Programação Gráfica

Professor Rodrigo Piva

Representação Matricial de Imagens

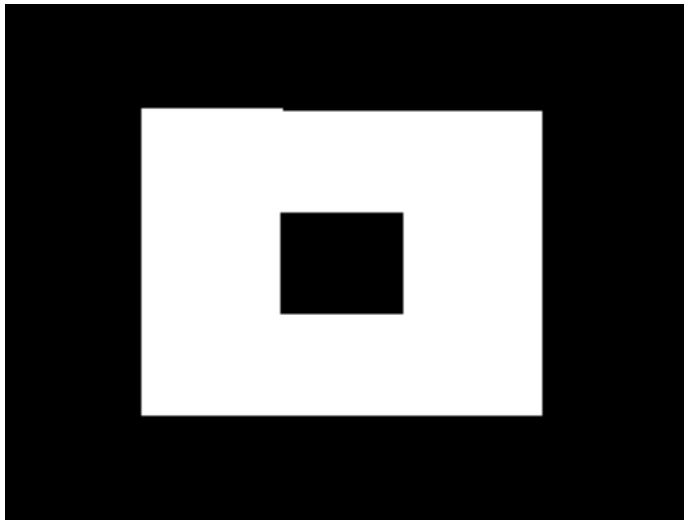
- A imagem é descrita por um conjunto de células em um arranjo espacial bidimensional, uma matriz.
- Cada célula representa um pixel da imagem.
- As imagens são formadas a partir da disposição correta dos pixels.

Representação Matricial de Imagens

- Imagens Matriciais também são conhecidas como bitmaps, ou mapa de bits.
- São utilizadas para formar a imagem na memória e na tela dos computadores.

Representação Matricial de Imagens

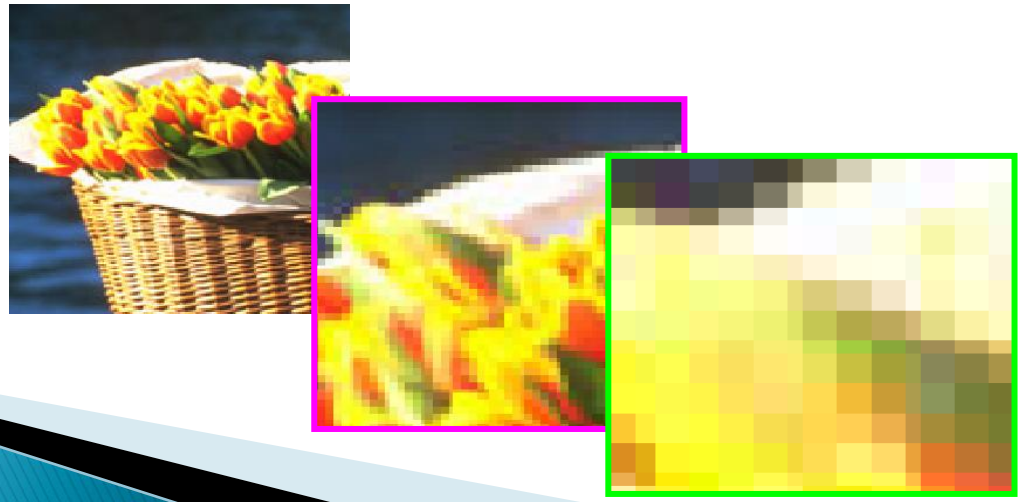
1	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	0	0	0	1
1	1	1	1	1



Pixel

- É o menor elemento em um dispositivo de exibição.
- Ele é indivisível.

- Em monitores coloridos um pixel é formado por 3 pontos: verde, vermelho e azul.



Sistemas de Coordenadas

- A ideia de localizar posições utilizando-se números, vem de longe e é usada em muitas situações da nossa vida.

- Os sistemas de coordenadas servem para nos dar uma referência em termos de medidas do tamanho e posição dos objetos dentro de nossa área de trabalho.

Sistema de Referência do Universo (SRU)

- É chamado de coordenadas do universo, ou do mundo, o sistema de referência utilizado para descrever os objetos em termos das coordenadas utilizadas pelo usuário em determinada aplicação.
- Sendo assim, cada tipo de aplicação especifica o seu universo de trabalho próprio, por exemplo, para sistemas de CAD de arquitetura, o universo poderá ser em metros ou centímetros; e para um CAD de mecânica de precisão, o universo provavelmente estará em milímetros ou nanômetros.

Sistema de Referência do Universo (SRU)

- Em OpenGL, esse Sistema de Referência do Universo consiste em um plano cartesiano com dois eixos (x e y) perpendiculares entre si e que se cruzam na origem.
- Todos os modelos e comandos OpenGL são definidos em relação a este sistema de referência.

Sistema de Referência do Dispositivo (SRD)

- Utiliza coordenadas que podem ser fornecidas diretamente para um dado dispositivo de saída específico.
- Por exemplo, em um vídeo esses valores podem ser o número máximo de pixels que podem ser acesos (640x480, 800x600 etc.) ou podem indicar a resolução especificada em determinada configuração do sistema operacional

Sistema de Referência do Dispositivo (SRD)

Neste sistema de referência fazemos como que cada objeto seja um mini-universo individual, ou seja, cada objeto tem suas particularidades descritas em função de seu sistema, muitas vezes coincidindo o centro do sistema de coordenadas com o seu centro de gravidade.

Plano Cartesiano

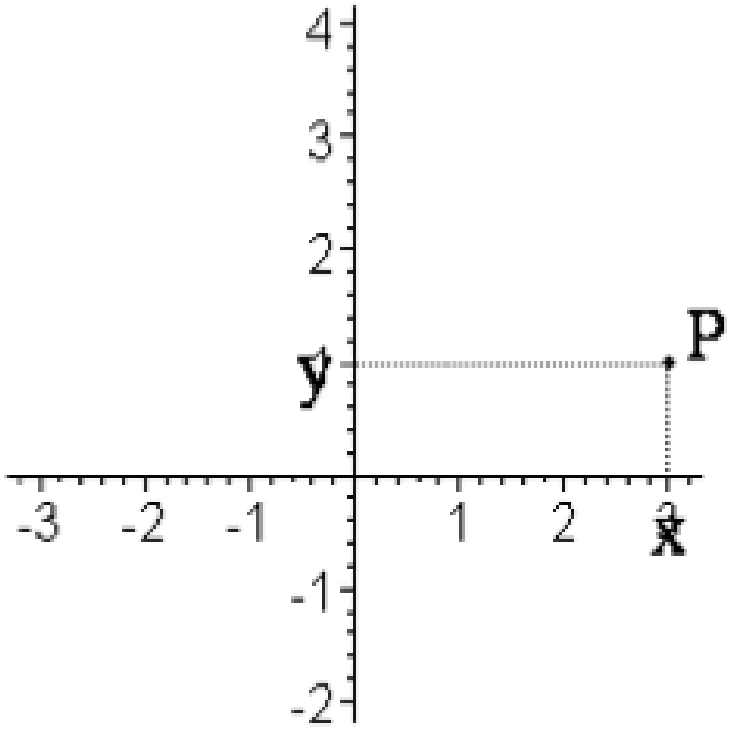
- Em 1619, o filósofo e matemático francês René Descartes (1596-1650) percebeu que a idéia de determinar posições utilizando retas
- Como o plano tem duas dimensões, para localizar pontos no plano, precisamos de dois números, ao invés de um.
- Descartes resolveu este problema usando duas retas numeradas, perpendiculares, cortando-se na origem.

Plano Cartesiano

- Esta reta será chamada eixo x ou eixo das abscissas. A outra reta, vertical com a direção positiva para cima, é chamada eixo y , ou eixo das ordenadas.

- Em openGL:

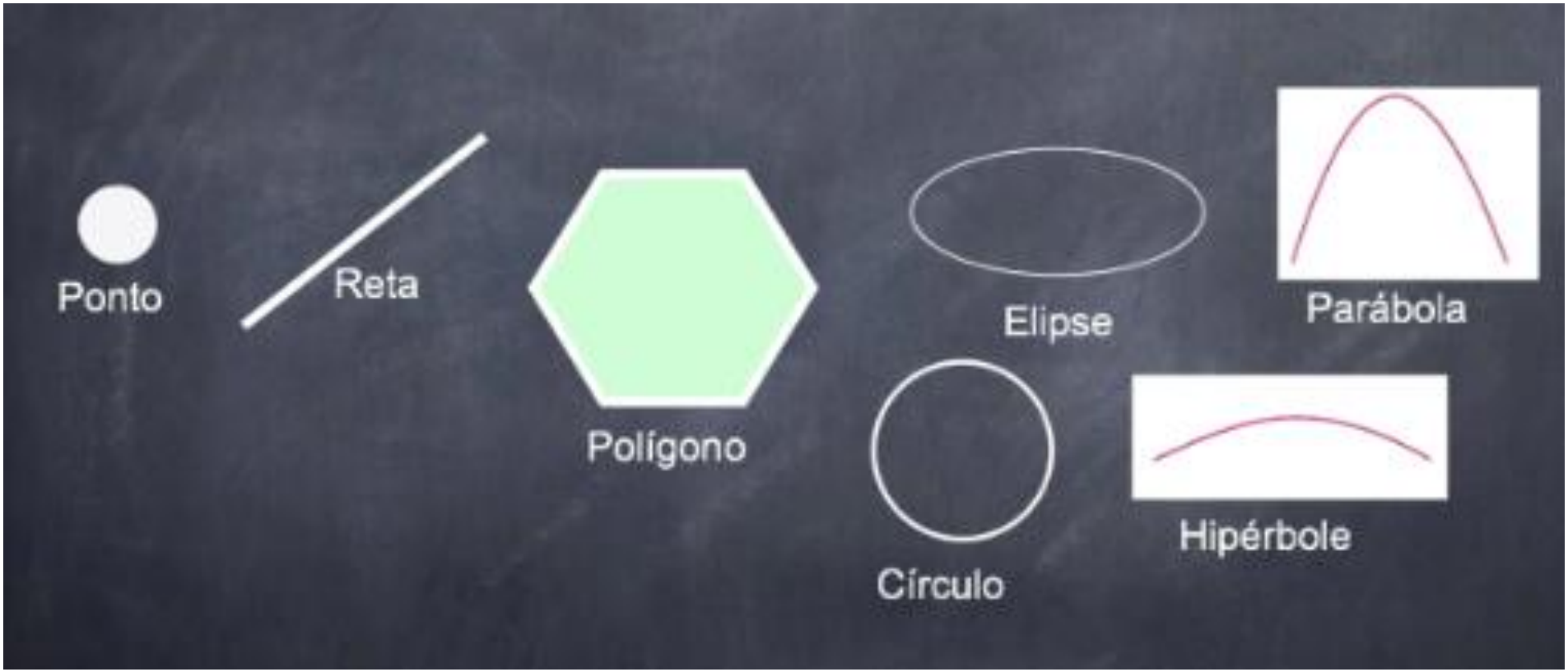
```
glBegin(GL_POINTS)
glVertex2i(3,1)
glEnd()
```



Primitivas Gráficas

- Primitivas gráficas consistem nos elementos básicos que compõem um desenho, tais como pontos, segmentos de reta e círculos.
- Em OpenGL, as primitivas são definidas em um sistema de coordenadas bidimensionais, por meio de um ou mais pares de coordenadas absolutas, chamadas de vértices (comando glVertex)

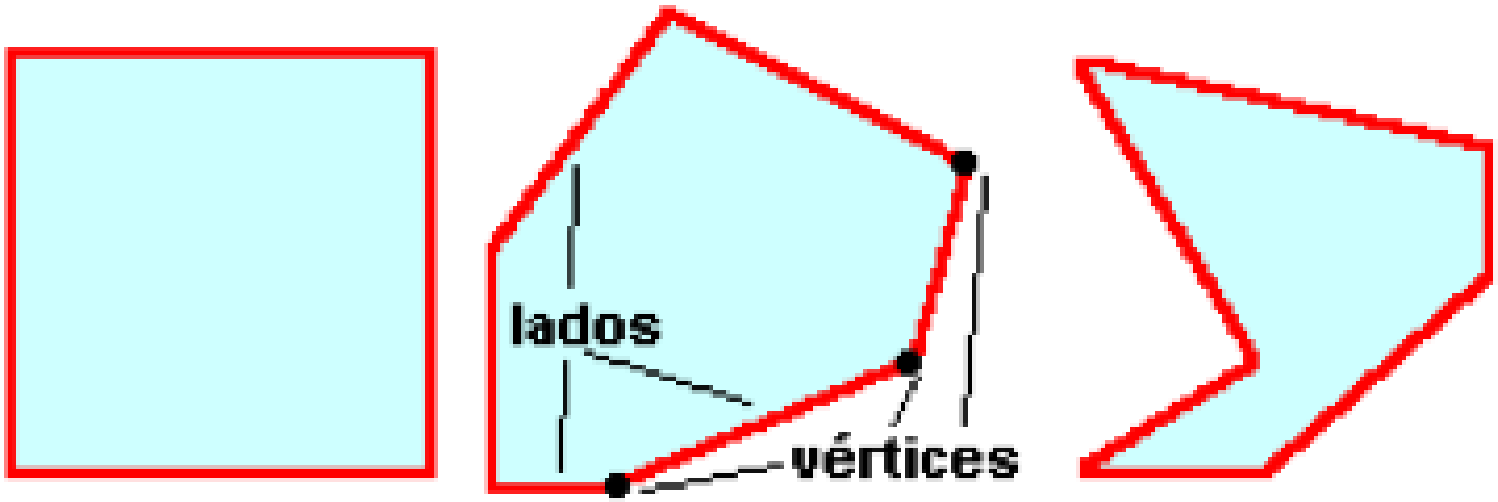
Primitivas Gráficas



Primitivas Gráficas
Polígono

- É uma figura plana formada por três ou mais segmentos de reta que se intersectam dois a dois. Os segmentos de reta são denominados lados do polígono.
- Os pontos de intersecção são denominados vértices do polígono. A região interior ao polígono é muitas vezes tratada como se fosse o próprio polígono

Primitivas Gráficas
Polígono

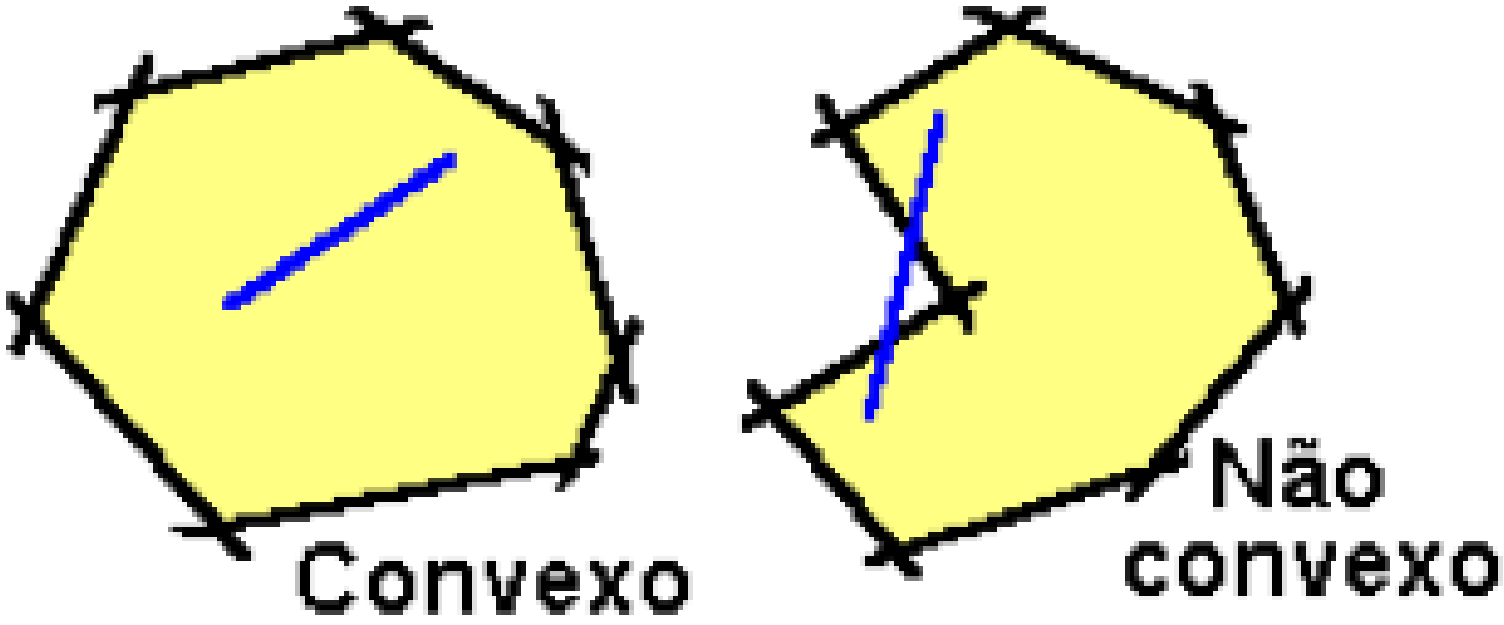


Primitivas Gráficas
Polígono

- Polígono convexo: É um polígono construído de modo que os prolongamentos dos lados nunca ficarão no interior da figura original. Se dois pontos pertencem a um polígono convexo, então todo o segmento tendo estes dois pontos como extremidades, estará inteiramente contido no polígono.

- Polígono não convexo: Um polígono é dito não convexo se dados dois pontos do polígono, o segmento que tem estes pontos como extremidades, contiver pontos que estão fora do polígono.

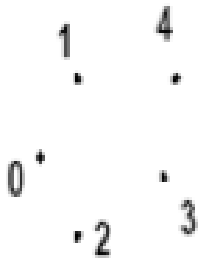
Primitivas Gráficas
Polígono



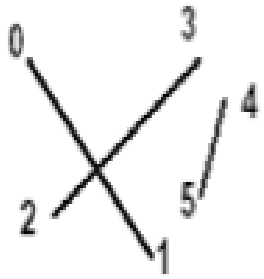
Primitivas Gráficas OpenGL

Valor	Descrição
GL_POINTS	Para desenhar pontos.
GL_LINES	Para desenhar segmentos de linha.
GL_LINE_STRIP	Para desenhar segmentos de linha conectados.
GL_LINE_LOOP	Para desenhar segmentos de linha conectados, unindo o primeiro ao último.
GL_POLYGON	Para desenhar um polígono convexo.
GL_TRIANGLES	Para desenhar triângulos.
GL_TRIANGLE_STRIP	Para desenhar triângulos conectados.
GL_TRIANGLE_FAN	Para desenhar triângulos a partir de um ponto central.
GL_QUADS	Para desenhar quadriláteros.
GL_QUAD_STRIP	Para desenhar quadriláteros conectados.

