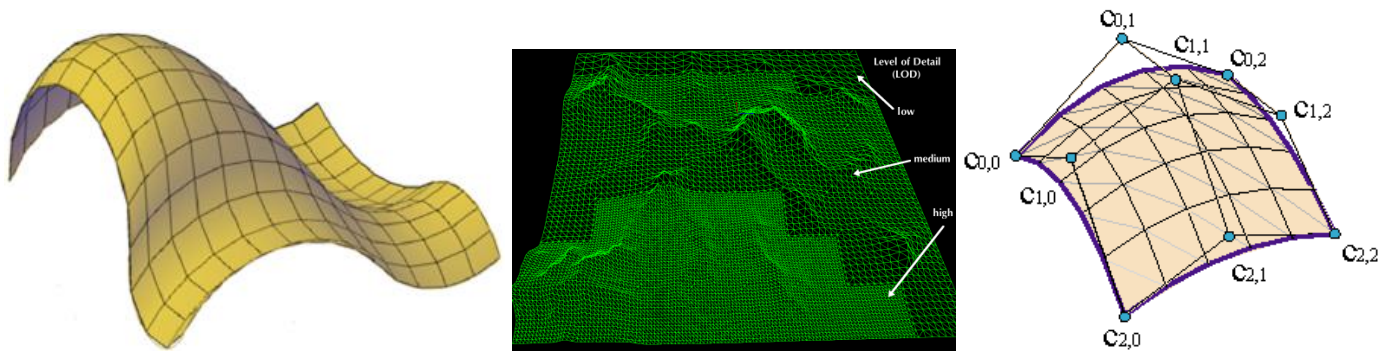


Trabalho 1 - Curvas B-Spline 3D e LOD



Descrição

Desenvolva um programa em C++, utilizando a API OpenGL, para renderizar uma superfície B-Spline. A definição da superfície deve ser definida por 10x10 pontos de controle. A visualização da superfície deve ser por malha em wireframe e com preenchimento.

Deve-se desenvolver/aplicar algum tipo de LOD nessa superfície. Pode-se utilizar a distância da câmera como heurística de resolução.

A solução deve permitir o refinamento da malha da superfície (valor fixo), em etapa de pré-processamento.

Objetivos

- Explorar o uso de vetores e suas operações
- Explorar o uso de curvas
- Explorar interação com o mouse e teclado.
- Explorar a API OpenGL
- Explorar algoritmos de LOD.

Requisitos Mínimos

- O usuário deve poder controlar a câmera (colada a superfície ou não)
- Exibir grafo de controle
- Exibir pontos de controle
- Deletar o código das callbacks não utilizadas! Deletar as classes não utilizadas! (caso usar o SCV)
- Opção para ativar/desativar LOD.
- Explorar T-Vertex.

Bônus

- Complexidade do algoritmo de LOD.
- Heurística de LOD
- Usar quadtree
- Etc.

Data e Formato de Entrega

- Data: 28/set/2015, horário de expediente. O trabalho deve ser apresentado individualmente.
- No email e no cabeçalho do arquivo, devem conter o nome completo e matrícula do aluno.
- O arquivo deve ser enviado para pozzer3@gmail.com, afrasson@inf.ufsm.br, tengel@inf.ufsm.br com o subject "CGA T1".
- Deve-se enviar fontes e o projeto para o compilador **Visual Studio 2013** ou **Codeblocks**. 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 indentação 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 a nota 0 (zero).