

Transformações Geométricas 2D – Parte 2

Pedro Ximenes

Universidade Católica de Pernambuco (UNICAP)

11 de maio de 2026

Sumário

- 1 Transformações Inversas
 - Visão Geral
 - Translação Inversa
 - Rotação Inversa
 - Escala Inversa
- 2 Transformações 2D Compostas
 - Motivação
 - Ordem das Matrizes
 - Sequências de Mesma Operação
 - Rotação com Ponto Arbitrário
 - Escala com Ponto Fixo
 - Escala com Ponto Fixo
 - Exemplo Completo
- 3 Outras Transformações 2D
 - Reflexão
 - Cisalhamento

Transformações Inversas – Introdução

O que são?

Para cada transformação geométrica básica (translação, rotação, escala) que aprendemos na aula passada, existe uma **versão inversa** – como se fosse “desfazer” a operação.

- A transformação inversa é representada com o símbolo -1 na matriz
- **Exemplo:** se você transladou um objeto 2 unidades para a direita, a translação inversa move 2 unidades para a **esquerda**
- São muito usadas nas **transformações compostas** (assunto desta aula!)

Resumo das inversas

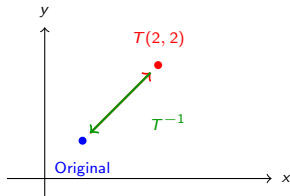
- **Translação inversa:** inverte o sinal de t_x e t_y
- **Rotação inversa:** inverte o sinal de θ
- **Escala inversa:** usa o **recíproco** dos fatores ($1/s_x$, $1/s_y$)

Translação Inversa

Ideia

Para “desfazer” uma translação com (t_x, t_y) , basta fazer uma translação com $(-t_x, -t_y)$.

$$T^{-1}(t_x, t_y) = \begin{pmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{pmatrix}$$



Cuidado!

Se o valor original de t_x era **negativo** (ex: $t_x = -3$), o inverso fica **positivo**:
 $-(-3) = +3$.

Rotação Inversa

Ideia

Para “desfazer” uma rotação de θ , basta rotacionar por $-\theta$ (ângulo oposto).

$$R^{-1}(\theta) = R(-\theta) = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Por que o cosseno não muda de sinal?

Lembrem da trigonometria básica:

- $\cos(-\theta) = \cos \theta$ (cosseno **não** é afetado pelo sinal)
- $\sin(-\theta) = -\sin \theta$ (seno é afetado pelo sinal)

Portanto, ao negativar θ , somente os senos trocam de posição!

Curiosidade

A matriz de rotação inversa é igual à **transposta** da matriz de rotação original:

$R^{-1}(\theta) = R^T(\theta)$. Compare as duas matrizes e perceba que basta trocar linhas por colunas!

Escala Inversa

Ideia

Para “desfazer” uma escala com fatores (s_x, s_y) , usamos os fatores **recíprocos**: $(1/s_x, 1/s_y)$.

$$S^{-1}(s_x, s_y) = \begin{pmatrix} 1/s_x & 0 & 0 \\ 0 & 1/s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Exemplo

Se a escala original duplicou o objeto ($s_x = s_y = 2$), a escala inversa usa $1/2$, reduzindo à metade. O objeto volta ao tamanho original.

Transformações Compostas – Por quê?

Na aula passada ficaram dois problemas em aberto:

Problema 1 – Rotação com ponto arbitrário

A matriz de rotação que aprendemos **só funciona** quando o ponto de rotação está na **origem** (0, 0). E se o ponto de rotação estiver em qualquer outra posição?

Problema 2 – Escala com ponto fixo

A escala que aprendemos, além de alterar o **tamanho**, tem o efeito colateral de **mover** o objeto (aproxima/afasta da origem). Como escalar **sem mover**?

Solução: Transformações Compostas!

Ambos os problemas são resolvidos com uma **sequência de transformações** (translação + rotação/escala + translação), combinadas em uma **única matriz composta**.

Graças às **coordenadas homogêneas**, todas as transformações são multiplicações de matrizes 3×3 . Uma sequência de transformações vira uma sequência de **multiplicações de matrizes**!

