Universidade Cruzeiro do Sul

# Computação Gráfica

- INTRODUÇÃO A MODELAGEM, MALHAS E PRIMITIVAS GRÁFICAS

amlucena@cruzeirodosul.edu.br

# Na última aula...

Computação Gráfica

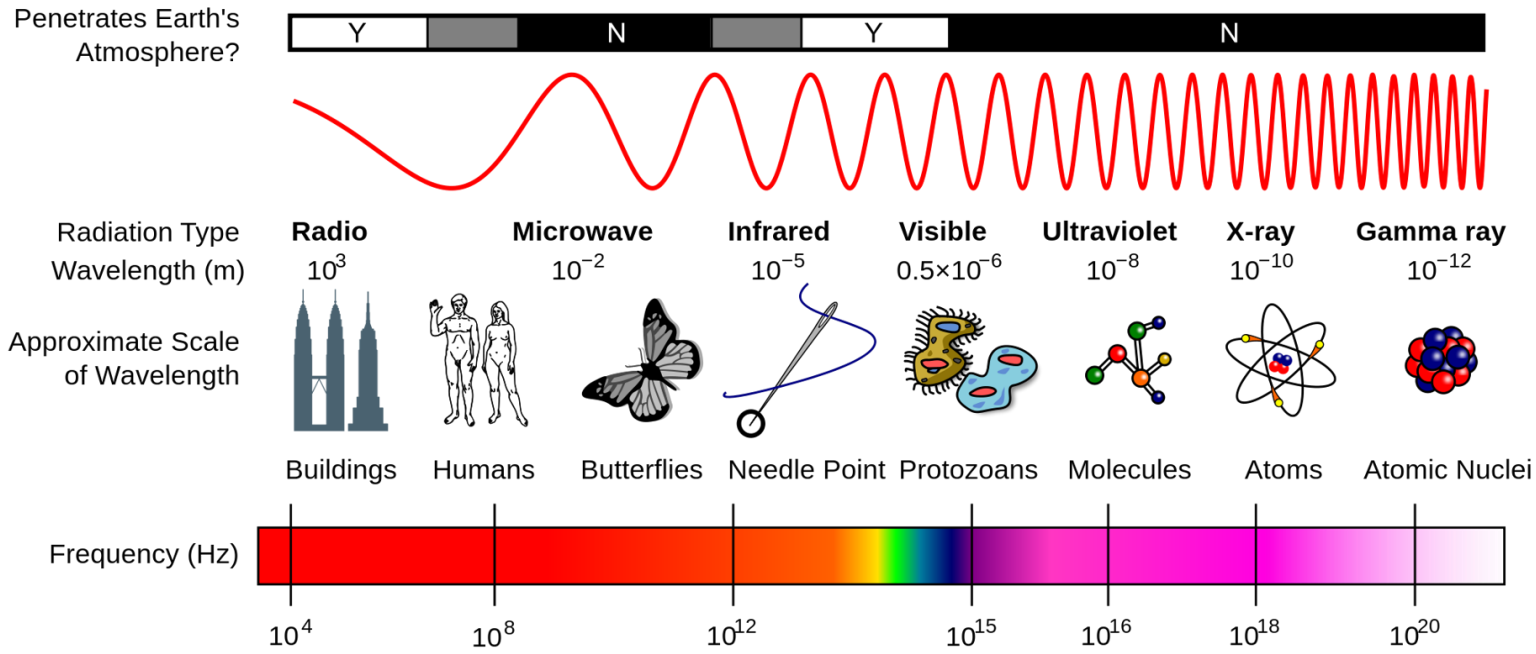
X

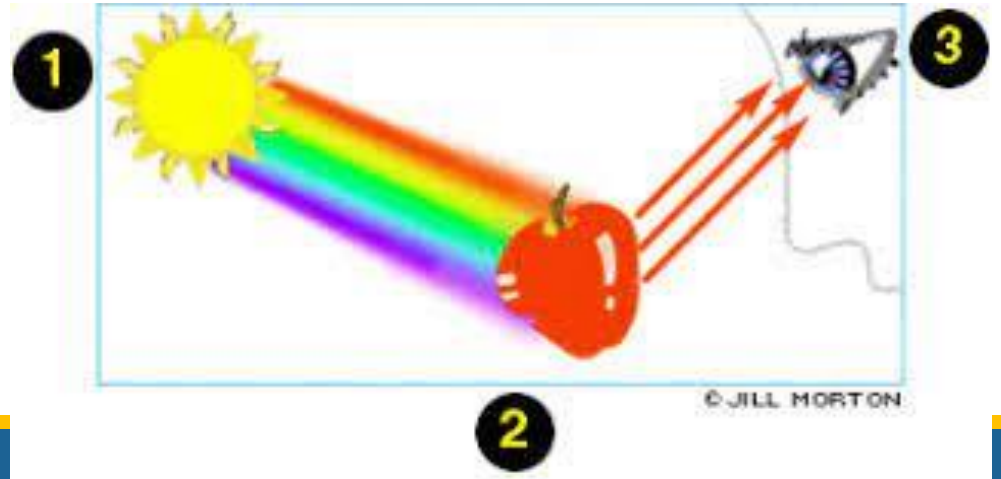
Processamento de Imagem

X

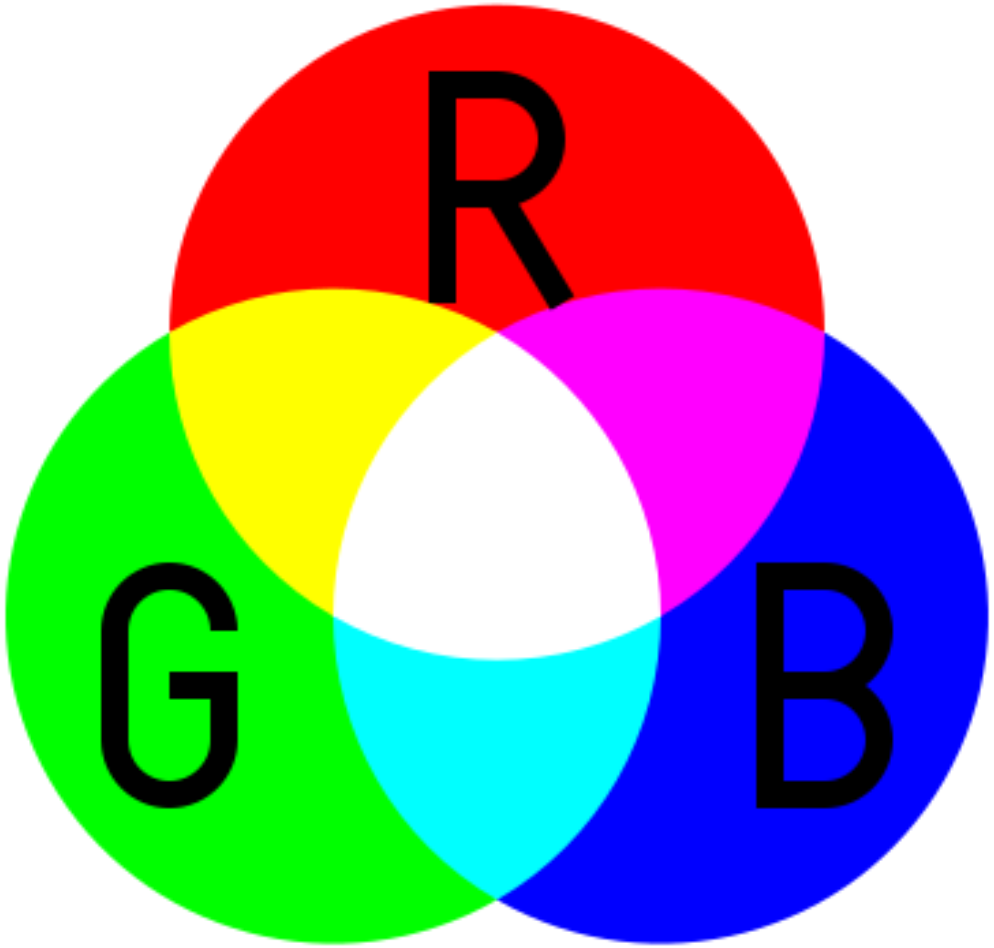
Visão Computacional


# Na última aula...

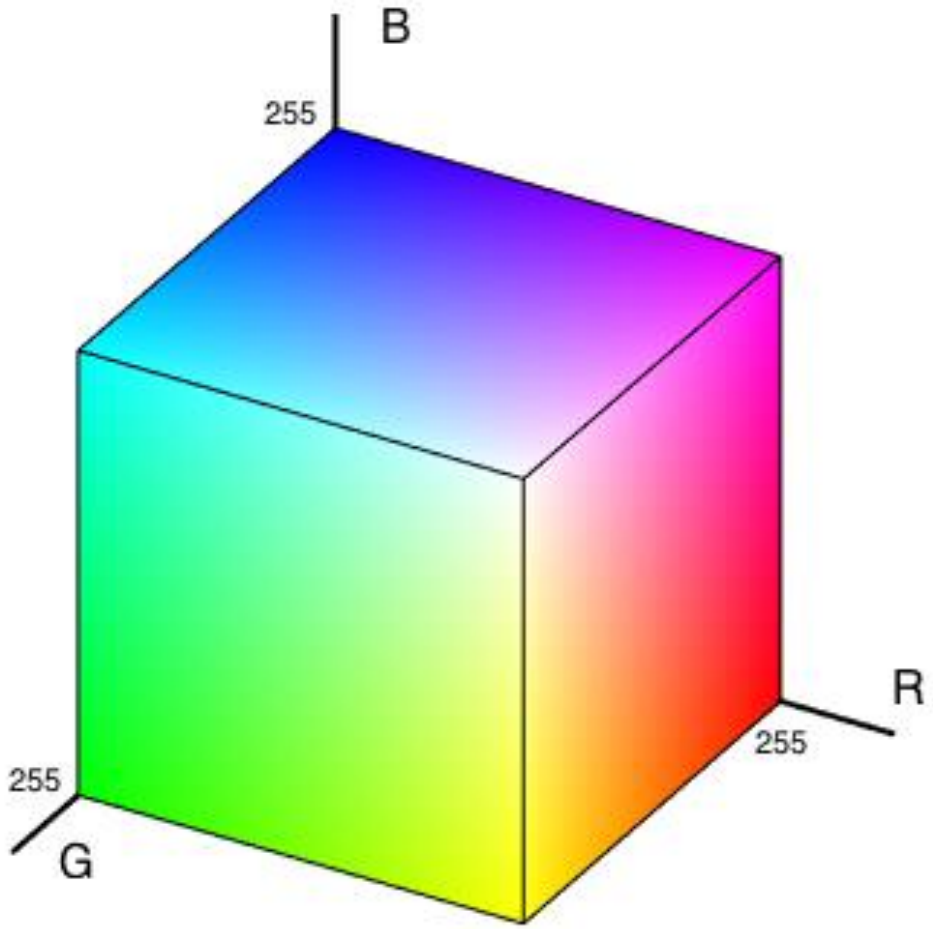


