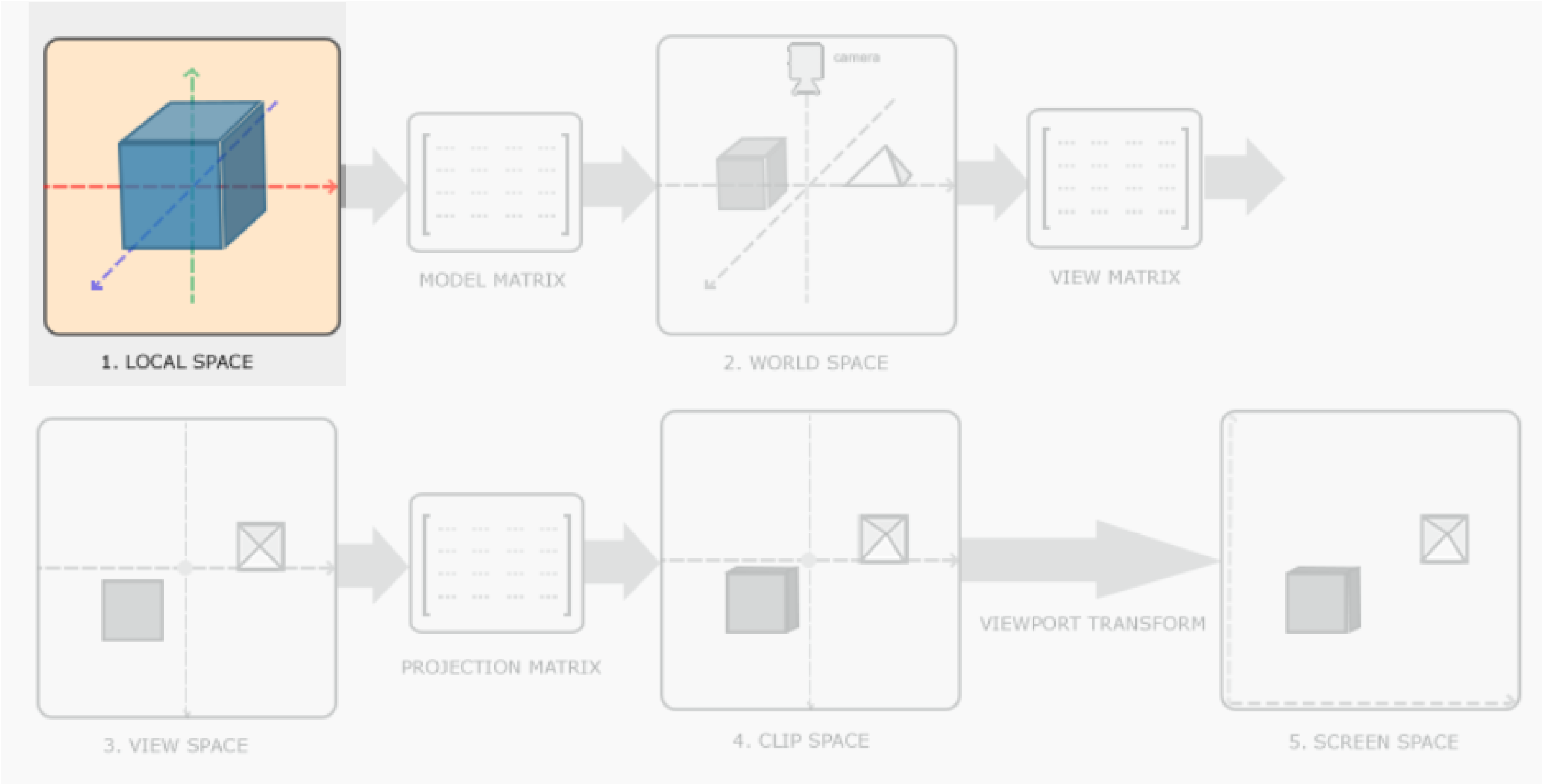
Computação Gráfica

Aula 08 (parte 2) – *View*

Prof. Jean R. Ponciano

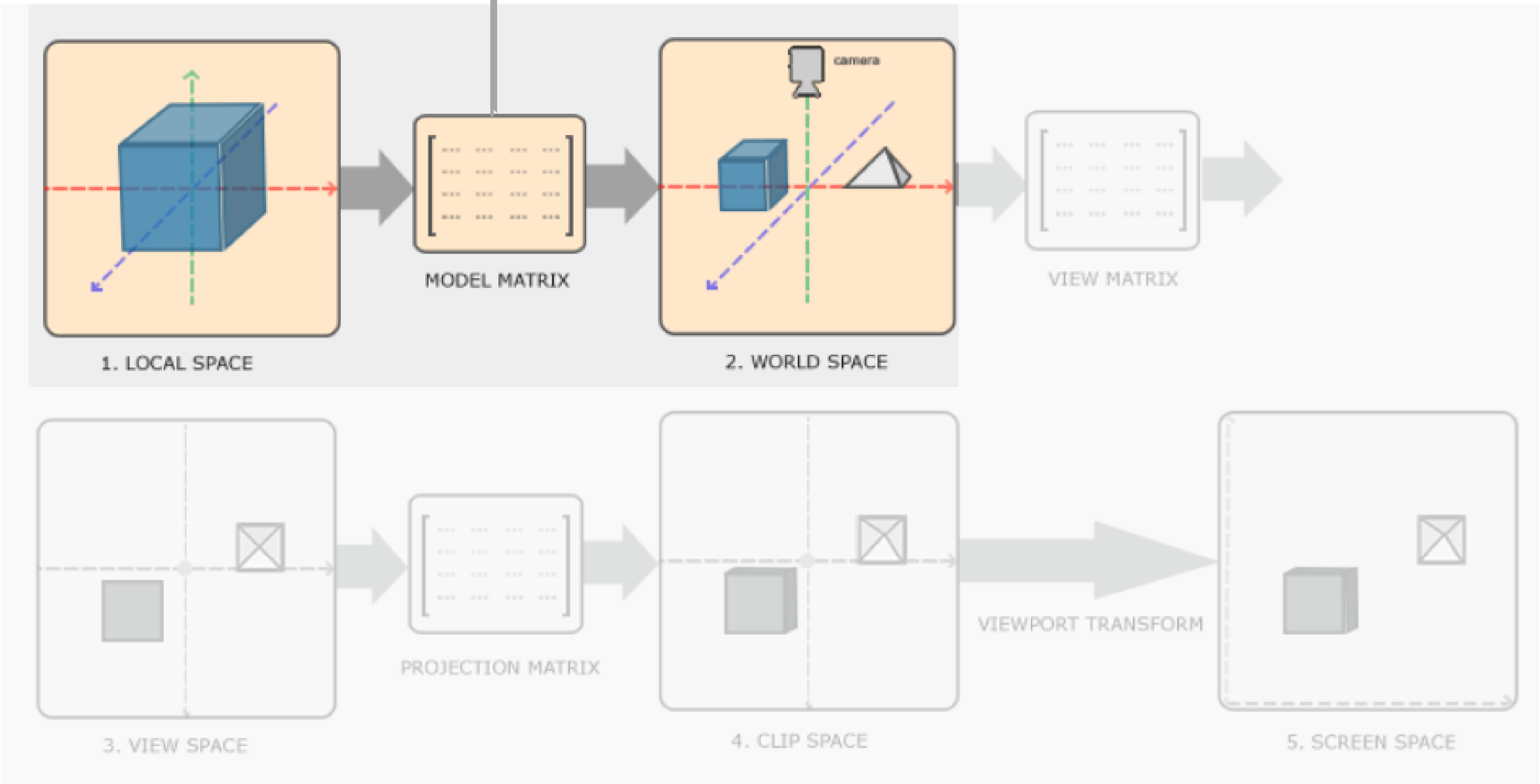
Pipeline de transformações

Coordenadas iniciais dos vértices (objetos)