Primitivas Gráficas OpenGL

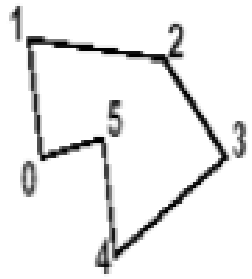


GL_POINTS

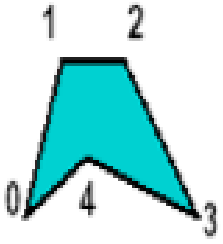


GL_LINES

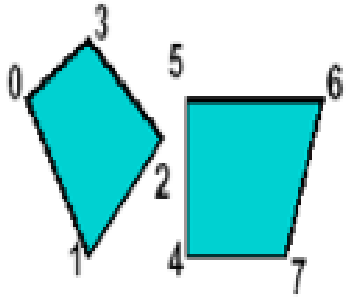
GL_LINE_STRIP



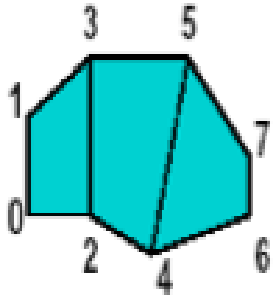
GL_LINE_LOOP



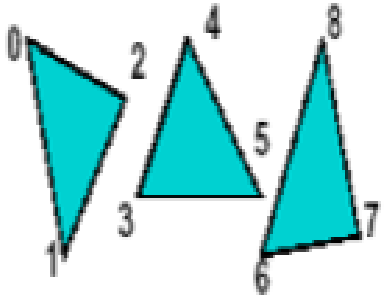
GL_POLYGON



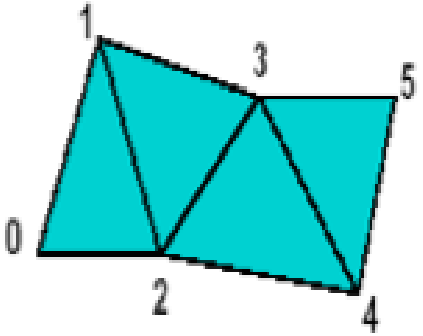
GL_QUADS



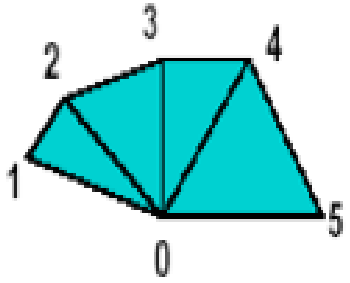
GL_QUAD_STRIP



GL_TRIANGLES



GL_TRIANGLE_STRIP



GL_TRIANGLE_FAN

Tipos de Tranformações

- Tipos de Transformações:
  - Translação
  - Escala
  - Rotação
  - Rotação 3D
  - Reflexão
  - Cisalhamento

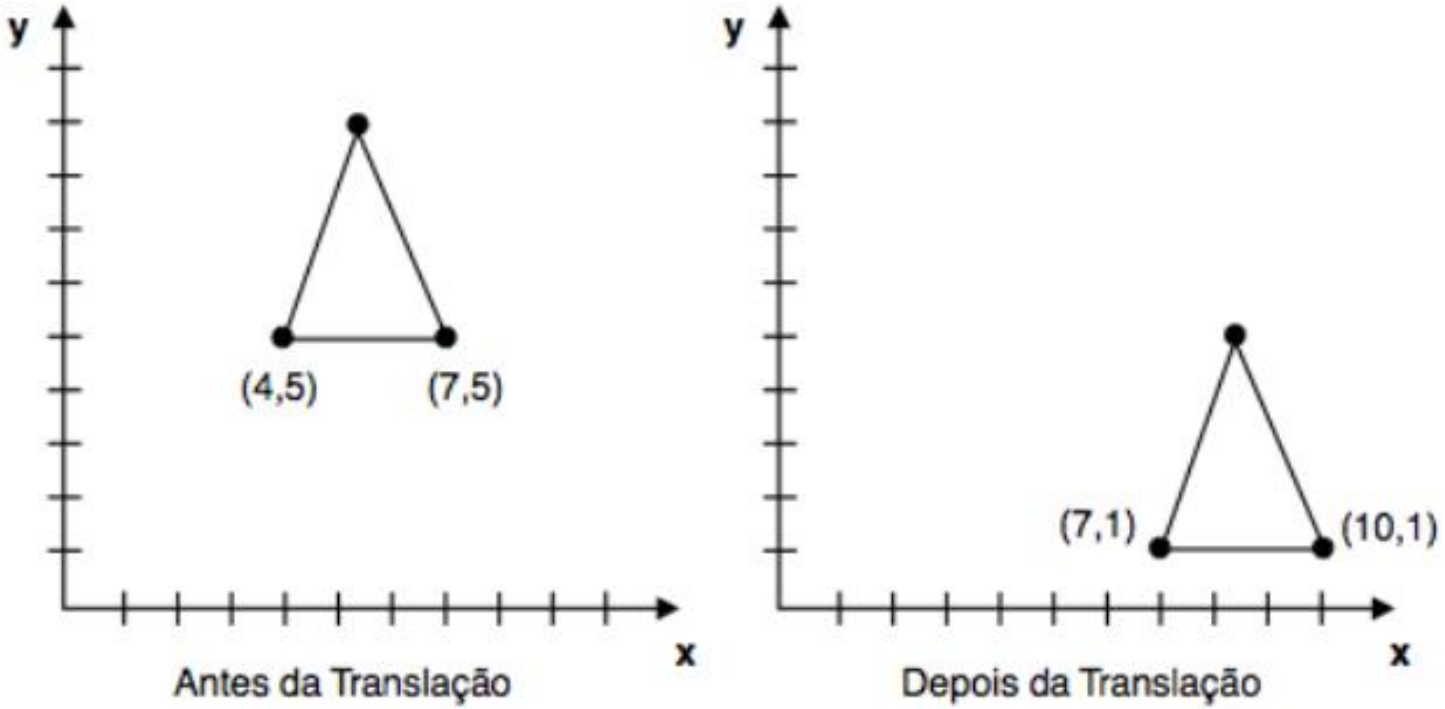
Translação

Definição:

Transladar significa movimentar o objeto. Para transladar um objeto deve-se transladar todos os seus pontos. Uma translação é basicamente uma operação matemática de adição ou subtração.

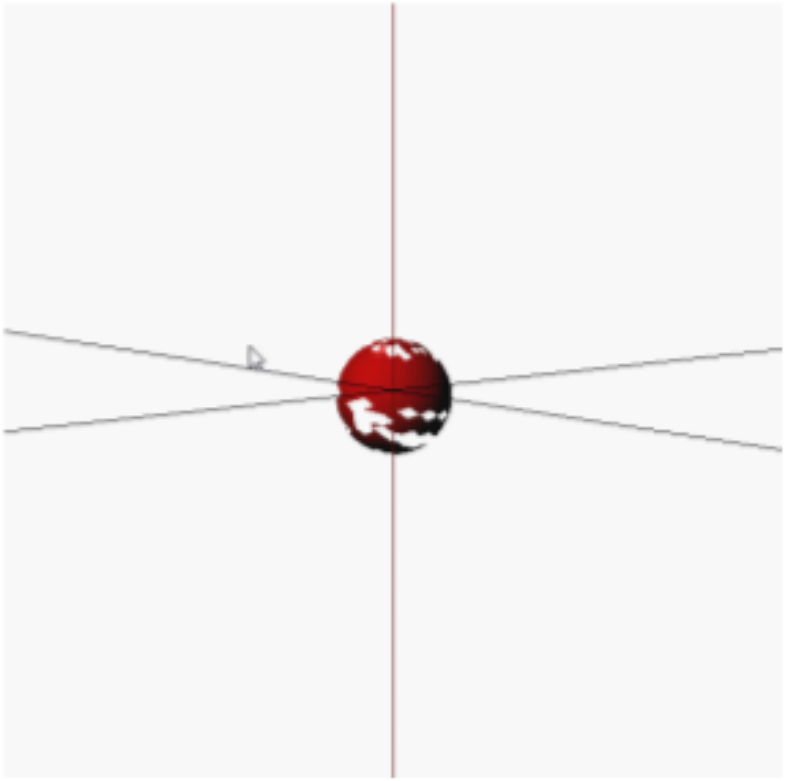
Translação

Exemplo:



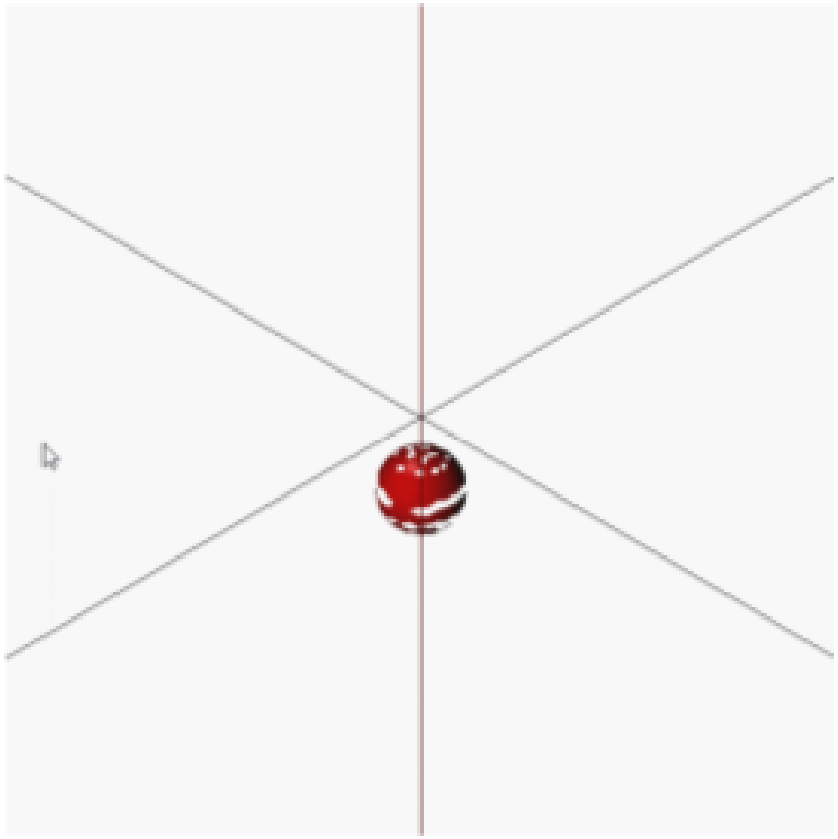
Translação

Exemplo:



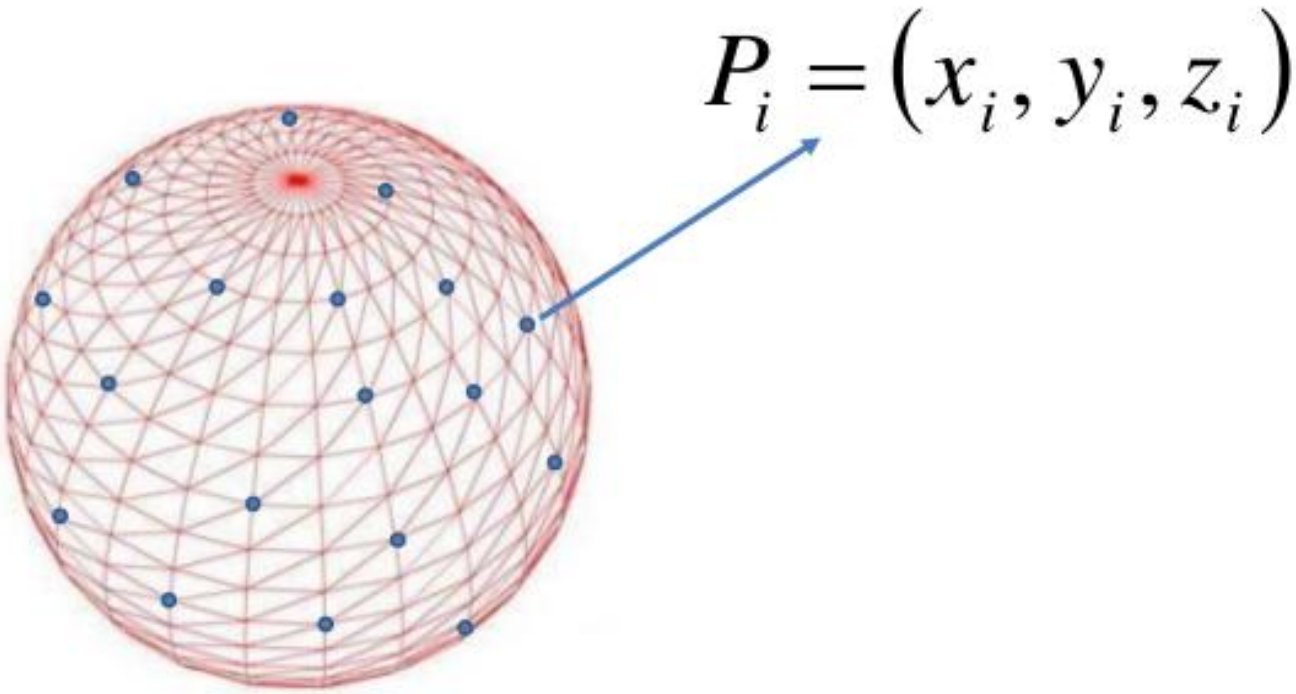
Translação

Exemplo:



Translação

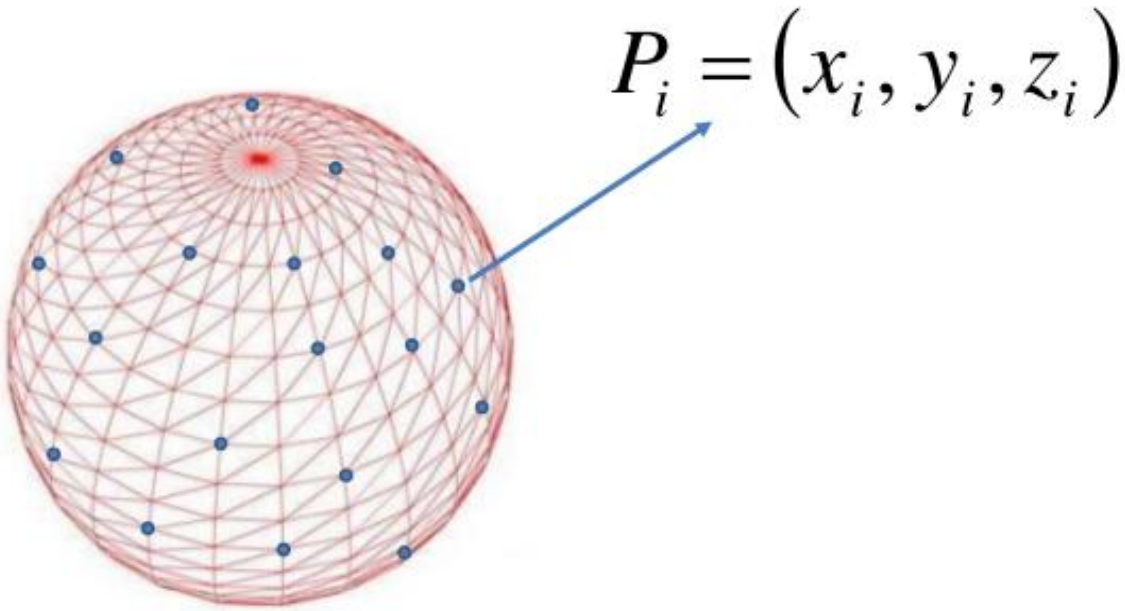
Exemplo:



Translação

Onde:

x, y e z são as coordenadas do ponto P.



Transformação de Translação

$$P_i = (x_i, y_i, z_i)$$

$$P_i = (x_i + \Delta x, y_i, z_i)$$

Transformação de Translação

$$P_i = (x_i, y_i, z_i)$$

$$P_i = (x_i + \Delta x, y_i + \Delta y, z_i)$$

Transformação de Translação

$$P_i = (x_i, y_i, z_i)$$

$$P_i = (x_i + \Delta x, y_i + \Delta y, z_i + \Delta z)$$

Transformação de Translação

$$P_i^{'} = (x_i + \Delta x, y_i + \Delta y, z_i + \Delta z)$$

$$P_i^{'} = \begin{bmatrix} x_i + \Delta x \\ y_i + \Delta y \\ z_i + \Delta z \end{bmatrix}$$

Transformação de Translação

$$P_i' = (x_i + \Delta x, y_i + \Delta y, z_i + \Delta z)$$

$$P_i' = \begin{bmatrix} x_i + \Delta x \\ y_i + \Delta y \\ z_i + \Delta z \end{bmatrix} = TP_i$$

Transformação de Translação

Onde:

- P são as coordenadas do ponto.
- T são os valores de deslocamento de x, y e z.
- P' são as coordenadas resultantes, ou seja, o novo ponto.

$$P_i^{'} = \begin{bmatrix} x_i + \Delta x \\ y_i + \Delta y \\ z_i + \Delta z \end{bmatrix} = TP_i$$

Transformação de Translação

$$P_i' = \begin{bmatrix} x_i + \Delta x \\ y_i + \Delta y \\ z_i + \Delta z \end{bmatrix} = T \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$$

Transformação de Translação

$$P_i' = \begin{bmatrix} x_i + \Delta x \\ y_i + \Delta y \\ z_i + \Delta z \end{bmatrix} = T \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$$

Transformação de Translação

$$P_i' = \begin{bmatrix} x_i + \Delta x \\ y_i + \Delta y \\ z_i + \Delta z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$

Transformação de Translação

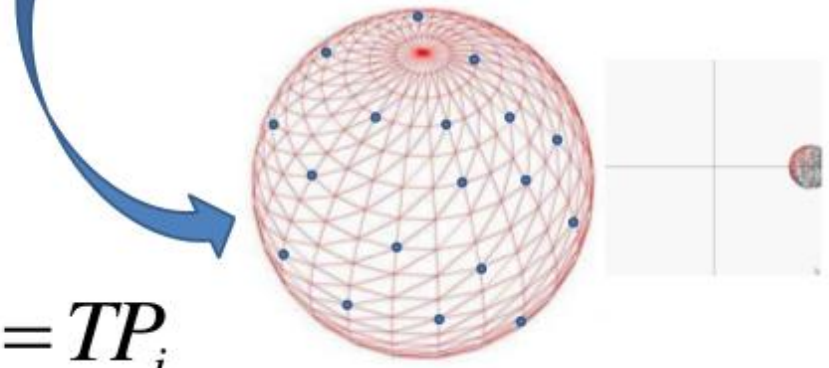
Matriz de Translação.

$$T = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformação de Translação

P Aplicando a matriz de translação T em todos os pontos do objeto, esse objeto será transladado.

$$T = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$P_i' = TP_i$$

Transformação de Translação

No **OpenGL** utiliza-se a seguinte função:

glTranslatef(tx, ty, tz**)**

Os parâmetros tx, ty e tz a serem informados são os valores de translação nos eixos x, y e z.

