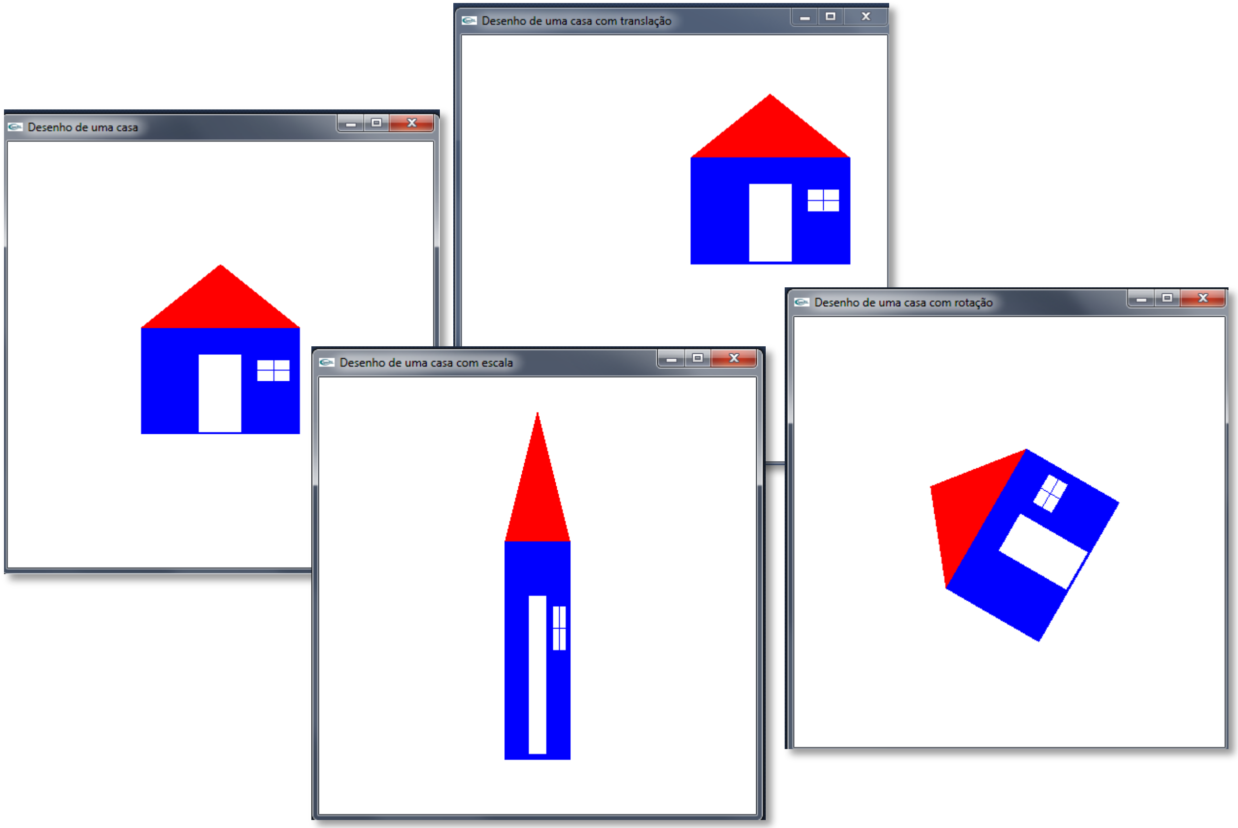


Transformações 2D

Rossana Baptista Queiroz

UNISINOS

27 de março de 2014

Matemática para CG

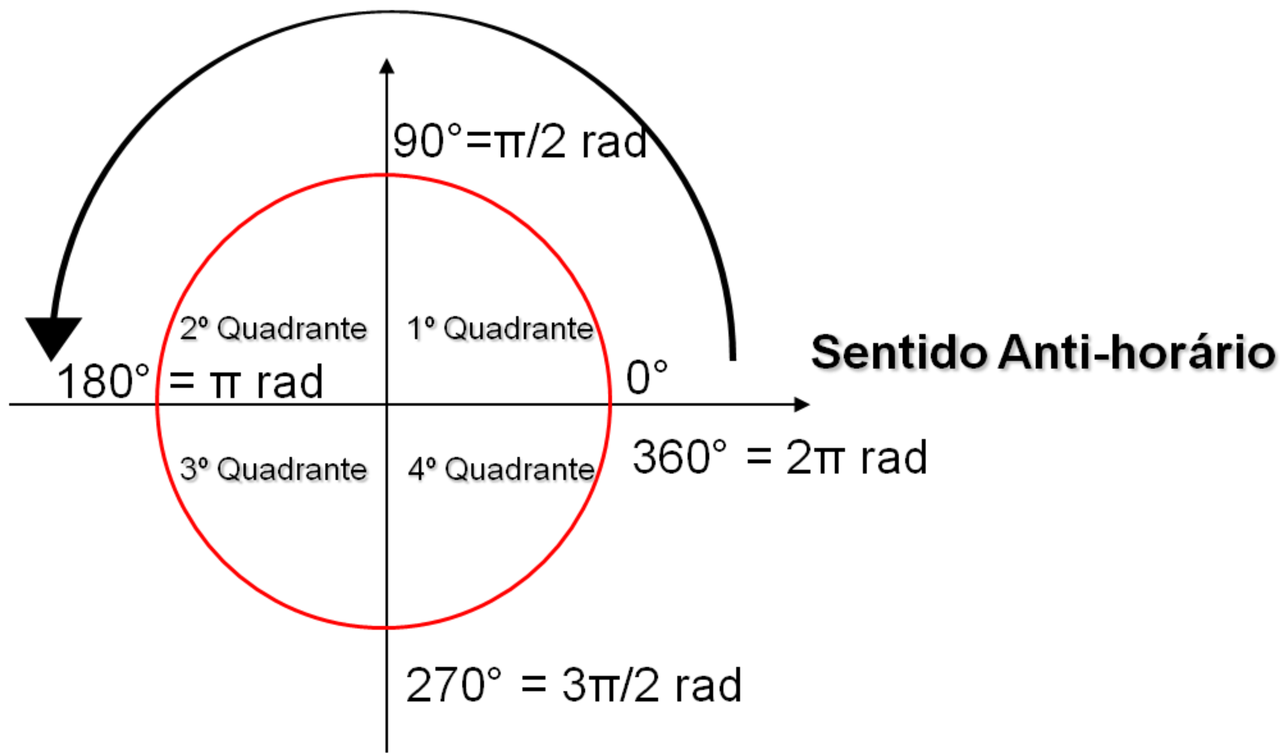
Revisão rápida...

Requisitos Matemáticos

- Trigonometria
- Matrizes

Trigonometria

Ciclo Trigonométrico



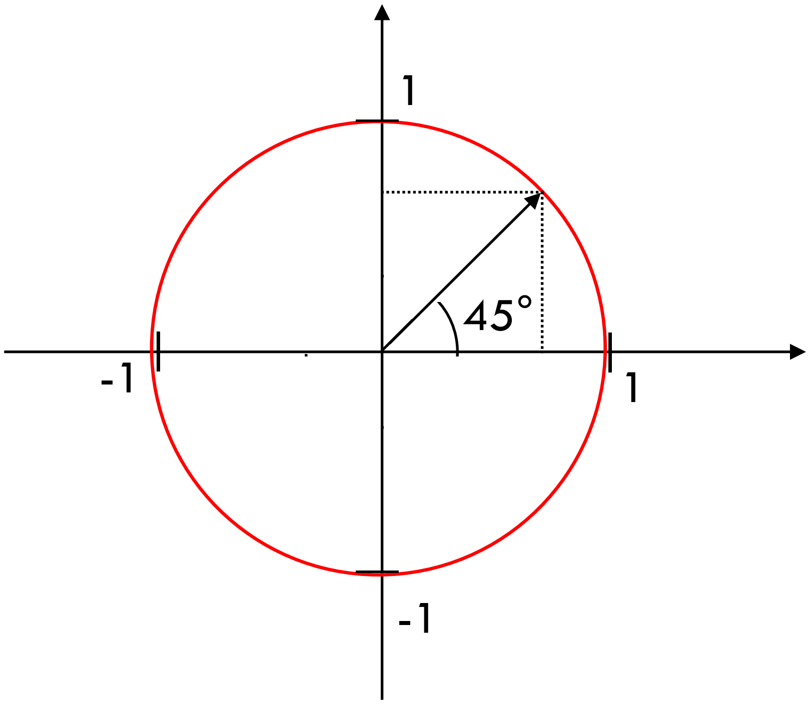
Unidades de medida

Grau: divisão da circunferência em 360 partes.