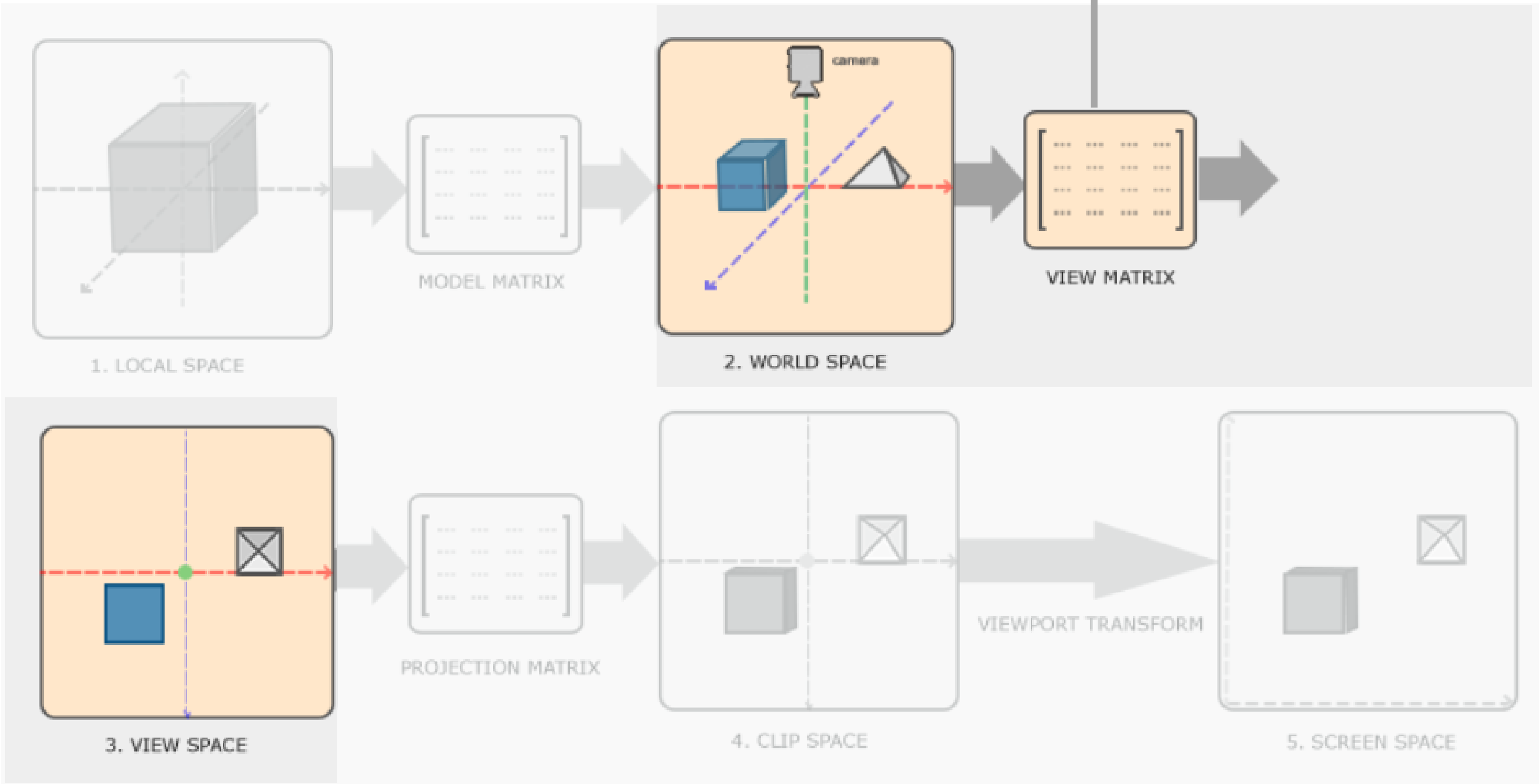
Pipeline de transformações

Transformações geométricas



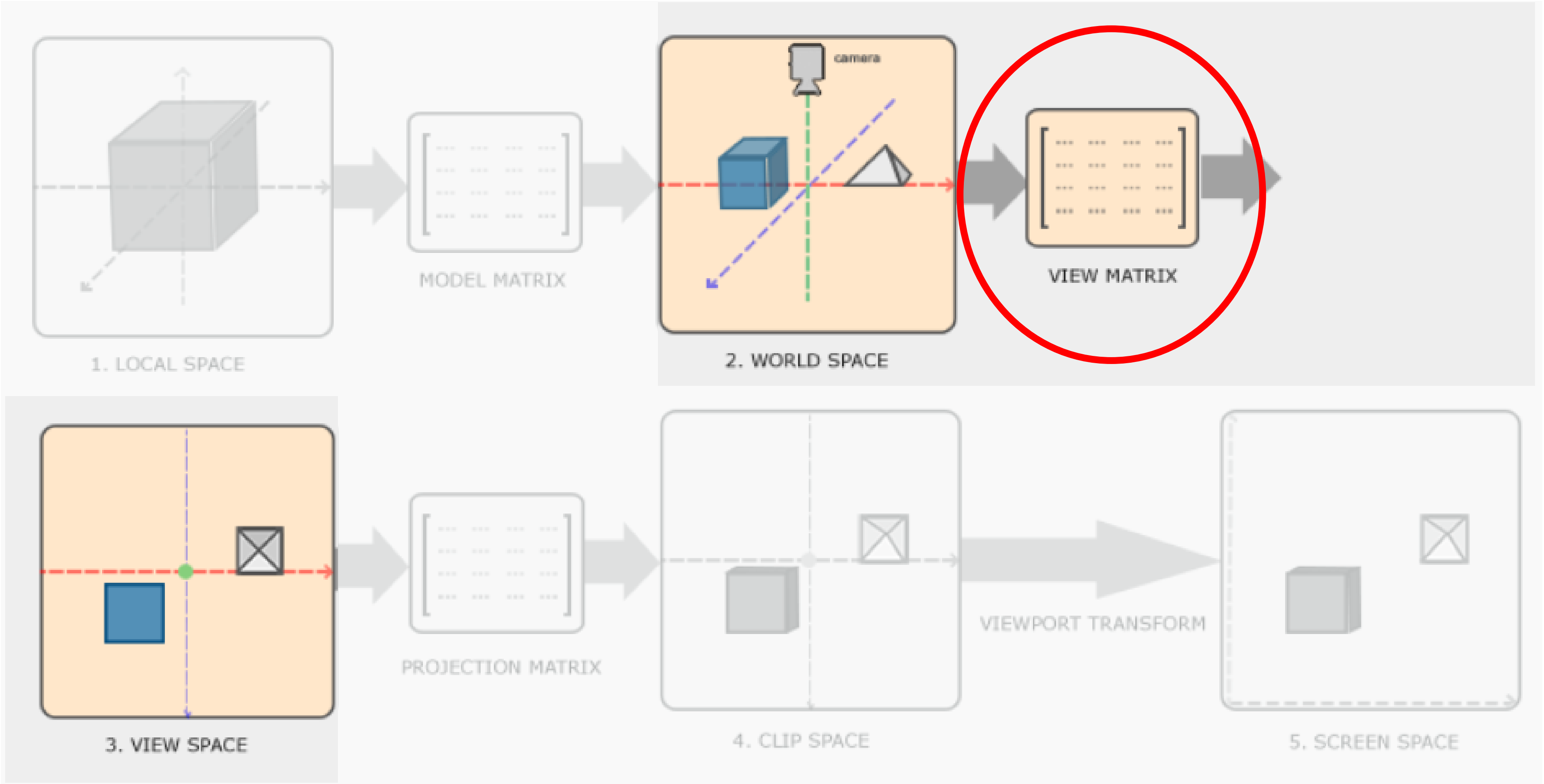
Pipeline de transformações

Transformação para visualizar os objetos a partir da câmera



View Matrix

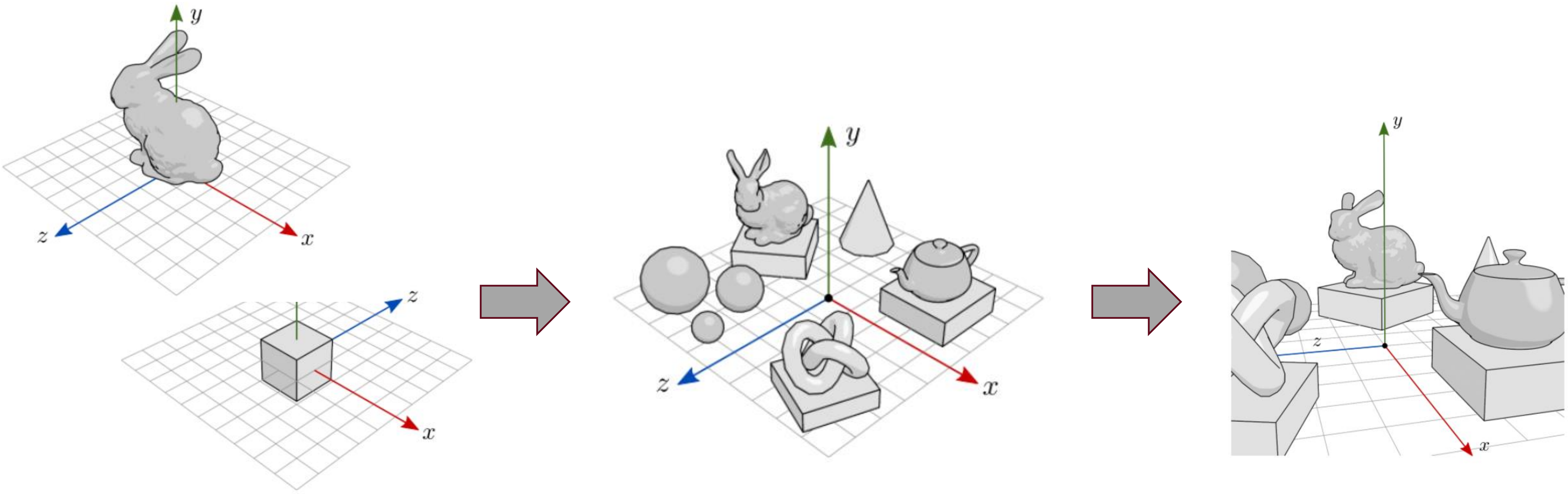
P' = Projection x **View** x Model x P



P' = Projection x **View** x Model x P

View Matrix

- Espaço Mundo → Espaço Visão (ou espaço da câmera)



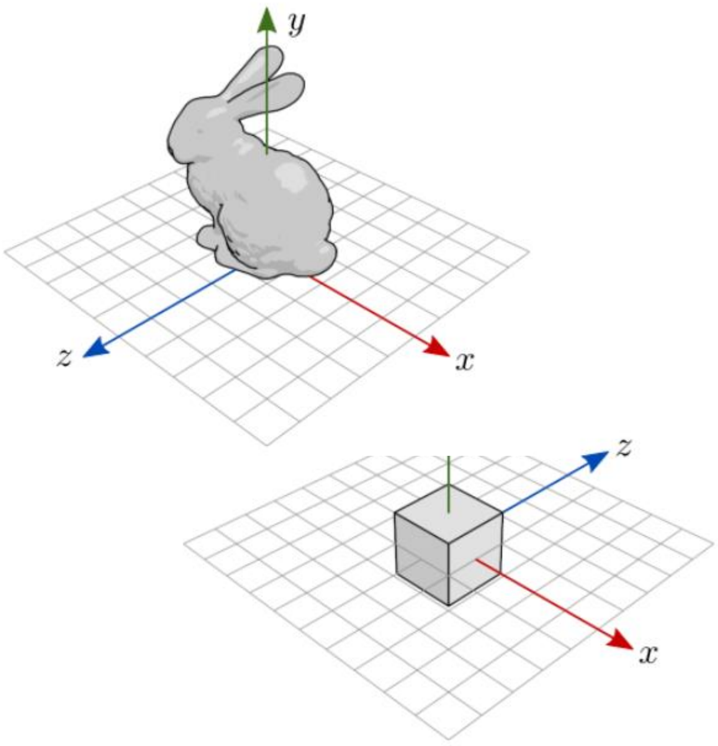
Espaço local | Espaço mundo | Espaço visão

P’ = Projection x **View** x Model x P

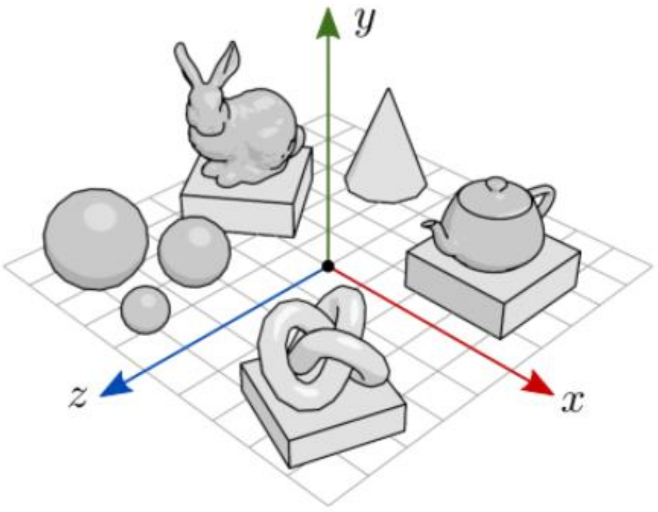
View Matrix

- Espaço Mundo → Espaço Visão (ou espaço da câmera)

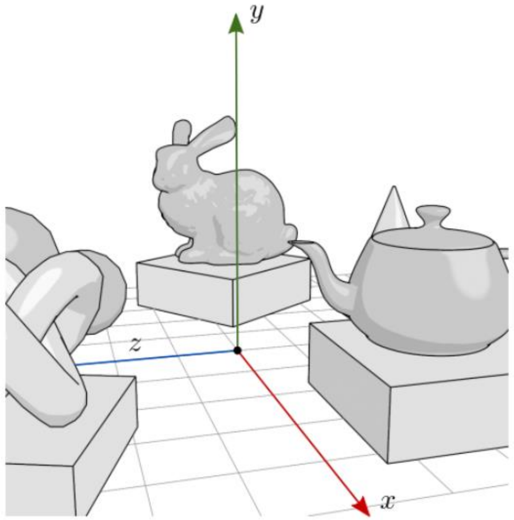
Model Matrix



Espaço local



Espaço mundo

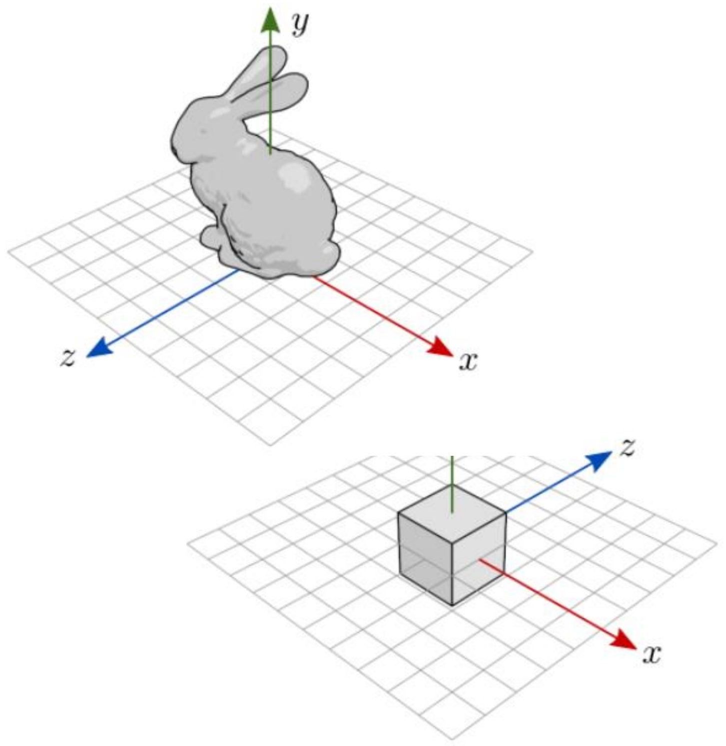


Espaço visão

P' = Projection x **View** x Model x P

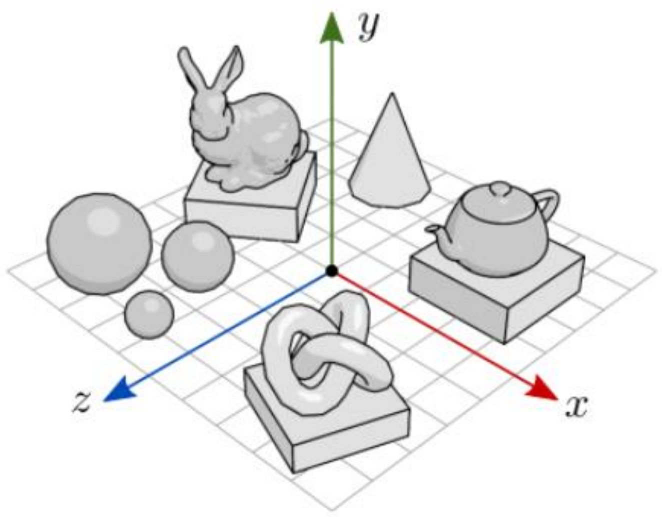
View Matrix

- Espaço Mundo → Espaço Visão (ou espaço da câmera)

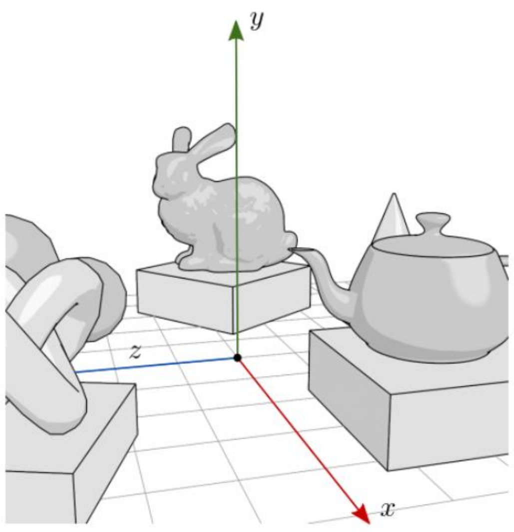


Espaço local

View Matrix



Espaço mundo



Espaço visão

P' = Projection x **View** x Model x P

View Matrix

- Espaço Mundo → Espaço Visão (ou espaço da câmera)

- Transferência de objetos do **sistema de coordenadas do mundo** para o **sistema de coordenadas da câmera**

P' = Projection x **View** x Model x P

View Matrix

- Espaço Mundo → Espaço Visão (ou espaço da câmera)

- Transferência de objetos do **sistema de coordenadas do mundo** para o **sistema de coordenadas da câmera**

Mas como é o sistema de coordenadas da câmera???

