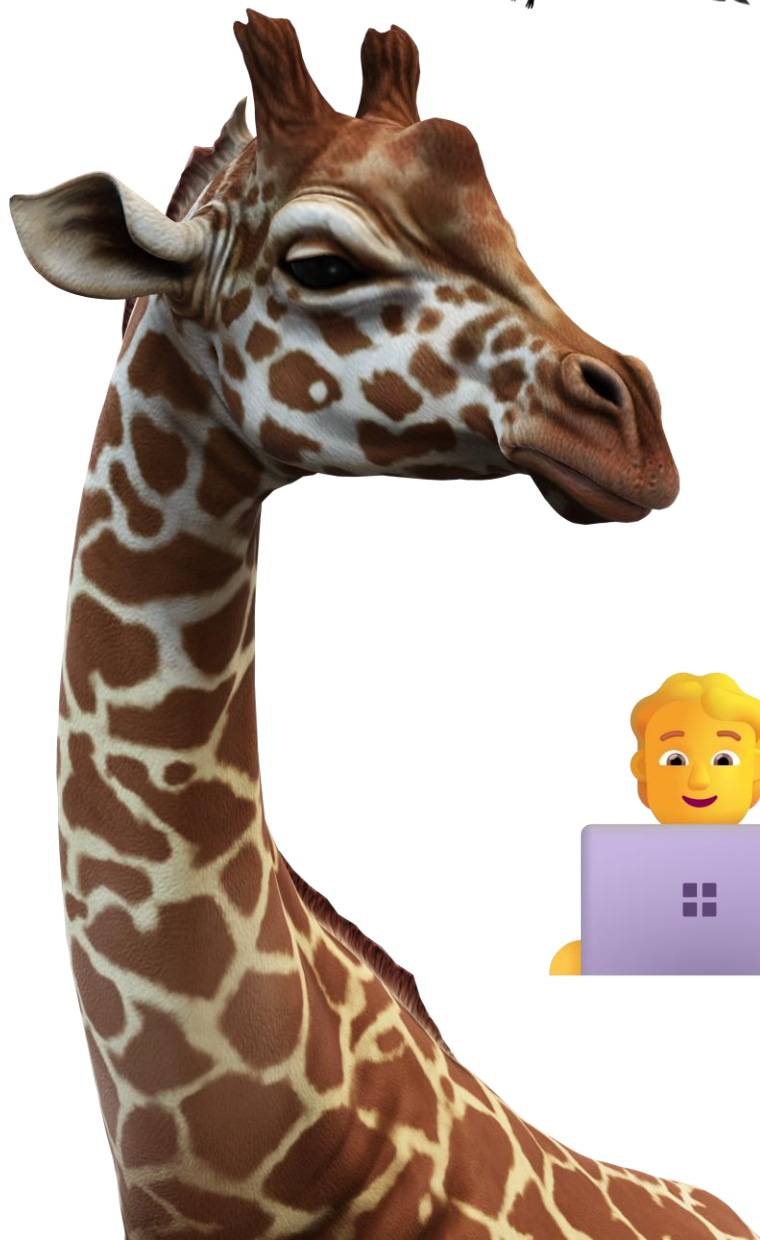
cena. cenário. palco. panorama. local. arena.



CAMERA

/'kam(ə)rə/ *substantivo.*

câmera. define a perspectiva do usuário,
controlando o que é visto e o que é exibido na tela



SCENE

/sēn/ *substantivo.*

cena. cenário. palco. panorama. local. arena.



CAMERA

/'kam(ə)rə/ *substantivo.*

câmera. define a perspectiva do usuário, controlando o que é visto e o que é exibido na tela



RENDERER

/'rendər/ *substantivo.*

responsável por transformar dados do modelo 3D em imagens 2D na tela, exibindo a cena virtual com texturas, iluminação e sombras.



SCENE

/sēn/ *substantivo.*

cena. cenário. palco. panorama. local. arena.



CAMERA

/'kam(ə)rə/ *substantivo.*

câmera. define a perspectiva do usuário, controlando o que é visto e o que é exibido na tela



RENDERER

/'rendər/ *substantivo.*

responsável por transformar dados do modelo 3D em imagens 2D na tela, exibindo a cena virtual com texturas, iluminação e sombras.





CAMERA

O ThreeJS possui vários tipos de cameras, CubeCamera, OrthographicCamera, mas nenhuma delas importa para gente.



Perspective Camera



CAMERA

O ThreeJS possui vários tipos de cameras, CubeCamera, OrthographicCamera, mas nenhuma delas importa para gente.



Perspective Camera

Ela mimifica o que nós humanos entendemos como olho. É a mais comum e usada para renderizar uma cena 3D.





Perspective Camera

Ela mimifica o que nós humanos entendemos como olho. É a mais comum e usada para renderizar uma cena 3D.

- FOV
- Aspect Ratio
- Near
- Far
- Wherever you are 🧑❤️🚢🤖





Perspective Camera

Ela mimifica o que nós humanos entendemos como olho. É a mais comum e usada para renderizar uma cena 3D.

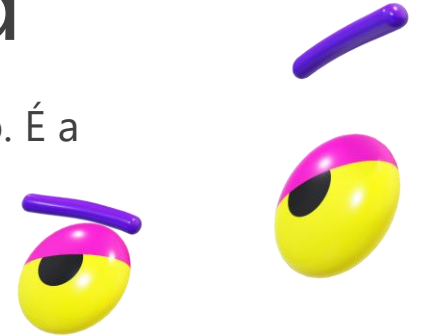
- FOV
- Aspect Ratio
- Near
- Far



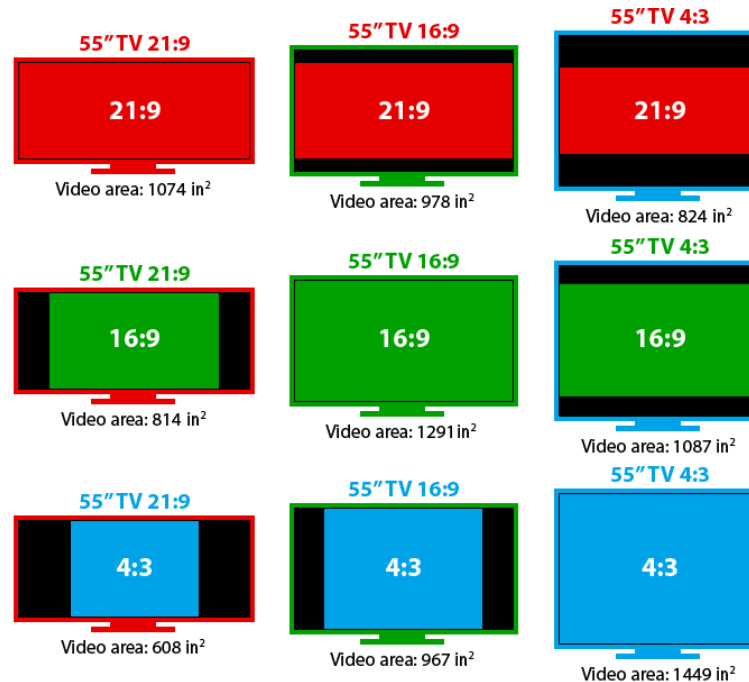


Perspective Camera

Ela mimifica o que nós humanos entendemos como olho. É a mais comum e usada para renderizar uma cena 3D.



- FOV: Field of View (Campo de visão) em graus
- Aspect Ratio: 4x3, 16x9 (Largura:Altura)
- Near
- Far

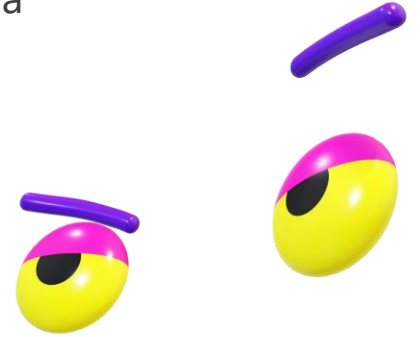




Perspective Camera

Ela mimifica o que nós humanos entendemos como olho. É a mais comum e usada para renderizar uma cena 3D.

