

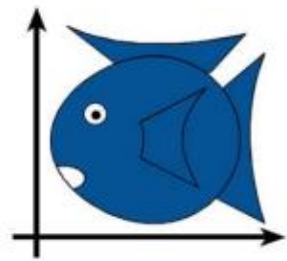
Computação Gráfica

✓ CAMERAS E PERSPECTIVA

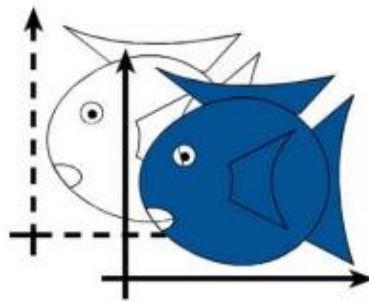
amlucena@cruzeirodosul.edu.br

Na última aula...

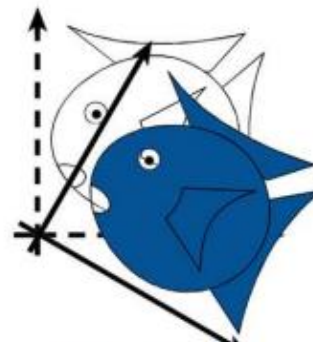
Sendo assim todas as transformações no espaço podem ser obtidas a partir de operações com vetores



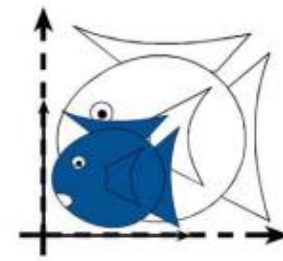
Identidade



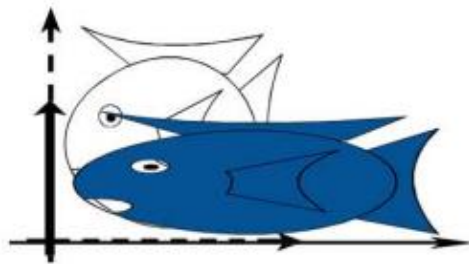
Translação



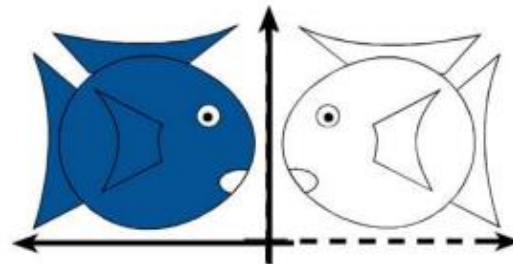
Rotação



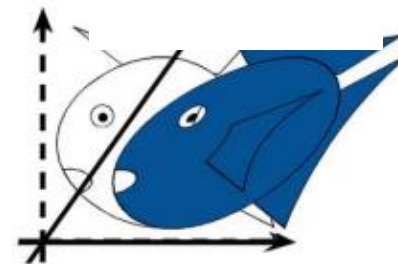
Escala
(Uniforme)



Escala



Reflexão



Cisalhamento

Na última aula...

- Translação

$$\begin{array}{c} \text{Translation in 3D} \\ \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

<https://matrix.resish.com/multiplication.php>

- Escala

$$\begin{array}{c} \text{Scale in 3D} \\ \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

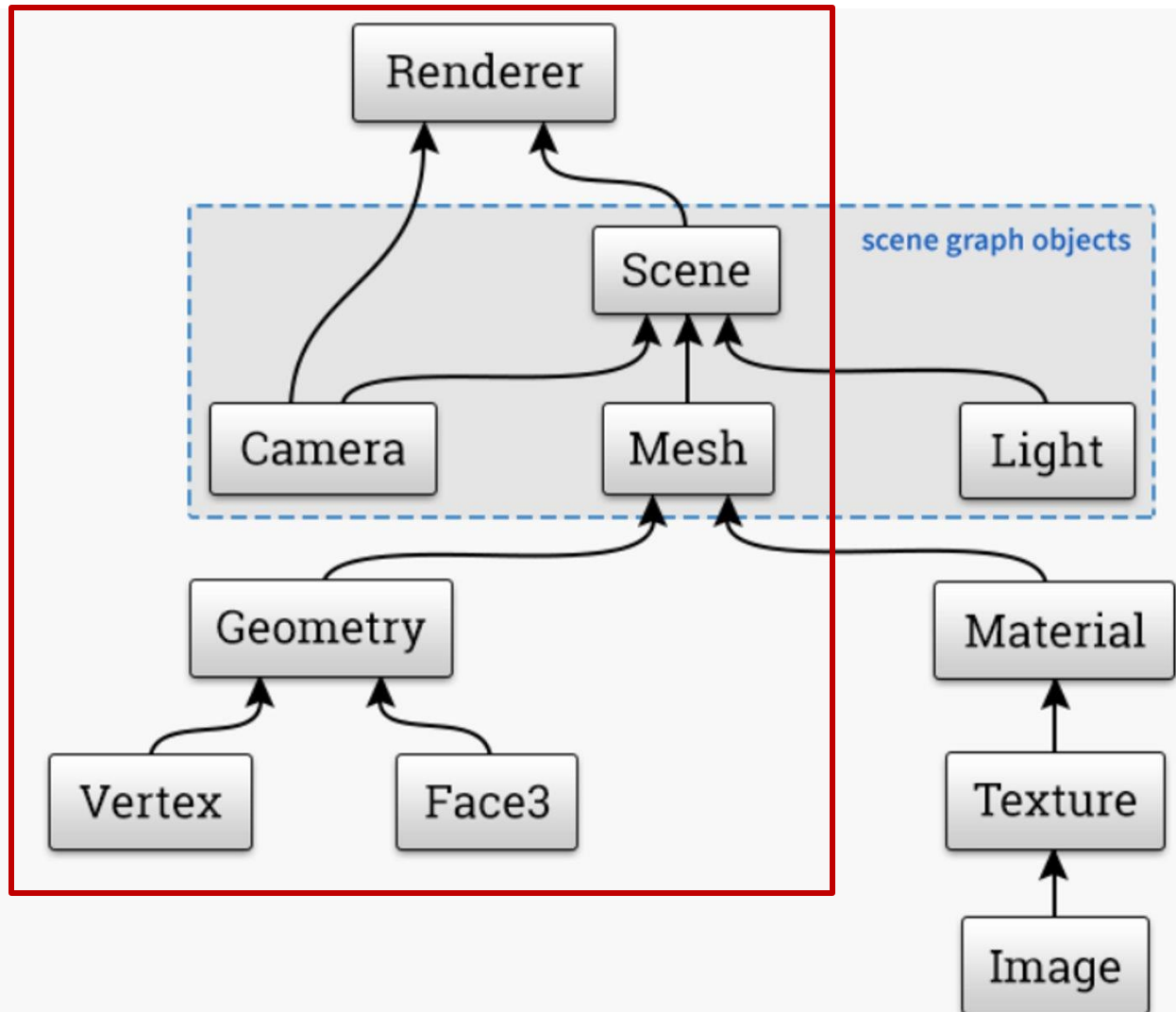
- Rotação (em torno do eixo)

$$\begin{array}{c} \text{X-Rotation in 3D} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

$$\begin{array}{c} \text{Y-Rotation in 3D} \\ \begin{bmatrix} \cos\phi & 0 & \sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

$$\begin{array}{c} \text{Z-Rotation in 3D} \\ \begin{bmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

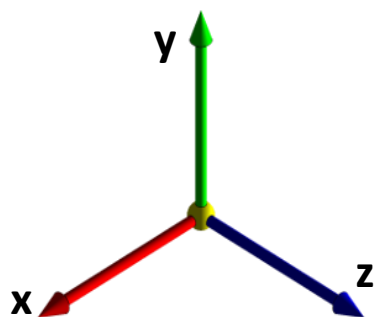
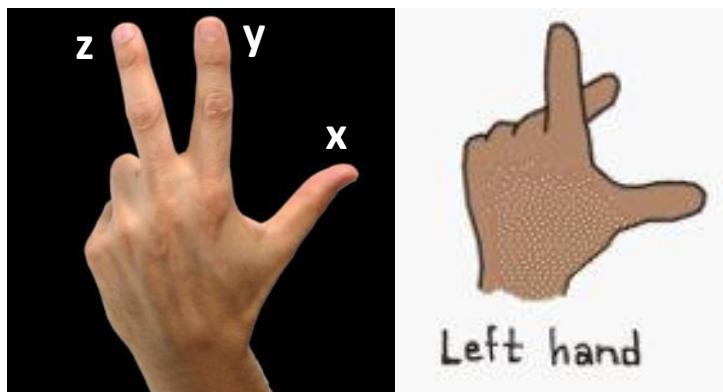
Na última aula...



