

# Computação Gráfica

## Pipeline de Visualização

Prof. Alaor Cervati Neto



2022/1

# Transformação Geométrica

Até o momento, aprendemos:

- ▶ Primitivas para desenhar objetos.
- ▶ Transformações geométricas (2D e 3D):
  - ▶ Translação.
  - ▶ Escala.
  - ▶ Rotação.

Tudo que fizemos está representado em um Espaço de Coordenadas Local.

# Espaço de Coordenadas

Nesta aula introduziremos outros espaços de coordenadas:

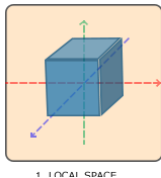
**Espaço Local** Coordenadas iniciais dos vértices (objetos).

**Espaço Mundo** Transformações nos vértices de forma a posicioná-los em relação ao nosso mundo 3D (escala, rotação, translação).

**Espaço Visão** Transformação nos vértices (mundo) para que os objetos sejam visualizados a partir de um ponto de referência (observador ou câmera).

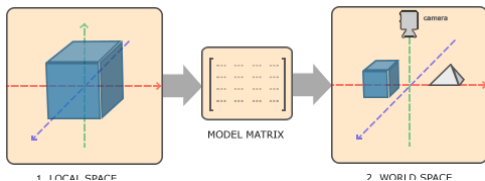
**Espaço Clip** Transformação nos vértices (visão) para determinar o que de fato será apresentado na tela.

# Espaço de Coordenadas



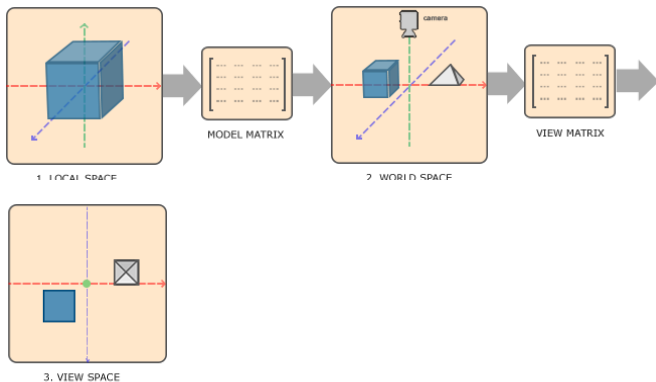
Coordenadas iniciais dos vértices (objetos).

# Espaço de Coordenadas



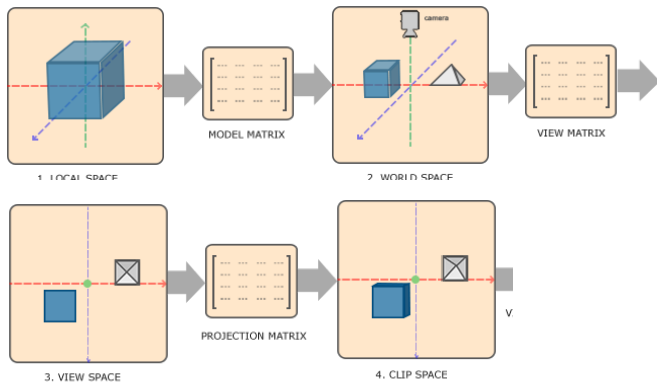
Transformações nos vértices de forma a posicioná-los em relação ao nosso mundo 3D (escala, rotação, translação).