# Na última aula...

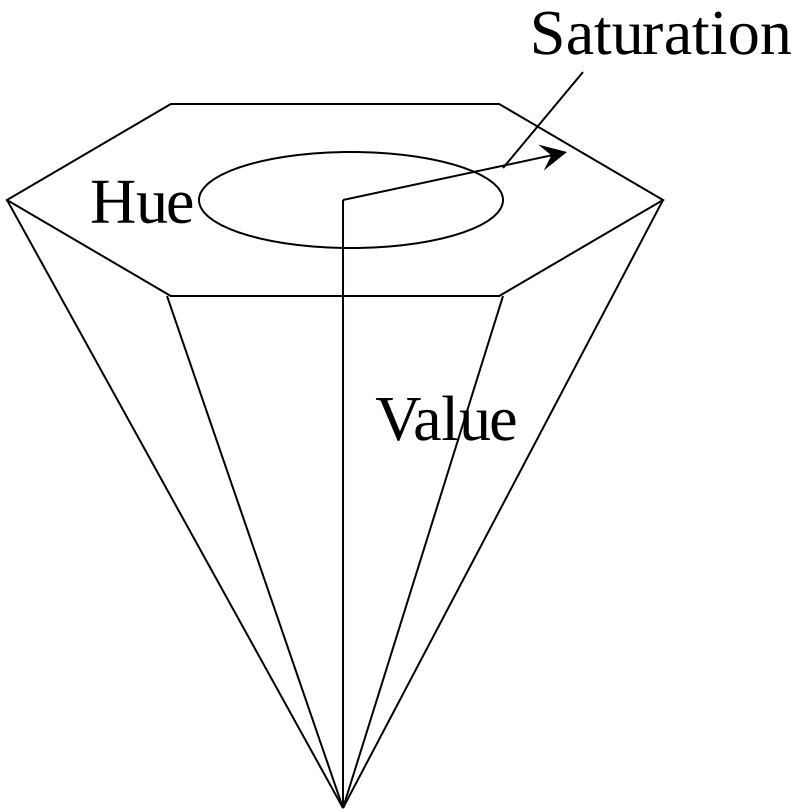


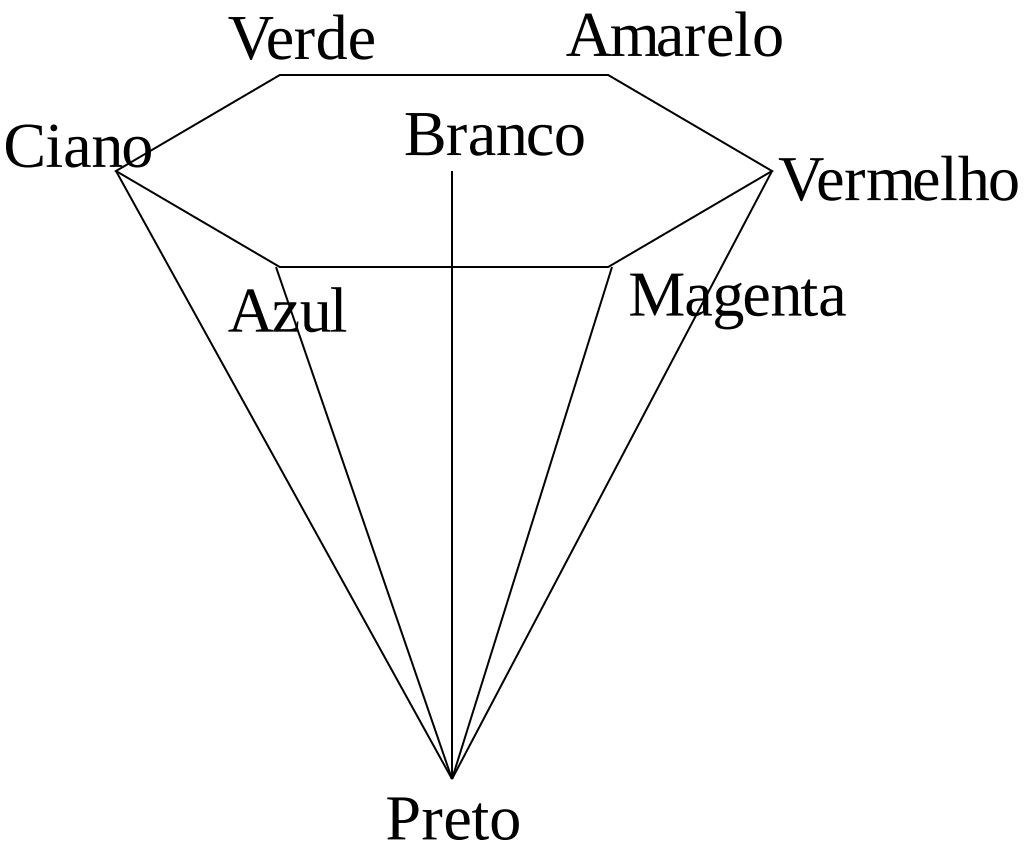
# Na última aula...




Young–Helmholtz theory of trichromatic color vision

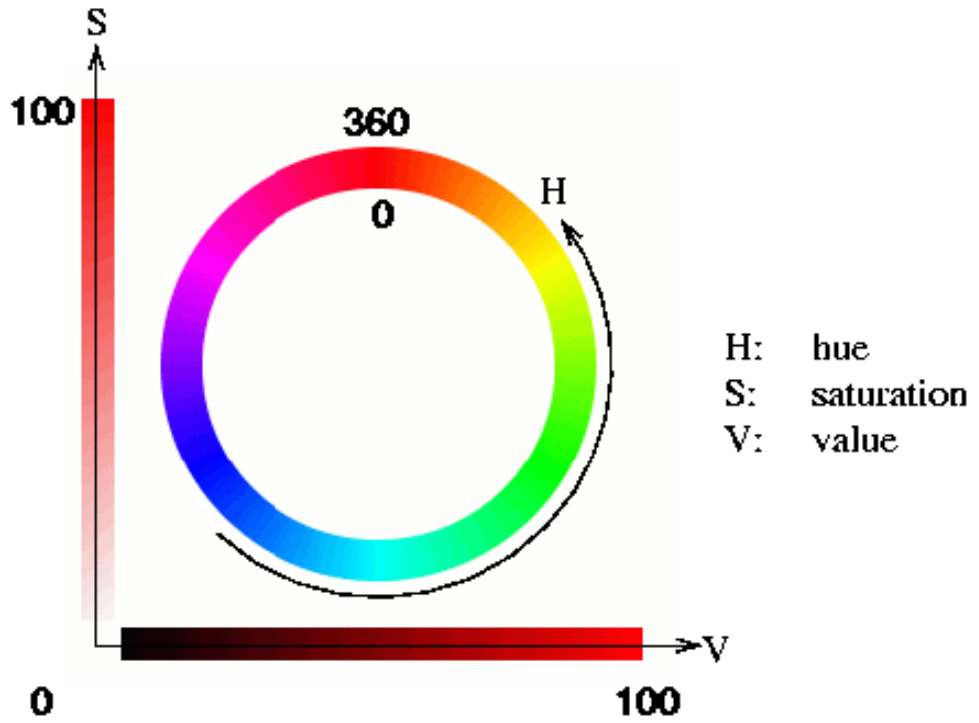
# Sólido de cor HSV



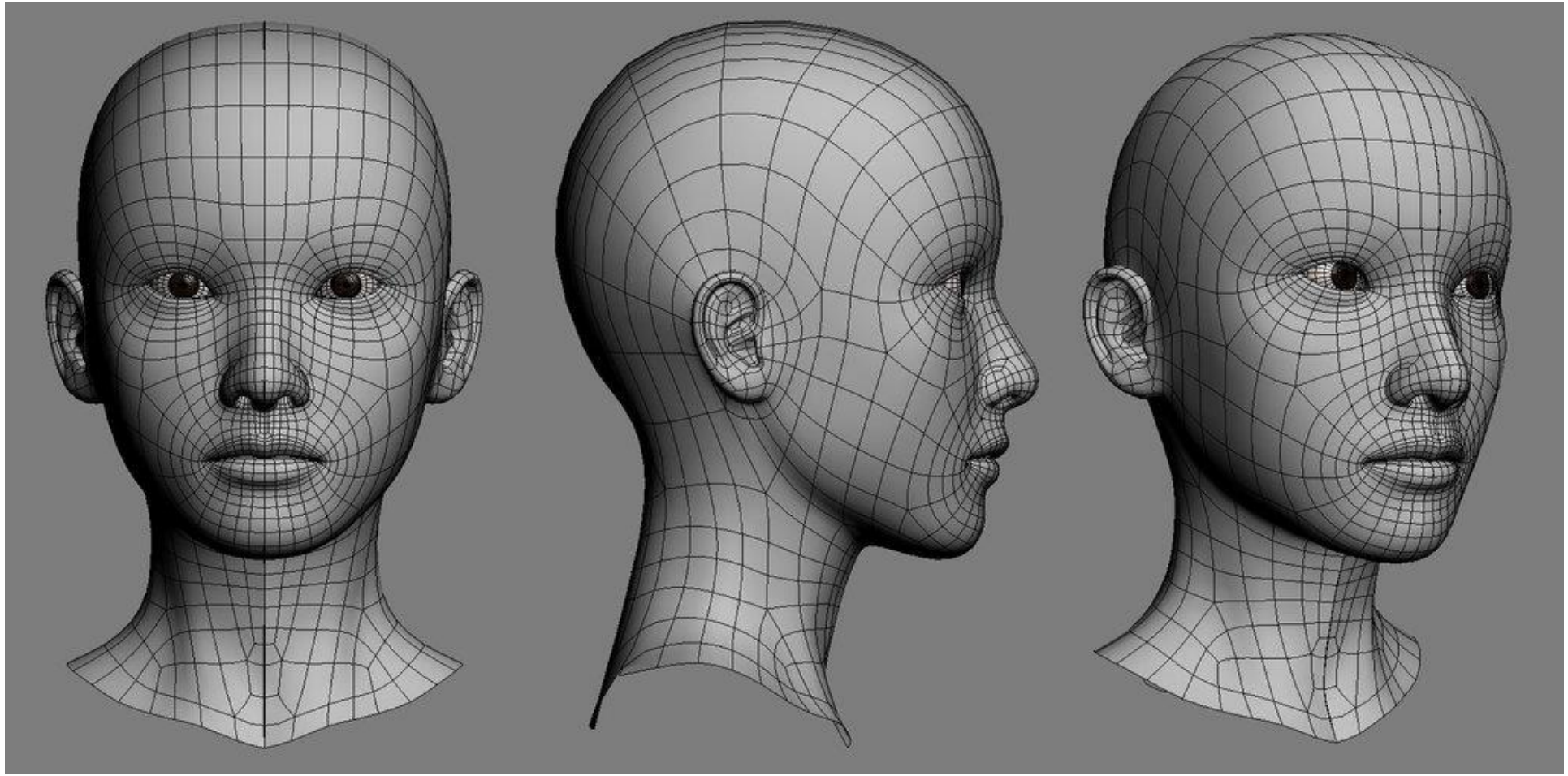



# Sistema HSV

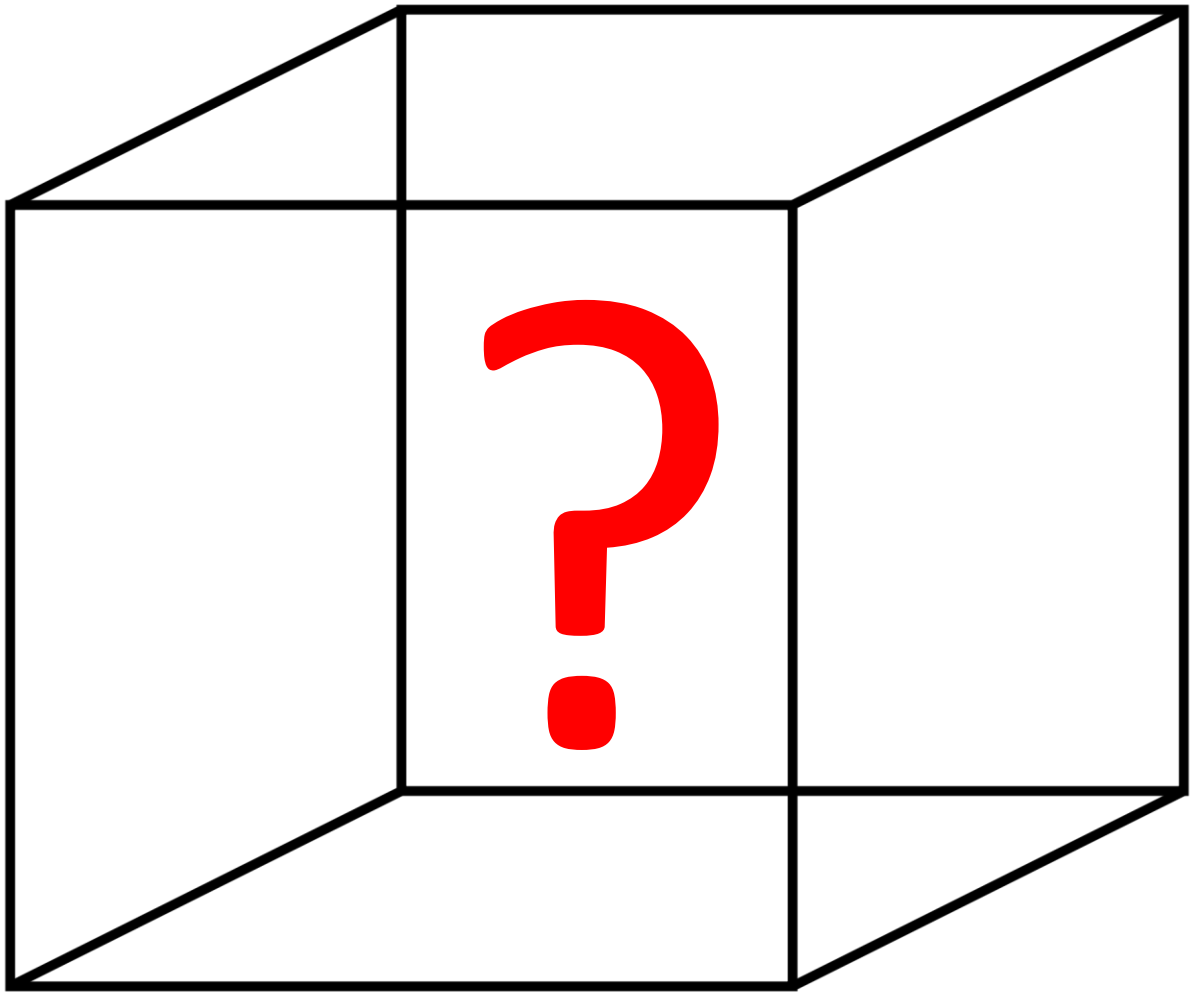
H (Hue) - Matiz da Cor
S (Saturation) - Saturação da cor
V (Value) - Relacionado com o brilho