Alguns complementos

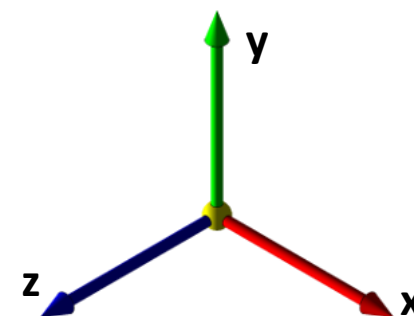
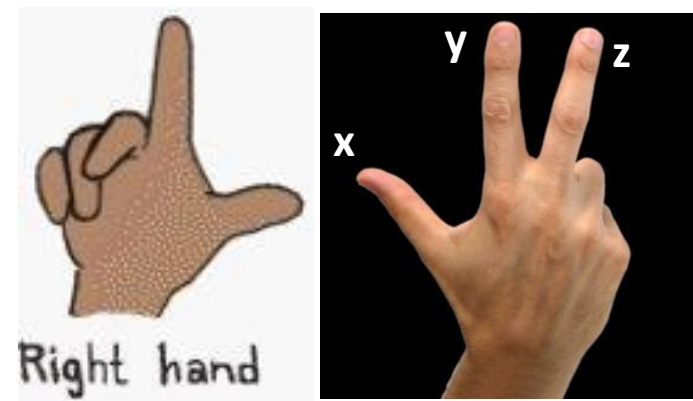
Convenção para o eixo de coordenadas:

Regra da mão esquerda



x

Regra da mão direita

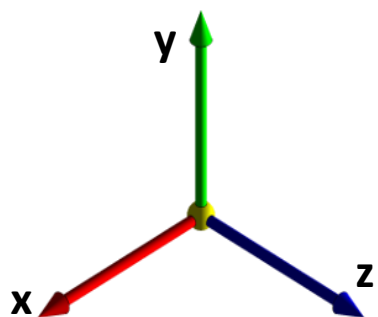
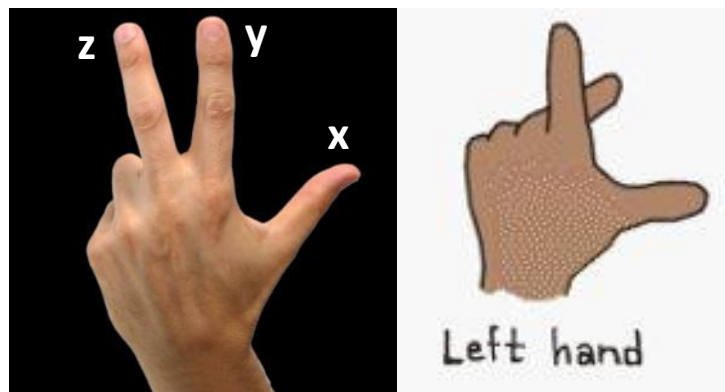


Não dá na mesma?
O lado sentido positivo para
o eixo z muda!

Alguns complementos

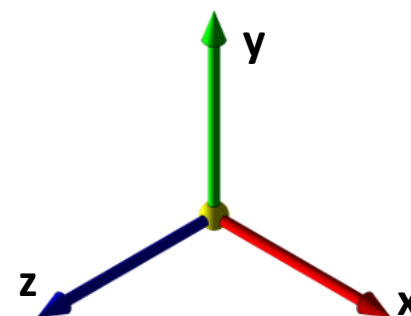
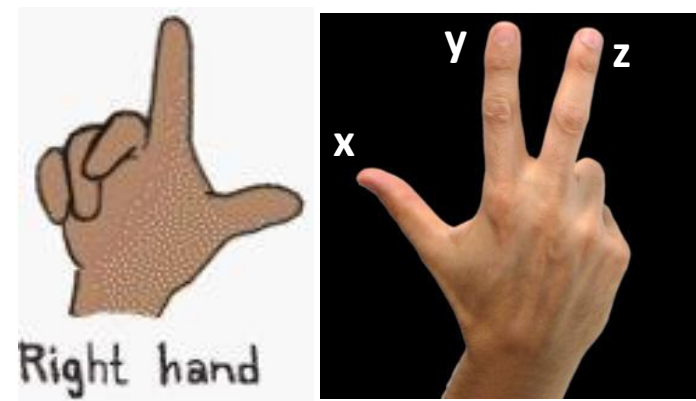
Convenção para o eixo de coordenadas:

Regra da mão esquerda



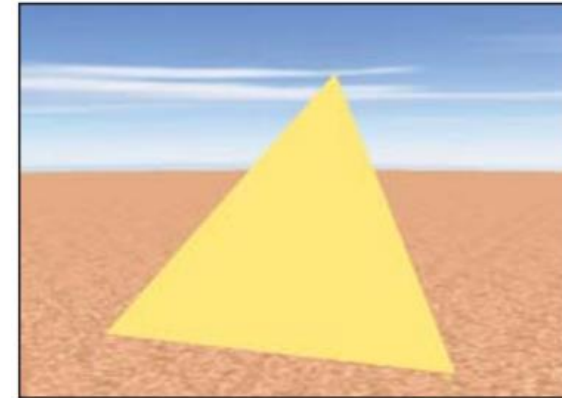
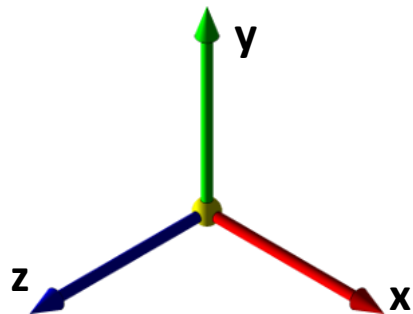
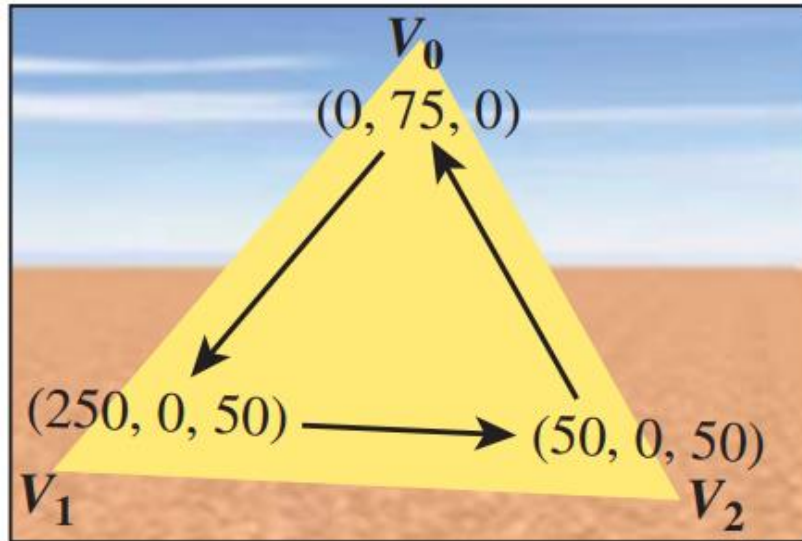
x

Regra da mão direita

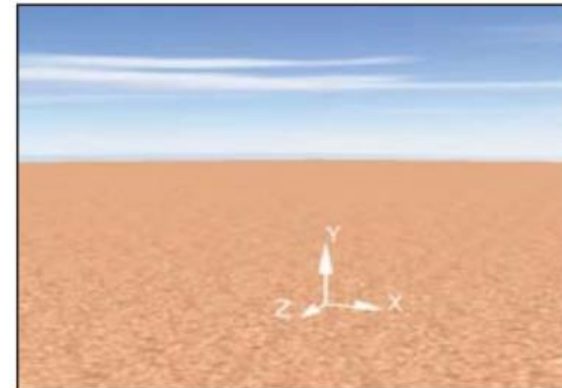


Não dá na mesma?
O lado sentido positivo para
o eixo z muda!