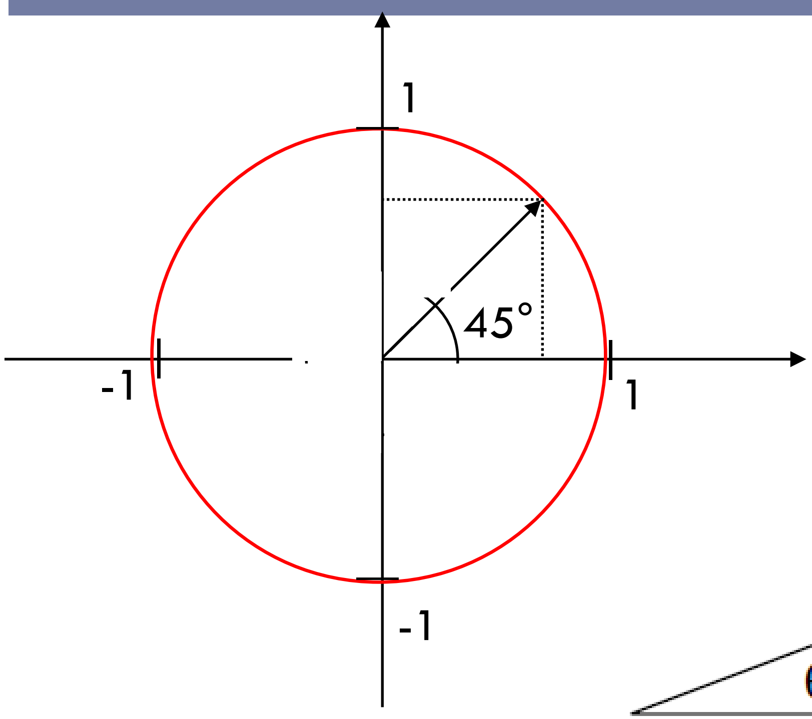
Radiano: um arco de um rad é igual ao raio

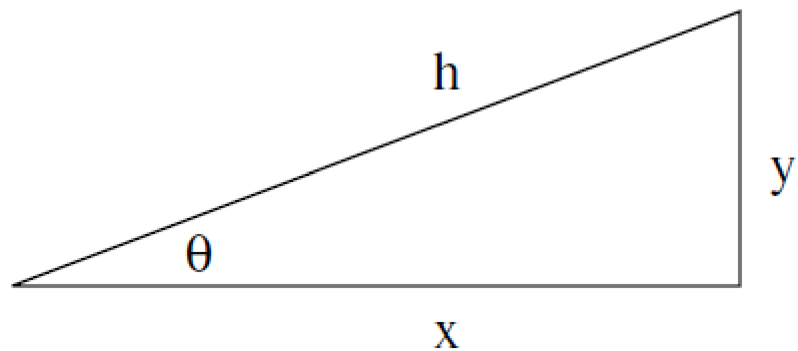
Trigonometria

Círculo Canônico



Trigonometria





$\sin\theta = y/r$

$\cos\theta = x/r$

$\tan\theta = y/x$

$x = r\cos\theta = y/\tan\theta$

$y = r\sin\theta = x\tan\theta$

$r = y/\sin\theta = x/\cos\theta$

$\sin^{-1}(y/r) = \theta$

$\cos^{-1}(x/r) = \theta$

$\tan^{-1}(y/x) = \theta$

Matrizes

Linhas

$$A_{mxn} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$$

Colunas

Operações sobre Matrizes

Adição

$$A = \begin{bmatrix} \boxed{a_{11}} & \boxed{a_{12}} & \boxed{a_{13}} \\ \boxed{a_{21}} & \boxed{a_{22}} & \boxed{a_{23}} \\ \boxed{a_{31}} & \boxed{a_{32}} & \boxed{a_{33}} \end{bmatrix} \quad B = \begin{bmatrix} \boxed{b_{11}} & \boxed{b_{12}} & \boxed{b_{13}} \\ \boxed{b_{21}} & \boxed{b_{22}} & \boxed{b_{23}} \\ \boxed{b_{31}} & \boxed{b_{32}} & \boxed{b_{33}} \end{bmatrix} \quad C = A + B$$

$$C = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

Obs: A e B devem ser do mesmo tamanho.

Operações sobre Matrizes

Subtração

$$A = \begin{bmatrix} \boxed{a_{11}} & \boxed{a_{12}} & \boxed{a_{13}} \\ \boxed{a_{21}} & \boxed{a_{22}} & \boxed{a_{23}} \\ \boxed{a_{31}} & \boxed{a_{32}} & \boxed{a_{33}} \end{bmatrix} \qquad B = \begin{bmatrix} \boxed{b_{11}} & \boxed{b_{12}} & \boxed{b_{13}} \\ \boxed{b_{21}} & \boxed{b_{22}} & \boxed{b_{23}} \\ \boxed{b_{31}} & \boxed{b_{32}} & \boxed{b_{33}} \end{bmatrix} \qquad C = A - B$$

$$C = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

Obs: A e B devem ser do mesmo tamanho.

Operações sobre Matrizes

Multiplicação de matriz por escalar

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad B = 3 * A \qquad B = 3 * \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$B = \begin{bmatrix} 3*a_{11} & 3*a_{12} & 3*a_{13} \\ 3*a_{21} & 3*a_{22} & 3*a_{23} \\ 3*a_{31} & 3*a_{32} & 3*a_{33} \end{bmatrix}$$

Operações sobre Matrizes

Multiplicação de matrizes

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix}$$

$$C_{m \times n} = A_{m \times k} \cdot B_{k \times n}$$

$$C_{2 \times 2} = A_{2 \times 3} \cdot B_{3 \times 2}$$

$$C = \begin{bmatrix} \quad \end{bmatrix} \qquad C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

Obs: A_n e B_m devem ser do mesmo tamanho.

Matrizes

Matriz transposta

Ocorre a troca entre os elementos m e n das matrizes.

$$\boxed{A_{m\times n} = A^{t}{}_{n\times m}}$$

$$A = \begin{bmatrix} 2 & 3 \\ 5 & 0 \\ 8 & 6 \end{bmatrix} \qquad A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

$$A^{t} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

Resumo Operações sobre Matrizes

Adição

$$C = \begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & a_{13}+b_{13} \\ a_{21}+b_{21} & a_{22}+b_{22} & a_{23}+b_{23} \\ a_{31}+b_{31} & a_{32}+b_{32} & a_{33}+b_{33} \end{bmatrix}$$

Obs: A e B devem ser do mesmo tamanho.

Subtração

$$C = \begin{bmatrix} a_{11}-b_{11} & a_{12}-b_{12} & a_{13}-b_{13} \\ a_{21}-b_{21} & a_{22}-b_{22} & a_{23}-b_{23} \\ a_{31}-b_{31} & a_{32}-b_{32} & a_{33}-b_{33} \end{bmatrix}$$

Multiplicação de matriz por escalar

$$B = n * \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Multiplicação de matrizes

$$C = \begin{bmatrix} a_{11}\cdot b_{11}+a_{12}\cdot b_{21}+a_{13}\cdot b_{31} & a_{11}\cdot b_{12}+a_{12}\cdot b_{22}+a_{13}\cdot b_{32} \\ a_{21}\cdot b_{11}+a_{22}\cdot b_{21}+a_{23}\cdot b_{31} & a_{21}\cdot b_{12}+a_{22}\cdot b_{22}+a_{23}\cdot b_{32} \end{bmatrix}$$

$$\boxed{C_{m\times n} = A_{m\times k}\cdot B_{k\times n}}$$

Pipeline de visualização 2D
Continuação

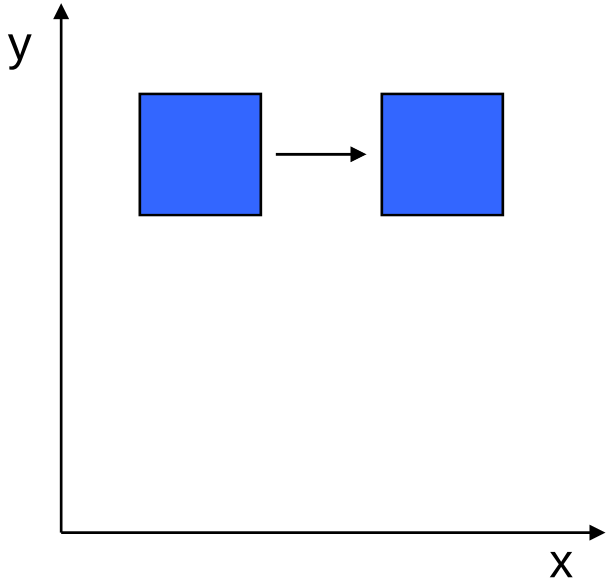
Transformações geométricas

Transformações Geométricas 2D

Translação

- A operação de translação movimenta todos os pontos de um polígono.

