Laboratórios de Informática II
Batalha Naval em C

André Freitas Bruno Sousa João Palmeira

28 de Maio de 2015

Conteúdo

1 Introdução

Na batalha naval, existem dois jogadores que tentam descobrir onde estão colocados os barcos um do outro, no entanto, no *puzzle* da batalha naval há apenas um jogador que tem de descobrir onde estão os barcos através de informações como saber o que se encontra em certas posições da grelha (água ou segmentos de barcos) e o número de segmentos em cada linha ou coluna.

Este projeto tem como objetivo criar uma aplicação na linguagem de programação C que resolva o *puzzle* da batalha naval, sendo constituído por três etapas.

2 Desenvolvimento

2.1 1ª Etapa

A aplicação desenvolvida deve ler os comandos do *standard input* que permitem desempenhar várias tarefas. Mais concretamente, para a primeira etapa, os comandos a ser desenvolvidos são "c", "m", "h", "v", "p"e "q"(criando, para isso, um interpretador de comandos). Para esse efeito utilizamos várias funções:

- *main* que chama o interpretador de comandos, dando-lhe um tabuleiro (da batalha naval) vazio;
- *interp* que é o interpretador de comandos e recebe o tabuleiro, sendo que enquanto não surgir o comando "q", vai correr um ciclo. Neste ciclo, quando um tabuleiro é lido, a função verifica o comando inserido e aplica-lhe as ações correspondentes;
- *lerTab* que é chamada quando o interpretador de comandos lê o comando "c", sendo que esta função lê o tabuleiro e devolve-o;
- *insM* que imprime o tabuleiro final e é chamada quando é inserido o comando "m"no interpretador de comandos;
- *insH* que substitui os valores indeterminados da linha do tabuleiro inserida por água. É chamada quando é inserido o comando "h"no interpretador de comandos;
- *insV* que é chamada quando é lido o comando "v"no interpretador e que tem um funcionamento semelhante à *insH*, mas para as colunas do tabuleiro em vez das linhas;
- *insP* que substitui o caracter escrito pelo que está na posição dada quando o comando "p"é lido no interpretador de comandos.

Para além destas funções, é também importante referir o que algumas designações utilizadas significam:

- O tipo *tabuleiro* designa o tabuleiro da batalha naval;
- O *array* de caracteres *tab* representa o tabuleiro da batalha naval sem os segmentos;
- O inteiro *lin* designa a quantidade de linhas do tabuleiro;

- O inteiro *col* representa a quantidade de colunas do tabuleiro;
- O *array* de inteiros *segl* designa os segmentos das linhas do tabuleiro da batalha naval;
- O *array* de inteiros *segc* é idêntico ao *segl*, mas para as colunas em vez das linhas;
- O inteiro *val* designa se o tabuleiro é válido ou não.

2.2 2ª Etapa

Para a segunda etapa, os comandos a ser desenvolvidos são "l", "e", "V", "E1", "E2", "E3"e "D", que vão ser usados através do interpretador de comandos de forma idêntica à da etapa anterior. Para isso, foi necessário ciar várias funções:

- *insL* que é chamada quando é inserido o comando 'l' no interpretador de comandos, sendo que esta função lê o tabuleiro a partir de um ficheiro externo;
- *insE* que escreve o tabuleiro num ficheiro externo e é chamada quando é inserido o comando 'e' no interpretador de comandos;
- *insE1* que coloca água que se deduza que vai existir à volta de todos os segmentos de barcos já colocados, sendo chamada pelo interpretador quando é inserido o comando "E1";
- *insE2* que coloca água nas linhas e colunas em que todos os segmentos de barcos já foram colocados e é chamada pelo interpretador quando é inserido o comando "E2";
- *insE3* que é chamada quando o comando "E3"é inserido no interpretador de comandos e que coloca segmentos de barcos nas linhas e colunas nas quais todos os espaços vazios (isto é, que contêm um '.') têm que conter segmentos de barcos para que o nº correspondente seja respeitado;
- *insVer* que é chamada quando é inserido comando "V"no interpretador, sendo que esta função verifica se o tabuleiro em questão é válido ou não;
- *insD* que desfaz o último comando inserido (ou seja, anula o comando anterior) e é chamada quando o comando "D"é inserido no interpretador de comandos.

Convém ainda referir o significado de mais algumas designações utilizadas neste etapa:

- O tipo *Dcom* contém alterações feitas a um tabuleiro da batalha naval;
- O tipo *Listch* contém as especificações de uma alteração feita a um tabuleiro;
- O tipo *Tabsload* contém uma *stack* de tabuleiros para as funcões em que compensa guardar o tabuleiro todo como, por exemplo, as funções de carregar tabuleiros e as estratégias.

2.3 3ª Etapa

Na terceira etapa,

3 Conclusão

Concluindo, a primeira etapa criou-nos algumas dificuldades como perceber exatamente o que era um interpretador de comandos e como iria funcionar, para além de problemas que surgiram no código, no entanto, julgamos que conseguimos cumprir os objetivos desta etapa. Ajudou-nos a perceber aplicações mais práticas da linguagem de programação C e ajudou-nos também a utilizar uma ferramenta muito útil como o *git* que também nos causou algumas dificuldades inicialmente. No total, tivemos cerca de dezoito horas de trabalho coletivo nas diversas partes necessárias para entrega nesta etapa.

Quanto à segunda etapa, revelou-se um pouco mais difícil, principalmente nos comandos "E3"e "D", mas conseguimos cumprir os seus objetivos. A separação do código em várias partes foi algo bastante útil que aprendemos a nível de organização, tornando mais fácil de perceber o código. Tivemos por volta de 22 horas de trabalho coletivo para ser possível a entrega de todas as partes necessárias nesta etapa.

Em relação à terceira etapa,