Qual a frente do polígono?



v_1, v_2, v_0
(frente para a câmera)



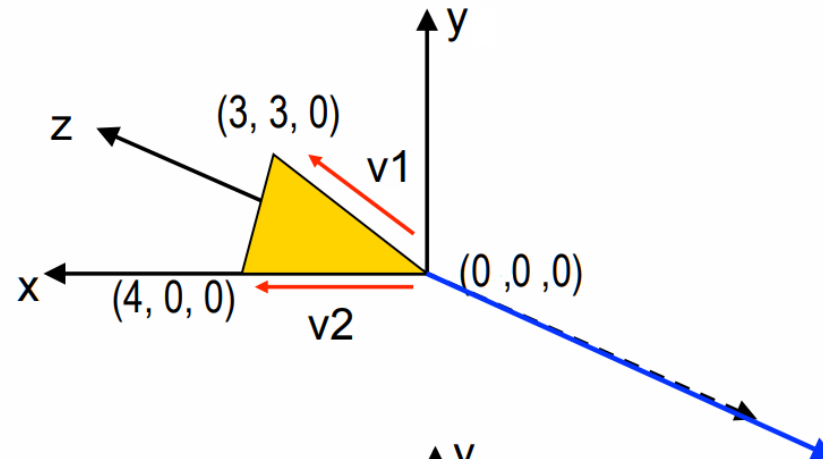
v_1, v_0, v_2
(fundo para a câmera)

Não é possível ver o polígono!

Produto vetorial entre os vetores determina o lado

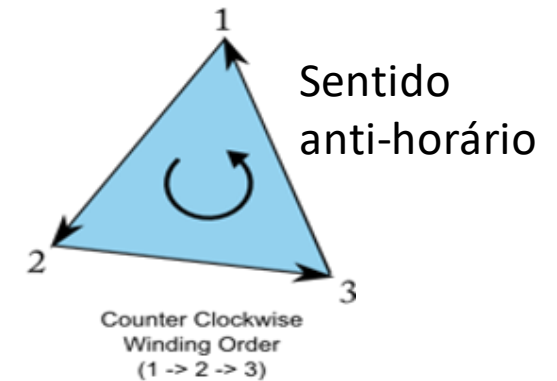
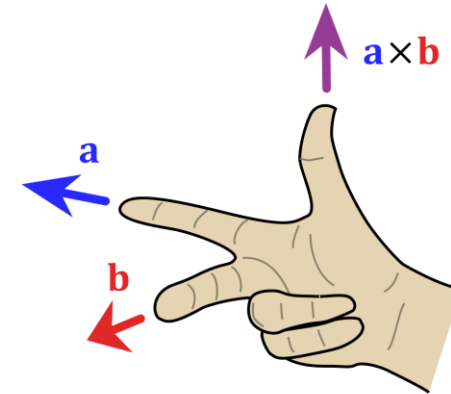
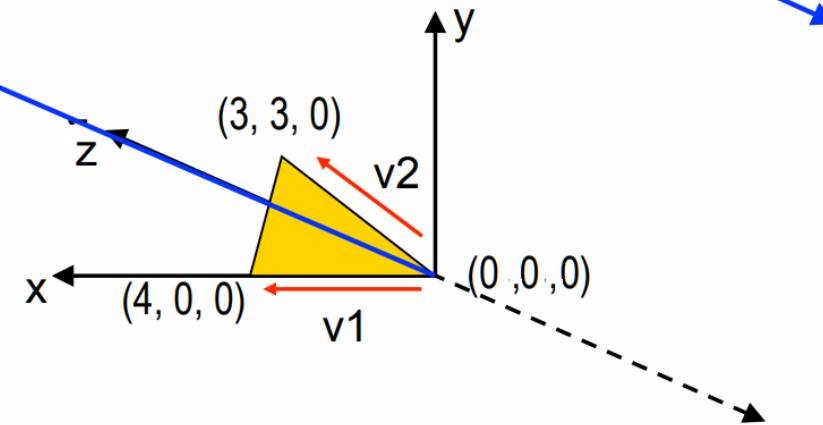
$$\vec{v1} = 3\mathbf{i} + 3\mathbf{j} + 0\mathbf{k}$$

$$\vec{v2} = 4\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}$$



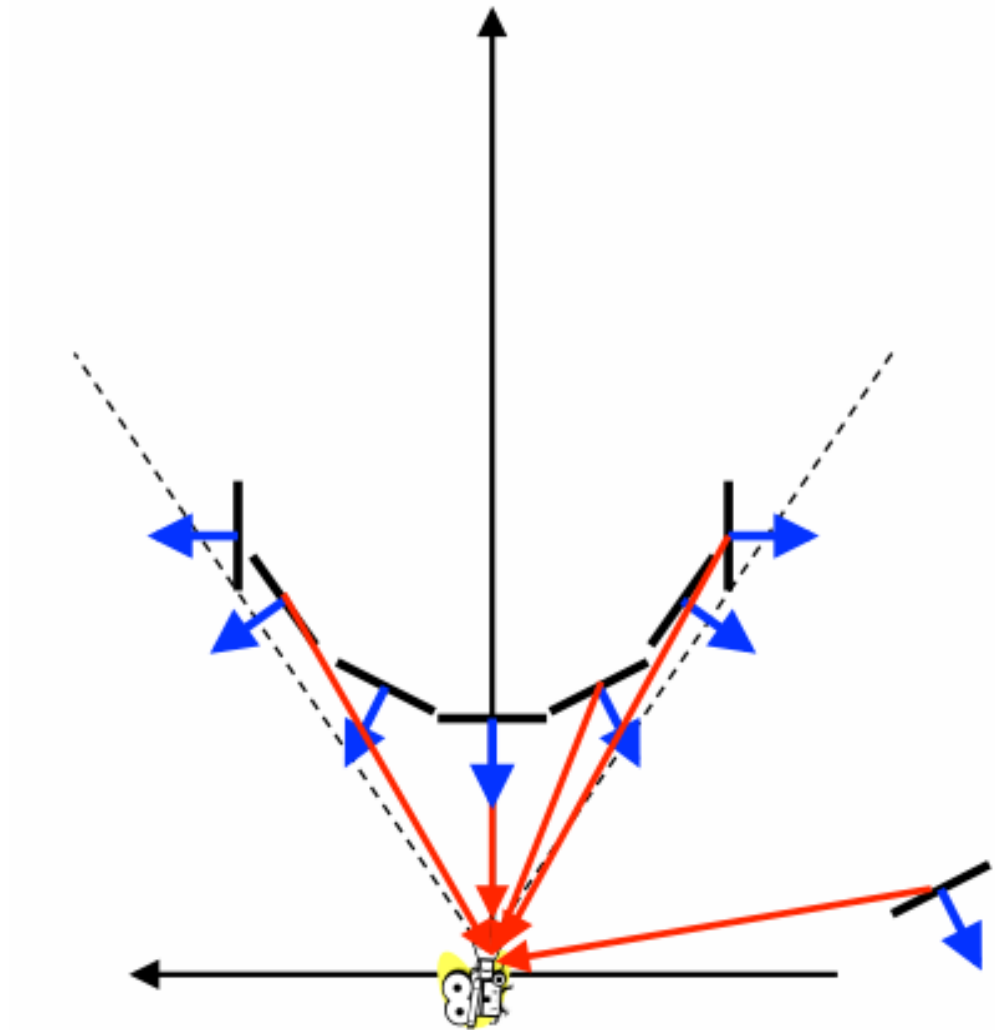
$$\vec{v1} = 4\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}$$

