

Relatório - Projeto 2 - Introdução à comp. Gráfica

Aluno: Nathan Rodrigo Cavalcante de Jesus

Matrícula: 20170159362

Aluno: João Hudson de Lacerda Oliveira.

Matrícula: 20170153752

Professor: Christian Azambuja Pagot

O projeto começou com foco no entendimento do papeline de acordo com os slides disponibilizados pelo professor Christian, para que pudéssemos entender cada matriz e sua função e como faríamos para montá-las e multiplicá-las no decorrer do programa.

Bibliotecas utilizadas na construção do programa:

- **Assimp**
 - Utilizada para importação dos vértices da Suzane (macaca)
- **GLM**
 - Biblioteca de operações matemáticas
- **Freeglut**
 - Biblioteca do OpenGL utilizada apenas para criar a tela

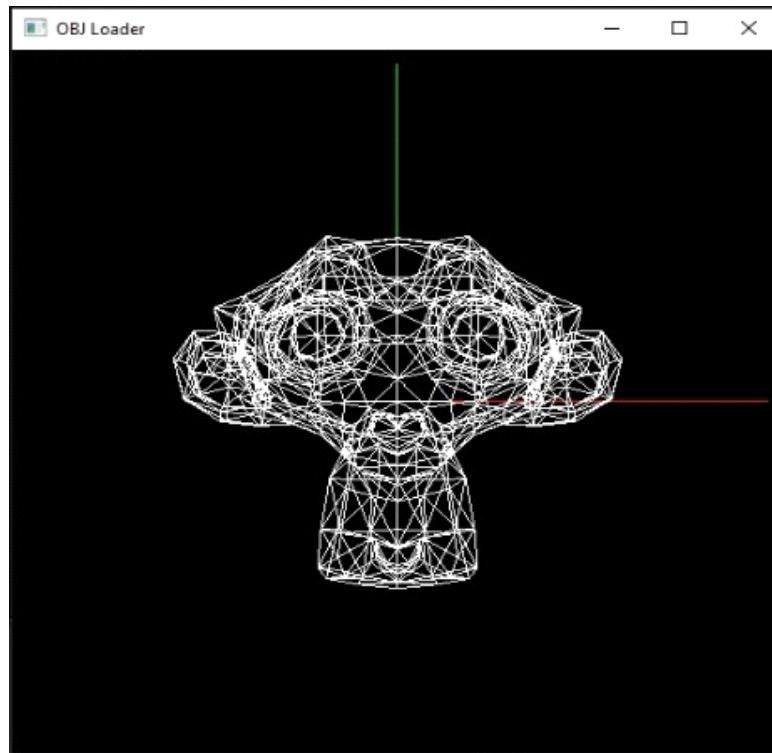
Na implementação:

Como orientado pelo professor, utilizamos a **Projeto 1**, que tinha como objetivo a rasterização de linhas e triângulos.

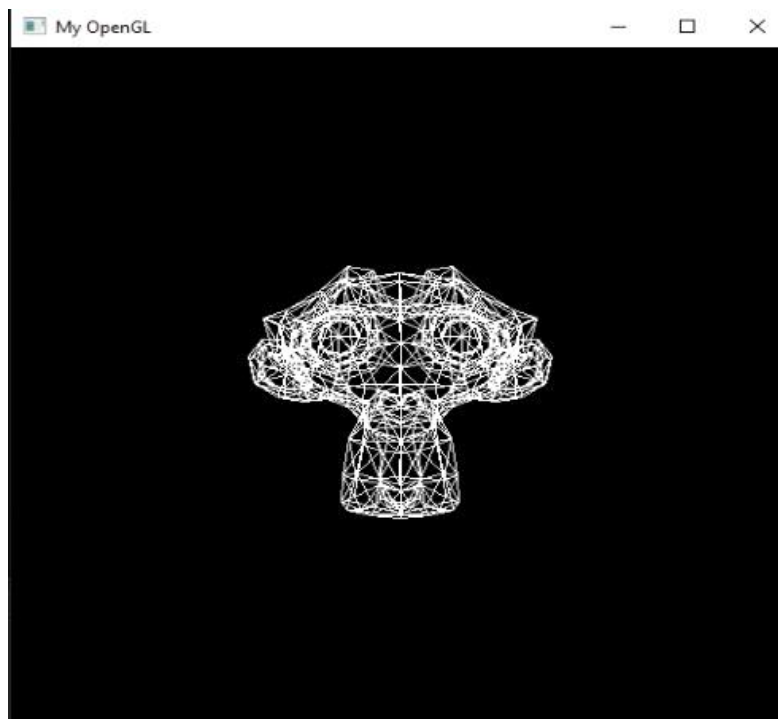
O programa foi feito sem a utilização de classes, apenas cabeçalhos auxiliares, como os arquivos "mygl.h" e "pipeline.h". Cada passo do uso das matrizes foi inspirada no programa exibido em sala de aula pelo professor e em relatórios encontrados na internet.