- FOV: Field of View (Campo de visão) em graus
- Aspect Ratio: 4x3, 16x9 (Largura:Altura)
- Near: De onde a camera começa a ver
- Far: Até onde a camera termina de ver

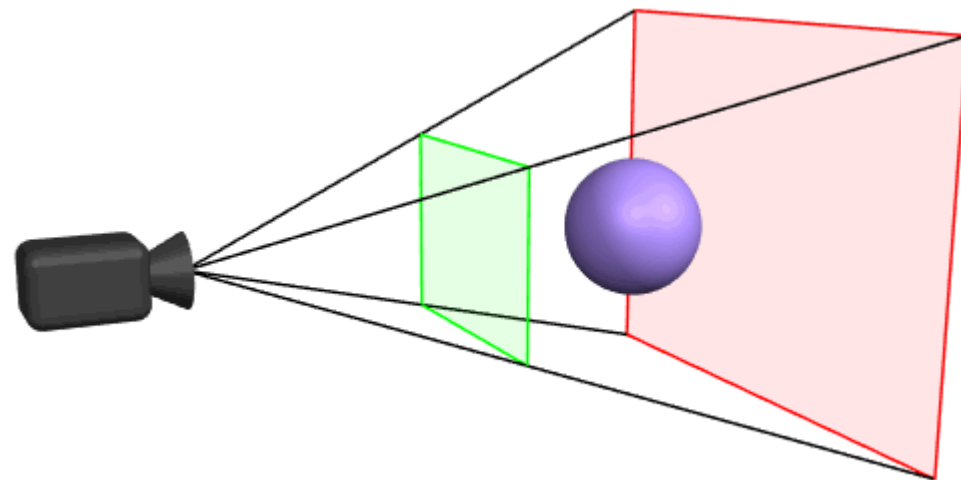
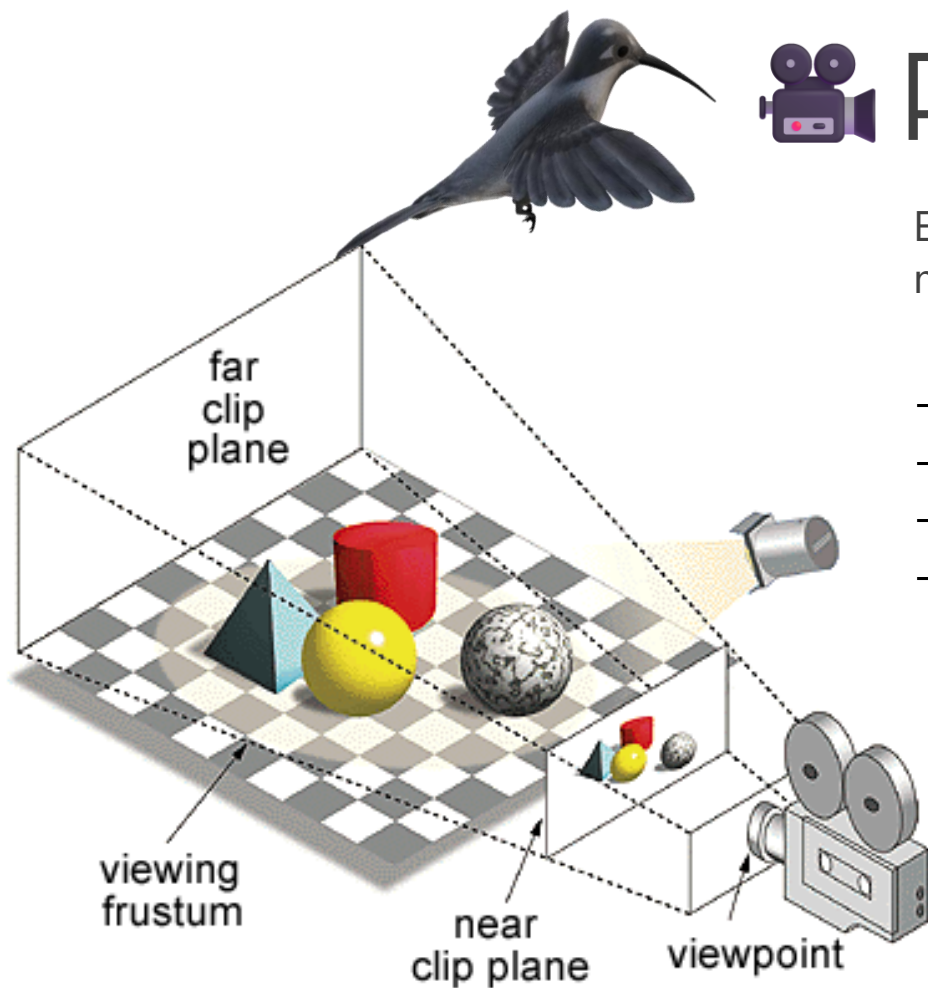




Perspective Camera

Ela mimifica o que nós humanos entendemos como olho. É a mais comum e usada para renderizar uma cena 3D.

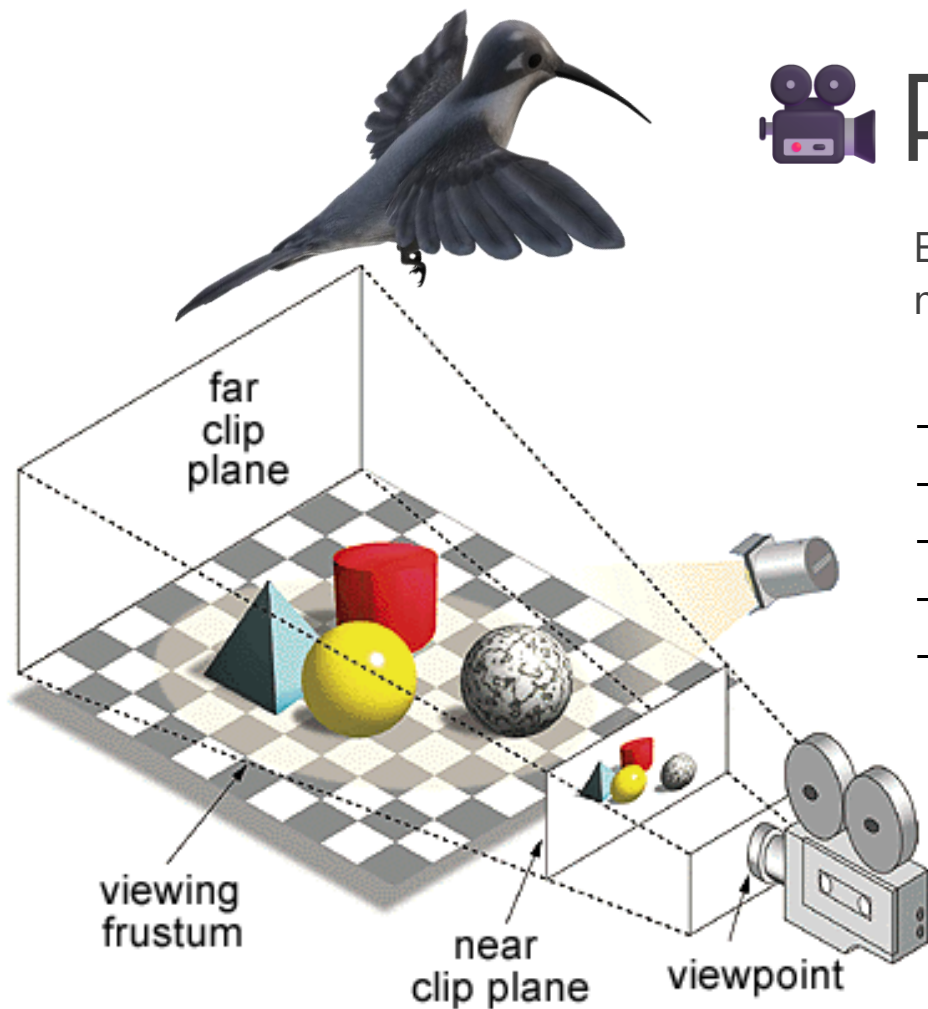
- FOV: Field of View (Campo de visão) em graus
- Aspect Ratio: 4x3, 16x9 (Largura:Altura)
- Near: De onde a camera começa a ver
- Far: Até onde a camera termina de ver





Perspective Camera

Ela mimifica o que nós humanos entendemos como olho. É a mais comum e usada para renderizar uma cena 3D.



