

COMPUTAÇÃO GRÁFICA

PIPELINE DE VISUALIZAÇÃO

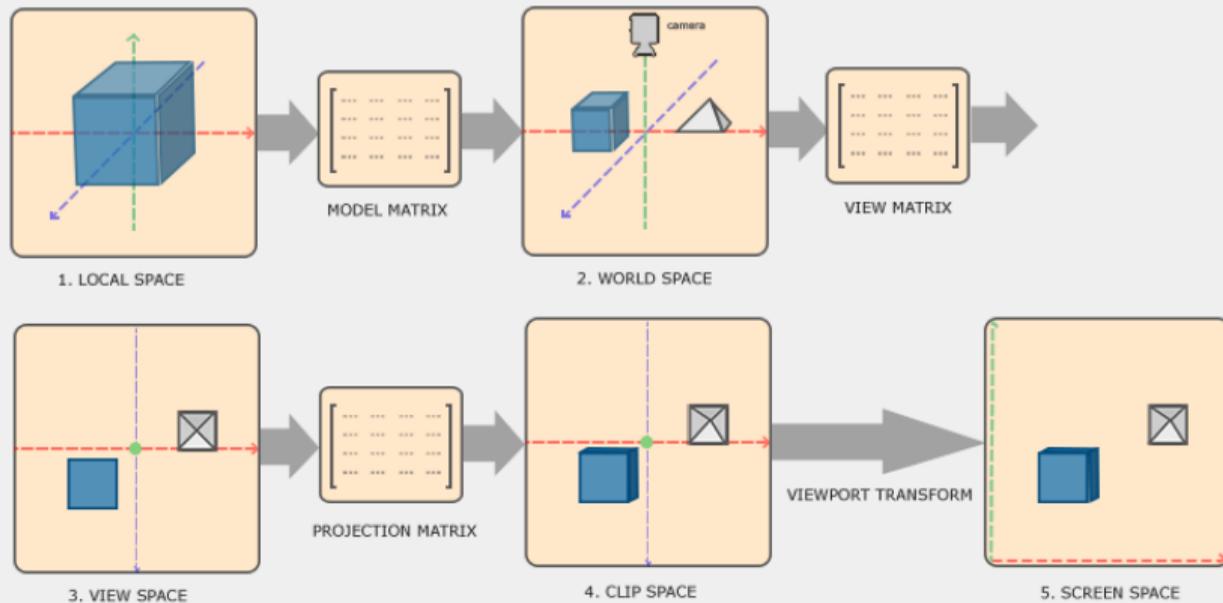
PROF. ALAOR CERVATI NETO

ICMC
SÃO CARLOS



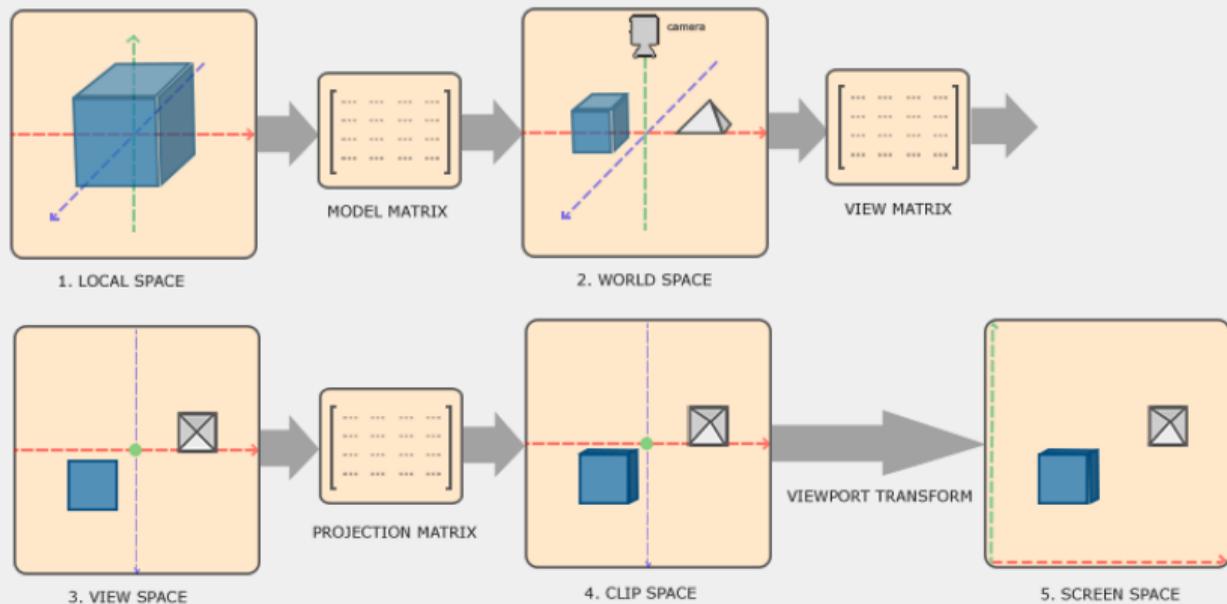
2021/2

ESPAÇO DE COORDENADAS



$$P' = \text{Projection} \times \text{View} \times \text{Model} \times P$$

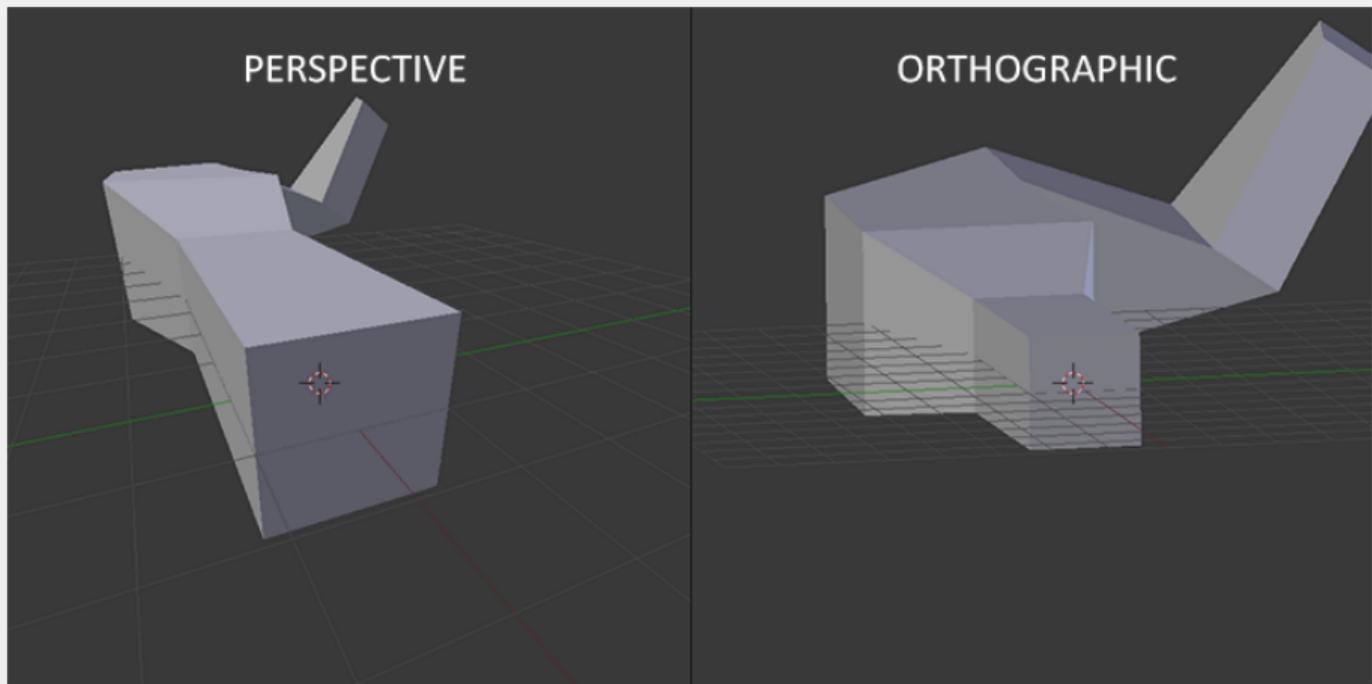
