

Relatório de Computação Gráfica

Trabalho 3

Gabriel Henrique Campos Scalici 9292970

Keith T. Sasaki 9293414

Enunciado do trabalho: Confeccionar um ambiente 3D no OpenGL, de forma que algum dos objetos tenha sido importado de um modelo 3D, onde é possível realizar diversos movimentos, tanto na câmera (personagem em primeira pessoas) quanto no objeto carregado de um modelo 3D (terceira pessoa). Onde os movimentos podem ser feitos em todos os eixos.

Plataforma utilizada: O referido trabalho foi confeccionado no sistema operacional MacOS e Ubuntu(Linux). As IDEs utilizadas foram, o ambiente Xcode (no MacOS) e Sublime (no Linux), o qual fazem uso da linguagem C++ para a implementação do trabalho. Para compilar foi usado o botão play automático da IDE e também o uso do arquivo Makefile, para caso o usuário queira rodar via linha de comando.

Lembrando que o Makefile é importante pois junta a biblioteca ".c" com a main ".cpp"

Tecnologias externas: Como tecnologia externa, foi usada a função dada de exemplo em sala de aula para movimentação da câmera ao apertar teclas específicas (wasd).

Como tecnologia externa também, foi usado uma biblioteca disponível no github, do autor "felipecustodio" em linguagem C, que possui funções para que possa ser colocado o modelo 3D no projeto com uma simples implementação e deixando o código organizado.

Para o modelo 3D foi usado um site que disponibiliza alguns modelos para download, tal como sua textura. Site: sketchfab.com

Biblioteca SOIL baixada para poder colocar a textura em modelos 3D, porém não utilizada pela dificuldade em compilar.

Principais dificuldades: A principal dificuldade que tivemos durante a criação foi a parte de integrar e usar as funções das bibliotecas, pois há muitos detalhes na hora de colocar no arquivo principal, como local onde deverá ir

determinado trecho de código dentre outros.

Outra dificuldade foi a parte de colocar a textura, para não ficar somente o modelo 3D formado por triângulos, o uso das funções é relativamente simples, o que dificulta é entender o que está acontecendo, o que cada função está fazendo, para que possam ser feitas as modelagens de acordo com o projeto, como mudar textura quando o objeto é pressionado, dentre outras. Portanto optamos por não fazer, apesar de ter a biblioteca implementada.

Uma das dificuldades que esteve presente em praticamente todos os projetos, foram colocar as coordenadas dos vértices corretamente de forma que a primitiva final seja a desejada.

Divisão de tarefas: O trabalho inteiro foi construído praticamente junto, onde um realizava as pesquisas sobre as funções corretas (e explicações) e como fixar erros que impediam a compilação, principalmente porque trabalhamos em dois sistemas operacionais diferentes. Trocando de função diversas vezes durante o processo, para que pequenos erros e detalhes pudessem ser corrigidos, de acordo com o tempo disponível de cada um.

Resumidamente a área de código escrito:

Keith : Movimento do modelo, trabalho com cores, ajustes nas variáveis, base do código, pesquisas de textura e erros.

Gabriel: Importação das bibliotecas, movimentos da personagem, pesquisa dos modelos, arrumação do código, divisão em métodos, Makefile

Tempo: Para a realização deste trabalho foi preciso trabalhar por 2 dias, com um bom tempo dedicado ao projeto, por mais que o trabalho seja relativamente simples, demora um pouco para ganhar familiaridade com o ambiente 3D, as funções necessárias para visualização em 3D, dentre outros problemas enfrentados na hora da implementação.

Trecho de código que o grupo julga mais importante:

```
//Funcao para desenhar a maca de objeto 3D
void draw_apple(int ang1, int ang2, int ang3){
    glColor3f(1.0, 1.0, 1.0 );
    glTranslatef(0, 0, 0);
    glScalef(-0.15, 0.15, 0.15);
    //glRotatef(ang, rot_x, rot_y, rot_z);
    glRotatef(ang1, 1.0, 0.0, 0.0);
    glRotatef(ang2, 0.0, 1.0, 0.0);
    glRotatef(ang3, 0.0, 0.0, 1.0);
    drawObjWireframe(apple);
}
```

Parte que envolve a importação e modelagem do modelo 3D, adaptando à escala do projeto, fazendo as transformações necessárias para torná-lo visível, e também o trabalho de rotação onde há três rotações em todos os eixos.

