Trabalho Prático Final – Semestre 2011/1 Turmas A e B

ENUNCIADO DEFINITIVO

MOTIVAÇÃO E OBJETIVOS

O objetivo deste trabalho é exercitar as habilidades e conceitos de programação desenvolvidos ao longo da disciplina pela implementação de um jogo em C, em grupos de 2 alunos.

Deverá ser feito um jogo inspirado no clássico jogo *Soukoban*, em modo texto, onde o usuário do programa (jogador) controlará a movimentação do personagem que move (empurra) caixas para os lugares demarcados através de um labirinto. Apenas uma caixa pode ser empurrada a cada jogada, e o objetivo principal do jogo é fazer com que todas as caixas cheguem aos locais marcados no menor número de movimentos. Uma implementação online deste jogo você encontra em:
http://www.game-remakes.com/play.php?id=279

O PROGRAMA

As duplas terão liberdade para a implementação do programa. Entretanto, os seguintes requisitos mínimos devem ser respeitados:

- o labirinto inicial deverá ser carregado através de um arquivo de texto chamado 'labirinto1.txt' que informe a posição inicial das paredes do labirinto, das caixas, do jogador, dos espaços caminháveis, e também as posições marcadas para destino das caixas;
- o número máximo de caixas fica definido em 5.
- o Jogador será representado pelo caractere 'Q', as caixas pelo caractere '*', as paredes pelo caractere "B", as posições destino das caixas pelo caractere 'X', e espaços em branco (caminháveis) pelo caractere ' ' (espaço em branco);

o labirinto deverá ter um formato retangular, com dimensões máximas de 80 caracteres de largura e 24 caracteres de altura). Ao ser exibido na tela, os caracteres 'B' serão substituídos pelo caractere '█' (código ASCII 219). Um exemplo de labirinto com os dados necessários é mostrado na Figura 1;

- as posições do jogador e das caixas deverão ser manipuladas através de um *struct*, contendo sua posição em x (coluna) e y (linha) na tela. As caixas não são movidas de modo autônomo, apenas pela ação do jogador;
- as seguintes teclas deverão ser utilizadas para controlar a interação no jogo:
  - q – encerra o jogo
  - w – muda a direção de movimentação do jogador para cima
  - x – muda a direção de movimentação do jogador para baixo
  - a – muda a direção de movimentação do jogador para a esquerda
  - d – muda a direção de movimentação do jogador para a direita
  - u – desfaz o último movimento;
- alternativamente podem-se usar as teclas de mudança de direção (setas) para controlar a direção de movimento do jogador (as teclas setas são representadas por dois caracteres, então é necessário fazer duas leituras consecutivas para interpretar uma seta);

- o jogador e as caixas não podem atravessar as paredes (deve haver uma detecção de colisão);
- o jogo deve reconhecer quando uma partida acabou, ou seja, quando todas as caixas estiverem nos lugares pré-determinados;
- o jogo deverá conter um contador de jogadas, a ser exibido em uma linha da tela logo abaixo do final do labirinto. Lembre-se de que o objetivo do jogo é mover as caixas ao local de destino no menor número de jogadas (movimentos de caixa);
- observe que a tecla 'u', que retorna o último movimento, deve ser válida para *n* movimentos. Para cada estado s_i do jogo, pressionar 'u' significa retornar a s_{i-1}, o que quer dizer que estando já no estado s_{i-1}, o jogo deve voltar ao estado s_{i-2} e assim sucessivamente até um limite razoável escolhido pelo programador;
- cada jogo é composto de 3 níveis, que correspondem a 3 labirintos diferentes com dificuldade crescente. A passagem de um nível para outro se dá ao completar com sucesso um determinado nível (os 3 níveis são determinados por 3 arquivos 'labirinto1.txt', 'labirinto2.txt' e 'labirinto3.txt)';
- ao terminar, o jogo mostra as melhores pontuações já obtidas (recordes), que devem ser armazenadas em um arquivo binário cuja estrutura é de livre escolha do programador; não esquecer de perguntar o nome do jogador para associá-lo à sua pontuação.

