Lista de Exercícios #2

Laís Saloum Deghaide, nUSP: 11369767

Dia de Aniversário = Variável D = [21]
Mês de Aniversário = Variável M = [3]

1) Inicialmente, todos os objetos que compõem uma cena são posicionados na origem, chamada de "espaço local". A matriz Model de cada objeto tem a função de aplicar as transformações de rotação, translação e escala em seu respectivo objeto, os levando ao espaço de Mundo. Feito isso, a matriz View posiciona um "observador" e define o ponto observado no espaço mundo, o transformando no espaço de visão. Então, a matriz Projection define o tamanho do espaço observável, distância máxima visível, e ângulo do campo de visão, "cortando" partes do espaço view que não devem ser vistas, o transformando no espaço de clip, resultado final desse pipeline.

2) Como temos apenas a translação de -3 no eixo z, essa matriz pode ser definida como uma matriz de translação em coordenadas uniformes:

$$Mmodel = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Vamos encontrar os novos vértices da pirâmide multiplicando a matriz Model pela matriz com as coordenadas da pirâmide:

$$Mpiramide = \begin{bmatrix} 0 & -0.5 & -0.5 & 0.5 & 0.5 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0.5 & -0.5 & 0.5 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$Mmodel * Mpiramide = \begin{bmatrix} 0 & -0.5 & -0.5 & 0.5 & 0.5 \\ 1 & 0 & 0 & 0 & 0 \\ -3 & \frac{-7}{2} & \frac{-5}{2} & \frac{-7}{2} & \frac{-5}{2} \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Então, os novos vértices da pirâmide são:
$(0, 1, -3)$, $(-0.5, 0, -3.5)$, $(-0.5, 0, -2.5)$, $(0.5, 0, -3.5)$, $(0.5, 0, -2.5)$

3) Seja $P_v = (0, 0, 1)$ o ponto de visão, $P_t = (0, 0, 0)$ o target, e $V_{up} = (0, 1, 0)$ o view up. Vamos determinar os vetores normais:

$N = P_v - P_t = (0, 0, 1)$ →vetor normal

$N_z = \frac{N}{||N||} = (0, 0, 1)$ →vetor normal no eixo z

$N_x = \frac{V_{up} \wedge N_z}{\left\|V_{up} \wedge N_z\right\|} = \frac{(1,0,0)}{1} = (1, 0, 0)$ →vetor normal no eixo x

$N_y = N_z \wedge N_x = (0, 1, 0)$ →vetor normal no eixo y

Usando N_x, N_y, N_z para rotacionar e $P_0 = (0, 0, 0)$ como a origem do sistema de coordenadas, temos a matriz View:

$$M_v = \begin{bmatrix} N_{xx} & N_{xy} & N_{xz} & -N_x * P_0 \\ N_{yx} & N_{yy} & N_{yz} & -N_y * P_0 \\ N_{zx} & N_{zy} & N_{zz} & -N_z * P_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_v = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Multiplicando pela matriz da pirâmide para encontrarmos os novos vértices:

$$Mv * Mpiramide = \begin{bmatrix} 0 & -0.5 & -0.5 & 0.5 & 0.5 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0.5 & -0.5 & 0.5 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Então, os novos vértices da pirâmide são:
$(0, 1, 0)$, $(-\ 0.5, 0, -\ 0.5)$, $(-\ 0.5, 0, 0.5)$, $(0.5, 0, -\ 0.5)$, $(0.5, 0, 0.5)$

4) Sendo o ângulo do campo de visão $\theta = \frac{\pi}{2} rad$, $z_{near} = 1$, $z_{far} = 10$e uma janela quadrada, $aspect = 1$, temos a matriz de Projeção Perspective:

$$M_p = \begin{bmatrix} \frac{cot(\frac{\theta}{2})}{aspect} & 0 & 0 & 0 \\ 0 & cot(\frac{\theta}{2}) & 0 & 0 \\ 0 & 0 & \frac{Znear+Zfar}{Znear-Zfar} & -\frac{2*Znear*Zfar}{Znear-Zfar} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$M_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1.1 & -2 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$Mp * Mpiramide = \begin{bmatrix} 0 & -0.5 & -0.5 & 0.5 & 0.5 \\ 1 & 0 & 0 & 0 & 0 \\ -2 & -1.45 & -2.55 & -1.45 & -2.55 \\ 0 & -0.5 & -0.5 & 0.5 & 0.5 \end{bmatrix}$$

Então, os novos vértices da pirâmide são:
$(0, 1, -\ 2)$, $(-\ 0.5, 0, -\ 1.45)$, $(-\ 0.5, 0, -\ 2.55)$, $(0.5, 0, -\ 1.45)$, $(0.5, 0, -\ 2.55)$