Translação no eixo X

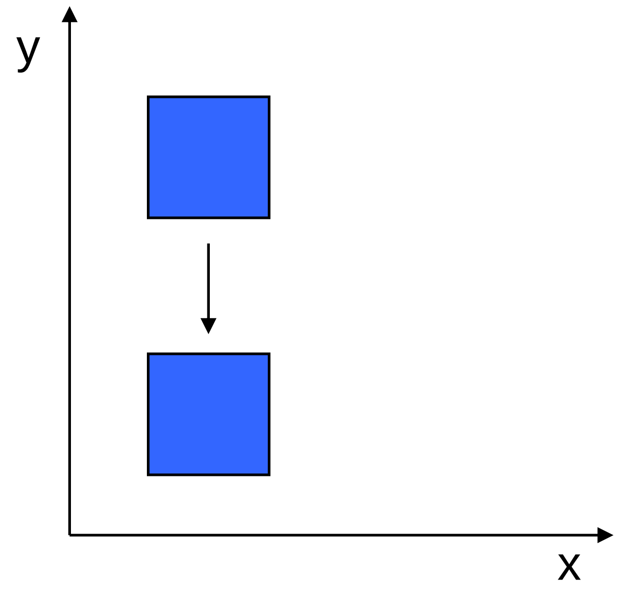


Transformações Geométricas 2D

Translação

- A operação de translação movimenta todos os pontos de um polígono.

Translação no eixo Y

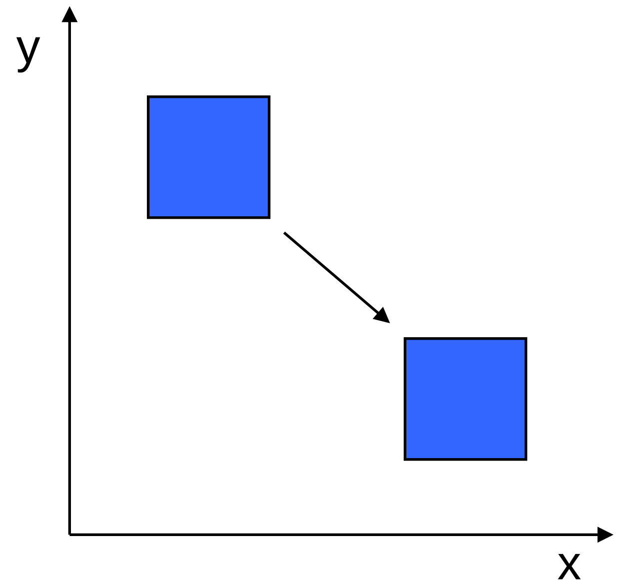


Transformações Geométricas 2D

Translação

- A operação de translação movimenta todos os pontos de um polígono.

Translação em X e Y



Transformações Geométricas 2D

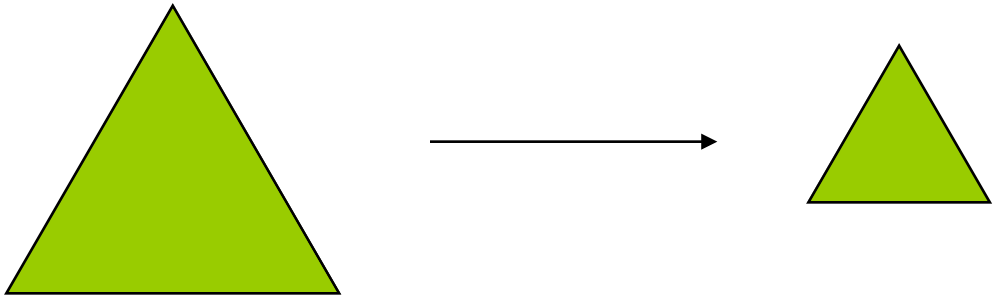
Translação

$$xt = x + Tx$$
$$yt = y + Ty$$

Transformações Geométricas 2D

Escala

- A operação de escala muda as dimensões de um polígono.

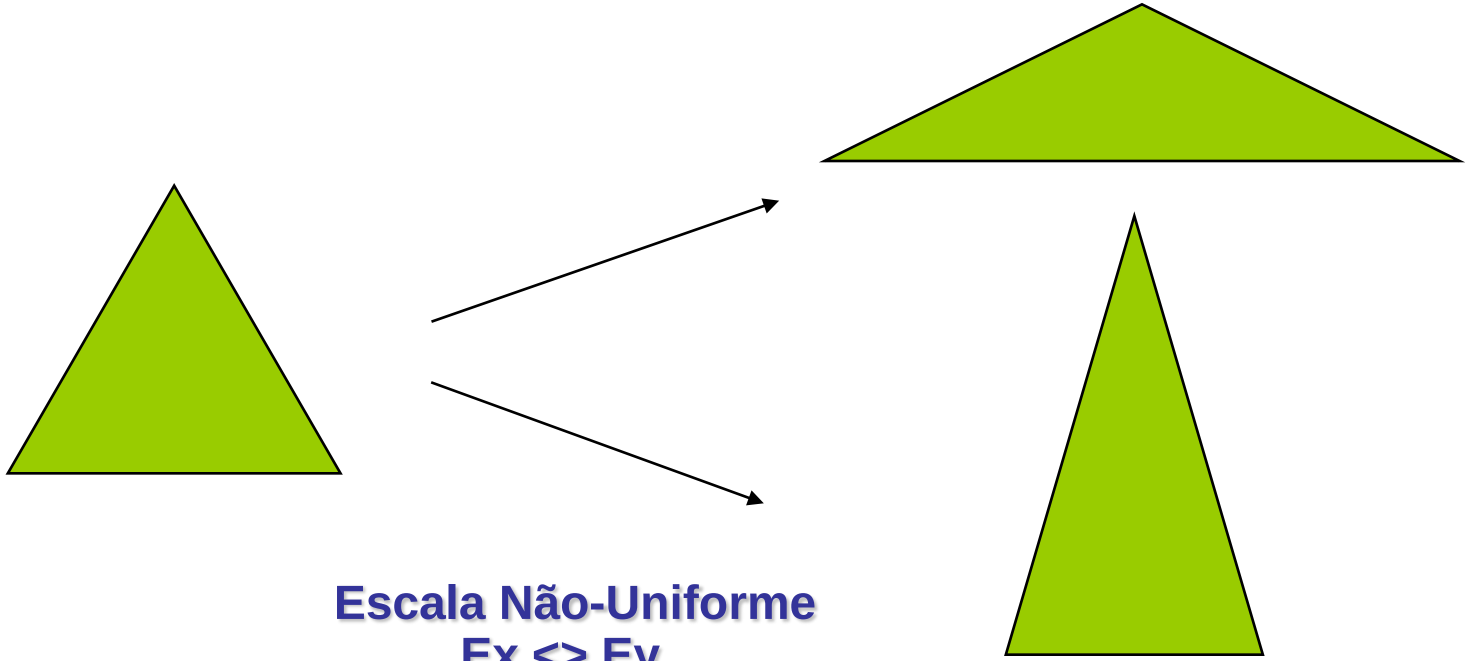


Escala Uniforme
Ex = Ey

Transformações Geométricas 2D

Escala

- A operação de escala muda as dimensões de um polígono.