$$\vec{v2} = 3\mathbf{i} + 3\mathbf{j} + 0\mathbf{k}$$



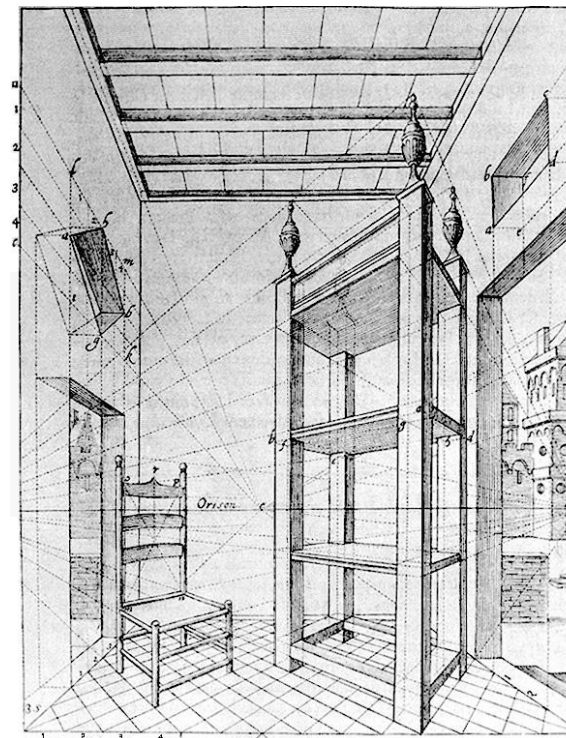
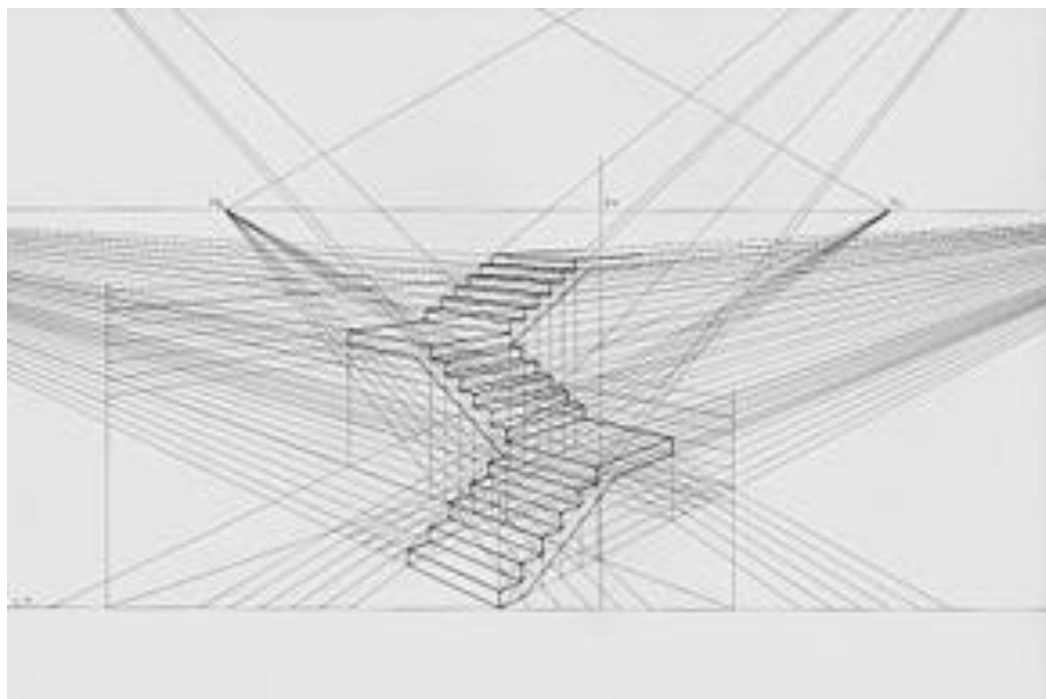
Portanto, a ordem com que são conectados os vértices importa!

Visibilidade pela Câmera



Câmera, Projeção e Perspectiva

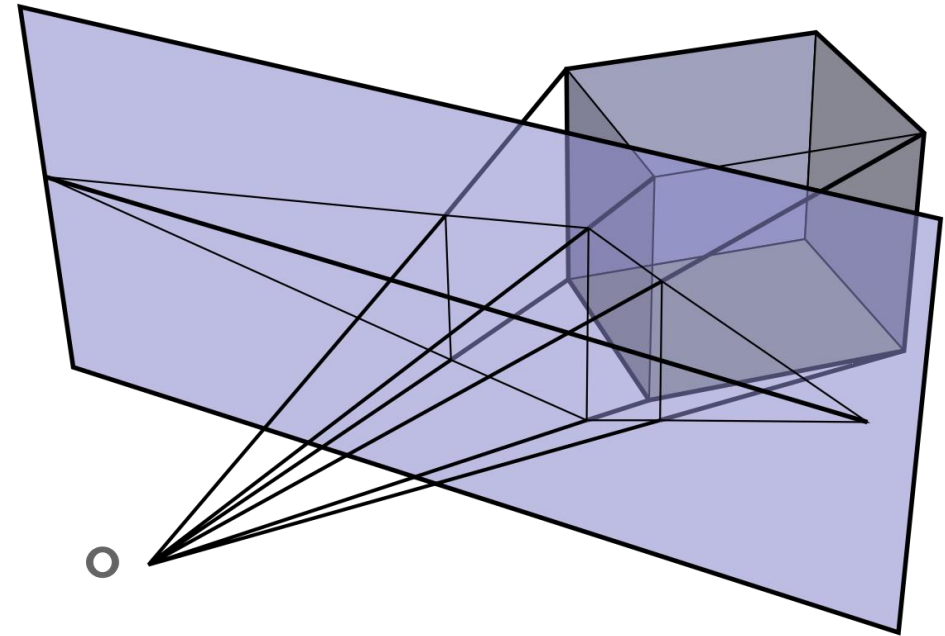
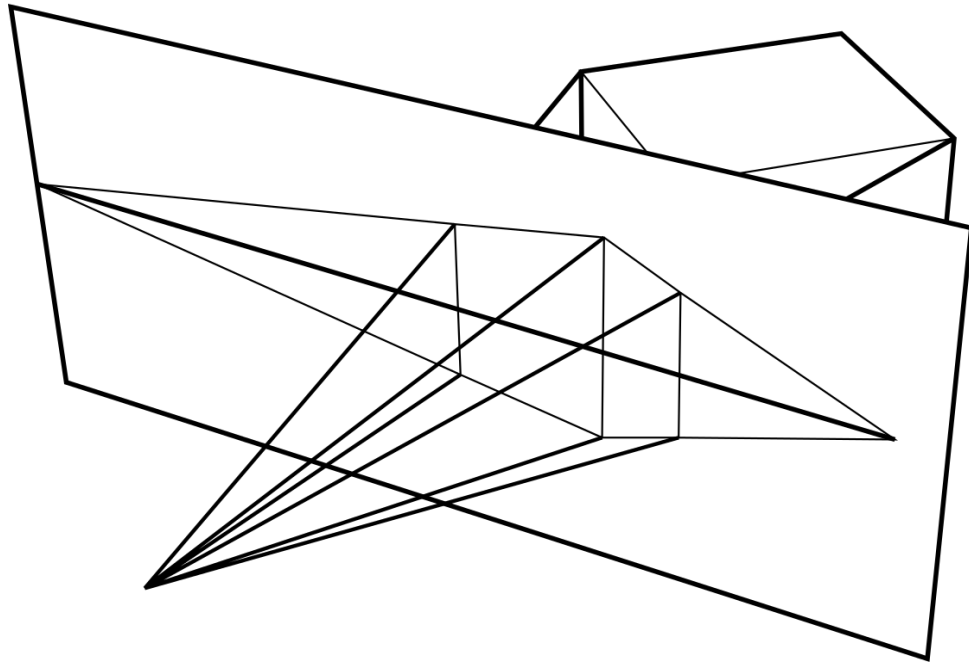
A perspectiva é uma representação de objetos e do espaço que os cerca, da maneira como são percebidos pelo olhar, a partir de um ponto de vista fixo.



