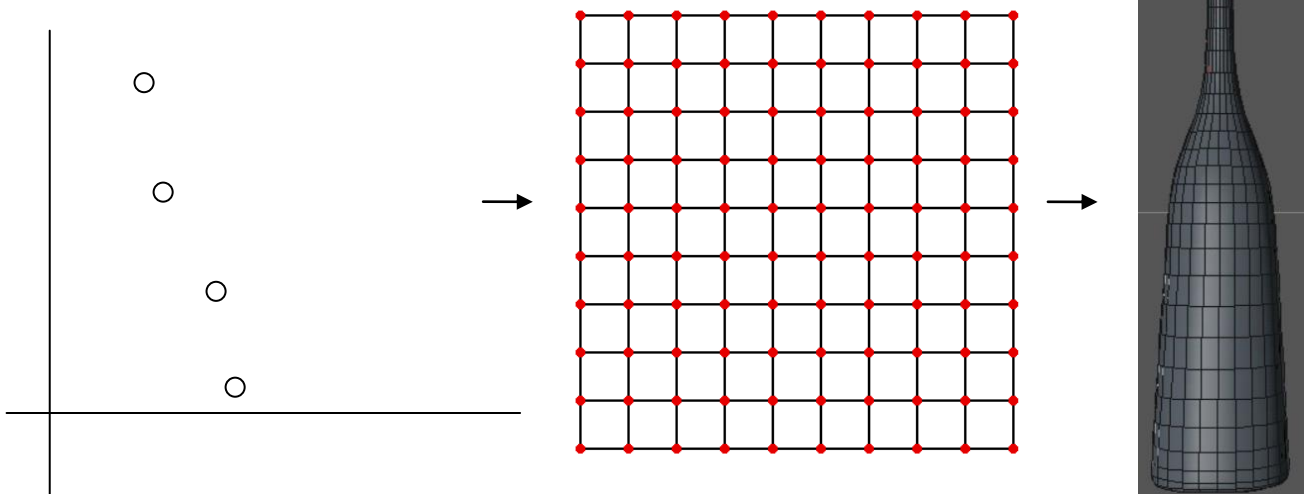


Trabalho 3- Geração Objeto 3D com Sweep e Curvas

Implemente um programa para fazer modelagem e visualização 3D de objetos gerados sweep rotacional e curvas de Bezier.

Linguagem C++, utilizando a API Canvas2D (disponível no site da disciplina) e IDE Code::Blocks, compilando com MinGW (disponível na versão 17.12 da IDE Code::Blocks).

O usuário deve informar uma sequência de 4 pontos de controle. A partir desses pontos, deve-se gerar uma curva de Bezier (com N pontos amostrados) que deve ser rotacionada (sweep rotacional) para geração do objeto tridimensional.



Os pontos de controle podem ser editados a qualquer momento, e deve refletir instantaneamente na forma do objeto gerado.

Requisitos básicos (Max: 9.0)

- Modelagem do objeto
- Animação (rotação)
- Visualização ortográfica e perspectiva (sob vários ângulos) em wireframe
- Parametrização do número de pontos e faces do objeto gerado.

Avançados

- Exibir vetores normais em cada face (até 1 ponto)

- Adição de mais de 4 pontos de controle (até 1 ponto)
- Remoção de faces ocultas – sem preenchimento (até 2 pontos)
- Preenchimento de polígonos com Iluminação por vértice (até 4 pontos)
- Preenchimento de polígonos com Iluminação por pixel (até 5 pontos)
- Etc.

O trabalho deve apresentar uma lista de instruções, explicando de forma como o usuário deve interagir com o programa. Enumere no início do código fonte (arquivo main.cpp) os quesitos que foram implementados.

Data e Formato de Entrega:

- Data: 05/06 até as 23:59. **Após esse prazo o trabalho será desconsiderado.**
- 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 e adrian.k.47@hotmail.com com o subject “CG T3”.
- 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. Deve-se enviar fontes. Não envie arquivos com extensão exe, obj, libs, etc. **Envie somente o que for necessário para compilação e o projeto Code::blocks.**
- 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.
- 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).