Matemática e Física para Jogos
Unidade 01
Conceitos Básicos

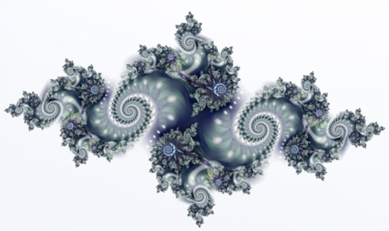
Matrizes de Transformação

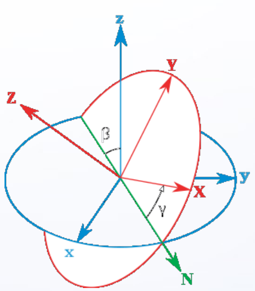
Gilvan Maia
gilvanmaia@virtual.ufc.br
Professor Adjunto

Instituto UFC Virtual
Universidade Federal do Ceará

Unidade 01
Conceitos Básicos

- Representação numérica no computador
- Funções
- Vetores
- **Matrizes**
- **Sistemas de Coordenadas**
- Quatérnios





Composição de Transformações

Matrizes, matrizes e matrizes

- Translações, rotações e escalas são úteis
- Mas como construir transformações mais complexas?
  - Basta produzir uma **matriz composta**
    - Produto de várias matrizes
  - Uma série de transformações torna-se uma só matriz
    - Além da beleza matemática, é computacionalmente eficiente
- É preciso entender como funciona a composição

Multiplicação Matricial

- Usando a notação em bloco

$$\mathbf{A} = \mathbf{BC} = \mathbf{B}\begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 & \mathbf{c}_4 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{B}\mathbf{c}_1 & \mathbf{B}\mathbf{c}_2 & \mathbf{B}\mathbf{c}_3 & \mathbf{B}\mathbf{c}_4 \end{bmatrix}$$

Cada produto $\mathbf{Bc}_i$ é válido e bem definido

B é uma matriz 4x4 e $\mathbf{c}_i$ é uma matriz-coluna 4x1

O resultado é matriz-coluna 4x1

- Assumindo matrizes 4x4
  - Cada coluna da segunda matriz é um vetor
  - Multiplicar duas matrizes equivale a multiplicar 4 vetores pela matriz da esquerda

Compondo Matrizes

- Analisemos a seguinte composição de matrizes

$$M_5 M_4 M_3 M_2 M_1 \mathbf{v}$$

- Qual transformação é realizada primeiro?

$$M_5 M_4 M_3 M_2 (M_1 \mathbf{v})$$
$$= M_5 \; M_4 M_3 M_2 (\mathbf{v}')$$
$$= M_5 M_4 M_3 (M_2 \mathbf{v}')$$
$$= M_5 \; M_4 M_3 (\mathbf{v}'')$$
$$\dots$$