Câmera, Projeção e Perspectiva

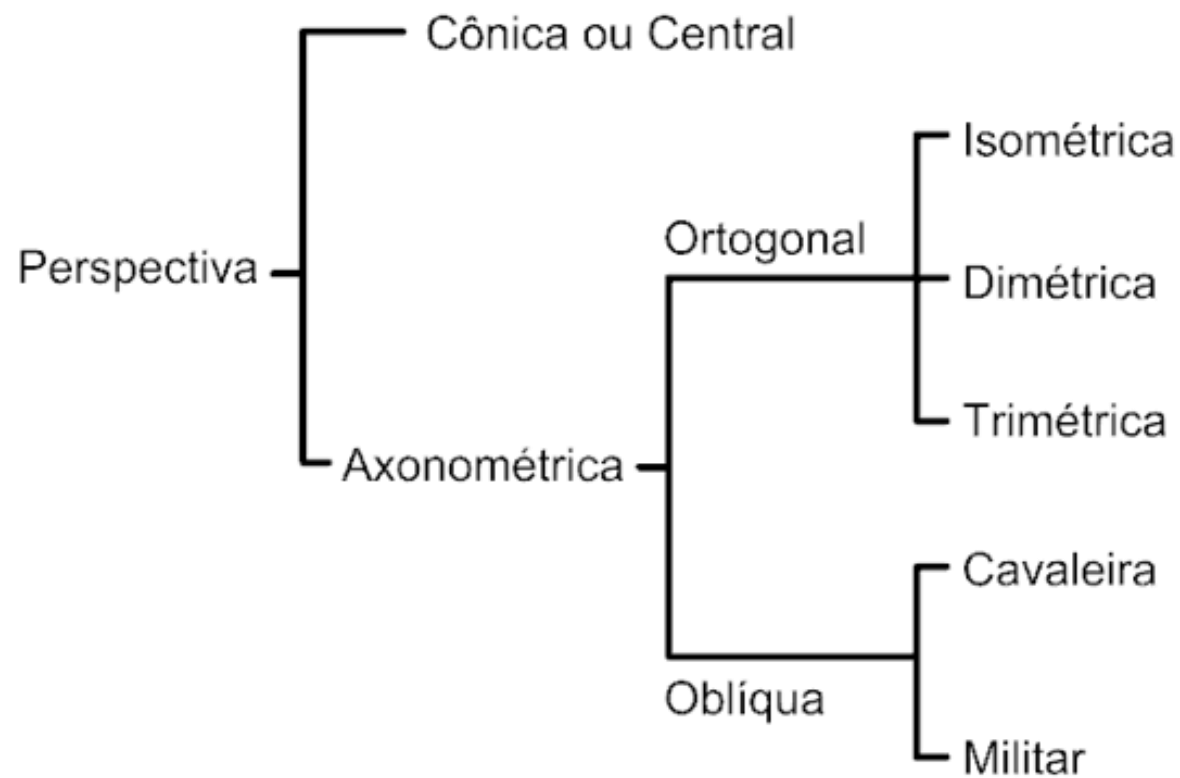
A perspectiva pode ser obtida por meio de uma projeção (3D para o 2D).

Exemplo: Projeção

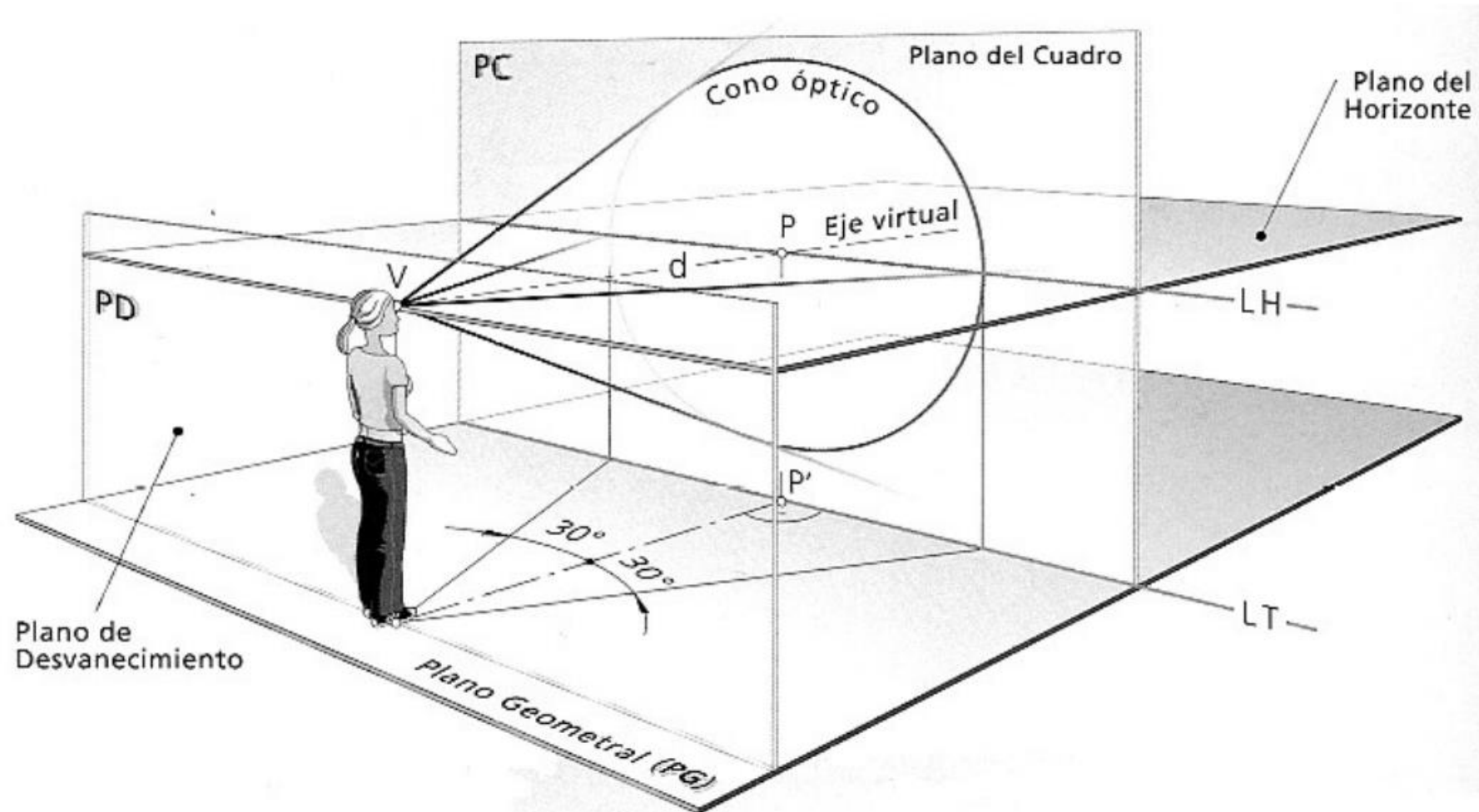


Câmera, Projeção e Perspectiva

Diferentes tipos de projeção:

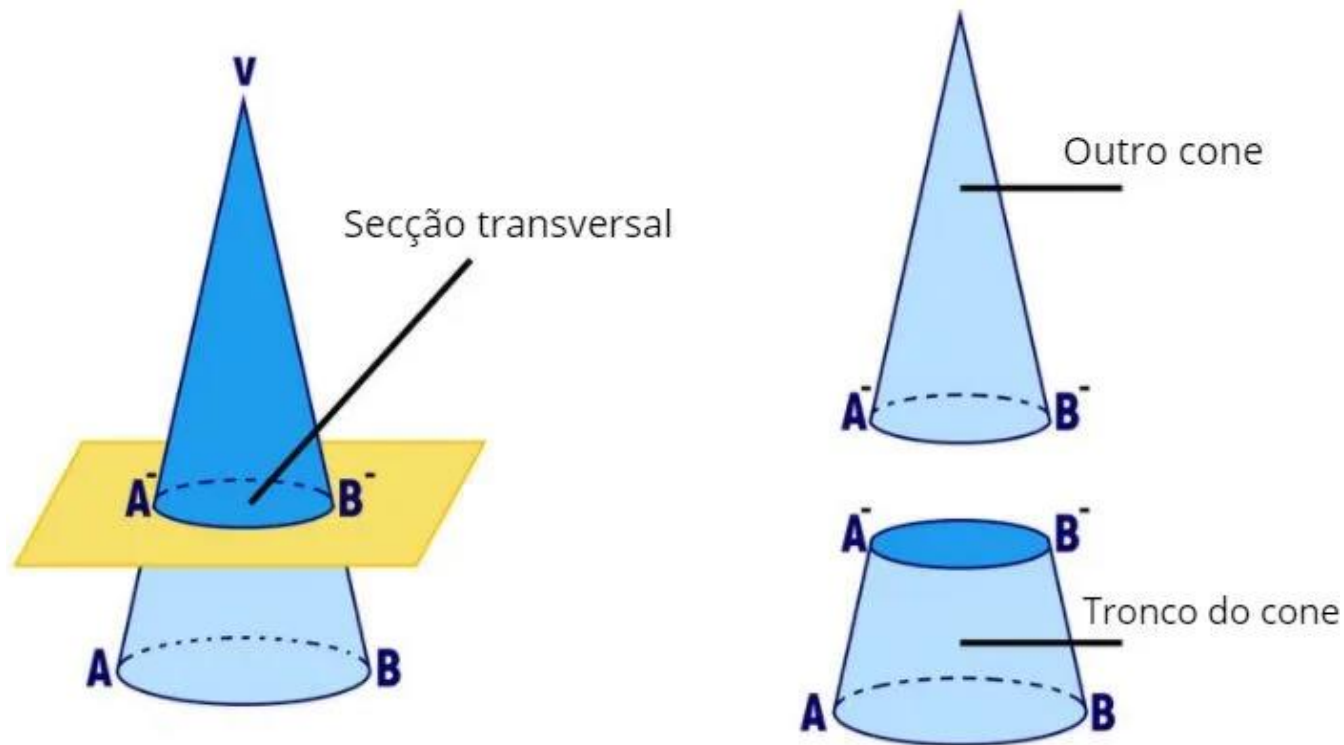


Perspectiva Cônica



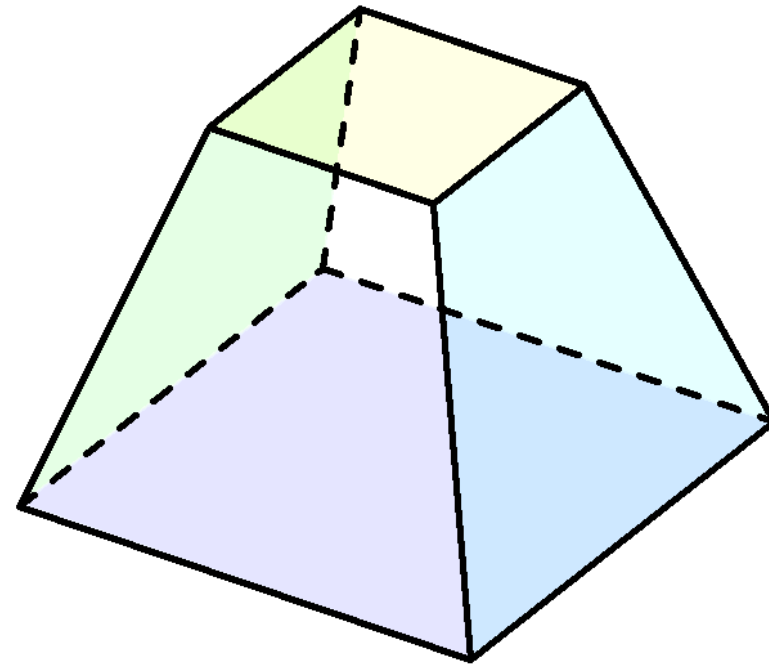
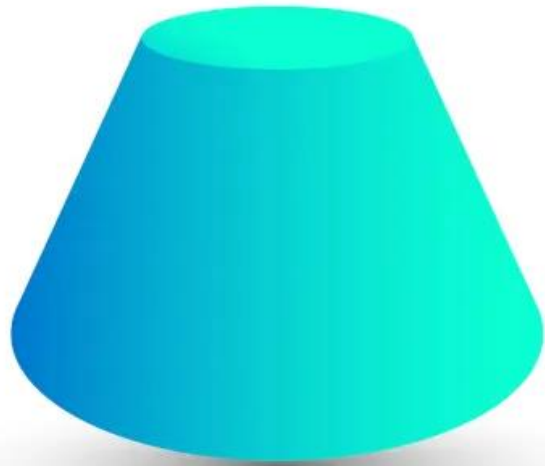
Perspectiva Cônica

Tronco de Cone:

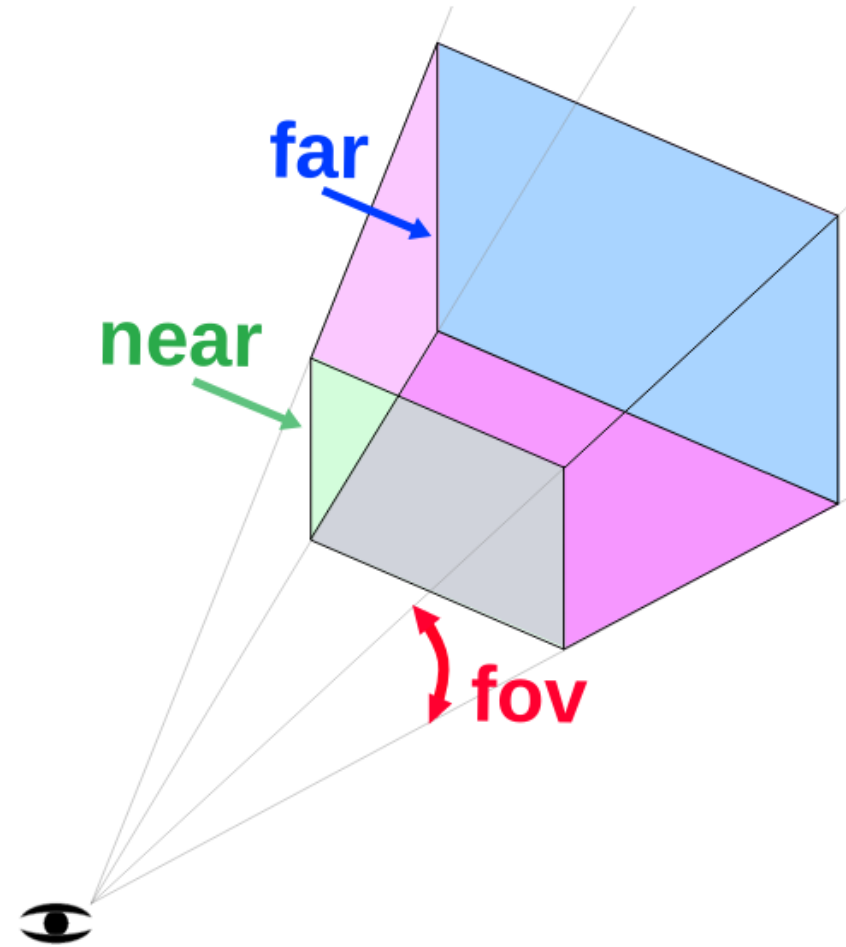


Perspectiva Cônica

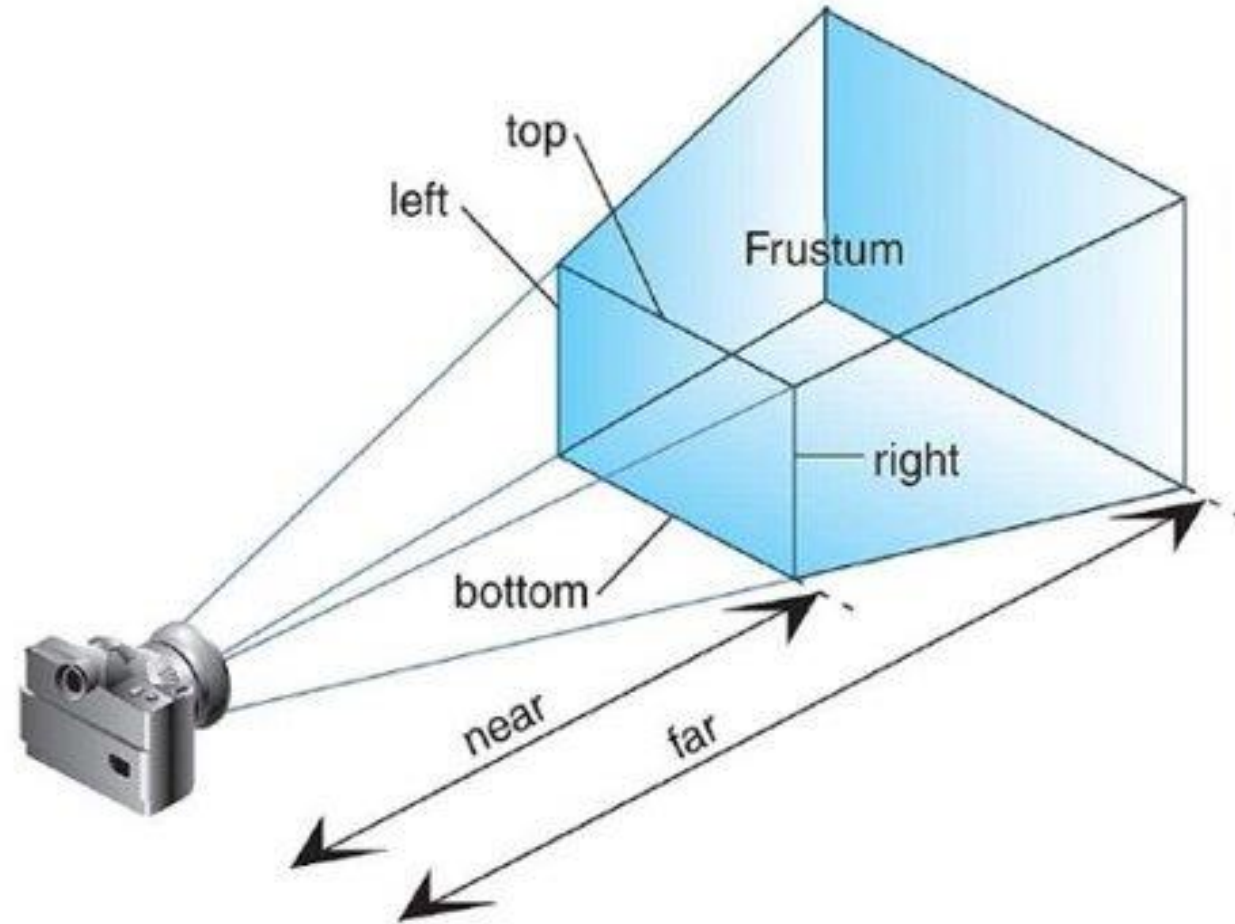
Frustum (Tronco de Pirâmide):