Escala Não-Uniforme
Ex <> Ey

Transformações Geométricas 2D

Escala

$$xe = x * Ex$$
$$ye = y * Ey$$

Transformações Geométricas 2D

Rotação

(x,y)

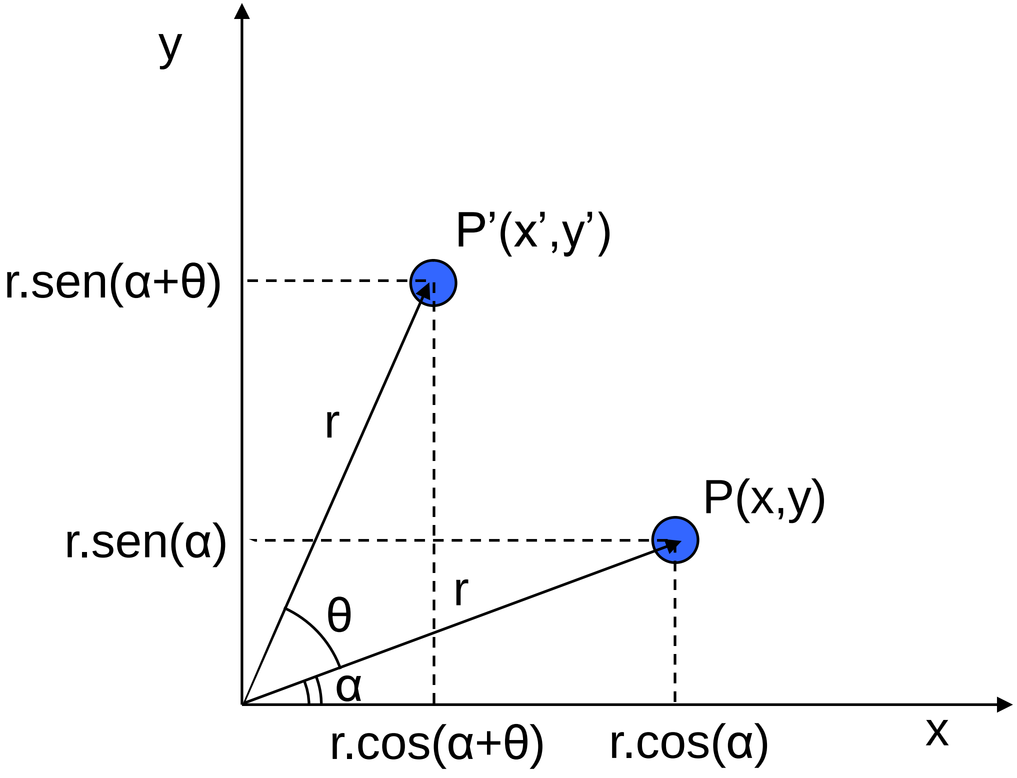
Distante $r=(x^2+y^2)^{1/2}$

$x = r.\cos(\alpha)$

$y = r.\text{sen}(\alpha)$

$x' = r.\cos(\alpha+\theta)$

$y' = r.\text{sen}(\alpha+\theta)$



$x' = r.\cos(\alpha+\theta)$

$y' = r.\text{sen}(\alpha+\theta)$

Transformações Geométricas 2D

Rotação

(x,y)

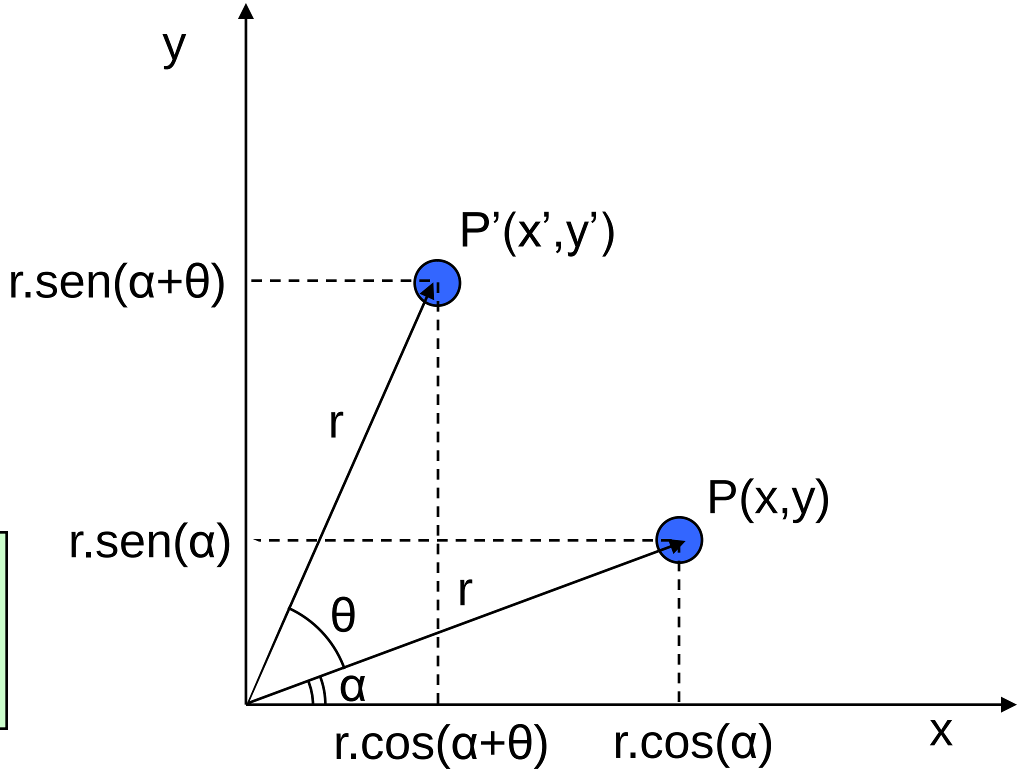
Distante $r=(x^2+y^2)^{1/2}$

$x = r.\cos(\alpha)$

$y = r.\text{sen}(\alpha)$

$x' = r.\cos(\alpha+\theta)$

$y' = r.\text{sen}(\alpha+\theta)$



$x' = r.\cos(\alpha).\cos(\theta) – r.\text{sen}(\alpha).\text{sen}(\theta)$

$y' = r.\text{sen}(\alpha).\cos(\theta) + r.\cos(\alpha).\text{sen}(\theta)$

Transformações Geométricas 2D

Rotação

(x,y)