View Matrix

- Espaço Mundo → Espaço Visão (ou espaço da câmera)

- Transferência de objetos do **sistema de coordenadas do mundo** para o **sistema de coordenadas da câmera**

- Vamos lembrar do nosso objetivo:

Efetuar sua apresentação

Transformação de Visualização

- Fazer a apresentação envolve:
  - Definição dos parâmetros da câmera
  - Mudança do sistema de coordenadas do mundo para o sistema da câmera (view matrix)
  - Transformações de Projeção (projection matrix)
  - Recorte/Eliminação de partes não visíveis da cena
  - Aplicação de iluminação/tonalização

Transformação de Visualização

- Fazer a apresentação envolve:
  - Definição dos parâmetros da câmera
  - Mudança do sistema de coordenadas do mundo para o sistema da câmera (view matrix)
  - Transformações de Projeção (projection matrix)
  - Recorte/Eliminação de partes não visíveis da cena
  - Aplicação de iluminação/tonalização

Essa sequência de passos define o **pipeline de observação** (*viewing pipeline*)

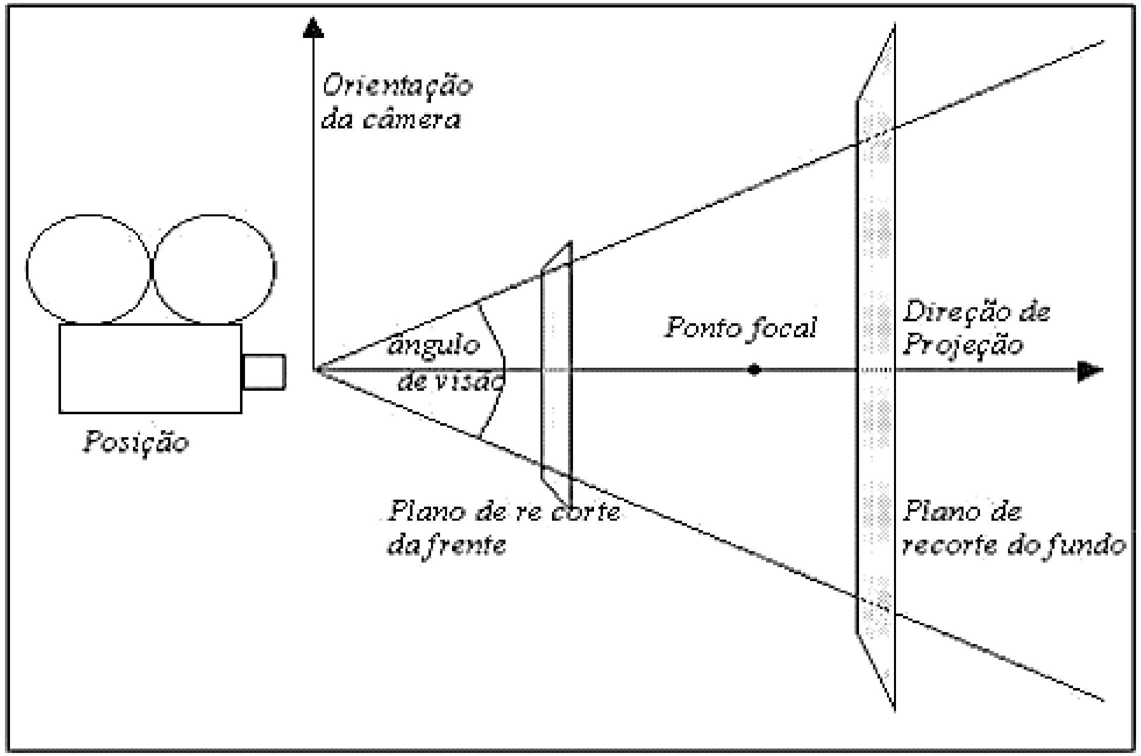
Transformação de Visualização

- Fazer a apresentação envolve:
  - **Definição dos parâmetros da câmera**
  - **Mudança do sistema de coordenadas do mundo para o sistema da câmera (view matrix)**
  - Transformações de Projeção (projection matrix)
  - Recorte/Eliminação de partes não visíveis da cena
  - Aplicação de iluminação/tonalização

Essa sequência de passos define o **pipeline de observação** (*viewing pipeline*)

Parâmetros da câmera

O observador vê a cena através das lentes de uma **câmera virtual**

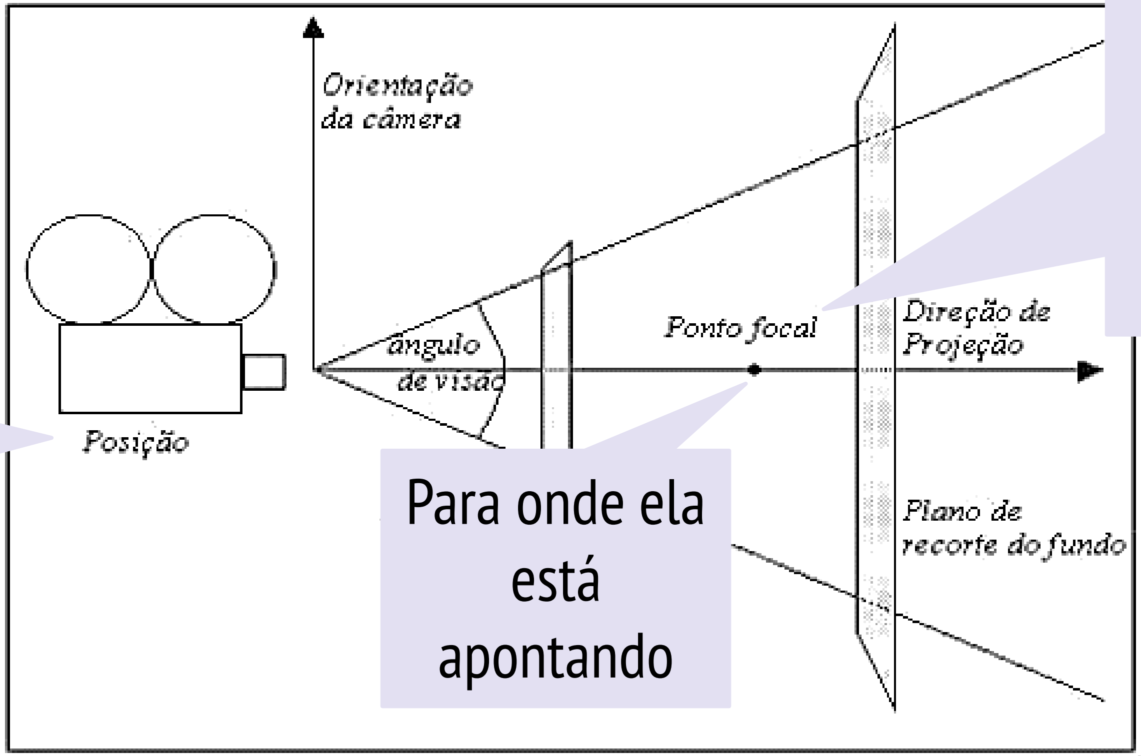


A câmera pode ser posicionada de forma a obter a imagem desejada da cena

(posição, orientação e ponto focal)

[Schröeder, 1998]

Parâmetros da câmera

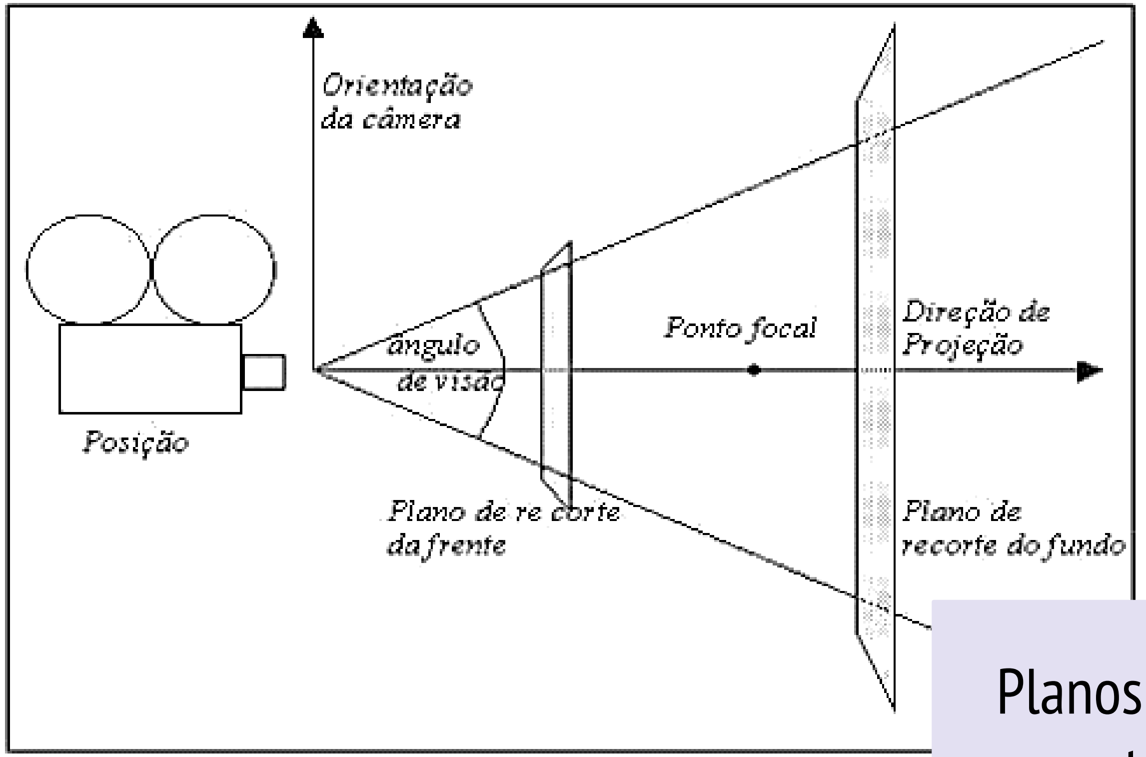


Onde a câmera está

Para onde ela está apontando

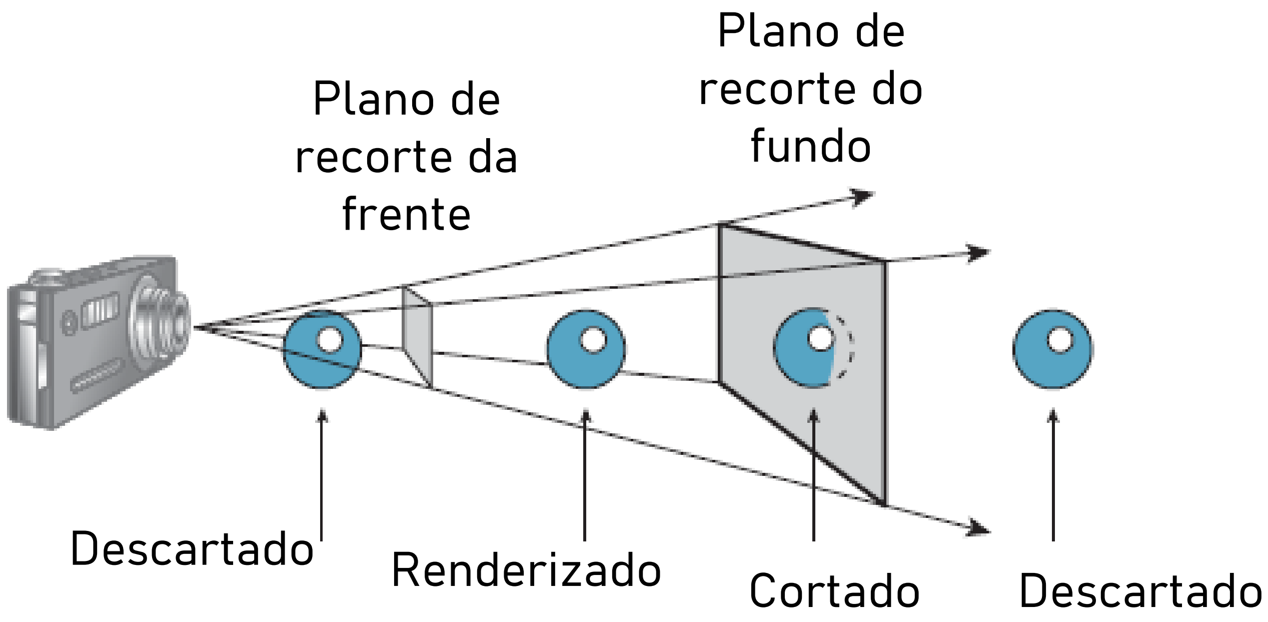
Plano de imagem posicionado no ponto focal e, tipicamente, perpendicular ao vetor direção de projeção

