

Relatório do Trabalho 2 - CIC270

Fernando José Ferreira Neto - 2018001665

Giovany da Silva Santos - 2018007758

Introdução sobre a aplicação desenvolvida

Neste trabalho desenvolvemos uma aplicação em 3D com OpenGL que utiliza dos movimentos de translação, rotação, além de utilizar escala de objetos e iluminação. Nesta aplicação, um diamante de cor azul gira e altera sua posição de altura sobre um cubo de mesma cor. O usuário pode mudar a visão da câmera para cima, ou jogar a câmera um pouco mais para a esquerda.

Apresentação das ferramentas, linguagens e bibliotecas utilizadas na implementação.

O código foi desenvolvido em C++ e as bibliotecas utilizadas foram as seguintes:

- `stdio`- Biblioteca que contém as funções básicas para os programas em C/C++.
- `glew` - fornece mecanismos eficientes de tempo de execução para determinar quais extensões OpenGL são suportadas na plataforma de destino.
- `glm` - possui as classes e métodos necessários para que seja possível trabalhar com translação, rotação e escala.
- `utils` - Armazena algumas funções úteis e básicas para os programas OPENGL, a exemplo funções como: definição de display, janelas, loop do programa .

Explicação dos detalhes importantes da implementação usando a biblioteca gráfica OpenGL.

- Imagens ou modelos 3D utilizados.

No nosso código existem dois modelos 3D, sendo eles um cubo e um diamante. A criação do cubo foi feita através de triângulos, onde para cada face temos 2 triângulos. As coordenadas dos triângulos são definidas no vetor `vertices2` na função `initData`, e na função `display` utilizamos a função `glDrawArrays(GL_TRIANGLES, 0, 36)` para desenhar o cubo. O diamante é composto por 2 pirâmides de tamanhos distintos, onde a que fica no sentido contrário é um pouco mais achatada para tentar fazer o modelo 3D se parecer mais com um diamante. As pirâmides são definidas por meio de 8 triângulos, sendo que no vetor `vertices` na função `initData` definimos as suas coordenadas, além disso usamos `glDrawArrays(GL_TRIANGLES, 0, 24)` na função `display` para desenhá-las.

- Transformações geométricas como translação, rotação e escala.

Entre as transformações geométricas usamos a rotação, rotação e a escala. A rotação foi usada tanto no cubo quanto no diamante. No cubo fizemos uma pequena rotação para que fosse possível ver suas 3 dimensões, já no diamante, utilizamos uma rotação constante no eixo y, para que ele fique “girando em torno dele mesmo”.

A translação também foi feita tanto no cubo quanto no diamante, no cubo fizemos uma translação única para posicioná-lo corretamente no lugar que desejado. No diamante é feita uma translação constante no eixo y, de modo que ele fique subindo e descendo continuamente. Para isso, definimos 3 variáveis (*sobe*, *altura* e *direction*), *sobe* define o quanto o diamante sobe no movimento de translação, a *altura* estabelece até que ponto ele desce e a variável *direction* define o sinal positivo na subida e negativo na descida. Para saber o valor de *direction* são realizados 2 estruturas de condições *if* na display como pode ser visto abaixo:

```
if(sobe>=altura)direction=-1;  
if(sobe<=-1.2)direction=1;
```

Se *sobe* for maior que a *altura*, ou seja, está acima do limite máximo que definimos para o diamante subir, então o *direction* é negativo. Se o *sobe* for menor que -1.2, ou seja, está abaixo do limite mínimo que definimos para o diamante descer, então o *direction* é positivo.

Ao saber o valor do *direction* verificamos utilizamos mais 2 estruturas para definir o quanto subir ou descer, incrementando ou decrementando a variável *sobe*:

```
if(direction==1)sobe+=0.005;  
else sobe-=0.005;
```

A escala foi feita somente no cubo, onde fizemos com que ele tenha 0.7 vezes o seu tamanho em todos os eixos, isso para melhorar a proporção de tamanho entre ele e o diamante.

- **Projeções ortográficas ou perspectivas.**

No desenvolvimento do nosso trabalho usamos projeção perspectiva para proporcionar uma melhor visualização dos modelos 3D que projetamos.

Para isso utilizamos a perspectiva com os seguintes parâmetros: *perspective(glm::radians(45.0f), (win_width/(float)win_height), 0.1f, 100.0f);*

- **Iluminação.**

Para a iluminação setamos a cor(Light color) para (1.0,1.0,1.0) nos comandos: *glGetUniformLocation(program, "lightColor")* e *glUniform3f(loc, 1.0, 1.0, 1.0)*. Essa cor foi aproveitada do código phong.cpp fornecido pela professora.

A posição da luz foi setada experimentalmente pelos comandos:

```
loc=glGetUniformLocation(program, "lightPosition") e glUniform3f(loc, 6.0, 19.2,  
6.0);
```

- **Funcionalidades da aplicação semelhante a um manual do usuário.**

O usuário tem as seguinte opções:

Seta para baixo: Vê a imagem 3D pela parte de cima

Seta para cima: Volta a câmera ao normal

Seta para esquerda: Movimenta a câmera um pouco para esquerda

Seta para direita: Volta a câmera ao normal

- **Conclusões sobre as dificuldades encontradas, problemas não resolvidos e conhecimentos adquiridos.**

Concluimos que o OpenGL possibilita o processamento de imagens 3D e que é possível trabalhar com rotação, translação, escala e iluminação dos objetos. Alguns dos problemas que não conseguimos resolver foram: Inicialmente nós planejamos fazer uma bola quicando em um cubo, porém o desenho da bola (esfera) não foi tão simples, e ao fazer pesquisas encontramos alguns materiais que utilizavam bibliotecas antigas. Além disso, outro problema encontrado e não resolvido, foi a translação da câmera, planejamos fazer com que a câmera girasse em torno do objeto, mas os objetos acabavam que por sumir da tela.

O trabalho foi bastante interessante pois aprendemos aplicar os conhecimentos de programação pura e matemática junto a aplicações de visualização. Foi algo bem novo, nos possibilitando a obter uma ideia de como funciona a animação de objetos.

- **Apresentação das referências utilizadas.**

<https://learnopengl.com/>

<https://pemavirtualhub.wordpress.com/2016/06/20/opengl-color-codes/>

Slides e códigos da Professora Elisa de Cássia Silva Rodrigues

ANEXO: Link para o vídeo:

<https://drive.google.com/file/d/1Bq4Lj8Ohjx3nBeiAisiAaXfw42oJ55Oe/view?usp=sharing>