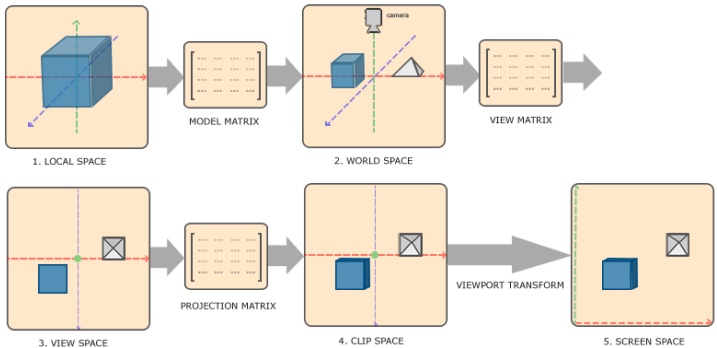
Computação Gráfica

Pipeline de Visualização

Prof. Alaor Cervati Neto

2023/1

Espaços de Coordenadas



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

Matriz *View*

Matriz *View*

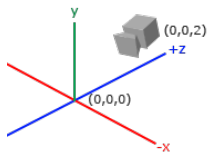
Espaço de Mundo→Espaço de Visão

- Transferência de objetos do cenário (mundo) para sistema de coordenada da visão.
- Dada a posição do observador/câmera:
  - Transladar o observador para a origem do sistema de coordenadas do mundo.
  - Rotacionar os eixos $x_{\text{view}}, y_{\text{view}}, z_{\text{view}}$ do observador para alinhar com os eixos do mundo $x_{\text{world}}, y_{\text{world}}, z_{\text{world}}$.
- A Matriz *View* é composta por Rotação e Translação.

Matriz *View*

Determinando Ponto de Visão e *Target*:

- É preciso definir as coordenadas da câmera em relação ao Espaço Mundo.
- É o ponto de visão (olho) do observador.



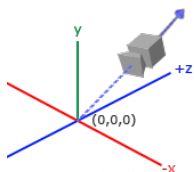
$$P_0 = (x_0, y_0, z_0)$$

Neste exemplo, o ponto de visão é a coordenada $(0, 0, 2)$ e o *Target* é a coordenada $(0, 0, 0)$.

Matriz *View*

Determinando o vetor normal *N*:

- A partir do ponto de visão e *target*, conseguimos uma "direção" da câmera.
- Obtém o eixo *z* da câmera (z_{view}).



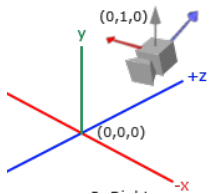
$$n = \frac{N}{|N|} = (n_x, n_y, n_z)$$

Uma forma de calcular o vetor normal *N* é subtrair o ponto de visão com o *target* e depois normalizar.

Matriz *View*

Determinando o vetor *view-up* V:

- O vetor V usualmente é definido como $(0, 1, 0)$.
- Usaremos para obter o eixo y da câmera (y_{view}).



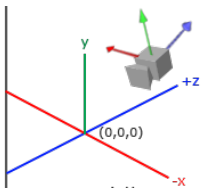
$$u = \frac{V \times n}{|V \times n|} = (u_x, u_y, u_z)$$

O vetor normalizado u é utilizado para obter o x_{view}. A partir de u e n, obtemos o y_{view}.

Matriz *View*

Determinando o vetor *view-up V*:

- O vetor *V* usualmente é definido como $(0, 1, 0)$.
- Usaremos para obter o eixo *y* da câmera (y_{view}).

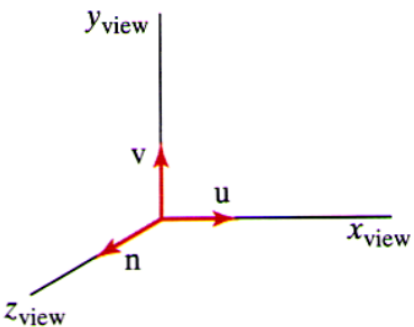


$$v = n \times u = (v_x, v_y, v_z)$$

O vetor normalizado *u* é utilizado para obter o x_{view}. A partir de *u* e *n*, obtemos o y_{view}.

