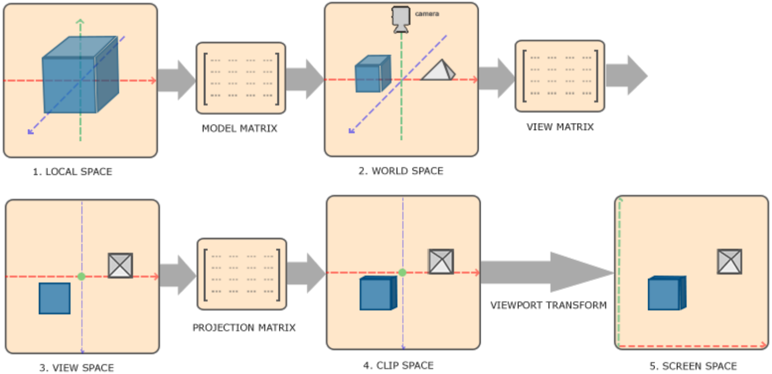
Computação Gráfica

Pipeline de Visualização

Prof. Alaor Cervati Neto

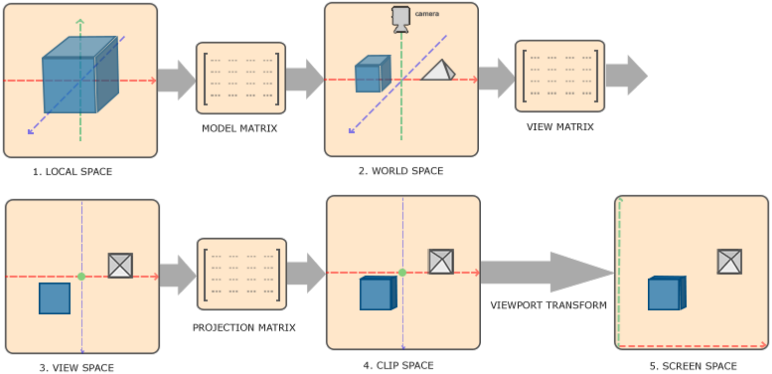
2022/1

Espaço de Coordenadas



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

Espaço de Coordenadas



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

Matriz *View*

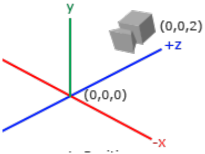
Espaço de Mundo→Espaço de Visão

- 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.



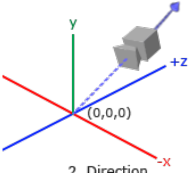
$$P_0 = (x_0, y_0, z_0)$$

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}).



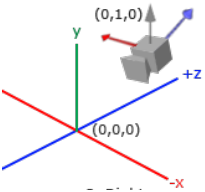
$$n = \frac{N}{|N|} = (n_x, n_y, n_z)$$

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}).



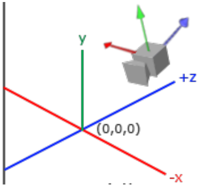
$$u = \frac{V \times n}{|V \times n|} = (u_x, u_y, u_z)$$

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}).

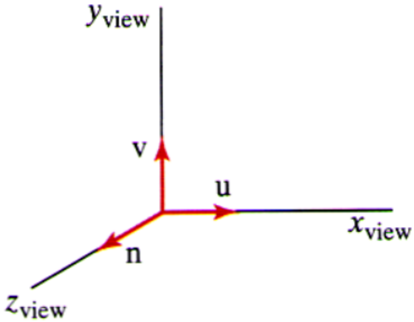


$$v = n \times u = (v_x, v_y, v_z)$$

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*

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

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: Aula 07. Slides de Ricardo M. Marcacini. Disciplina SCC0250/0650, ICMC/USP, 2021.

Exercício complementar

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