

Transformações Geométricas 2D



Professor: Odilon Corrêa da Silva

Curso: Engenharia de Computação

Disciplina: Computação Gráfica

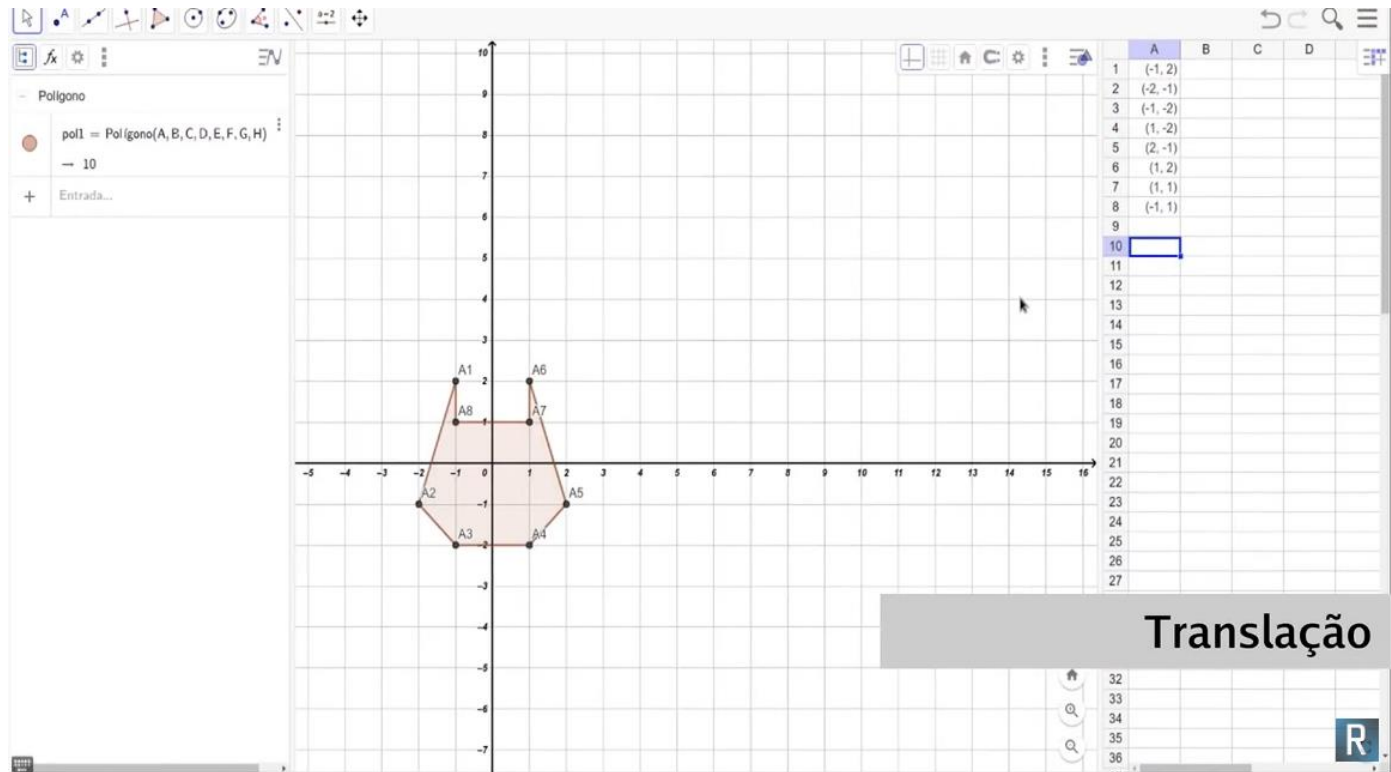
Transformações Geométricas 2D

- Transformações Geométricas

- As transformações geométricas são usadas para manipular um modelo, isto é, através delas é possível mover, rotacionar ou alterar a escala de um objeto.
- A habilidade de representar um objeto em várias posições no espaço é fundamental para compreender sua forma. A possibilidade de submetê-lo a diversas transformações é importante em diversas aplicações da computação gráfica (AZEVEDO e CONCI, 2003).

Transformações Geométricas 2D

- Transformações Geométricas
 - Transformação de Translação
 - Transladar significa movimentar o objeto.