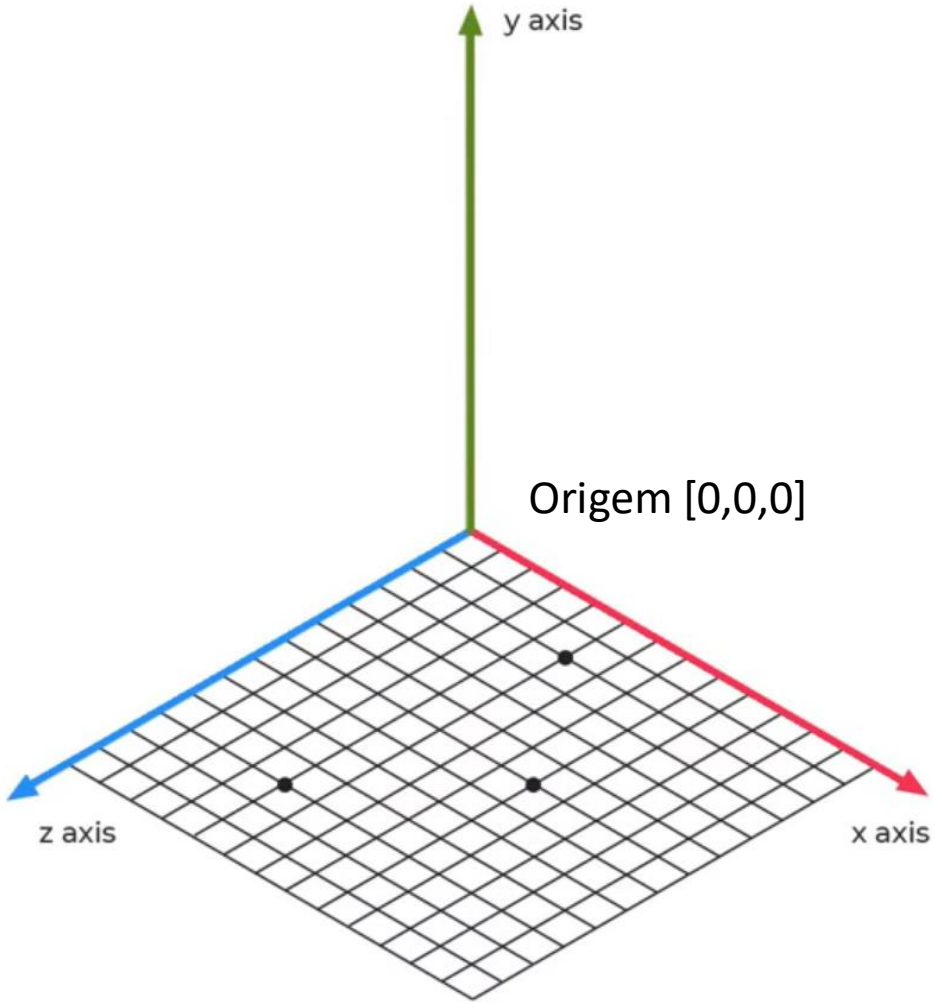
# Modelagem 3D



# Como são formados os objetos em 3D?



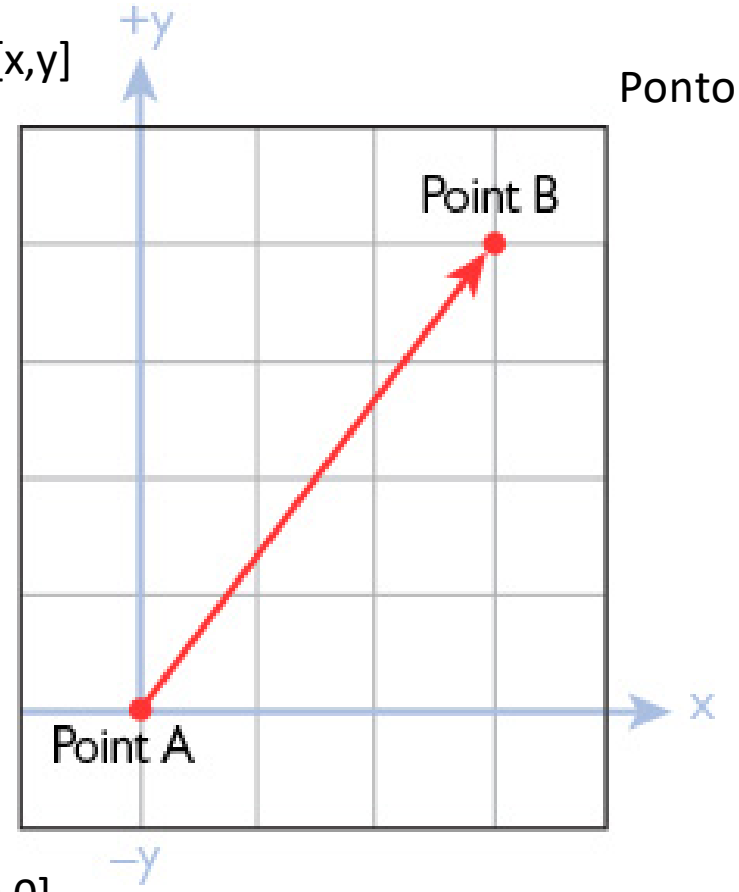
# Como são formados os objetos em 3D?




Origem [0,0,0]

Coordenadas 3D - [x,y,z]

Coordenadas 2D - [x,y]

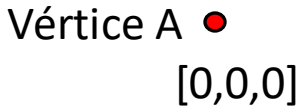



Ponto B [3,4]

Ponto A [0,0]
Origem

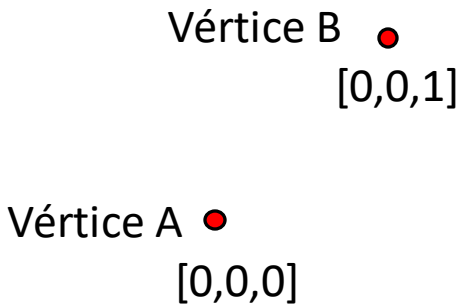
# **Vértices**, Arestas, Faces...

Vértice: Um ponto com informação de suas coordenadas (geralmente no espaço 3D) .



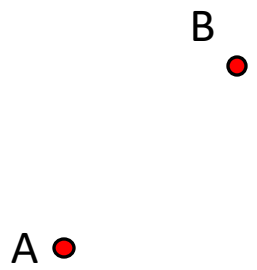

1 ponto

# **Vértices**, Arestas, Faces...



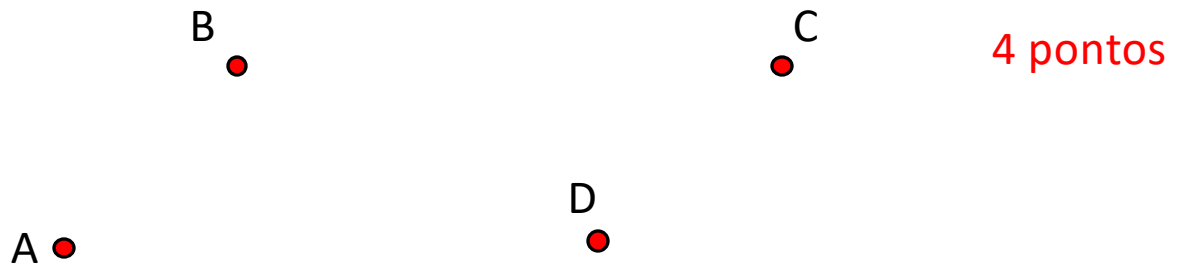

2 pontos

# **Vértices**, Arestas, Faces...

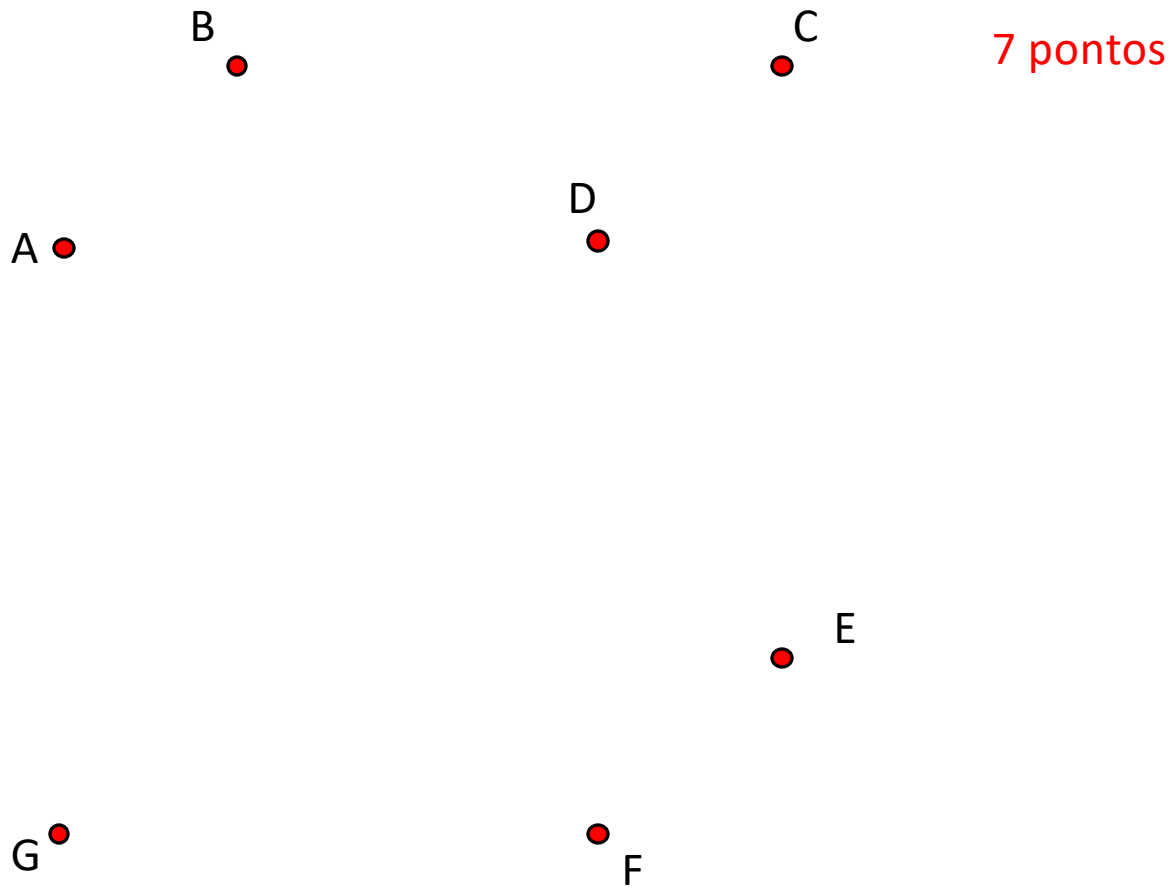



2 pontos

# **Vértices**, Arestas, Faces...

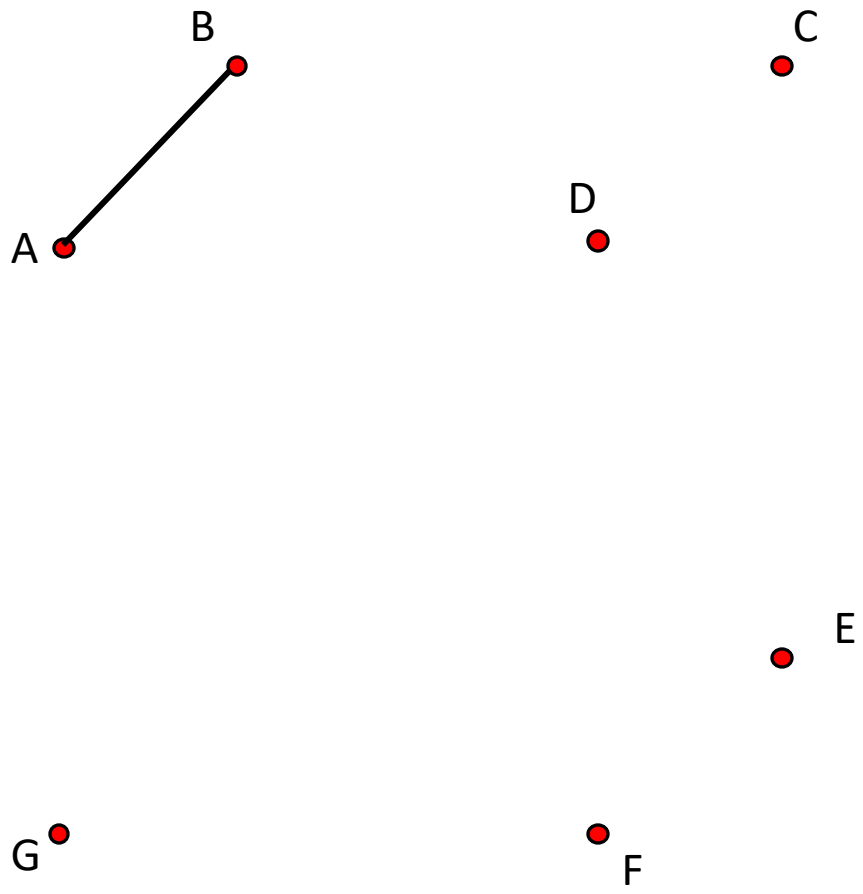


# **Vértices**, Arestas, Faces...



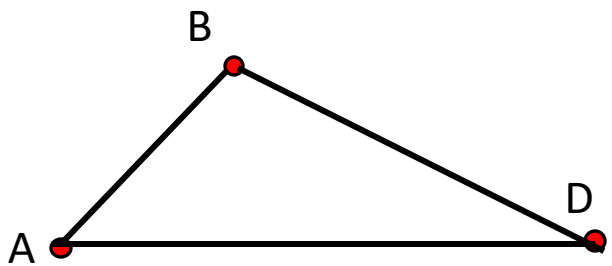
# Vértices, **Arestas**, Faces...

