Introdução à Computação Gráfica Implementando Pipeline Gráfico

Aluno: Renan Ribeiro Lage

Matrícula: 201600728

Professor: Christian Pagot

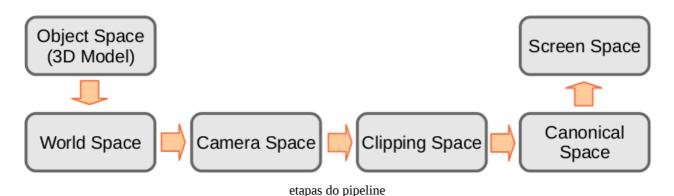
1. Introdução

Neste segundo trabalho para a disciplina de Introdução à Computação Gráfica, ministrada pelo professor Christian Pagot.

Neste trabalho foi solicitado a implementação de um pipeline gráfico completo, parecido ao utilizado pelo OpenGL, para essa implementação utilizaríamos os conhecimentos adquiridos em sala e a rasterização seria realizada pelo código que desenvolvemos na atividade TI1.

O que é um pipeline gráfico?

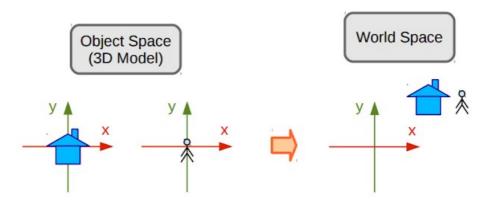
É uma sequência de passos (uma receita de bolo) que deve ser seguida no intuito de transformar uma cena 3D em uma cena 2D. Esses passos são descritos pela figura abaixo:



Será descrito a seguir como funciona cada um desses passos e o processo relacionado.

A) Espaço do Objeto para Espaço do Universo

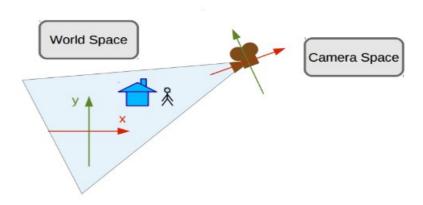
O espaço do objeto é um espaço tridimensional onde, dado objeto está inserido no seu próprio sistema de coordenadas centrado na origem no sistema de coordenadas. Ao aplicar nos vértices transformações de rotação, translação, escala e shear, passamos do espaço do objeto para o espaço do universo.



Na realização desses passos são utilizadas matrizes, que multiplicadas uma pela outra geral a chamada matriz de modelagem. Caso o objeto não sofra alterações a matriz de modelagem é a identidade. A exceção é a translação que por ser uma transformação afim não pode ser representada por matrizes.

B) Espaço do Universo para Espaço da Câmera

A passagem do espaço do universo se dá pela utilização de outra matriz denominada de Visualização (View), cria-se um sistema de coordenadas da câmera que é composto por uma translação e uma rotação. A translação é utilizada para igualar as origens e a rotação para igualar as bases do espaço do universo com o espaço da câmera, na implementação dessa matriz se necessita de algumas informações como a posição da câmera (onde ela se encontra) e o vetor direção.



C) Espaço da Câmera para Espaço de Recorte

Para a passagem para o espaço de recorte utiliza-se outra matriz chamada de Matriz de Projeção. Essa matriz cria a perspectiva de realidade fazendo com que objetos que estejam mais próximos da câmera pareçam maiores e objetos mais distantes pareçam menores.

$$M_{pt} = M_{p} M_{t} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & -\frac{1}{d} & 0 \end{bmatrix}$$

Exemplo de Matriz de Projeção

D) Espaço de Recorte para Espaço Canônico

Nesta etapa as imagens começam a ficar parecidos com o que estamos acostumados a ver no dia a dia. Transforma os vértices do espaço de recorte para um espaço canônico. Dividimos todas as coordenadas dos vértices pelas coordenadas homogênea de cada um.

E) Espaço Canônico para Espaço de Tela

No espaço canônico garante-se que todos os vértices da cena visível possui os valores de suas coordenadas entre -1 e 1. Este espaço é obtido quando, após multiplicar os vértices pelas matrizes model, view e projection, dividi-se as coordenadas dos vétices por sua coordenada w (coordenada homogênea).

Após o espaço canônico é preciso preparar os vértices para serem rasteirados na tela. Este processo e feito multiplicando os vértice por uma matriz chamada viewport. Essa matriz leva os vértices do espaço canônico para o espaço da tela e é formada pela multiplicação das matrizes mostradas na figura abaixo:

$$\begin{bmatrix} 1 & 0 & 0 & \frac{w-1}{2} \\ 0 & 1 & 0 & \frac{h-1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{w}{2} & 0 & 0 & 0 \\ 0 & \frac{h}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
Transtale. Scale along X and Y acis. Invert Y axis direction

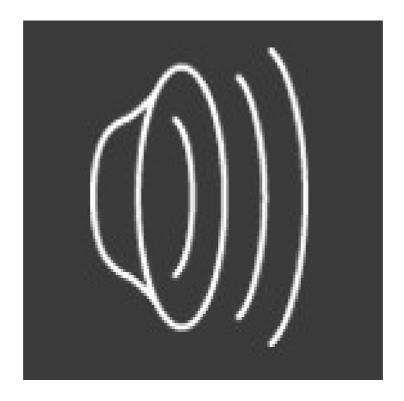
Na figura acima, w representa o numero de colunas de pixels da tela e h o número de linhas de pixel da tela. Estas matrizes consistem em duas escalas e uma translação. Uma das escalas é para inverter as coordenadas y dos vértices, pois na tela o eixo y cresce para baixo. A outra escala é pra fazer com que todo o espaço entre 1 e -1 fique nas mesmas dimensões da tela. E a translação é para levar os objetos para o centro da tela.

2. Vídeo

Pipeline Feito:



Pipeline do OpenGl:



Como pode se notar no vídeo o resultado e a comparação entre os dois videos percebe-se um resultado bem semelhante.

3. Dificuldades Encontradas

Como foi rodado no Sistema Operacional Windowns apresentou certa lentidão na execução.

4. Referencias

Notas de aulas cedidas pelo Professor Christian Pagot.