Distante $r=(x^2+y^2)^{1/2}$

$x = r.\cos(\alpha)$

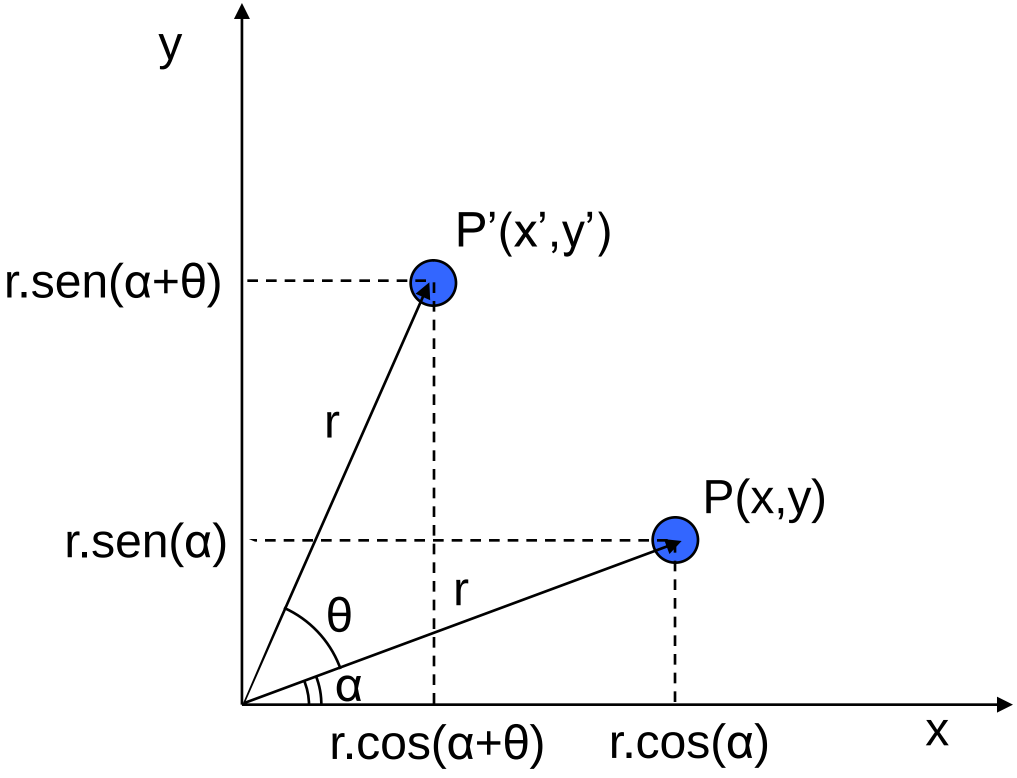
$y = r.\mathrm{sen}(\alpha)$

$x' = r.\cos(\alpha+\theta)$

$y' = r.\mathrm{sen}(\alpha+\theta)$

$x' = x.\cos(\theta) - y.\mathrm{sen}(\theta)$

$y' = y.\cos(\theta) + x.\mathrm{sen}(\theta)$



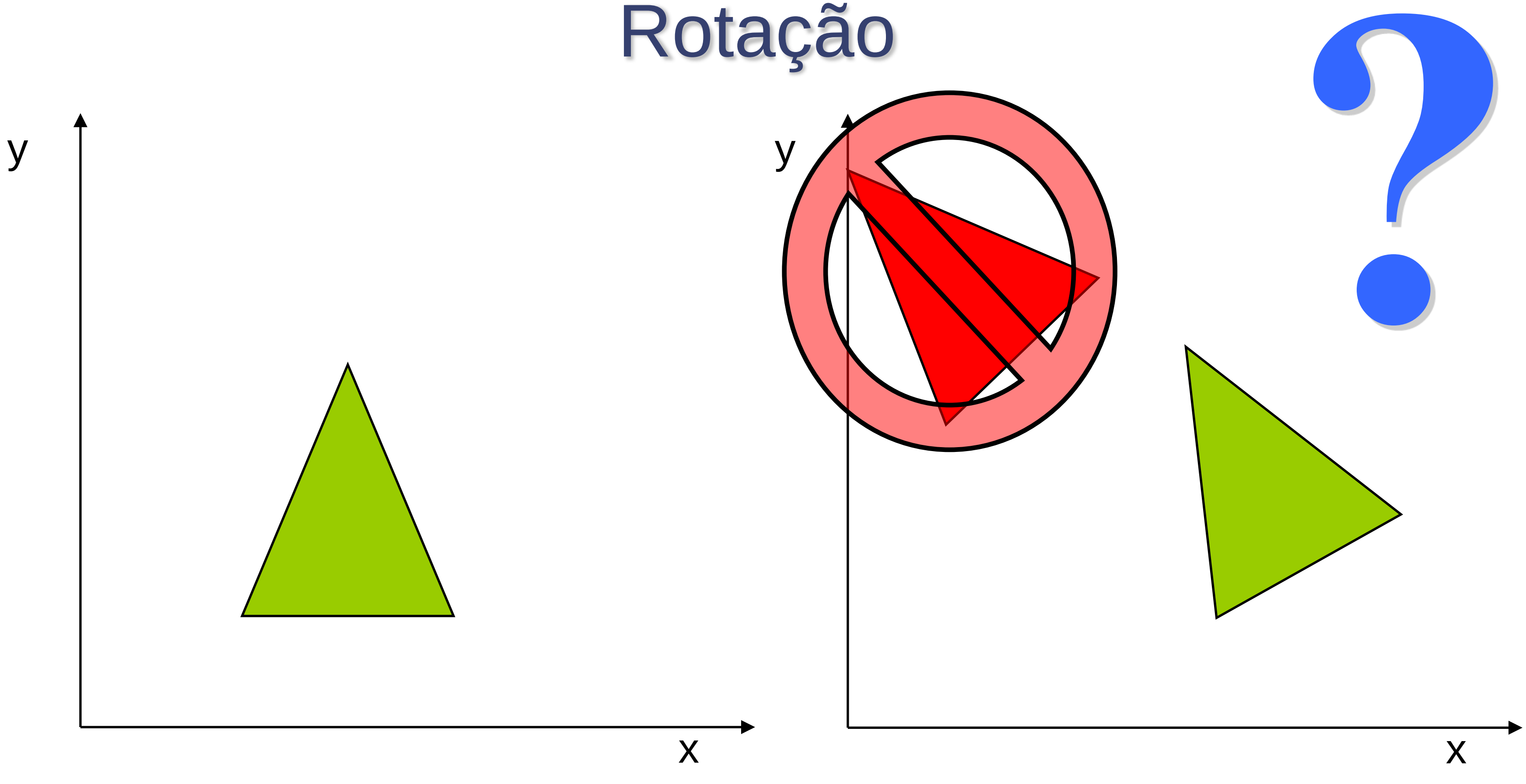
Transformações Geométricas 2D

Rotação

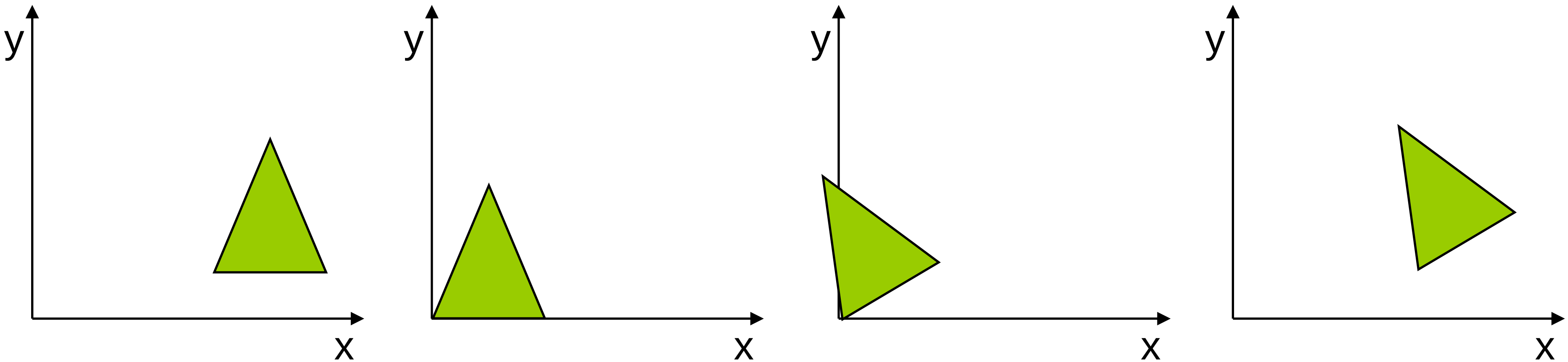
$$xr = x.\cos(\theta) - y.\mathrm{sen}(\theta)$$
$$yr = y.\cos(\theta) + x.\mathrm{sen}(\theta)$$

Transformações Geométricas 2D

Rotação



Transformações Geométricas 2D

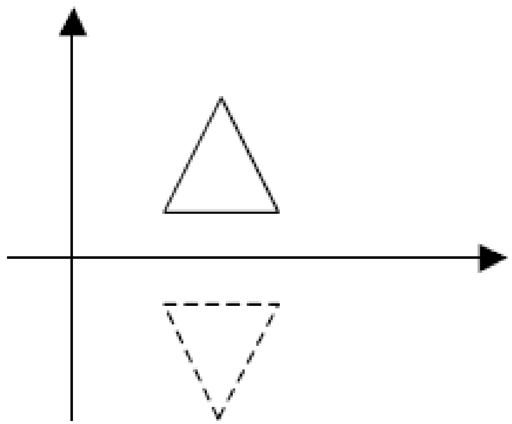


Objeto Original | **Translada de P1 até a origem** | **Rotaciona** | **Translada da origem à posição original**

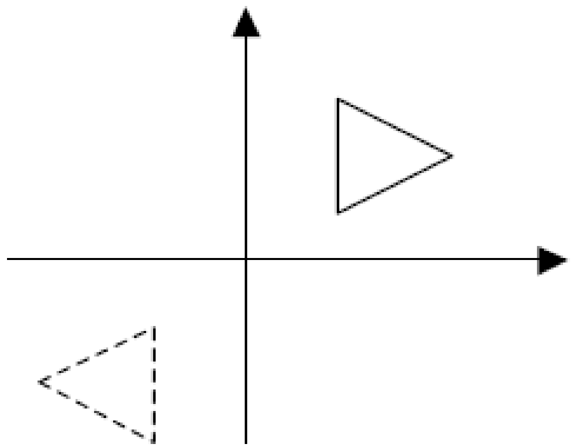
$$x' = (x - xp).\cos(\theta) - (y - yp).\sin(\theta) + xp$$
$$y' = (x - xp).\sin(\theta) + (y - yp).\cos(\theta) + yp$$