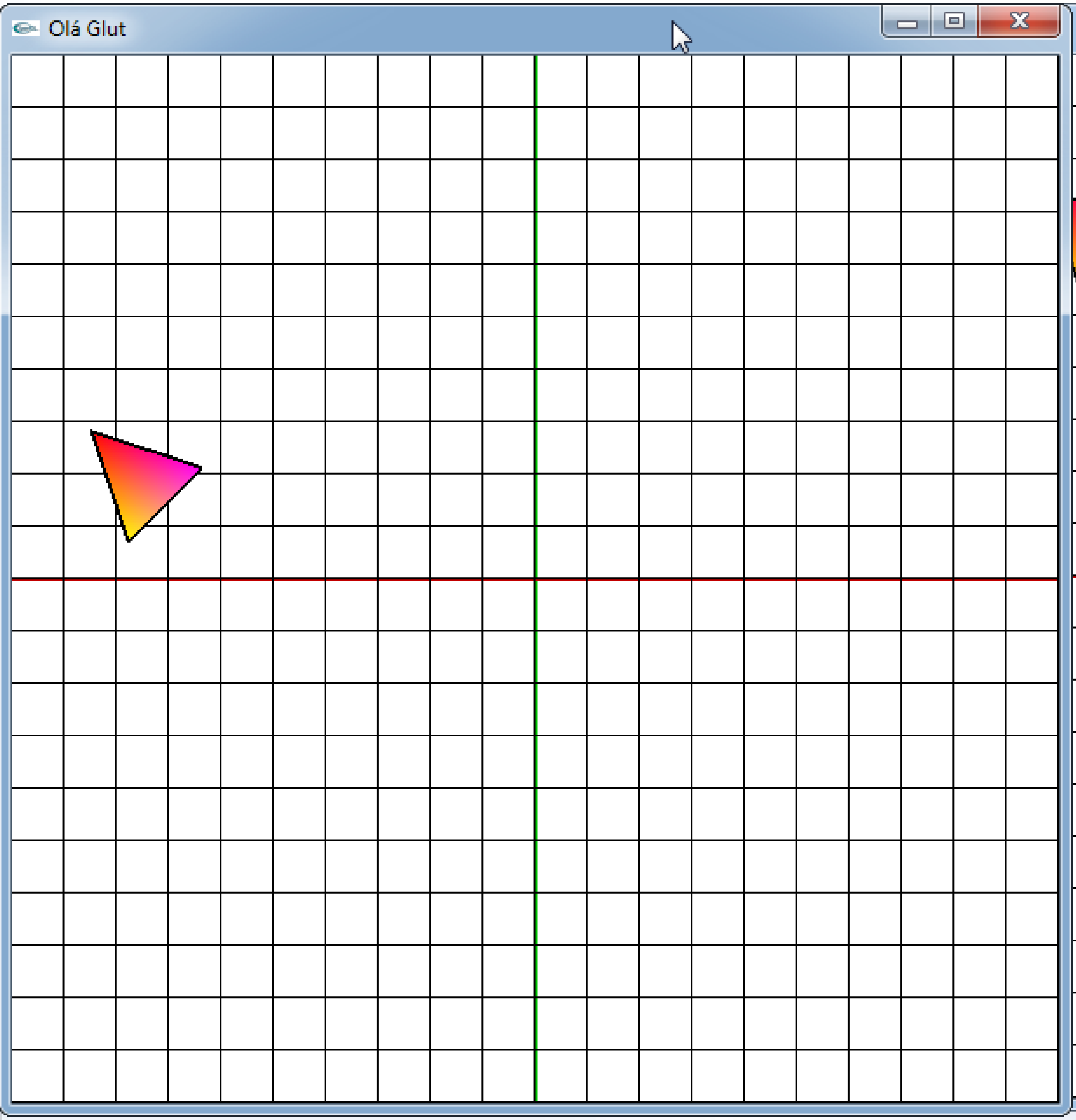
A Solução Está na Equação

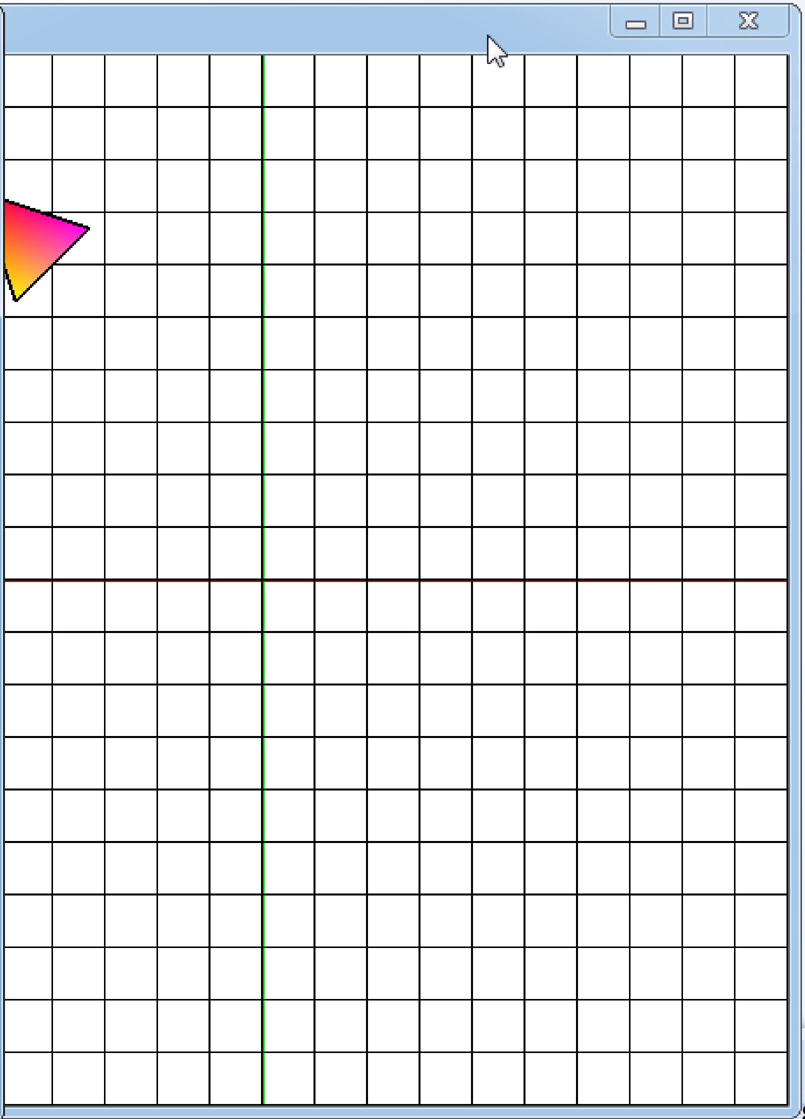
- Analisemos a seguinte composição de matrizes