Matriz *View*

Coordenadas da Visão



$$n = \frac{N}{|N|} = (n_x, n_y, n_z)$$

$$u = \frac{V \times n}{|V \times n|} = (u_x, u_y, u_z)$$

$$v = n \times u = (v_x, v_y, v_z)$$

Matriz *View*

Gerada por Translação e Rotação

Se a origem do sistema de visão for $P_0 = (x_0, y_0, z_0)$, a matriz de translação será:

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

Matriz *View*

Gerada por Translação e Rotação

A matriz de rotação pode ser obtida dos vetores $u = (u_x, u_y, u_z)$, $v = (v_x, v_y, v_z)$, e $n = (n_x, n_y, n_z)$:

$$R = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matriz *View*

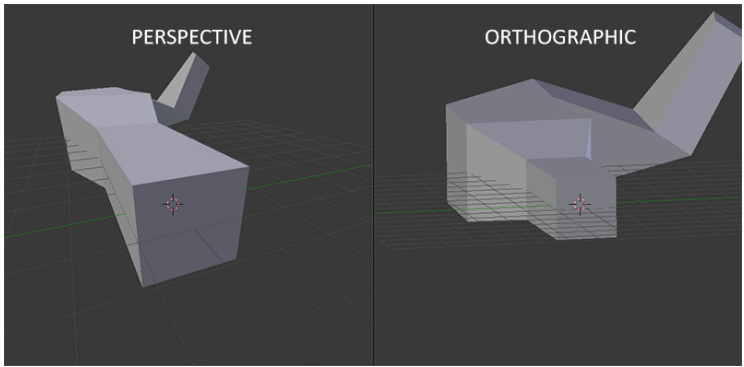
Gerada por Translação e Rotação

$$R \cdot T = \begin{bmatrix} u_x & u_y & u_z & -u \cdot P_0 \\ v_x & v_y & v_z & -v \cdot P_0 \\ n_x & n_y & n_z & -n \cdot P_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

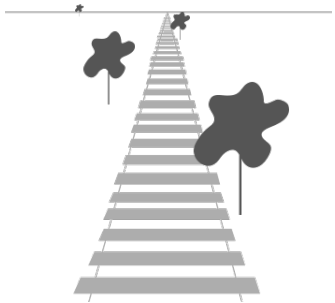
Matriz *Projection*

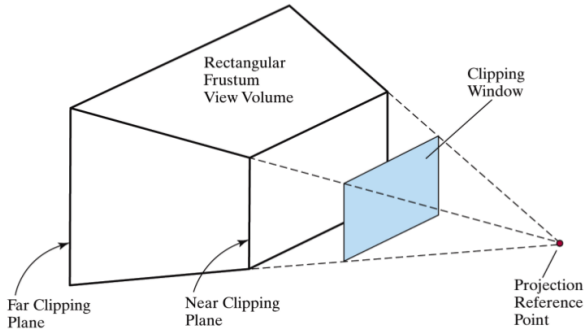
Matriz *Projection*

Dois tipos de projeção:



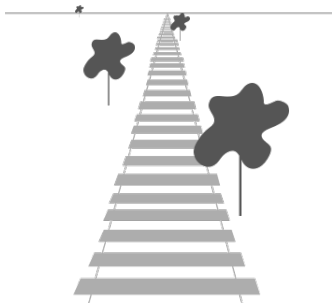
Projeção Perspectiva

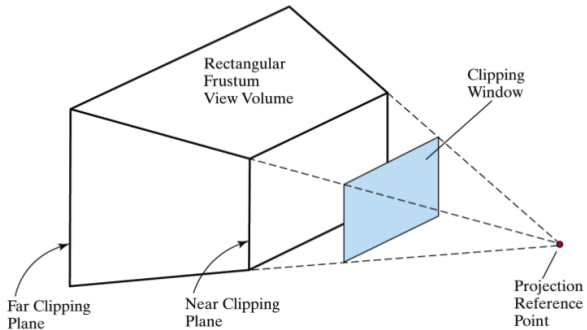




$\left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w}\right)$

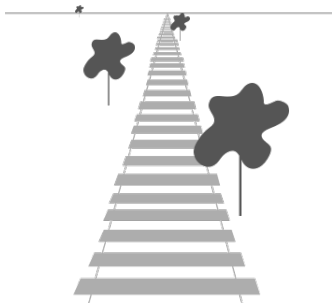
Projeção Perspectiva

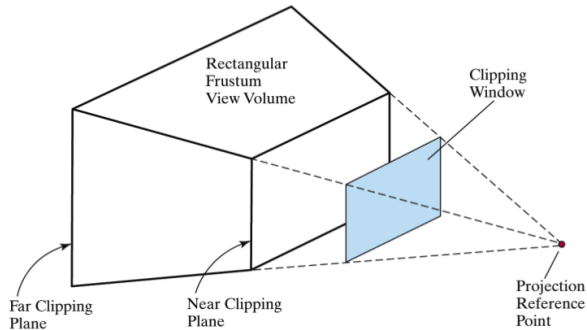




Janela de Recorte: cenas 3D projetadas/visíveis no plano de projeção.