Ordem das Matrizes – De Trás para Frente

Regra fundamental

Quando temos uma sequência de transformações, as matrizes são **pós-multiplicadas**: a **primeira** operação a ser aplicada é a **última** matriz na equação.

$$P' = \boxed{M_3} \cdot \boxed{M_2} \cdot \boxed{M_1} \cdot \boxed{P}$$

↓ ↓ ↓
 3ª operação 2ª operação 1ª operação

Multiplicação de matrizes NÃO é comutativa!

$$M_1 \cdot M_2 \neq M_2 \cdot M_1 \quad (\text{no geral})$$

Se você **trocar a ordem**, o resultado geralmente será **diferente**!

Porém, a multiplicação de matrizes é **associativa**:

$$(M_3 \cdot M_2) \cdot M_1 = M_3 \cdot (M_2 \cdot M_1)$$

Você pode multiplicar as matrizes na ordem que quiser, **desde que não troque a posição das matrizes**!

Associatividade – Cuidado com a Ordem!

Dadas três matrizes M_3 , M_2 e M_1 (de trás para frente), você pode calcular de duas formas:

Opção A

- 1 Calcule $M_3 \times M_2 = R_1$
- 2 Calcule $R_1 \times M_1 = M$

Opção B

- 1 Calcule $M_2 \times M_1 = R_2$
- 2 Calcule $M_3 \times R_2 = M$

Ambas dão o **mesmo** resultado M !

ERROS comuns (o que NÃO fazer)

- **NÃO:** $M_2 \times M_3$ (trocou a ordem!)
- **NÃO:** $M_1 \times M_2$ (trocou a ordem!)
- **NÃO:** Resultado $\times M_3$ (matriz no lado errado!)

A chance de errar é alta! **Nunca inverte a posição** das matrizes durante a multiplicação.

Sequência de Translações

Quando combinamos **duas translações**, basta **somar** os fatores:

$$T(t_{2x}, t_{2y}) \cdot T(t_{1x}, t_{1y}) = \begin{pmatrix} 1 & 0 & t_{2x} \\ 0 & 1 & t_{2y} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & t_{1x} \\ 0 & 1 & t_{1y} \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_{1x} + t_{2x} \\ 0 & 1 & t_{1y} + t_{2y} \\ 0 & 0 & 1 \end{pmatrix}$$

Resultado

$$T(t_{2x}, t_{2y}) \cdot T(t_{1x}, t_{1y}) = T(t_{1x} + t_{2x}, t_{1y} + t_{2y})$$

Exemplo

$$T(3, 2) \cdot T(1, 4) = T(4, 6)$$

Transladar (1, 4) e depois (3, 2) é o mesmo que transladar (4, 6) de uma vez.

Sequência de Rotações

Quando combinamos **duas rotações**, basta **somar** os ângulos:

$$R(\theta_2) \cdot R(\theta_1) = \begin{pmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c_{12} & -s_{12} & 0 \\ s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

onde $c_i = \cos \theta_i$, $s_i = \sin \theta_i$, e $c_{12} = \cos(\theta_1 + \theta_2)$, $s_{12} = \sin(\theta_1 + \theta_2)$.

Resultado

$$R(\theta_2) \cdot R(\theta_1) = R(\theta_1 + \theta_2)$$

Exemplo

$$R(30^\circ) \cdot R(20^\circ) = R(50^\circ)$$

Rotacionar 20° e depois 30° é o mesmo que rotacionar 50° de uma vez.