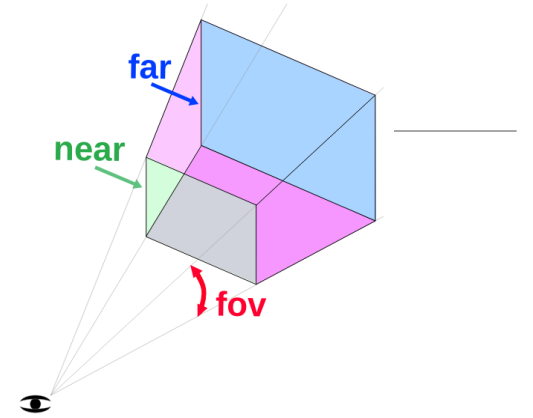
Perspectiva Cônica



Perspectiva Cônica



```
const fov = 75;  
const aspect = 2; // the canvas default  
const near = 0.1;  
const far = 5;  
const camera = new THREE.PerspectiveCamera(fov, aspect, near, far);
```



fov é a abreviação de campo de visão (*field of view*). Nesse caso, 75 graus na dimensão vertical. Observe que a maioria dos ângulos em three.js é em radianos, mas por algum motivo a `PerspectiveCamera` recebe o valor em Graus.

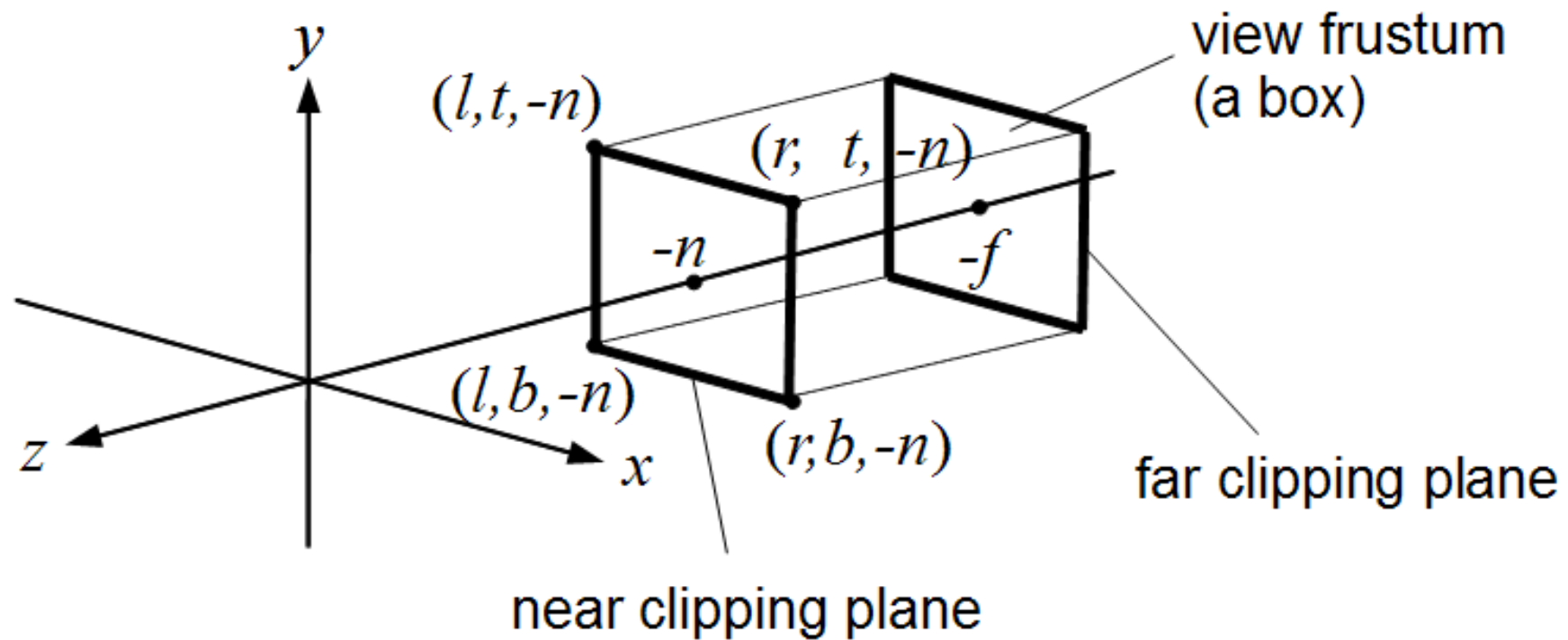
aspect é o aspecto de exibição da tela. Se um canvas tem 300 x 150 pixels, o aspecto é dados por 300/150 ou 2.

near / far representam o espaço na frente da câmera que será renderizado. Qualquer coisa antes desse intervalo ou depois dele será cortada (não desenhada).

Demonstração Interativa: <https://threejs.org/manual/examples/cameras-perspective-2-scenes.html>

Perspectiva Axonométrica ou Paralela

Variação do sólido de visualização:



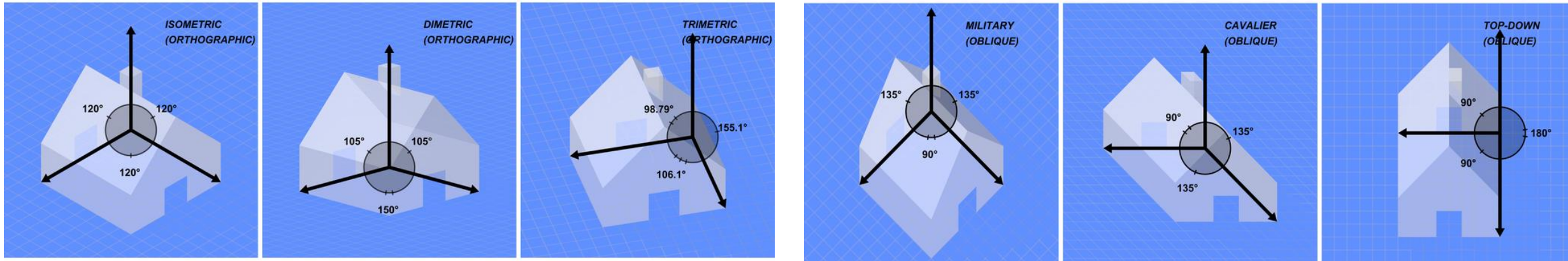
Perspectiva Axonométrica ou Paralela

Diferentes tipos de projeção de acordo com a posição da câmera:

Ortográfica

x

Oblíqua



Demonstração Interativa: <https://threejs.org/manual/examples/cameras-orthographic-2-scenes.html>

Leitura: <https://mechanicalland.com/explanation-of-the-axonometric-projections-in-technical-drawing/>

Alguns exemplos

Q*Bert
Isométrica
(Ortográfica)