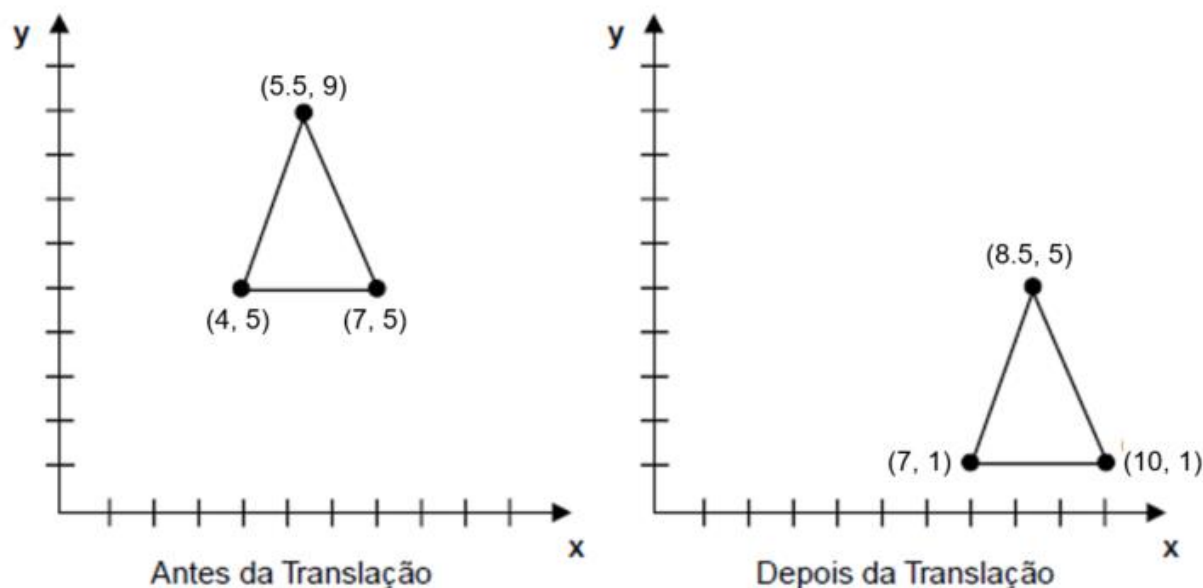
Fonte: https://youtu.be/XxXXhI6f8Es?list=PLvat2X-KHJNbL_JsbO_qiDP7863-dgzee&t=182

Transformações Geométricas 2D

- Transformações Geométricas

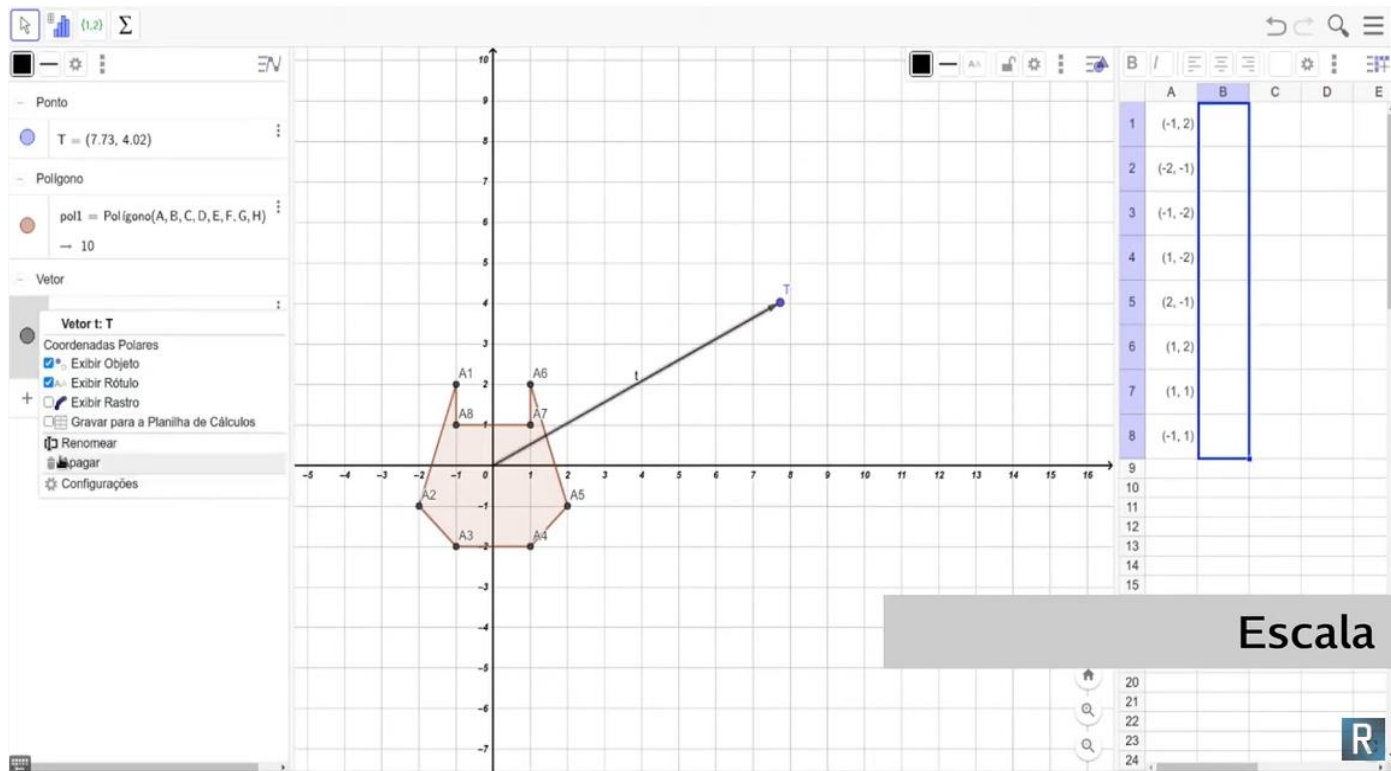
- Transformação de Translação

- É possível efetuar a translação de pontos no plano (x,y) adicionando quantidades às suas coordenadas. Assim, cada ponto em (x,y) pode ser movido por T_x unidades em relação ao eixo x , e por T_y unidades em relação ao eixo y . Logo, a nova posição do ponto (x,y) passa a ser (x',y') .