Dica(s):

- a biblioteca conio2 contém varias funções úteis, como *putchxy, kbhit* e *getch*. A versão para o compilador MinGW que usamos no CodeBlocks pode ser baixada em: http://www.inf.ufrgs.br/~amaciel/software/conio2.zip (vejam as instruções no leiame.txt)
- para o jogo não executar muito rápido, pode-se utilizar a função *sleep*;
- para representar todos os símbolos da tabela ASCII estendida é necessário que a variável correspondente seja declarada como '*unsigned char*';

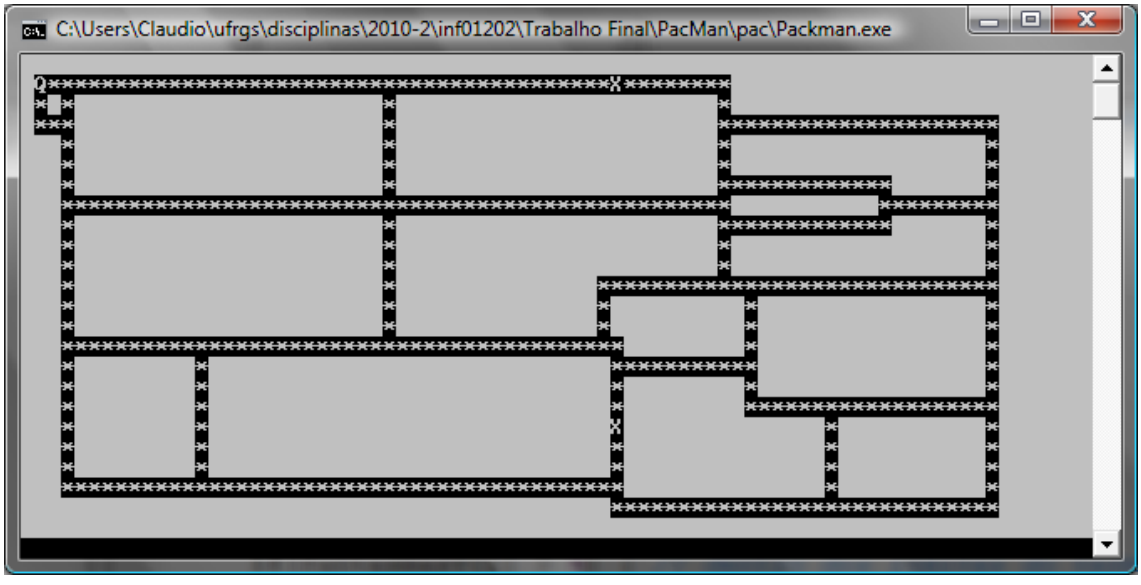

Figura 1: Exemplo de labirinto

REQUISITOS:

- O trabalho deverá ser feito em duplas. Informar os componentes da dupla até o dia **27 de maio**, por e-mail ao professor
- Até o dia **5 de julho**, a dupla deverá submeter via Moodle um arquivo zip cujo nome deve conter o(s) nome(s) do(s) aluno(s). O arquivo zip deve conter:
    - Uma descrição do trabalho realizado contendo a especificação completa das estruturas usadas e uma explicação de como usar o programa
    - Os programas-fonte devidamente organizados e documentados (arquivos .c)
    - Executável do programa (Windows)
- O trabalho será obrigatoriamente apresentado durante a aula prática do dia **8 de julho**. Ambos membros da dupla deverão saber responder perguntas sobre qualquer trecho do código.
- No dia da apresentação serão fornecidos novos arquivos labirinto para testar o programa.
- Os seguintes itens serão considerados na avaliação do trabalho: estruturação do código em módulos (funções, bibliotecas.h, etc.), documentação geral do código (comentários, indentação), "jogabilidade" do jogo e atendimento aos requisitos definidos.
- **Importante:** trabalhos copiados não serão considerados. Saibam que há ferramentas que possibilitam a detecção automática de plágio.