ESPAÇO DE COORDENADAS



$$P' = \boxed{\text{Projection}} \times \text{View} \times \text{Model} \times P$$

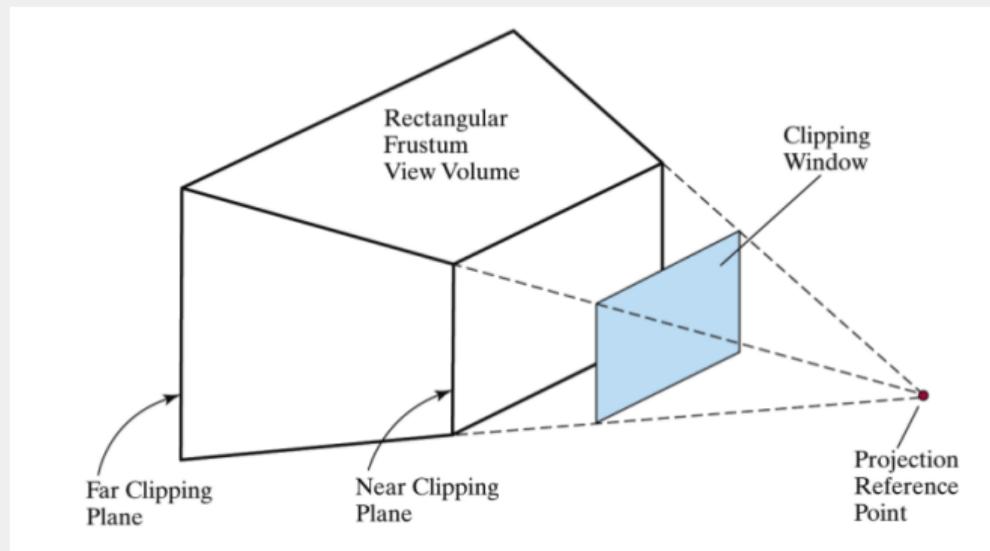
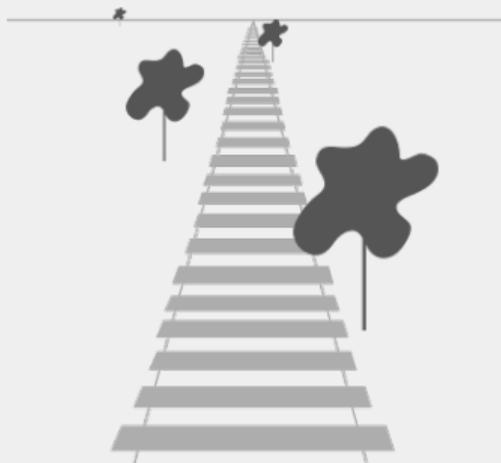
MATRIZ PROJECTION

Dois tipos de projeção:



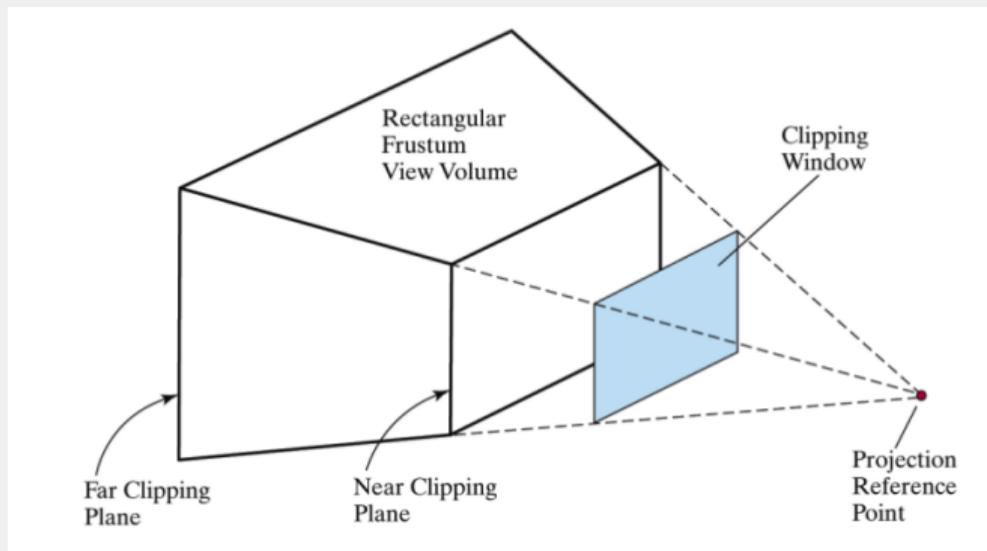
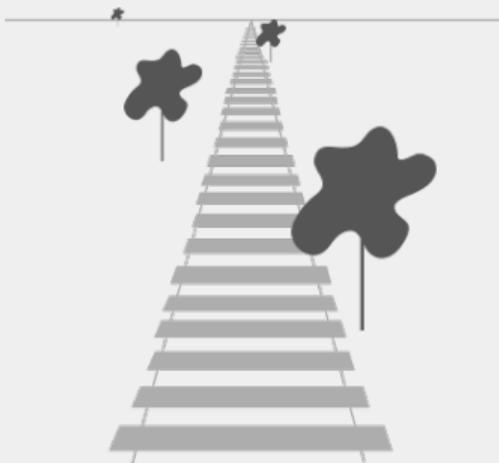
PROJEÇÃO PERSPECTIVA

