

OPENGL – PRÁTICA 04

Translação

Define a nova posição de um ou mais objetos através de constantes de deslocamento. As coordenadas (x, y, z) de cada vértice do(s) objeto(s) terão um novo valor de acordo com a constante aplicada sobre seus respectivos eixos.

Matematicamente:

$$\begin{aligned}x' &= x + T_x \\y' &= y + T_y \\z' &= z + T_z\end{aligned}$$

Sendo T_x , T_y e T_z , as constantes de deslocamento aplicadas sobre os eixos x , y e z . A função OpenGL responsável por transladar um ou mais objetos de uma cena:

```
void glTranslatef(GLfloat tx, GLfloat ty, GLfloat tz);
```

A função recebe como parâmetros as constantes de deslocamento para os eixos x , y e z , respectivamente, e desloca o(s) objeto(s) a partir de sua posição inicial. A translação é aplicada na origem do sistema de referência. Dessa forma, aconselha-se que os objetos sejam sempre posicionados ao redor do ponto $(0,0,0)$.

Escala

Define o novo tamanho de um ou mais objetos através de constantes que estabelecem o fator de escala. As coordenadas (x, y, z) de cada vértice do(s) objeto(s) terão um novo valor de acordo com o fator de escala aplicado sobre seus eixos.

Matematicamente:

$$\begin{aligned}x' &= x * E_x \\y' &= y * E_y \\z' &= z * E_z\end{aligned}$$

Sendo E_x , E_y e E_z , os fatores de escala aplicados sobre os eixos x , y e z . A função OpenGL responsável por escalonar um ou mais objetos de uma cena:

```
void glScalef(GLfloat ex, GLfloat ey, GLfloat ez);
```

A função recebe como parâmetros as constantes de fator escala para os eixos x , y e z , respectivamente, e redimensiona o(s) objeto(s) a partir de seu tamanho original. Assim como na translação, a transformação de escala é aplicada na origem do sistema de referência. Dessa forma, aconselha-se que os objetos sejam sempre posicionados ao redor do ponto $(0,0,0)$.

Rotação

Define o valor do ângulo que será utilizado como referência para girar um ou mais objetos. Matematicamente, esta operação consiste em aplicar uma composição de cálculos empregando o seno e o cosseno do ângulo de rotação a todas as coordenadas das primitivas gráficas que compõe o objeto ou cena. Por exemplo, uma rotação em torno do eixo z pode ser assim expressa:

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

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

A Função OpenGL responsável pela rotação de um ou mais objetos de uma cena:

```
void glRotatef(GLfloat angulo, GLfloat x, GLfloat y, GLfloat z);
```

A função recebe como parâmetros o ângulo de rotação e os eixos x , y e z , respectivamente, dos quais devem ser aplicados a rotação sobre o(s) objeto(s). Assim como na translação e escala, a transformação de rotação é aplicada na origem do sistema de referência. Dessa forma, aconselha-se que os objetos sejam sempre posicionados ao redor do ponto (0,0,0).

Matriz de Transformação

Como os comandos de visualização e transformações geométricas aplicadas sobre os objetos são executados por meio de operações com matrizes, torna-se necessário especificar qual matriz deve ser utilizada. A função `glMatrixMode` especifica a matriz que se deseja transformar. Já a função `LoadIdentity` faz com que a matriz de transformação corrente seja inicializada com a matriz identidade, indicando que nenhuma transformação foi aplicada.

É muito importante observar que a ordem na qual as transformações geométricas são aplicadas altera o resultado final. Utilize o **Projeto01** para compreender como esses conceitos são aplicados. Primeiro aplicou-se uma translação de 15 unidades no eixo x , depois uma rotação de 90° em torno do eixo z e por último uma escala de 2 no eixo x .