5) Definindo os pontos $W_{min} = (-1, -1, -1)$ e $W_{max} = (1, 1, 1)$, temos:

$$M_o = \begin{bmatrix} \frac{2}{XWmax - XWmin} & 0 & 0 & -\frac{XWmax+XWmin}{XWmax-XWmin} \\ 0 & \frac{2}{YWmax-YWmin} & 0 & -\frac{YWmax+YWmin}{YWmax-YWmin} \\ 0 & 0 & \frac{-2}{Znear-Zfar} & \frac{Znear+Zfar}{Znear-Zfar} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_o = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Mo * Mpiramide = \begin{bmatrix} 0 & -0.5 & -0.5 & 0.5 & 0.5 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0.5 & -0.5 & 0.5 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Então, os novos vértices da pirâmide são:
$(0, 1, 0)$, $(-0.5, 0, -0.5)$, $(-0.5, 0, 0.5)$, $(0.5, 0, -0.5)$, $(0.5, 0, 0.5)$

6) Os parâmetros Near e Far tem por objetivo definir a distância mínima e máxima do ponto de visão a serem exibidas. Tudo que estiver a uma distância menor que z_{min} ou maior que z_{max} do observador não é mostrado na tela.

7) Na computação gráfica 3D, o Frustum é a região do espaço no mundo modelado que pode aparecer na tela, é o campo de visão de um sistema de câmera virtual em perspectiva. Nada fora do Frustum aparece na tela.

8) Textura é uma técnica que, quando aplicada ao modelo geométrico junto ao modelo de iluminação, procura dar mais realismo. Uma vez mapeadas a um polígono, as texturas estão sujeitas a todas as transformações que ocorrem naquele polígono. Elas irão rotacionar, mover ou escalar juntamente com o polígono. Podemos ver a textura como se fosse a "pele" de nossa geometria.
O mapeamento consiste em utilizar uma imagem e mapeá-la sobre a superfície. A imagem é denominada mapa de textura. O padrão pode ser repetido. O mapeamento ocorre como uma mudança de coordenadas. Podemos resumir o mapeamento em 4 passos:

1. Projeção do pixel sobre a superfície
2. Parametrização
3. Função de mapeamento + Função inversa
   (Dado um fragmento, queremos saber a que ponto do objeto corresponde Dado um ponto no objeto, queremos saber a que ponto na textura ele corresponde)
4. Cor média dos texels, proporcional a área coberta pelo quadrilátero

Nesse mapeamento, são marcados pontos na imagem de textura, os quais formam faces que correspondem às faces presentes no objeto renderizado.
Alguns tipos de mapeamento:

Esférico: coordenadas uvsão mapeadas segundo coordenadas polares esféricas
Planar: coordenadas uvmapeadas ortogonalmente.
Cilíndrico: coordenadas uvsão mapeadas segundo coordenadas polares cilíndricas.

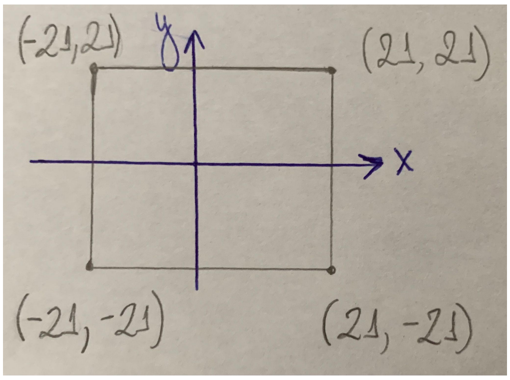
9) Os texels são fragmentos de textura, e os pixels são fragmentos do objeto renderizado. Um texel pode ocupar vários pixels ou um pixel pode conter vários texels.

10) Os parâmetros são aplicados quando há repetição da textura. O REPEAT repete a textura em várias instâncias, lado a lado, enquanto o CLAMP "alonga" o último pixel da textura.

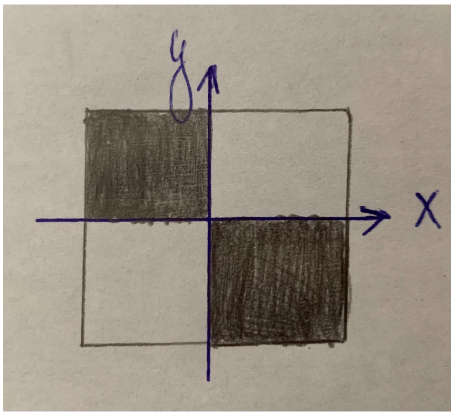
11) Esses filtros devem ser aplicados em caso de magnificação (polígono maior que a textura) ou minificação (textura maior que o polígono). O filtro NEAREST define a cor do pixel com o texel mais próximo da coordenada de textura, enquanto o LINEAR define essa cor como a cor média dos texels próximos.

12)

a)



b)



c)