PROJEÇÃO PERSPECTIVA



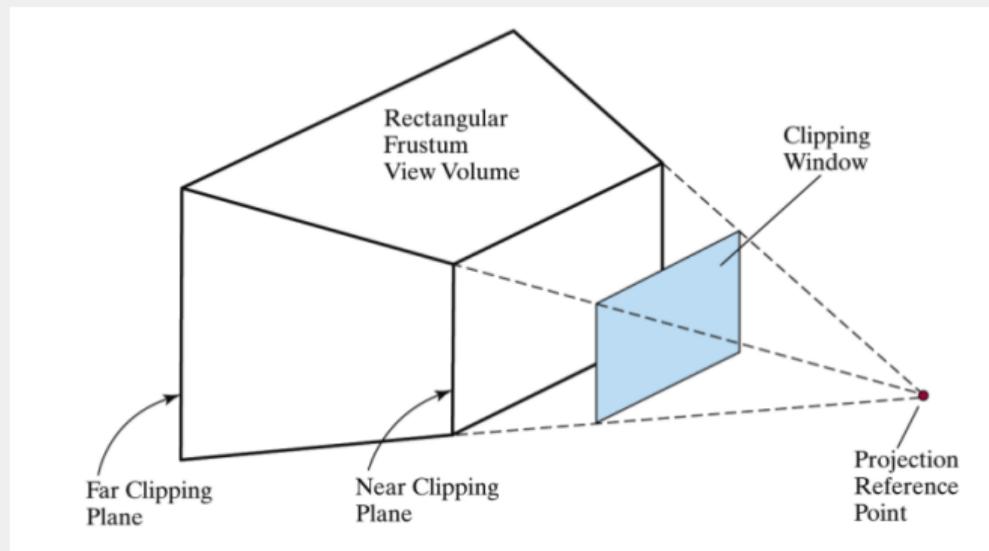
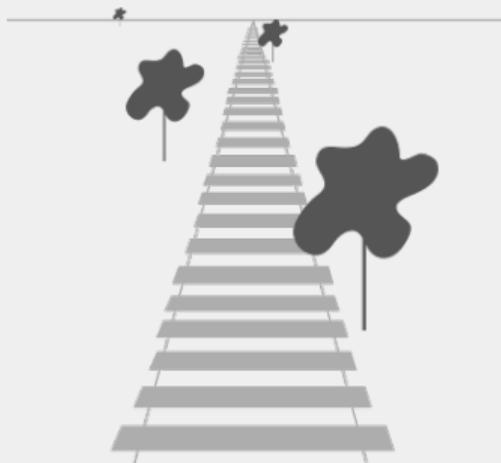
$$\left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w} \right)$$

PROJEÇÃO PERSPECTIVA



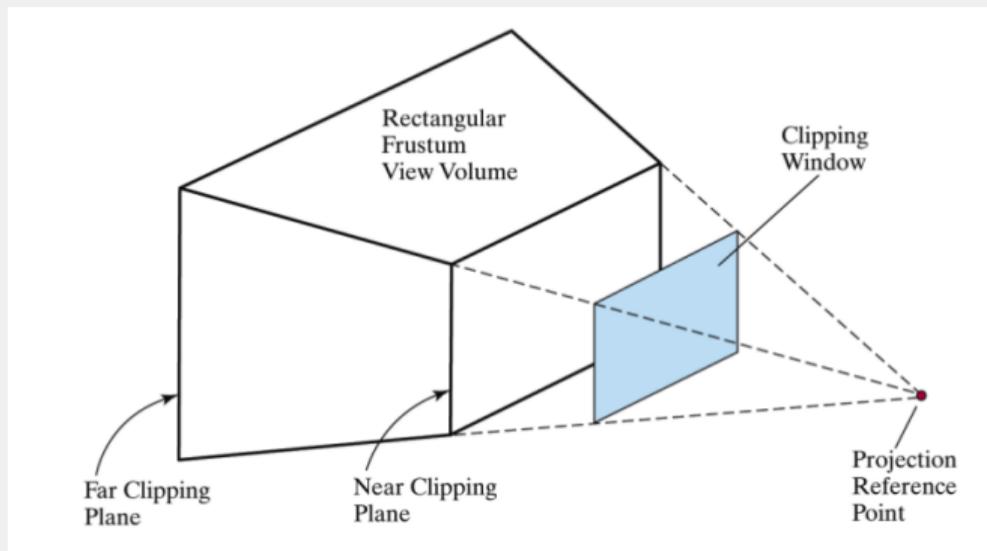
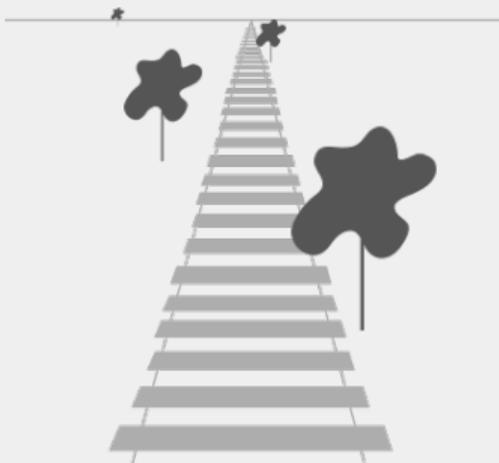
Janela de Recorte: cenas 3D projetadas/visíveis no plano de projeção.

