

Universidade Federal de Santa Maria
Curso de Ciência da Computação
Disciplina: Computação Gráfica
Primeiro Semestre de 2024
Prof. Cesar Tadeu Pozzer
Data: 20/03/2024

Trabalho 1 – Editor de imagens

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.

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, etc.

Objetivos

Desenvolver habilidades na programação e manipulação de:

- Estruturação de código: Estruturas e/ou Orientação a objetos.
- Loop de renderização
- Componentes de interface gráfica e interação
- Manipulação de Cores
- Eventos de mouse
- Eventos de teclado
- Sistema de coordenadas

Descrição

Desenvolva um programa em C++ para fazer manipulações em imagens. O programa deve:

- Carregar imagens colorida (RGB) em formato BMP. Utilize imagens retangulares pequenas (entre 200 e 300 pixels) para facilitar a exibição dentro da canvas2D.
- Mover cada imagem individualmente dentro da canvas2D com uso do mouse. Deve-se ter o conceito de imagem corrente selecionada para que sejam aplicados processamentos sobre ela. Pode-se indicar a imagem selecionada com uma moldura. A imagem selecionada deve aparecer na frente das outras.
- Reexibir a imagem selecionada, em uma quarta imagem, das seguintes formas:
 - Em canais separados RGB
 - Em tons de cinza (luminância)
- Rotacionar a imagem no eixo vertical e horizontal (flip)
- Plotar o histograma de cada canal RGB e da luminância.

O programa deve dispor de botões para realizar as diversas operações. Pode-se (e sugere-se) que seja implementado um “manager” de botões, panels, checkbox, sliders, scrolls, etc. Utilize seus

conhecimentos de programação. Esse manager (uma classe) poderá ser usado nos demais trabalhos da disciplina. Um slider pode ser usado, por exemplo, para ajustar o brilho de uma imagem.

Para facilitar a implementação da interface de carregamento de imagens, pode-se:

- criar botões que carregam a imagem "a.bmp", "b.bmp" e "c.bmp". Não é necessário carregar mais que 3 imagens, mas o programa deve ser genérico para N imagens.
- As imagens devem estar no diretório images (como está no demo da canvasGLUT). Deve-se enviar as três imagens juntamente com o trabalho, com tamanhos diferentes e resoluções pares e ímpares.

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.

Faça um planejamento (numa folha de papel) de como os elementos serão distribuídos dentro da canvas2D. O programa não precisa ser bonito, mas tem que ser bem organizado e de fácil utilização. Defina se o Y cresce para cima ou para baixo.

Avaliação:

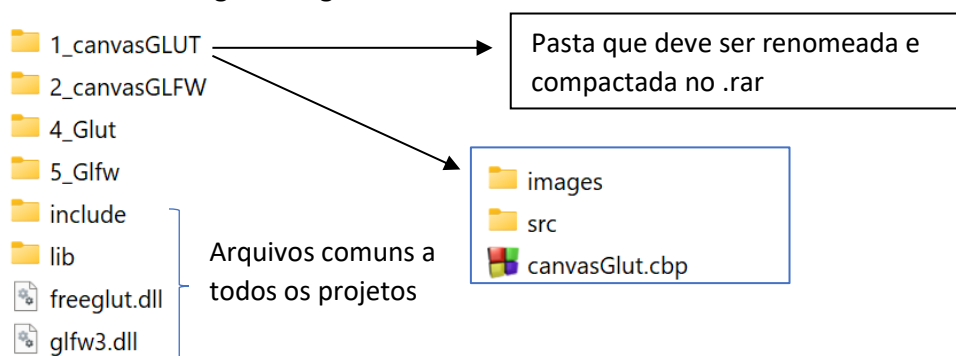
Será avaliada a estruturação do código (nomes de variáveis, funções, métodos, comentários, etc). Trabalhos com códigos replicados perderão **muita** nota (ex: um código para tratar cada botão individualmente). Não utilize números mágicos. Deve-se utilizar laços de repetição sempre que possível. Para quem não tem muito conhecimento em C++, utilizar estruturas e arrays para organizar e generalizar o código.

Bônus (Para nota acima de 9.0):

- Aplicar uma rotação qualquer a imagem. A rotação deve ser feita com o uso do mouse. (até 2 pts)
- Clarear ou escurecer (até 1 pt)
- Display de 7 segmentos para exibição de números (inteiros e reais), letras, hora, etc. (até 2pts)
- Redimensionar a imagem com o uso do mouse (até 2 pt)
- Etc.

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.

- 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.

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).