$$M_5 M_4 M_3 M_2 M_1 \mathbf{v}$$

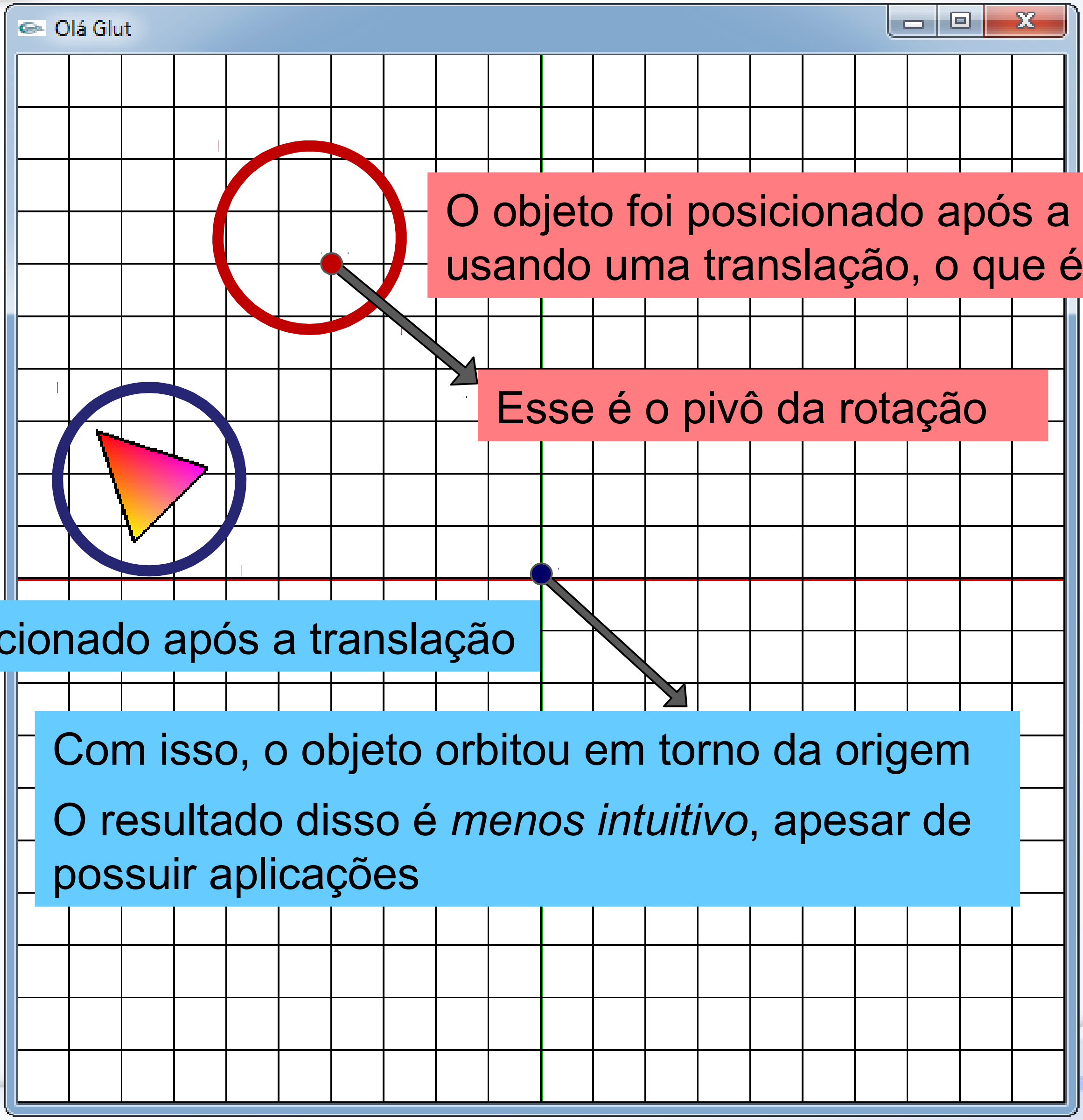
- Qual transformação é realizada primeiro?
  - As matrizes ***mais próximas ao vetor vêm primeiro***
    - Esse mnemônico é bastante intuitivo
  - A ordem afeta radicalmente o resultado final
    - É preciso cuidado ao construir transformações complexas