Transformações Geométricas 2D

- Transformações Geométricas
 - Transformação de Escala
 - Escalonar significa mudar as dimensões de escala



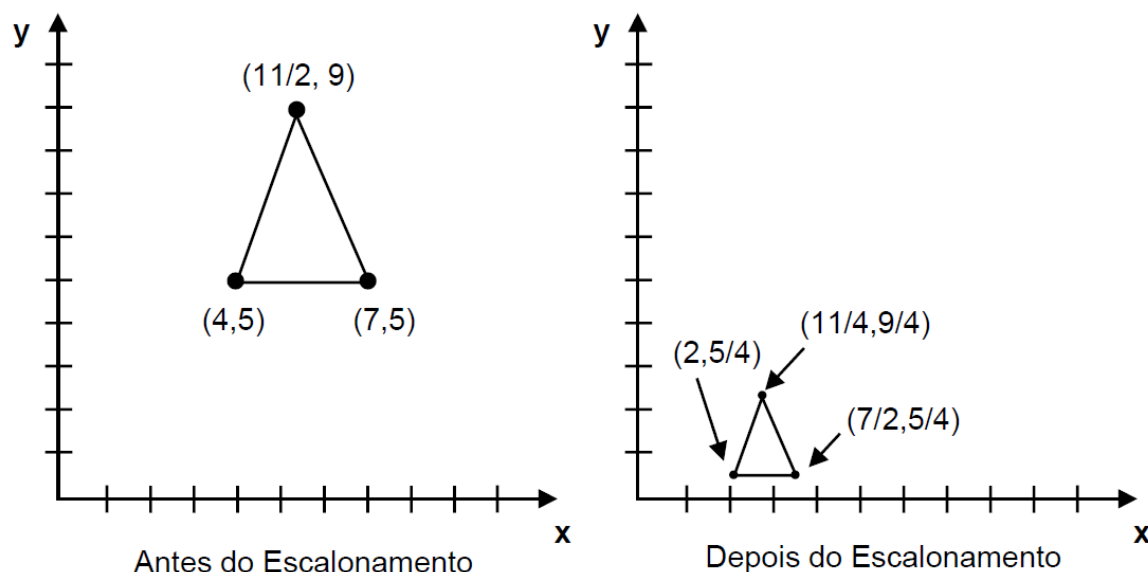
Fonte: https://youtu.be/XxXXhI6f8Es?list=PLvat2X-KHJNbL_JsbO_qiDP7863-dgzee&t=285

Transformações Geométricas 2D

- Transformações Geométricas

- Transformação de Escala

- Para fazer com que uma imagem definida por um conjunto de pontos mude de tamanho, teremos de multiplicar os valores de suas coordenadas por um fator de escala.