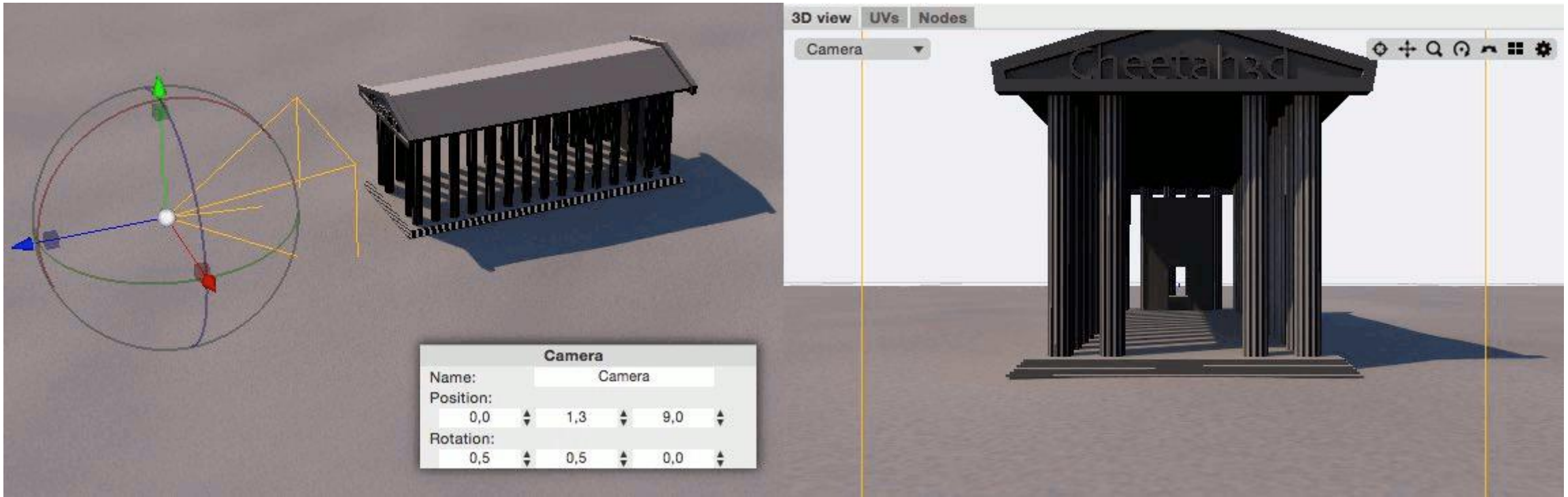
- FOV: Field of View (Campo de visão) em graus
- Aspect Ratio: 4x3, 16x9 (Largura:Altura)
- Near: De onde a camera começa a ver
- Far: Até onde a camera termina de ver
- Position: Aonde a camera está



Perspective Camera

Ela mimifica o que nós humanos entendemos como olho. É a mais comum e usada para renderizar uma cena 3D.

- FOV: Field of View (Campo de visão) em graus
- Aspect Ratio: 4x3, 16x9 (Largura:Altura)
- Near: De onde a camera começa a ver
- Far: Até onde a camera termina de ver
- Position: Aonde a camera está





SCENE

/sēn/ *substantivo.*

cena. cenário. palco. panorama. local. arena.



CAMERA

/'kam(ə)rə/ *substantivo.*

câmera. define a perspectiva do usuário,
controlando o que é visto e o que é exibido na tela



RENDERER

/'rendər/ *substantivo.*

responsável por transformar dados do modelo 3D
em imagens 2D na tela, exibindo a cena virtual com
texturas, iluminação e sombras.



RENDERER

Assim como a camera o ThreeJS tem vários Renderes possíveis... Todos eles baseados no WebGL





RENDERER

O que é WebGL vc me pergunta!?

Web Graphics Library (JavaScript API para renderizar coisas 2D e 3D), o que traduzindo significa: um motor que cria gráficos 3D e animações diretamente no navegador.



Precisará determinar a área em que será renderizado na página.



SCENE

/sēn/ *substantivo.*

cena. cenário. palco. panorama. local. arena.



CAMERA

/'kam(ə)rə/ *substantivo.*

câmera. define a perspectiva do usuário,
controlando o que é visto e o que é exibido na tela



RENDERER

/'rendər/ *substantivo.*

responsável por transformar dados do modelo 3D
em imagens 2D na tela, exibindo a cena virtual com
texturas, iluminação e sombras.



PARA COMEÇAR

- <https://bit.ly/threejs-begginers>
- Ter noção de HTML
- Precisa manjar de JavaScript
- Usar o projeto base em *[superviz/threejs-for-beginners](#)*



[threejs-for-begginers](#) - CodeSandbox



O QUE JÁ TEM PRONTO? 🤔

- Projeto base de vite (react-ts)
- Pacotes instalados:
 - three
 - @types/three*
 - @superviz/sdk



[threejs-for-begginers - CodeSandbox](#)