Transformações Geométricas 2D

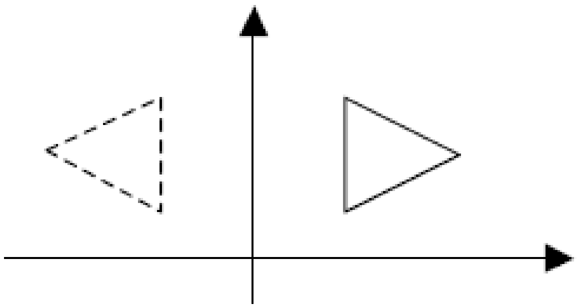
Reflexão



Eixo X
$x' = x$
$y' = -y$



Origem
$x' = -x$
$y' = -y$

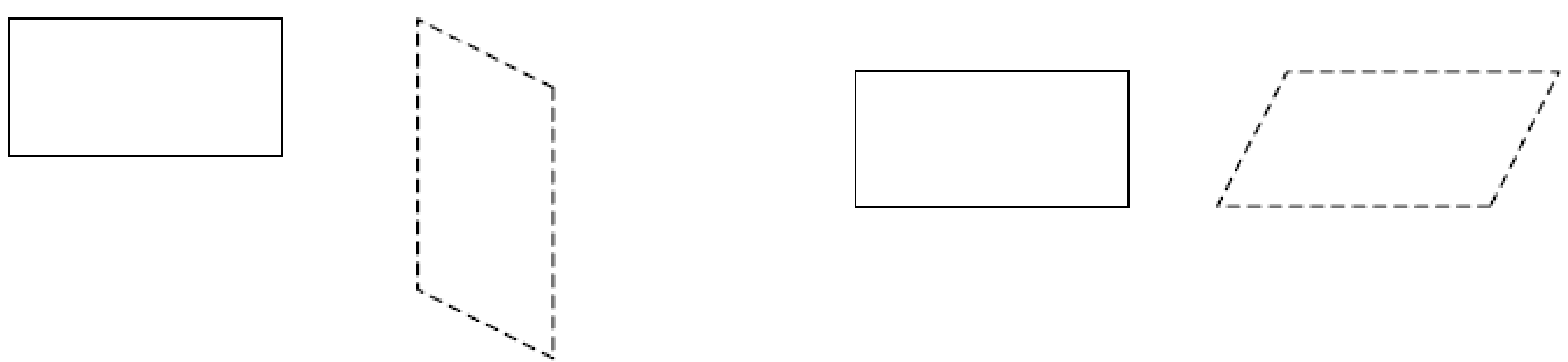


Eixo Y
$x' = -x$
$y' = y$

Transformações Geométricas 2D

Deslizamento (Shearing)

xs = x + Sx
ys = y + Sy



Transformações Geométricas 2D

Translação

xt = x + Tx
yt = y + Ty

Escala

xe = x * Ex
ye = y * Ey

Rotação

xr = x.cos(θ) - y.sen(θ)
yr = y.cos(θ) + x.sen(θ)

Forma Matricial

Translação

$$P' = P + T =$$
$$\begin{bmatrix} x & y \end{bmatrix} + \begin{bmatrix} Tx & Ty \end{bmatrix}$$

Escala

$$P' = P \cdot S =$$
$$\begin{bmatrix} x & y \end{bmatrix} \cdot \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

Rotação

$$P' = P \cdot S =$$
$$\begin{bmatrix} x & y \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

Coordenadas Homogêneas

- Adiciona uma terceira coordenada: w
- O ponto 2D vira um vetor com 3 coordenadas: $\begin{bmatrix} x \\ y \\ w \end{bmatrix}$

- Homogeneizar: dividir x, y e w por w, sendo w = 1.
- Pontos homogeneizados: $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

Transformações 2D Homogêneas

$$T(Tx, Ty) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Tx & Ty & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$R(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$E(Ex, Ey) = \begin{bmatrix} Ex & 0 & 0 \\ 0 & Ey & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Composição de Transformações

- Para realizar uma composição de transformações, basta efetuar uma multiplicação de matrizes.
  - Ex.: Composição de uma rotação com uma translação

    M = R.T

- Multiplicação das matrizes não é comutativa:

  A ordem das operações influencia diretamente.
  - Rotação seguida de translação é diferente de translação seguida de rotação.

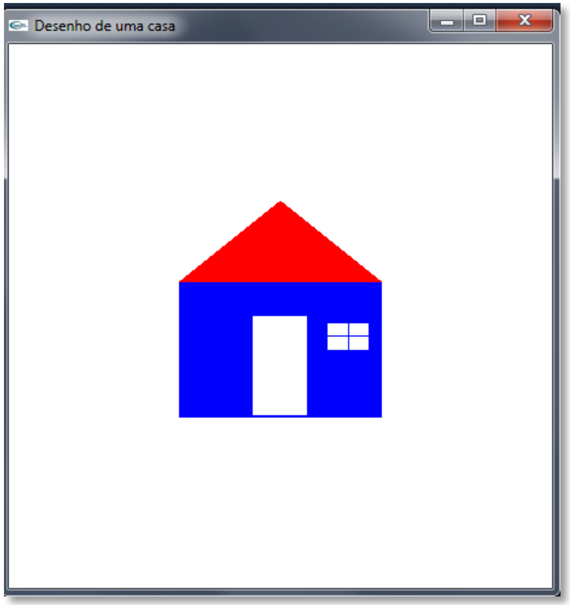
E como programar isso?

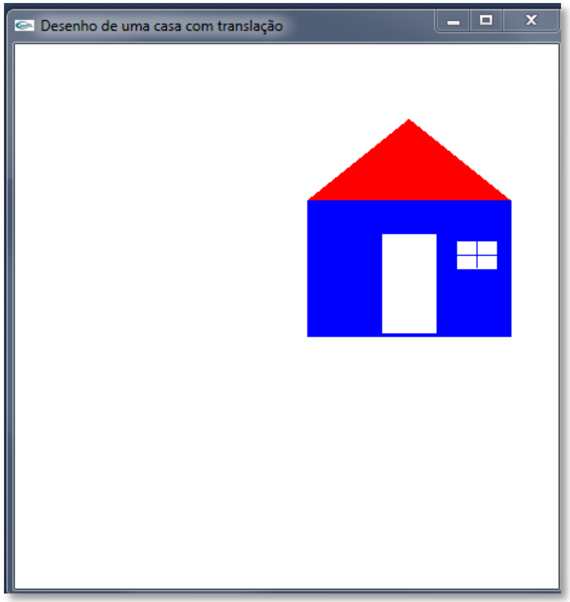
- Por meio de matrizes, diretamente ☺

- OpenGL possui rotinas para cada transformação
  - Ela também usa matrizes
  - É necessário entender como funciona

Translação na OpenGL


```
// Aplica uma translação sobre a casinha que será desenhada
glTranslatef(18.0f, 12.0f, 0.0f);
```

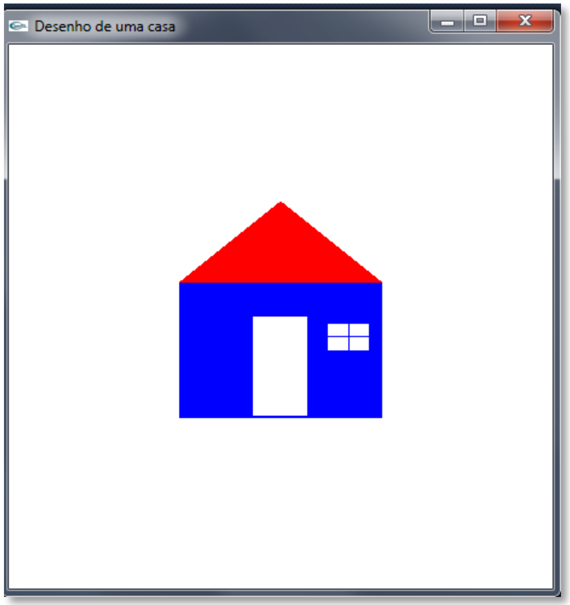


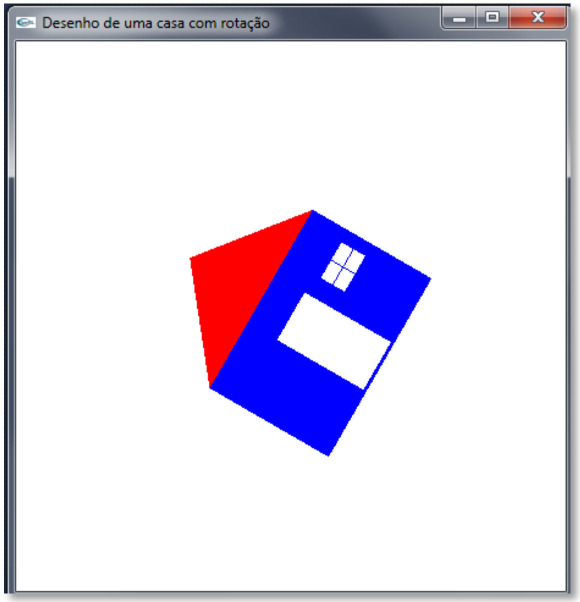


A **translação** é feita através da função *glTranslatef(Tx, Ty, Tz)*, que pode receber três números *float* ou *double* (*glTranslated*) como parâmetro. Neste caso, a matriz atual é multiplicada por uma matriz de translação baseada nos valores dados.

