

Universidade Federal de Santa Maria
Curso de Ciência da Computação
Disciplina: Computação Gráfica
Segundo Semestre de 2020
Prof. Cesar Tadeu Pozzer
Data: 05 / 11 / 2020

Trabalho 1 – Visualizador de Imagens e API de widgets

Ferramentas:

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.

Objetivos:

- desenvolver e habilidade de programação gráfica
- criar e implementar uma API de widgets na Canvas2D
- manipular imagens
- programação em C++

Descrição:

Desenvolva um programa em C++ com interface gráfica para fazer a visualização de imagens em formato BMP. Pode ser utilizado o loader da classe Bmp disponível no demo gl_14_texture (disponível no site da disciplina), que pode ser aprimorado para suportar maior número de configurações de arquivos BMP.

Em relação a interface gráfica, o programa deve disponibilizar widgets dos seguintes tipos:

- Botão
- Sliders
- Checkbox
- Janelas de mensagens
- Etc.

Os widgets devem estar organizados em classes (em um formato de API) de forma que possam ser reutilizados nos próximos trabalhos. Deve-se implementar os “listeners” de cada widget para tratamento dos eventos.

Tendo-se o loader e os widgets, deve-se criar interfaces para fazer a manipulação da imagem carregada, dentre elas:

- Girar 90 graus
- Reduzir pela metade (escala)
- Visualizar cada componente de cor separadamente

- Visualizar a imagem em tons de cinza.

Procure não criar na canvas2D janela com dimensão maior que 1200 x 1000, ou limite a resolução do seu monitor ou notebook.

OBS: O programa deve estar estruturado em classes. Tire todas os códigos não utilizados do demo da canvas2D. Podem ser feitas modificações na canvas2D que julgarem necessárias. Não se pode utilizar recursos do OpenGL que não estejam na Canvas2D. Envie com o trabalho a(as) imagem(ns) utilizadas para testar a aplicação. As imagens devem ter no máximo 400x400 pixels. Copiar trabalhos ou parte de trabalhos de outros alunos implica em nota zero.

Extras (para nota acima de 9,0):

- (+1) Aplicar escalas genéricas.
- (+2) Barra de rolagem.
- (+1) Exibir a imagem em janela que pode ser movida dentro da canvas com o uso do mouse.
- (+1) Plotar histograma de cada canal.
- (+x) Etc.

Entrega:

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, até a data limite.
- 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. Deve-se enviar a Canvas2D.

Antes do envio, teste se o projeto contido no seu .rar funciona em qualquer diretório que ele seja colocado.

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.
- README: incluir um arquivo “README.txt” contendo informações sobre quais funcionalidades foram implementadas (requisitos e extras) e instruções de uso do programa caso o aluno julgue necessário ou caso tenha sido implementado uma funcionalidade extra que exija explicação.
- 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).