Aresta: Uma conexão entre dois vértices




7 pontos
1 aresta

# Vértices, **Arestas**, Faces...




C

E

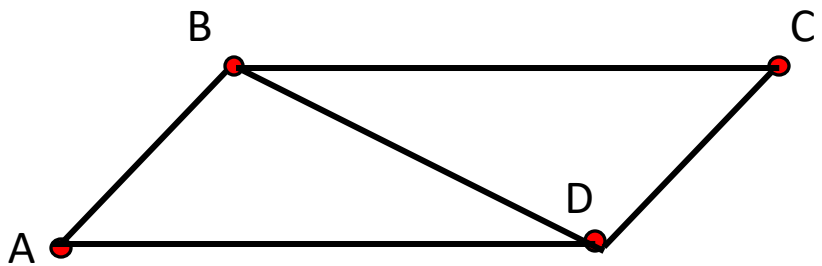
G

F

7 pontos

3 arestas

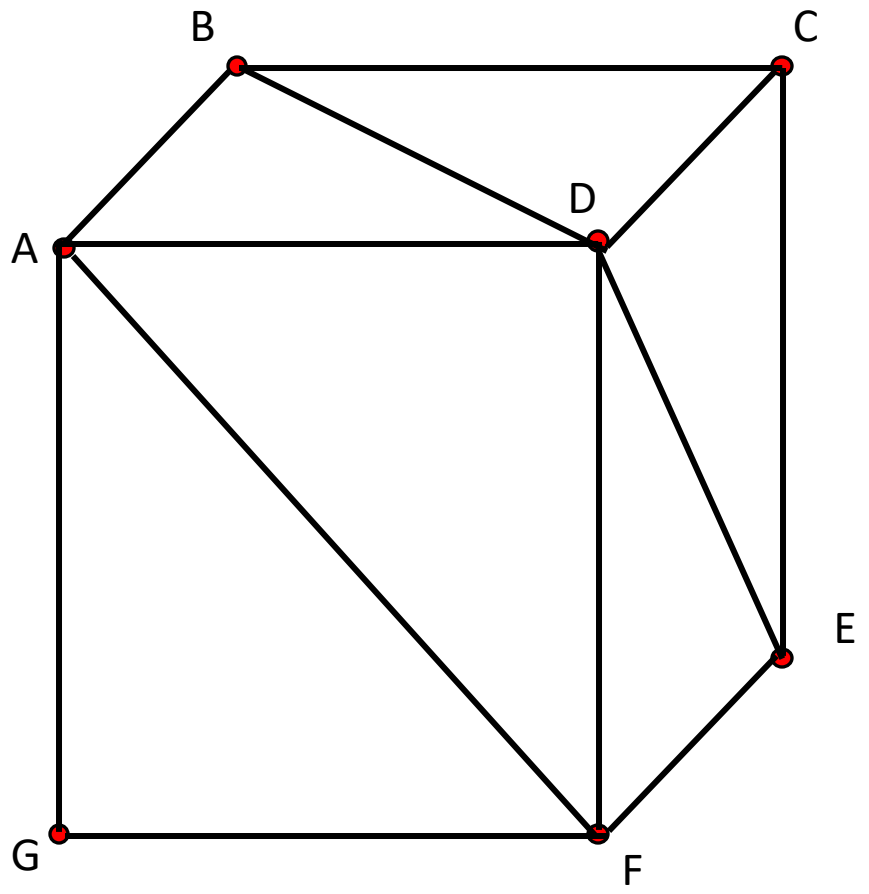
# Vértices, **Arestas**, Faces...




7 pontos
5 arestas

# Vértices, Arestas, **Faces**...

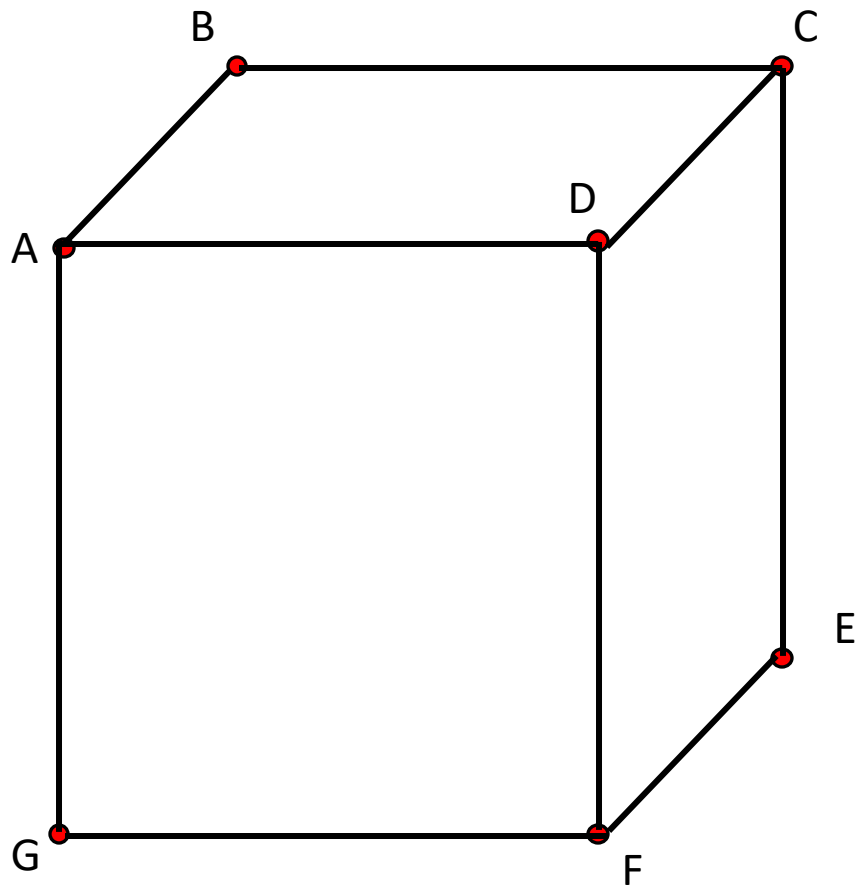
Face: Um conjunto fechado de arestas, na qual a **face triangular tem 3 arestas**, e a quadrangular 4 arestas (quad)




7 pontos
12 arestas

# Vértices, Arestas, **Faces**...

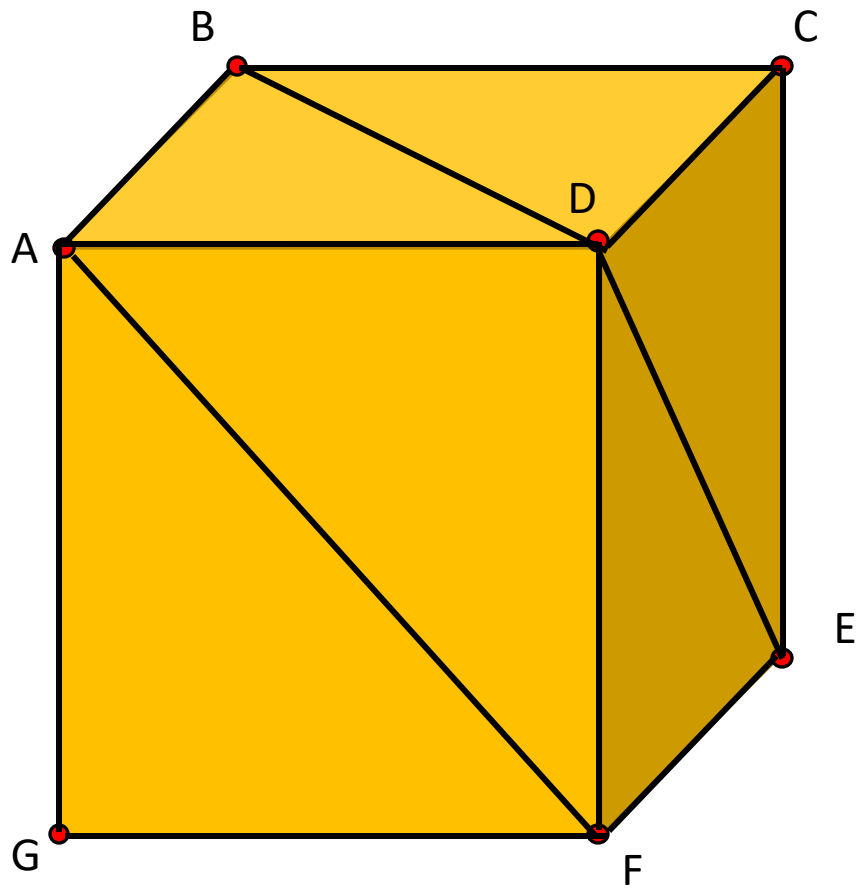
Face: Um conjunto fechado de arestas, na qual a face triangular tem 3 arestas , **e a quadrangular 4 arestas (quad)**




7 pontos
9 arestas

# Vértices, Arestas, **Faces**...

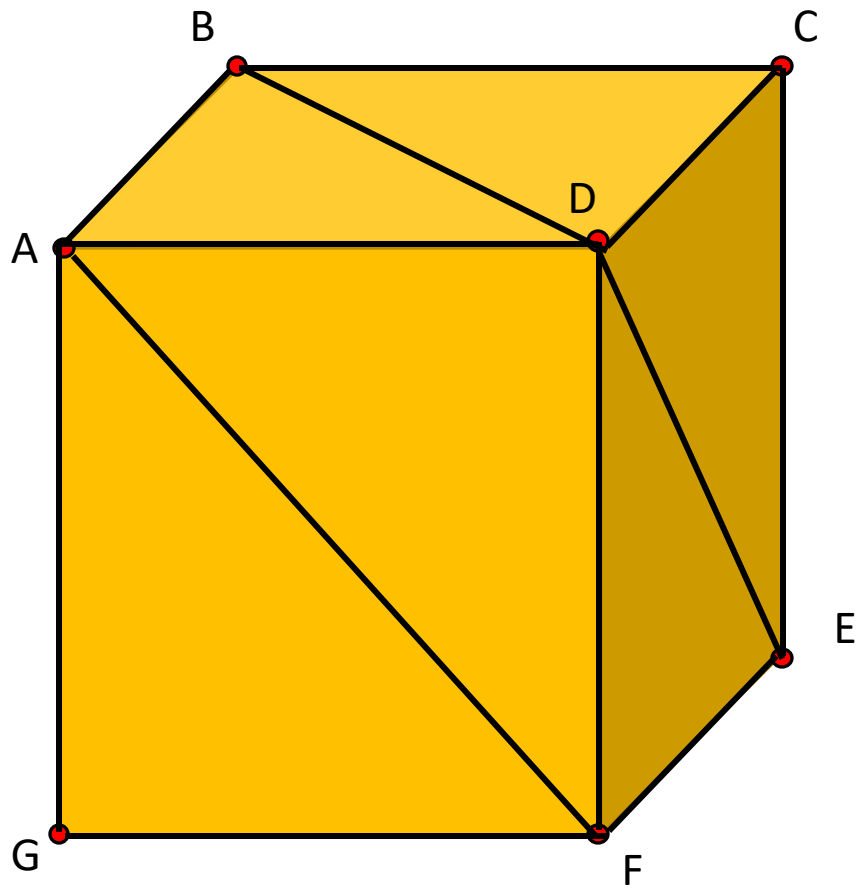
Face: Um conjunto fechado de arestas, na qual a face triangular tem 3 arestas , e a quadrangular 4 arestas (quad)




7 pontos
9 arestas
6 faces

# Vértices, Arestas, **Faces**...

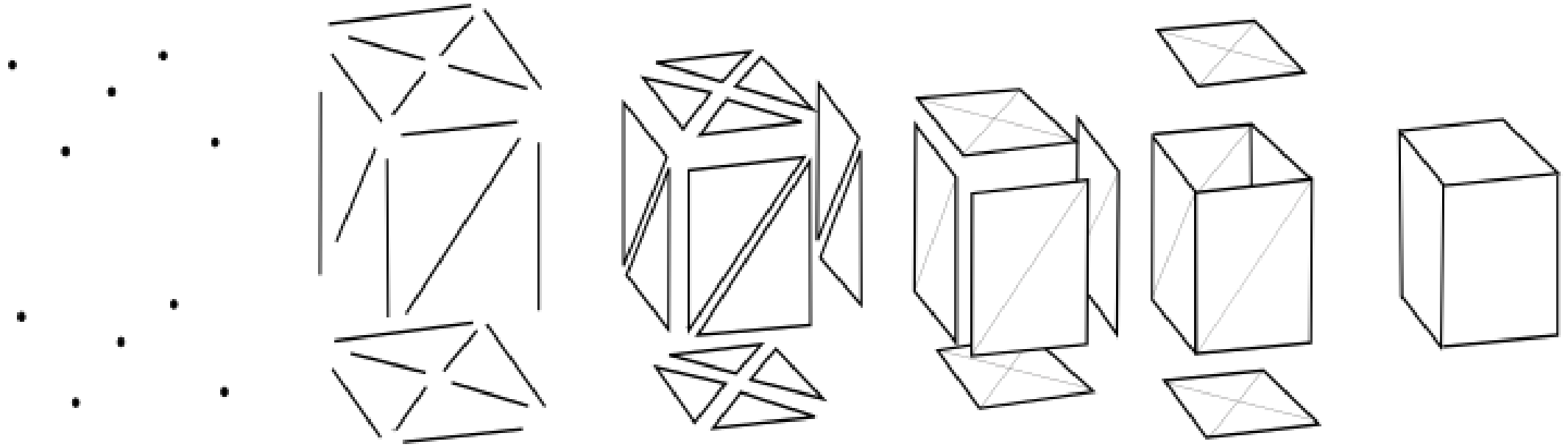
Face: Um conjunto fechado de arestas, na qual a face triangular tem 3 arestas , e a quadrangular 4 arestas (quad)




7 pontos
9 arestas
6 faces
3 polígonos

Polígono:
Um polígono é um conjunto coplanar de faces.

