

Trabalho 2 (peso 2,5)

Este trabalho tem por objetivo desenvolver um Sistema Simples Visualizador de Objetos 3D utilizando OpenGL. Para tal, as funcionalidades seguintes devem ser implementadas:

- Os três eixos principais X, Y e Z devem estar sempre presentes na janela de visualização, em que as cores para os eixos, respectivamente, são: verde, azul e vermelho.
- Visualizar objetos a partir de malha poligonal triangular, ou seja, triângulos carregados a partir de um arquivo.obj, cujo formato está descrito na seção 2 e cujo nome será passado como parâmetro pela linha de comando ao executar o programa numa janela DOS.
- Utilizar a projeção perspectiva para a visualização da curva abrindo uma janela de tamanho inicial 640x480 pixels que possa ser redimensionada livremente pelo usuário mantendo a proporcionalidade dos objetos.
- Exibir os objetos, porém a cor do objeto deverá ser diferente da cor do fundo da tela (ver Figura 1). O modo padrão de exibição é apresentar o modelo de arame objeto. A tecla P mudará a representação do objeto para um dos 2 modos possíveis, a saber: modelo de arame e modelo preenchido. No modo preenchido a cor das arestas dos polígonos deverá ser diferente da cor de preenchimento dos polígonos (veja a figura 1, em que os polígonos estão na cor azul e as arestas estão em preto). No modo aramado, apenas as arestas dos polígonos são exibidas.
- Permitir ao usuário ligar e desligar uma fonte de luz no ambiente. A tecla L liga e desliga a luz virtual. A posição da fonte de luz é livre, desde que fique claro na cena que ela está presente.
- Carregar, também, do arquivo .obj a sequência de transformações 3D que será executada quando o usuário apertar a barra de espaço. As transformações 3D serão executadas da seguinte forma:
 - Inicialmente será exibida na tela a figura original sem nenhuma transformação.
 - Assim que o usuário apertar a barra de espaço pela primeira vez, apenas a primeira transformação 3D será aplicada ao objeto.
 - Quando o usuário apertar a barra de espaço pela segunda vez, a transformação composta da primeira e da segunda transformação será aplicada ao objeto.
 - E assim por diante, até que todas as transformações 3D presentes no arquivo.obj tiverem sido aplicadas ao objeto.
 - Quando o usuário pressionar novamente a barra de espaço o objeto será exibido na tela novamente sem nenhuma transformação 3D.
 - Exiba também o objeto antes da transformação, porém o objeto deverá estar com uma cor diferente. Portanto, depois da aplicação da primeira transformação, sempre aparecerá dois objetos na cena, isto é, o objeto na posição final e a na posição anterior.

- Encerrar a aplicação ao ser pressionada a letra Q ou a tecla ESC. (-0,5 ponto se esta funcionalidade não estiver implementada).

O arquivo obj:

- Nunca conterá erros, portanto não precisa verificar um formato inválido.
- Contém uma malha poligonal triangular, ou seja, uma lista de vértices com coordenadas (x, y, z) seguida de uma lista de faces que apontam para os vértices que a compõe.
- # é um comentário no mesmo estilo do comentário de uma linha de C++ (//).
- Linhas que iniciem com v, indicam um vértice e são seguidas pelas suas coordenadas (x, y, z). As linhas que iniciem com f, indicam faces e são seguidas pelos vértices que a compõe.
- Linhas que iniciem com s, t, x, y, z, c ou e indicam uma transformação 3D seguida pelos parâmetros necessários. Veja o arquivo de exemplo abaixo para uma melhor compreensão.
- O arquivo obj abaixo ilustra uma malha poligonal triangular de um cubo, bem como algumas transformações 3D. O arquivo tem 8 vértices e 12 triângulos.

```
#-----
# cubo.obj
# Malha poligonal triangular para um cubo.
#-----
# Lista de vértices:
#-----
v -25 25 25
v -25 -25 25
v 25 -25 25
v 25 25 25
v -25 25 -25
v -25 -25 -25
v 25 -25 -25
v 25 25 -25
#-----
# Lista de faces:
#-----
f 1 2 3
f 8 7 6
f 4 3 7
f 5 1 4
f 5 6 2
f 2 6 7
f 1 3 4
f 8 6 5
f 4 7 8
f 5 4 8
f 5 2 1
f 2 7 3
#-----
# Transformações 3D:
#-----
s 2 2.5 3    #Escala
t 0 3 -3     #Translação
x 30         #Rotação de 30 graus no eixo X
```

```
y -45      #Rotação de -45 graus no eixo Y
z 22.6     #Rotação de 22.6 graus no eixo Z
c 2 3 0    #Cisalhamento no eixo Z com fator 2 em X e 3 em Y
c 0 2.5 4  #Cisalhamento no eixo X com fator de 2.5 em Y e 4 em Z
e 1 1 0    #Espelhamento em relação ao plano XY
e 1 0 1    #Espelhamento em relação ao plano XZ
e 0 1 1    #Espelhamento em relação ao plano YZ
```

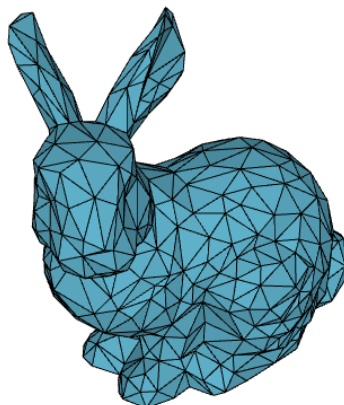


Figura 1: Visualização de um coelho com 1623 vértices e 1000 triângulos. Fonte: <https://www.meshlab.net/>

Sobre a entrega:

As equipes devem se organizar para elaborar um **desenho interessante e original**, ou seja, cada equipe terá seu próprio desenho. Para isso, o arquivo obj deve conter, pelo menos, 1000 triângulos. Vale ressaltar que será criada uma tarefa no Teams para as equipes entregarem o trabalho num único arquivo .ZIP contendo a pasta com todos os arquivos do projeto OpenGL utilizando o Visual Studio C++ 2022.