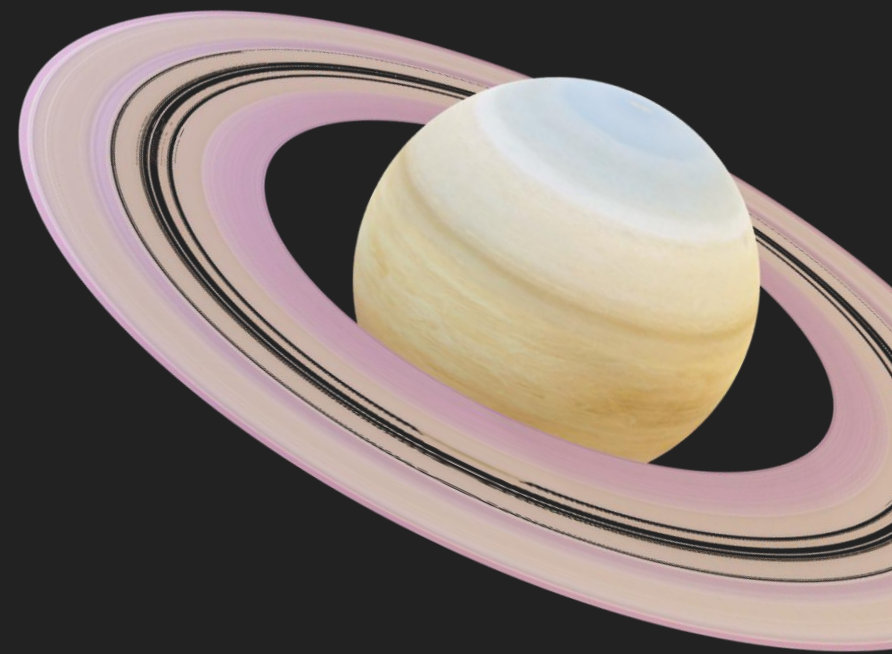
CRIANDO UMA CENA

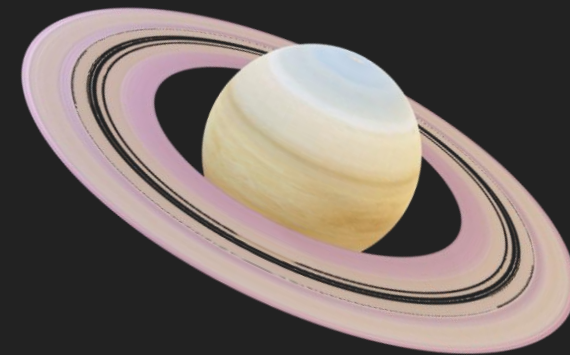
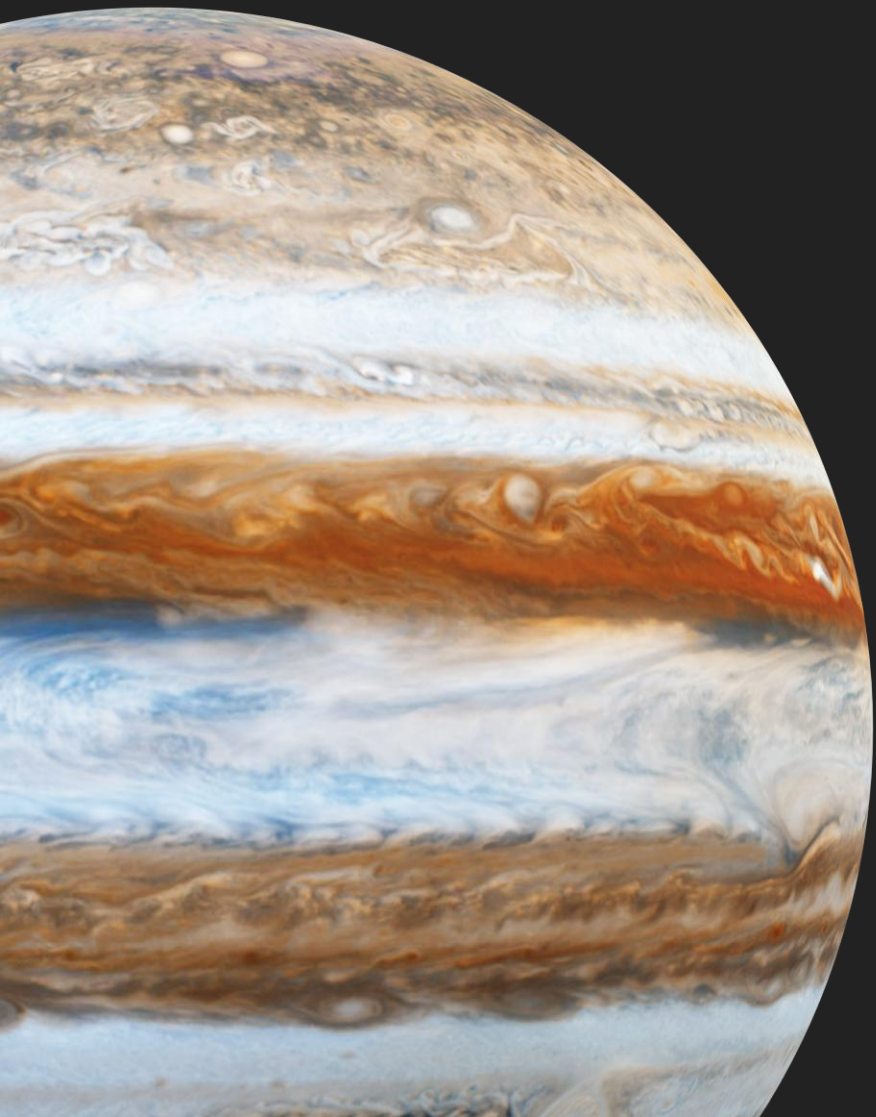


Vamos traduzir o que vimos até agora em código

SHOW, BELEZA

Mas só tenho uma tela em branco, cadê o 3D?





DICIONÁRIO 3D

Bora conhecer mais algumas palavras



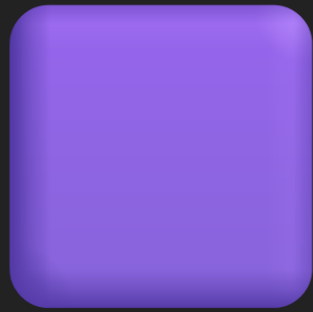
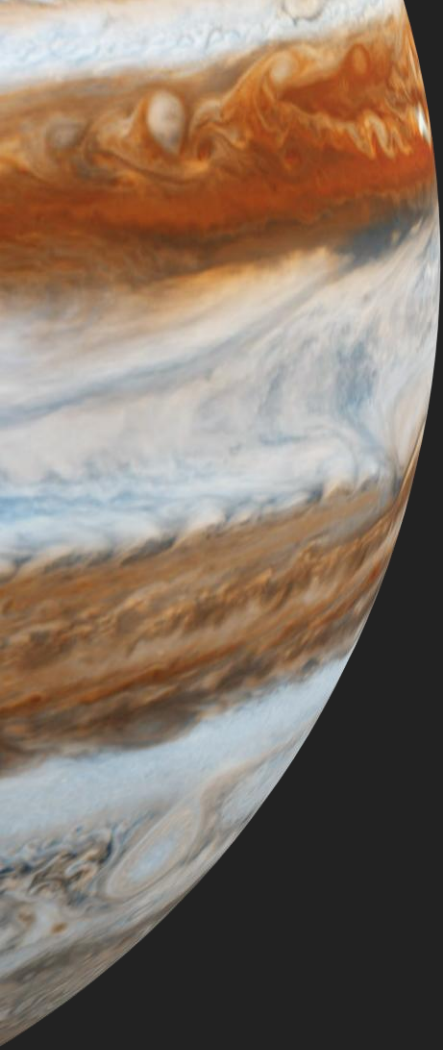
DICIONÁRIO 3D

Bora conhecer mais algumas palavras



BOX GEOMETRY

Variavel que contem todos os pontos (vertices) e preenchimentos (faces) de um objeto 3D



BOX GEOMETRY

Variavel que contem todos os pontos (vertices) e preenchimentos (faces) de um objeto rectangular 3D

Geometries

[BoxGeometry](#)

CapsuleGeometry

CircleGeometry

ConeGeometry

CylinderGeometry

DodecahedronGeometry

EdgesGeometry

ExtrudeGeometry

IcosahedronGeometry

LatheGeometry

OctahedronGeometry

PlaneGeometry

PolyhedronGeometry

RingGeometry

ShapeGeometry

SphereGeometry

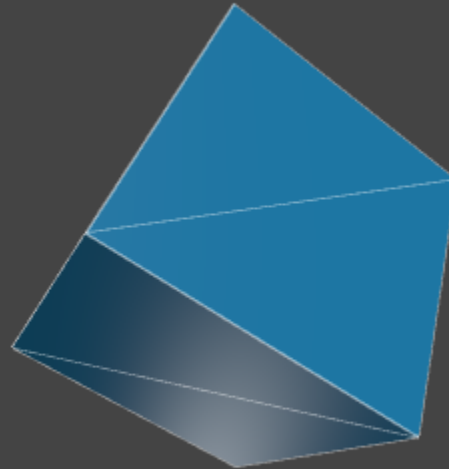
TetrahedronGeometry

TorusGeometry

TorusKnotGeometry

TubeGeometry

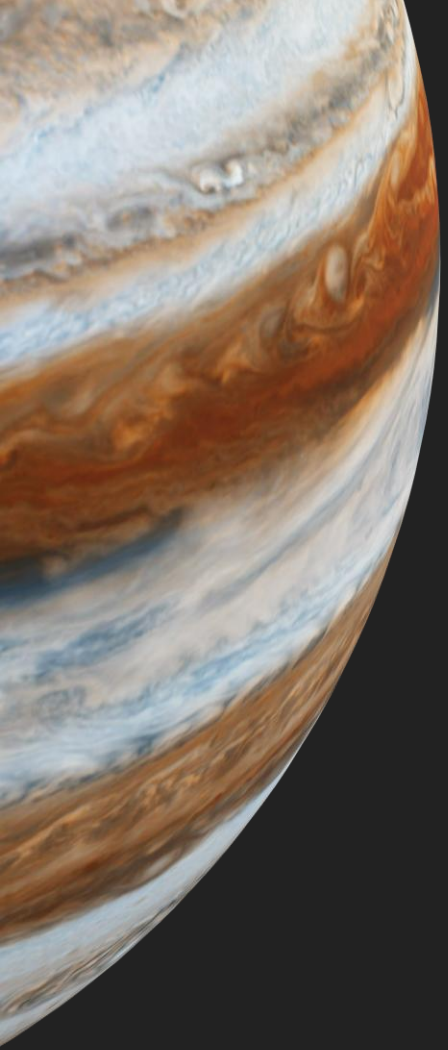
WireframeGeometry



Controls

THREE.BoxGeometry

width	<input type="text"/>	15
height	<input type="text"/>	15
depth	<input type="text"/>	15
widthSegments	<input type="text"/>	1
heightSegments	<input type="text"/>	1
depthSegments	<input type="text"/>	1