O (De)efeito da Ordem

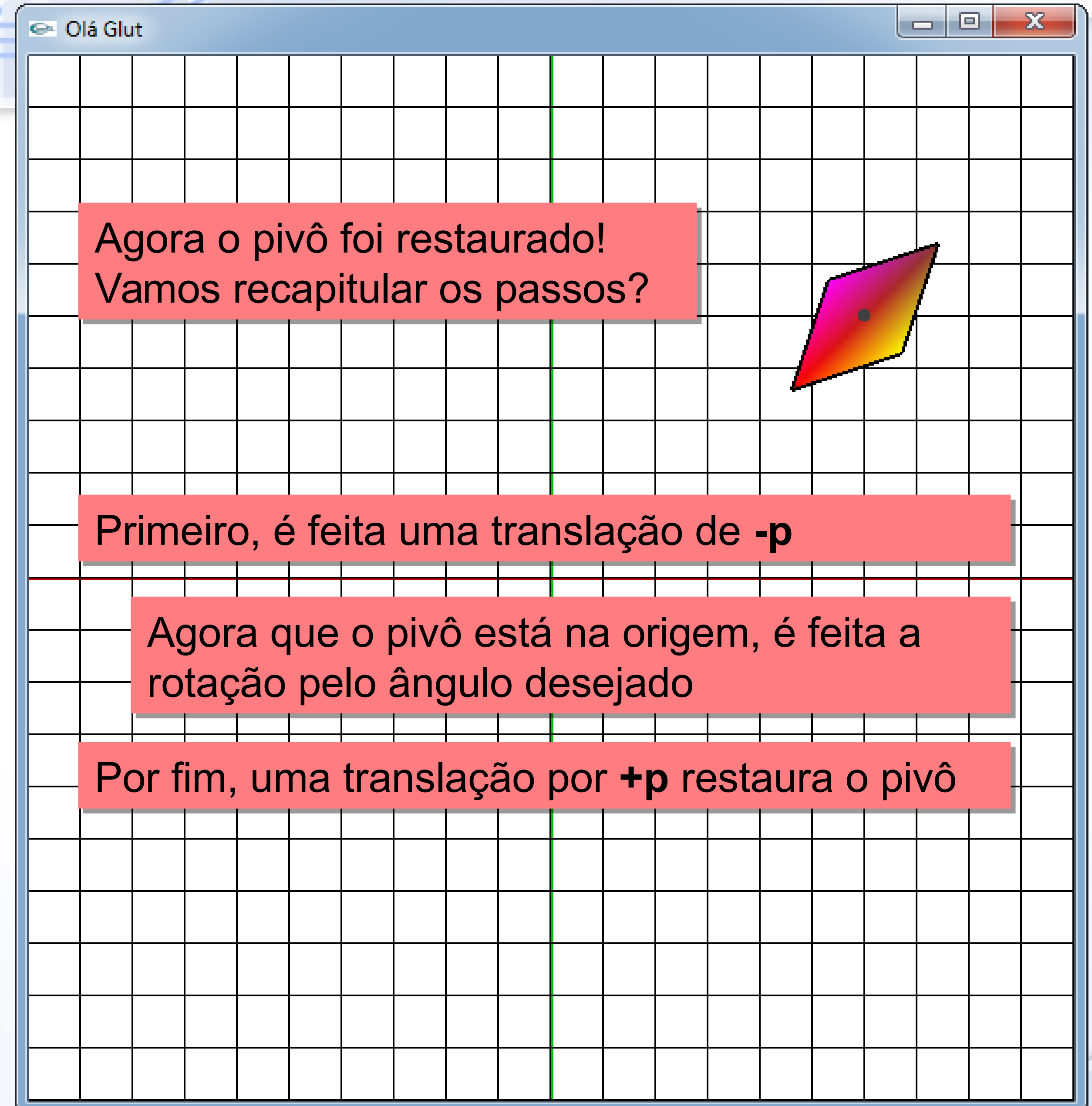




O (De)efeito da Ordem



Rotação em Torno de um Ponto Genérico



Agora o pivô foi restaurado!
Vamos recapitular os passos?

Primeiro, é feita uma translação de **-p**

Agora que o pivô está na origem, é feita a rotação pelo ângulo desejado

Por fim, uma translação por **+p** restaura o pivô

Usando da Matrizes...

