

## Trabalho 2 - Transformada Discreta de Cosseno

### OPCIONAL

#### 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.

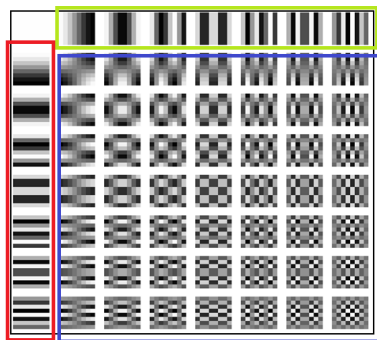
A Canvas2D pode ser customizada, porém não é permitido o uso de funcionalidades OpenGL que não estão presentes na Canvas2D (ex: texturas, shaders). Pode-se criar novas sobrecargas de funções, novos métodos, classes, enums, templates, etc.

#### Descrição:

Desenvolva um programa que aplique a Transformada Discreta de Cosseno (DCT) e sua inversa (IDCT) em uma imagem BMP. O programa deve exibir a

- imagem original,
- a imagem reconstruída e
- a diferença entre as duas imagens. A imagem reconstruída deve ser “idêntica” a original se não aplicado nenhum processo de quantização.

O objetivo do programa é explorar o papel das frequências na reconstrução da imagem. Vamos considerar 3 tipos de frequência: vertical, horizontal e combinação.



Após aplicar a DCT, geram-se os coeficientes de frequência. Desenvolva interfaces gráficas para reduzir ou aplicar frequências específicas da imagem (ou grupos de frequências) e reconstruir novamente, para ver qual efeito isso causa na imagem. A redução de altas frequências causa um borramento. O **aumento** de altas frequências vai causar um realce de bordas?

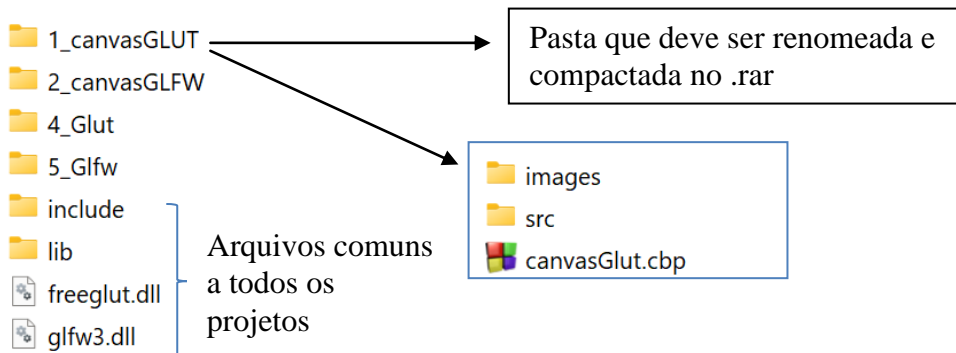
Por exemplo, pode-se aumentar a frequência de todos os coeficientes da primeira linha, ou da segunda linha, ou do elemento [7][7] (o de maior frequência), ou outro qualquer. Preste atenção no coeficiente DC.

**Monte um relatório,** em formato PDF, mostrando os casos de testes e as conclusões que chegaram alterando frequências específicas.

Procure utilizar vários tipos de imagens como entrada. Envie as imagens testadas. Não envie imagens maiores que 400x400 pixels.

### 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).