Tratamento de eventos do teclado e teclas especiais

Além do tratamento de teclas que produzem um resultado visível na tela (letras, números e sinais de pontuação), existem as chamadas teclas especiais (HOME, END, F1, F2 etc.) que podem ser usadas, mas não são tratadas pelo *callback* de teclado. Para essas teclas, existe uma função de *callback* específica. A função `glutSpecialFunc` da GLUT é responsável por definir a rotina de tratamento de teclas especiais. Utilize o **Projeto02** para compreender como esses recursos são aplicados.

Escopo das transformações

OpenGL também permite restringir o escopo das transformações geométricas. Por exemplo, se existem dois objetos em uma cena e apenas um deles deve ser rotacionado. Se aplicarmos a rotação e depois desenharmos os dois objetos, ambos serão rotacionados. A OpenGL resolve esse problema pela implementação de uma pilha de matrizes de transformação. Duas funções podem ser utilizadas para definir o escopo das transformações:

- A função `glPushMatrix` guarda a matriz de transformação corrente na pilha.
- A função `glPopMatrix` recupera a matriz de transformação no topo da pilha e substitui a matriz de transformação corrente utilizada pela OpenGL.

Quando `glPushMatrix` é chamada uma cópia do conteúdo da matriz corrente é armazenada no topo da pilha. E ao chamar `glPopMatrix` a matriz armazenada no topo da pilha é retirada e atribuída para a matriz de transformação corrente. Sendo assim, o estado anterior é restaurado. Utilize o **Projeto3** para compreender os conceitos de escopo das transformações apresentados.

Exercício

Implemente um programa em OpenGL que crie uma cena (Figura 1) com quatro quadrados e um círculo com 10 vértices.

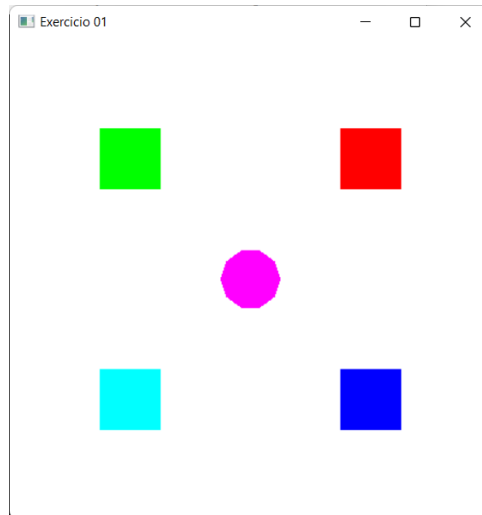


Figura 1 – Exercício 01

Comportamento do programa ao pressionar as teclas:

Tecla	Código da tecla	Comportamento
direcional esquerda	GLUT_KEY_LEFT	mover os quadrados para esquerda e rotacionar o círculo em sentido horário
direcional direita	GLUT_KEY_RIGHT	mover os quadrados para direita e rotacionar o círculo em sentido anti-horário
direcional acima	GLUT_KEY_UP	mover os quadrados para cima e rotacionar o círculo em sentido anti-horário
direcional abaixo	GLUT_KEY_DOWN	mover os quadrados para baixo e rotacionar o círculo em sentido horário
home	GLUT_KEY_HOME	aumentar o tamanho dos quadrados
end	GLUT_KEY_END	diminuir o tamanho dos quadrados
+	43	aumentar o tamanho do círculo
-	45	diminuir o tamanho do círculo

A proposta do exercício foi retirada e adaptada do exemplo desenvolvido na videoaula "Transformações Geométricas 2D no OpenGL - Parte 02" da prof. Ana Luísa (<https://www.youtube.com/watch?v=tMxGscrtHVV>).

Referências

Todas as informações descritas neste roteiro foram retiradas do livro "OpenGL: Uma abordagem prática e objetiva", presente no plano didático da disciplina, e do site da Profa. Isabel Harb Manssour (<https://www.inf.pucrs.br/~manssour/OpenGL/Tutorial.html>)