PROJEÇÃO PERSPECTIVA



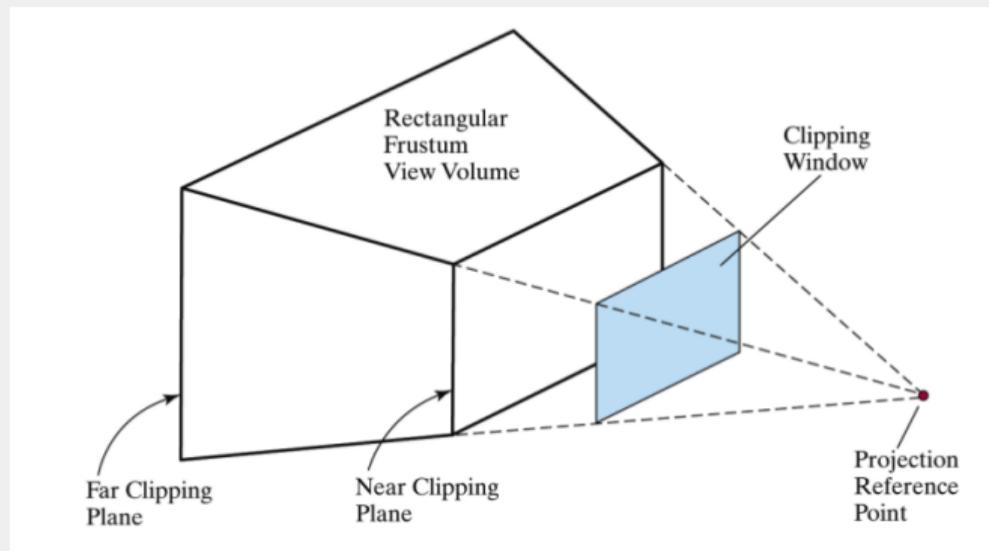
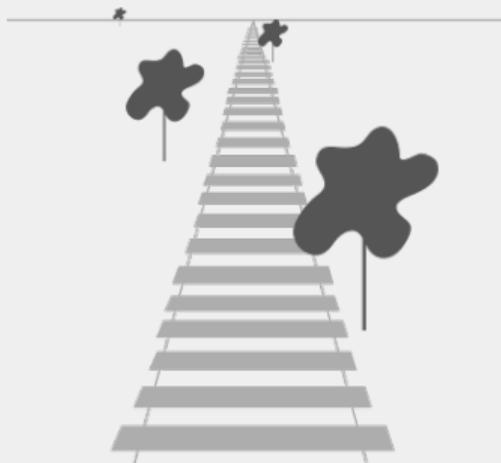
θ : ângulo do campo de visão.

PROJEÇÃO PERSPECTIVA



Planos de Recorte (*Near/Far*): perpendiculares ao eixo z_{view} .

PROJEÇÃO PERSPECTIVA



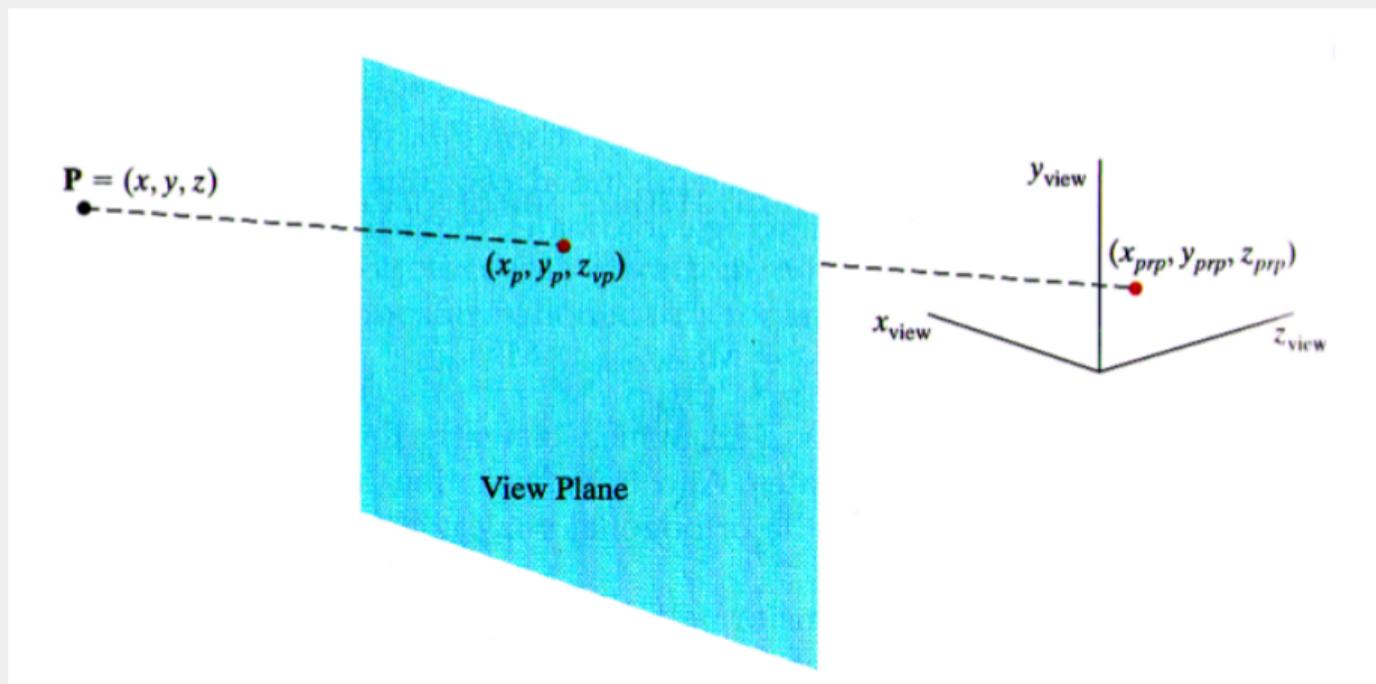
Frustum: porção visível do mundo.