# Como são formados os objetos em 3D?



vertices (vértices)

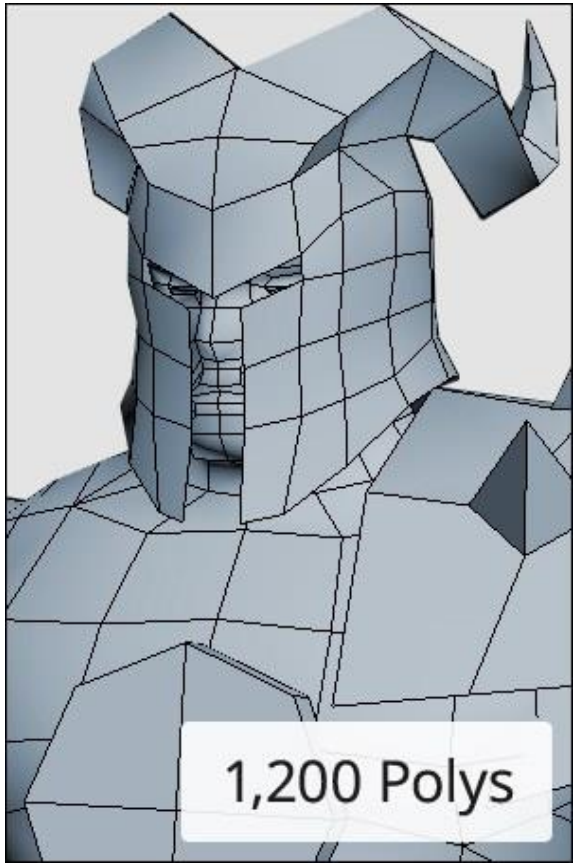
edges (arestas)

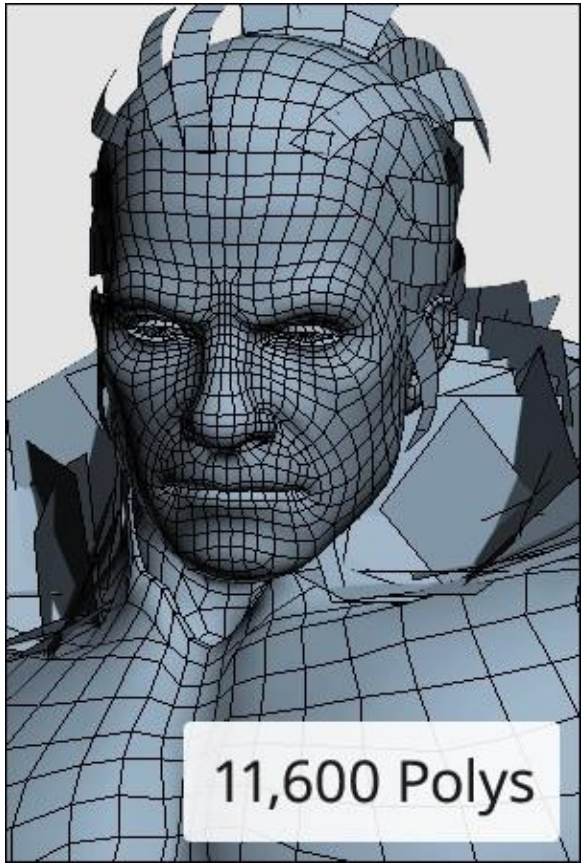
faces (faces)

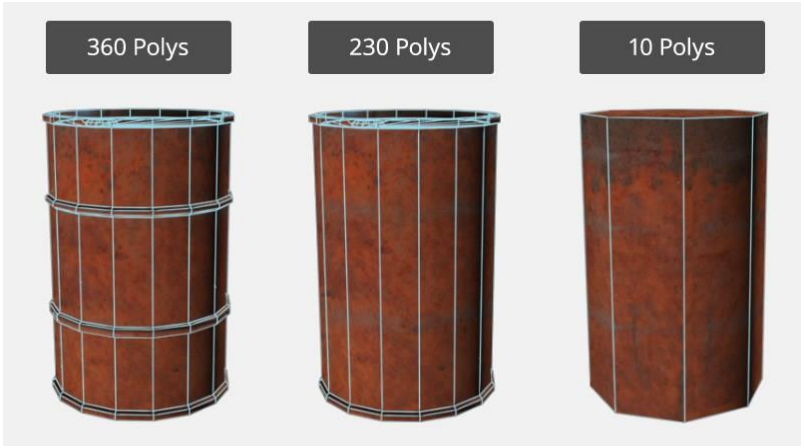
polygons (polígonos)

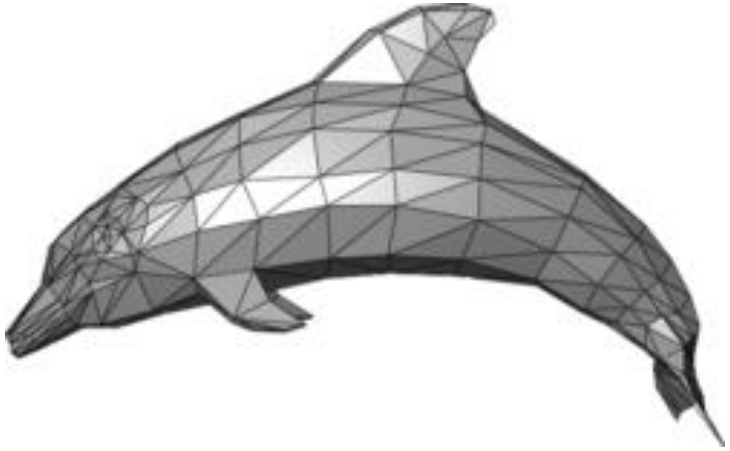
surfaces (superfícies)

