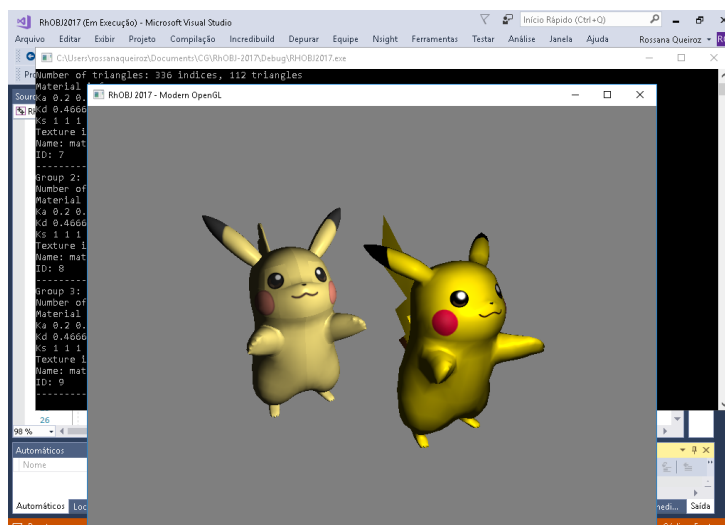


Trabalho Prático – Computação Gráfica

LEITOR/VISUALIZADOR DE CENAS 3D COM OPENGL MODERNA – Versão 1.0



O objetivo deste trabalho, ao longo de toda atividade de Computação Gráfica, é desenvolver um leitor e visualizador de cenas tridimensionais aplicando os conceitos vistos em aula. Para o Grau A, os conhecimentos sobre câmera sintética, modelagem 3D e transformações nos objetos serão aplicados. Nesta etapa, o programa deve possuir, como requisitos mínimos:

1. *Parser* do arquivo `.obj` ou `.ply`: a leitura pode ser feita utilizando a biblioteca [Assimp](#) (Asset Importer) **ou implementada diretamente em C++**
 - vértices e faces
 - apenas triângulos (pois o shader só processa triângulos)
 - normais (ainda não serão usadas, mas já devem ser armazenadas)
 - coordenadas de textura (ainda não serão usadas, mas já devem ser armazenadas)
 - grupos (também chamados de *mesh* ou *submesh* do objeto)
2. *Parser* dos arquivos com o código dos *shaders* (pode simplesmente implementar do tutorial)
3. Correta criação e utilização dos *buffers*
 - VAO, VBO(s) e EBO (este último, opcional)
 - Lembre-se de criar um VAO para cada grupo
4. Controle do modelo 3D (com input de teclado e/ou mouse):
 - Rotação em x, y, e z
 - Translação em x, y e z
 - Escala (pode ser o mesmo fator de escala nos 3 eixos para manter a proporção)
 - Seleção dos objetos (acrescentando um efeito na cor)

5. Controle de câmera em primeira pessoa (com input de teclado e/ou mouse)

EXTRAS (opcionais):

- Já acrescentar as texturas (similar ao visto em PG, mas com as coordenadas recuperadas do arquivo)
- Interface gráfica com menu, botões e *sliders* (usando, por exemplo a [imgui](#))

OBSERVAÇÕES:

- **TRABALHO INDIVIDUAL OU DUPLAS** - mesmo que vocês façam em colaboração com outros colegas (é totalmente OK se ajudarem), cada dupla deve entregar o seu leitor. No dia da entrega vamos ter um momento para cada um me mostrar o trabalho, e eu farei algumas perguntas sobre o código para cada um. A sincronia entre a dupla (ambos tendo domínio) é parte fundamental da avaliação, podendo causar diferenças nas notas entre os participantes.
- APENAS REFORÇANDO: é totalmente OK e recomendável seguir os tutoriais (learnopengl, anton, opengl-tutorials etc) para desenvolver o trabalho. Como é um trabalho "padrão", não tem muito como ter códigos super diferentes dos tutoriais. A arquitetura das classes pode variar, mas o mapeamento de vértices, seus atributos e a topologia da malha com os buffers sempre será parecida. O importante é que vocês ENTENDAM como o pipeline gráfico programável funciona!!!

BOM TRABALHO! ☺

Entrega via Moodle (conferir data no sistema)