MATRIZ DE PROJEÇÃO PERSPECTIVA

$$M_{\text{normpers}} = M_{\text{xyscale}} \cdot M_{\text{pers}}$$
$$M_{\text{xyscale}} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, M_{\text{pers}} = \begin{bmatrix} \frac{\text{width} \cdot \cot\left(\frac{\theta}{2}\right)}{2 \cdot \text{aspect}} & 0 & -X_{\text{prp}} & X_{\text{prp}} Z_{\text{vp}} \\ 0 & \frac{\text{width} \cdot \cot\left(\frac{\theta}{2}\right)}{2 \cdot \text{aspect}} & -Y_{\text{prp}} & Y_{\text{prp}} Z_{\text{vp}} \\ 0 & 0 & S_z & t_z \\ 0 & 0 & -1 & Z_{\text{prp}} \end{bmatrix}$$
$$M_{\text{normpers}} = \begin{bmatrix} \frac{\cot\left(\frac{\theta}{2}\right)}{\text{aspect}} & 0 & 0 & 0 \\ 0 & \cot\left(\frac{\theta}{2}\right) & 0 & 0 \\ 0 & 0 & \frac{Z_{\text{near}} + Z_{\text{far}}}{Z_{\text{near}} - Z_{\text{far}}} & -\frac{2 \cdot Z_{\text{near}} \cdot Z_{\text{far}}}{Z_{\text{near}} - Z_{\text{far}}} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

MATRIZ DE PROJEÇÃO PERSPECTIVA

Primeiramente, definimos um cenário simplificado:



MATRIZ DE PROJEÇÃO PERSPECTIVA

Escrevemos x_p e y_p em função dos outros pontos:

$$x_p = x \left(\frac{z_{prp} - z_{vp}}{z_{prp} - z} \right) + x_{prp} \left(\frac{z_{vp} - z}{z_{prp} - z} \right)$$
$$y_p = y \left(\frac{z_{prp} - z_{vp}}{z_{prp} - z} \right) + y_{prp} \left(\frac{z_{vp} - z}{z_{prp} - z} \right)$$

Convertendo estas equações em uma matriz de transformação geométrica 3D:

- Coordenadas homogêneas: $x_p = \frac{x_h}{h}$, $y_p = \frac{y_h}{h}$.
- Parâmetro homogêneo: $h = z_{prp} - z$.

Portanto:

$$x_p = x (z_{prp} - z_{vp}) + x_{prp} (z_{prp} - z_{vp})$$
$$y_p = y (z_{prp} - z_{vp}) + y_{prp} (z_{prp} - z_{vp})$$

MATRIZ DE PROJEÇÃO PERSPECTIVA

Definindo uma matriz perspectiva M_{pers} para $P_h = M_{\text{pers}} \cdot P$ onde:

- P é o ponto a ser projetado.
- P_h é o ponto projetado em coordenadas homogêneas.

Uma matriz possível:

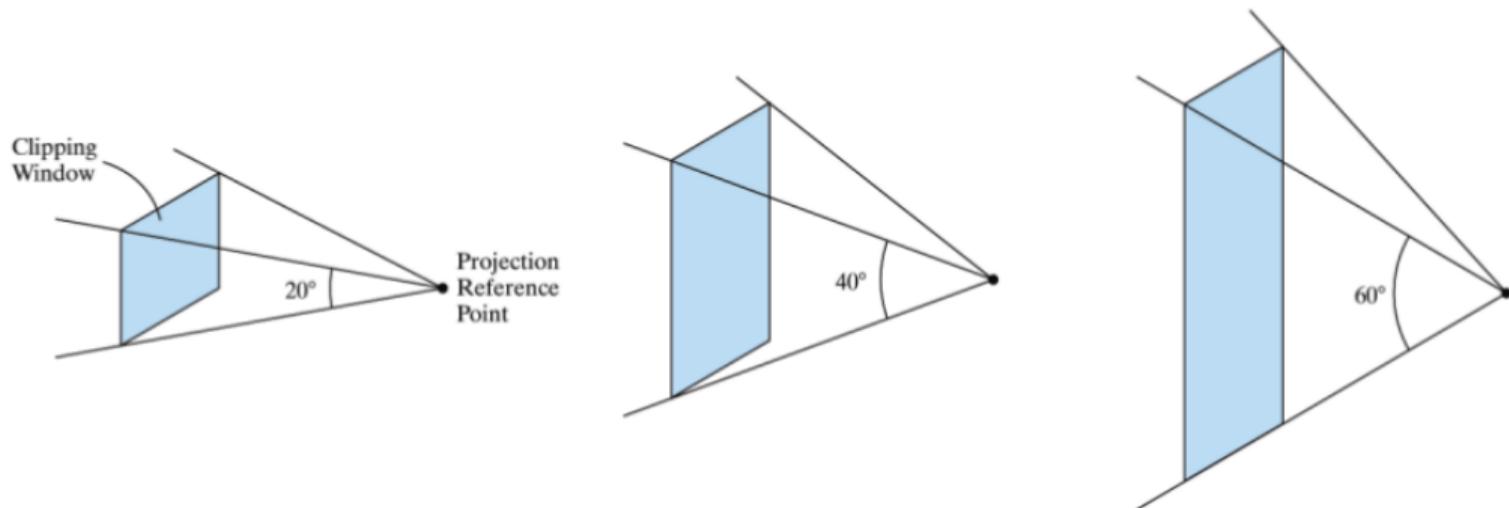
$$M_{\text{pers}} = \begin{bmatrix} z_{\text{prp}} - z_{\text{vp}} & 0 & -x_{\text{prp}} & x_{\text{prp}}z_{\text{vp}} \\ 0 & z_{\text{prp}} - z_{\text{vp}} & -y_{\text{prp}} & y_{\text{prp}}z_{\text{vp}} \\ 0 & 0 & s_z & t_z \\ 0 & 0 & -1 & z_{\text{prp}} \end{bmatrix}$$

M_{pers} pode ser obtida de forma direta:

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} z_{\text{prp}} - z_{\text{vp}} & 0 & -x_{\text{prp}} & x_{\text{prp}}z_{\text{vp}} \\ 0 & z_{\text{prp}} - z_{\text{vp}} & -y_{\text{prp}} & y_{\text{prp}}z_{\text{vp}} \\ 0 & 0 & s_z & t_z \\ 0 & 0 & -1 & z_{\text{prp}} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

MATRIZ DE PROJEÇÃO PERSPECTIVA

Precisamos normalizar a projeção em relação à escala, considerando a janela de recorte.



