Relatório Exercício Prático 7

Gustavo Ciotto Pinton 117136
EA979 - Processamento de Imagens

Explicação

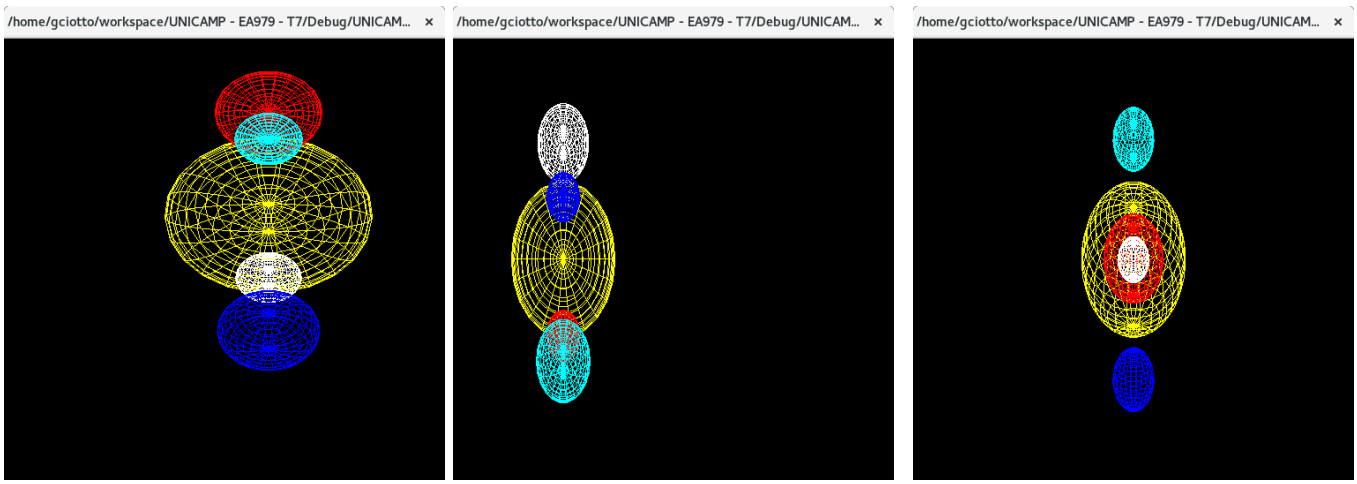
1. A função `glViewport()` mapeia as coordenadas da cena, em três dimensões, para as coordenadas da tela, que apresentam, por sua vez, apenas duas dimensões. Tal função recebe quatro parâmetros, sendo eles x e y, que especificam as coordenadas do canto inferior esquerdo, a largura w e a altura h da janela. A modificação dos dois primeiros produz um efeito de translação na imagem, sendo que valores positivos de x *transladam* os objetos para a direita e, de y, para cima. A imagem 1a é o efeito da execução do programa para $x = 50$ e $y = 50$. Nota-se, de fato, o efeito de translação. Abaixo, está representada a função com os respectivos parâmetros:

```
void reshape (int w, int h) {
glViewport (50, 50, (GLsizei) w, (GLsizei) h);
```

   A mudança do parâmetro w para um valor inferior que a largura efetiva da janela ($w = 0.5 * w$, imagem 1b) produz, por sua vez, o efeito de encolhimento (ou escala) nos elementos no eixo x, uma vez que a biblioteca *OpenGL* considera apenas aquele intervalo reduzido na renderização. Temos, abaixo, a chamada da função responsável por gerar a respectiva imagem.

```
void reshape (int w, int h) {
glViewport (0, 0, (GLsizei) w * 0.5, (GLsizei) h);
```