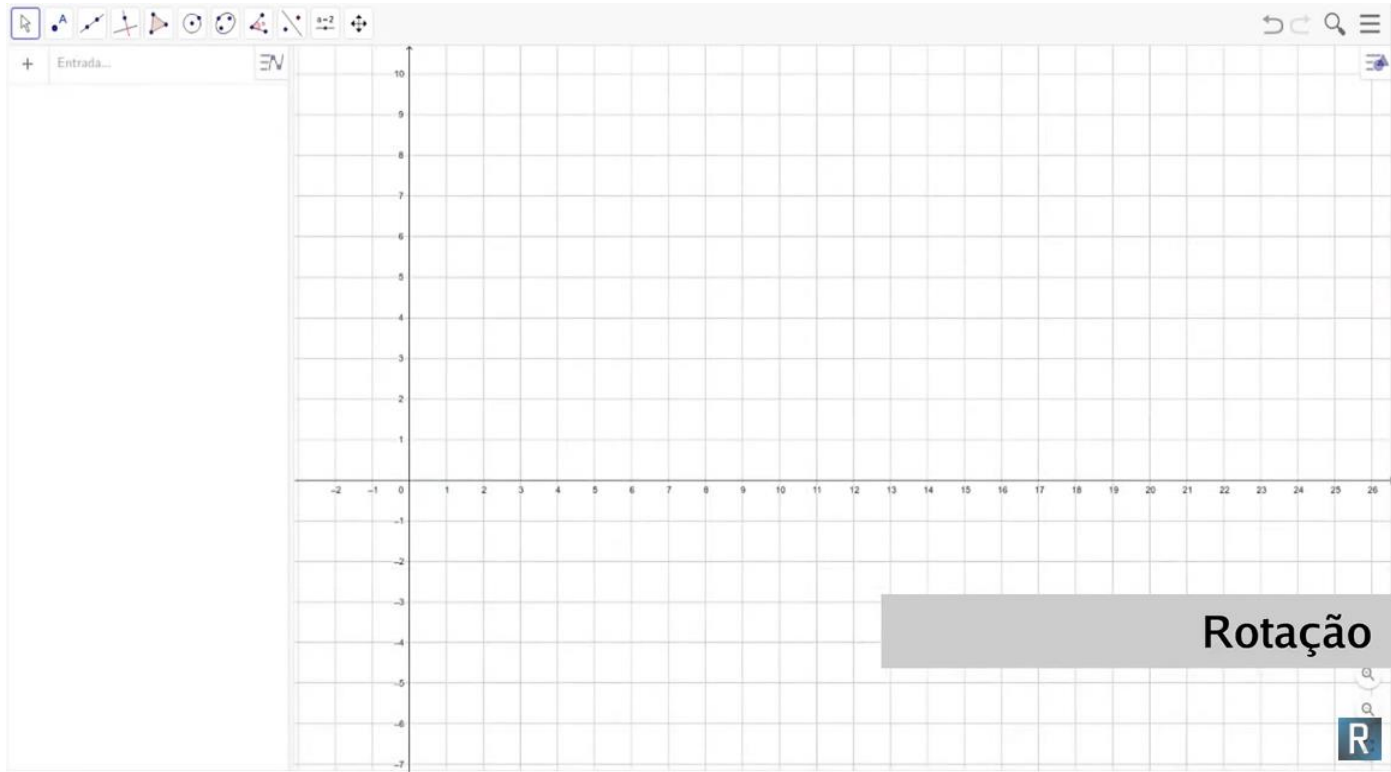


Fonte: Azevedo e Conci (2003, p. 40).

- A mesma figura antes e depois de uma mudança de escala genérica, de $1/2$ na horizontal e $1/4$ na vertical

Transformações Geométricas 2D

- Transformações Geométricas
 - Transformação de Rotação
 - Rotacionar significa girar



Fonte: https://youtu.be/XxXXhI6f8Es?list=PLvat2X-KHJNbL_JsbO_qiDP7863-dgzee&t=841

Transformações Geométricas 2D

- Transformações Geométricas

- Transformação de Rotação

- Se um ponto de coordenada (x,y) , distante $r=(x^2+y^2)^{1/2}$ da origem do sistema de coordenadas, for rotacionado de um ângulo θ em torno da origem, suas coordenadas, que antes eram definidas como: $x = r * \cos(\Phi)$, $y = r * \sin(\Phi)$, passam a ser descritas como (x', y') dadas por:

$$x' = x \cos(\theta) - y \sin(\theta)$$

$$y' = y \cos(\theta) + x \sin(\theta)$$

- Essas expressões podem ser descritas pela multiplicação do vetor de coordenadas do ponto $(x \ y)$ pela matriz:

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

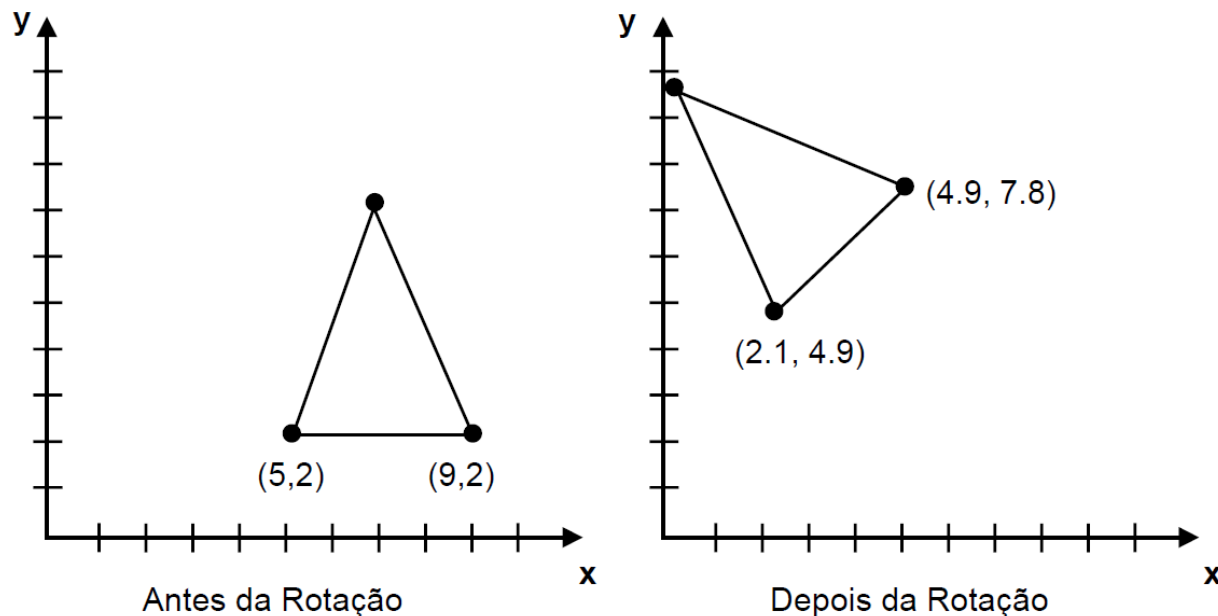
- Essa matriz é denominada matriz de rotação no plano xy por um ângulo .

Transformações Geométricas 2D

- Transformações Geométricas

- Transformação de Rotação

- No caso de o objeto não estar definido na origem do sistema de coordenadas, a multiplicação de suas coordenadas por uma matriz de rotação também resulta em uma translação.



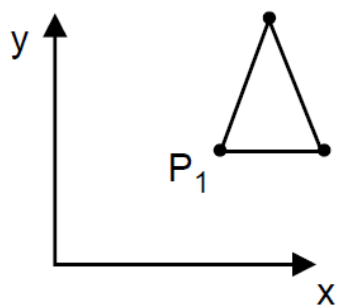
Fonte: Azevedo e Conci (2003, p. 43).

