

MC930A - Computação Gráfica
Prof: Helio Pedrini

Trabalho 1

Entrega: 08/09/2009

Aluno: Gustavo Serra Scalet
RA: 061358

Introdução:

Nesse exercício, descrito no documento¹, foi o primeiro trabalho de OpenGL da disciplina com o intuito de conhecermos superficialmente o poder do OpenGL para desenhos gráficos.

A minha implementação foi feita em Python pois é uma linguagem a qual eu possuo maior experiência e satisfação em programar. A prototipagem do trabalho foi rápida e a maior dificuldade encontrada foi para conhecer as funções básicas do OpenGL, pois embora exista muitas funções já prontas, é necessário conhecê-las.

Comandos (exibidos na inicialização do programa):

Gerais:

ESC key: Quit.

SPACE key: Cycles through the objects (selected object is drawn in white!).

Transformação de escala:

+ key: Makes the object bigger.

- key: Makes the object smaller.

Transformação de translação:

ARROW keys: Moves the object.

Transformação de rotação:

R key: +Rolls the object (through the X axis).

r key: -Rolls the object (through the X axis).

Y key: +Yaws the object (through the Y axis).

y key: -Yaws the object (through the Y axis).

P key: +Piches the object (through the Z axis).

p key: -Piches the object (through the Z axis).

Descrição alto-nível das funções (partes lógicas)

`light(color, obj, alpha = 1.0):`

Essa função era encarregado de dar a cor `@color` (dada em `rgb`) para o próximo objeto a ser construído. O parâmetro `@obj` era utilizado para saber se o objeto em questão era o mesmo que a variável global `selected`, pois se fosse o programa deveria pintá-lo de branco, ignorando o parâmetro `@color`. Desta forma é possível identificar qual objeto na cena estaremos modificando pelas teclas.

`InitGL(Width, Height)`

Inicializa a tela do OpenGL com fundo preto e em uma perspectiva tal que seja coerente com o tamanho da tela (definido pelos parâmetros `@Width` e `@Height`).

`ReSizeGLScene(Width, Height)`

Trata do redimensionamento da janela, fazendo-a manter a perspectiva adequada para o novo tamanho `@Width x @Height`.

`DrawGLScene()`

Função principal de desenho. É nela que são desenhados o cone, esfera, cubo e chaleira com suas cores, tamanhos, posições e rotações desejadas.

`keyPressed(*args)`

Tratadora de eventos do teclado. Nela é feita as modificações dos parâmetros dos objetos devido à entrada especiais advindas da entrada padrão (teclado).

`mouseMoved(x,y)`

Protótipo para rotacionar a cena, infelizmente não consegui implementar os parâmetros corretos nessa função (talvez via `gluLookAt`).

`main()`

Inicializa a janela com o atributo *double buffered visual*, define-se o tamanho da janela como 640x480 e atribui-se as rotinas especiais para o OpenGL, como a função de desenho (`DrawGLSCENE`), a de redimensionamento (`ReSizeGLScene`) e as tratadoras de eventos advindas de entrada (`keyPressed` e `mouseMoved`).

Especificidades da minha solução

- Adotei o tamanho padrão dos objetos como 0.5 (atributo *Scale* de seus métodos construtores)
- Utilizei 20 cortes para a resolução do cone e da esfera.
- A posição padrão dos elementos é como sendo -1.5, -0.5, 0.5 e 1.5 no eixo X, 0 no eixo Y e -7 no eixo Z. Deste modo os objetos ficaram alinhados e igualmente simétricos em relação à origem.
 - OBS: todos esses valores são hardcoded e foram parametrizados.
- O menu é impresso na saída padrão.
- O objeto a ser modificado por transformações é o objeto pintado de branco.

Limitações:

Atualmente a minha solução não permite movimentação do ponto de observação (câmera).

Referências:

1 - <http://www.ic.unicamp.br/~helio/disciplinas/MC930/trabalho1.pdf>