

Noções Básicas

sexta-feira, 19 de fevereiro de 2021

Tudo é definido por triângulos

- É a figura mais simples que existe
- É garantidamente convexo - mais fácil de pintar
- É garantidamente plano - não há ambiguidades

Pintar a cena

A iluminação é o que nos dá a percepção de 3D

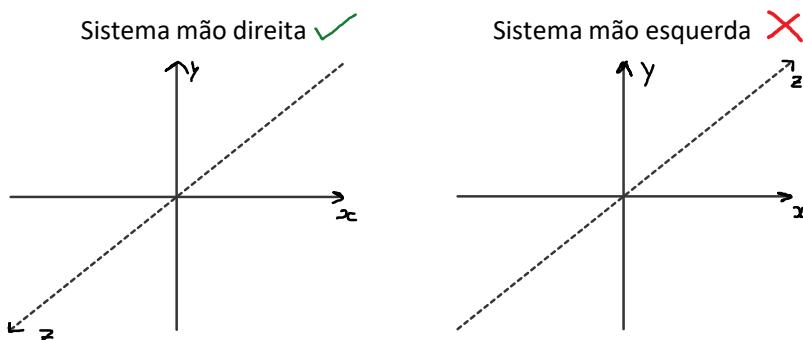
Técnicas de otimização

- Eliminação de geometria não visível
- Partição especial
- Transferências de memória

Vetores

$$p'' = \alpha_1 p + \alpha_2 p'$$

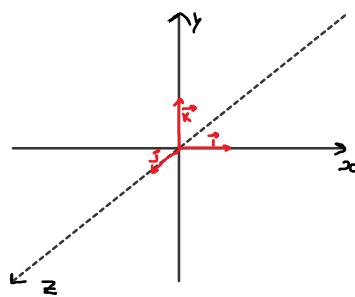
Interpolação linear



★ | Y é o eixo vertical

Versores

- Direção do eixo
- Definem unidade



Definição de pontos

Os pontos em computação gráfica serão sempre definidos da seguinte forma:

$$P = (P_x, P_y, 0)$$

Ou seja:

$$P = (P_x, P_y, 1)$$

Por sua vez, os vetores, serão definidos da seguinte forma:

$$v = (v_x, v_y, 0)$$

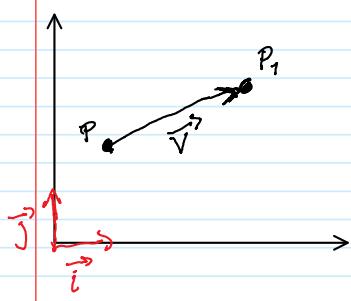
Eventos

segunda-feira, 22 de fevereiro de 2021

- Premir uma tecla
- Premir um botão do rato
- Movimento do rato
- Mudar o tamanho da janela
- Janela precisa de ser pintada
- ...

Translações e Rotações

sexta-feira, 26 de fevereiro de 2021



Deduções das matrizes no caderno

$$\begin{bmatrix} 1 & 0 & V_x \\ 0 & 1 & V_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ 1 \end{bmatrix} = \begin{bmatrix} P_x + V_x \\ P_y + V_y \\ 1 \end{bmatrix}$$

$$P_1 = T_V \cdot P$$

⚠️ Primeiro a translação, depois o desenho

Translação

$$T = \begin{bmatrix} 1 & 0 & 0 & V_x \\ 0 & 1 & 0 & V_y \\ 0 & 0 & 1 & V_z \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Matriz genérica

Rotação

$$R = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$$

Matrizes inversas

sexta-feira, 12 de março de 2021

$$MG = I$$

$$\left[\begin{array}{c|c} RS & T \\ \hline 0 & 1 \end{array} \right] \cdot \left[\begin{array}{c|c} A & B \\ \hline 0 & 1 \end{array} \right] = \left[\begin{array}{c|c} RS \cdot A & RS \cdot B + T \\ \hline 0 & 1 \end{array} \right]$$

