



**FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA**

Departamento de Engenharia Electrotécnica

PROGRAMAÇÃO DE MICROPROCESSADORES

2012 / 2013

Mestrado Integrado em Engenharia Electrotécnica
e Computadores

1º Ano

1º Semestre

**Trabalho Final
Jogo de Xadrez**

André Lourenço
Angelo Veiga
Carlos Sousa
Francisco Marques
José Barata

1 Introdução

O trabalho final consiste num conjunto de procedimentos de dificuldade crescente que levarão o aluno a tratar a matéria dada na disciplina por forma a construir um **Jogo de Xadrez**. O problema envolve a gestão de todas as peças do tabuleiro e dos seus movimentos.

Os alunos serão encorajados a seguir um processo de implementação em que a cada etapa se incrementará a complexidade da construção e pela qual se regerá a avaliação do trabalho.

2 Estruturas de Dados

O jogo de xadrez estará centrado no tratamento do conjunto das peças do tabuleiro, nomeadamente, como cada peça se movimentará pelo mesmo. Assim, estipula-se neste trabalho final que cada peça será codificada numa estrutura de dados — **PIECE**, que reúne os seguintes campos:

int state: indica o estado da peça: ou em jogo (1) ou eliminada (0);

int type: representa o tipo de peça:

- Peão: 100
- Cavalo: 300
- Bispo: 305
- Torre: 500
- Rainha: 900
- Rei: 2000

char color: reproduz a cor da peça com os caracteres 'w' ou 'b' para, respectivamente, branca ou preta.

SQUARE square: outra estrutura com a localização da peça no tabuleiro, que aglomera os seguintes dados:

char file: coluna com o valor de 'a' até 'h'

char rank: linha entre '1' e '8';

Como exemplo, considere a situação da Figura 1:

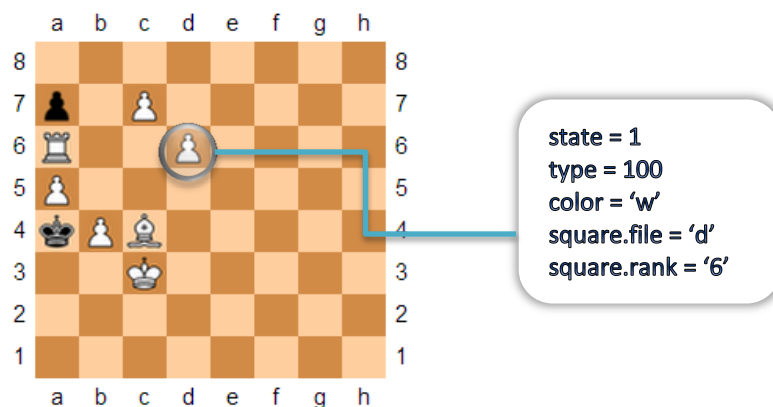


Figura 1 – Exemplo do conteúdo de todos os campos da estrutura **PIECE** relativa a um peão: **state** = 1 : em jogo; **type** = 100 : peão; **color** = 'w' : peça branca; **square.file** = 'd' : linha no tabuleiro; **square.rank** = '6' : coluna no tabuleiro.

3 Programa

Para a implementação do programa propomos ao aluno que siga uma ordem específica, dividida por funcionalidades distintas e de crescente complexidade.

O programa base deve ter as seguintes funcionalidades (**Menu Principal**):

- **Novo Jogo**: em que todas as peças do tabuleiro serão (re)inicializadas (posições e estado);
- **Carregar Jogo do Ficheiro**: em que se lê do ficheiro toda a informação referente às peças (na disposição das estruturas referidas no capítulo anterior) bem como o turno do jogo (branco e preto);
- **Guardar Jogo**: escrever no ficheiro essa mesma informação;
- **Continuar Jogo Corrente**: continuar com o jogo com a disposição anteriormente definida (Reiniciado ou Carregado), acedendo-se ao **Menu de Jogo**;
- **Sair do Jogo**: (explicação implícita...).

O jogador deve aceder a este **Menu Principal**, escolher se **Carrega o Jogo do Ficheiro** ou inicializar o tabuleiro com **Novo Jogo** e, por fim, seleccionar **Continuar Jogo** para aceder ao **Menu de Jogo**. No **Menu de Jogo** deve-se ora introduzir as coordenadas das células do tabuleiro respectivas ao movimento pretendido (por determinada peça) ora voltar ao **Menu Principal**.