Demonstração de funcionalidades:

Foi desenhado um "chão" na tela apenas para ficar mais fácil de visualizar a imagem andando para os lados.

Como usar:

Movimento da câmera/personagem primeira pessoa

W - para frente

S - para trás

A - para a esquerda

D - para a direita

R - Voar para cima

F - Voar para baixo

Movimento do objeto carregado de um modelo 3D (apple)

I - rotaciona para cima

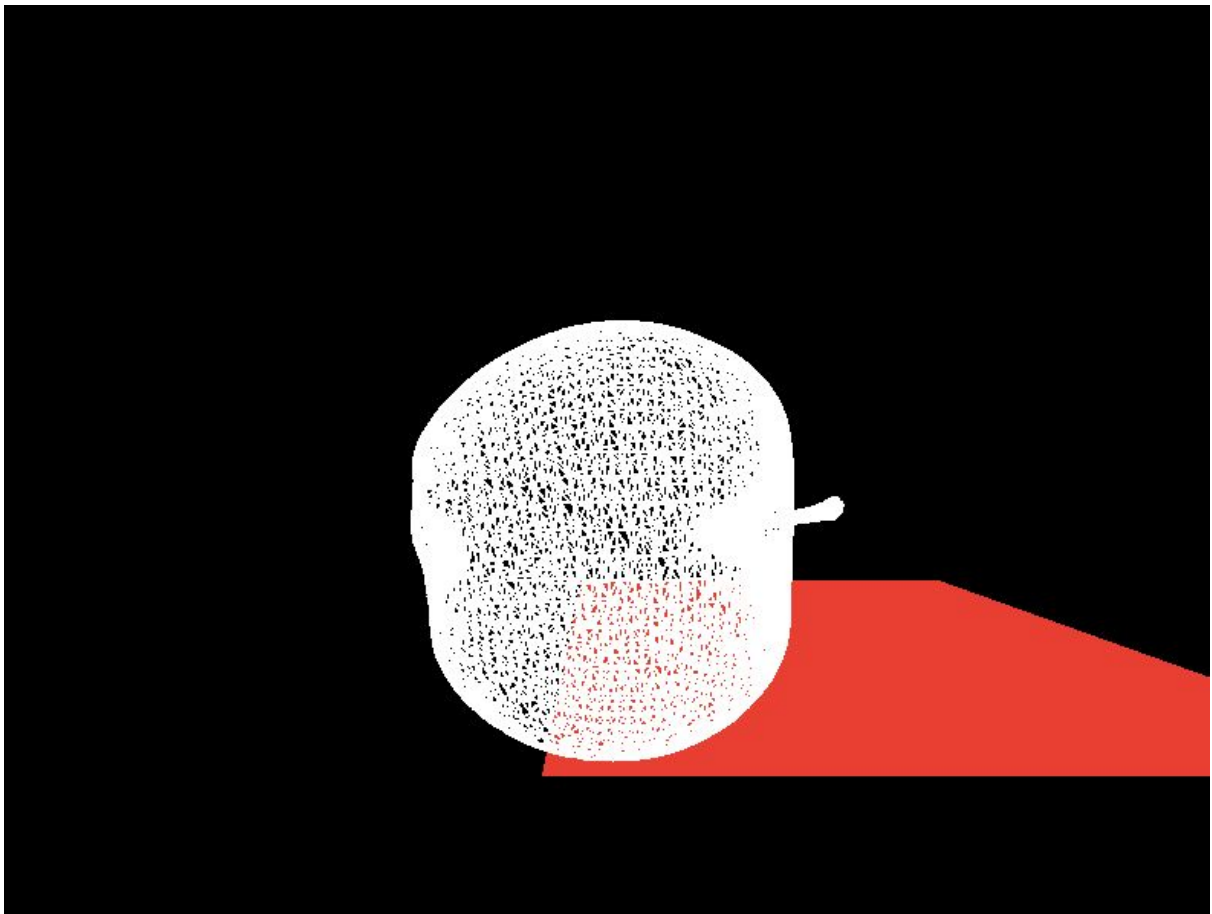
K - rotaciona para baixo

L - rotaciona para a esquerda

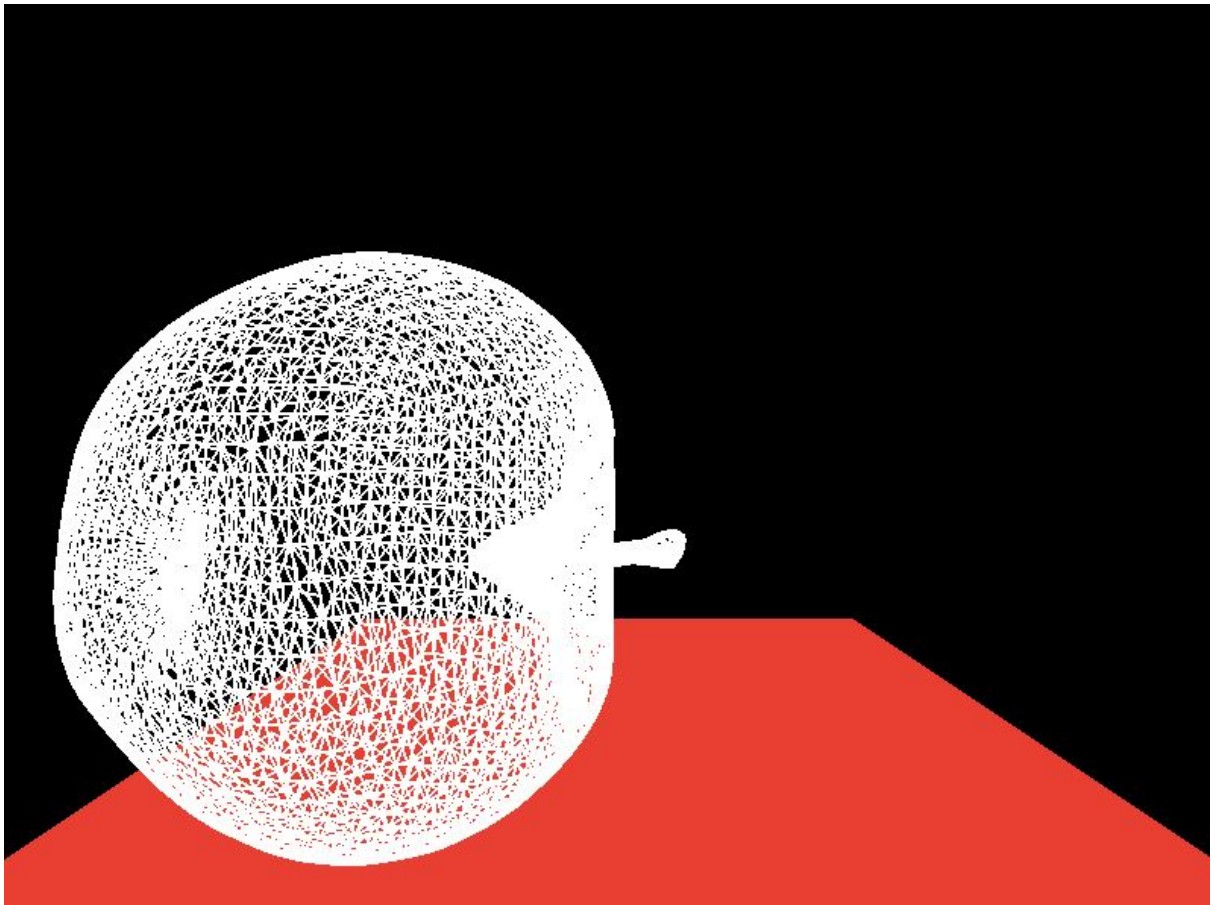
J - rotaciona para a direita

P - rotaciona para a direita no proprio eixo

O - rotaciona para a esquerda no proprio eixo



(Câmera andou para o lado e maçã rotacionada)



(Câmera andou para trás e maçã foi rotacionada)