24 de novembro de 2022

CIC270 - Computação Gráfica

Profª Elisa de Cássia Silva Rodrigues

Lista de Exercícios 2

(*Valor: 10,0 pontos - Peso 1,5 na nota N2*)

OBSERVAÇÃO:

*- Entrega vale **PRESENÇA** para a aula extra do dia **26/11** (sábado) - reposição da aula do dia 09/11.*

Descrição

Nesta atividade, o código apresentado nos arquivos **lista1.cpp** ou **lista1.py** (implementação da Lista 1 da disciplina), deverá ser modificado a fim de:

- Modificar o *array* de entrada do programa para permitir o desenho de alguma outra forma geométrica 2D de sua escolha que utilize triângulos em sua composição, por exemplo: polígono, estrela, seta, cruz, etc.
  - Para desenhar o objeto escolhido, utilize a primitiva mais adequada e crie o *array* corretamente de acordo com as instruções apresentadas no Módulo 6 da disciplina.
  - Caso o objeto escolhido permita alterar entre outras primitivas além de `GL_POINTS`, considere a ideia apresentada no exemplo `primitives.cpp` ou `primitives.py` do Módulo 6, que utiliza as teclas numéricas (**"1", "2"**, etc.) para troca entre primitivas.
- Modificar o *array* de entrada do programa para possibilitar a inclusão de uma cor específica para cada vértice do objeto, ou seja, cada vértice incluído no *array* DEVE conter 6 coordenadas (x, y, z, R, G, B). Para isso considere a ideia apresentada no exemplo `retangle2.cpp` ou `retangle2.py` do Módulo 6,
- Manter a opção para modificar o tipo de primitiva para `GL_POINTS` (nuvem de pontos) através das teclas **"V' ou "v"**.
- Manter as opções para modificar o tipo de preenchimento do triângulo através das teclas **"F' ou "f"**.
- Retirar a opção para modificar o número de triângulos, ou seja, a opção que altera o objeto entre triângulo e retângulo, que na Lista 1 era feita através das teclas "O' ou "o". Caso o objeto escolhido para ser desenhado permita essa opção, utilize as teclas **"H" ou "h"**.

- Incluir na função `keyboard()` as seguintes opções para realização de transformações geométricas do tipo:
  - **Translação:** as teclas abaixo devem definir a direção que o modelo será incrementalmente deslocado:
    "W' ou "w": deslocamento positivo em y
    "S' ou "s": deslocamento negativo em y
    "D' ou "d": deslocamento positivo em x
    "A' ou "a": deslocamento negativo em x
  - **Rotação:** as teclas abaixo devem definir o sentido que o modelo será incrementalmente rotacionado:
    "P' ou "p": rotação positiva (sentido anti-horário)
    "N' ou "n": rotação negativa (sentido horário)
  - **Escala (*zoom*):** as teclas abaixo devem definir a direção que o modelo será incrementalmente escalado:
    "I' ou "i": fator de escala maior que 1 (aumenta o objeto em ambas as direções x e y)
    "O' ou "o": fator de escala menor que 1 (diminui o objeto em ambas as direções x e y)

  OBSERVAÇÃO: *os valores de incremento para as translações, rotações e escalas devem ser escolhidos de forma que as transformações não sejam nem abruptas nem imperceptíveis visualmente. Rotações e escalas devem ser feitas usando o centro do modelo como referência (para não haver translações indesejadas).* OBS: as transformações geométricas foram discutidas na última aula atráves dos programas `transform.cpp` e `transform2.cpp`.

Implementação e Execução

As implementações devem ser feitas em linguagem C/C++ e compilar, respectivamente, usando os compiladores gcc/g++. Um arquivo Makefile deve ser disponibilizado para a compilação. A compilação deve ser feita usando o comando *make* em um terminal de comandos do Linux. Alternativamente, poderá ser usada a linguagem Python.

Em qualquer caso, o aplicativo deve executar no sistema operacional Linux usando a biblioteca "moderna" do OpenGL, com o uso de *shaders* para mostrar a visualização do modelo. *Não devem ser usadas construções como glBegin e glEnd das versões antigas do OpenGL!.*

O programa DEVE ter o nome "lista2.cpp'. A execução deve iniciada em um terminal de comandos do Linux, como no exemplo abaixo.

```
$ ./lista2
```

Caso seja implementado em Python, o programa deve ter o nome "lista2.py", sendo executado como mostrado abaixo em um terminal de comandos do Linux.

```
$ python3 lista2.py
```

Entrega

A entrega deverá ser realizada através do formulário indicado no SIGAA até às 23:59 horas do dia **30/11/2022** (quarta-feira).

- **Código:** submeter um arquivo .zip contendo o código (lista2.cpp ou lista2.py) e o Makefile, se for o caso. Se utilizar outras bibliotecas ou outro sistema operacional, incluir um arquivo README.txt.