Sequência de Escalas

Quando combinamos **duas escalas**, basta **multiplicar** os fatores:

$$S(s_{2x}, s_{2y}) \cdot S(s_{1x}, s_{1y}) = \begin{pmatrix} s_{2x} & 0 & 0 \\ 0 & s_{2y} & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_{1x} & 0 & 0 \\ 0 & s_{1y} & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s_{1x} \cdot s_{2x} & 0 & 0 \\ 0 & s_{1y} \cdot s_{2y} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Resultado

$$S(s_{2x}, s_{2y}) \cdot S(s_{1x}, s_{1y}) = S(s_{1x} \cdot s_{2x}, s_{1y} \cdot s_{2y})$$

Exemplo

$$S(3, 3) \cdot S(2, 2) = S(6, 6)$$

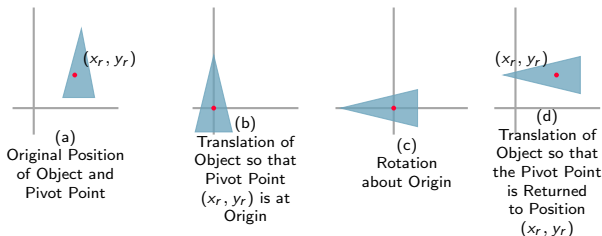
Escalar por (2, 2) e depois por (3, 3) é o mesmo que escalar por (6, 6) de uma vez.

Rotação com Ponto de Rotação Arbitrário

Problema

Queremos rotacionar um objeto por θ ao redor de um ponto (x_r, y_r) que **não** está na origem.

Solução em 3 passos:



- 1 **Mover** o ponto de rotação para a origem: $T(-x_r, -y_r)$
- 2 **Rotacionar** pelo ângulo θ : $R(\theta)$
- 3 **Mover** o ponto de rotação de volta: $T(x_r, y_r)$

Rotação com Ponto Arbitrário – Matriz Composta

Equação da Matriz Composta

$$R(x_r, y_r, \theta) = \underbrace{T(x_r, y_r)}_{3^{\text{a}} \text{ op.}} \cdot \underbrace{R(\theta)}_{2^{\text{a}} \text{ op.}} \cdot \underbrace{T(-x_r, -y_r)}_{1^{\text{a}} \text{ op.}}$$

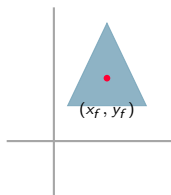
Expandindo as matrizes:

$$\begin{pmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{pmatrix}$$

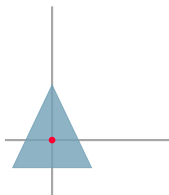
Lembrete – Ordem de leitura

De trás para frente: a 1ª operação aplicada ($T(-x_r, -y_r)$) é a **última** na equação.

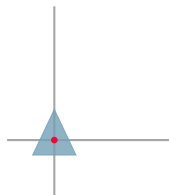
Escala com Ponto Fixo



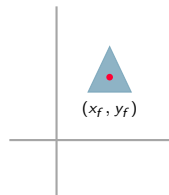
(a)
Original Position
of Object and
Fixed Point



(b)
Translate Object
so that Fixed Point
(x_f, y_f) is at Origin



(c)
Scale Object
with Respect
to Origin



(d)
Translate Object
so that the Fixed
Point is Returned
to Position (x_f, y_f)

Solução em 3 passos (idêntica à rotação com ponto arbitrário):

- 1 **Mover** o ponto fixo para a origem: $T(-x_f, -y_f)$
- 2 **Escalar** com os fatores s_x, s_y : $S(s_x, s_y)$
- 3 **Mover** o ponto fixo de volta: $T(x_f, y_f)$

Escala com Ponto Fixo

Problema

A escala básica, além de alterar o tamanho, **move** o objeto (aproxima/afasta da origem). Queremos escalar **sem mover** o objeto, usando um **ponto fixo** (x_f, y_f) como “âncora”.

Solução em 3 passos (idêntica à rotação com ponto arbitrário):

- ❶ **Mover** o ponto fixo para a origem: $T(-x_f, -y_f)$
- ❷ **Escalar** com os fatores s_x, s_y : $S(s_x, s_y)$
- ❸ **Mover** o ponto fixo de volta: $T(x_f, y_f)$

Equação da Matriz Composta

$$S(x_f, y_f, s_x, s_y) = \underbrace{T(x_f, y_f)}_{3^{\text{a}} \text{ op.}} \cdot \underbrace{S(s_x, s_y)}_{2^{\text{a}} \text{ op.}} \cdot \underbrace{T(-x_f, -y_f)}_{1^{\text{a}} \text{ op.}}$$

O ponto fixo funciona como uma **âncora**: o objeto muda de tamanho ao redor dele, mas a âncora **não** se move.