Rotação na OpenGL


```
// Aplica uma rotação sobre a casinha que será desenhada
glRotatef(60.0f, 0.0f, 0.0f, 1.0f);
```

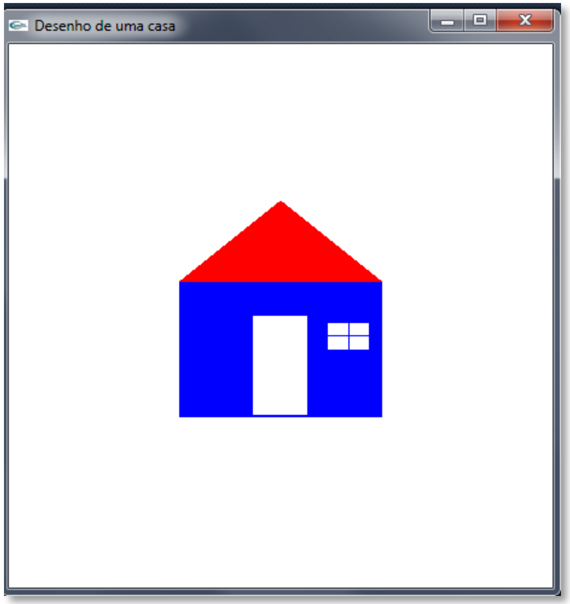


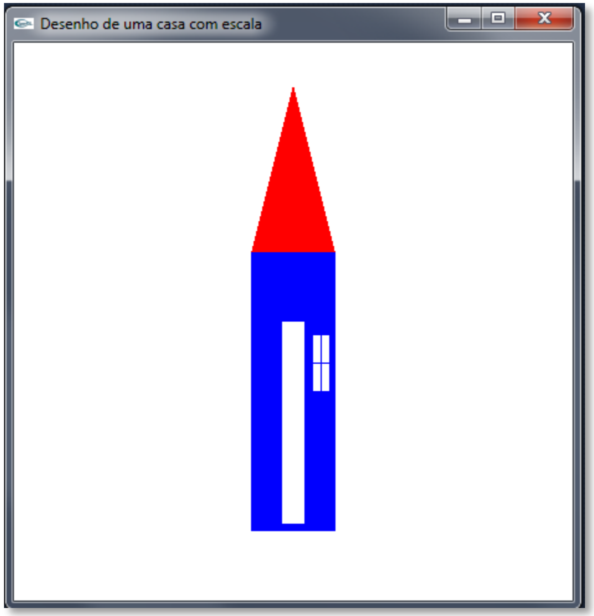


A **rotação** é feita através da função *glRotatef(Ângulo, x, y, z)*, que pode receber quatro números *float* ou *double* (*glRotated*) como parâmetro. Neste caso, a matriz atual é multiplicada por uma matriz de rotação de "Ângulo" graus ao redor do eixo definido pelo vetor "x,y,z" no sentido anti-horário.

Escala na OpenGL


```
// Aplica uma escala sobre a casinha que será desenhada
glScalef(0.4f, 2.0f, 1.0f);
```





A **escala** é feita através da função *glScalef(Ex, Ey, Ez)*, que pode receber três números *float* ou *double* (*glScaled*) como parâmetro. Neste caso, a matriz atual é multiplicada por uma matriz de escala baseada nos valores dados.

Matrizes de Transformação

- Para possibilitar a combinação das transformações geométricas, de maneira a reduzir a quantidade de operações matemáticas a serem aplicadas a cada vértice do modelo, **os cálculos são implementados utilizando matrizes com coordenadas homogêneas**.

- Tanto os comandos de visualização como as transformações geométricas são executados por meio de operações por matrizes.
  - Ou seja, o que vimos antes ☺

Matrizes de Transformação

- Para isso, precisamos chamar as funções que especificam e inicializam as matrizes.

- glMatrixMode
  - Seleciona a matriz a ser utilizada. O parâmetro *mode* deve ser uma constante que pode receber um dos seguintes valores:
    - GL_MODELVIEW
    - GL_PROJECTION
    - GL_TEXTURE (para selecionar a matriz de textura)

glMatrixMode(GL_PROJECTION); e ***glLoadIdentity();*** servem, respectivamente, para avisar a OpenGL que todas as futuras alterações, tais como operações de escala, rotação e translação, irão afetar a "câmera" (ou observador), e para inicializar o sistema de coordenadas antes da execução de qualquer operação de manipulação de matrizes. Sem este comando, cada chamada sucessiva de *gluOrtho2D* poderia resultar em uma corrupção do volume de visualização. Em outras palavras, a matriz de projeção é onde o volume de visualização, que neste caso é um plano, é definido; a função *gluOrtho2D* não estabelece realmente o volume de visualização utilizado para fazer o recorte, apenas modifica o volume existente; ela multiplica a matriz que descreve o volume de visualização corrente pela matriz que descreve o novo volume de visualização, cujas coordenadas são recebidas por parâmetro.

glMatrixMode(GL_MODELVIEW); avisa a OpenGL que todas as futuras alterações, tais como operações de escala, rotação e translação, irão afetar os modelos da cena, ou em outras palavras, o que é desenhado. A função *glLoadIdentity();* chamada em seguida, faz com que a matriz corrente seja inicializada com a matriz identidade (nenhuma transformação é acumulada)

Matrizes de Transformação

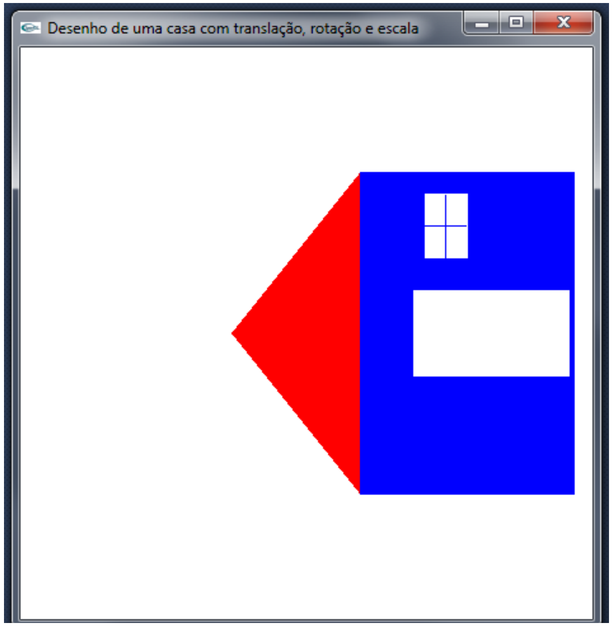
- glLoadIdentity()
  - Faz com que a matriz de transformação corrente seja inicializada com a **matriz identidade**, indicando que nenhuma transformação foi aplicada

Matrizes de Transformação

- Exemplo

```
// Muda para o sistema de coordenadas do modelo
glMatrixMode(GL_MODELVIEW);
// Inicializa a matriz de transformação corrente
glLoadIdentity();

(...)
// Aplica uma translação sobre a casinha que será desenhada
glTranslatef(15.0f, 0.0f, 0.0f);
// Aplica uma rotação sobre a casinha que será desenhada
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
// Aplica uma escala sobre a casinha que será desenhada
glScalef(1.5f, 1.5f, 1.0f);
```