# Malha Poligonal (Mesh)

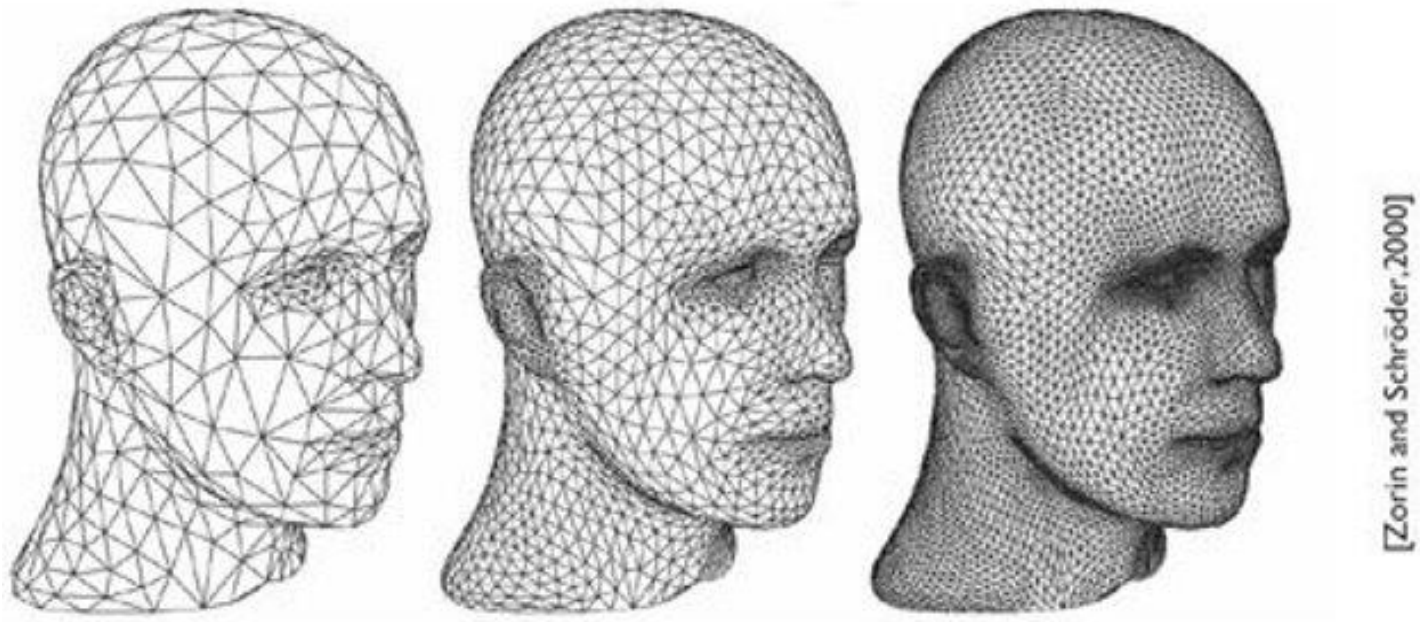


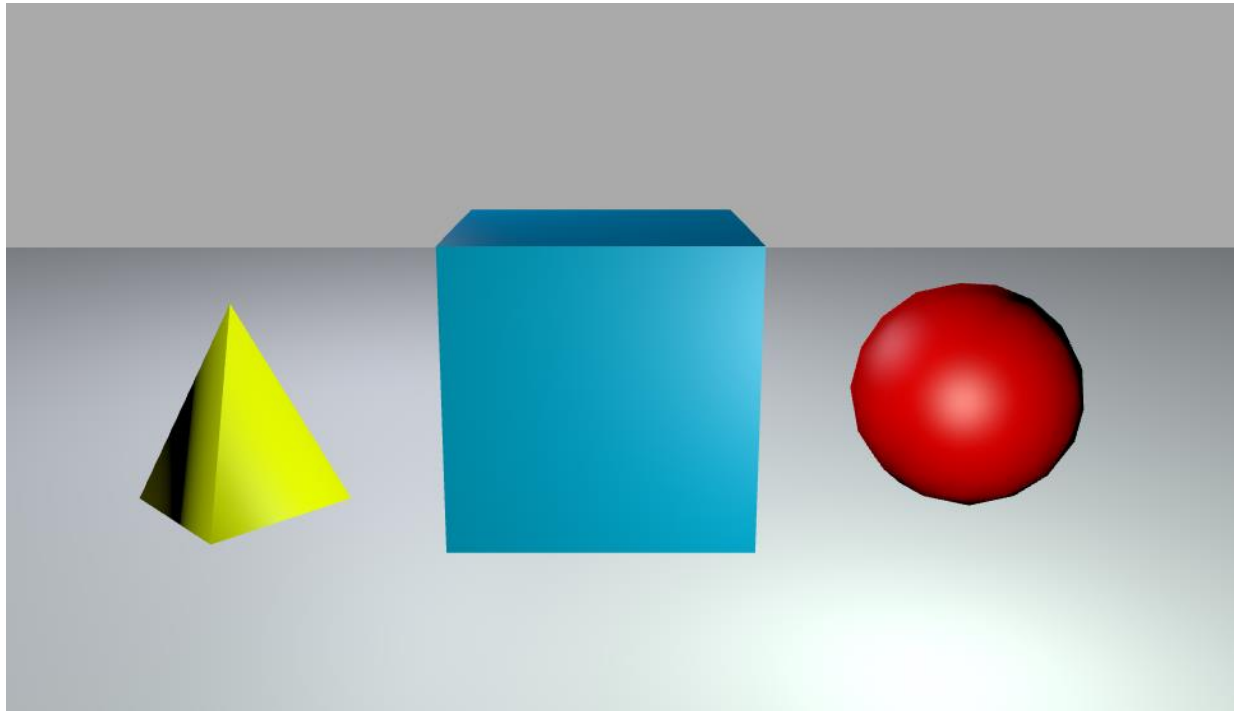



# Malha Poligonal (Mesh)

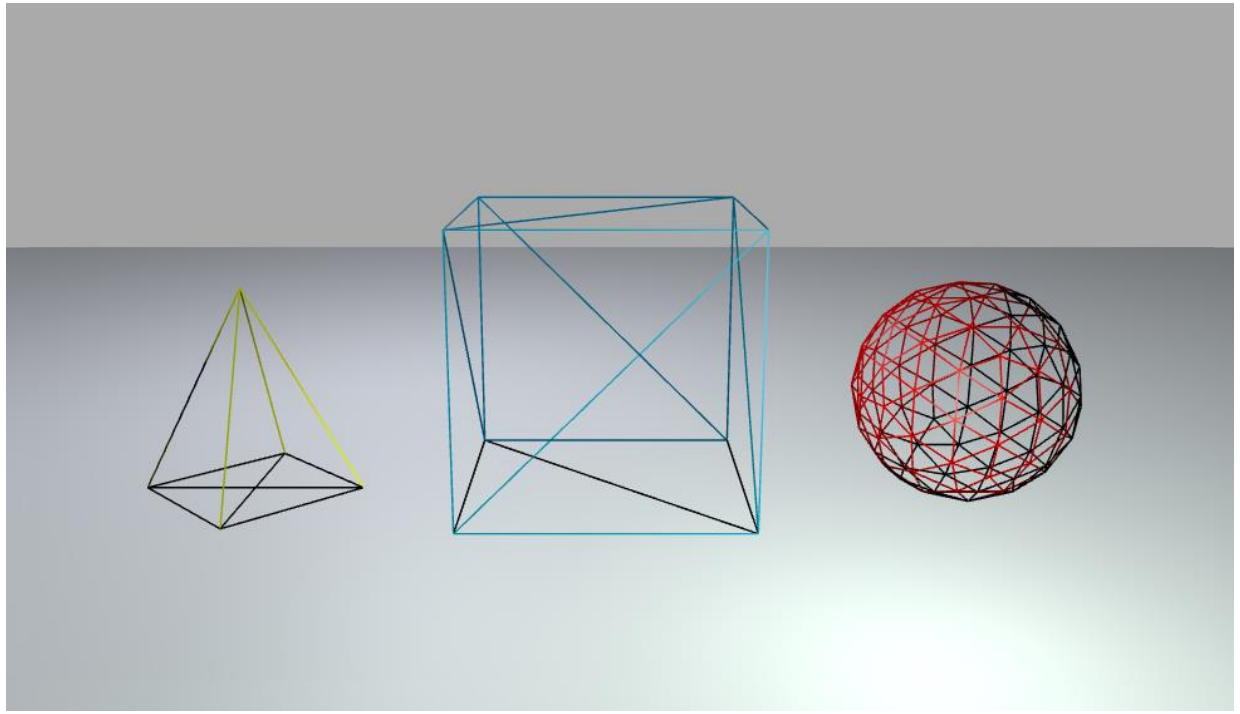


[Zorin and Schröder, 2000]

# Visualização: Wireframe



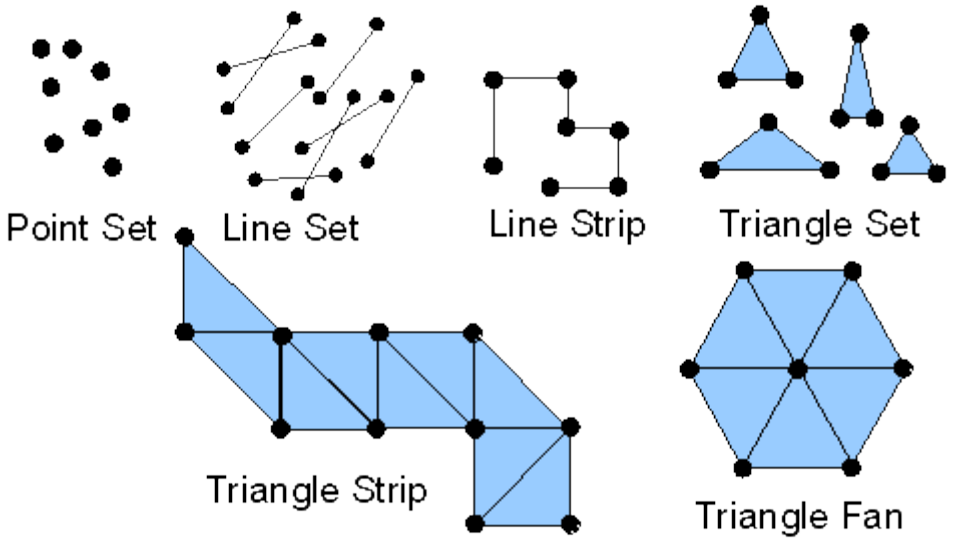
# Visualização: Wireframe



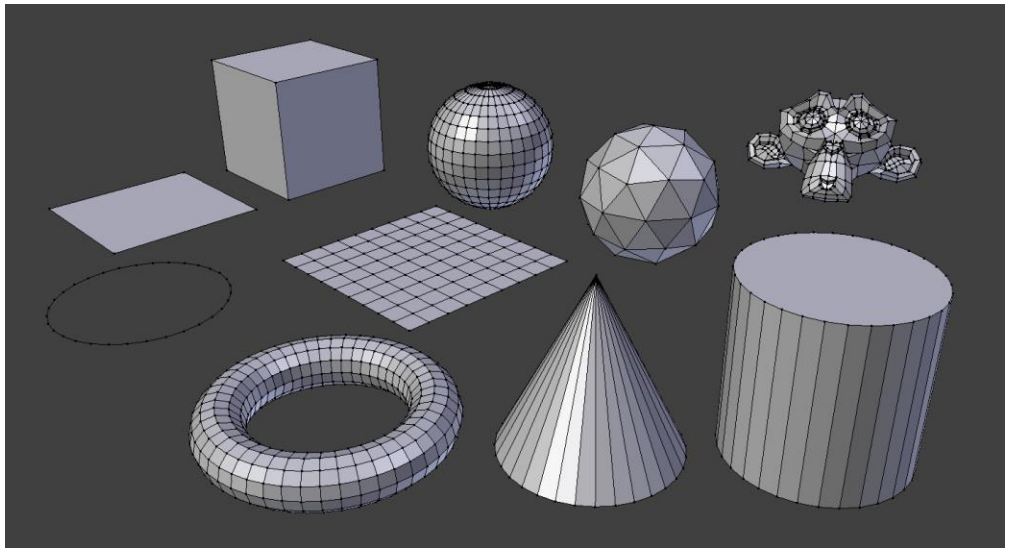
Visualização apenas das arestas dos objetos

"Estrutura de arame"