3.1 Novo Jogo

A opção de **Novo Jogo** deve implicar que as estruturas das peças (**PIECE**) estejam criadas e inicializadas. Deve-se ter especial atenção à colocação das peças (afecção dos campos **file** e **rank** da estrutura **SQUARE** de cada peça). A Figura 2 mostra a disposição das peças pelo tabuleiro no início de um jogo, em que as peças brancas se apresentam nas linhas (**ranks**) '1' e '2', enquanto que as peças pretas nas linhas '7' e '8', cobrindo as colunas (**files**) de 'a' a 'h'.

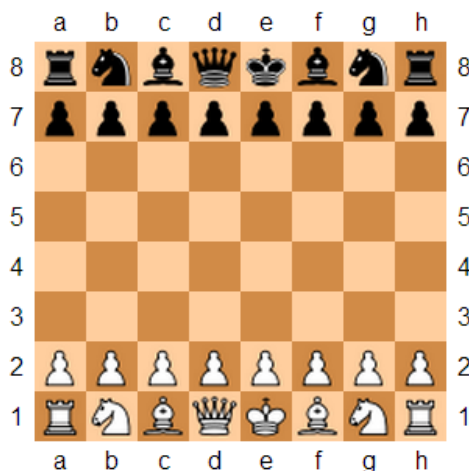


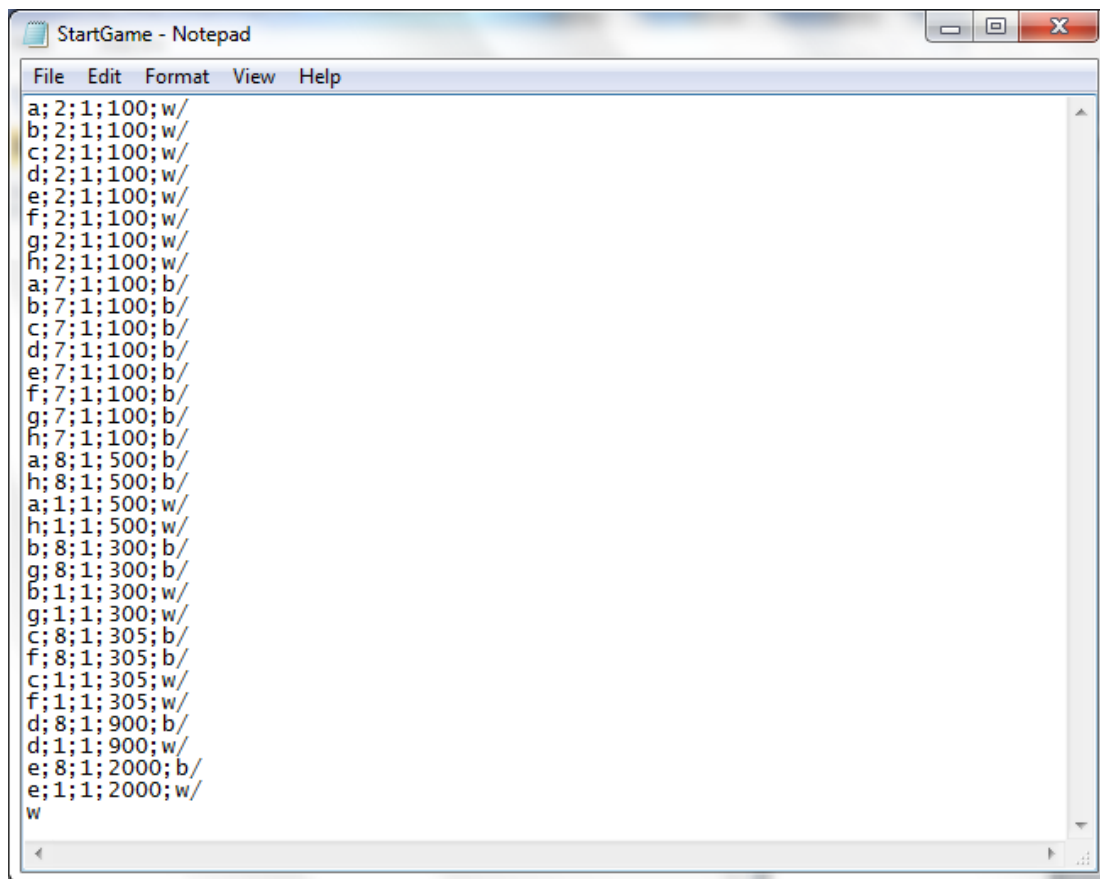
Figura 2 – Configuração do Tabuleiro no início de um jogo.