CAPSULE GEOMETRY

Geometries

[BoxGeometry](#)

CapsuleGeometry

CircleGeometry

ConeGeometry

CylinderGeometry

DodecahedronGeometry

EdgesGeometry

ExtrudeGeometry

IcosahedronGeometry

LatheGeometry

OctahedronGeometry

PlaneGeometry

PolyhedronGeometry

RingGeometry

ShapeGeometry

SphereGeometry

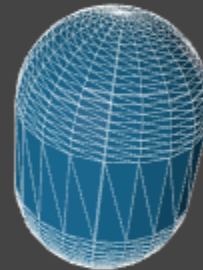
TetrahedronGeometry

TorusGeometry

TorusKnotGeometry

TubeGeometry

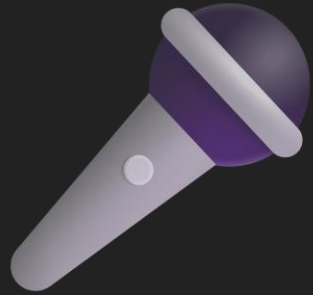
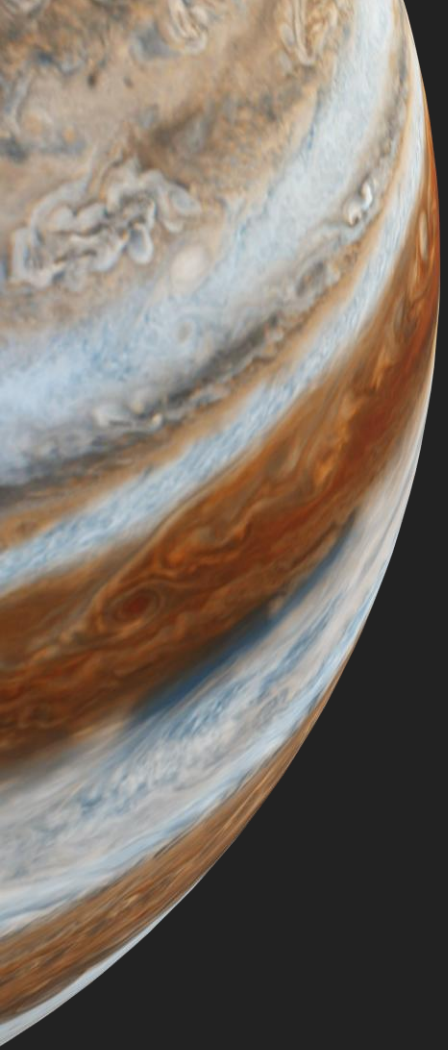
WireframeGeometry



Controls

THREE.CapsuleGeometry

radius	<input type="text"/>	5
length	<input type="text"/>	5
capSegments	<input type="text"/>	10
radialSegments	<input type="text"/>	20



CONE GEOMETRY

Geometries

[BoxGeometry](#)

CapsuleGeometry

CircleGeometry

ConeGeometry

CylinderGeometry

DodecahedronGeometry

EdgesGeometry

ExtrudeGeometry

IcosahedronGeometry

LatheGeometry

OctahedronGeometry

PlaneGeometry

PolyhedronGeometry

RingGeometry

ShapeGeometry

SphereGeometry

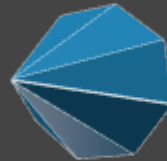
TetrahedronGeometry

TorusGeometry

TorusKnotGeometry

TubeGeometry

WireframeGeometry



Controls

THREE.ConeGeometry

radius	<input type="text"/>	5
height	<input type="text"/>	10
radialSegments	<input type="text"/>	8
heightSegments	<input type="text"/>	1
openEnded	<input type="checkbox"/>	
thetaStart	<input type="text"/>	0
thetaLength	<input type="text"/>	6.283185



SPHERE GEOMETRY

Geometries

[BoxGeometry](#)

CapsuleGeometry

CircleGeometry

ConeGeometry

CylinderGeometry

DodecahedronGeometry

EdgesGeometry

ExtrudeGeometry

IcosahedronGeometry

LatheGeometry

OctahedronGeometry

PlaneGeometry

PolyhedronGeometry

RingGeometry

ShapeGeometry

SphereGeometry

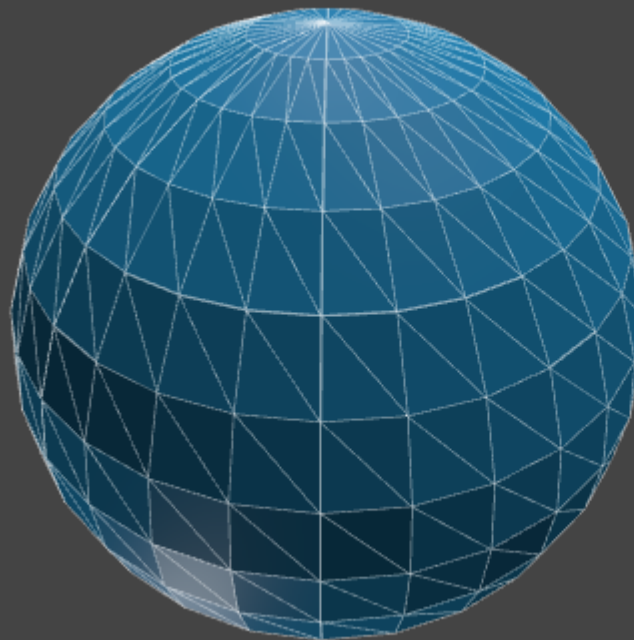
TetrahedronGeometry

TorusGeometry

TorusKnotGeometry

TubeGeometry

WireframeGeometry



Controls

THREE.SphereGeometry

radius	<input type="text"/>	15
widthSegments	<input type="text"/>	32
heightSegments	<input type="text"/>	16
phiStart	<input type="text"/>	0
phiLength	<input type="text"/>	6.283185
thetaStart	<input type="text"/>	0
thetaLength	<input type="text"/>	3.141592



TORUS KNOT GEOMETRY

Geometries

[BoxGeometry](#)

