

Universidade Federal de Santa Maria
Curso de Ciência da Computação
Disciplina: Computação Gráfica
Primeiro Semestre de 2024
Prof. Cesar Tadeu Pozzer
Data: 18/06/2024

Trabalho 4 - Bomba

Ferramentas

Linguagem C++, utilizando a API Canvas2D (disponível no site da disciplina) e IDE Code::Blocks, compilando com MinGW 32 bits. **Não podem ser utilizadas bibliotecas auxiliares.** Desenvolva o trabalho sobre o demo 1_CanvasGlut ou 2_CanvasGlfw. Antes de enviar, retire todas as funções e arquivos não utilizados. Se alguém fizer em Linux, deve ajustar todos os paths para execução em Windows. Teste uma máquina Windows antes de enviar. O melhor neste caso é rodar o Windows em uma máquina virtual, como o Virtual Box ou Parallels.

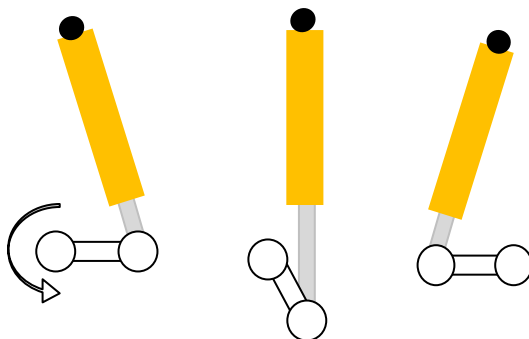
A Canvas2D pode ser customizada, porém não é permitido o uso de funcionalidades OpenGL que não estão presentes na Canvas2D (ex: texturas, shaders). Pode-se criar novas sobrecargas de funções, novos métodos, classes, enums, templates, etc.

Objetivo do trabalho:

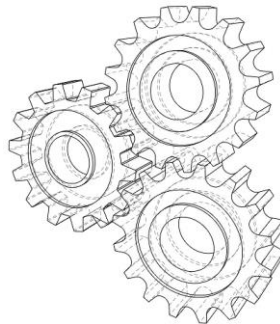
Desenvolver raciocínio matemático de vetores e operações. Aplicar transformações de visualização 3D. Modelagem de objetos 3D.

Descrição

Implemente um programa para fazer modelagem e visualização de uma bomba, composta por um pistão, um cilindro e um virabrequim. A visualização deve ser de duas formas: 2D e 3D, como mostrado nas seguintes figuras. A medida que o virabrequim gira, o pistão se movimenta dentro do cilindro. O cilindro é fixado em uma extremidade e sua direção muda conforme o virabrequim gira. (8 pontos)



Conectado ao Virabrequim, deve haver uma engrenagem tocando outra engrenagem. A relação de dentes deve ser observada na velocidade das duas engrenagens. (2 Pontos)



Deve-se **inicialmente** fazer a versão 2D para entender como calcular a direção do cilindro, pistão e movimentação do virabrequim. Deve-se utilizar vetores e suas operações. A ligação do virabrequim com o pistão deve ser perfeita. Se a matemática estiver errada não vai funcionar. Desenvolva a solução matemática no papel para então implementar. A posição do pistão é dada pela rotação do eixo, que deve ser constante.

O programa deve permitir a seleção do que vai ser exibido. Ex: virabrequim, cilindro, pistão, etc. O vídeo baixo mostra o virabrequim e o pistão em movimento. O trabalho é parecido com isso. O que muda é que o cilindro se move.

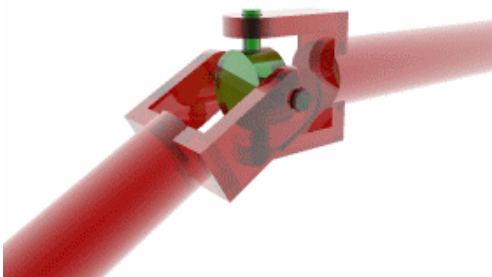
https://www.youtube.com/watch?v=DKF5dKo_r_Y,

Requisitos básicos (Max: 10.0)

1. Rotação do eixo variável (RPM informada pelo usuário).
2. Visualização ortográfica e perspectiva (sob vários ângulos) em wireframe.
3. O trabalho será compilado em Windows.

Avançados

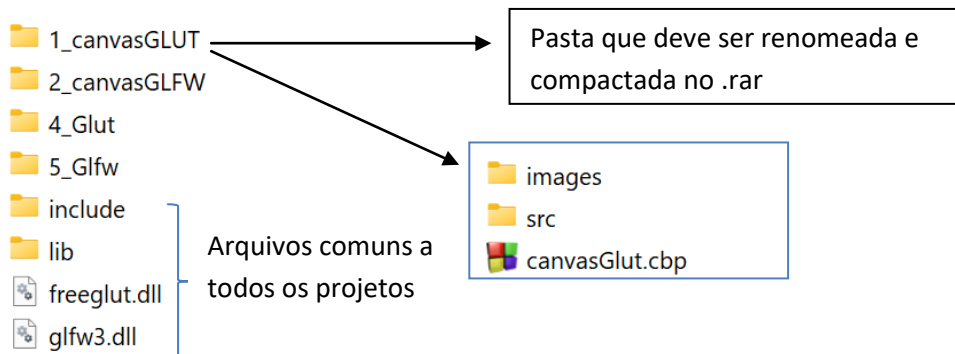
1. Exibir vetores normais em cada face (até 1 ponto)
2. Adição de detalhes aos componentes (até 2 pontos)
3. Preenchimento de polígonos com iluminação por pixel (até 5 pontos)
4. Adicionar eixo cardã conectado ao virabrequim. O ângulo do eixo deve poder ser ajustado em tempo real, entre 0 e 45 graus. (até 6 pontos) https://pt.wikipedia.org/wiki/Eixo_card%C3%A3
5. 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.

Formato de Entrega:

- O trabalho deve ser entregue pelo Google Classroom.
- Deve-se utilizar como base o projeto 1_canvasGLUT disponível nos demos da disciplina, como ilustrado na seguinte figura.



- A pasta 1_canvasGLUT tem todos os códigos fonte e recursos (images, src e projeto). Esta pasta **deve ser renomeada** com o nome do aluno. Ex: Trab1Maria, Trab2Paulo, Trab3Pedro, Trab4JoaoPedro, etc. Esta estrutura vai facilitar a execução e correção dos trabalhos. Todos os arquivos do trabalho devem estar dentro desta pasta, que deve ser a única pasta enviada, compactada em formato .rar, cujo nome deve ser o nome do aluno. Ex: FulanoSobrenome.rar. **Os caminhos relativos para as pastas include, lib e para as dlls devem ser mantidos, e no padrão Windows.**
- Esta estrutura de pastas não pode ser modificada.
- **Não devem ser enviadas lib, exe, obj, DLL, pdf, doc.**
- Retire todo código não utilizado no trabalho (arquivos, métodos, variáveis, etc), bem como printiefis de depuração.
- O trabalho será compilado em Windows .
- Perderá muita nota o trabalho que não seguir essas regras.

CrITÉrios de avaliação:

- Documentação: descrever no cabeçalho de cada arquivo a ideia geral do código e comentar o que cada método e classe faz.
- Clean code: estrutura do código e nomeação de métodos, classes e variáveis devem ser fáceis de ler e entender. Procurar manter o código o mais simples e organizado possível. Utilizar diferentes arquivos para diferentes classes.
- Pontualidade: Trabalhos não entregues na data não serão avaliados e receberão nota zero.
- 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).