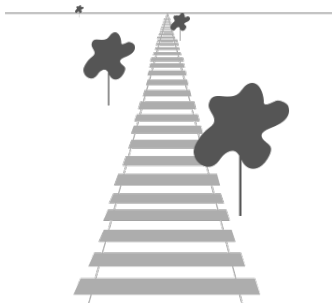
Projeção Perspectiva

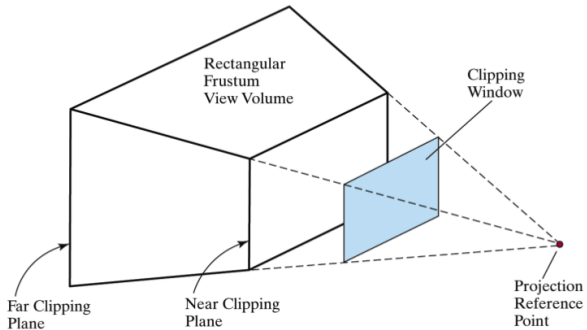




θ: ângulo do campo de visão.

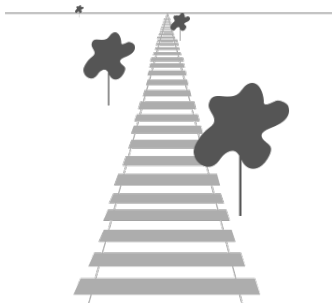
Projeção Perspectiva

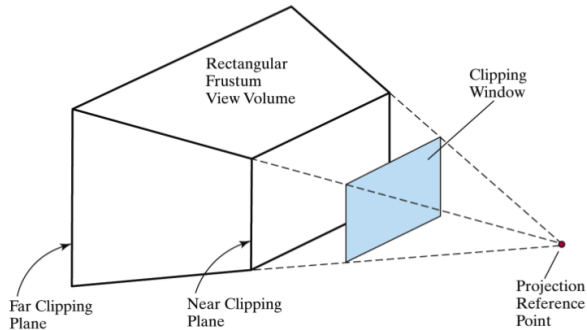




Planos de Recorte (*Near*/*Far*): perpendiculares ao eixo z_{view}.

Projeção Perspectiva





Frustum: porção visível do mundo.

Matriz de Projeção Perspectiva

$$M_{\text{normpers}} = M_{xy\text{scale}} \cdot M_{\text{pers}}$$

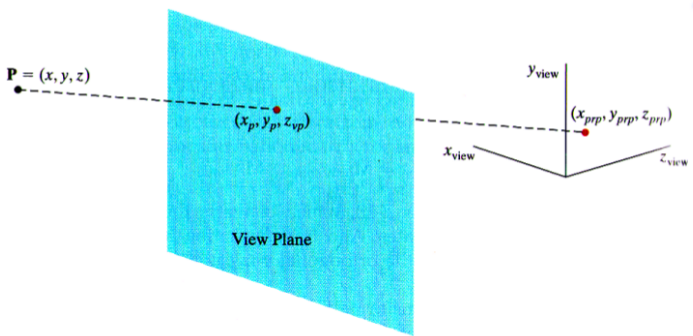
$$M_{xy\text{scale}} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, M_{\text{pers}} = \begin{bmatrix} \frac{width \cdot \cot\left(\frac{\theta}{2}\right)}{2 \cdot aspect} & 0 & -x_{prp} & x_{prp} z_{vp} \\ 0 & \frac{width \cdot \cot\left(\frac{\theta}{2}\right)}{2 \cdot aspect} & -y_{prp} & y_{prp} z_{vp} \\ 0 & 0 & s_z & t_z \\ 0 & 0 & -1 & z_{prp} \end{bmatrix}$$

Matriz de Projeção Perspectiva

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

Matriz de Projeção Perspectiva

Primeiramente, definimos um cenário simplificado:



Matriz de Projeção Perspectiva

Escrevemos x_p e y_p em função dos outros pontos:

