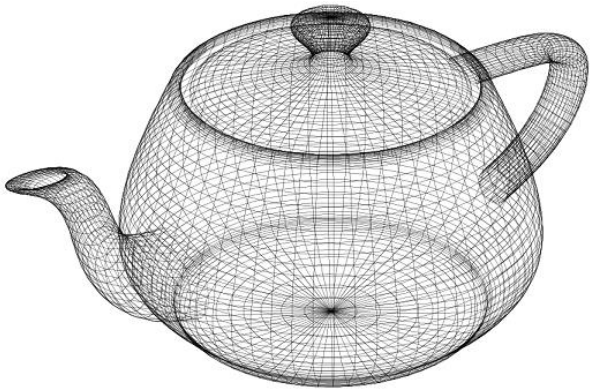
Universidade Federal de Santa Maria
Curso de Ciência da Computação
Disciplina: Computação Gráfica Avançada
Segundo semestre de 2016
Prof. Cesar Tadeu Pozzer

Trabalho 1 – The Utah Teapot



Descrição

Desenvolva um programa em C++, utilizando a API OpenGL 4.x, para renderizar o famoso Utah Teapot. Deve-se utilizar VAO e VBO. A coordenadas dos vértices podem ser encontradas na internet. O objeto deve ter normais que podem ser passadas pela aplicação ou calculadas no Geometry Shader. Deve-se aplicar algum tipo de iluminação ao objeto, via Fragment Shader.

O programa deve permitir movimentação da câmera. Deve ter controle de FPS.

Objetivos

- Explorar programação de shader
- Explorar a API OpenGL
- Explorar algoritmos avançados

Bônus

- Algoritmos de iluminação avançados, como reflexão
- Uso de Geometry shader para cálculo de normais
- Uso de tesselation para refinamento da malha
- Adição de sombra
- Etc.

Ferramentas: Linguagem C++, compilando com Visual Studio 2015. Não podem ser utilizadas bibliotecas auxiliares. Deve ser usada a API OpenGL 4.x.

Data e Formato de Entrega

- Data: 06/set/2016, horário de expediente. O trabalho deve ser apresentado individualmente.
- No email e no cabeçalho do arquivo devem conter o nome completo do aluno.
- O arquivo deve ser enviado para pozzer3@gmail.com, mdalcin@inf.ufsm.br com o subject "CGA T1".
- Deve-se enviar fontes e o projeto para o compilador **Visual Studio 2015**. Envie **apenas** os arquivos **de projeto, código fonte e modelos.**
- O programa deve ser enviado em um arquivo compactado fulano.rar (fulano = login ou nome do aluno). Dentro deste arquivo deve haver um diretório com o mesmo nome do arquivo e dentro deste diretório os arquivos do trabalho.
- Ex: o arquivo pozzer.rar deve conter um diretório chamado pozzer, e dentro do diretório devem estar os arquivos do trabalho.

Critério de Avaliação

- Documentação: descrever no cabeçalho de cada arquivo a ideia geral do código e detalhes específicos de partes que mereçam uma explicação – não comente por exemplo o que faz b++.
- README.txt: incluir um arquivo "README.txt" contendo informações sobre quais funcionalidades foram implementadas (requisitos e extras).
- Pontualidade: Trabalhos não entregues na data não serão avaliados e receberão nota zero.
- Legibilidade: nome de variáveis, estruturação do código. O código digital a ser entregue deve ter 4 espaços de identação e não deve possuir tabulações.
- Clareza: facilidade de compreensão – evite códigos complexos e desnecessários. Adote a solução mais simples possível.
- Funcionalidade: o programa deve satisfazer todos os requisitos. Programas que não compilarem ou que não atenderem nenhum requisito receberão nota 0 (zero).

Você pode discutir estratégias e ajudar o colega na implementação, porém evite passar código fonte. Programas semelhantes terão nota 0 (zero).