Transformações Geométricas 2D

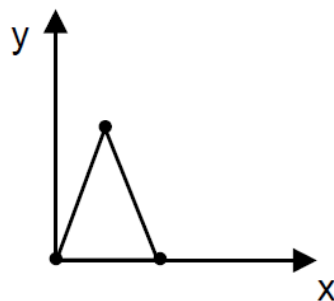
- Transformações Geométricas

- Transformação de Rotação

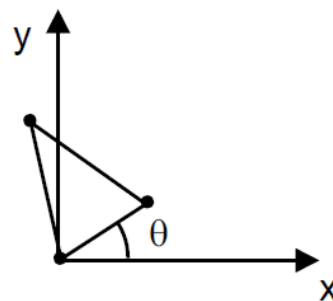
- Para uma rotação correta, deve-se combinar a transformação de rotação com a transformação de translação. **O mesmo procedimento pode ser utilizado com a transformação de escala.** Antes de aplicar a rotação de um ângulo θ no plano das coordenadas em torno de um ponto, realizar uma translação para localizar esse ponto na origem do sistema, aplicando a rotação desejada e, então, uma translação inversa para localizar o dado ponto na origem



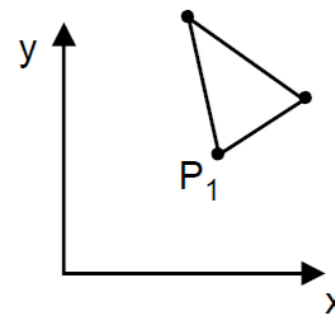
Objeto Original



Depois da Translação de P_1 à origem



Após Rotação



Após Translação que retorna a posição original

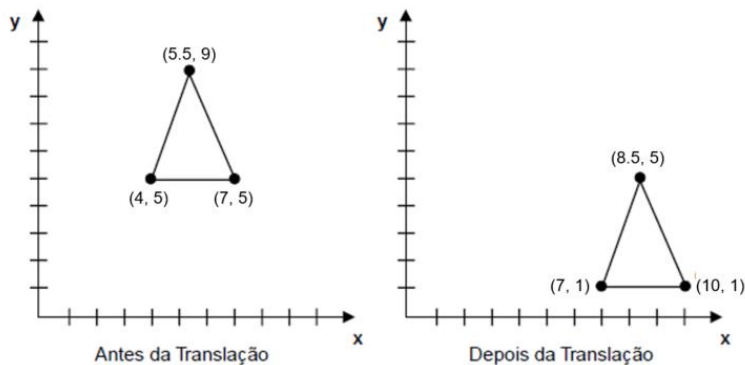
Transformações Geométricas 2D

- Transformações Geométricas no OpenGL
 - A biblioteca gráfica OpenGL é capaz de executar transformações de translação, escala e rotação através de uma multiplicação de matrizes.
 - 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.
 - 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.
 - 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.

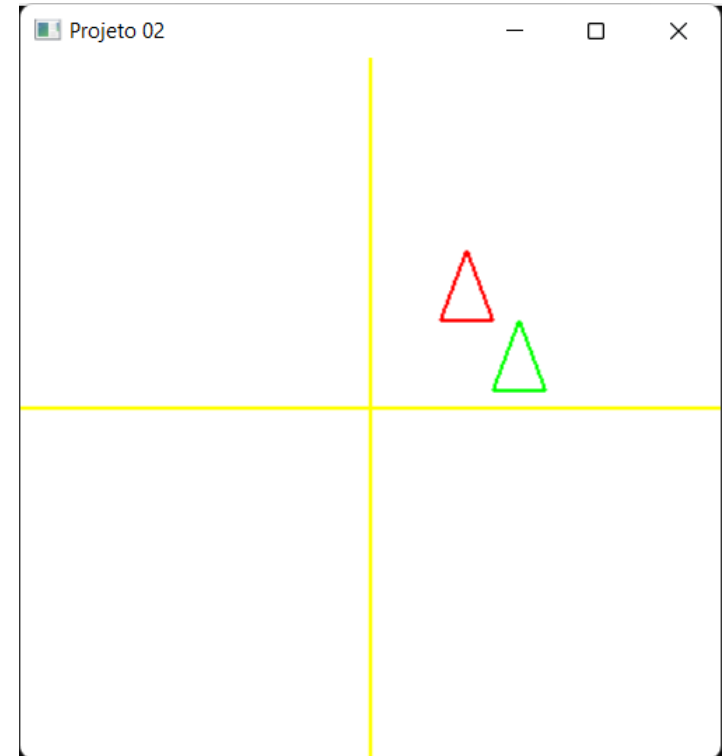
Transformações Geométricas 2D

- Transformações Geométricas no OpenGL

- Exemplos



```
void DesenhaTriangulo() {  
    glBegin(GL_LINE_LOOP);  
        glVertex2f(4.0, 5.0);  
        glVertex2f(7.0, 5.0);  
        glVertex2f(5.5, 9.0);  
    glEnd();  
}  
  
void Desenha(void) {  
    glColor3f(1.0, 0.0, 0.0); //vermelho em RGB  
    DesenhaTriangulo();  
  
    glTranslatef(3.0, -4.0, 0.0f);  
    glColor3f(0.0f, 1.0f, 0.0f); //verde em RGB  
  
    DesenhaTriangulo();  
}
```