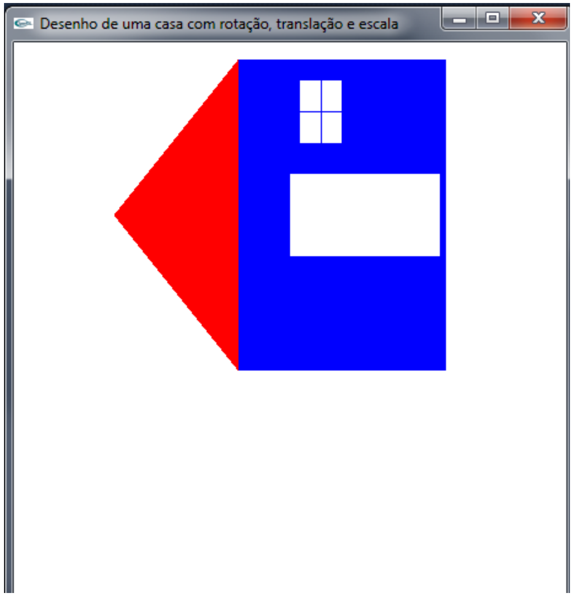


Matrizes de Transformação

- A ordem dos fatores altera o produto!!!

```
// Muda para o sistema de coordenadas do modelo
glMatrixMode(GL_MODELVIEW);
// Inicializa a matriz de transformação corrente
glLoadIdentity();

(...)
// Aplica uma rotação sobre a casinha que será desenhada
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
// Aplica uma translação sobre a casinha que será desenhada
glTranslatef(15.0f, 0.0f, 0.0f);
// Aplica uma escala sobre a casinha que será desenhada
glScalef(1.5f, 1.5f, 1.0f);
```



Escopo das Transformações

- Para restringir o escopo das transformações geométricas, OpenGL usa **pilhas de matrizes de transformação**.
  - glPushMatrix(void)
    - Esta função, que não possui argumentos, é utilizada para guardar a matriz de transformação corrente na pilha
  - glPopMatrix(void)
    - Esta função, que também não possui argumentos, é utilizada para recuperar a matriz de transformação corrente da pilha.

Escopo das Transformações

- Exemplo

```
// Guarda a matriz de transformação corrente na pilha
glPushMatrix();

// Aplica uma translação
glTranslatef(10.0f, 0.0f, 0.0f);
// Altera a cor do desenho para azul
glColor3f(0.0f, 0.0f, 1.0f);
// Desenha a casa
glBegin(GL_QUADS);
        glVertex2f(-15.0f,-15.0f);
        glVertex2f(-15.0f,  5.0f);
        glVertex2f( 15.0f,  5.0f);
        glVertex2f( 15.0f,-15.0f);
glEnd();
```

Escopo das Transformações


```
// Guarda a matriz de transformação corrente na pilha
glPushMatrix();

// Aplica uma rotação
glRotatef(30.0f, 0.0f, 0.0f, 1.0f);
// Altera a cor do desenho para cinza
glColor3f(0.7f, 0.7f, 0.7f);
// Desenha a porta e a janela
glBegin(GL_QUADS);
        glVertex2f(-4.0f,-14.5f);
        glVertex2f(-4.0f,  0.0f);
        glVertex2f( 4.0f,  0.0f);
        glVertex2f( 4.0f,-14.5f);
        glVertex2f( 7.0f,-5.0f);
        glVertex2f( 7.0f,-1.0f);
        glVertex2f(13.0f,-1.0f);
        glVertex2f(13.0f,-5.0f);
glEnd();
```

Escopo das Transformações

- Exemplo (continuação)

```
// Restaura a matriz de transformação corrente da pilha
glPopMatrix();

// Altera a cor do desenho para branco
glColor3f(1.0f, 1.0f, 1.0f);
// Desenha as "linhas" da janela
glBegin(GL_LINES);
        glVertex2f( 7.0f,-3.0f);
        glVertex2f(13.0f,-3.0f);
        glVertex2f(10.0f,-1.0f);
        glVertex2f(10.0f,-5.0f);
glEnd();
```

Escopo das Transformações

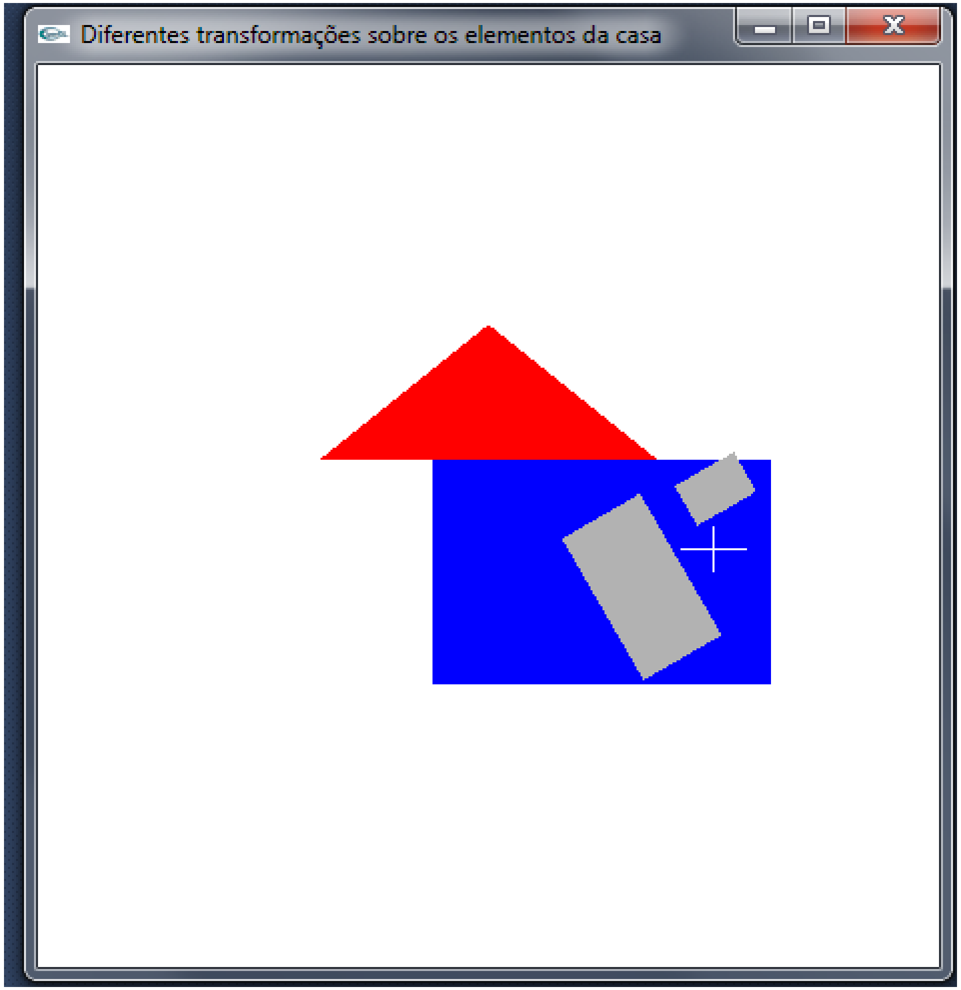
- Exemplo (continuação)

```
// Restaura a matriz de transformação corrente da pilha
glPopMatrix();

// Altera a cor do desenho para vermelho
glColor3f(1.0f, 0.0f, 0.0f);
// Desenha o telhado
glBegin(GL_TRIANGLES);
            glVertex2f(-15.0f, 5.0f);
            glVertex2f(  0.0f,17.0f);
            glVertex2f( 15.0f, 5.0f);
glEnd();
```

Escopo das Transformações

- Exemplo



Exercício

- Mobilie uma casa usando a OpenGL
  - Sugestão: Desenhe uma grid para facilitar a visualização. Quando estiver projetando, você pode desenhar em uma folha quadriculada para facilitar a visualização.
  - Desenhe a planta baixa usando primitivas GL_LINE ou GL_LINE_LOOP
    - No mínimo 5 cômodos
  - Crie representações para móveis básicos (cama, mesa, sofá, cadeira, armário, etc) usando primitivas. No mínimo 10 móveis diferentes.
    - Crie funções de desenho para cada uma
      - DesenhaMesa
      - DesenhaCamaCasal
      - DesenhaCamaSolteiro
  - Você deve mobiliar a casa chamando as rotinas de desenho dos móveis básicos, precedidas das transformações geométricas necessárias para que os móveis fiquem no lugar desejado
  - No mínimo (mínimo meeesmo) 25 móveis na casa
  - Necessário haver todas as transformações (translação, rotação, escala) pelo menos 3 vezes.
  - Link para inspirar
    - http://casa.abril.com.br/decorar/brinque/

Exemplo...

- Não precisa imagens, só primitivas
- Pode acumular transformações
- Entregar até semana que vem!
  - **<u>03/04/2014</u>**