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

Matriz de Projeção Perspectiva

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:

$$x_h = x\left(z_{prp} - z_{vp}\right) + x_{prp}\left(z_{prp} - z_{vp}\right)$$
$$y_h = y\left(z_{prp} - z_{vp}\right) + y_{prp}\left(z_{prp} - z_{vp}\right)$$

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:

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

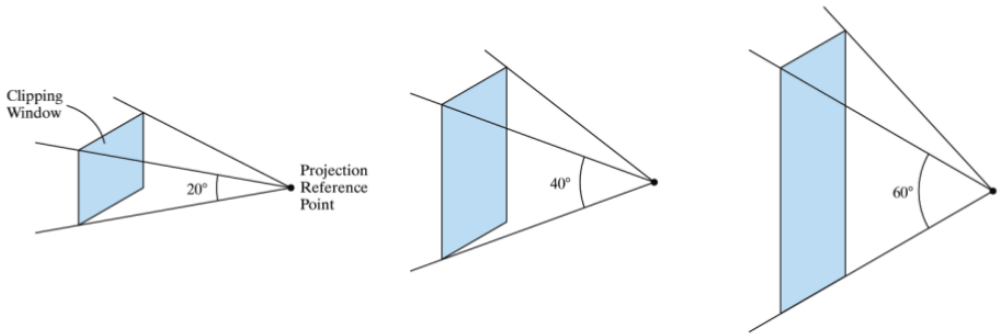
Matriz de Projeção Perspectiva

M_{pers} pode ser obtida de forma direta:

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

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:

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

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

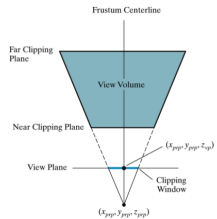
- Move o ponto de projeção para próximo do plano de visão.
- *Zoom-out* da cena.

$$M_{xyscale} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matriz de Projeção Perspectiva

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

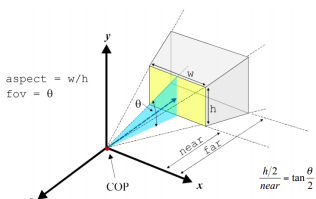
- A linha que liga o centro de projeção ao meio do plano de visão, perpendicular a esse, é a linha central do *frustum* simétrico.



Matriz de Projeção Perspectiva

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

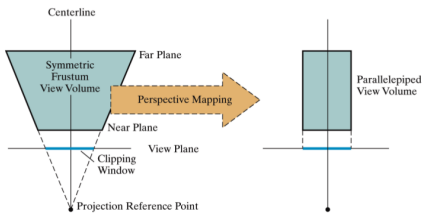
- A direção de observação é paralela a $-z$.
- O centro de projeção está na origem: $(x_{prp}, y_{prp}, z_{prp}) = (0, 0, 0)$.
- O plano de visão está sobre o plano de recorte: $z_{vp} = z_{\text{near}}$.



Matriz de Projeção Perspectiva

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

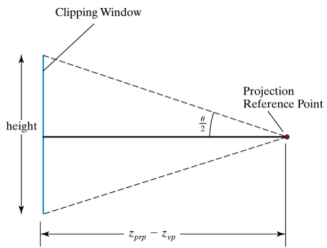
- Simplifica localizações dentro do *frustum* para um paralelepípedo retangular.



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.



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

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:

$$M_{\text{pers}} = \begin{bmatrix} \frac{width \cdot \cot\left(\frac{\theta}{2}\right)}{2 \cdot aspect} & 0 & -x_{prp} & x_{prp} z_{vp} \\ 0 & \frac{width \cdot \cot\left(\frac{\theta}{2}\right)}{2 \cdot aspect} & -y_{prp} & y_{prp} z_{vp} \\ 0 & 0 & s_z & t_z \\ 0 & 0 & -1 & z_{prp} \end{bmatrix}$$

Matriz de Projeção Perspectiva

Agora, calculando de fato a matriz normalizada:

$$M_{\text{normpers}} = M_{xy\text{scale}} \cdot M_{\text{pers}}$$

$$M_{\text{normpers}} = \begin{bmatrix} (z_{prp} - z_{vp})\, s_x & 0 & (-x_{prp})\, s_x & (x_{prp} z_{vp})\, s_x \\ 0 & (z_{prp} - z_{vp})\, s_y & (-y_{prp})\, s_y & (y_{prp} z_{vp})\, s_y \\ 0 & 0 & s_z & t_z \\ 0 & 0 & -1 & z_{prp} \end{bmatrix}$$