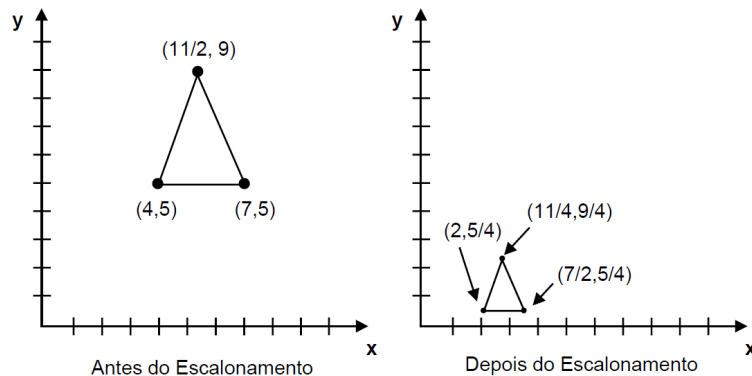


Projeto01: Translação

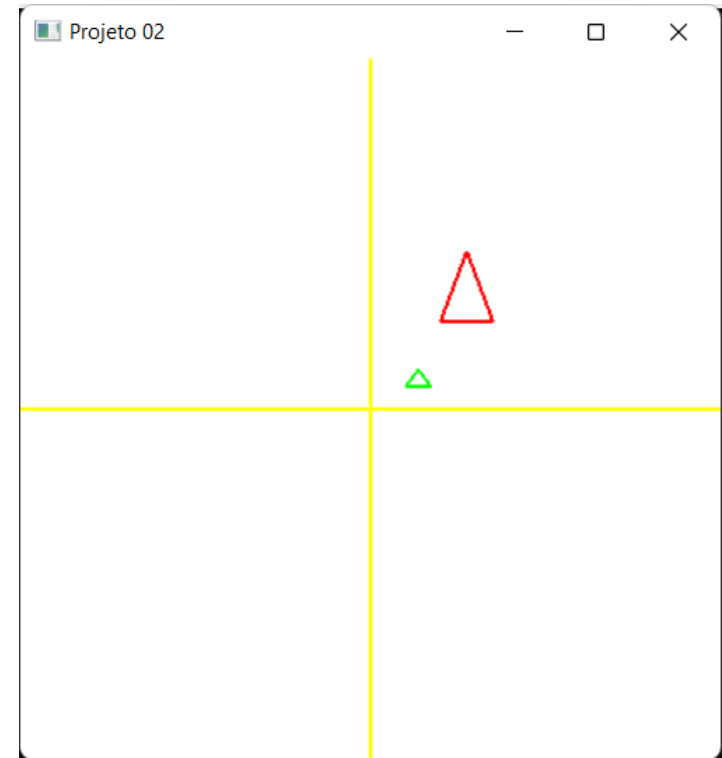
Transformações Geométricas 2D

- Transformações Geométricas no OpenGL

- Exemplos



```
void DesenhaTriangulo() {  
    glBegin(GL_LINE_LOOP);  
        glVertex2f(4.0, 5.0);  
        glVertex2f(7.0, 5.0);  
        glVertex2f(5.5, 9.0);  
    glEnd();  
}  
  
void Desenha(void) {  
    glColor3f(1.0, 0.0, 0.0); //vermelho em RGB  
    DesenhaTriangulo();  
  
    glScalef(0.5, 0.25, 0.0f);  
    glColor3f(0.0f, 1.0f, 0.0f); //verde em RGB  
    DesenhaTriangulo();  
}
```