Apesar de nos ter sido fornecida uma biblioteca pelo professor (obj_loader e obj_parser), não conseguimos passar as coordenadas dos vértices para o armazenador de vetores da biblioteca glm, por isso utilizamos a Assimp, onde encontramos

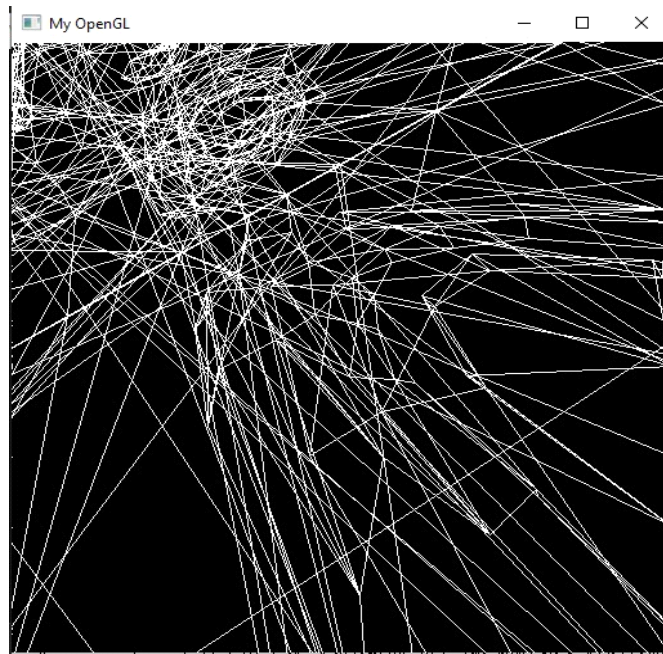
Resultado obtido a partir do carregamento de malhas com OpenGL:



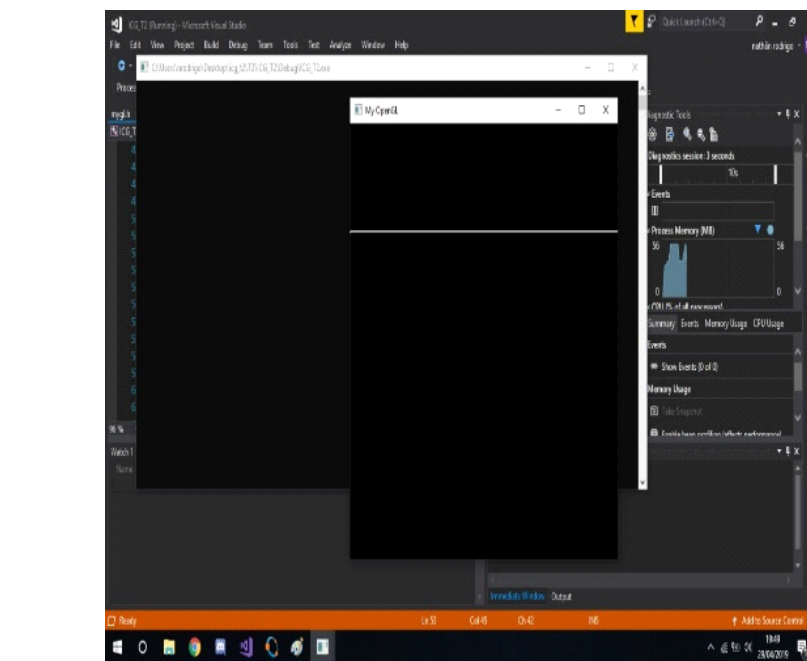
Resultado do nosso papeline:



Alguns bugs encontrados ao longo do caminho:



(Suzane vista por dentro e distorcida devido a um erro na projeção)



Primeiras matrizes e iniciação dos vetores da câmera:

```
glm::mat4x4 S1 = glm::mat4x4( // Inverte eixo Y
    1, 0, 0, 0,
    0, 1, 0, 0,
    0, 0, 1, 0,
    0, 0, 0, 1);

glm::mat4x4 M_model = glm::mat4x4(
    1, 0, 0, 0,
    0, 1, 0, 0,
    0, 0, 1, 0,
    0, 0, 0, 1);

M_model = M_model * S1;

/*Parametros da camera*/
glm::vec3 camera_pos = glm::vec3(0, 0, -3);
glm::vec3 camera_lookat = glm::vec3(0, 0, 0);
glm::vec3 camera_up = glm::vec3(0, 1, 0);
```

Possíveis melhorias:

- Acrescentar matriz de rotação;
- Seguir o padrão do eixo z para fora da tela(ele ficou invertido);
- Refatorar o código para fornecer rotinas de desenhos reutilizáveis;
- Acrescentar recursos extras como: textura e iluminação;
- Fornecer estruturas de dados mais abstratas com métodos de desenho simplificado para renderização dos objetos.

Algumas dificuldades encontradas foram:

- Entendimento das matrizes referentes a passagem do objeto para o espaço da tela e canônico;
- A combinação de todas as matrizes de forma correta.

Referências:

<https://stackoverflow.com/questions/43610347/cant-access-correct-number-of-vertices-for-a-3d-model-obj-using-assimp>

<http://luanareiscg.blogspot.com/2016/05/pipeline-grafico.html>

Todos os slides da primeira unidade passados pelo professor via SIGAA.