Escala com Ponto Fixo – Matrizes

Expandindo as três matrizes:

$$\underbrace{\begin{pmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{pmatrix}}_{T(x_f, y_f)} \cdot \underbrace{\begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{S(s_x, s_y)} \cdot \underbrace{\begin{pmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{pmatrix}}_{T(-x_f, -y_f)}$$

Resultado da multiplicação (matriz composta)

$$S(x_f, y_f, s_x, s_y) = \begin{pmatrix} s_x & 0 & x_f(1 - s_x) \\ 0 & s_y & y_f(1 - s_y) \\ 0 & 0 & 1 \end{pmatrix}$$

Observe: a última linha continua sendo (0 0 1). Se ao calcular você obtiver algo diferente, **algo está errado!**

Multiplicação de matrizes não é comutativa

- Multiplicação de matrizes **não é comutativa**:

$$M_2 \cdot M_1 \neq M_1 \cdot M_2$$

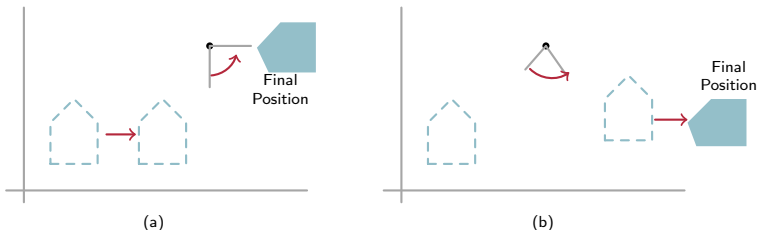
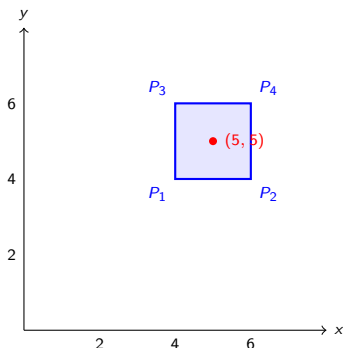


Figura: (a) primeiro o objeto é transladado depois rotacionado em 45° ;

(b) primeiro o objeto é rotacionado em 45° , depois transladado.

Exemplo – Rotação de um Quadrado

Problema: Um quadrado com vértices $P_1 = (4, 4)$, $P_2 = (6, 4)$, $P_3 = (4, 6)$, $P_4 = (6, 6)$. Rotacionar 45° usando o **centroide** como ponto de rotação.



- Centroide: ponto “médio” do quadrado
- Canto inferior esquerdo: $(4, 4)$, superior direito: $(6, 6)$
- Centroide: $\left(\frac{4+6}{2}, \frac{4+6}{2}\right) = (5, 5)$
- Logo: $x_r = 5$, $y_r = 5$

Dica

$$\cos 45^\circ = \sin 45^\circ = \frac{\sqrt{2}}{2} \approx 0,7071$$

Exemplo – Montando as Matrizes

Passo 1: Translação $T(-5, -5)$ (mover ponto de rotação para a origem):

$$T(-5, -5) = \begin{pmatrix} 1 & 0 & -5 \\ 0 & 1 & -5 \\ 0 & 0 & 1 \end{pmatrix}$$

Passo 2: Rotação $R(45^\circ)$:

$$R(45^\circ) = \begin{pmatrix} 0,7071 & -0,7071 & 0 \\ 0,7071 & 0,7071 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Passo 3: Translação $T(5, 5)$ (volta à posição original):

$$T(5, 5) = \begin{pmatrix} 1 & 0 & 5 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{pmatrix}$$

Ordem na equação

$$M = T(5, 5) \cdot R(45^\circ) \cdot T(-5, -5) \quad (\text{de trás para frente!})$$

Exemplo – Matriz Composta e Resultado

Multiplicando as três matrizes, obtemos a **matriz composta**:

$$M = \begin{pmatrix} 0,7071 & -0,7071 & 5,0000 \\ 0,7071 & 0,7071 & -2,0711 \\ 0 & 0 & 1 \end{pmatrix}$$