# Primitivas Geométricas

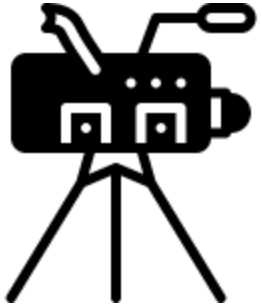



2D

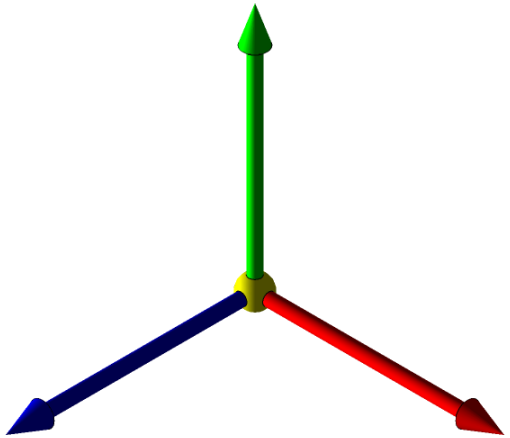


3D

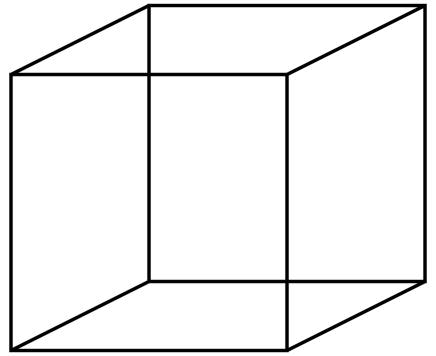
# Elementos de uma cena 3D



Câmera:
Perspectiva do usuário

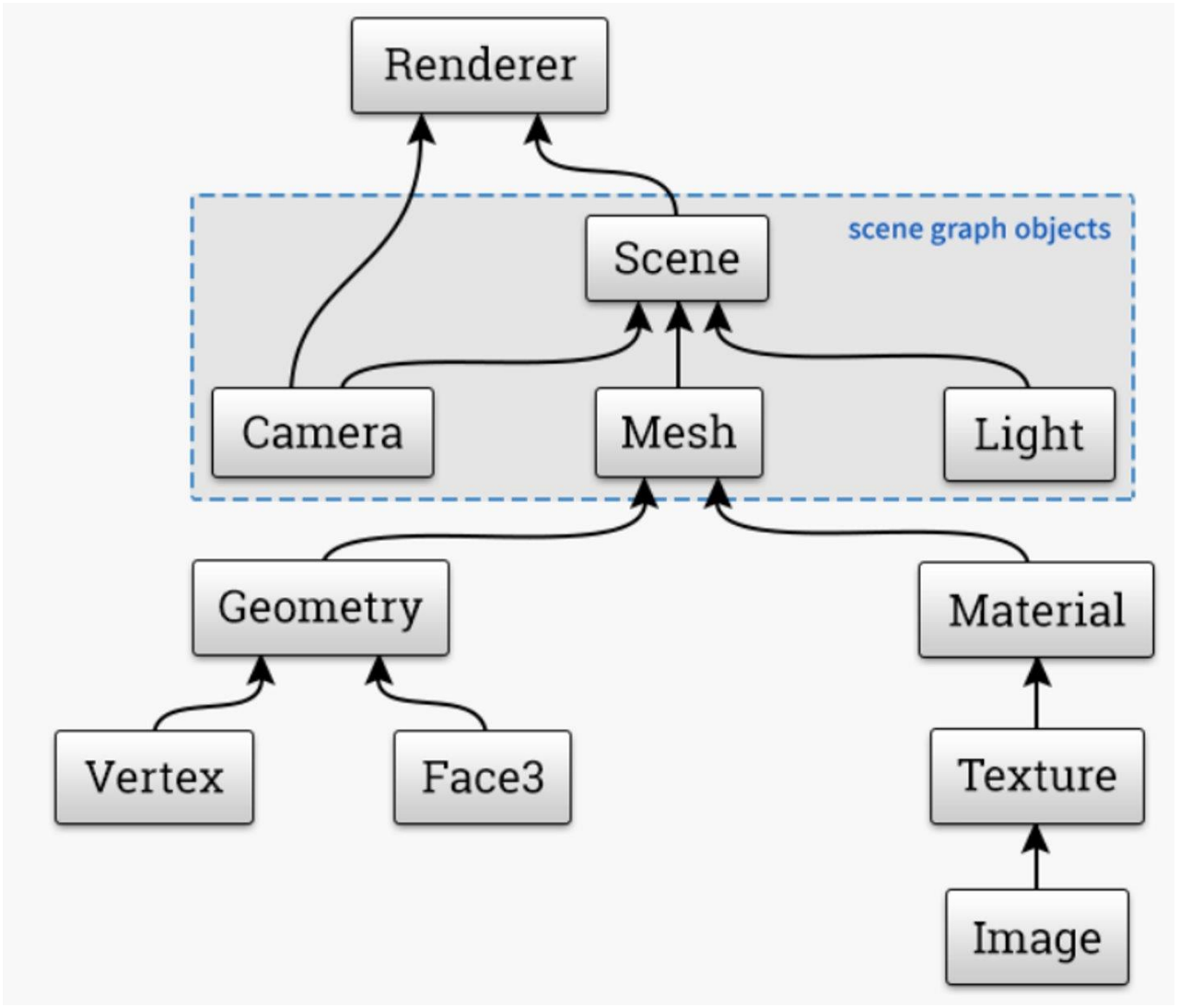


Eixo de coordenadas:
Referencial para posicionamento dos elementos

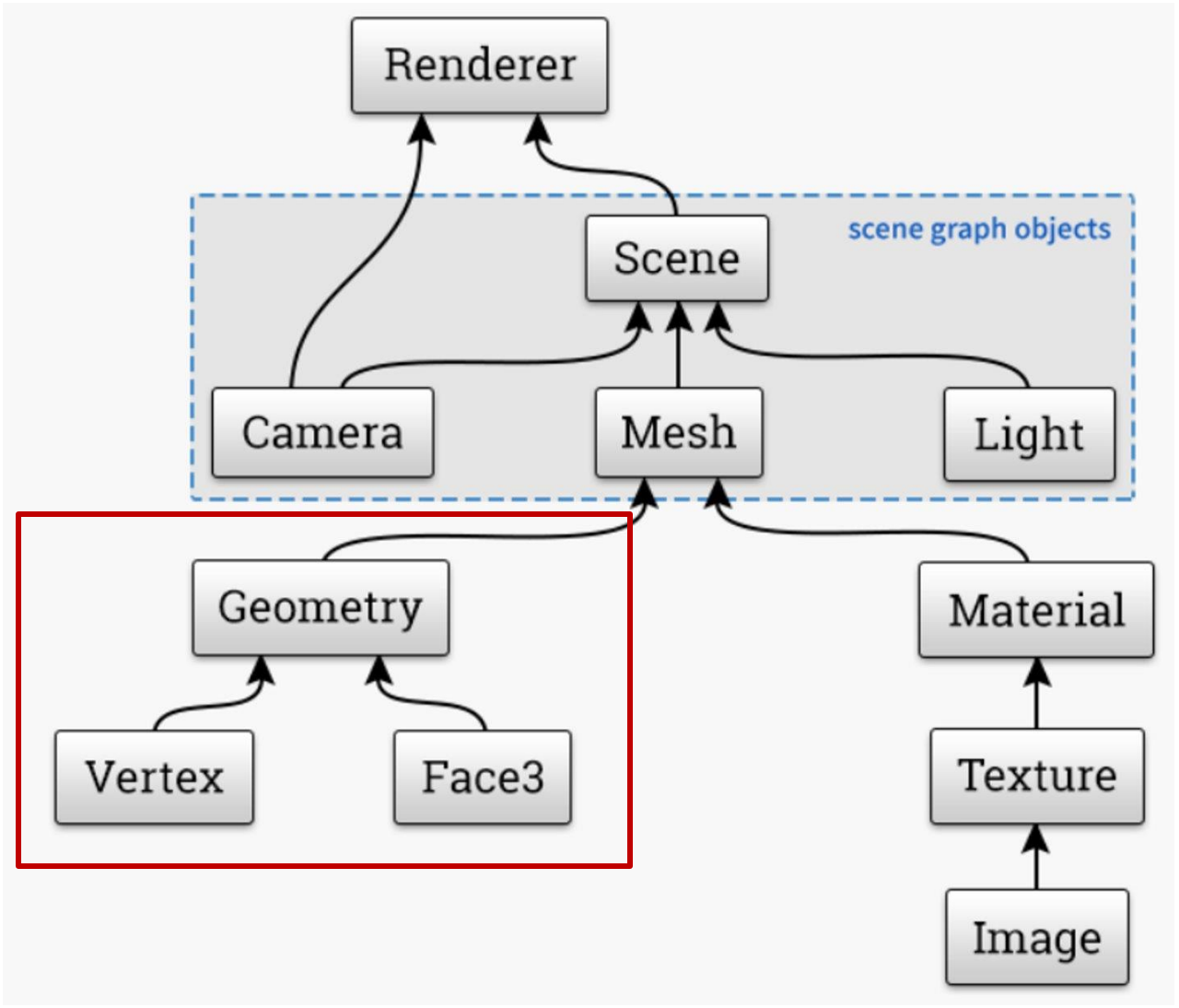


Objeto 3D:
Objeto de interesse

# Fluxo de processamento Three.JS

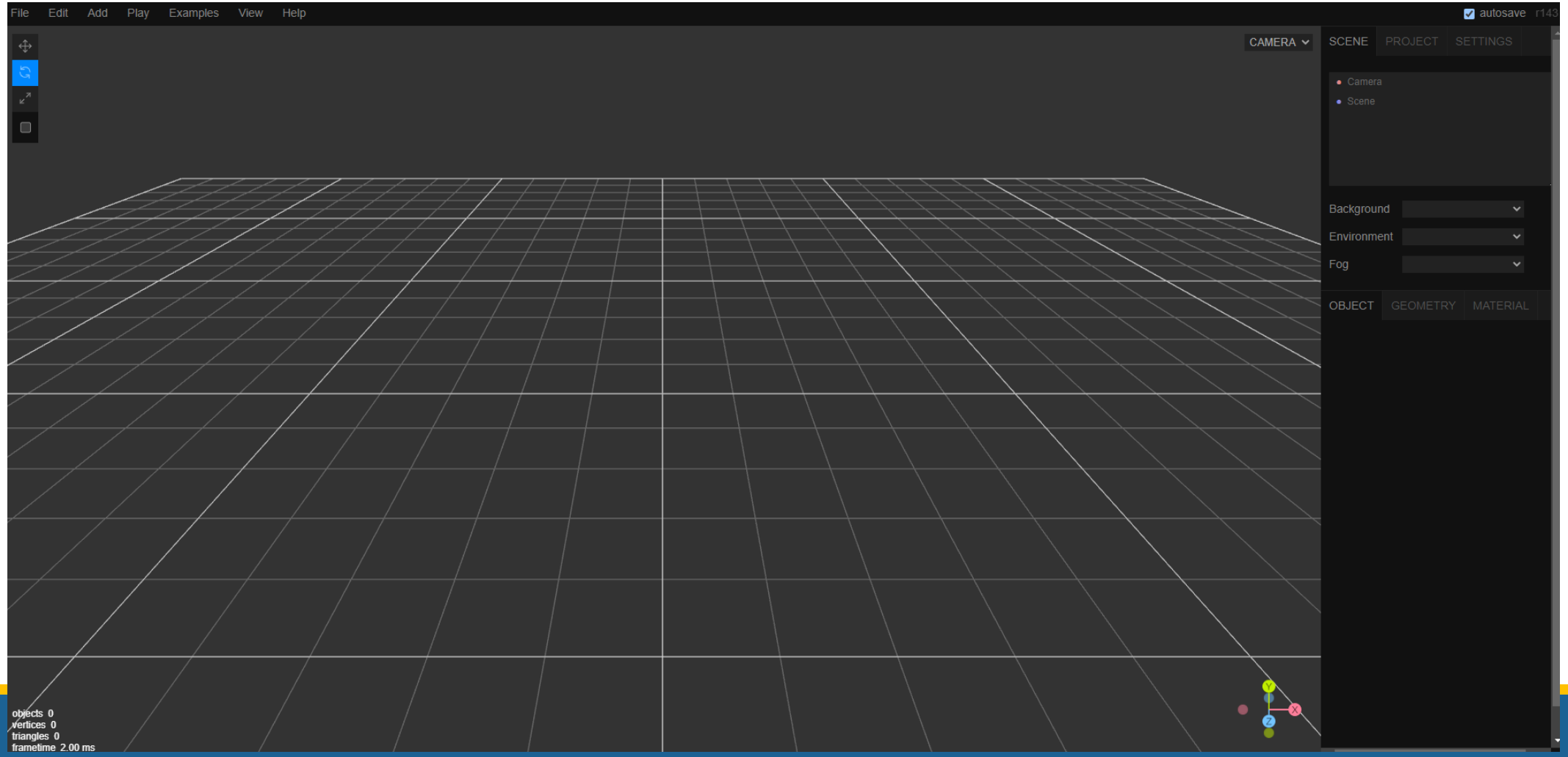


# Fluxo de processamento Three.JS

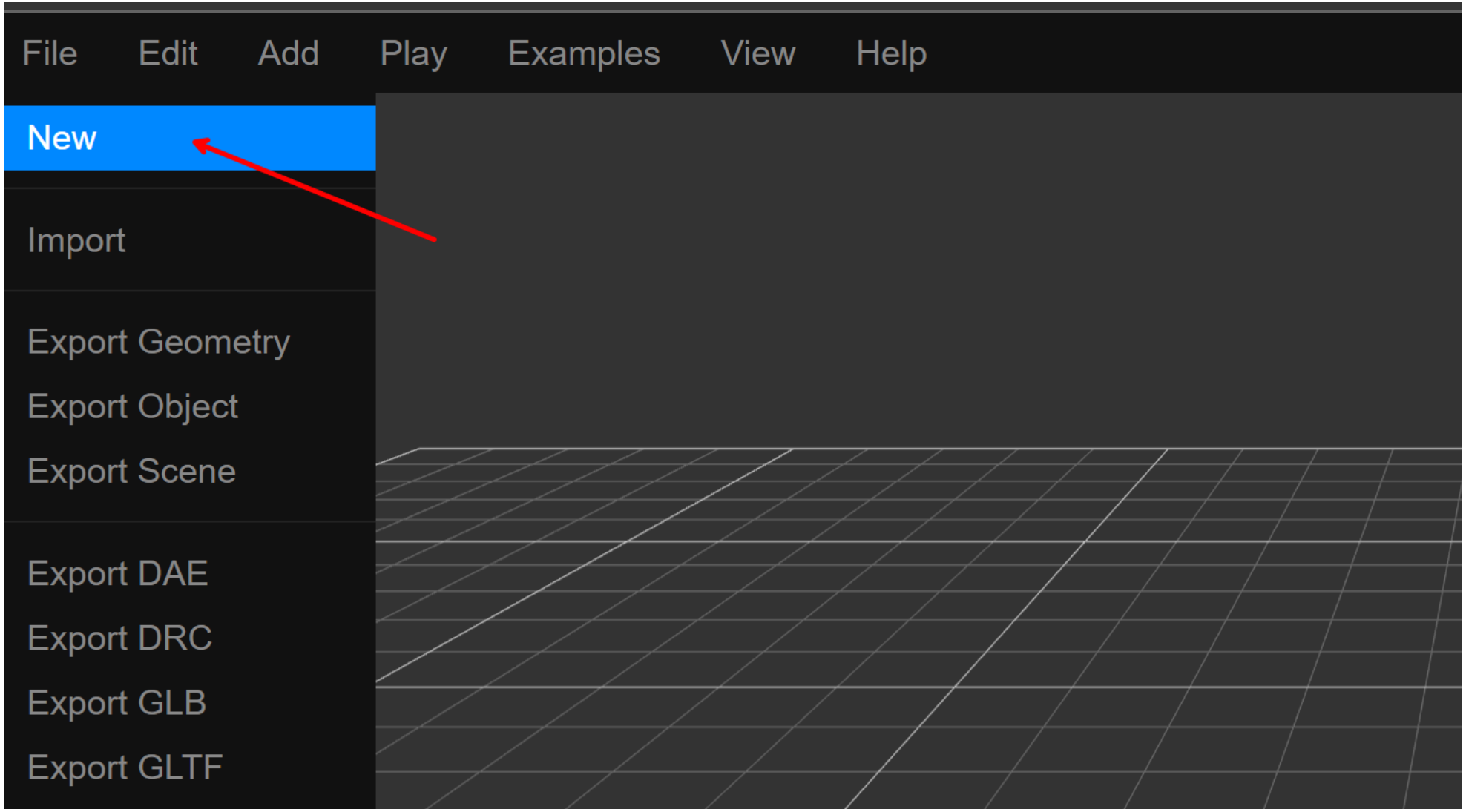


# Praticando no Three.js

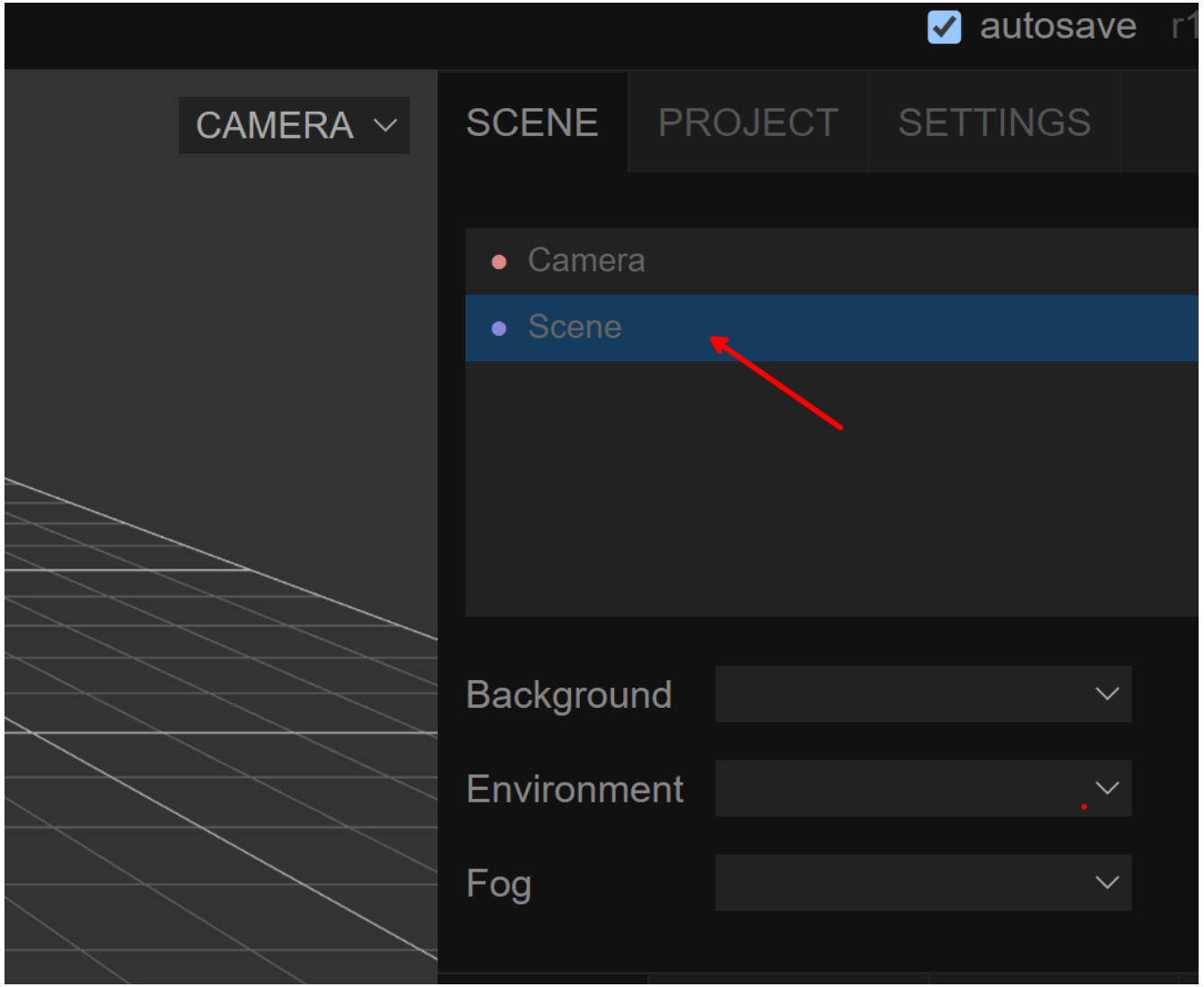
https://threejs.org/editor/



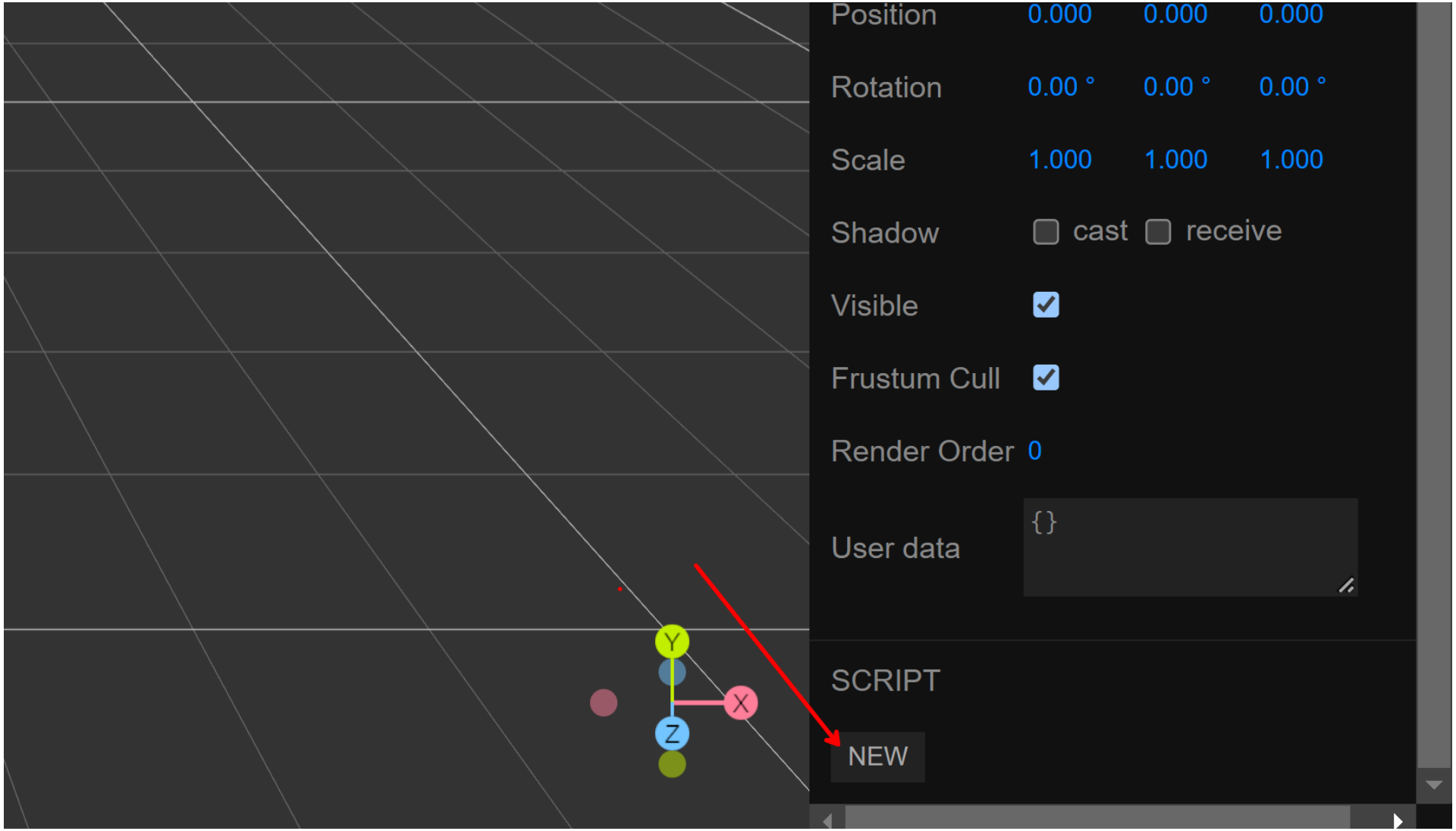
# Novo arquivo



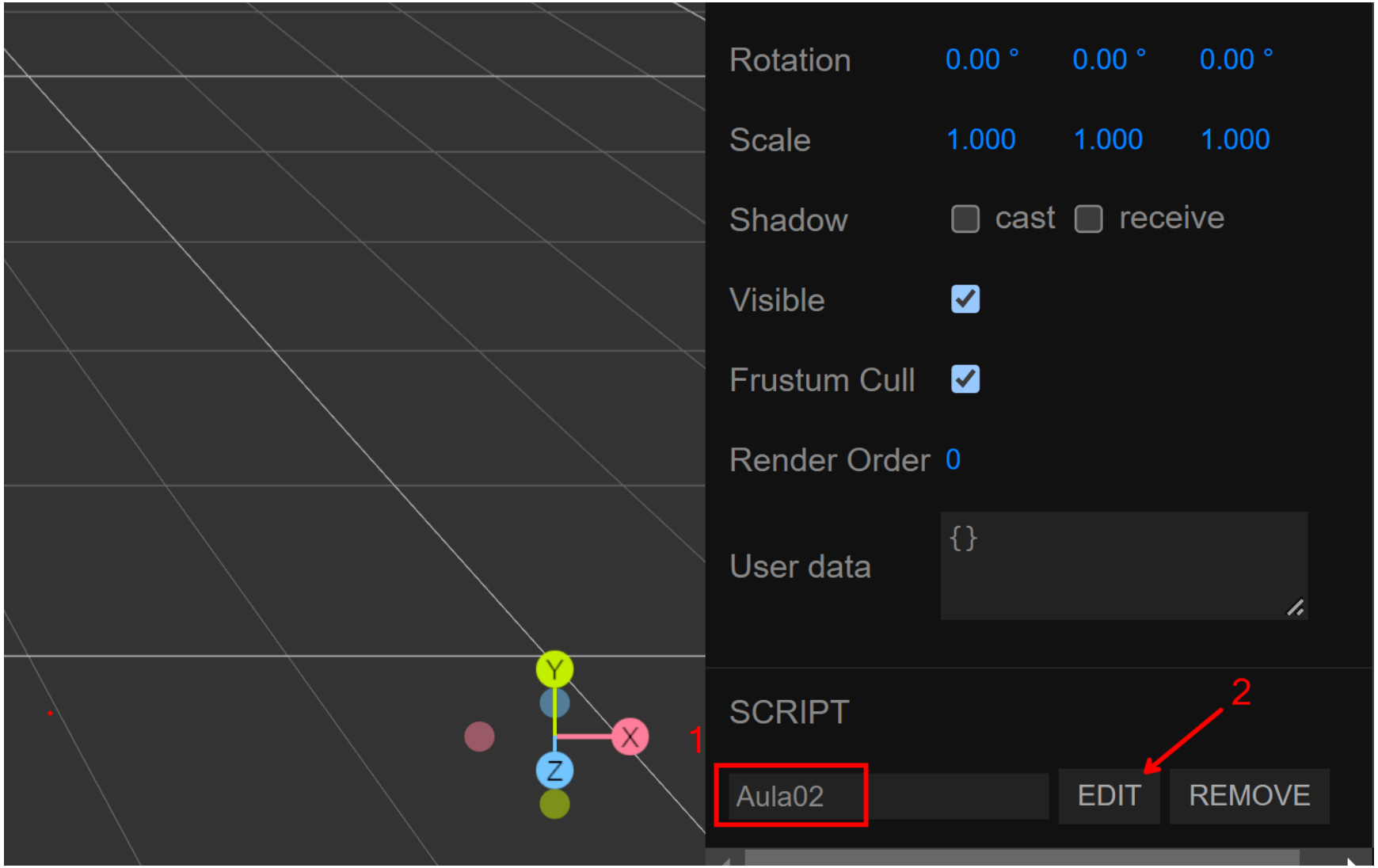
# Clicar em "Scene"



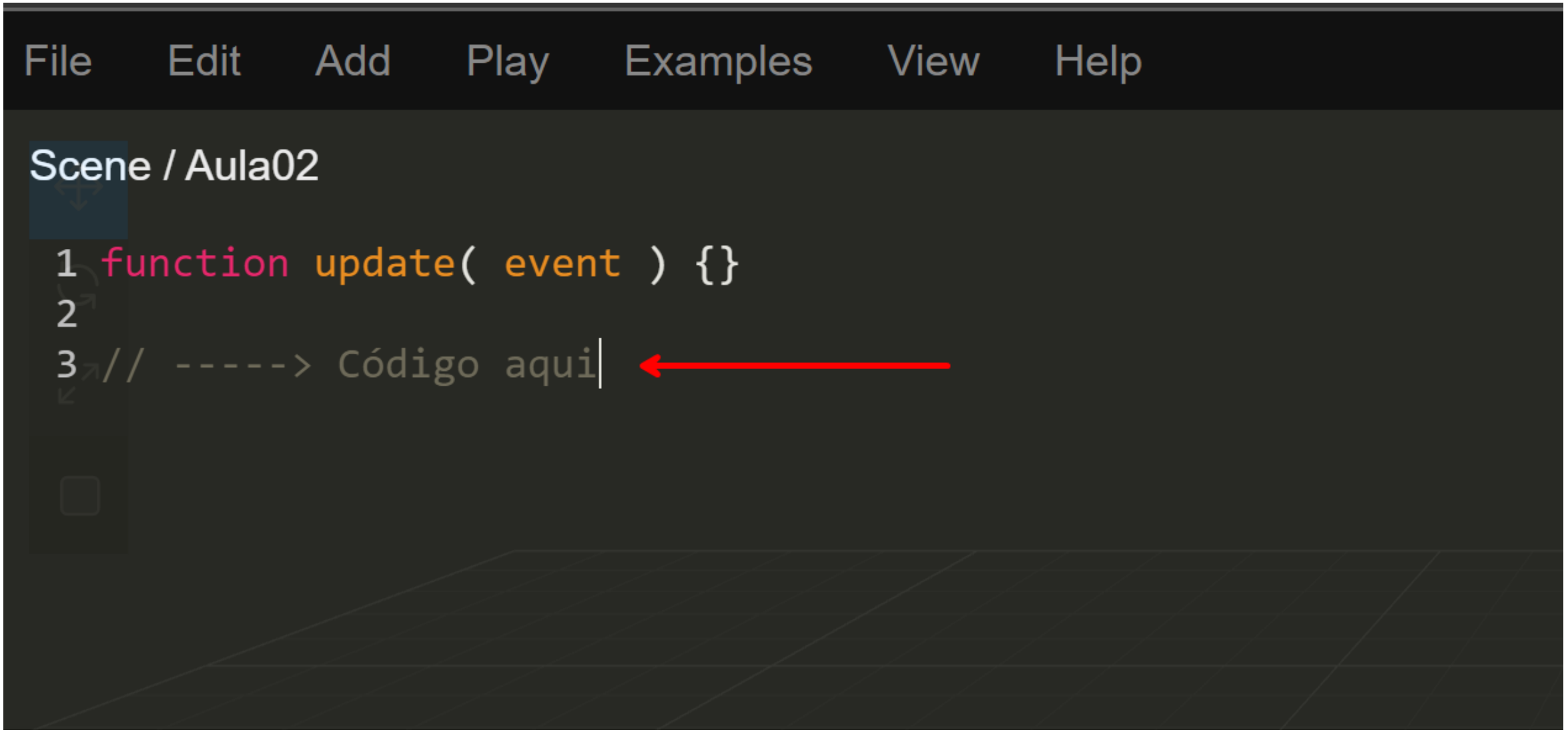
# Clicar em "NEW"



# Nome do arquivo e depois "EDIT"



# Área para escrever o código



# Exemplo básico

```
function update( event ) {}

// Criando Poligono a partir dos Vértices

const axesHelper = new THREE.AxesHelper(5); // Cria um guia para os eixos cartesianos
scene.add(axesHelper);                      // Adiciona os eixos na cena


// Cria um triangulo simples
//  vx = [ x, y, z];