# Espaço de Coordenadas



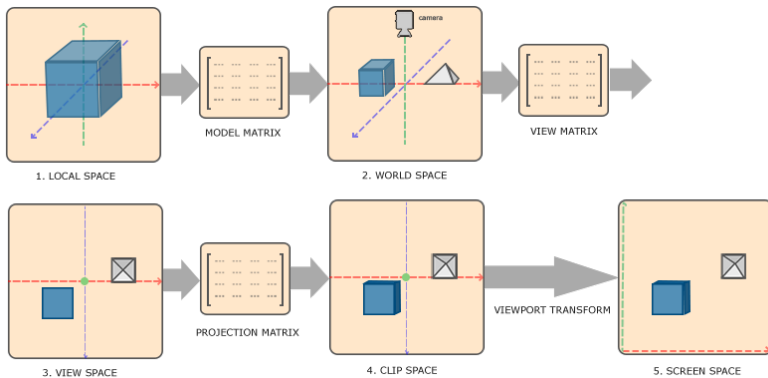
Transformação nos vértices (mundo) para que os objetos sejam visualizados a partir de um ponto de referência (observador ou câmera).

# Espaço de Coordenadas



Transformação nos vértices (visão) para determinar o que será apresentado na tela.

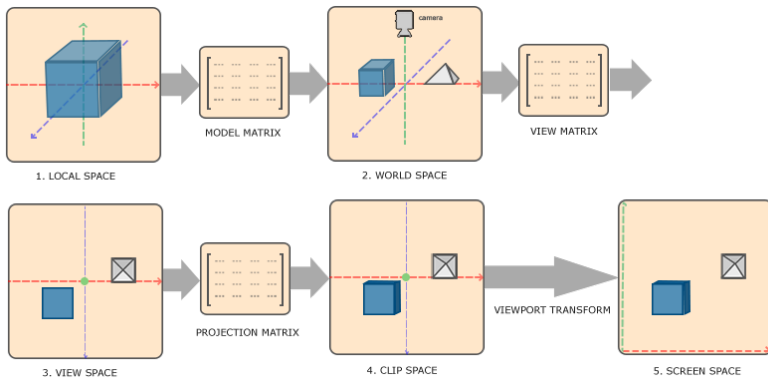
# Espaço de Coordenadas



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



# Espaço de Coordenadas

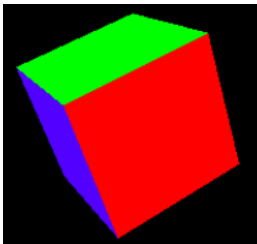


$$P' = \text{Projection} \times \text{View} \times \boxed{\text{Model}} \times P$$

# Matriz *Model*

Espaço Local → Espaço do Mundo

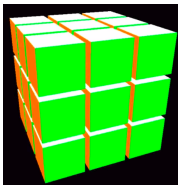
- ▶ É comum que cada objeto seja inicializado com seu centro na origem do sistema de coordenadas local.
- ▶ Cada objeto tem sua própria matriz *Model* para posicioná-lo no mundo.



# Matriz *Model*

Espaço Local → Espaço do Mundo

- ▶ É comum que cada objeto seja inicializado com seu centro na origem do sistema de coordenadas local.
- ▶ Cada objeto tem sua própria matriz *Model* para posicioná-lo no mundo.
- ▶ Permite a construção de cenários e compor objetos complexos a partir de objetos simples.



# Matriz *Model*

A matriz *Model* é composta por transformações geométricas 3D:

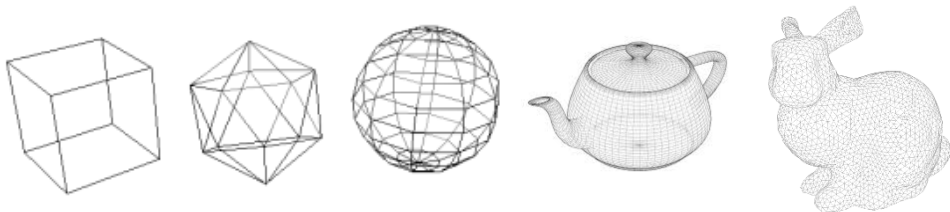
- ▶ Translação.
- ▶ Rotação.
- ▶ Escala.

Já é conhecida.