CapsuleGeometry

CircleGeometry

ConeGeometry

CylinderGeometry

DodecahedronGeometry

EdgesGeometry

ExtrudeGeometry

IcosahedronGeometry

LatheGeometry

OctahedronGeometry

PlaneGeometry

PolyhedronGeometry

RingGeometry

ShapeGeometry

SphereGeometry

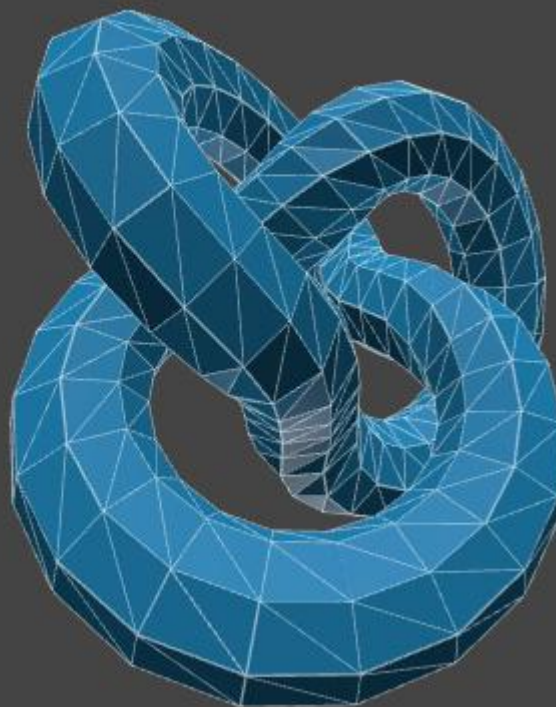
TetrahedronGeometry

TorusGeometry

TorusKnotGeometry

TubeGeometry

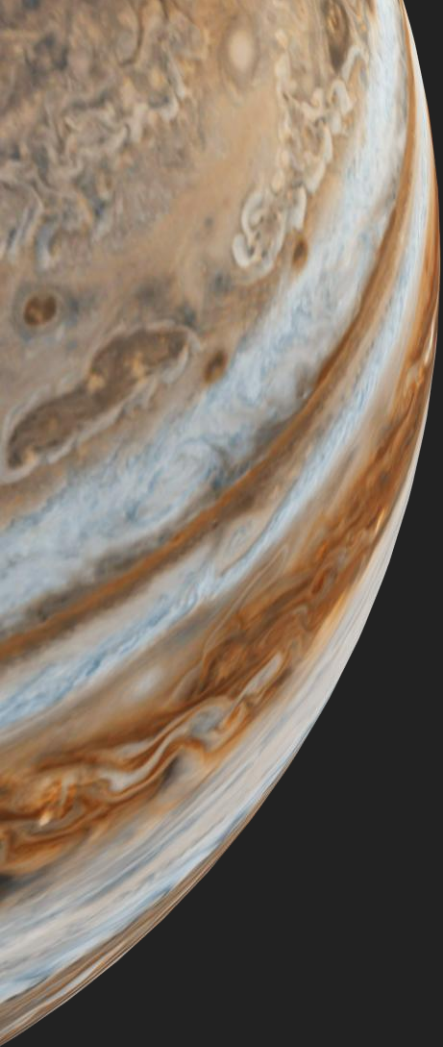
WireframeGeometry



Controls

THREE.TorusKnotGeometry

radius	<input type="text"/>	10
tube	<input type="text"/>	3
tubularSegments	<input type="text"/>	64
radialSegments	<input type="text"/>	8
p	<input type="text"/>	2
q	<input type="text"/>	3



BASIC MATERIAL

Material para colorir a nossa geometria. Para manter simples, usaremos o MeshBasicMaterial que é só cores com valores HEX (#FF00FF)

Materials

[LineBasicMaterial](#)

LineDashedMaterial

Material

MeshBasicMaterial

MeshDepthMaterial

MeshDistanceMaterial

MeshLambertMaterial

MeshMatcapMaterial

MeshNormalMaterial

MeshPhongMaterial

MeshPhysicalMaterial

MeshStandardMaterial

MeshToonMaterial

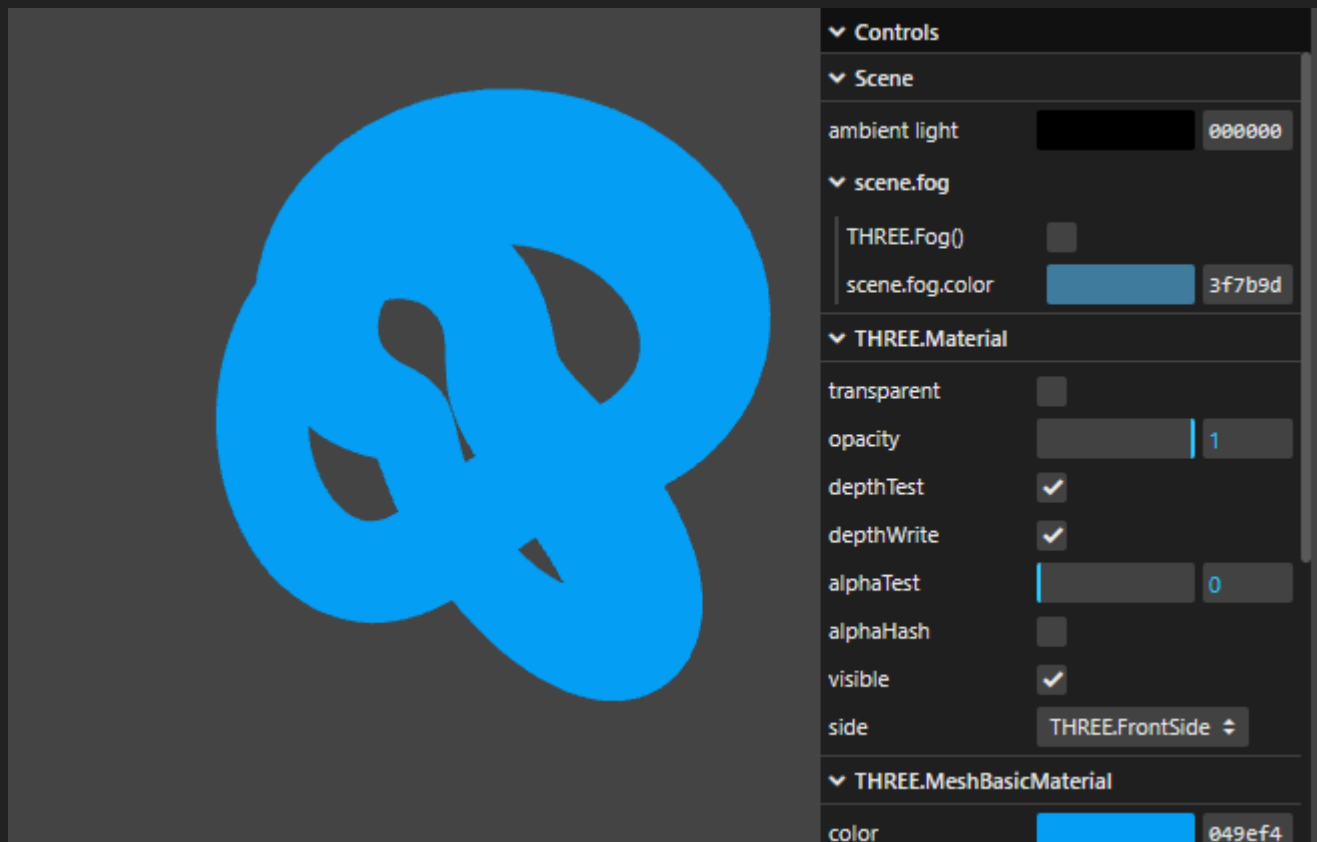
PointsMaterial

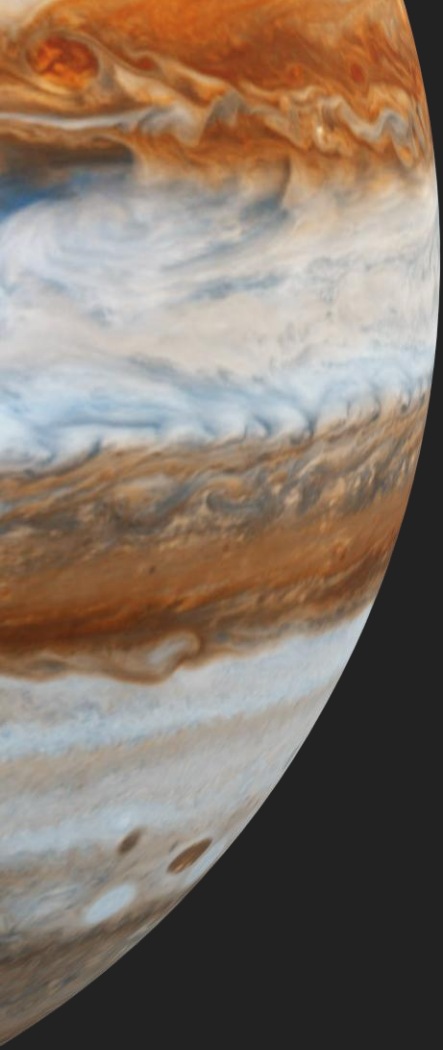
RawShaderMaterial

ShaderMaterial

ShadowMaterial

SpriteMaterial





DEPTH MATERIAL

Materials

[LineBasicMaterial](#)