Parâmetros de observação



Planos de recorte definem o que entra na projeção da cena

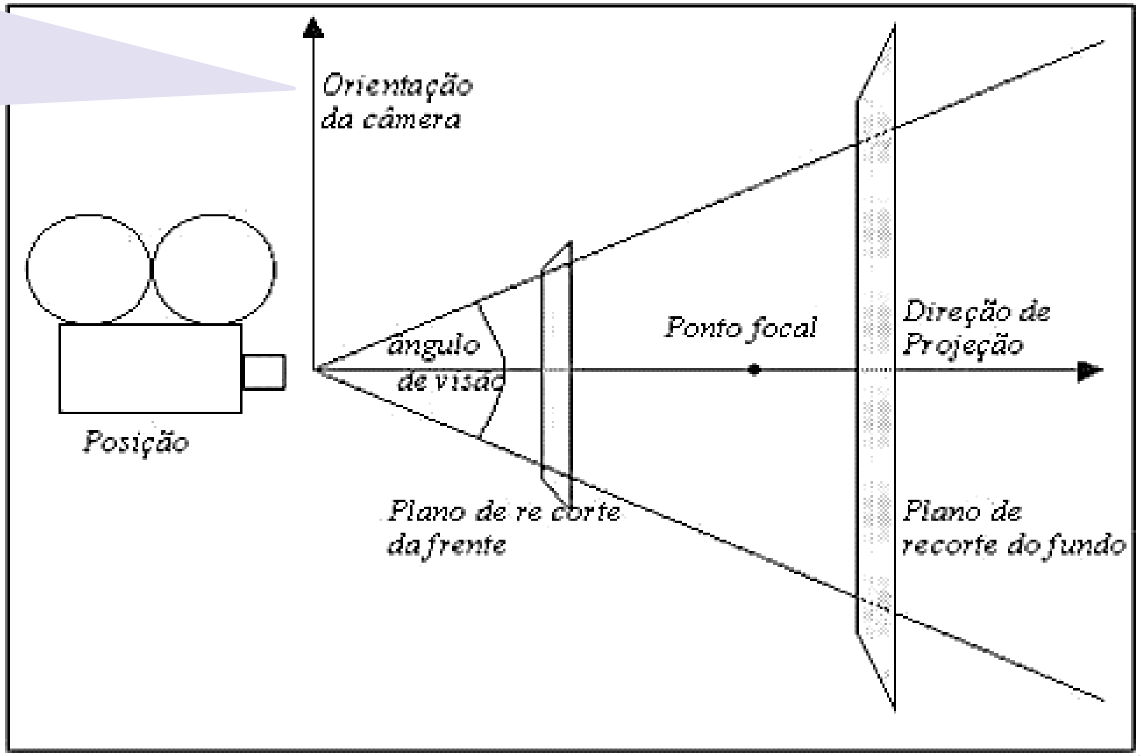
Parâmetros de observação



Planos de recorte definem o que entra na projeção da cena.
Veremos mais detalhes ao estudar a **projection matrix**.

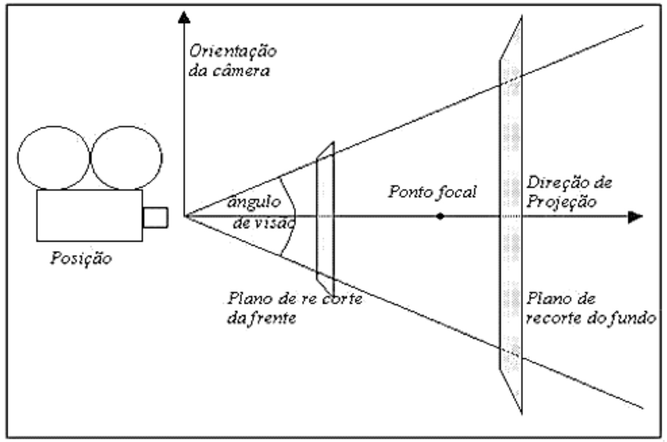
Parâmetros da câmera

Orientação controlada pela **posição**, **ponto focal** e um vetor chamado $\overrightarrow{\boldsymbol{UP}}$.



Esses três parâmetros definem completamente a câmera

Parâmetros da câmera

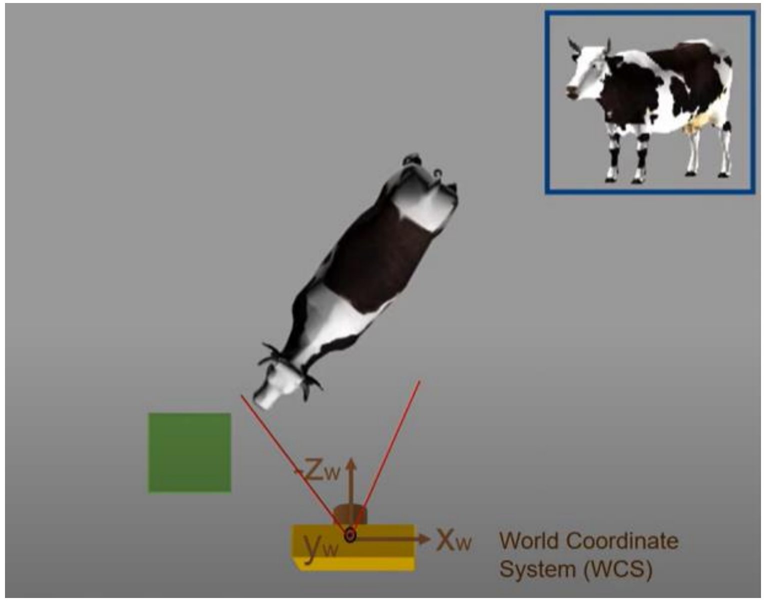


- O ponto **Pos = (p_x, p_y, p_z)**, que representa a posição dela no espaço mundo
  - Ele é a origem do sistema de coordenadas da câmera
- O ponto **ponto_focal = (l_x, l_y, l_z)**, que representa o local que é o foco de observação
- O vetor $\overrightarrow{UP}$ **= (up_x, up_y, up_z)**, que indica o "lado de cima" da câmera

Parâmetros da câmera

- Podemos caminhar na cena e fotografar de qualquer ângulo, de várias distâncias e com diferentes orientações da câmera.

A escolha dos parâmetros da câmera afetará o que será exibido

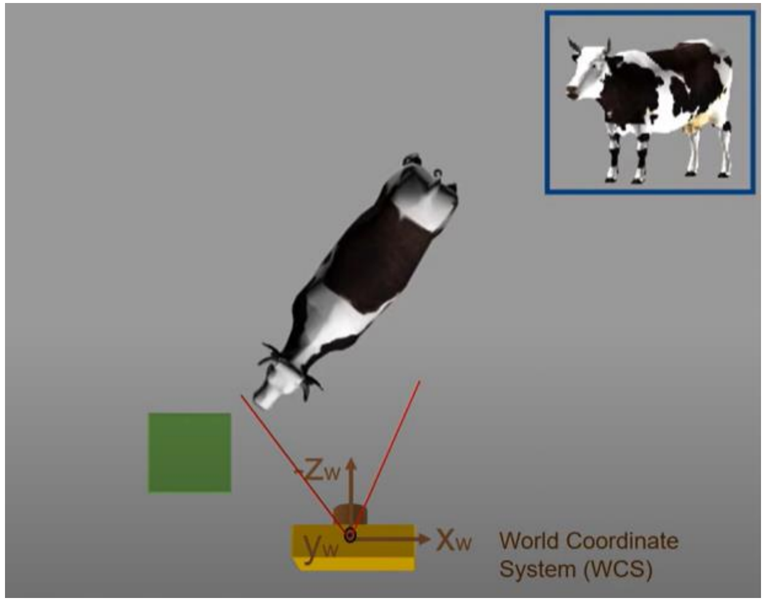


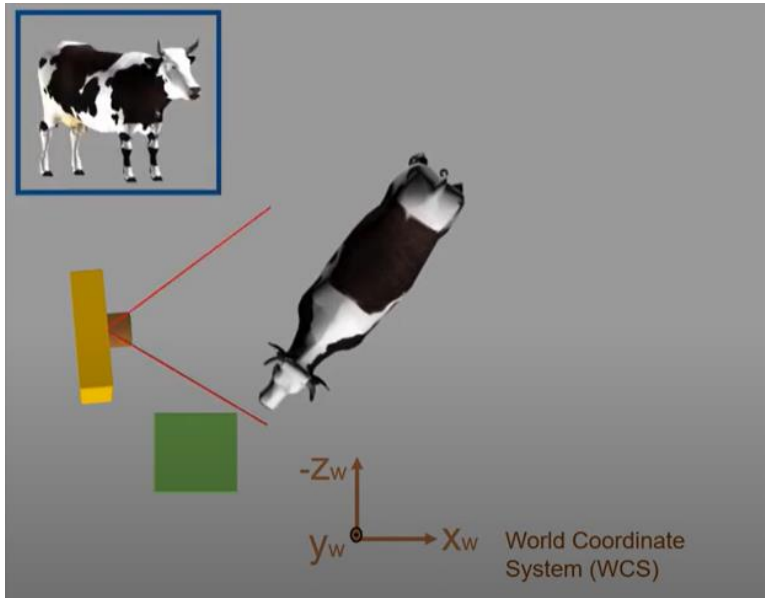
Fonte da imagem: vídeo-aula prof. Manuel M. Oliveira (https://youtu.be/v0IwXIZqUDM?list=PLQ7b--ynwIXhP-MrKHdEKDu3T4plN5jYe)

Parâmetros da câmera

- Podemos caminhar na cena e fotografar de qualquer ângulo, de várias distâncias e com diferentes orientações da câmera.

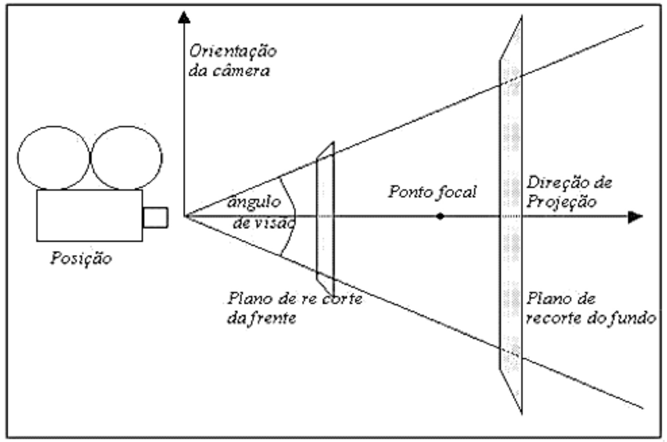
A escolha dos parâmetros da câmera afetará o que será exibido





Fonte da imagem: vídeo-aula prof. Manuel M. Oliveira (https://youtu.be/v0IwXIZqUDM?list=PLQ7b--ynwIXhP-MrKHdEKDu3T4plN5jYe)

Parâmetros da câmera



- O ponto **Pos = (p_x, p_y, p_z)**, que representa a posição dela no espaço mundo
  - Ele é a origem do sistema de coordenadas da câmera
- O ponto **ponto_focal = (l_x, l_y, l_z)**, que representa o local que é o foco de observação
- O vetor $\overrightarrow{UP}$ **= (up_x, up_y, up_z)**, que indica o "lado de cima" da câmera

Sistema de Coordenadas da Câmera

- Com os pontos Pos e Ponto_focal, e com o vetor $\overrightarrow{UP}$, podemos definir o sistema.

- Para isso, vamos obter os três **vetores unitários** que representam a **base ortonormal** do sistema.

Sistema de Coordenadas da Câmera

- Com os pontos Pos e Ponto_focal, e com o vetor $\overrightarrow{UP}$, podemos definir o sistema.

- Para isso, vamos obter os três **vetores unitários** que representam a **base ortonormal** do sistema.

Chamarei os vetores unitários de $\overrightarrow{X_c}$, $\overrightarrow{Y_c}$, $\overrightarrow{Z_c}$ para deixar mais intuitivo.

Na literatura, vocês poderão ver esses vetores sendo chamados de $\vec{u}$, $\vec{v}$, $\vec{n}$

Sistema de Coordenadas da Câmera

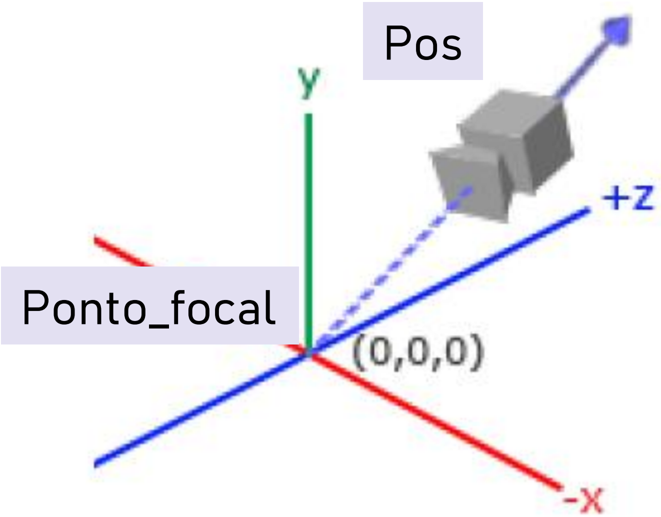
- Primeiro, o vetor unitário $\overrightarrow{Z_c}$, que representa o eixo Z do sistema da câmera
  - Regra da mão direita (vetores apontam na direção contrária de onde a câmera está olhando)

Sistema de Coordenadas da Câmera

- Primeiro, o vetor unitário $\overrightarrow{Z_c}$, que representa o eixo Z do sistema da câmera
  - Regra da mão direita (vetores apontam na direção contrária de onde a câmera está olhando)

$$\overrightarrow{LP} = Pos - Ponto\_focal$$

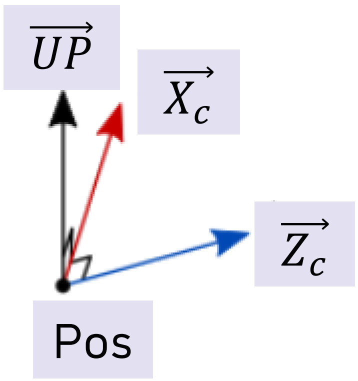
$$\overrightarrow{Z_c} = \frac{\overrightarrow{LP}}{|\overrightarrow{LP}|}$$ (vetor normalizado)