M G

$$RS \cdot A = I_3$$

$$RS \cdot B + T = 0$$

$$A = RS^{-1}$$

$$B = -RS^{-1} \cdot T$$

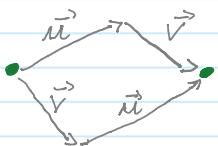
Se a matriz for orthonormal

isto é: as linhas ou colunas são perpendiculares entre si
Têm comprimento 1

$$RS^{-1} = RS^T$$

Translação

$$T_1 \times T_2 = T_2 \times T_1 \quad \checkmark$$



$$\left[\begin{array}{c|c} I & t_1 \\ \hline 0 & 1 \end{array} \right] \left[\begin{array}{c|c} I & t_2 \\ \hline 0 & 1 \end{array} \right] = \left[\begin{array}{c|c} I & t_1 + t_2 \\ \hline 0 & 1 \end{array} \right] \quad \checkmark$$

$$\left[\begin{array}{c|c} I & t_2 \\ \hline 0 & 1 \end{array} \right] \left[\begin{array}{c|c} I & t_1 \\ \hline 0 & 1 \end{array} \right] = \left[\begin{array}{c|c} I & t_2 + t_1 \\ \hline 0 & 1 \end{array} \right] \quad \checkmark$$

Escala

$$S_1 \times S_2 = S_2 \times S_1 \quad \checkmark$$

$$\left[\begin{array}{ccc} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{array} \right] \left[\begin{array}{ccc} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & 1 \end{array} \right] = \left[\begin{array}{ccc} S_x \times K_x & 0 & 0 \\ 0 & S_y \times K_y & 0 \\ 0 & 0 & 1 \end{array} \right] \quad \checkmark$$

$$\left[\begin{array}{ccc} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & 1 \end{array} \right] \left[\begin{array}{ccc} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{array} \right] = \left[\begin{array}{ccc} K_x \times S_x & 0 & 0 \\ 0 & K_y \times S_y & 0 \\ 0 & 0 & 1 \end{array} \right] \quad \checkmark$$

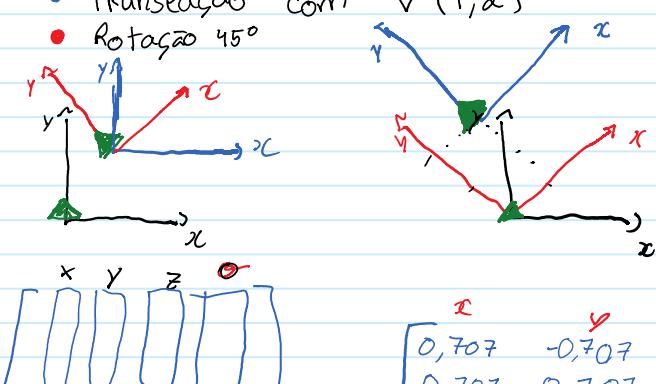
Rotação

$$R_1 \times R_2 = R_2 \times R_1$$

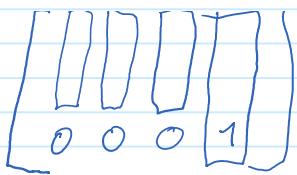
Se o eixo de rotação for o mesmo! \checkmark

Se o eixo não for o mesmo (3D) \times

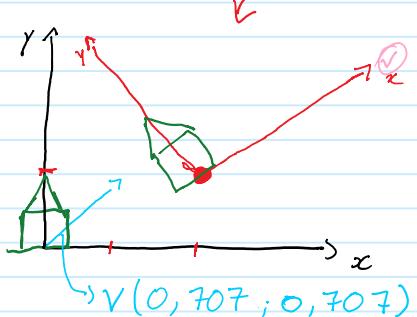
- Translações com $\vec{v}(1, 2)$
- Rotação 45°



$$TR \neq RT$$



$$\begin{bmatrix} x & y & \textcircled{z} \\ 0,707 & -0,707 & 2 \\ 0,707 & 0,707 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$



Escala não homogénea
° Rotação

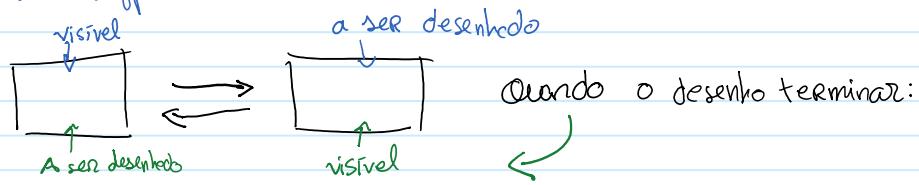
Mas apenas depois
de ~~todas as operações~~