3.2 Carregar e Guardar Jogo

Com **Carregar Jogo do Ficheiro** entenda-se a leitura do ficheiro de todas as peças do jogo para estruturas **PIECE** (variáveis do jogo), independentemente destas peças estarem correntemente em jogo ou previamente removidas (campo **state**, respectivamente a 1 ou 0), e do turno de jogo (numa variável aparte das restantes). A opção **Guardar Jogo em Ficheiro** deve ter o processo exactamente inverso, i.e., escrever no mesmo formato e ordem.

O ficheiro que se manipulará no programa será necessariamente de texto em que o referido formato e ordem deve ser coerente na leitura e escrita do ficheiro, ainda que a disposição da informação de peças e turno esteja ao critério do aluno.

A docência fornece com este enunciado dois ficheiros que deve tomar como exemplo para o formato e ordem dos processos de leitura e escrita. A Figura 3 apresenta essa mesma configuração no ficheiro **StartGame** (configuração inicial).



```
a;2;1;100;w/
b;2;1;100;w/
c;2;1;100;w/
d;2;1;100;w/
e;2;1;100;w/
f;2;1;100;w/
g;2;1;100;w/
h;2;1;100;w/
a;7;1;100;b/
b;7;1;100;b/
c;7;1;100;b/
d;7;1;100;b/
e;7;1;100;b/
f;7;1;100;b/
g;7;1;100;b/
h;7;1;100;b/
a;8;1;500;b/
h;8;1;500;b/
a;1;1;500;w/
h;1;1;500;w/
b;8;1;300;b/
g;8;1;300;b/
b;1;1;300;w/
g;1;1;300;w/
c;8;1;305;b/
f;8;1;305;b/
c;1;1;305;w/
f;1;1;305;w/
d;8;1;900;b/
d;1;1;900;w/
e;8;1;2000;b/
e;1;1;2000;w/
w
```

Figura 3 - Ficheiro **StartGame** (Notepad)

Em cada linha respectiva a uma peça estão dispostos, separados por ponto e vírgula, os campos da peça consecutivamente: **square.file**, **square.rank**, **state**, **type** e **color**, terminado pela barra '/'.
w

3.3 Continuar Jogo

Nesta opção entra-se num submenu que gere o curso do jogo. Com isto, pretende-se obviamente um ciclo que alterne o turno de jogo entre o jogador das peças brancas e pretas e uma opção de saída, que retorne o utilizador ao **Menu Principal**.

Assim cada turno deve começar por exibir o tabuleiro, a disposição das peças e a vez do jogador. Neste ponto, o aluno tem a tarefa facilitada pois a forma como o tabuleiro e as peças são de facto mostradas na consola por uma função fornecida pela docência.

A dita função apresenta o seguinte cabeçalho:

```
void showBoard(PIECE pieces[32])
```

O processo exige que se passe como argumento um vector de estruturas do tipo **PIECE** com todas as peças (em jogo ou retiradas) sem ordem predefinida. O resultado (*output*) desta função para um vector com as peças no seu estado e posição iniciais é o da Figura 4.

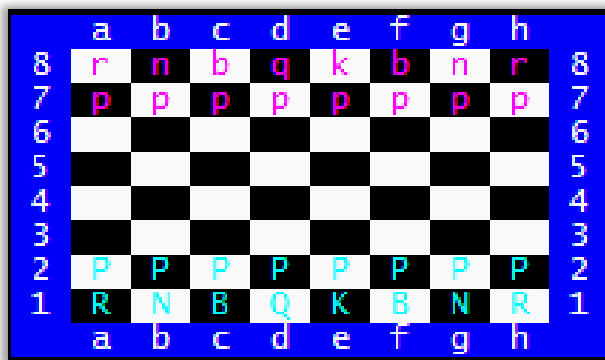


Figura 4 - Output da função **showBoard** na consola.