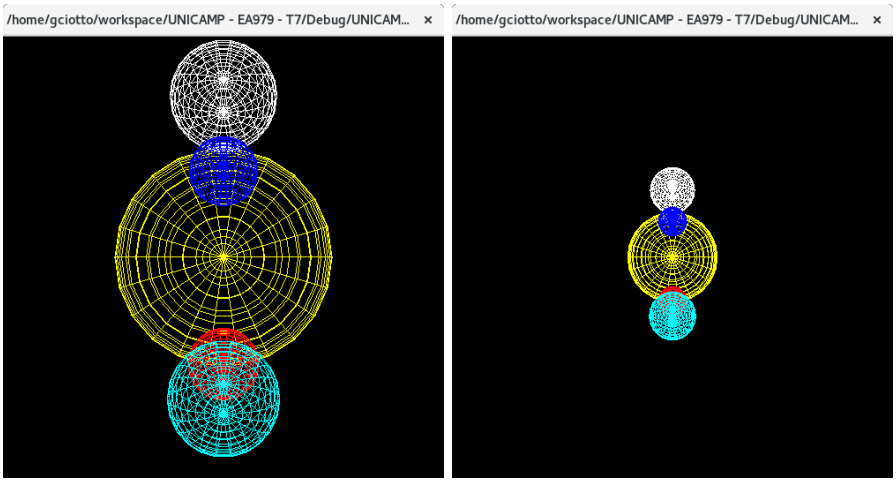
   Enfim, a imagem 1c combina as duas manipulações anteriores de forma a centralizar a cena na área da janela.



(a) Modificação de x e y. (b) Modificação de w. (c) Modificação de x, y e w.

Figura 1: Modificação dos parâmetros da função `glViewPort()`.

   A função `gluPerspective` imita o funcionamento olho humano e é responsável por adicionar efeitos de profundidade. A modificação das posições de `zNear` e `zFar` não modificam o tamanho com que os objetos são desenhados, uma vez que especificam os limiares dos dois planos. Por outro lado, os parâmetros *fovy* e *scale* produzem resultados diferentes. O primeiro é o ângulo de visão e o segundo, o fator de proporcionalidade que determina o campo de visão na direção x. A modificação do primeiro para um ângulo superior aumenta o campo de visão e produz o efeito de *distanciamento* da cena, conforme figura 2b, em que tal parâmetro vale 90º. A mudança do segundo gera alterações semelhantes àquelas produzidas por `glViewPort()`.



(a) *fovy* = 45º. (b) *fovy* = 90º.

Figura 2: Modificação do parâmetro *fovy* da função `gluPerspective()`.

2. A fim de desenharmos uma pirâmide, escrevemos uma função chamada `draw_pyramid()`, que chama, por sua, a função `glDrawElements()` com o parâmetro `GL_TRIANGLES`. Cada pirâmide pode ser desenhada com 6 triângulos: 2 para a base quadrada e 4 para os lados.

   Dentro da função `display()`, realizamos 3 chamadas de `draw_pyramid()`, modificando, em cada caso, a escala e a translação do sistema de coordenadas através das funções `glScalef` e `glTranslatef`. A maior pirâmide possui um z mais negativo e, a menor, um menos negativo. O resultado da execução pode ser conferido na figura 3. Nota-se que a biblioteca *OpenGL* gerencia os casos de oclusão gerados pela construção explicada acima, conforme esperado.

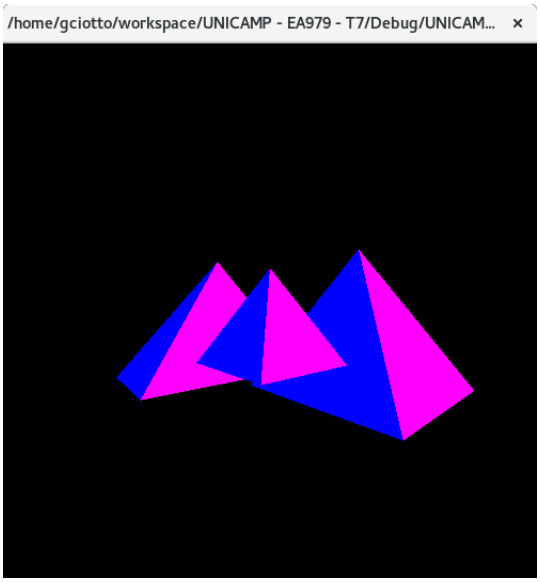


Figura 3: Renderização de 3 pirâmides.

Referências

1. `glViewport` *reference page*, disponível em `https://www.opengl.org/sdk/docs/man/html/glViewport.xhtml`.

2. `gluPerspective` *reference page*, disponível em `https://www.opengl.org/sdk/docs/man2/xhtml/gluPerspective.xml`.