Matriz de Projeção Perspectiva

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:

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} -z_{\text{near}} s_x & 0 & 0 & 0 \\ 0 & -z_{\text{near}} s_y & 0 & 0 \\ 0 & 0 & s_z & t_z \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

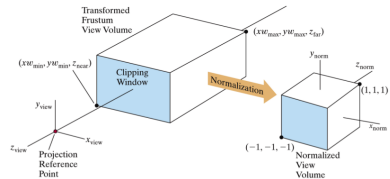
Matriz de Projeção Perspectiva

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 $(x, y, z) = (xw_{\text{min}}, yw_{\text{min}}, z_{\text{near}})$ em $(x_p, y_p, z_p) = (-1, -1, -1)$ e $(x, y, z) = (xw_{\text{max}}, yw_{\text{max}}, z_{\text{far}})$ em $(x_p, y_p, z_p) = (1, 1, 1)$.

Matriz de Projeção Perspectiva

Portanto, temos:

$$xw_{\text{min}} = -\frac{width}{2}, xw_{\text{max}} = \frac{width}{2}$$
$$yw_{\text{min}} = -\frac{height}{2}, yw_{\text{max}} = \frac{height}{2}$$

Matriz de Projeção Perspectiva

Substituímos 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 anteriores[1], encontramos:

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

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

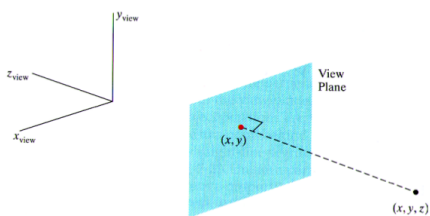
Matriz de Projeção Perspectiva

Substituindo s_x, s_y, s_z, e t_z, encontramos a matriz de Projeção Perspectiva normalizada:

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

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}.



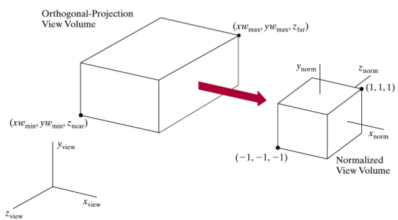
$$x_p = x$$

$$y_p = y$$

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).

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

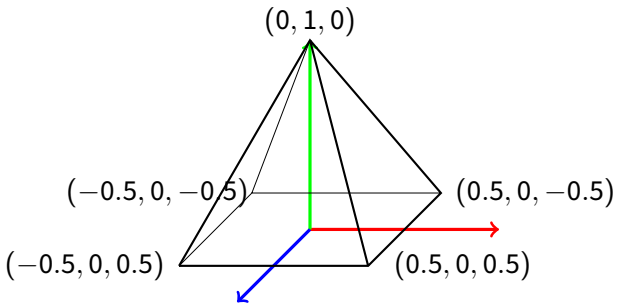
Material de base para a aula

- 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: Aulas 07 e 08. Slides de Ricardo M. Marcacini. Disciplina SCC0250/0650, ICMC/USP, 2021.

Exercícios

Exercícios I

Para a resolução, use o dia de seu nascimento como D e o mês como M. Considere a pirâmide, em seu espaço de coordenadas local:



Exercícios II

1. Multiplique o tamanho dos lados da base da pirâmide por M e sua altura por D, e posicione-a totalmente no 1º octante, isto é, com $\forall x, y, z \geq 0$. Mostre a matriz *Model* que realiza estas transformações e use-a para calcular as coordenadas da pirâmide no espaço de Mundo.
2. Tome o Ponto de Visão $P_0 = (0, 0, M)$, o *Target* na origem, e o vetor *View-Up* $V = (0, 1, 0)$. Exiba a matriz *View* obtida com estes parâmetros e utilize-a para converter as coordenadas do exercício anterior para o espaço de Visão.

Exercícios III

3. Considere o plano de projeção próximo posicionado em $z_{\text{near}} = \frac{D}{100}$ e o distante em $z_{\text{far}} = 10D$. Apresente uma matriz de Projeção Perspectiva Normalizada (*Projection*), construída usando estes parâmetros, com o ângulo e aspecto da projeção a sua escolha. Transforme as coordenadas obtidas no exercício anterior para o espaço de *Clip* usando esta matriz.