

## Trabalho 3- Geração de Engrenagens 3D

Implemente um programa para fazer modelagem e visualização 3D de engrenagens.

**Ferramentas:** Linguagem C++ com a API Qt (Versão 5.7.1), ferramenta QT Creator, utilizando as funções de desenho da Canvas2D – demo CanvasQT, compilando com MinGW (disponível na versão 16.01 da IDE Code::Blocks). <https://www.qt.io/developers/>. **Não podem ser utilizadas bibliotecas auxiliares.** Não pode ser usada a API OpenGL.

Vários tipos de engrenagens podem ser modelados:

- a) Dente com lado plano (Spur gear),
- b) Dente com lado curvado (Spur gear),
- c) com raios distintos (Spiral bevels),
- d) com dentes curvos (Helical gear),
- e) dentes curvos e raios distintos (bevel gear)



(básicas)



(avançadas)

Vários parâmetros podem ser definidos:

- a) Raios
- b) Número de dentes

- c) Angulação dos dentes
- d) Número de faces de cada dente
- e) Etc.

#### Requisitos básicos (Max: 9.0)

- Modelagem de engrenagem tipo spur gear
- Animação (rotação)
- Visualização ortográfica e perspectiva (sob vários ângulos) em wireframe
- Parametrização do número de dentes e raios

#### Avançados

- Exibir vetores normais em cada face (até 1 ponto)
- Demais tipos de engrenagens (em ordem de complexidade, até 1 + 2 + 3 pontos)
- Conectar duas engrenagens iguais com movimentação, com eixos conectados (até 2 pontos)
- Preenchimento de polígonos com iluminação por vértice (até 4 pontos)
- Preenchimento de polígonos com iluminação por pixel (até 5 pontos)

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: 02/06 até as 23:59.
- 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](mailto:pozzer3@gmail.com) e [rtrindade@inf.ufsm.br](mailto:rtrindade@inf.ufsm.br) 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, projeto Qt Creator. Não envie arquivos com extensão pdb, ilk, idb, obj, etc. **Envie somente o que for necessário para compilação.** Não devem ser enviadas as libs.
- 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).**