

# Laboratórios de Informática II

## LEI (1º ano)

### Projecto Integrado

#### Guião Laboratorial - 3ª Fase - Jogo do minesweeper

Ano Lectivo de 2006/07

## 1 Introdução

Este documento é parte do guião para apoio ao funcionamento das sessões laboratoriais da disciplina de LI2 durante a 3ª fase de desenvolvimento do projecto integrador, até ao terceiro ponto de controlo e avaliação intercalar, na semana de 21-25 Maio de 2007 sendo apenas apresentados os requisitos em relação ao desenvolvimento do jogo *minesweeper*. Os detalhes referentes aos restantes jogos são apresentados em documentos autónomos.

## 2 Tecnologias a utilizar

- Ambiente Linux
- Editor de texto - vim, Emacs, ...;
- Compilador da linguagem C- gcc;
- Debugger - gdb;

## 3 Objectivos a atingir

Nesta fase do projecto, e em relação a este jogo em particular, pretende-se que os seguintes objectivos sejam alcançados:

1. Descrição das estruturas de dados e do algoritmo do jogo *minesweeper*.
2. Implementação do jogo de forma interactiva, integrado na *Arca de Jogos* que tem vindo a ser desenvolvida.
3. Implementação de uma versão do jogo em linha de comando que permita ler ficheiros que definam o estado inicial do jogo, uma sequência de jogadas e retorne o estado final. Este programa será utilizado para validar o funcionamento do código implementado.

## 4 Requisitos a cumprir

### 4.1 Versão interactiva

Na sua vertente interactiva o jogo do *minesweeper* deverá ser convenientemente integrado com a *Arca de Jogos* que tem vindo a ser desenvolvida. Assim, o jogo fará parte do menú de jogos oferecido, podendo ser jogado uma ou várias vezes até o programa voltar a esta opção. A pontuação deverá ser guardada mantendo-se para cada utilizador um historial contendo no mínimo o número de vezes que jogou cada jogo e a sua melhor pontuação.

Na implementação da versão interactiva do jogo devem ser cumpridos os requisitos listados abaixo.

- O quadro deverá ser gerado, colocando-se aleatoriamente as bombas, gerando quadros sempre distintos de cada vez que o utilizador joga e de cada vez que se arranca a aplicação.
- Ao longo do jogo, um jogador deverá poder realizar duas operações: abrir uma célula e marcar uma célula como sendo uma bomba.
- O jogador perde se abrir uma célula que contenha uma bomba.
- Quando o jogador marca uma célula pela primeira vez, é colocado um símbolo de marcação; não deve ser dada nenhuma indicação sobre se a marcação foi ou não correcta (i.e., se há ou não bomba nessa casa). Uma célula pode ser desmarcada se for chamada a mesma opção sobre uma casa já marcada.
- Uma casa marcada não poderá ser aberta tendo de ser desmarcada antes.
- Para cada célula aberta que não contenha bombas deverá ser indicado o número de vizinhos com bomba (em que vizinho é qualquer célula que se pode atingir movendo-se uma única célula na horizontal, na vertical ou na diagonal).
- Quando se abre uma célula que não tenha vizinhos com bombas, devem ser abertas todas as células vizinhas; este processo deve ser repetido recursivamente.
- O jogador ganha o jogo se abrir todas as células que não contenham bombas (NOTA: o jogo não termina mesmo que o utilizador marque correctamente todas as casas com bomba, havendo a necessidade mesmo nesse caso de abrir as casas que não têm bombas)
- Implementação de vários níveis de jogo (e.g. iniciado, médio, especialista) sendo que pelo menos num destes terá que ser dada hipótese ao utilizador de escolher a dimensão do quadro e o número de bombas a colocar.

Serão ainda contemplados os seguintes requisitos opcionais:

- Interface gráfica mais "agradável".
- Marcação do tempo que um jogador leva a terminar o jogo.
- Implementação de ajudas durante uma partida (e.g. mostrando ou abrindo uma casa sem bombas).

## 4.2 Versão linha de comando

Deverá implementar um programa que corra na linha de comando com o seguinte formato:

```
minesweeper fich_quadro.txt fich_jogadas.txt
```

Este programa receberá o nome de dois ficheiros de texto como argumentos, tendo estes a estrutura e a função que se detalha de seguida:

- Ficheiro de definição do quadro: na primeira linha dois inteiros (separados por espaços) indicando a dimensão da matriz (N) e o número de bombas (NB) a colocar (assume-se que a matriz é quadrada); nas NB linhas seguintes (uma por cada bomba): dois inteiros x e y (separados por espaços) indicando a linha e a coluna de cada casa com bomba (valores de x e y no intervalo [0,N-1]).
- Ficheiro de sequência de jogadas: conjunto de linhas com o formato:  $A\ x\ y$  ou  $M\ x\ y$ . No primeiro caso, significa que se deverá abrir a casa na linha x e coluna y (valores de x e y no intervalo [0,N-1]). No segundo caso, significa que será marcada/ desmarcada a casa na linha x e coluna y. Estas jogadas terão a semântica indicada na secção anterior (note em particular que abrir uma casa marcada não é possível devendo estas linhas ser ignoradas ao executar as jogadas).

O programa a implementar deverá ler os dois ficheiros, criar o quadro respectivo, efectuar as operações respeitantes a cada jogada (até o jogador perder ou ganhar ou até o ficheiro de jogadas não ter mais linhas). No final, o programa deverá escrever um ficheiro de texto com o nome *mine.out* com o estado do tabuleiro com a seguinte forma:

- cada linha do quadro corresponde a uma linha de texto; em cada linha, os valores das colunas são separados por espaços;
- para cada célula imprime-se um caracter:
  - o caracter M se for uma casa marcada como bomba;
  - o caracter \* se for uma bomba explodida (no caso de o jogador ter perdido);
  - um caracter entre 0 e 8 se for uma casa já aberta, indicando com o dígito o número de vizinhos com bomba;
  - o caracter - se a casa ainda não tiver sido aberta nem marcada como bomba.

Para além deste ficheiro, o programa deverá escrever para o stdout a seguinte linha de informação:

- $P\ nj$  - no caso em que o jogador perdeu, indicando a jogada (nj) onde tal aconteceu;
- $G\ nj$  - no caso em que o jogador ganhou, indicando a jogada em que tal aconteceu;
- $I\ nj$  - no caso em que o jogo não terminou, tendo sido efectuadas as nj jogadas incluídas no ficheiro.

Serão colocados no site ficheiros de teste com os resultados respectivos para que possa testar o seu programa. Outros ficheiros distintos podem ser usados pelos docentes para testar o código nas aulas práticas ou na entrega do trabalho no final do semestre.

## 5 Documentação do Trabalho

Estes requisitos deverão ser reflectidos numa versão do relatório, seguindo o template proposto, e que deverá ser entregue até à aula prática da semana respectiva. Esta deverá integrar os algoritmos pedidos, as estruturas de dados usadas bem como a análise do problema, O código C pedido deverá ser enviado como anexo.

Os pontos omissos neste enunciado deverão ser abordados pelos alunos que tomarão as opções que julgarem mais adequadas.

Todas as opções relevantes no desenvolvimento dos algoritmos e do programa deverão constar do respectivo relatório.