Sistema de Coordenadas da Câmera

- Agora o vetor unitário $\overrightarrow{X_c}$, que representa o eixo X da câmera
  - Perpendicular a $\overrightarrow{UP}$ e $\overrightarrow{Z_c}$ simultaneamente

$$\overrightarrow{X_c} = \frac{\overrightarrow{UP} \times \overrightarrow{Z_c}}{|\overrightarrow{UP} \times \overrightarrow{Z_c}|}$$



Sistema de Coordenadas da Câmera

- Agora o vetor unitário $\overrightarrow{X_c}$, que representa o eixo X da câmera
  - Perpendicular a $\overrightarrow{UP}$ e $\overrightarrow{Z_c}$ simultaneamente

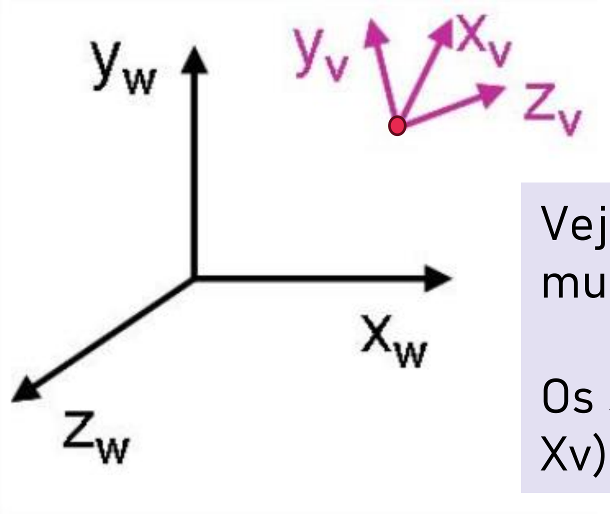
$$\overrightarrow{X_c} = \frac{\overrightarrow{UP} \times \overrightarrow{Z_c}}{|\overrightarrow{UP} \times \overrightarrow{Z_c}|}$$

- Finalmente, o vetor $\overrightarrow{Y_c}$ (perpendicular a $\overrightarrow{Z_c}$ e $\overrightarrow{X_c}$ ao mesmo tempo)

$$\overrightarrow{Y_c} = \overrightarrow{Z_c} \times \overrightarrow{X_c}$$

Transformação de Visualização

- Com o sistema de coordenadas da câmera especificado, podemos fazer a mudança do sistema de coordenadas do mundo ($X_w Y_w Z_w$) para o sistema de coordenadas da câmera ($X_v Y_v Z_v$)

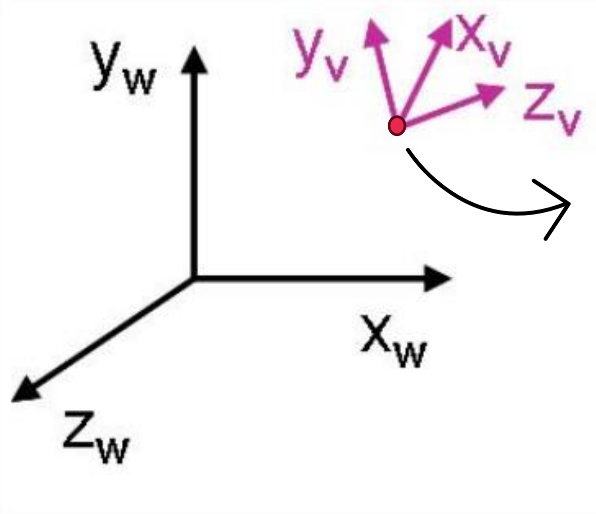


Veja que a figura usa 'w' para simbolizar as coordenadas do mundo (**w**orld) e 'v' para as coordenadas da câmera (**v**iew).

Os slides anteriores usam 'c' ao invés de 'v' (por ex, Xc ao invés de Xv), mas é a mesma coisa!

Transformação de Visualização

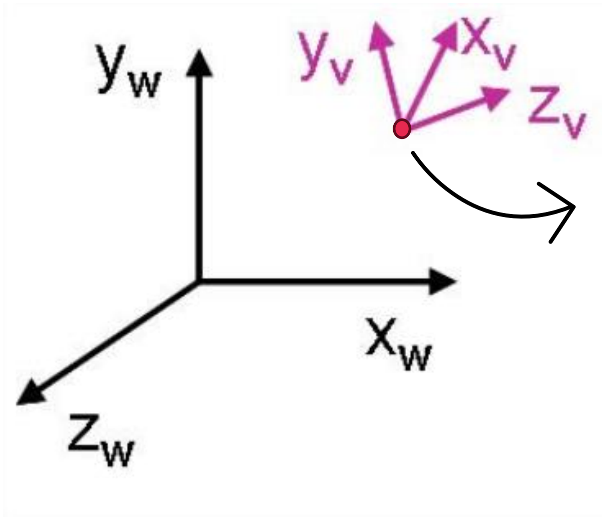
- Com o sistema de coordenadas da câmera especificado, podemos fazer a mudança do sistema de coordenadas do mundo ($X_w Y_w Z_w$) para o sistema de coordenadas da câmera ($X_v Y_v Z_v$)



Ponto Pos = (p_{xw}, p_{yw}, p_{zw}) [em coordenadas mundo]

Transformação de Visualização

- Com o sistema de coordenadas da câmera especificado, podemos fazer a mudança do sistema de coordenadas do mundo ($X_wY_wZ_w$) para o sistema de coordenadas da câmera ($X_vY_vZ_v$)



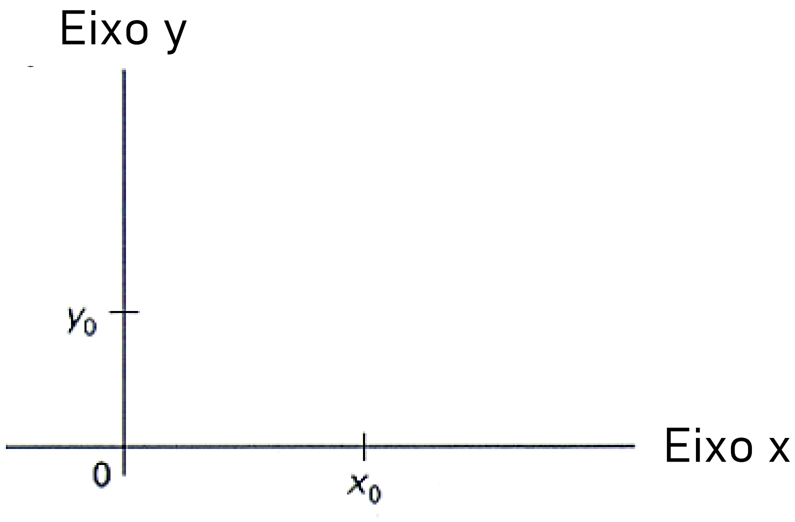
Ponto Pos = (p_{xw}, p_{yw}, p_{zw}) [em coordenadas mundo]

E agora? Como mudar de um sistema para o outro?

Mudança de sistemas de coordenadas

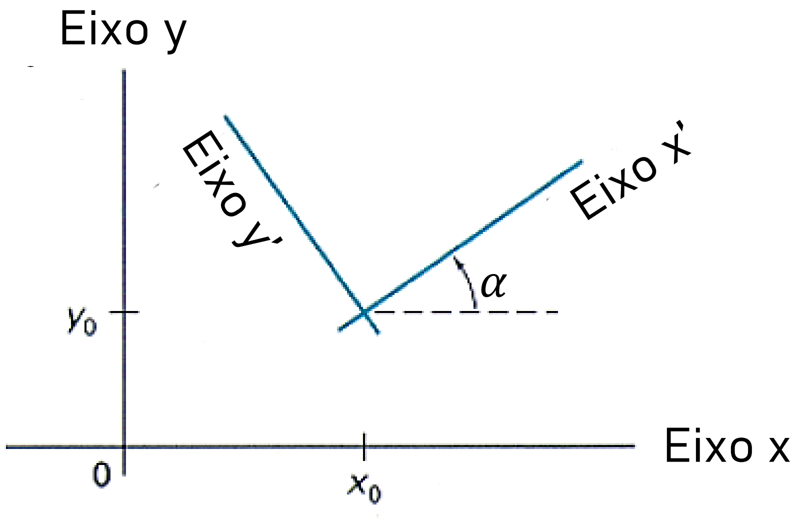
Mudança de sistemas de coordenadas (2D)

- Mudar de um sistema (cartesiano) xy para outro sistema (cartesiano) x'y'



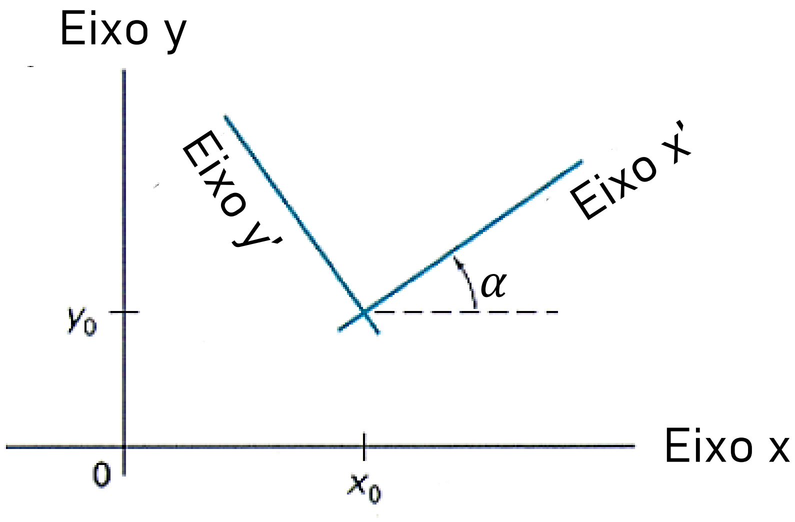
Mudança de sistemas de coordenadas (2D)

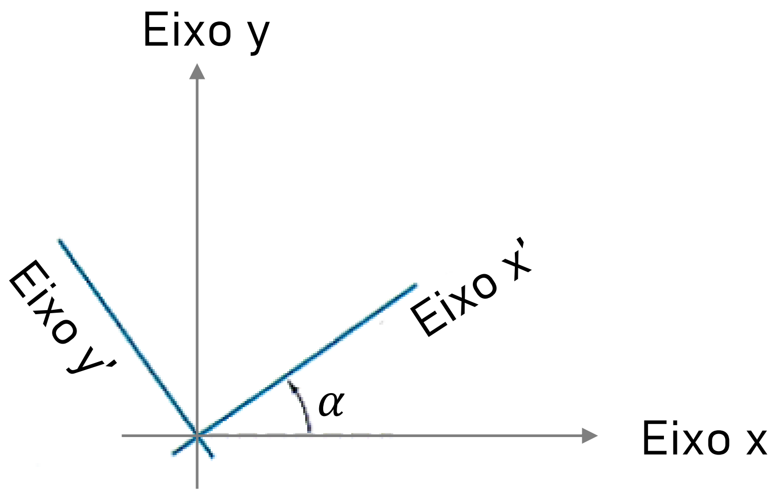
- Mudar de um sistema (cartesiano) xy para outro sistema (cartesiano) x'y'



Mudança de sistemas de coordenadas (2D)

- Mudar de um sistema (cartesiano) xy para outro sistema (cartesiano) x'y'



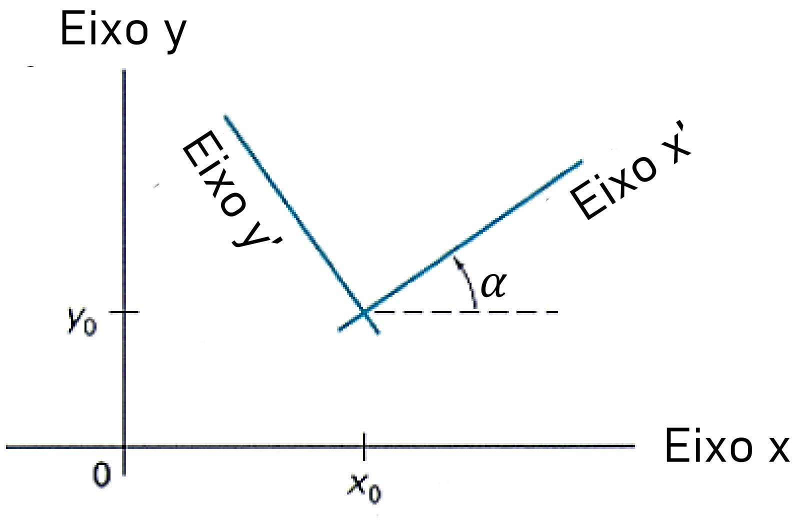


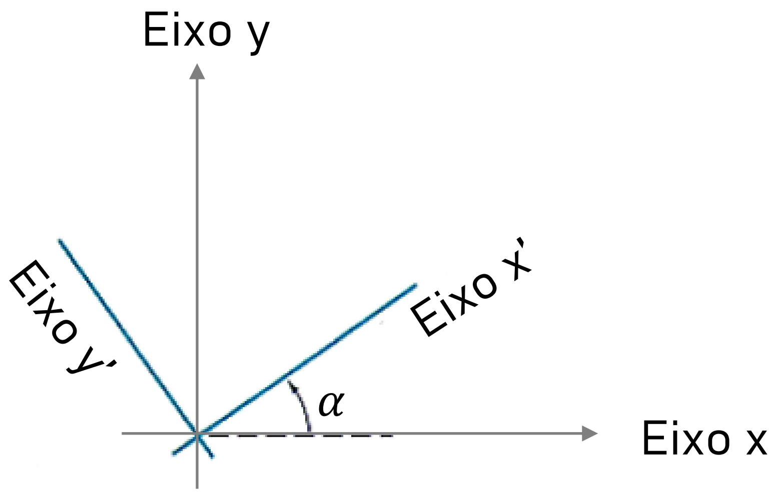
Translação para coincidir
as duas origens