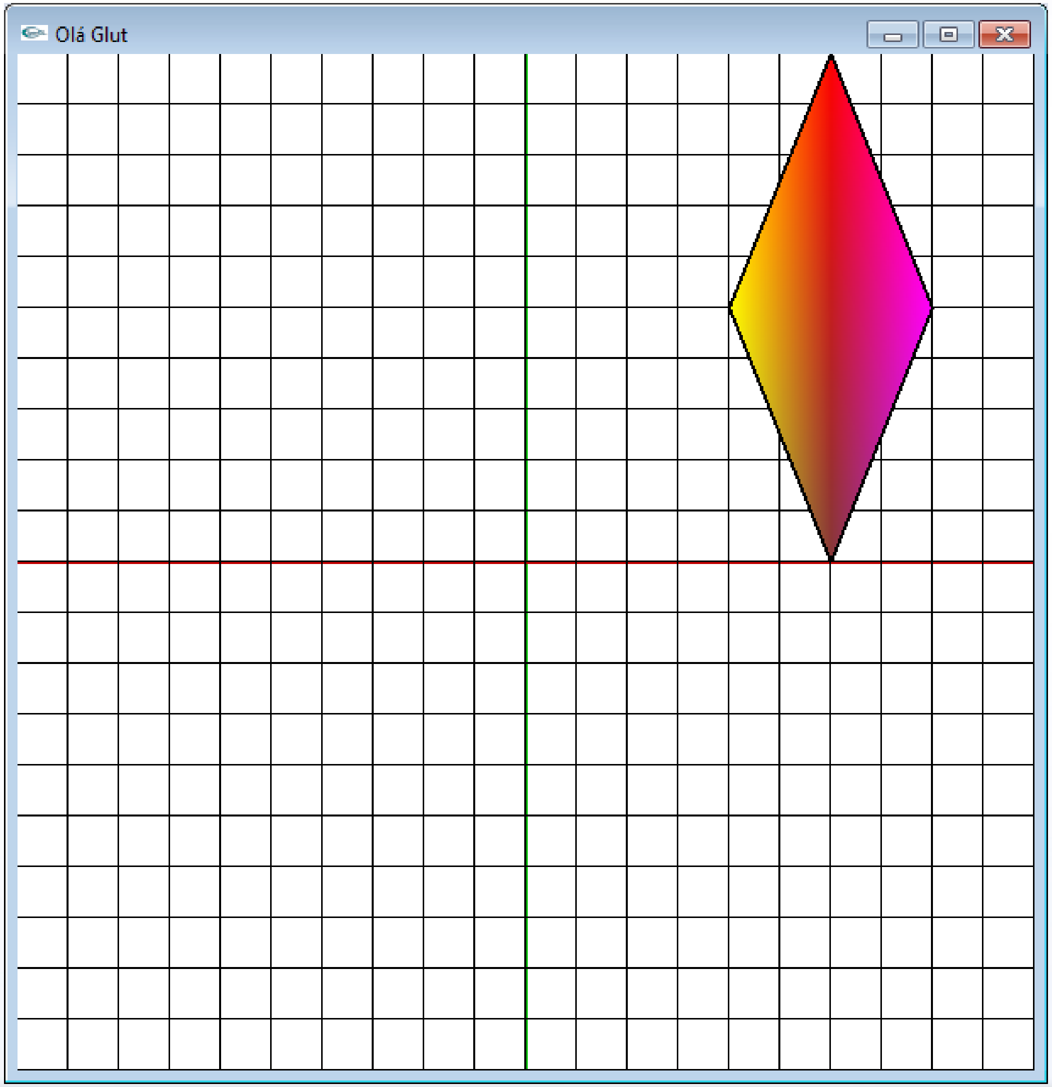
- A transformação é simples de se expressar

$$M = T(\mathbf{p})R(\boldsymbol{\theta})T(-\mathbf{p})$$

- Não se preocupe tanto com o cálculo
  - Escreva um bom código de produto de matrizes
  - Deixe que o computador trabalhar ^^