MATRIZ DE PROJEÇÃO PERSPECTIVA

Diminuir o ângulo do campo de visão diminui a janela de recorte:

- Mover o ponto de projeção para longe do plano de visão.
- *Zoom-in* de uma pequena região da cena.

Aumentar o ângulo do campo de visão aumenta a janela de recorte:

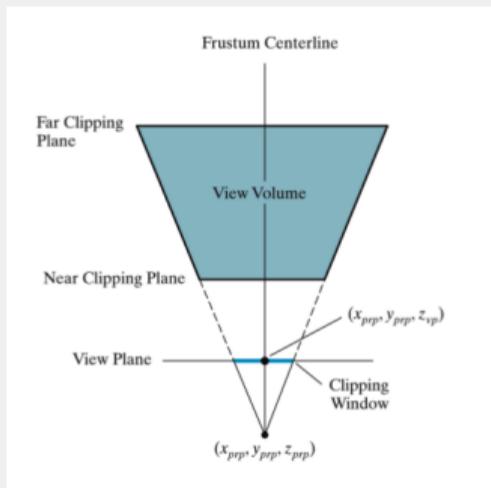
- Mover o ponto de projeção para próximo do plano de visão.
- *Zoom-out* da cena.

$$M_{xyscale} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

MATRIZ DE PROJEÇÃO PERSPECTIVA

Usamos um sistema de visão simplificada, que utiliza *frustum* simétrico:

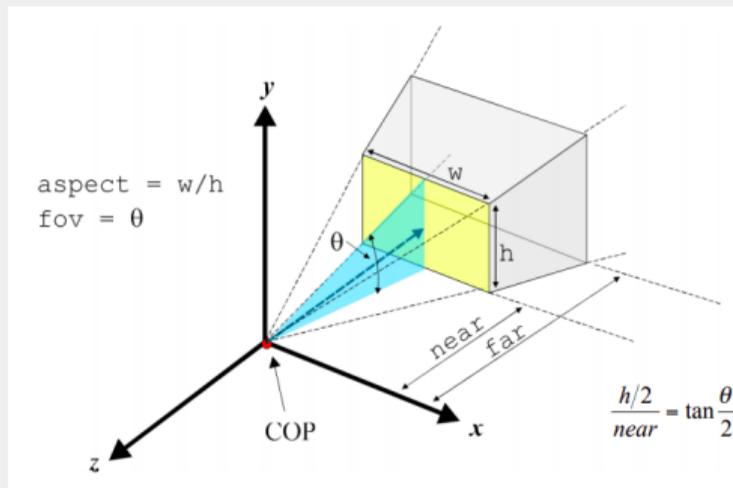
- A linha que liga o centro de projeção ao meio do plano de visão, perpendicular a esse, é a linha central do *frustum* simétrico.



MATRIZ DE PROJEÇÃO PERSPECTIVA

Usamos um sistema de visão simplificada, que utiliza *frustum* simétrico:

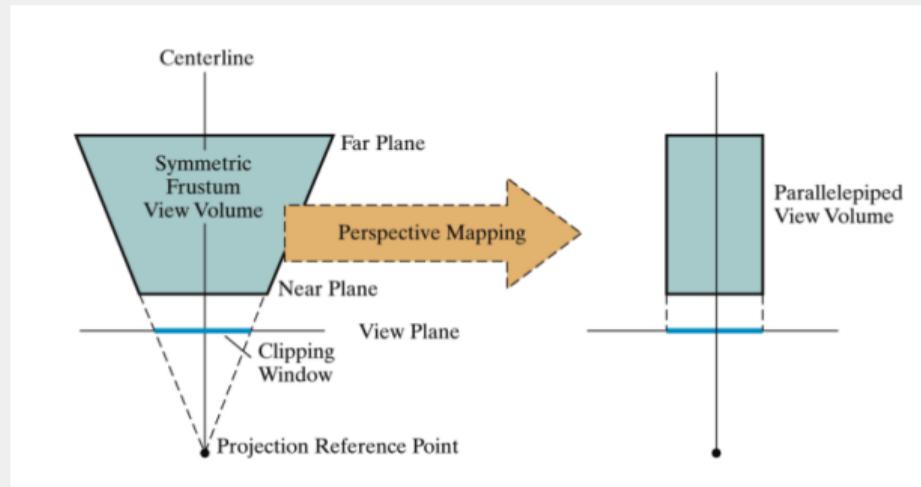
- A direção de observação é paralela a $-z$.
- O centro de projeção está na origem: $(x_{prp}, y_{prp}, z_{prp}) = (0, 0, 0)$.
- O plano de visão está sobre o plano de recorte: $z_{vp} = z_{near}$.



MATRIZ DE PROJEÇÃO PERSPECTIVA

Usamos um sistema de visão simplificada, que utiliza *frustum* simétrico:

- Simplifica localizações dentro do *frustum* para um paralelepípedo retangular.

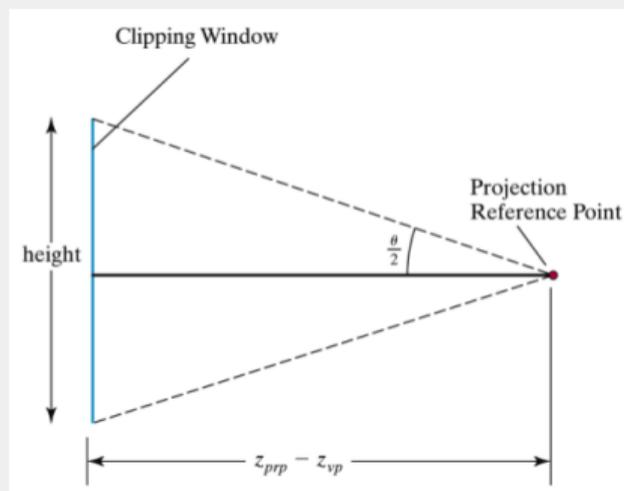


MATRIZ DE PROJEÇÃO PERSPECTIVA

Dado um *frustum* simétrico, normalizamos a projeção em relação à escala.

- Encontramos a largura e altura da janela de recorte (*clipping window*) conforme o ângulo.

$$\tan\left(\frac{\theta}{2}\right) = \frac{\frac{\text{height}}{2}}{z_{prp} - z_{vp}}$$



MATRIZ DE PROJEÇÃO PERSPECTIVA

Dado um *frustum* simétrico, normalizamos a projeção em relação à escala.