LineDashedMaterial

Material

MeshBasicMaterial

MeshDepthMaterial

MeshDistanceMaterial

MeshLambertMaterial

MeshMatcapMaterial

MeshNormalMaterial

MeshPhongMaterial

MeshPhysicalMaterial

MeshStandardMaterial

MeshToonMaterial

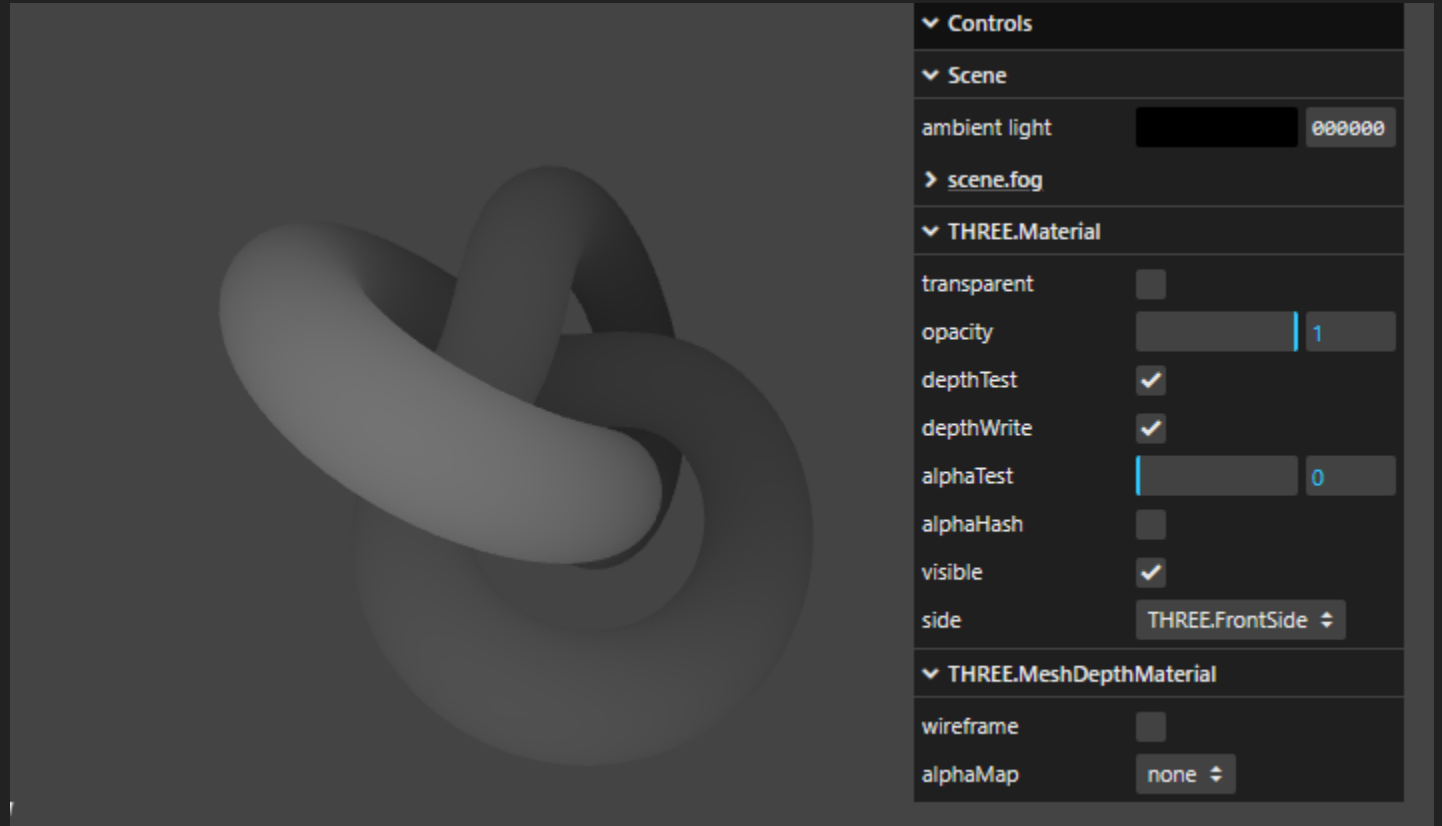
PointsMaterial

RawShaderMaterial

ShaderMaterial

ShadowMaterial

SpriteMaterial





BOX GEOMETRY

Variavel que contem todos os pontos (vertices) e preenchimentos (faces) de um objeto rectangular 3D



BASIC MATERIAL

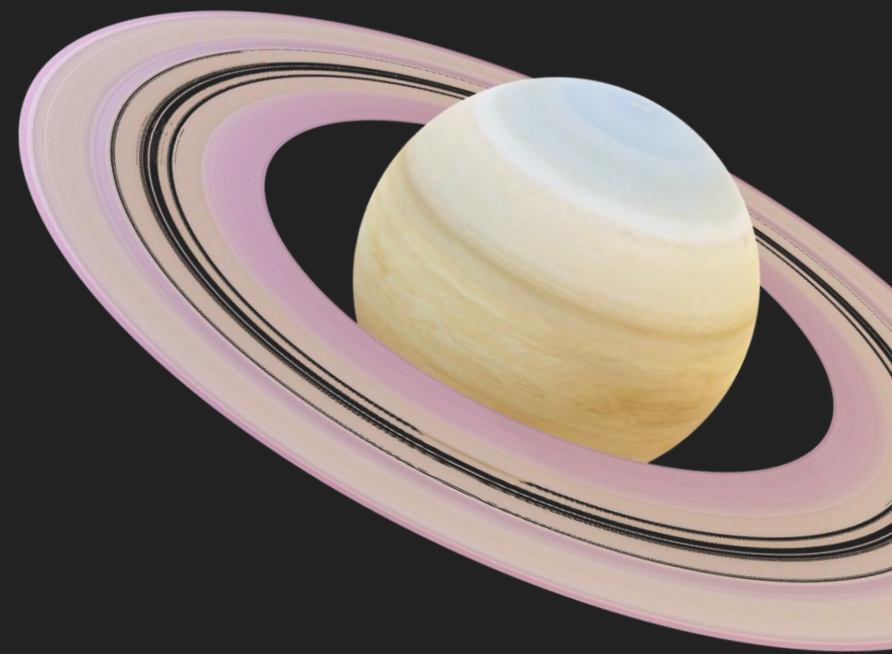
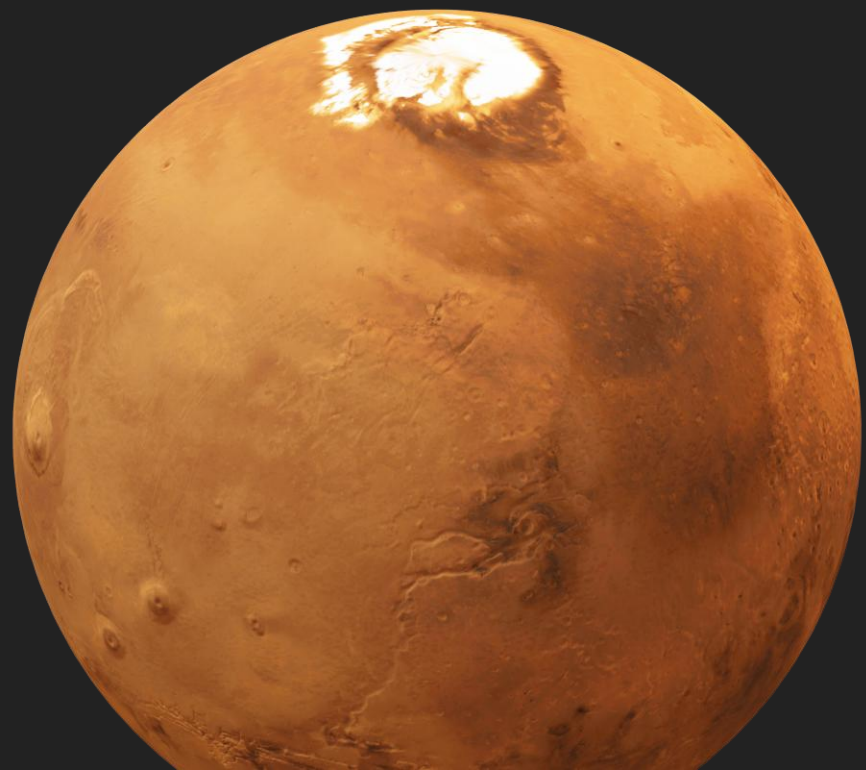
Material para colorir a nossa geometria. Para manter simples, usaremos o MeshBasicMaterial que é só cores com valores HEX (#FF00FF)



MESH

O que vai juntar nossa geometria ao material,

SHOW, BELEZA



CUBO EM CENA

MAS EU NÃO VEJO O CUBO?!?!?! Cadê ele?



CUBO EM CENA

MAS EU NÃO VEJO O CUBO?!?!?! Cadê ele?