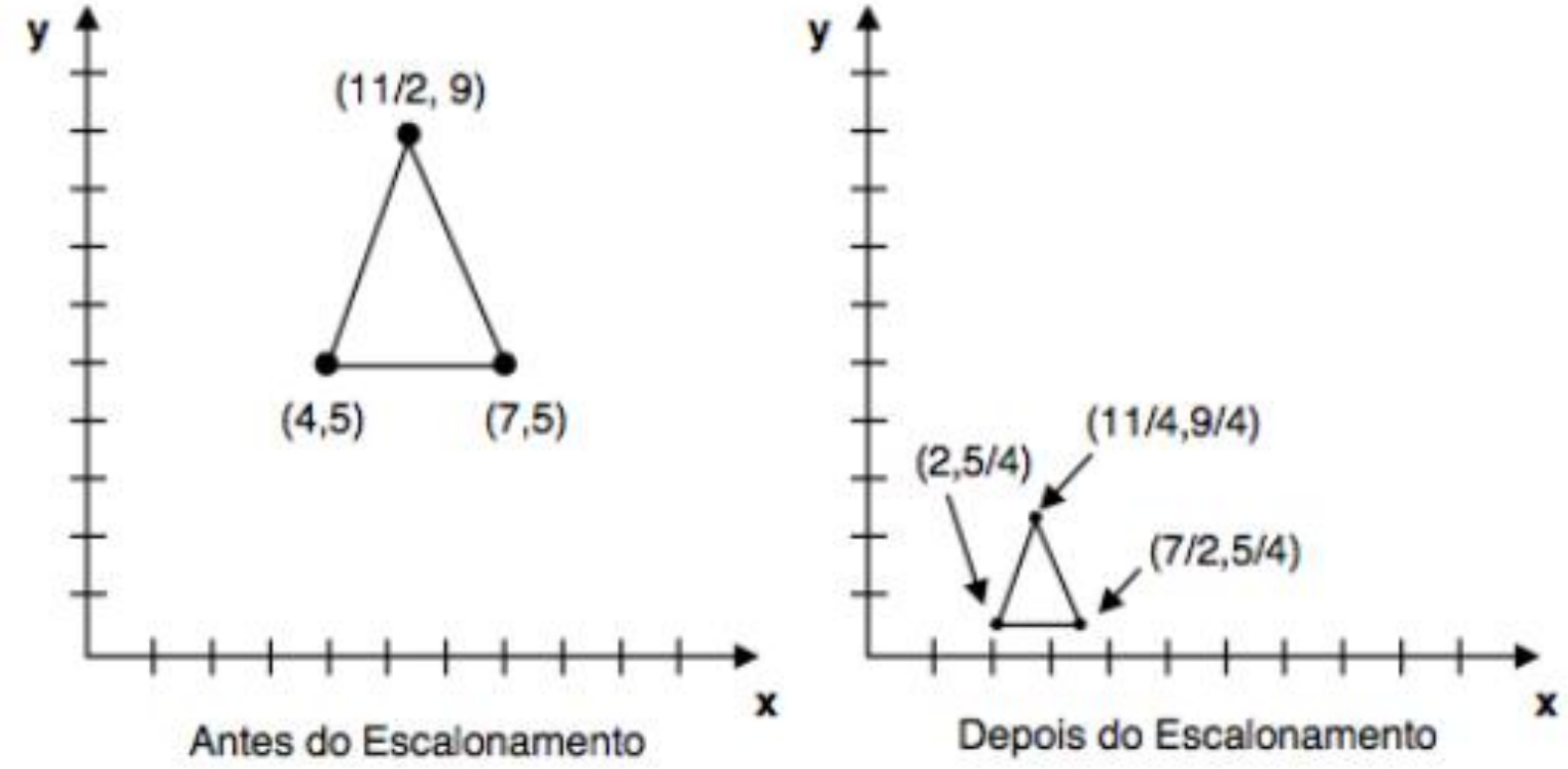
Escala

Definição:

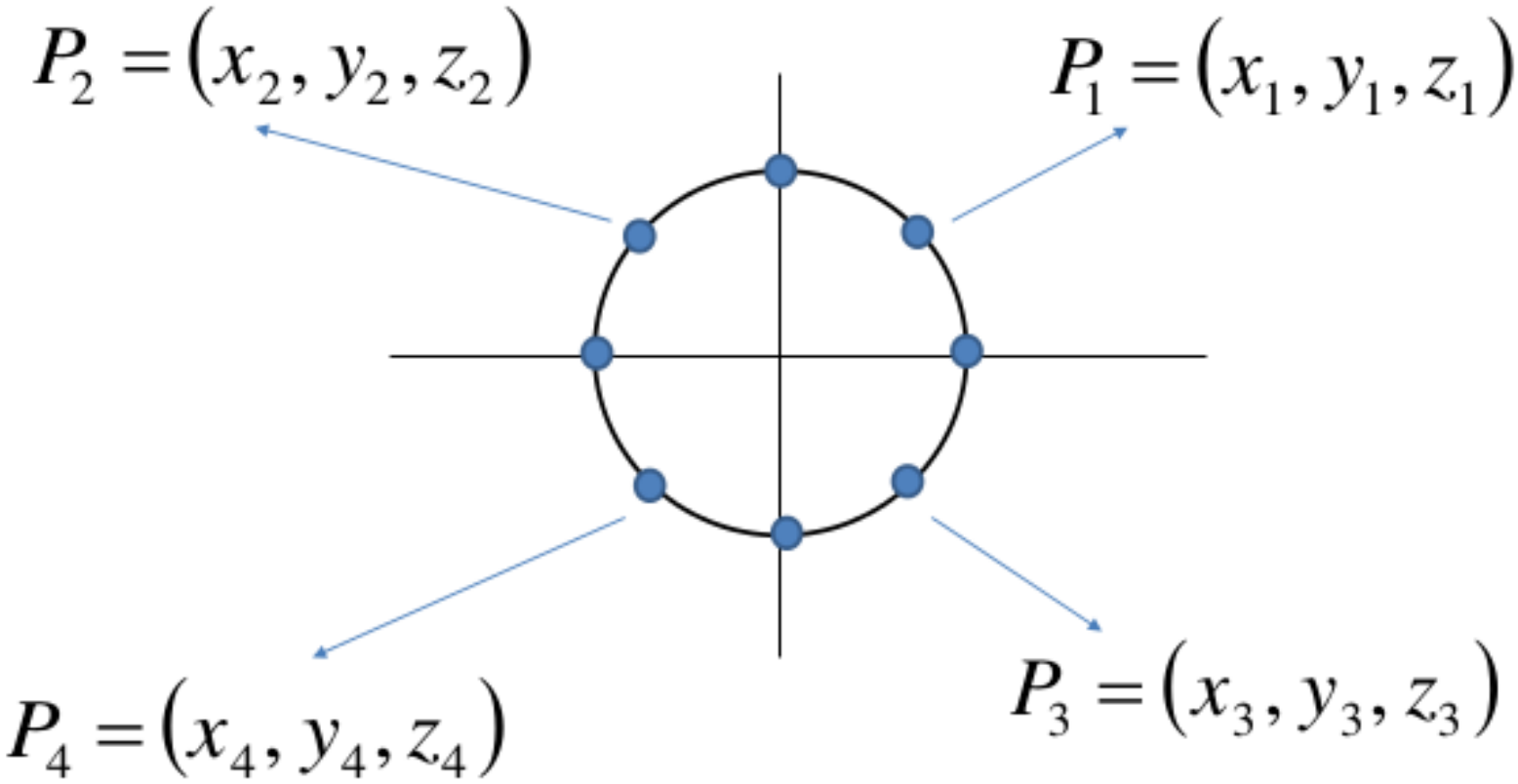
Escalonar significa mudar as dimensões de escala do objeto. Simplificando, escalonar é alterar o tamanho do objeto. O escalonamento de um objeto é realizado através da multiplicação dos seus pontos por um fator de escala..

