

CIV 2802 – SISTEMAS GRÁFICOS PARA ENGENHARIA – 2011.1

Trab6: Interface gráfica, visualização 2D e tratamento de eventos de *Mouse*

Objetivo

O objetivo deste trabalho é:

- (a) Criar uma interface gráfica (menus, botões etc.) de um programa utilizando o sistema de interface IUP.
- (b) Complementar funções que gerenciam a janela (*window*) de visualização em coordenadas do objeto (*world coordinates*) do programa. Essas funções são utilizadas para dar *zoom*, mover enquadramento etc.
- (c) Complementar as funções que gerenciam a entrada de dados via *mouse* atuando no *canvas* para a criação e manipulação de primitivas gráficas.

Especificação

Deve-se complementar algumas funções do programa que se encontram incompletas. Para fazer as complementações necessárias é preciso que se entenda completamente o programa fornecido. **Este entendimento faz parte do trabalho.**

Para complementar as funções que gerenciam a entrada de dados via *mouse* atuando no *canvas*, é preciso entender a sequência básica dos eventos associados a esta tarefa: (a) botão do *mouse* pressionado registra o primeiro ponto; (b) *mouse* arrastado registra as posições intermediárias do segundo ponto e dá um eco (retorno) visual e temporário da ação a ser executada; e (c) botão do *mouse* liberado registra a posição final do segundo ponto.

A parte do trabalho que compreende a criação da interface gráfica (menus, botões etc.) do programa deve ser implementada com a complementação do arquivo *trab6.led* que especifica a configuração da interface gráfica na linguagem LED. O arquivo fornecido cria um diálogo de interface onde já são especificados uma área de desenho (*canvas*), uma área de mensagens para o usuário e a associação de teclas do teclado a funções do programa. O que se pede é a criação de menus e botões para acionar as mesmas funções que já são acionadas pelo teclado e para acionar algumas funções novas. Obrigatoriamente devem ser criados menus de barra (no topo do diálogo) e botões adjacentes.

Abaixo estão relacionadas as teclas usadas e as correspondentes funções acionadas:

- “h” ou “H” (*Help*): Informações sobre as ações do programa.
 - “Ctrl+Q” (*Quit*): Saida do programa.
 - “Ctrl+I” (*Info*): Informações sobre o trabalho.
 - “Ins” (*Insert*): Insere (cria) uma nova primitiva.
 - “w” (*White*): Define branco como cor corrente.
 - “k” (*black*): Define preto como cor corrente.
 - “m” (*Magenta*): Define magenta como cor corrente.
 - “b” (*Blue*): Define azul como cor corrente.
 - “c” (*Cyan*): Define ciano como cor corrente.
 - “g” (*Green*): Define verde como cor corrente.
 - “y” (*Yellow*): Define amarelo como cor corrente.
 - “r” (*Red*): Define vermelho como cor corrente.
- (Para obter cores escuras tecle Shift+tecla.)
- “Ctrl+P” (*Print*): Imprime o conteúdo da janela de desenho.

- “Ctrl+C” (*Copy*): Cópia para o *clipboard* o conteúdo da janela de desenho.
- “f” (*Fit*): Ajusta a visualização total do modelo na janela.
- “+” (*Zoom +*): Amplia o modelo na janela de desenho.
- “-” (*Zoom -*): Reduz o modelo a janela de desenho.
- “←” (*Pan Left*): Move o enquadramento da janela de desenho para a esquerda.
- “→” (*Pan Right*): Move o enquadramento da janela de desenho para a direita.
- “↓” (*Pan Down*): Move o enquadramento da janela de desenho para baixo.
- “↑” (*Pan Up*): Move o enquadramento da janela de desenho para cima.

Fornecido e Pedido

Os seguintes arquivos são fornecidos:

- *trab6.led*: Arquivo em linguagem LED que especifica a configuração da interface gráfica. Este arquivo deve ser complementado como parte deste trabalho.
- *drv.cpp*: *Driver* que faz a interface do programa para o sistema de interface IUP e que dirige o fluxo de controle do programa. Todas as funções de *callback* registradas no IUP devem ficar neste arquivo. Este arquivo deve ser complementado adicionando-se variáveis correspondentes aos elementos de interface que serão criados na interface gráfica, bem como implementando funções de *callback* que ficarão associadas às ações relacionadas a esses elementos de interface.
- *prj.cpp*: Arquivo onde estão localizadas as funções de projeto do programa, isto é, as funções que distribuem as tarefas para os outros módulos do programa. Essas funções são definidas como membros da classe *Prj*. Parte das funções a serem complementadas se encontram neste arquivo (os locais a serem complementados estão indicados pelo comentário **/***/ COMPLETE AQUI: PRJ-XX */**). O texto de uma variável *char** de informação, definida neste arquivo, deve ser alterado de forma a indicar o autor do trabalho.
- *prj.h*: Arquivo com a definição da classe *Prj*.
- *dsp.cpp*: Arquivo onde estão localizadas as funções que gerenciam a janela de visualização (*window*) para desenhar um modelo no *canvas*. Essas funções estão organizadas na classe *Dsp* (veja comentários **/***/ COMPLETE AQUI: DSP-XX */**).
- *dsp.h*: Arquivo com a definição da classe *Dsp*.
- *model.cpp*: Arquivo onde se encontra a definição de um modelo hipotético com um retângulo e um círculo, que serve como exemplo do programa. O modelo é um objeto da classe *Model*.
- *model.h*: Arquivo com a definição da classe *Model*.
- Bibliotecas de funções do sistema de interface IUP e do sistema gráfico CD, e os correspondentes arquivos de definições e protótipos.
- Arquivos para compilação e criação do executável (Visual C, versão 9).

Pede-se um executável do programa final e os arquivos que foram modificados para implementar o trabalho. Pede-se também, em papel, somente as linhas que foram adicionadas nos arquivos. Estas linhas devem ser indicadas da seguinte forma:

**COMPLETE AQUI: XXX-XX
LINHAS ADICIONADAS**