- Encontramos a largura e altura da janela de recorte (*clipping window*) conforme o ângulo.

$$height = 2 (z_{prp} - z_{vp}) \tan \left(\frac{\theta}{2} \right)$$

$$z_{prp} - z_{vp} = \frac{height}{2} \cot \left(\frac{\theta}{2} \right)$$

MATRIZ DE PROJEÇÃO PERSPECTIVA

Dado um *frustum* simétrico, normalizamos a projeção em relação à escala.

- Encontramos a largura e altura da janela de recorte (*clipping window*) conforme o ângulo.
- No caso da largura, vamos usar o "aspecto" da janela: $aspect = \frac{width}{height}$.

$$z_{prp} - z_{vp} = \frac{width \cdot \cot\left(\frac{\theta}{2}\right)}{2 \cdot aspect}$$

MATRIZ DE PROJEÇÃO PERSPECTIVA

Substituindo, temos a matriz perspectiva:

$$M_{\text{pers}} = \begin{bmatrix} \frac{\text{width} \cdot \cot\left(\frac{\theta}{2}\right)}{2 \cdot \text{aspect}} & 0 & -X_{\text{prp}} & X_{\text{prp}}Z_{\text{vp}} \\ 0 & \frac{\text{width} \cdot \cot\left(\frac{\theta}{2}\right)}{2 \cdot \text{aspect}} & -Y_{\text{prp}} & Y_{\text{prp}}Z_{\text{vp}} \\ 0 & 0 & S_z & t_z \\ 0 & 0 & -1 & Z_{\text{prp}} \end{bmatrix}$$

MATRIZ DE PROJEÇÃO PERSPECTIVA

Agora, calculando de fato a matriz normalizada:

$$M_{\text{normpers}} = M_{\text{xyscale}} \cdot M_{\text{pers}}$$
$$M_{\text{normpers}} = \begin{bmatrix} (z_{\text{prp}} - z_{\text{vp}}) S_x & 0 & (-x_{\text{prp}}) S_x & (x_{\text{prp}} z_{\text{vp}}) S_x \\ 0 & (z_{\text{prp}} - z_{\text{vp}}) S_y & (-y_{\text{prp}}) S_y & (y_{\text{prp}} z_{\text{vp}}) S_y \\ 0 & 0 & S_z & t_z \\ 0 & 0 & -1 & z_{\text{prp}} \end{bmatrix}$$

Utilizando a simplificação do *frustum* simétrico:

$$M_{\text{normpers}} = \begin{bmatrix} -z_{\text{near}} S_x & 0 & 0 & 0 \\ 0 & -z_{\text{near}} S_y & 0 & 0 \\ 0 & 0 & S_z & t_z \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

MATRIZ DE PROJEÇÃO PERSPECTIVA

Coordenadas homogêneas:

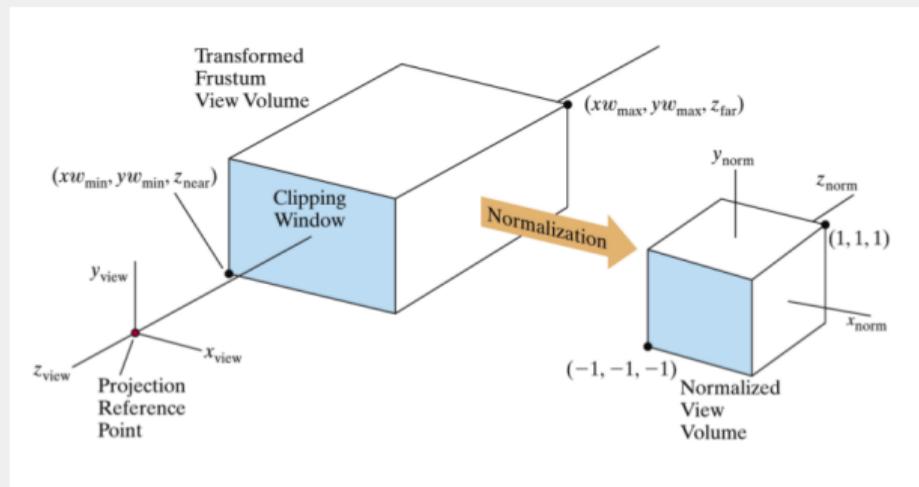
$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = M_{\text{normpers}} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Resultando nas coordenadas de projeção (parâmetro homogêneo $h = z_{prp} - z$):

$$x_p = \frac{x_h}{h} = \frac{-z_{\text{near}} \cdot s_x \cdot X}{-z}$$
$$y_p = \frac{y_h}{h} = \frac{-z_{\text{near}} \cdot s_y \cdot Y}{-z}$$
$$z_p = \frac{z_h}{h} = \frac{s_z \cdot Z + t_z}{-z}$$

MATRIZ DE PROJEÇÃO PERSPECTIVA

Entendendo a normalização:



Queremos mapear os pontos (x, y, z) para (x_p, y_p, z_p) de forma a respeitar os intervalos $(x, y, z) = (xw_{\min}, yw_{\min}, z_{\text{near}})$ em $(x_p, y_p, z_p) = (-1, -1, -1)$ e $(x, y, z) = (xw_{\max}, yw_{\max}, z_{\text{far}})$ em $(x_p, y_p, z_p) = (1, 1, 1)$.

MATRIZ DE PROJEÇÃO PERSPECTIVA

Portanto, temos:

$$xW_{\min} = -\frac{\text{width}}{2}, xW_{\max} = \frac{\text{width}}{2}$$
$$yW_{\min} = -\frac{\text{height}}{2}, yW_{\max} = \frac{\text{height}}{2}$$

Substituimos e resolvemos:

$$x_p = \frac{x_h}{h} = \frac{-z_{\text{near}} \cdot s_x \cdot x}{-z}$$
$$y_p = \frac{y_h}{h} = \frac{-z_{\text{near}} \cdot s_y \cdot y}{-z}$$
$$z_p = \frac{z_h}{h} = \frac{s_z \cdot z + t_z}{-z}$$

MATRIZ DE PROJEÇÃO PERSPECTIVA

Resolvendo as equações anteriores¹, encontramos:

$$s_x = \frac{2}{width}, s_y = \frac{2}{height}$$
$$s_z = \frac{z_{near} + z_{far}}{z_{near} - z_{far}}, t_z = -\frac{2 \cdot z_{near} \cdot z_{far}}{z_{near} - z_{far}}$$