var v = []; // criado apenas para facilitar a concatenação
var v0 = [-1.0,-1.0, 1.0];
var v1 = [ 1.0,-1.0, 1.0];
var v2 = [ 1.0, 1.0, 1.0];

// Inicializa o objeto que será utilizado para armazenar nosso vertices
const geometry = new THREE.BufferGeometry();
// Para criar um poligono/triangulo é necessário colocar as coordenadas dos 3 vertices
const vertices = new Float32Array(v.concat(
    v0,v1,v2
//  v2,v3,v0 // Cada linha 1 triangulo
));

// itemSize = 3 pois há 3 valors (componentes/coordenadas) por vértice
geometry.setAttribute( 'position', new THREE.BufferAttribute( vertices, 3 ) );
const material = new THREE.MeshBasicMaterial( { color: 0xff0000 } ); // Material Básico RGB Hexadecimal
const mesh = new THREE.Mesh( geometry, material );
scene.add(mesh);
//console.log(mesh);
//mesh.material.wireframe = true; // mostra Wireframe (apenas arestas)
```

# Praticando no Three.js

Utilizando o código elaborado em sala:

Crie um código que implemente uma **pirâmide** e um **cubo** (como os da imagem ao lado, podem ser códigos separados), a partir das coordenadas dos vértices como demonstrado em aula.

Em seguida responda as perguntas da atividade.

Obs.: Não é necessário incluir luzes, fundo e etc...