Stardew Valley
Top-Down
(Oblíqua)



SimCity 4
Trimétrica
(Ortográfica)

Câmeras do Three.js

Perspectiva:

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

Ortográfica:

```
const camera = new THREE.OrthographicCamera(left, right, top, bottom, near, far);
```

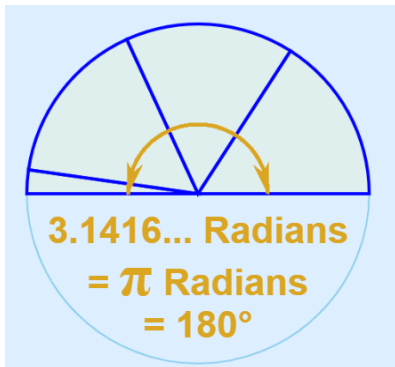
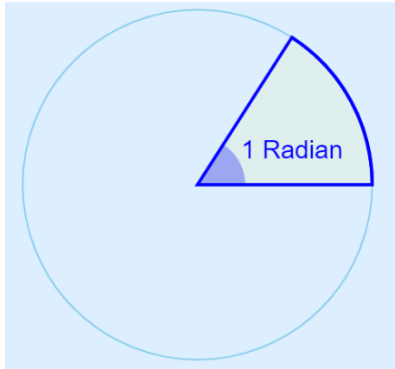
Apenas duas?

Todas as projeções podem ser obtidas a partir do ajuste dos parâmetros.

Recapitulando: Conversão Graus x Radiano

Todas as funções relacionadas a ROTAÇÃO no Three.js recebem os valores em RADIANOS.

O que é radiano?



Como fazer a conversão de graus para radiano?

Se **π radianos** é equivalente a **180 graus**, quanto vale o ângulo de 45 graus em radianos?

$$\begin{array}{l} 180^\circ = \pi \\ 45 = x \end{array}$$

$$\begin{array}{l} 180^\circ = \pi \\ 45 = x \end{array}$$

$$\begin{array}{l} 45 * \pi = 180 x \\ \pi/4 = x \end{array}$$

De onde podemos usar a fórmula:

$$\text{rad} = \text{graus} * \pi / 180;$$

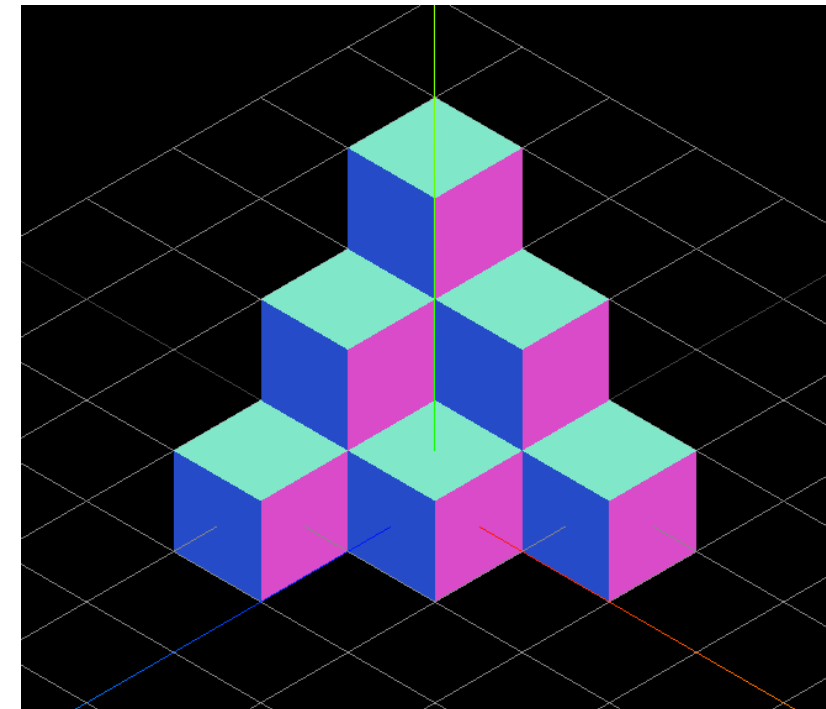
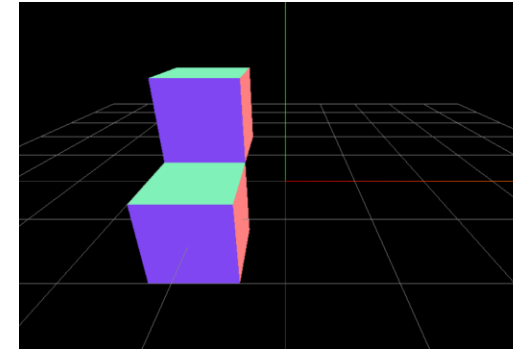
Praticando no Three.js

Exercício 1:

Utilizando os códigos elaborados em sala, modifique o arquivo 'Aula04_Ex1' para transformar a cena da primeira imagem no resultado da segunda imagem.

Lembre-se de:

- Modificar a câmera.
- Adicionar mais cubos.
- Posicionar os cubos corretamente.



Praticando no Three.js

Exercício 2:

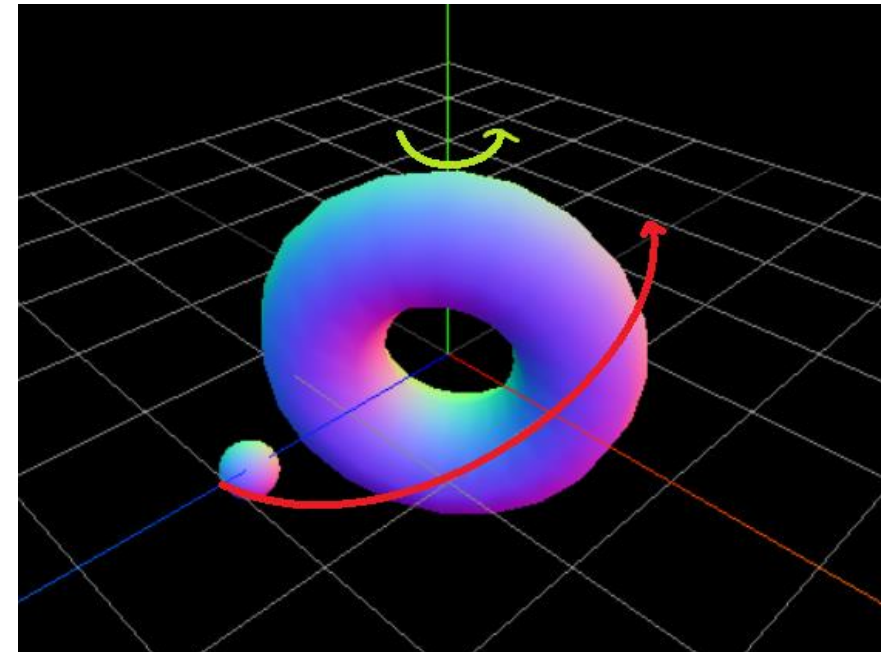
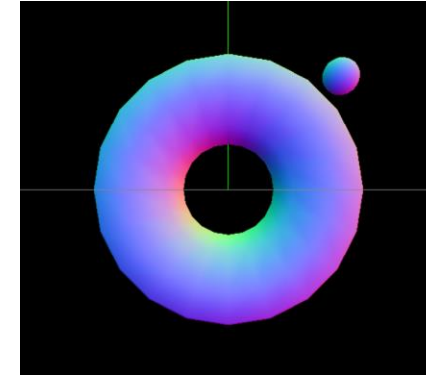
Utilizando os códigos elaborados em sala, modifique o arquivo 'Aula04_Ex2' para transformar a cena da primeira imagem no resultado da segunda imagem.

Requisitos:

- O anel deve girar em torno do eixo y.
- A esfera deve "orbitar" o anel com um ângulo de 45°.

Lembre-se de:

- Modificar a câmera.
- Adicionar rotação ao "anel".
- Adicionar rotação a esfera.



Praticando no Three.js

Exercício 3:

Utilizando os códigos elaborados em sala, modifique o arquivo 'Aula04_Ex3' para transformar a cena da primeira imagem no resultado da segunda imagem.

Lembre-se de:

- Adicionar novos triângulos.
 - Posiciona-los corretamente.
 - Adicionar a animação conforme a referência.
-
- Referência: [Clique aqui para acessar o link](#)