# Malha Poligonal

Malha poligonal é uma coleção de faces, onde cada face é um conjunto de vértices:

- ▶ Formam triângulos (ou quadriláteros).
- ▶ Triângulos facilitam a renderização de objetos complexos.



# Malha Poligonal

- ▶ Até o momento os vértices dos objetos foram escritos no código.
- ▶ Não é muito prático.
- ▶ Carregaremos modelos a partir de arquivos.
- ▶ Facilita reutilizar diferentes modelos pré-existentes (de outros autores).
- ▶ Formato *WaveFront* (Extensão .obj).

# Formato *WaveFront*

- ▶ Objetos/Modelos no formato *WaveFront*.
- ▶ Extensão .obj.
- ▶ São arquivos de texto com conteúdo facilmente interpretado.
- ▶ Um dos formatos mais utilizados.

# Formato *WaveFront*

Cubo.obj

```
# Blender3D v249 OBJ File
v 1.000000 -1.000000 -1.000000
v 1.000000 -1.000000 1.000000
v -1.000000 -1.000000 1.000000
v -1.000000 -1.000000 -1.000000
v 1.000000 1.000000 -1.000000
v 0.999999 1.000000 1.000001
v -1.000000 1.000000 1.000000
v -1.000000 1.000000 -1.000000

f 5/1/1 1/2/1 4/3/1
f 5/1/1 4/3/1 8/4/1
f 3/5/2 7/6/2 8/7/2
f 3/5/2 8/7/2 4/8/2
f 2/9/3 6/10/3 3/5/3
f 6/10/4 7/6/4 3/5/4
f 1/2/5 5/1/5 2/9/5
f 5/1/6 6/10/6 2/9/6
f 5/1/7 8/11/7 6/10/7
f 8/11/7 7/12/7 6/10/7
f 1/2/8 2/9/8 3/13/8
f 1/2/8 3/13/8 4/14/8
```

O caractere inicial de cada linha indica a função:

- # Comentários
- v Vértices
- vn Normal
- vt Coordenadas de textura
- f Faces



# Formato *WaveFront*

Cubo.obj

```
# Blender3D v249 OBJ File
v 1.000000 -1.000000 -1.000000
v 1.000000 -1.000000 1.000000
v -1.000000 -1.000000 1.000000
v -1.000000 -1.000000 -1.000000
v 1.000000 1.000000 -1.000000
v 0.999999 1.000000 1.000001
v -1.000000 1.000000 1.000000
v -1.000000 1.000000 -1.000000

f 5/1/1 1/2/1 4/3/1
f 5/1/1 4/3/1 8/4/1
f 3/5/2 7/6/2 8/7/2
f 3/5/2 8/7/2 4/8/2
f 2/9/3 6/10/3 3/5/3
f 6/10/4 7/6/4 3/5/4
f 1/2/5 5/1/5 2/9/5
f 5/1/6 6/10/6 2/9/6
f 5/1/7 8/11/7 6/10/7
f 8/11/7 7/12/7 6/10/7
f 1/2/8 2/9/8 3/13/8
f 1/2/8 3/13/8 4/14/8
```

O arquivo deve ter uma lista de vértices,  
como visto anteriormente.

# Formato *WaveFront*

Cubo.obj

```
# Blender3D v249 OBJ File
v 1.000000 -1.000000 -1.000000
v 1.000000 -1.000000 1.000000
v -1.000000 -1.000000 1.000000
v -1.000000 -1.000000 -1.000000
v 1.000000 1.000000 -1.000000
v 0.999999 1.000000 1.000001
v -1.000000 1.000000 1.000000
v -1.000000 1.000000 -1.000000

f 5/1/1 1/2/1 4/3/1
f 5/1/1 4/3/1 8/4/1
f 3/5/2 7/6/2 8/7/2
f 3/5/2 8/7/2 4/8/2
f 2/9/3 6/10/3 3/5/3
f 6/10/4 7/6/4 3/5/4
f 1/2/5 5/1/5 2/9/5
f 5/1/6 6/10/6 2/9/6
f 5/1/7 8/11/7 6/10/7
f 8/11/7 7/12/7 6/10/7
f 1/2/8 2/9/8 3/13/8
f 1/2/8 3/13/8 4/14/8
```

- ▶ Cada face é um triângulo.
- ▶ O conjunto de triângulos dará forma ao modelo (malha).