É que bem, você não disse onde o cubo está nesse palco virtual, então ele está na posição 0,0,0.

Acontece que a camera também está nesta posição.



CUBO EM CENA

MAS EU NÃO VEJO O CUBO?!?!?! Cadê ele?

É que bem, você não disse onde o cubo está nesse palco virtual, então ele está na posição 0,0,0 (x, y, z).

Acontece que a camera também está nesta posição.

Vamos mover a camera!



CUBO EM CENA

MAS EU AINDA NÃO VEJO O CUBO?!?!?! Cadê ele?
VOCÊ ME PROMETEU?



CUBO EM CENA

MAS EU AINDA NÃO VEJO O CUBO?!?!?! Cadê ele?
VOCÊ ME PROMETEU?

É que assim, a gente não tá renderizando
ele...



RENDERIZAR NA TELA

Precisamos desenhar na tela a informação que temos construído até agora, e o ideal – se você é de jogos vai entender – é que seja em 60 quadros por segundo (FPS).



RENDERIZAR NA TELA

Precisamos desenhar na tela a informação que temos construído até agora, e o ideal – se você é de jogos vai entender – é que seja em 60 quadros por segundo (FPS).

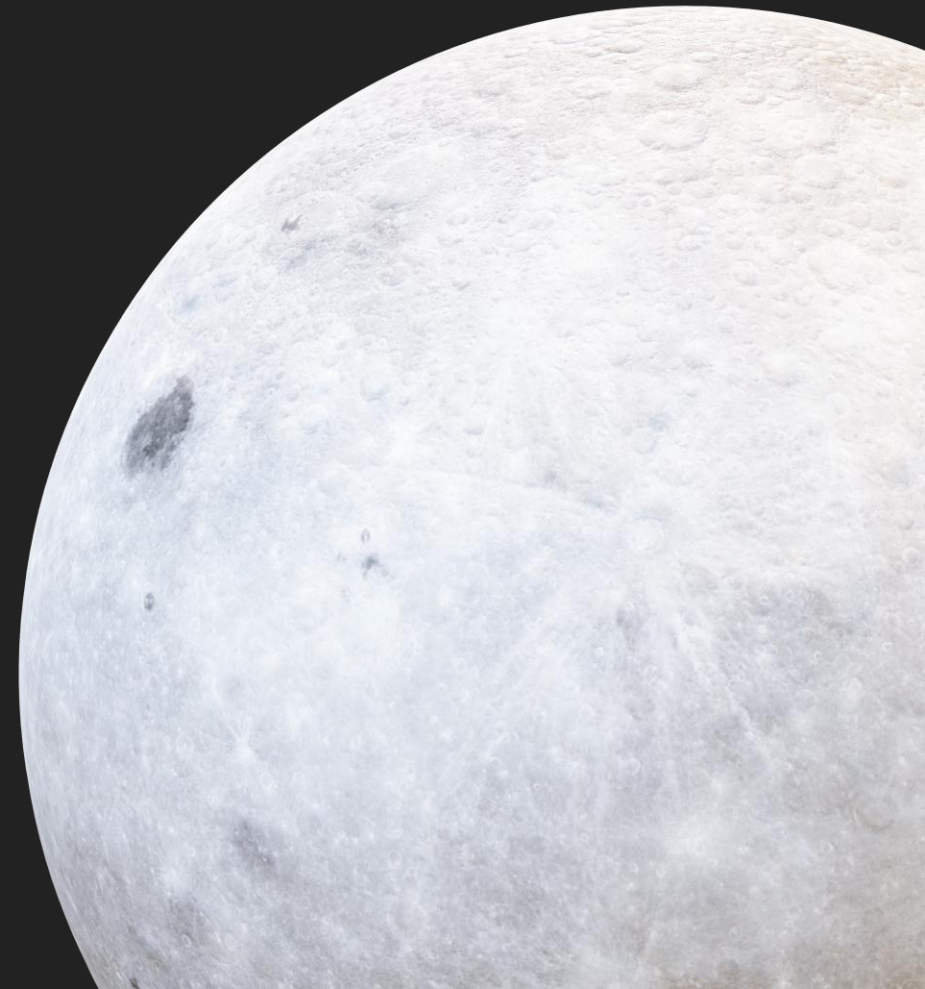


RENDERIZAR NA TELA

Precisamos desenhar na tela a informação que temos construído até agora, e o ideal – se você é de jogos vai entender – é que seja em 60 quadros por segundo (FPS).

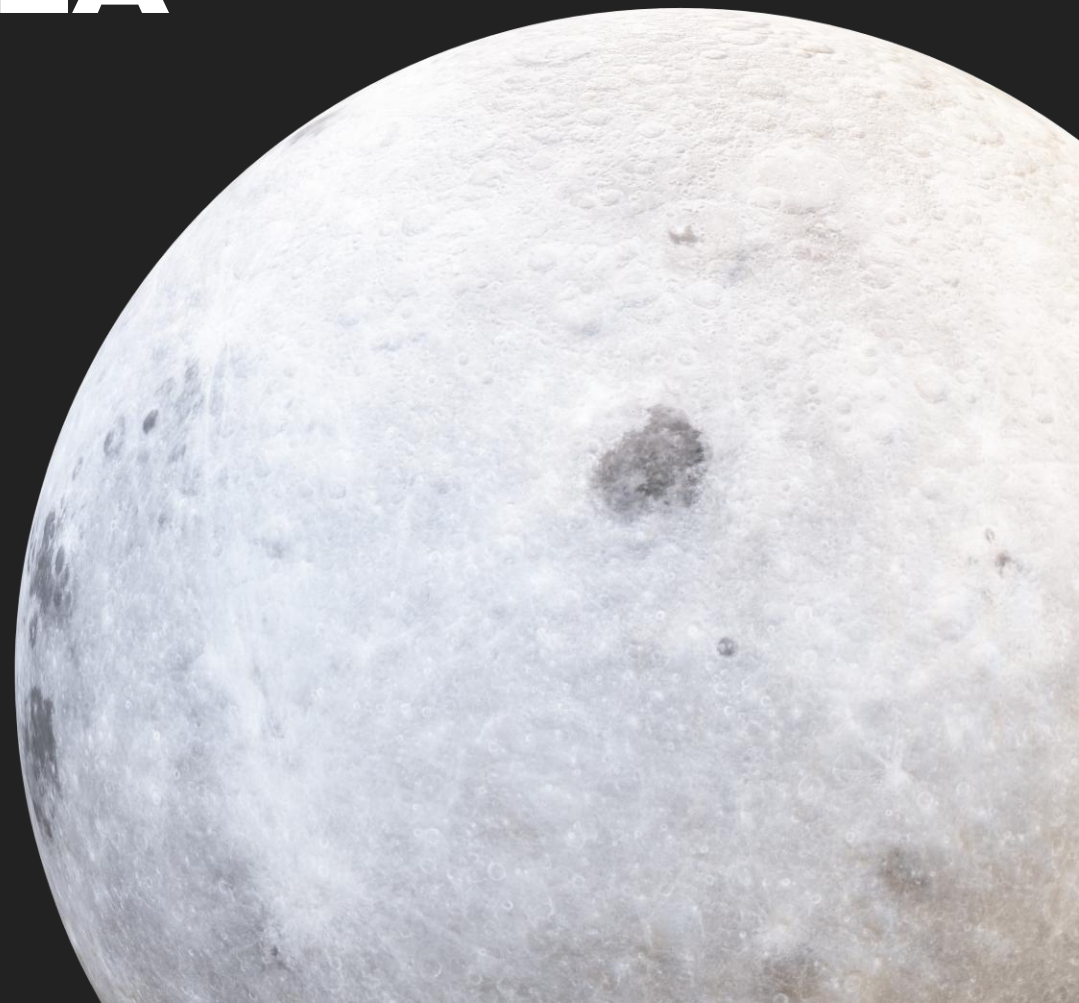


Isto é, criar uma animação!



RENDERIZAR NA TELA

Como é isso no código?



READY PLAYER ME

Bora adicionar pessoas dentro do nosso mundo



[Ready Player Me Avatar Creator](https://readyplayer.me)



ADICIONE PESSOAS NO SEU MUNDO