$T(-x_0, -y_0)$

Mudança de sistemas de coordenadas (2D)

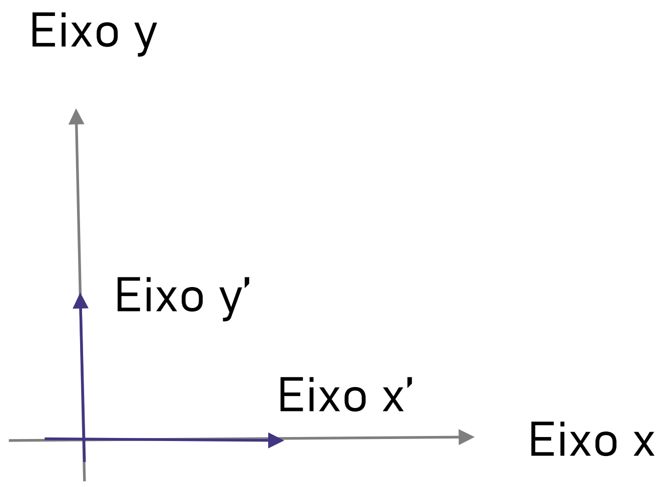
- Mudar de um sistema (cartesiano) xy para outro sistema (cartesiano) x'y'





Translação para coincidir as duas origens

$T(-x_0, -y_0)$



Rotação para coincidir o eixo x' com o x

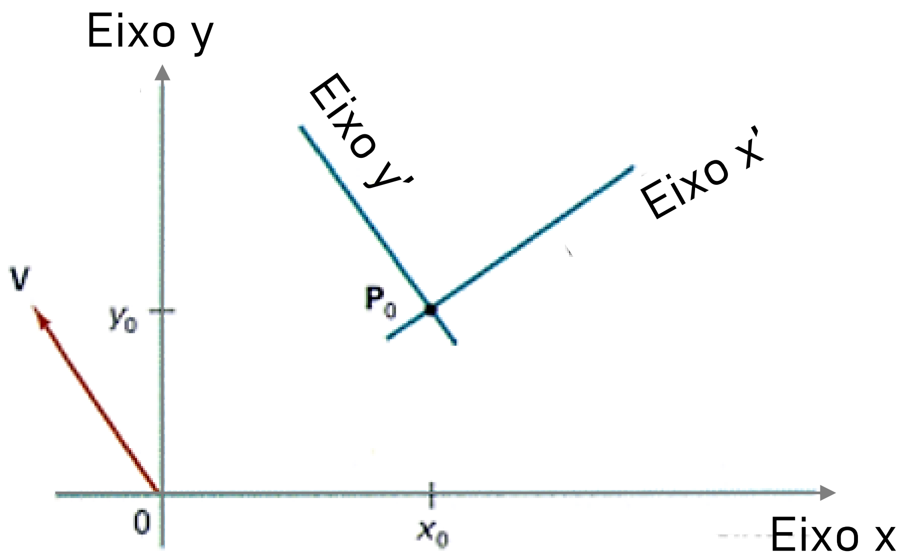
$R(-\alpha)$

Mudança de sistemas de coordenadas (2D)

- Definindo a matriz de rotação sem depender do ângulo α
  - Usando a orientação final desejada
  - Precisamos obter os vetores unitários u e v que dizem a direção do eixo x' e do eixo y'

Mudança de sistemas de coordenadas (2D)

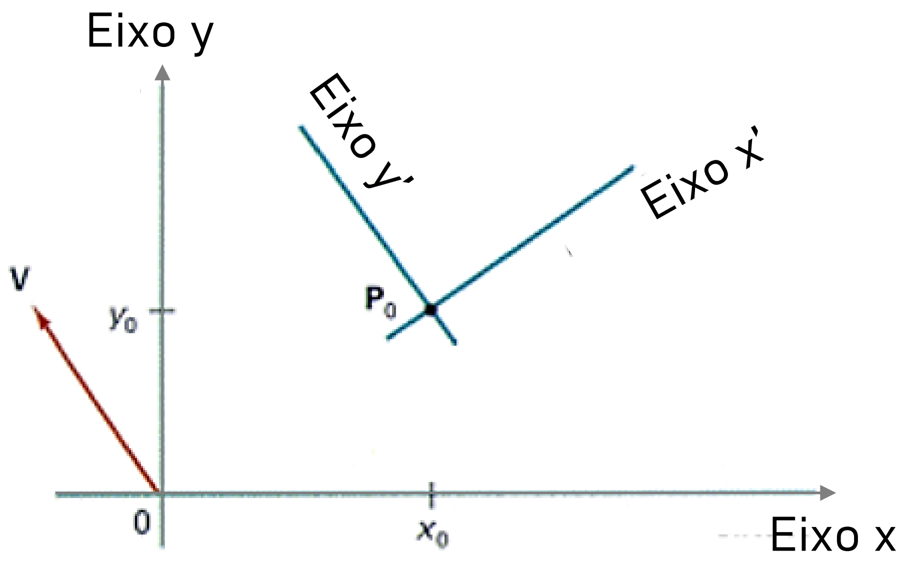
- Definindo a matriz de rotação sem depender do ângulo α
  - Usando a orientação final desejada
  - Precisamos obter os vetores unitários u e v que dizem a direção do eixo x' e do eixo y'



Vetor v paralelo a y'
passando pela origem de xy

Mudança de sistemas de coordenadas (2D)

- Definindo a matriz de rotação sem depender do ângulo α
  - Usando a orientação final desejada
  - Precisamos obter os vetores unitários u e v que dizem a direção do eixo x’ e do eixo y’

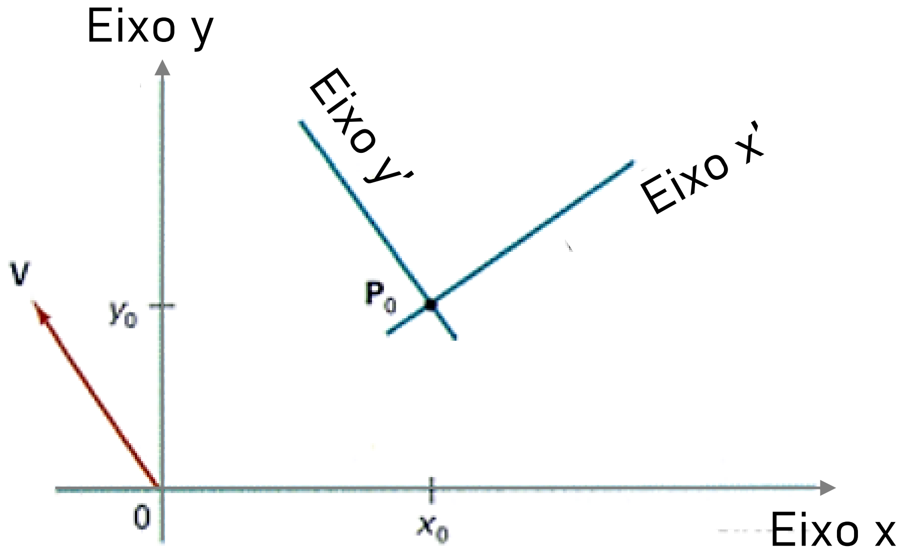


$$v = \frac{V}{|V|} = (v_x, v_y)$$

Vetor v paralelo a y’
passando pela origem de xy

Mudança de sistemas de coordenadas (2D)

- Definindo a matriz de rotação sem depender do ângulo α
  - Usando a orientação final desejada
  - Precisamos obter os vetores unitários u e v que dizem a direção do eixo x' e do eixo y'



Vetor v paralelo a y'
passando pela origem de xy

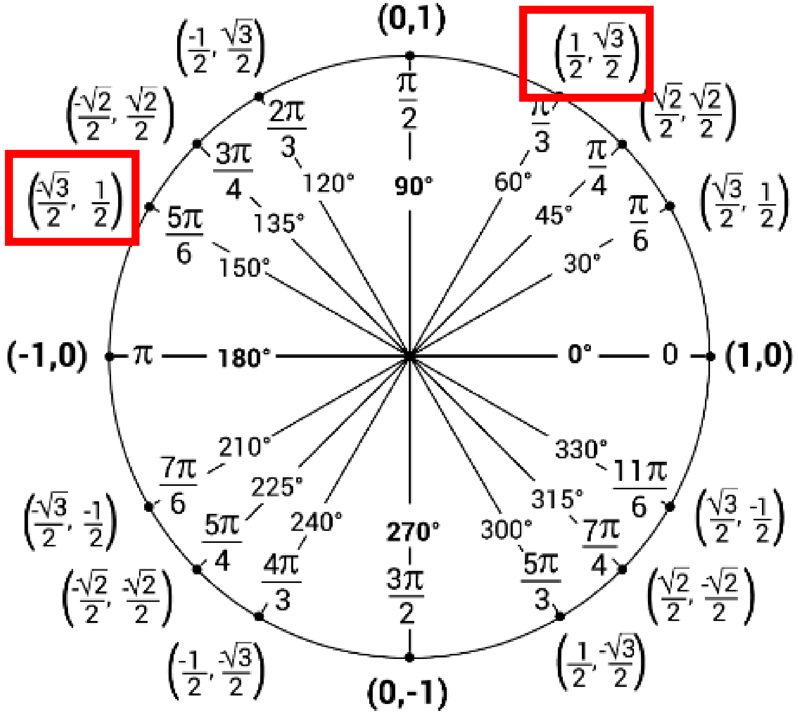
$$v = \frac{V}{|V|} = (\boxed{v_x}, \boxed{v_y})$$

$$u = (\boxed{v_y}, -\boxed{v_x}) = (u_x, u_y)$$

Por quê?

Mudança de sistemas de coordenadas (2D)

- Definindo a matriz de rotação sem depender do ângulo α
  - Usando a orientação final desejada
  - Precisamos obter os vetores unitários u e v que dizem a direção do eixo x' e do eixo y'



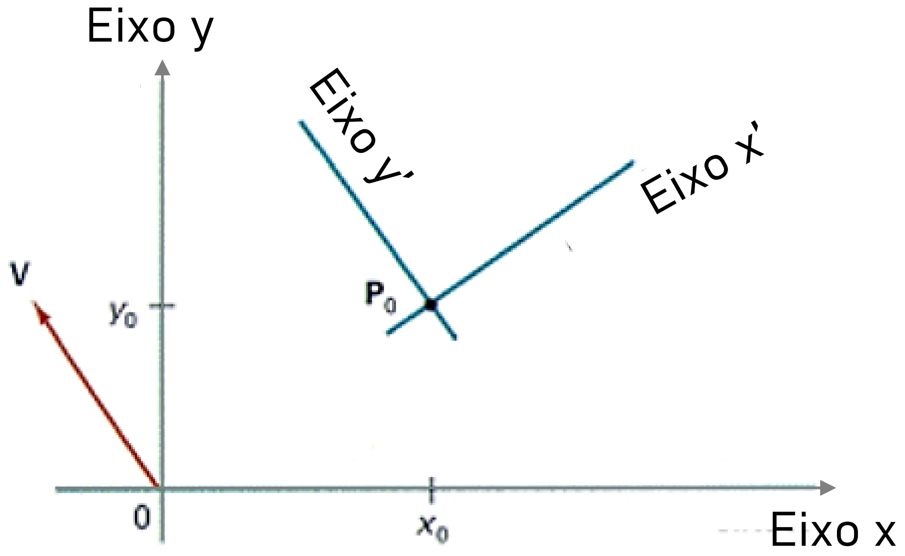
$$v = \frac{V}{|V|} = (\boxed{v_x}, \boxed{v_y})$$

$$u = (\boxed{v_y}, -\boxed{v_x}) = (u_x, u_y)$$

Por quê?

