BattleShip - Laboratório de Programação

João Varelas

Junho, 2020

Instruções

O código fonte encontra-se no diretório `src/`.

É possível alternar a escolha da estrutura de dados a ser utilizada, quadtree ou matriz, através da definição de uma *macro* como flag do compilador, nomeadamente, `-D_QUADTREE_` e `-D_MATRIX_`.

Por exemplo, através do `Makefile`:

- `make clean client CMACRO=-D_QUADTREE_` ou
- `make clean client CMACRO=-D_MATRIX_`

e depois `make run`.

Para facilitar este processo, estão incluidos scripts `.sh` com os comandos necessários para a compilação.

O binário executável do `client` será gerado no diretório `bin/`.

Configuração do Jogo

É permitida a configuração prévia do jogo no menu `Settings`. Pode ser alterado o tamanho do tabuleiro entre 20x20 e 40x40 e o número de barcos entre 5 e SIZE*SIZE/25, onde SIZE é o tamanho atual do lado do tabuleiro.

As definições do jogo serão posteriormente transmitidas aos jogadores estabeleçam ligação de forma a iniciar uma partida (quem hospeda o jogo é quem define a configuração do tabuleiro).

As configurações são guardadas de forma persistente no diretório `settings/`. O formato do ficheiro é o seguinte:

- 1ª linha: Tamanho SIZE do lado do tabuleiro (SIZExSIZE)
- 2ª linha: Número N de barcos
- N linhas: Configuração do barco 5x5 em bitmap de forma contígua

Um exemplo do ficheiro `settings` é o seguinte:

- 20
- 5
- 00000000000001000000000000
- 00000000000011000000000000
- 00000000000111100000000000
- 00000000001111100000000000
- 00000011100010000010000000

Deve ser evitada a modificação direta do ficheiro. O próprio jogo permite alterar estas definições.

Existe um ficheiro `settings_default` que permite restaurar as definições originais.

Os barcos podem ser "desenhados" no menu `Settings`. Após definir o número N de barcos são pedidas N configurações que consiste no posicionamento de pixeis numa matriz de 5x5.

É necessário que os pixeis colocados sejam adjacentes entre si sem contar com diagonais (poliominó).

Modos de Jogo

A lógica/base do jogo mantém-se para os 3 modos possíveis. A principal diferença está na atribuição dos respetivos *file descriptors* de forma a permitir que cada modo de jogo comunique com o adversário da maneira correta, consoante o caso.

Modo offline

No caso do modo offline, o processo principal é dividido em 2 através da *syscall* `fork()` e posteriormente, estes dois processos são sincronizados com um *named semaphore* para permitir que os jogadores tenham a sua vez de interagir com o programa (no mesmo terminal). A comunicação é establecida por `pipe()` e respetivos *fd's*.

Modo online (mesmo computador)

Neste modo, a comunicação é feita através de *named pipes* com recurso à chamada `mkfifo()`. São criados ficheiros especiais *FIFO* com um certo ID random no filename e posteriormente partilhado entre os 2 jogadores para que consigam establecer ligação.

Modo online (computadores diferentes)

À semelhança dos anteriores, exceto que os *fd's* são inicializados através de `socket()` para establecer uma ligação *TCP/IP*. O anfitrião define a porta que deverá ficar à escuta (o *bind address* é 0.0.0.0 por definição).

Início do Jogo

O jogo dispõe, na maioria das vezes, menus numéricos em que é necessário introduzir um dígito a partir de `1` para selecionar a opção pretendida.

No caso das coordenadas, elas devem ser enviadas na forma `x y` como por exemplo, `4 2`. O `x` representa as linhas do tabuleiro e `y` as colunas.

Para iniciar um jogo deve ser selecionada a opção `1 - New game` e de seguida a forma de jogo:

- `1` - por turnos no mesmo terminal (mas em processos diferentes)
- `2` - em terminais diferentes (na mesma máquina)
- `3` - em terminais diferentes (em máquinas diferentes)

Depois é pedido um nickname do jogador e qual a estratégia pretendida: `1 - Random Strategy` ou `2 - Custom Strategy`.

No caso da `1 - Random Strategy` os barcos são colocados no tabuleiro de forma aleatória e é apresentado o tabuleiro no seu estado final.

Na 2ª opção `2 - Custom Strategy`, o jogador tem a oportunidade de colocar os barcos um a um no tabuleiro, podendo movimentar de 4 formas (`Up`, `Down`, `Left`, `Right`) e ainda rodar a matriz do barco 90º no sentido dos ponteiros do relógio.

Quando o barco estiver na posição desejada, pode-se confimar com `6 - Done`.

Após os dois jogadores finalizarem a colocação de barcos nos seus tabuleiros, o jogo inicia.

É escolhido um jogador aleatóriamente para iniciar o jogo (jogador 1 ou jogador 2) e de imediato são pedidas coordenadas `(x y)` para fazer um disparo.

Cada célula do tabuleiro tem 3 estados possíveis: `UNKNOWN`, `HIT` e `MISSED`.

Quando um jogador acerta num pixel do barco do inimigo, repete a jogada, caso contrário, passa a vez ao adversário.

Um jogador vence quando o seu adversário fica sem barcos disponíveis, i.e., quando os pixeis de todos os barcos são atingidos.

Testes Automáticos

De modo a facilitar os testes *mooshak style*, estão incluidos scripts `run_test_*.sh` para fornecer um input pré-definido com o objetivo de efetuar ações automaticamente tais como selecionar menus, enviar coordenadas, etc.

Por exemplo, o comando `./run_test_quadtree.sh tests/gen_rand_board.inp` executa o `client` e fornece o input de `gen_rand_board.inp`. Neste input em particular, é gerado um tabuleiro aleatório sobre uma estrutura *quadtree*.

Todos os testes (inputs) estão dentro do diretório `tests/`. A descrição de cada teste está disponível em `tests/INFO.txt`.