Escala em Relação a um Ponto Genérico

Mesmo Princípio da Rotação



Matriz Composta

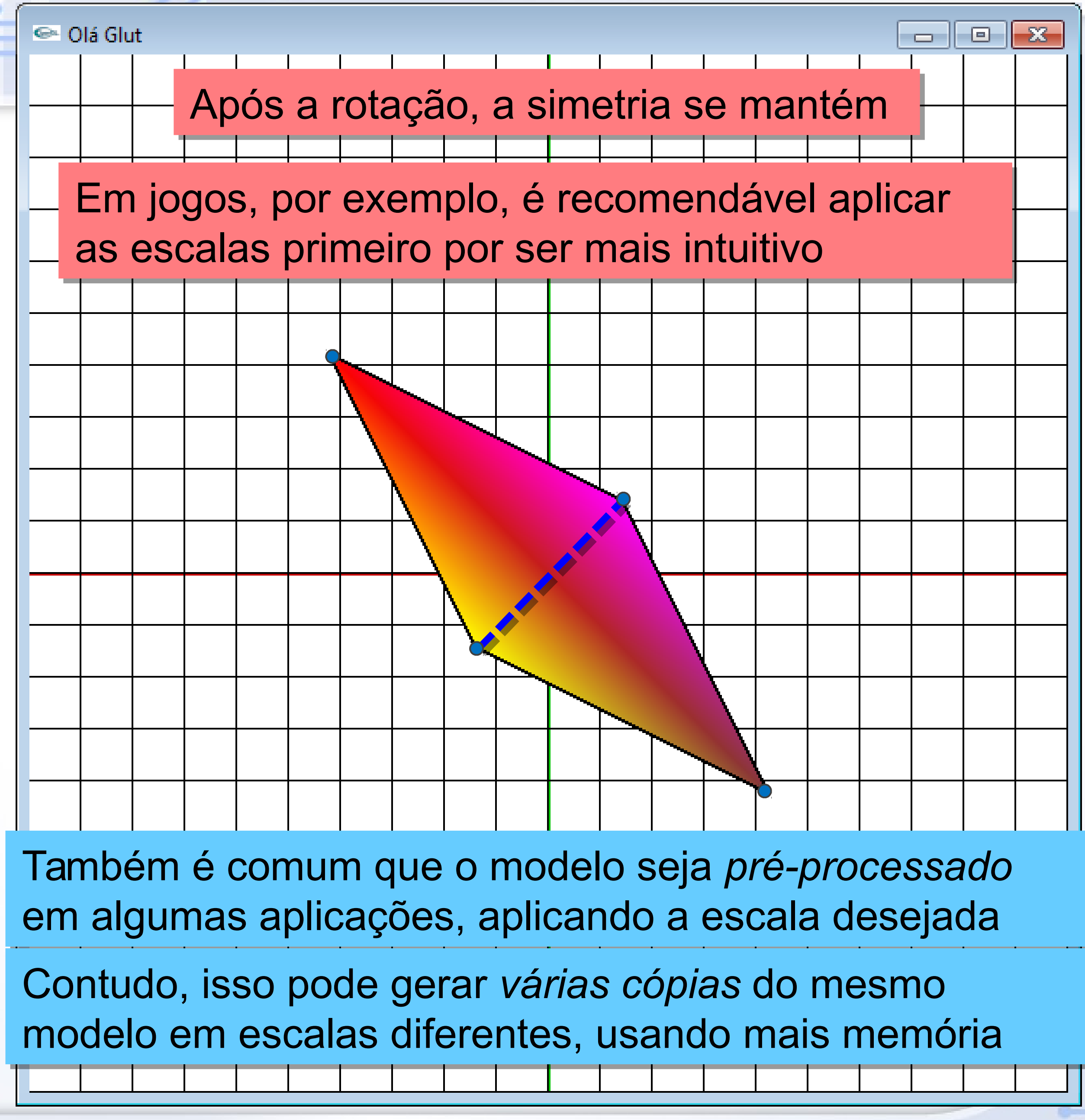
- A simples expressão descreve a transformação

$$M = T(\mathbf{p})S\left(s_x, s_y, s_z\right)T(-\mathbf{p})$$

Compondo Escalas e Rotações

Após a rotação, a simetria se mantém

Em jogos, por exemplo, é recomendável aplicar as escalas primeiro por ser mais intuitivo



Também é comum que o modelo seja *pré-processado* em algumas aplicações, aplicando a escala desejada

Contudo, isso pode gerar *várias cópias* do mesmo modelo em escalas diferentes, usando mais memória

Objetos, Matrizes e a Cena

- Instâncias de objetos na cena
  - Modelos geométricos replicados em vários pontos
- Descrição fundamental da instância
  - Posição
  - Tamanho
  - Orientação
- Representação intuitiva, amplamente disponível
  - Motores de jogos
  - Editores de cenários
  - Editores de modelos e de animações

Transformação de Modelo

- Sejam as seguintes matrizes
  - S, escala
  - R, rotação
  - T, translação
- As mesmas são compostas como
  - A escala vem primeiro, seguida de rotação e translação
- Expandindo, temos

$$M = TRS$$

- Que tal desenvolvermos um pouco essa matriz?