Double Buffering

sexta-feira, 26 de março de 2021 14:06

Problema: Sincronizar com o ecrã a meio do desenho

Solução: dois buffers



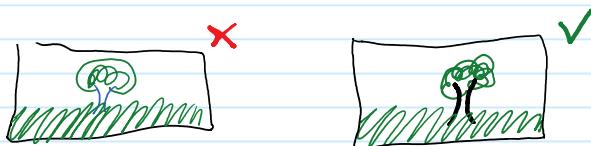
Isto é traduzido pelo comando "GLUT_DOUBLE"

⚠ Isto não significa que possa ter Single buffering. O comando é um pedido, a placa pode recusar.

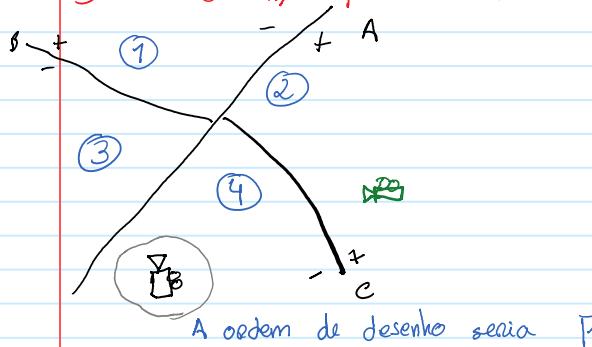
Depth

sexta-feira, 26 de março de 2021 14:21

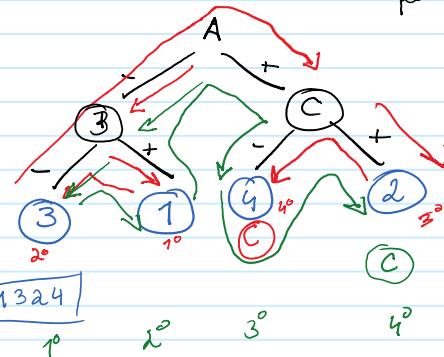
Problema: é necessário saber a ordem relativa dos objetos em relação à câmera



BSP (Binary Space Partition)



Este sistema já não é usado desta forma, mas sim para não pintar pixels a mais.

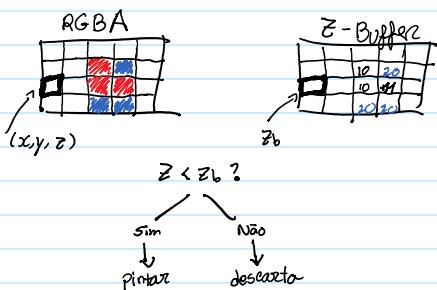


- Divisão recursiva do "mundo" com planos
- BSP permite determinar a ordem relativa do desenho

Depth Buffering ou Z-Buffer

- Buffer de cor
- Buffer de profundidade

Nota: Isto acontece no nível do Hardware



$$P' = P \cdot p$$

$$\begin{bmatrix} - & - & - \\ - & - & - \\ 0 & 0 & \frac{-\text{far} + \text{near}}{\text{far} - \text{near}} \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} - \\ - \\ \text{Ze} \\ 1 \end{bmatrix}$$

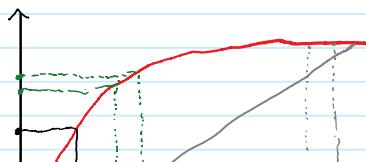
$$P'(X_w, Y_w, Z_w, 1]$$

$$Z_w = \frac{\text{far} + \text{near}}{\text{far} - \text{near}} + \frac{2 \cdot \text{near} \cdot \text{far}}{\text{far} - \text{near}} \times \frac{1}{\text{Ze}}$$

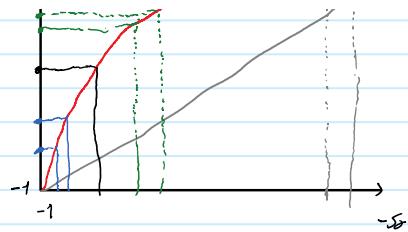
→ Relação Não Linear

$$\text{near} = 1$$

$$\text{far} = 50$$



Os valores de profundidade são guardados como inteiros



Os valores de profundidade são considerados inteiros