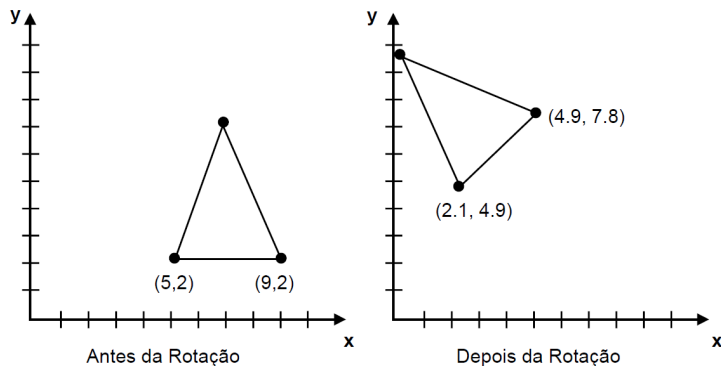


Projeto01: Escala

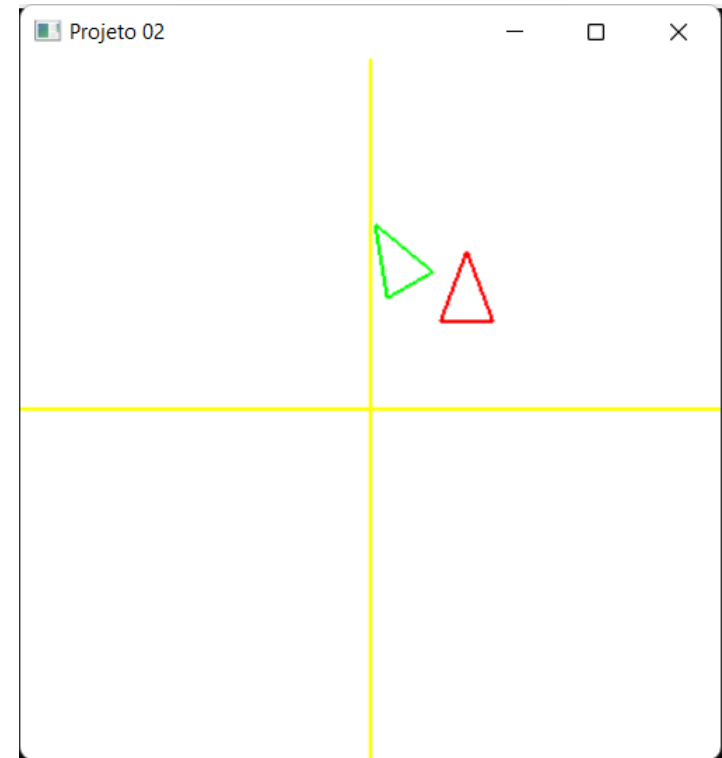
Transformações Geométricas 2D

- Transformações Geométricas no OpenGL

- Exemplos



```
void DesenhaTriangulo() {  
    glBegin(GL_LINE_LOOP);  
        glVertex2f(4.0, 5.0);  
        glVertex2f(7.0, 5.0);  
        glVertex2f(5.5, 9.0);  
    glEnd();  
}  
  
void Desenha(void) {  
    glColor3f(1.0, 0.0, 0.0); //vermelho em RGB  
    DesenhaTriangulo();  
  
    glRotatef(30.0, 0.0, 0.0, 1.0f);  
    glColor3f(0.0f, 1.0f, 0.0f); //verde em RGB  
    DesenhaTriangulo();  
}
```

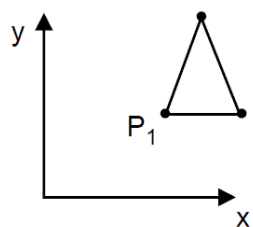


Projeto01: Rotação

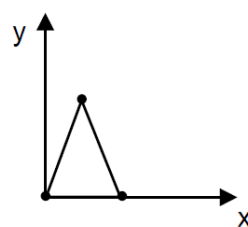
Transformações Geométricas 2D

- Transformações Geométricas no OpenGL

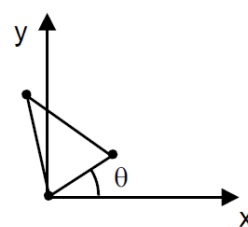
- Exemplo:



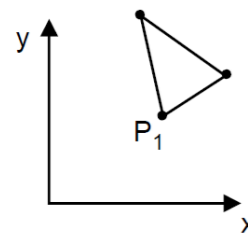
Objeto Original



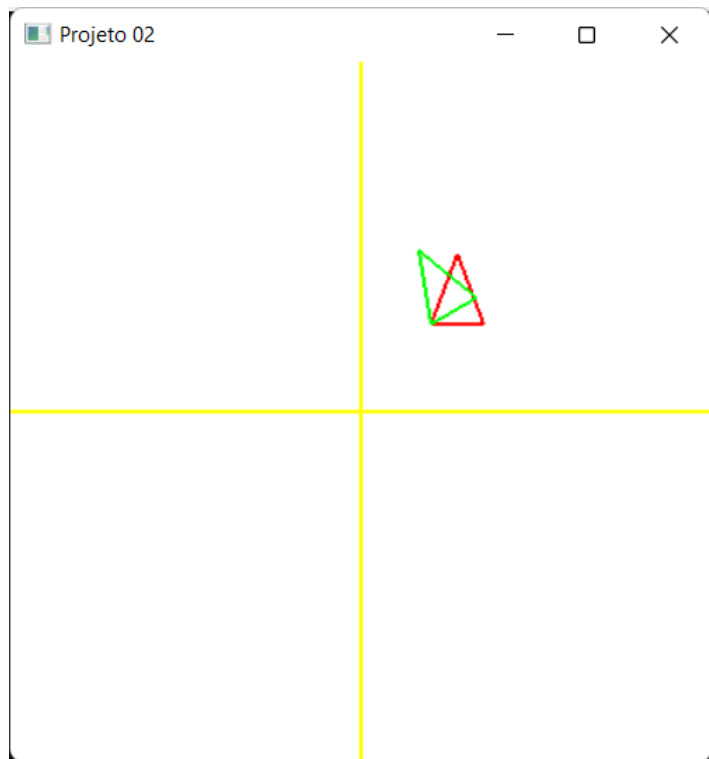
Depois da Translação de P_1 à origem



Após Rotação



Após Translação que retorna a posição original



Projeto01: Rotação correta

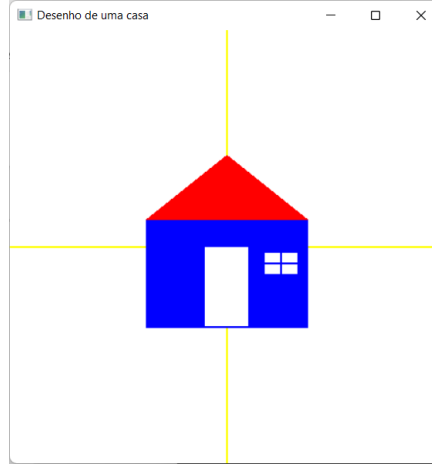
```
void DesenhaTriangulo() {  
    glBegin(GL_LINE_LOOP);  
        glVertex2f(4.0, 5.0);  
        glVertex2f(7.0, 5.0);  
        glVertex2f(5.5, 9.0);  
    glEnd();  
}  
  
void Desenha(void) {  
    glColor3f(1.0, 0.0, 0.0); //vermelho em RGB  
    DesenhaTriangulo();  
  
    glTranslatef(4.0, 5.0, 0.0f);  
    glRotatef(30.0, 0, 0, 1.0f);  
    glTranslatef(-4.0, -5.0, 0.0f);  
  
    glColor3f(0.0f, 1.0f, 0.0f); //verde em RGB  
    DesenhaTriangulo();  
}
```

