

Trabalho 4 - Geração de Engrenagens 3D

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](#)). **Não podem ser utilizadas bibliotecas auxiliares. Não pode ser usada a API OpenGL.** A API canvas2D pode ser customizada, bem como a classe Vector2.

Implemente um programa para fazer modelagem e visualização 3D de engrenagens **em wireframe**.
Uma dica importante: Faça uma versão 2D básica antes de fazer a versão 3D.

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)



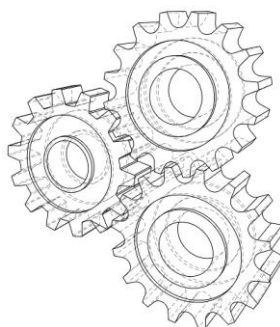
(básicas)



(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:

- O arquivo deve ter o nome do aluno (ou uma abreviação nome+sobrenome, ou seja, algo que identifique bem o aluno). O arquivo deve ser enviado pelo Google Classroom.
- 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: código fonte (.cpp, .h, .hpp) e o projeto (.cbp). Não devem ser enviadas libs, executáveis, DLLs.
- Antes do envio, certifique-se de que o projeto contido no seu .rar funciona em qualquer diretório que ele seja colocado e não dependa de outros arquivos não incluídos no envio 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).