Agora aplicamos M a cada vértice. Exemplo com $P_1 = (4, 4)$:

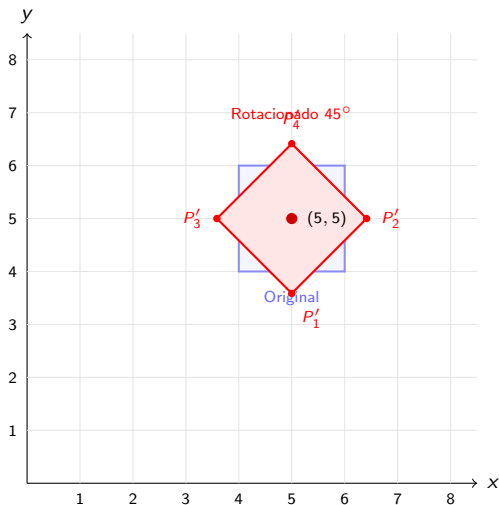
$$P'_1 = M \cdot \begin{pmatrix} 4 \\ 4 \\ 1 \end{pmatrix} = \begin{pmatrix} 5,0000 \\ 3,5858 \\ 1 \end{pmatrix}$$

Resultados para todos os vértices

Vértice	Original	Após rotação
P_1	(4, 4)	(5,0000; 3,5858)
P_2	(6, 4)	(6,4142; 5,0000)
P_3	(4, 6)	(3,5858; 5,0000)
P_4	(6, 6)	(5,0000; 6,4142)

Observe: a última coordenada sempre dá 1. Se deu diferente, **algo está errado!**

Exemplo – Resultado Visual



A **mesma matriz composta** é aplicada a todos os vértices. Quanto mais vértices tiver o objeto, mais multiplicações serão necessárias, mas a matriz é sempre a mesma!

Propriedade Importante no OpenGL

Matrizes pós-multiplicadas

As matrizes que aprendemos são todas **pós-multiplicadas** (de trás para frente). Esse mesmo comportamento acontece no **OpenGL** e no **WebGL**.

O que isso significa na prática?

- No código OpenGL/WebGL, as transformações são especificadas na **ordem inversa** em que são aplicadas
- A última transformação no código é a primeira a ser aplicada ao objeto
- É exatamente o “de trás para frente” que aprendemos!

Resultado diferente ao trocar ordem

Se você inverter a ordem das transformações no código, o resultado visual será **completamente diferente**. Muito cuidado!

Reflexão – Conceito

O que é?

A reflexão produz uma imagem “espelhada” do objeto em relação a um **eixo de reflexão**. É como se colocássemos um espelho no eixo escolhido.

Existem três casos básicos de reflexão:

- 1 Reflexão em relação ao **eixo** x ($y = 0$)
- 2 Reflexão em relação ao **eixo** y ($x = 0$)
- 3 Reflexão em relação à **origem** ($x = 0$ e $y = 0$)

Transformação mais simples

A reflexão é uma das transformações geométricas **mais fáceis** de calcular – em muitos casos dá para fazer de cabeça, bastando **negativar** a coordenada correta.

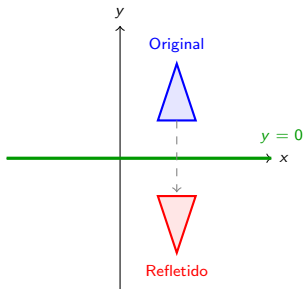
Reflexão em Relação ao Eixo x ($y = 0$)

Refletir em torno do eixo x significa **negativar a coordenada y** :

$$x' = x, \quad y' = -y$$

Matriz de Reflexão ($y = 0$)

$$F_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



Exemplo

Ponto $(3, 5)$ refletido em $y = 0$: $(3, -5)$. Basta trocar o sinal do y !

