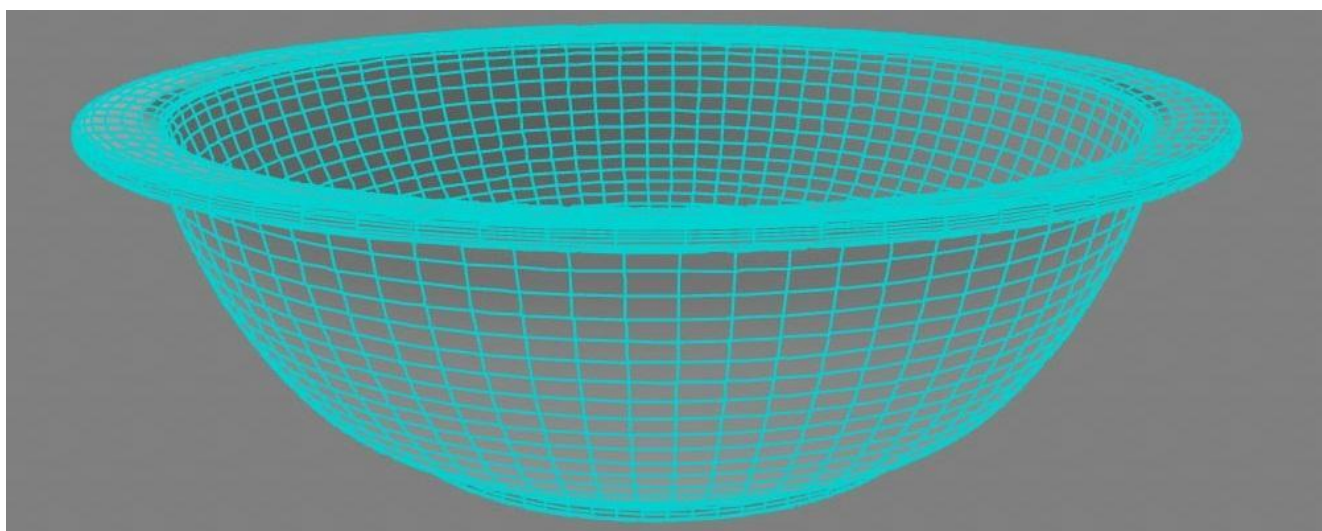


Trabalho 3 – Visualização de Objetos em 3D

Descrição: Implemente um programa em C++ para fazer a modelagem e visualização de um objeto 3D, representado por meio de sweep, usando uma câmera sintética. Deve-se criar uma interface **amigável** de interação (mouse, teclado, menus, etc). A visualização deve ser por meio de projeção perspectiva e ortográfica (selecionável).



O trabalho tem vários níveis de complexidade, e são incrementais. Somente pode ser feito o próximo nível após os anteriores (exceto para o etc):

- 1) Fazer a geração de um objeto por sweep rotacional, onde o usuário pode clicar com o mouse para definir a forma do objeto. Os pontos podem ser movidos pelo usuário. Com isso pode-se gerar objetos cilíndricos como copos, vasos, etc. Deve-se poder visualizar usando pontos e linhas, com opção para projeção ortográfica e perspectiva (wireframe) – **Nota 9**
- 2) Fazer também a geração de uma mola, também por sweep (rotacional e transacional). O usuário deve definir como é a forma da seção transversal da mola. Deve-se poder visualizar usando pontos e linhas (wireframe) (**2 pontos de Bônus**)
- 3) Permitir que o usuário modele o perfil do objeto por meio de uma curva Bezier 2D (ou bspline). Devem ser usados no mínimo 4 pontos de controle (para bezier). Os pontos podem ser movidos pelo usuário. Para a visualização, deve-se amostrar a curva em pelo menos 20 pontos. Após aplique sweep rotacional sobre os pontos amostrados. (**2 pontos de Bônus**)
- 4) Implementar funções de rotação e translação para melhor visualizar o objeto (**1 ponto de Bônus**)

- 5) Fazer a remoção de polígonos não visíveis (os polígonos que estão atrás do objeto) da malha em wireframe. (2 pontos de bônus)
- 6) Aplicar sobre os objetos gerados alguma forma de preenchimento de polígonos com iluminação. Não pode ser usada a API OpenGL. (5 pontos de bônus)
- 7) Etc. (x pontos de bônus)

Deve-se utilizar a API Qt ou o demo do canvas para implementar o trabalho.
<https://www.qt.io/developers/>.

Dica: faça inicialmente a visualização com projeção em perspectiva de um cubo centrado na origem. Faça a projeção sem o uso de matrizes de transformação. Crie uma função `void projeta(float x, float y, float z, int *xt, int *yt)` que retorna as coordenadas de tela x_t e y_t para cada ponto dado em função do centro de projeção da câmera. Assuma a existência de 1 ponto de fuga. Quando isso funcionar, comece a implementar o resto.

Data e Formato de Entrega:

- Data: 14/062016.
- 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, mdalcin@inf.ufsm.br e gbackes@inf.ufsm.br com o *subject* “CG T3”. Deve-se enviar fontes e o projeto para o compilador **Code::Blocks** (canvas), **Visual Studio 2013** (Qt) ou projeto do **Qt Creator** (preferencial)
- 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 somente os fontes e o arquivo do projeto. **Envie somente o que for necessário para compilação.** Não deve ser enviada as libs do canvas e nem DLLs em geral. **Deve-se enviar o arquivo do projeto, juntamente com os códigos fonte.** Caso for usada alguma ferramenta de geração de código, os arquivos devem ser copiados para `...\src`.

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++`.
- **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).