Substituindo s_x , s_y , s_z e t_z , encontramos a matriz de projeção perspectiva normalizada:

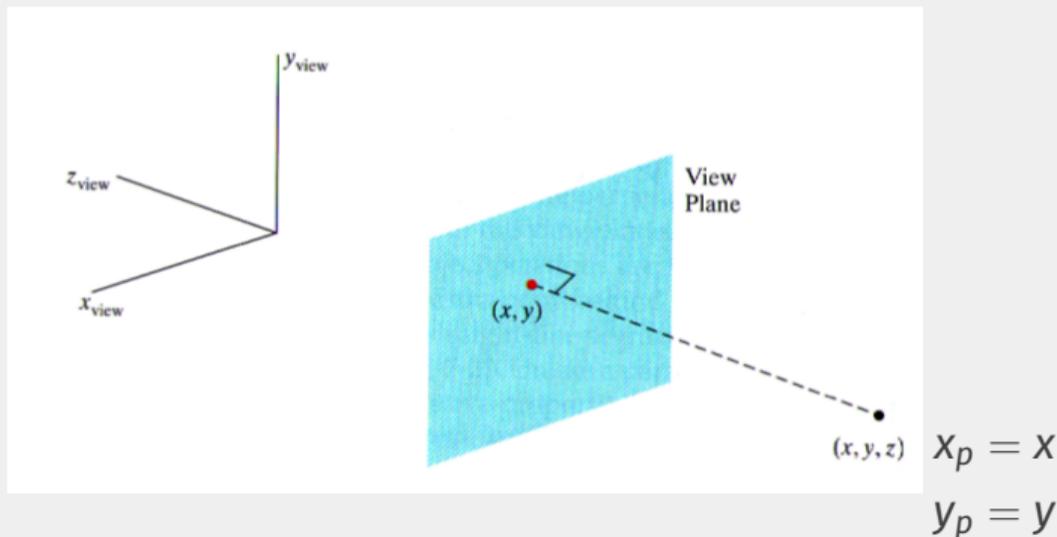
$$M_{normpers} = \begin{bmatrix} \frac{\cot(\frac{\theta}{2})}{aspect} & 0 & 0 & 0 \\ 0 & \cot(\frac{\theta}{2}) & 0 & 0 \\ 0 & 0 & \frac{z_{near} + z_{far}}{z_{near} - z_{far}} & -\frac{2 \cdot z_{near} \cdot z_{far}}{z_{near} - z_{far}} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

¹Resolução: http://www.songho.ca/opengl/gl_projectionmatrix.html

PROJEÇÃO ORTOGONAL

PROJEÇÃO ORTOGONAL

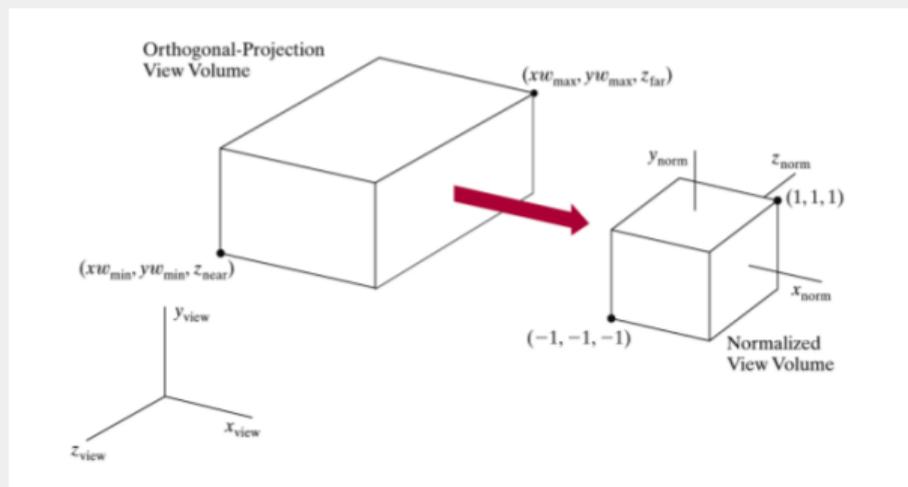
- Um vértice (x, y, z) em uma projeção ortogonal é mapeado diretamente para (x, y) .
- As coordenadas dentro do volume de visão já são as coordenadas de projeção.
- Paralela ao eixo Z_{view} .



PROJEÇÃO ORTOGONAL

Dado os parâmetros z_{near} , z_{far} e a janela de recorte, é preciso apenas mapear para um volume normalizado.

- Primeiro, estruturar o sistema de equações que faz a normalização.



PROJEÇÃO ORTOGONAL

Dado os parâmetros z_{near} , z_{far} e a janela de recorte, é preciso apenas mapear para um volume normalizado.

- Primeiro, estruturar o sistema de equações que faz a normalização.
- Segundo, "transformar" as equações no modelo de sistema de coordenadas homogêneas (matriz 4×4).

$$M_{\text{ortho,norm}} = \begin{bmatrix} \frac{2}{XW_{\text{max}} - XW_{\text{min}}} & 0 & 0 & -\frac{XW_{\text{max}} + XW_{\text{min}}}{XW_{\text{max}} - XW_{\text{min}}} \\ 0 & \frac{2}{YW_{\text{max}} - YW_{\text{min}}} & 0 & -\frac{YW_{\text{max}} + YW_{\text{min}}}{YW_{\text{max}} - YW_{\text{min}}} \\ 0 & 0 & -\frac{2}{Z_{\text{near}} - Z_{\text{far}}} & \frac{Z_{\text{near}} + Z_{\text{far}}}{Z_{\text{near}} - Z_{\text{far}}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- 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.
- LearnOpenGL. Coordinate-Systems.
<https://learnopengl.com/Getting-started/Coordinate-Systems>.
Acesso em Abril/2020.
- Computação Gráfica: Aula 08. Slides de Ricardo M. Marcacini. Disciplina SCC0250/0650, ICMC/USP, 2021.

EXERCÍCIO PARA CÔMPUTO DE PRESENÇA

Modifique o exercício da aula anterior para considerar a matriz *Projection*.
Demonstre o resultado da operação sobre os vértices já alterados pelas matrizes *Model* e *View*.