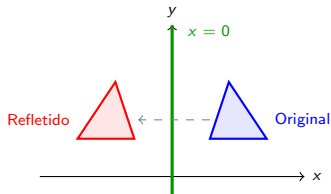
Reflexão em Relação ao Eixo y ($x = 0$)

Refletir em torno do eixo y significa **negativar a coordenada x** :

$$x' = -x, \quad y' = y$$

Matriz de Reflexão ($x = 0$)

$$F_y = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



Exemplo

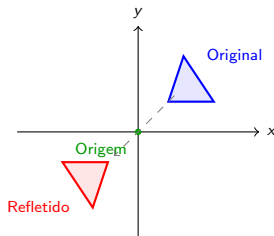
Ponto $(3, 5)$ refletido em $x = 0$: $(-3, 5)$. Basta trocar o sinal do x !

Reflexão em Relação à Origem

Refletir em torno da origem significa **negativar ambas** as coordenadas: $x' = -x$, $y' = -y$

Matriz de Reflexão (origem)

$$F_o = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



Exemplo

Ponto $(3, 5)$ refletido na origem: $(-3, -5)$. Basta trocar o sinal de ambas as coordenadas!

Cisalhamento (Shear) – Conceito

O que é?

O cisalhamento **distorce** o formato do objeto em uma direção (x ou y), transformando, por exemplo, um quadrado em um **paralelogramo**.

- O fator de cisalhamento é representado por sh (do inglês *shear*)
- Existem dois tipos:
 - Cisalhamento na **direção** x : distorce horizontalmente
 - Cisalhamento na **direção** y : distorce verticalmente

Notação

sh_x = fator de cisalhamento na direção x ; sh_y = fator na direção y . O livro usa “ sh ” (de *shear*) para manter consistência com a notação em inglês.

Cisalhamento na Direção x

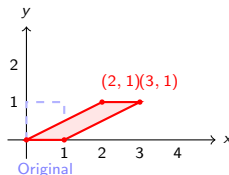
Equações

$$x' = x + sh_x \cdot y, \quad y' = y$$

A coordenada y **não muda**. A coordenada x é deslocada proporcionalmente ao y .

Matriz

$$SH_x = \begin{pmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



Exemplo com $sh_x = 2$ – vértices $(0,0)$, $(1,0)$, $(1,1)$, $(0,1)$

- $(0,0) \rightarrow (0,0)$; $(1,0) \rightarrow (1,0)$ (base não muda, $y = 0$)
- $(0,1) \rightarrow (2,1)$; $(1,1) \rightarrow (3,1)$ (topo deslocou!)

Cisalhamento na Direção y

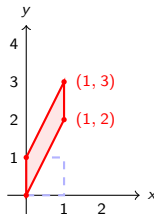
Equações

$$x' = x, \quad y' = y + sh_y \cdot x$$

A coordenada x **não muda**. A coordenada y é deslocada proporcionalmente ao x .

Matriz

$$SH_y = \begin{pmatrix} 1 & 0 & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



Exemplo com $sh_y = 2$ – vértices $(0,0)$, $(1,0)$, $(1,1)$, $(0,1)$

- $(0,0) \rightarrow (0,0)$; $(0,1) \rightarrow (0,1)$ (lado esquerdo, $x = 0$)
- $(1,0) \rightarrow (1,2)$; $(1,1) \rightarrow (1,3)$ (lado direito deslocou!)

Resumo – Todas as Matrizes 2D

Transformações Básicas (da aula passada)

$$T = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \quad R = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad S = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Reflexão e Cisalhamento (esta aula)

$$F_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} F_y = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} SH_x = \begin{pmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} SH_y = \begin{pmatrix} 1 & 0 & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Regra geral para todas as matrizes

Última linha **sempre** (0 0 1) e pontos representados como $(x, y, 1)^T$

Referências



Angel, E.; Shreiner, D. *Interactive Computer Graphics: A Top-Down Approach with WebGL*. 8th edition. Capítulo 5.



Mount, D. *CMSC 427 – Computer Graphics Lecture Notes*. University of Maryland. Aulas 8–9.



Material de Introdução à Computação Gráfica – IME-USP. Disponível em: <https://panda.ime.usp.br/introcg/static/introcg>