Mudança de sistemas de coordenadas (2D)

- Definindo a matriz de rotação sem depender do ângulo α
  - Usando a orientação final desejada
  - Precisamos obter os vetores unitários u e v que dizem a direção do eixo x' e do eixo y'



Vetor v paralelo a y' passando pela origem de xy

$$v = \frac{V}{|V|} = (\boxed{v_x}, \boxed{v_y})$$

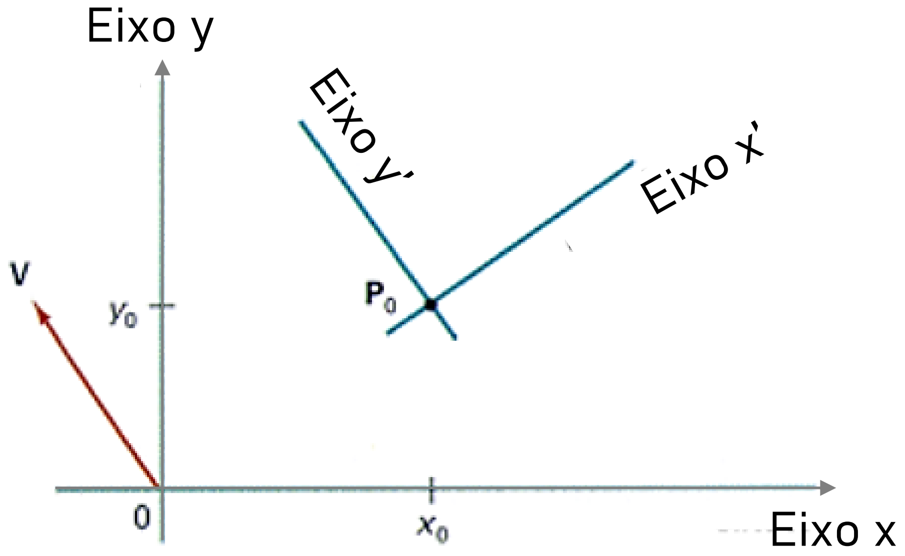
$$u = (\boxed{v_y}, -\boxed{v_x}) = (u_x, u_y)$$

$$R = \begin{bmatrix} u_x & u_y & 0 \\ v_x & v_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Detalhes aqui (seções 4.2 a 4.4)

Mudança de sistemas de coordenadas (2D)

- Definindo a matriz de rotação sem depender do ângulo α
  - Usando a orientação final desejada
  - Precisamos obter os vetores unitários u e v que dizem a direção do eixo x’ e do eixo y’



Vetor v paralelo a y’ passando pela origem de xy

$$v = \frac{V}{|V|} = (\boxed{v_x}, \boxed{v_y})$$

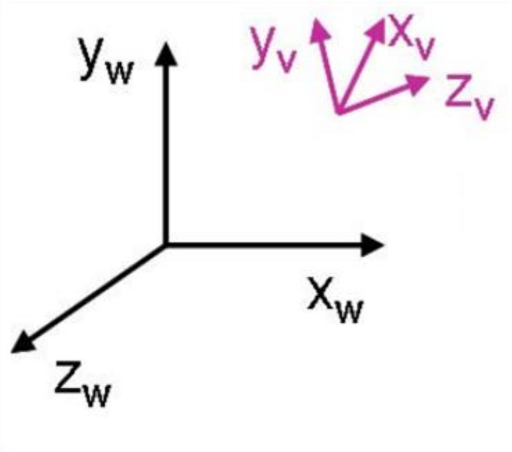
$$u = (\boxed{v_y}, -\boxed{v_x}) = (u_x, u_y)$$

$$R = \begin{bmatrix} u_x & u_y & 0 \\ v_x & v_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Vocês se lembram o porquê disso?

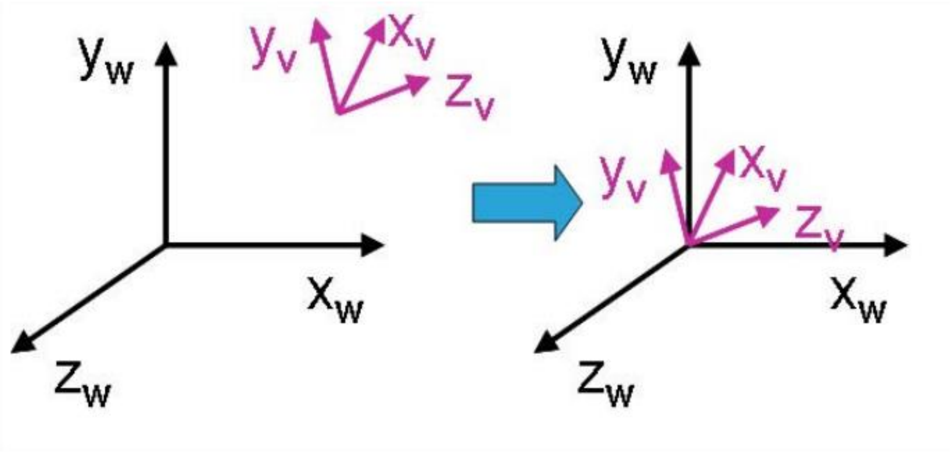
Mudança de sistemas de coordenadas (3D)

- Processo análogo ao 2D.
- Vamos mudar do sistema $X_wY_wZ_w$ para o $X_vY_vZ_v$



Mudança de sistemas de coordenadas (3D)

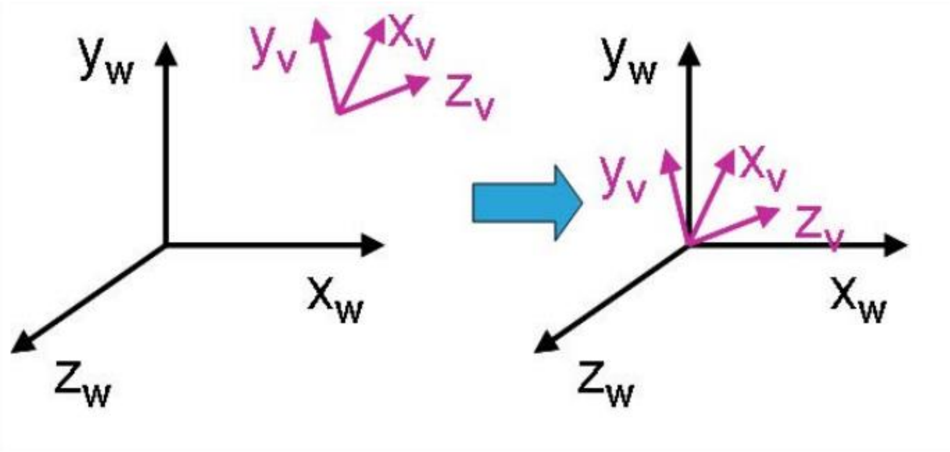
- Processo análogo ao 2D.
- Vamos mudar do sistema $X_wY_wZ_w$ para o $X_vY_vZ_v$



1º passo = translação

Mudança de sistemas de coordenadas (3D)

- Processo análogo ao 2D.
- Vamos mudar do sistema $X_w Y_w Z_w$ para o $X_v Y_v Z_v$



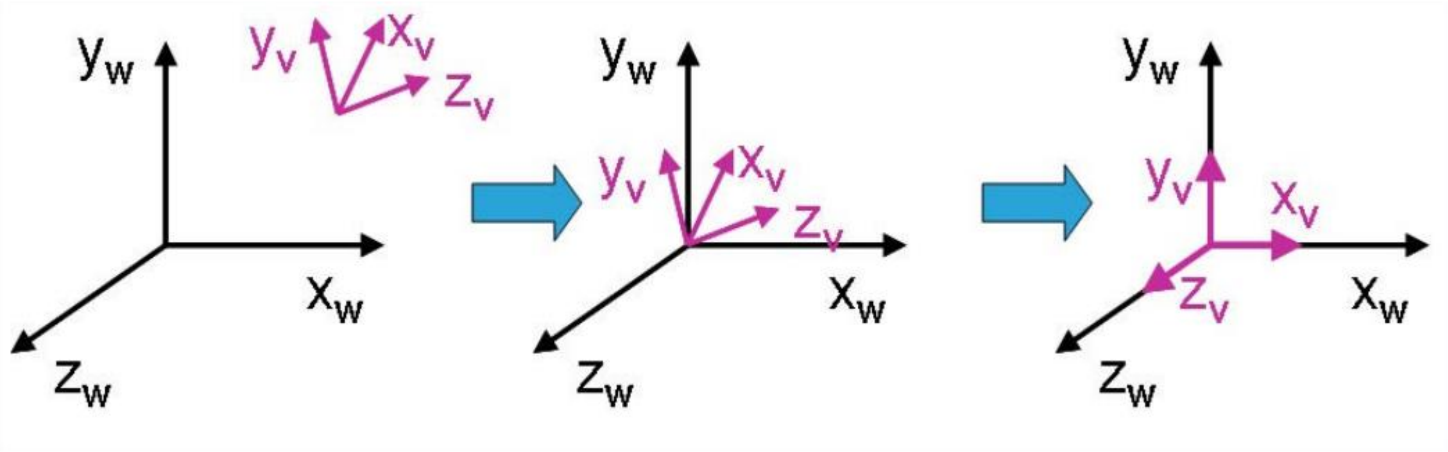
1º passo = translação

Assumindo a origem do sistema $X_v Y_v Z_v$ sendo o ponto (x_0, y_0, z_0):

$$T = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Mudança de sistemas de coordenadas (3D)

- Processo análogo ao 2D.
- Vamos mudar do sistema $X_wY_wZ_w$ para o $X_vY_vZ_v$



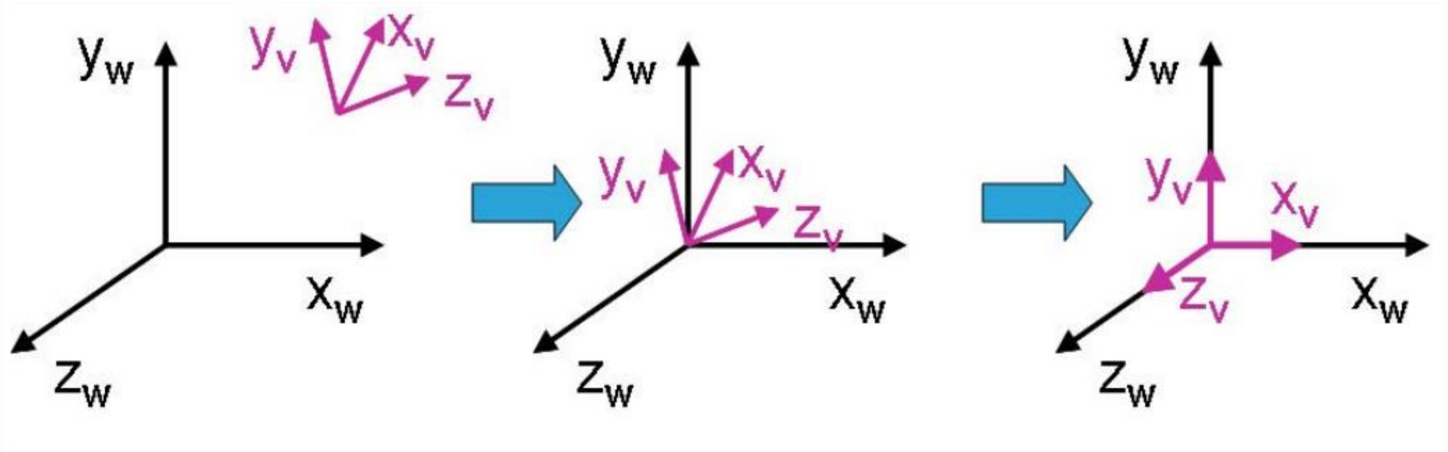
2º passo = rotações

$$R = \begin{bmatrix} i_{xv} & j_{xv} & k_{xv} & 0 \\ i_{yv} & j_{yv} & k_{yv} & 0 \\ i_{zv} & j_{zv} & k_{zv} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

, onde cada linha contém o vetor unitário de um eixo.

Mudança de sistemas de coordenadas (3D)

- Processo análogo ao 2D.
- Vamos mudar do sistema $X_wY_wZ_w$ para o $X_vY_vZ_v$

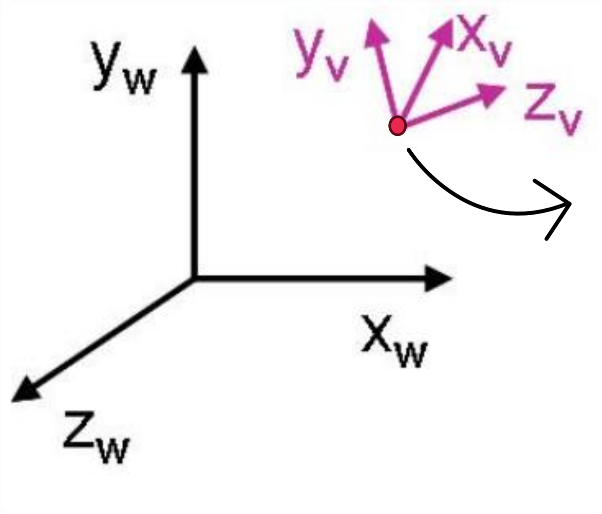


Para mudar de um sistema para o outro, faz-se $R \cdot T$

De volta ao problema da câmera...

Transformação de Visualização

- Com o sistema de coordenadas da câmera especificado, podemos fazer a mudança do sistema de coordenadas do mundo ($X_w Y_w Z_w$) para o sistema de coordenadas da câmera ($X_v Y_v Z_v$)

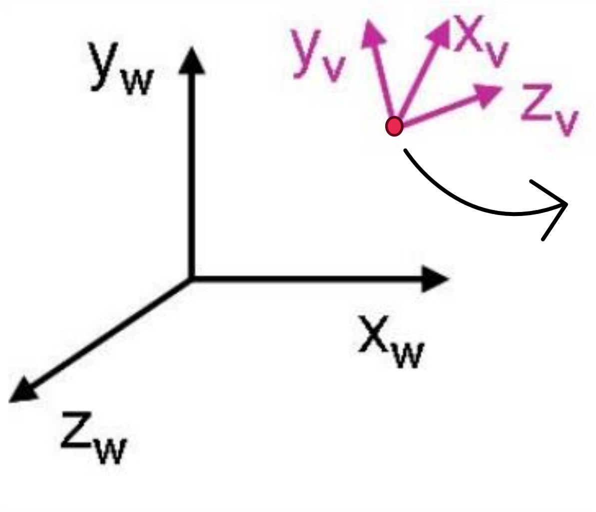


Ponto Pos = (p_x, p_y, p_z) [em coordenadas mundo]
(é onde a câmera está)

Agora já sabemos fazer essa mudança! =)

Transformação de Visualização

- Com o sistema de coordenadas da câmera especificado, podemos fazer a mudança.