Matriz TRS

Matriz TRS

$$M = TRS$$

deslocamentos

fatores de escala

$$= \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \vec{\mathbf{r}}_x & \vec{\mathbf{r}}_y & \vec{\mathbf{r}}_z & \vec{\mathbf{0}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

*vetores ortonormais**

**unitários e perpendiculares entre si*

Matriz TRS

$$= \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \vec{\mathbf{r}}_x & \vec{\mathbf{r}}_y & \vec{\mathbf{r}}_z & \vec{\mathbf{0}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

vetor deslocamento

$$= \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x\vec{\mathbf{r}}_x & s_y\vec{\mathbf{r}}_y & s_z\vec{\mathbf{r}}_z & \vec{\mathbf{0}} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_x\mathbf{r}_x & s_y\mathbf{r}_y & s_z\mathbf{r}_z & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Essa forma matricial é bastante usada em motores gráficos e grafos de cena

É possível montar a matriz rapidamente conhecendo suas partes

Inversa da Matriz TRS

Todas essas inversas são triviais

$$\mathbf{M}^{-1} = (\mathbf{TRS})^{-1} = \mathbf{S}^{-1}\mathbf{R}^{-1}\mathbf{T}^{-1}$$

$$\mathbf{S}^{-1} = \mathbf{S}(s_x, s_y, s_z)^{-1} = \mathbf{S}\left(\frac{1}{s_x}, \frac{1}{s_y}, \frac{1}{s_z}\right)$$

$$\mathbf{R}^{-1} = \mathbf{R}^T$$

$$\mathbf{T}^{-1} = \mathbf{T}(t_x, t_y, t_z)^{-1} = \mathbf{T}(-t_x, -t_y, -t_z)$$

Inversa da Matriz TRS

$$\mathbf{M}^{-1} = (\mathbf{TRS})^{-1} = \mathbf{S}\left(\frac{1}{s_x}, \frac{1}{s_y}, \frac{1}{s_z}\right)\mathbf{R}^T\,\mathbf{T}(-t_x, -t_y, -t_z)$$

$$\mathbf{M}^{-1} = \begin{bmatrix} \dfrac{\vec{\mathbf{r}}_x^{\,T}}{s_x} & -\dfrac{\vec{\mathbf{r}}_x \vec{\mathbf{t}}}{s_x} \\ \dfrac{\vec{\mathbf{r}}_y^{\,T}}{s_y} & -\dfrac{\vec{\mathbf{r}}_y \vec{\mathbf{t}}}{s_y} \\ \dfrac{\vec{\mathbf{r}}_z^{\,T}}{s_z} & -\dfrac{\vec{\mathbf{r}}_z \vec{\mathbf{t}}}{s_z} \\ \vec{\mathbf{0}}^T & 1 \end{bmatrix}$$

A porção 3x3 da matriz é obtida por simples divisões de vetor por escalar

Como o vetor é transposto, temos três vetores-linhas *1x3*

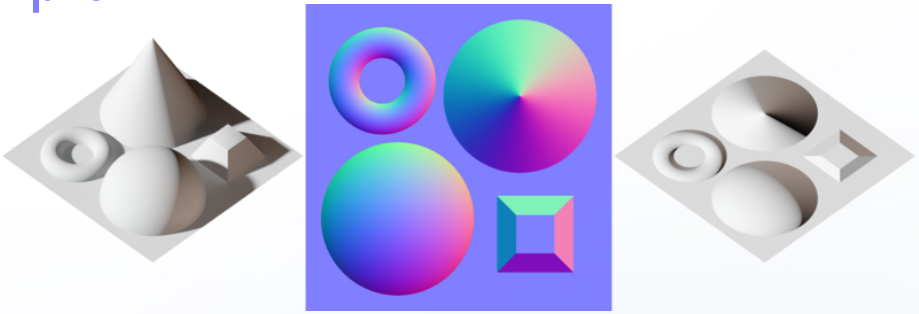
A última linha é idêntica à da matriz identidade

A última coluna é obtida por simples produtos escalares

Obviamente, cada valor é um *escalar*

Mas porque inversas?

- São necessárias em inúmeras aplicações
  - Selecionar um vértice de um modelo para editá-lo
  - Realizar testes de detecção de colisões
  - Implementar técnicas avançadas de iluminação como *normal mapping*, por exemplo



- Mas essa não é nossa preocupação agora
  - Deixemos esses tópicos avançados para o futuro
  - Falemos de câmeras e projeção