Transformação de Escala

Exemplo:



Transformação de Escala



Transformação de Escala

$$P_1 = (x_1, y_1, z_1) \quad \Rightarrow P_1^{'} = (x_1 \Delta x, y_1 \Delta y, z_1 \Delta z)$$

$$P_2 = (x_2, y_2, z_2) \quad \Rightarrow P_2^{'} = (x_2 \Delta x, y_2 \Delta y, z_2 \Delta z)$$

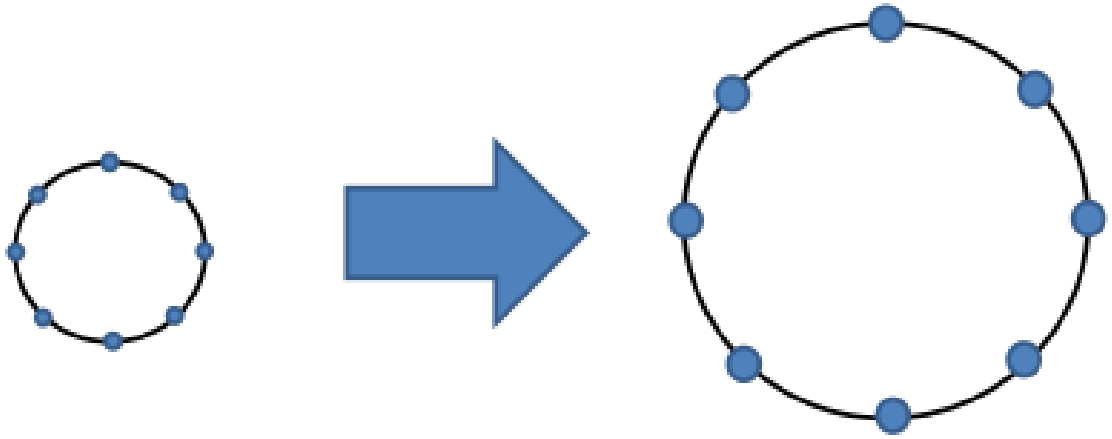
$$P_3 = (x_3, y_3, z_3) \quad \Rightarrow P_3^{'} = (x_3 \Delta x, y_3 \Delta y, z_3 \Delta z)$$

$$P_4 = (x_4, y_4, z_4) \quad \Rightarrow P_4^{'} = (x_4 \Delta x, y_4 \Delta y, z_4 \Delta z)$$

Transformação de Escala

Para Ampliar:

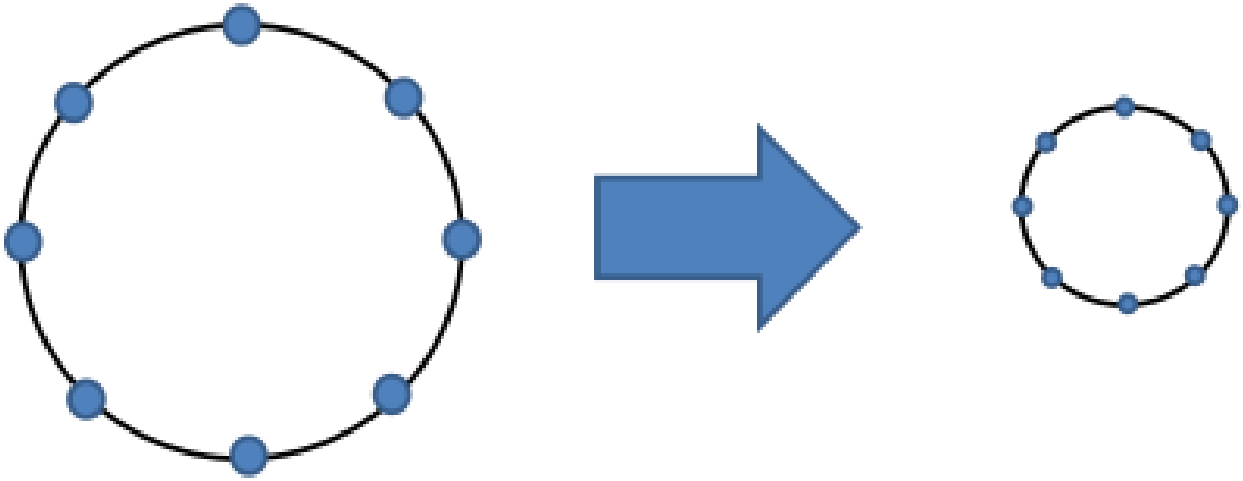
Fazer $\Delta x = \Delta y = \Delta z > 1.0$



Transformação de Escala

Para Reduzir:

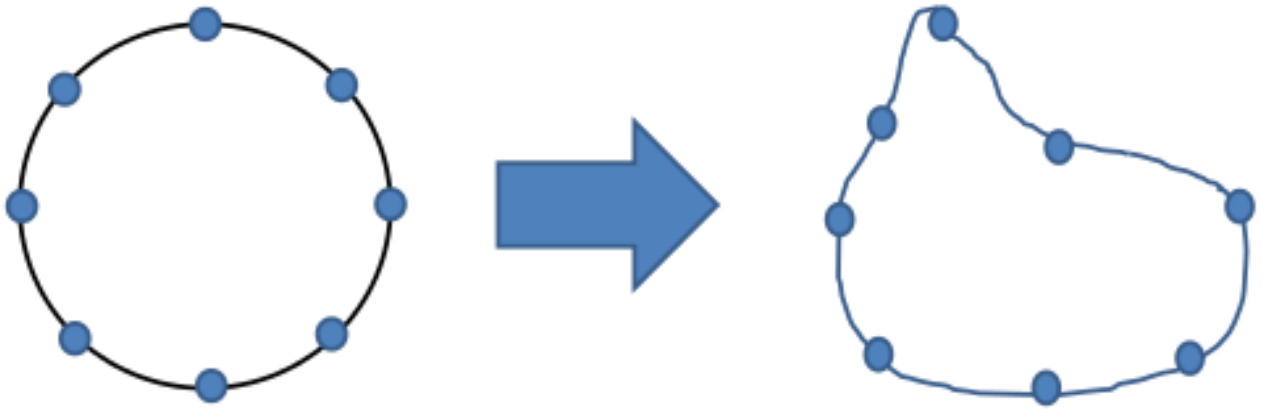
$$\text{Fazer} \quad \Delta x = \Delta y = \Delta z < 1.0$$



Transformação de Escala

Se o escalonamento for desigual:

$$\Delta x \neq \Delta y \neq \Delta z$$



Transformação de Escala

$$P_i^{'} = (x_i \Delta x, y_i \Delta y, z_i \Delta z)$$

$$P_i^{'} = \begin{bmatrix} x_i \Delta x \\ y_i \Delta y \\ z_i \Delta z \end{bmatrix}$$

Transformação de Escala

$$P_i' = \begin{bmatrix} x_i \Delta x \\ y_i \Delta y \\ z_i \Delta z \end{bmatrix} = S \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$$

Transformação de Escala

$$P_i' = \begin{bmatrix} x_i \Delta x \\ y_i \Delta y \\ z_i \Delta z \end{bmatrix} = \begin{bmatrix} \Delta x & 0 & 0 \\ 0 & \Delta y & 0 \\ 0 & 0 & \Delta z \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$$

Transformação de Translação

Matriz de Escalonamento.

$$S = \begin{bmatrix} \Delta x & 0 & 0 \\ 0 & \Delta y & 0 \\ 0 & 0 & \Delta z \end{bmatrix}$$

Transformação de Escala

No OpenGL utiliza-se seguinte função:

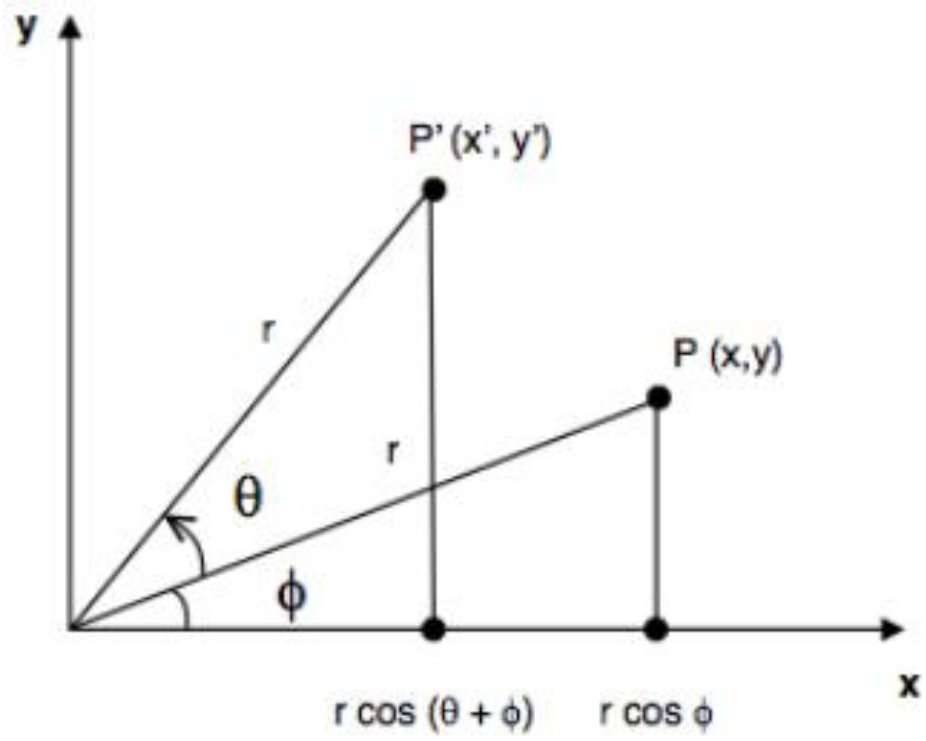
glScalef(Sx, Sy, Sz)

Os parâmetros Sx, Sy e Sz a serem informados são os fatores de escala para os eixos x, y e z.