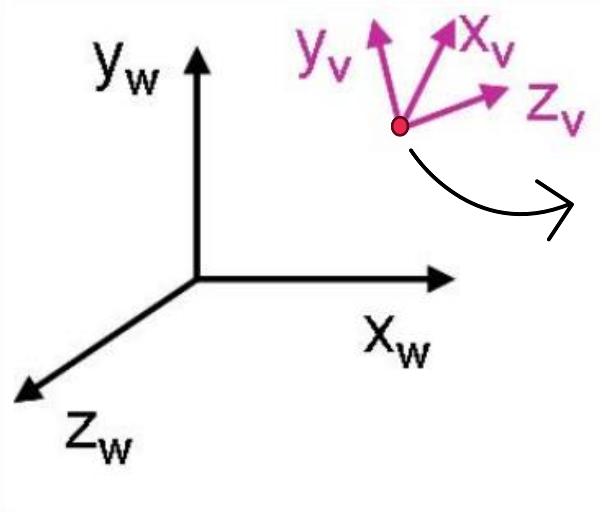
Ponto Pos = (p_x, p_y, p_z) [em coordenadas mundo]

1º passo = translação

$$T = \begin{bmatrix} 1 & 0 & 0 & -p_x \\ 0 & 1 & 0 & -p_y \\ 0 & 0 & 1 & -p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformação de Visualização

- Com o sistema de coordenadas da câmera especificado, podemos fazer a mudança.



Ponto Pos = (p_x, p_y, p_z) [em coordenadas mundo]

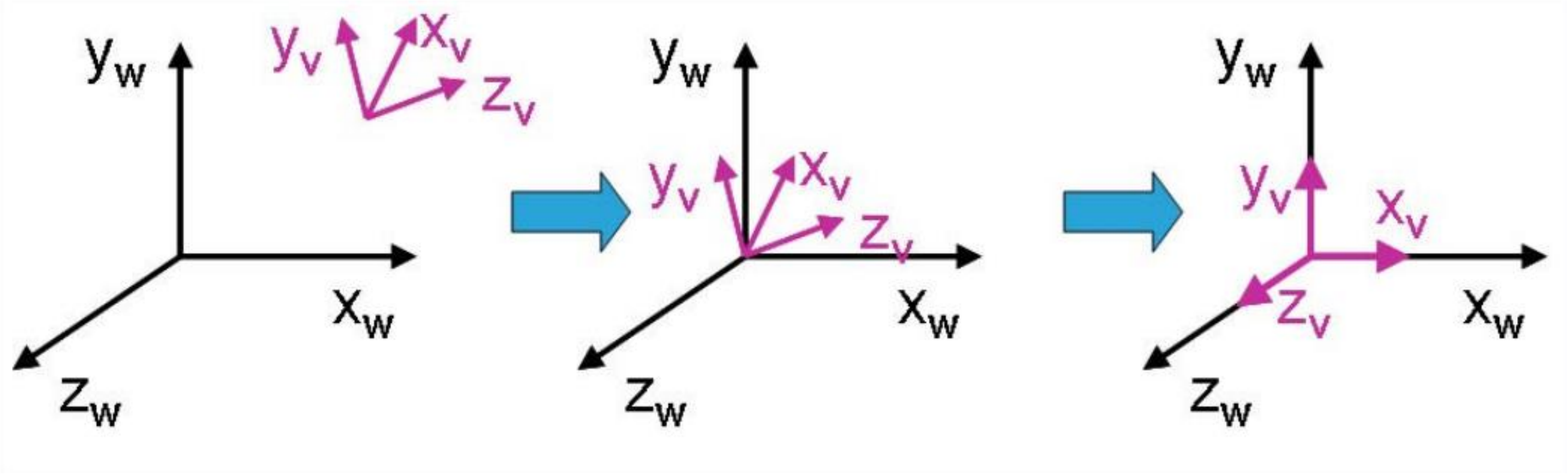
2º passo = rotações

$$R = \begin{bmatrix} i_{xc} & j_{xc} & k_{xc} & 0 \\ i_{yc} & j_{yc} & k_{yc} & 0 \\ i_{zc} & j_{zc} & k_{zc} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\overrightarrow{X_c} = (i_{xc}, j_{xc}, k_{xc})$$
$$\overrightarrow{Y_c} = (i_{yc}, j_{yc}, k_{yc})$$
$$\overrightarrow{Z_c} = (i_{zc}, j_{zc}, k_{zc})$$

Transformação de Visualização

- Transformação realizada!



Transformação de Visualização - View matrix

- Multiplicando as matrizes de translação e rotação, chegamos na matriz conhecida por **view matrix**, responsável pela transformação de visualização.

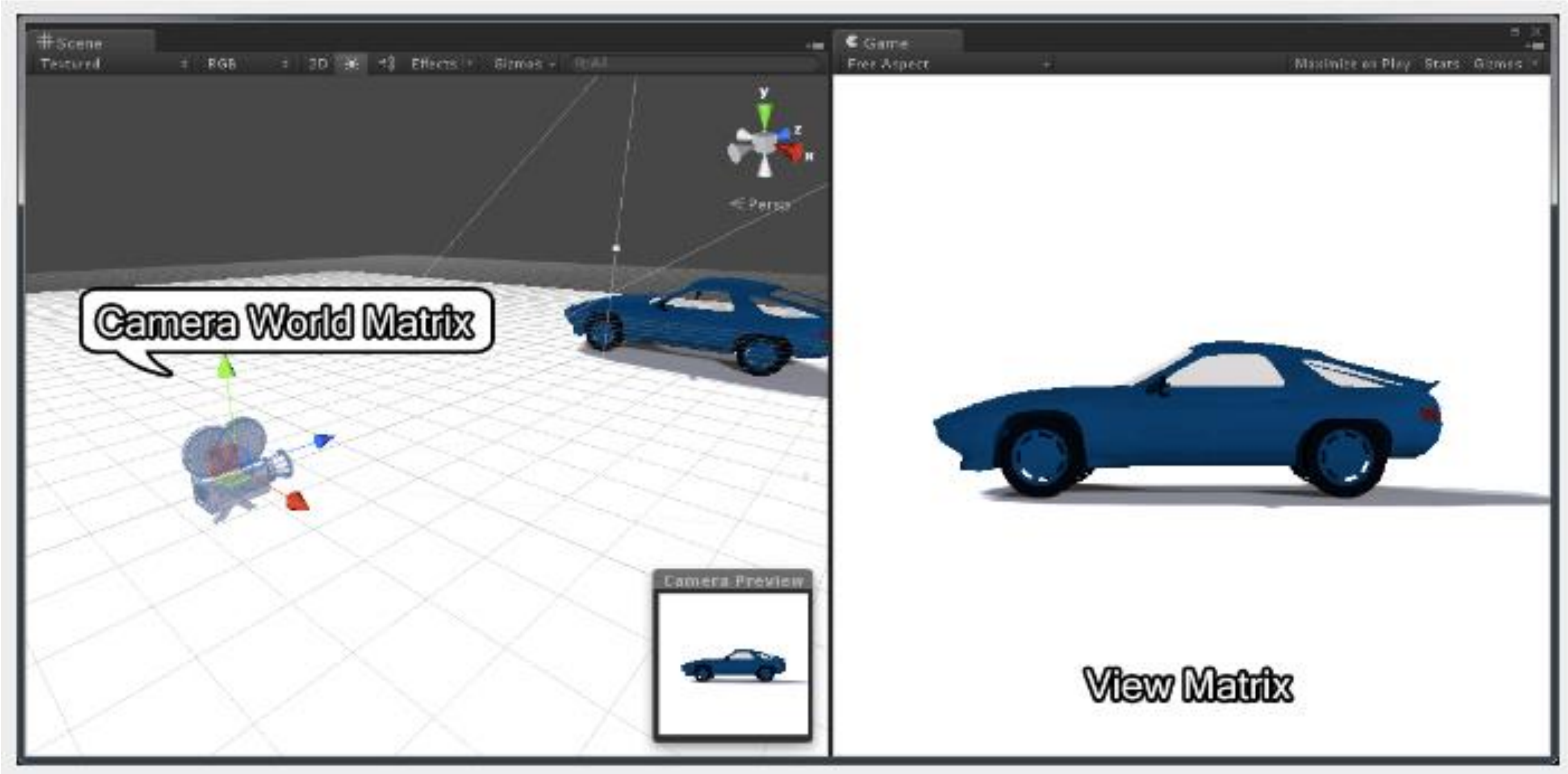
- Para fazer a mudança do sistema do mundo para o sistema da câmera, basta **multiplicar as coordenadas de cada vértice** de cada objeto por essa matriz.

Vetor coluna à direita da matriz

$$\text{Mview} = R \cdot T = \begin{bmatrix} i_{xc} & j_{xc} & k_{xc} & -Pos \cdot Xc \\ i_{yc} & j_{yc} & k_{yc} & -Pos \cdot Yc \\ i_{zc} & j_{zc} & k_{zc} & -Pos \cdot Zc \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\overrightarrow{X_c} = (i_{xc}, j_{xc}, k_{xc})$
$\overrightarrow{Y_c} = (i_{yc}, j_{yc}, k_{yc})$
$\overrightarrow{Z_c} = (i_{zc}, j_{zc}, k_{zc})$
$Pos = (p_x, p_y, p_z)$

Transformação de Visualização - View matrix



Fonte da imagem: https://www.3dgep.com/understanding-the-view-matrix/

View matrix

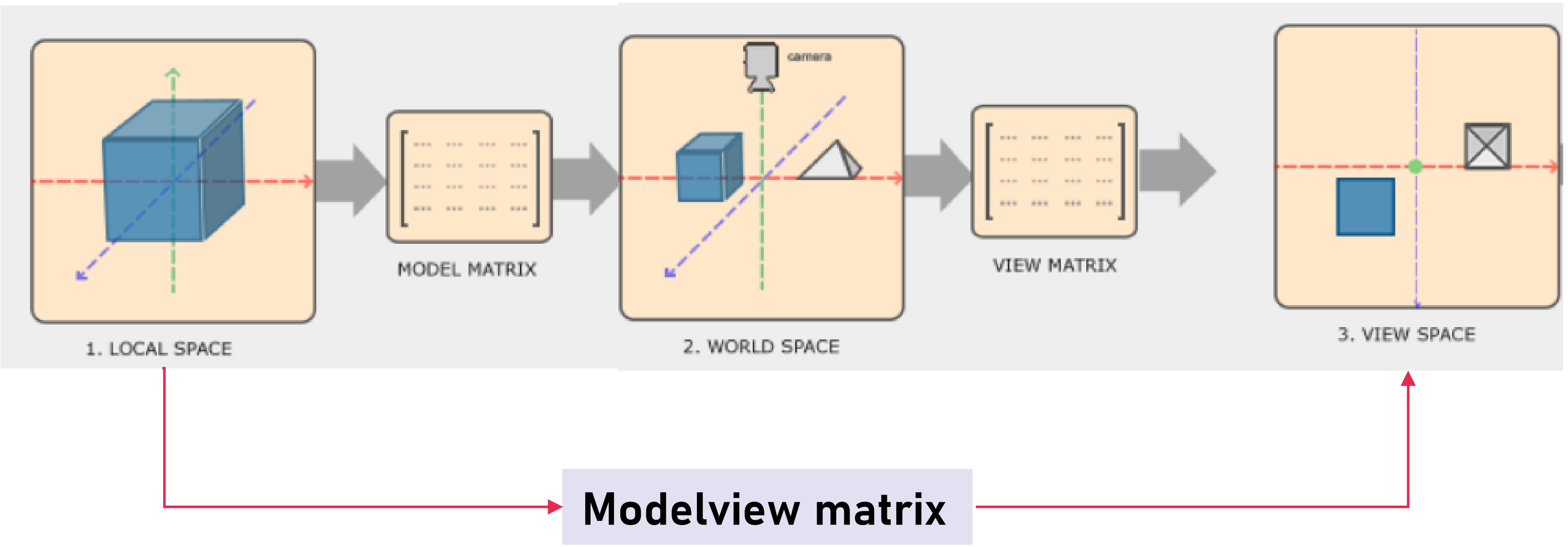
Jupyter Notebooks:

Aula08.Ex03

“Carregando Modelos” (mesmo exemplo visto na aula “Model”)

ModelView matrix

- Uma outra matriz, super importante para o pipeline gráfico, pode ser derivada agora: a **modelview matrix**



- Dada a model matrix M e a view matrix V, a modelview matrix ModelView é definida como:

$$\text{ModelView} = V \cdot M$$

Bibliografia

- Essa aula foi baseada no seguinte material:

- https://www.brunodorta.com.br/cg/glspaces *(Acessado em 23/08/2024)*

- Computação Gráfica, Slides de Ricardo Marcacini, Maria Cristina, Alaor Cervati. ICMC/USP.

- Hughes, J. F., Van Dam, A., Foley, J. D., McGuire, M., Feiner, S. K., & Sklar, D. F. (2014). Computer graphics: principles and practice. Terceira Edição. Pearson Education.

- 4ª edição do livro "Computer Graphics with OpenGL" (Hearn; Baker; Carithers, 2011)
  - Capítulo 10