

Universidade Federal de Santa Maria  
Curso de Ciência da Computação  
Disciplina: Estruturas de Dados A  
Segundo Semestre de 2011  
Prof. Cesar Tadeu Pozzer  
Data: 14/09/2011

## Trabalho *I* – Aba de Botões no MPC

Implemente em C ou em C++, utilizando o **MPC 4.6.2** (Multi-Platform Console), uma Aba de Botões. Nesta aba deve haver botões com funções para formatação de texto. Abaixo desta aba deve haver uma região para inserir texto e imagens. As imagens e os textos só devem ser inseridos na região de edição (parte mais escura da imagem abaixo).



### Botões que devem ser implementados:

- Negrito
- Itálico
- Sublinhado
- Cor de Background (Fundo do texto)
- Cor de Foreground (Cor do texto)
- Edição de Texto (escrever, apagar, home, end)

### Níveis do Trabalho:

- 1- Botões da aba devem ser do tipo on/off para exibir o tipo de fonte selecionado.
- 2- Deve-se poder escrever textos na região editável. (Inserir caracteres na posição do cursor, remover caracteres com backspace e com delete, teclas de navegação home e end).
- 3- O tamanho da Aba de botões deve ser genérico se redimensionando de acordo com a largura do MPC.
- 4- As opções definidas pela aba de botões devem refletir no texto que é escrito.
- 5- Imagem (bônus)
- 6- Quebra de Linha (bônus)
- 7- Mover figuras (bônus)
- 8- Selecionar texto com o mouse (bônus)
- 9- Etc

Extras gerarão nota maiores (pode passar de 10). **Pense bem** em todas as funcionalidades que quer implementar pois algumas são muito difíceis de adaptar em programas mal-estruturados.

Defina uma função para criação da Aba de Botões. Por exemplo, nesta função deve-se passar o tamanho da janela (em caracteres);

Defina uma função para criação da Caixa de Texto. Por exemplo, nesta função deve-se passar o a largura e altura (em caracteres);

### **Dicas:**

A Região Editável pode ser estruturada como um vetor de caracteres ou uma lista de caracteres, e apenas na hora de imprimir deve-se pular as linhas. Este caractere será uma estrutura que conterá o próprio caractere e seus atributos (cor, negrito, itálico, sublinhado).

Procure fazer um código padronizado e bem legível. Caso esse programa for bem planejado poderá servir como parte do **trabalho final** da disciplina de ED. Procure não sair fazendo código diretamente, procure pensar em como exatamente será toda a estrutura do programa e em como será sua organização (considere sempre os seus prazos) para criação de algo **genérico e reutilizável**.

### **MPC:**

Estudando o arquivo “**mpc.h**” pode-se ter uma noção de quais funções o MPC disponibiliza para criação elementos na janela criada pelo próprio MPC. Qualquer dúvida sobre o MPC e/ou suas funções falar com os monitores (Frederico Artur Limberger ou Bernardo Henz).

### **Data e Formato de Entrega:**

O trabalho deve ser entregue até o dia 07/10/2011.

No email e no cabeçalho do arquivo fonte, deve conter o nome completo e matrícula do aluno. O programa deve ser enviado em um arquivo compactado fulano.rar (fulano = login da inf). Dentro deste arquivo deve haver um diretório com o mesmo nome do arquivo e dentro deste diretório os arquivos do trabalho. O arquivo deve ser enviado para **pozzer3@gmail.com** e

**fredericoal@gmail.com** (Frederico Artur Limberger e Bernardo Henz – Monitores) com o assunto “ED – T1”.

Deve-se enviar somente:

- Arquivos fontes (**somente os arquivos necessários**)
- Imagens que forem utilizadas (limitado a 30KB por imagem) e no máximo 10 imagens
- Projeto Codeblocks
- **Não** deve ser enviado a lib do MPC. Será usada para compilação a lib que está no site.
- Não devem ser enviados arquivos .exe, .obj, etc. O trabalho pode ser feito em C ou C++ (sem uso de templates).

### **Critério de Avaliação:**

- **Identificação:** Deixar explicado no cabeçalho do arquivo principal (main) os níveis implementados e os extras (se implementar algo fora da lista).
- **Documentação:** descrever no cabeçalho de cada arquivo a idéia geral do código e detalhes específicos de partes que mereçam uma explicação – não comente por exemplo o que faz i++.
- **Pontualidade:** **Trabalhos não entregues na data não serão avaliados e receberão nota zero.**
- **Legibilidade:** nome de variáveis, estruturação do código. O código digital a ser entregue deve ter *3 ou 4 espaços de indentação*.
- **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).
- **Estruturação:** Como é um programa grande, deve-se elaborar estruturas bem organizadas e modularizadas. A modularização neste trabalho pode ser utilizada em trabalhos futuros relacionados, além de disponibilizar um treinamento para se ter um código mais padrão e compreensivo.

Você pode discutir estratégias e ajudar o colega na implementação, porém evite passar código-fonte. Programas semelhantes terão a nota 0 (zero).