# Formato *WaveFront*

Cubo.obj

```
# Blender3D v249 OBJ File
v 1.000000 -1.000000 -1.000000
v 1.000000 -1.000000 1.000000
v -1.000000 -1.000000 1.000000
v -1.000000 -1.000000 -1.000000
v 1.000000 1.000000 -1.000000
v 0.999999 1.000000 1.000001
v -1.000000 1.000000 1.000000
v -1.000000 1.000000 -1.000000

f 5/1/1 1/2/1 4/3/1
f 5/1/1 4/3/1 8/4/1
f 3/5/2 7/6/2 8/7/2
f 3/5/2 8/7/2 4/8/2
f 2/9/3 6/10/3 3/5/3
f 6/10/4 7/6/4 3/5/4
f 1/2/5 5/1/5 2/9/5
f 5/1/6 6/10/6 2/9/6
f 5/1/7 8/11/7 6/10/7
f 8/11/7 7/12/7 6/10/7
f 1/2/8 2/9/8 3/13/8
f 1/2/8 3/13/8 4/14/8
```

Cada triângulo é formado por três vértices, representados por  $v/vt/vn$ .

Exemplo: 5/1/1

- ▶ A primeira parte indica o vértice na posição cinco ( $v$  1.000000 1.000000 -1.000000).
- ▶ As outras partes indicam textura e normal (serão ignoradas por enquanto).

# Formato *WaveFront*

Cubo.obj

```
# Blender3D v249 OBJ File
v 1.000000 -1.000000 -1.000000
v 1.000000 -1.000000 1.000000
v -1.000000 -1.000000 1.000000
v -1.000000 -1.000000 -1.000000
v 1.000000 1.000000 -1.000000
v 0.999999 1.000000 1.000001
v -1.000000 1.000000 1.000000
v -1.000000 1.000000 -1.000000

f 5/1/1 1/2/1 4/3/1
f 5/1/1 4/3/1 8/4/1
f 3/5/2 7/6/2 8/7/2
f 3/5/2 8/7/2 4/8/2
f 2/9/3 6/10/3 3/5/3
f 6/10/4 7/6/4 3/5/4
f 1/2/5 5/1/5 2/9/5
f 5/1/6 6/10/6 2/9/6
f 5/1/7 8/11/7 6/10/7
f 8/11/7 7/12/7 6/10/7
f 1/2/8 2/9/8 3/13/8
f 1/2/8 3/13/8 4/14/8
```

Triângulo: f 5/1/1 1/2/1 4/3/1

Vetor 5: v 1.000000 1.000000 -1.000000

Vetor 1: v 1.000000 -1.000000  
-1.000000

Vetor 4: v -1.000000 -1.000000  
-1.000000

Após carregar e processar o arquivo,  
podemos usar o GL\_TRIANGLES, por  
exemplo.

# 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 06. Slides de Ricardo M. Marcacini. Disciplina SCC0250/0650, ICMC/USP, 2021.

# Exercício complementar

Com base nos exemplos da aula, importar um objeto no formato *WaveFront* (.obj) e aplicar as operações da Matriz *Model* (translação, rotação, escala) sobre ele. Devem ser exibidos os vértices do objeto antes e depois da operação. Algumas sugestões de onde encontrar objetos:

- ▶ <https://free3d.com/3d-models/> (filtrar por .obj).
- ▶ <https://www.blendswap.com/> (necessário abrir com *Blender* e exportar para .obj).
- ▶ Pesquise. Há muitos outros sites com modelos gratuitos.