Rotação

Definição:

Rotacionar significa girar. Veja o exemplo:



Rotação

Definição:

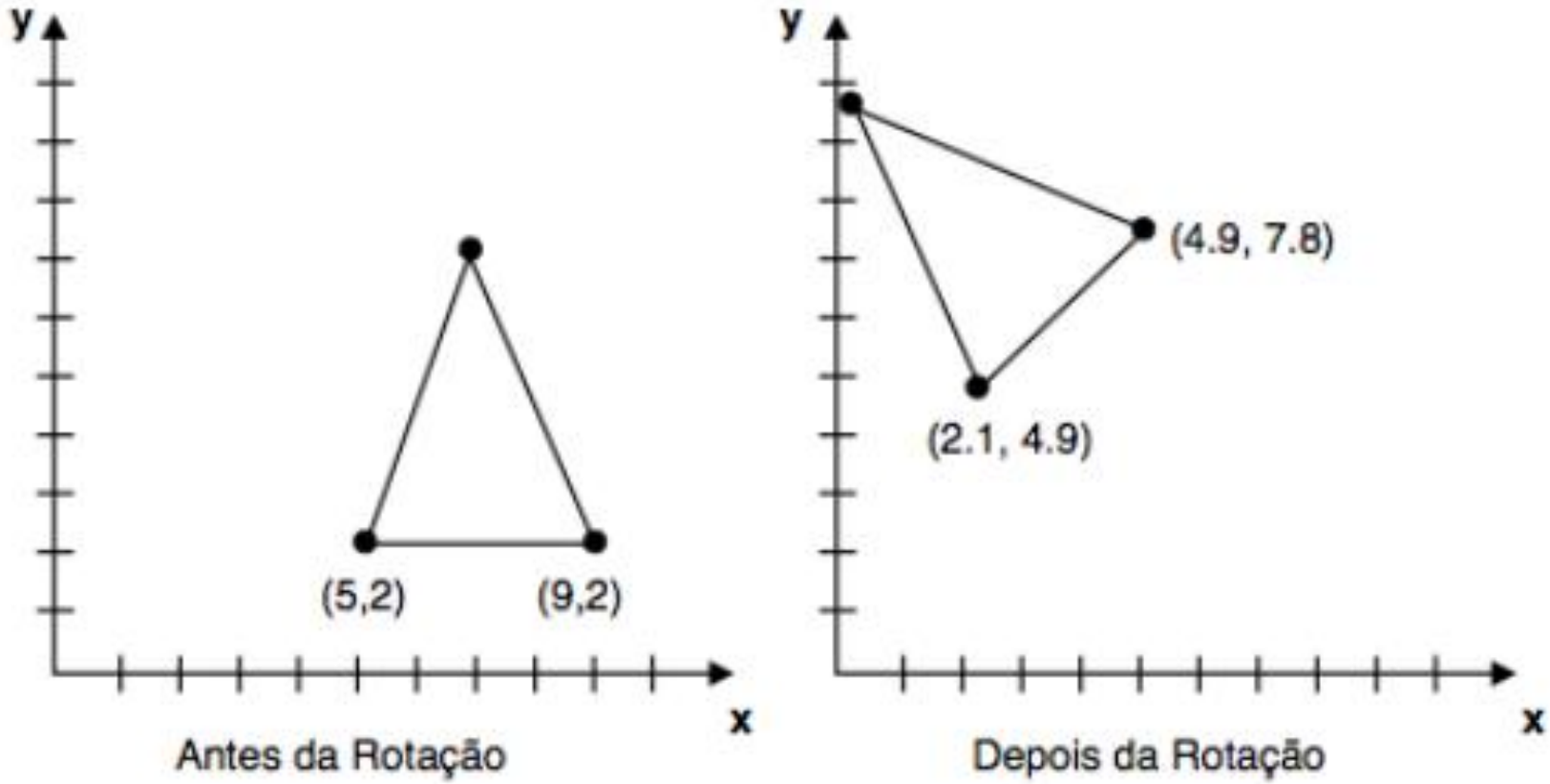
A rotação, geralmente, ocorre com base ao ponto de origem.
Por exemplo, dado um ponto P com coordenadas (x, y), com distância r = (x 2 + y 2) 1/2 da origem, for rotacionado de um ângulo θ, as coordenadas antes definidas por: $x = r * \cos(\varphi)$ e $y = r * \sin(\varphi)$, tornam-se agora (x , y) definidos como:

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

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

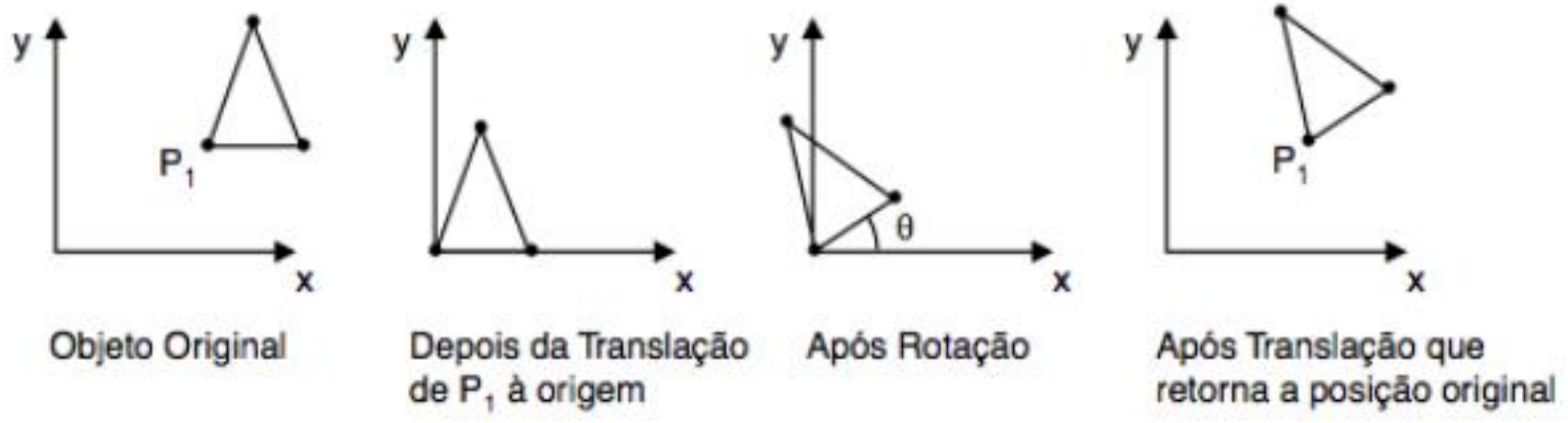
Rotação

Rotação a partir do ponto de origem:



Rotação + Translação

Para rotacionar um objeto a partir de um ponto que não seja a origem, deve-se primeiro realizar uma translação alinhando o ponto desejado com a origem, aplicar a rotação de um ângulo θ, e depois uma nova translação para retornar o objeto ao seu lugar na cena. Veja no exemplo:



Rotação 3D

Definição:

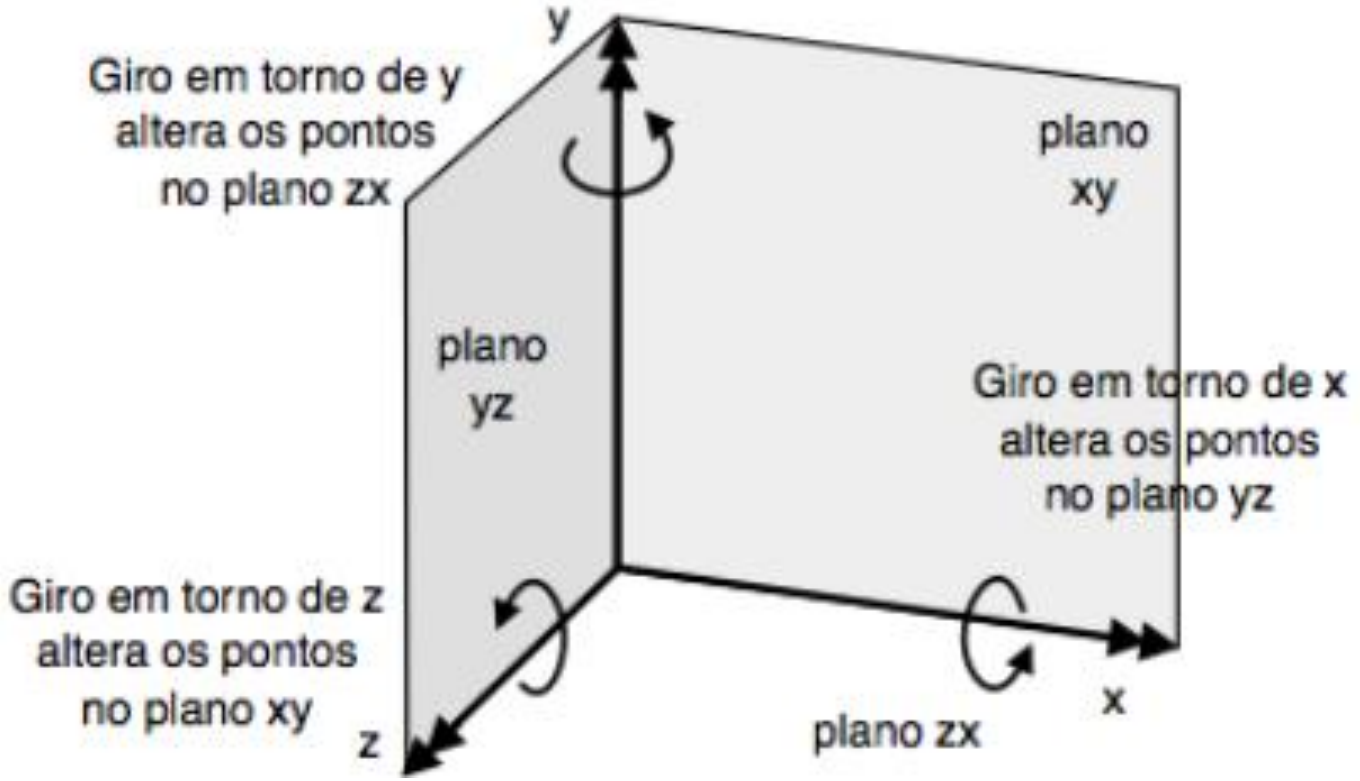
Euler, matemático do século XVIII, mostrou que:

- A combinação de um número qualquer de rotações em 3D pode ser representada por uma única rotação ao redor de um vetor apropriado.
- Qualquer rotação 3D pode ser dividida em 3 rotações ao redor do sistema de coordenadas.

Rotação 3D

Definição dos 3 ângulos de Euler em relação aos eixos x, y, z:

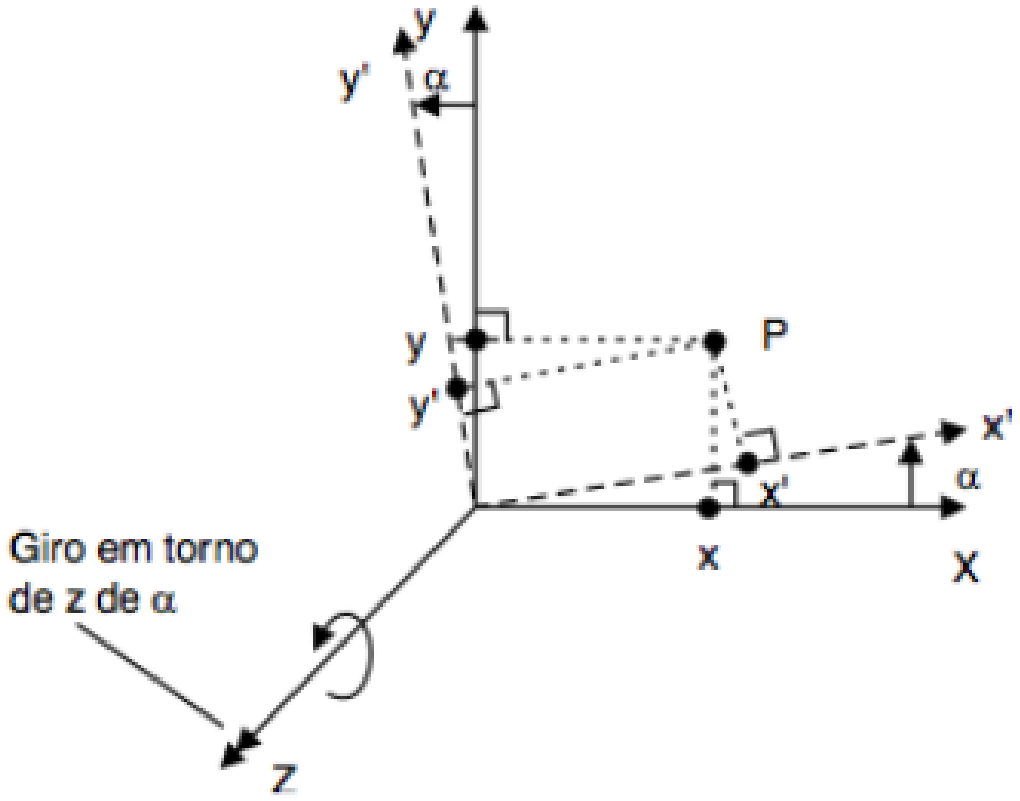
Rotacionar objetos no espaço em um certo ângulo



Rotação 3D

Angulo de Euler em torno do eixo z:

Só os pontos do plano xy são alterados. Rotacionar o próprio espaço com o ângulo.



Rotação 3D

Ângulo de Euler em torno do eixo z:

Como funciona a rotação do eixo **z** matematicamente:

$$[x' \;\; y' \;\; z'] = [x \;\; y \;\; z] * \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotação 3D

Ângulo de Euler em torno do eixo x:

Como funciona a rotação do eixo **x** matematicamente:

$$[x' \ y' \ z'] = [x \ y \ z] * \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta) & \sin(\beta) \\ 0 & -\sin(\beta) & \cos(\beta) \end{bmatrix}$$

Rotação 3D

Ângulo de Euler em torno do eixo y:

Como funciona a rotação do eixo **y** matematicamente:

$$[x' \ y' \ z'] = [x \ y \ z] * \begin{bmatrix} \cos(\gamma) & 0 & -\sin(\gamma) \\ 0 & 1 & 0 \\ \sin(\gamma) & 0 & \cos(\gamma) \end{bmatrix}$$

Rotação 3D

No OpenGL utiliza-se seguinte função:

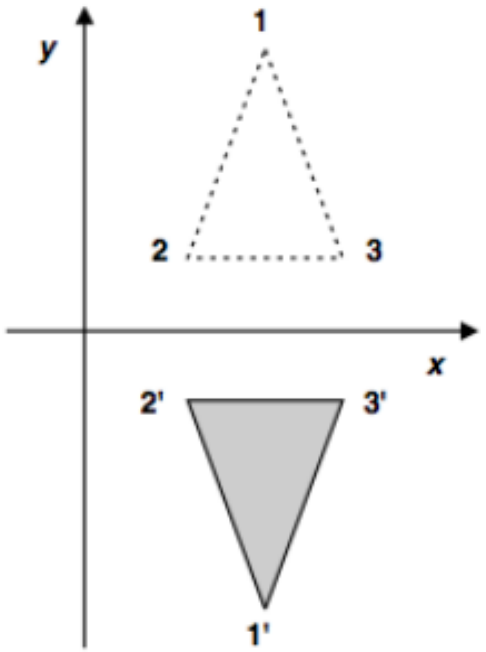
glRotatef(ângulo, X, Y, Z)

Os parâmetros a serem informados são os valores para o ângulo de rotação e os eixos que podem ser aplicada a rotação.

Reflexão

Definição:

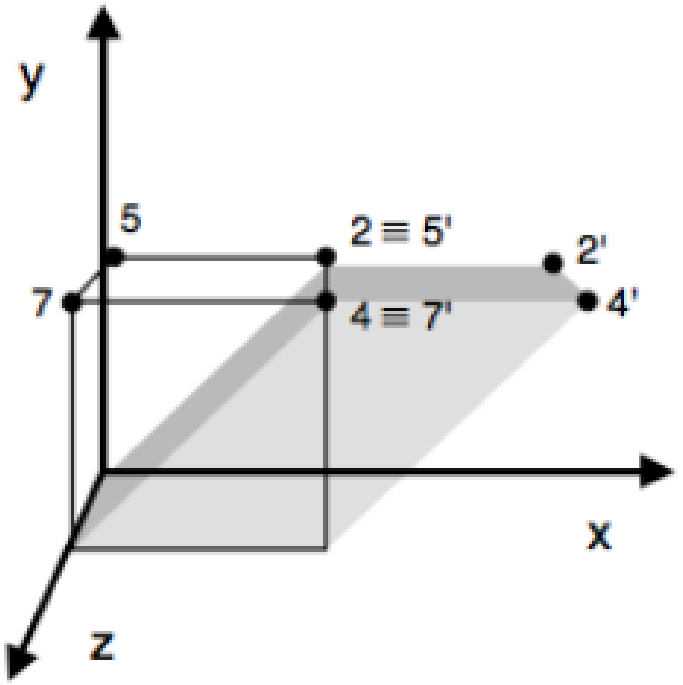
Também conhecida como espelhamento ou flip ocorre sempre em torno de um eixo. O processo cria um objeto similar ao anterior só que invertido como se fosse visto através de um espelho. Veja o exemplo:



Cisalhamento

Definição:

Também conhecida como Shearing ou Skew é uma transformação que distorce o formato de um determinado objeto. Veja o exemplo:



Cisalhamento

Matriz de Cisalhamento 2D:

Cisalhar um objecto é deformá-lo linearmente ao longo do eixo x ou do eixo y ou de ambos.
Deixar Kx = 1 e Ky = 0

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \kappa_x & 0 \\ \kappa_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$