Após este ponto deve-se pedir as coordenadas da peça que o jogador pretende mover e as respectivas à sua célula de destino.

Com esta informação, entra-se na parte mais importante da construção do jogo, ou seja, todo o algoritmo de verificações e controlo sobre o movimento das peças.

Como deve imaginar o seu programa deve ter em consideração toda a validação correspondente ao movimento da peça dadas as coordenadas de entrada e chegada da mesma. Com isto, pretende-se reforçar a importância de que as coordenadas estão por exemplo, dentro dos limites do tabuleiro, ou que as coordenadas de partida correspondam não só a uma peça do tabuleiro, mas também pertencente ao jogador do turno. Tenha então como referência os seguintes pontos:

- ✓ As coordenadas submetidas devem estar dentro dos limites do tabuleiro;
- ✓ As coordenadas de entrada têm obrigatoriamente de conter uma peça da cor do jogador;

Para lá destas necessárias verificações, outros controlos relativos às regras do jogo de Xadrez como o próprio movimento respectivo aos dois pares de coordenadas serão tomadas em consideração. Isto é, sabendo-se o tipo de peça correspondente às coordenadas de partida deve-se verificar um conjunto de condições:

- ✓ As coordenadas de destino devem corresponder a um movimento válido dentro das regras desse tipo de peça;
- ✓ Consoante a peça das coordenadas de partida, deve-se, ou não, exigir que o caminho (as posições intermédias) esteja livre (atenda-se ao movimento do cavalo e de todas as outras peças);
- ✓ Dependente da peça seleccionada, o movimento pode ser habilitado dependendo da presença de uma peça na posição de saída (lembre-se das regras de movimento da peça Peão).

Como referência para consulta destas regras de movimentos das peças, consulte por exemplo <http://en.wikipedia.org/wiki/Chess>.

4 Ficheiros Fornecidos

Em conjunto com este documento serão fornecidos dois ficheiros de texto: um com a disposição inicial do tabuleiro (**StartGame**); e outro com o jogo num estado mais avançado, com algumas peças capturadas (**SaveGame0**). Estes são ficheiros típicos em que se carregarão e guardarão as peças, para que possa praticar os procedimentos de leitura e escrita de ficheiros.

Será ainda disponibilizado o ficheiro (**showboard.txt**) que contém a referida função **showBoard** que recebe o vector de peças e mostra na consola o tabuleiro.

Por fim, será também anexado um executável com o jogo implementado pela docência, para que tenha uma referência durante a construção do seu próprio programa.

5 Avaliação

Tal como foi referido no capítulo da introdução, estipulou-se um percurso de etapas de crescente complexidade que o aluno deve seguir:

- ✓ **10 Valores:** O programa deve ser capaz de guardar e carregar jogos com o recurso a ficheiros. A peça cavalo deve ser implementada tendo em conta o seu movimento e a captura de outra peça.
- ✓ **12 Valores:** A peça rei deve estar implementada.
- ✓ **14 Valores:** A peça torre deve estar implementada.
- ✓ **15 Valores:** A peça bispo deve estar implementada.
- ✓ **16 Valores:** A peça rainha deve estar implementada.
- ✓ **18 Valores:** A peça peão deve estar implementada. Tenha em atenção que para o peão deve-se distinguir os movimentos de captura dos de simples deslocação.
- ✓ **20 Valores:** Implementação de Cheque.

6 Entrega do Trabalho

O trabalho deve ser entregue num arquivo **.rar** que junta o ficheiro de código C pelo correio electrónico do seu docente das aulas práticas. O nome do ficheiro deve indicar o turno do grupo e os numero dos alunos da seguinte forma:

PX_XXXXX_XXXXX.rar
 / / /
Turno n.º aluno n.º aluno

A data limite para entrega do trabalho é dia 16 de Dezembro de 2012, às 23:59 horas e não serão permitidas entregas posteriores (a nota será zero valores após aquela data).

Planeie metas ao longo deste período para não deixar acumular trabalho na última semana e com isso prejudicar as outras disciplinas (nos seus testes e trabalhos).