Empilha
e
desempilha

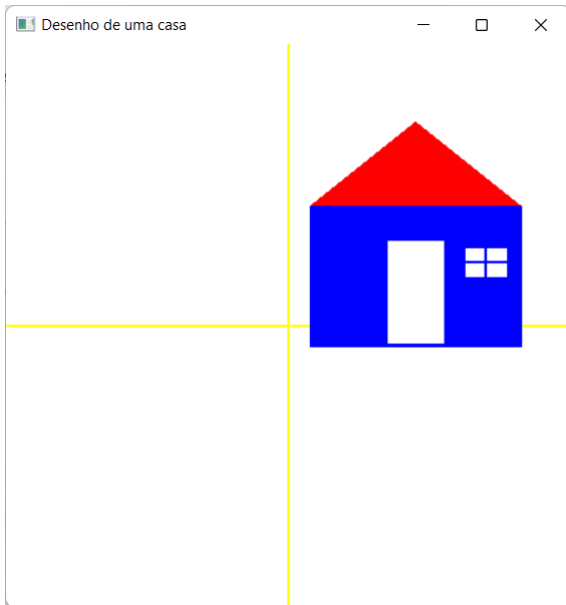
Transformações Geométricas 2D

- Transformações Geométricas no OpenGL

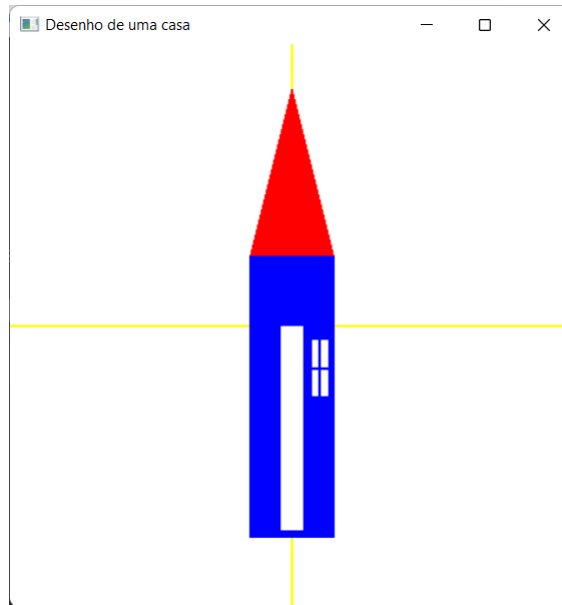
- Exemplos



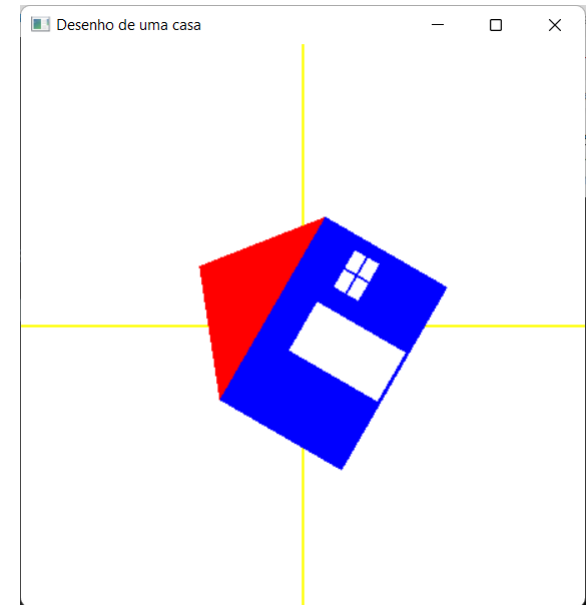
Projeto02



Projeto02: Translação



Projeto02: Escala



Projeto02: Rotação

Transformações Geométricas 2D

- Sugestão de videoaula
 - Transformações Geométricas 2D
 - <https://www.youtube.com/watch?v=XxXXhl6f8Es>
 - Transformações Geométricas 2D no OpenGL - Parte 01
 - <https://www.youtube.com/watch?v=NGwzZ9rzims>
 - Transformações Geométricas 2D no OpenGL - Parte 02
 - <https://www.youtube.com/watch?v=tMxGscrtHVY>
- Sugestão de leitura
 - Capítulo 2 do livro “Computação gráfica: geração de imagens” do livro do Azevedo e Conci
 - Capítulo 9 do livro “OpenGL - Uma Abordagem Prática e Objetiva” do Cohen e Manssour

Transformações Geométricas 2D

- Referência Bibliográfica
 - AZEVEDO, Eduardo; CONCI, Aura. Computação gráfica: geração de imagens. Rio de Janeiro: Campus, Elsevier, 2003
 - COHEN, Marcelo; MANSSOUR, Isabel. OpenGL - Uma Abordagem Prática e Objetiva. São Paulo: Novatec, 2006. 486 p.