

Trench

Relatório Intercalar



Mestrado Integrado em Engenharia Informática e
Computação

Programação em Lógica

Grupo xx:

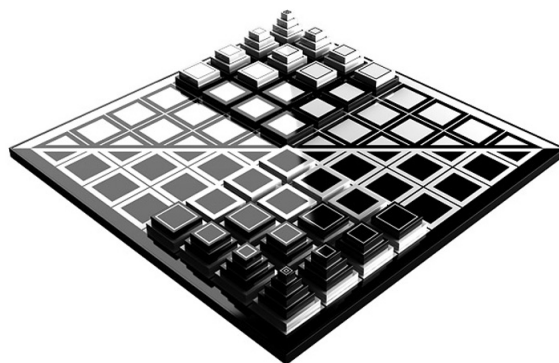
Kevin Amorim - 201207231

Luís Magalhães - 201207224

Faculdade de Engenharia da Universidade do Porto
Rua Roberto Frias, sn, 4200-465 Porto, Portugal

12 de Outubro de 2014

1 O Jogo TRENCH



Trench é um jogo de tabuleiro criado em Portugal por Rui Alípio Monteiro, em 2013. Este é um jogo para 2 jogadores baseado na guerra de trincheiras da 1ª Guerra Mundial.

Os algoritmos aplicados no jogos seguem os princípios referidos no livro "The Art of the War", de Sun Tzu. A disposição das peças do jogo em losango é inspirada na formação em diamante, originária no exército romano.

1.1 Objetivo do Jogo

O objetivo do jogo é capturar todas as peças inimigas. No entanto, nem sempre é possível que tal aconteça (pelas limitações das peças e do tabuleiro), pelo que a vitória ou a derrota regem-se por um sistema de pontuação, explicado em baixo.

1.2 Início do Jogo

O jogador que possuir as peças de cor preta inicia o jogo. Cada jogador tem direito a uma jogada por turn. O jogo termina ao fim de 25 jogadas se nenhuma peça for capturada nesse intervalo.

1.3 Tabuleiro de Jogo

O tabuleiro representa o campo de batalha, com as respectivas trincheiras. O tabuleiro de jogo tem uma forma em diamante, dividido por uma linha diagonal, que representa a linha das trincheiras (Cada metade do tabuleiro tem uma cor predominante: preto para um jogador, branco para o outro). O tabuleiro é constituído por 64 casas (8x8), divididas em dois territórios opostos.

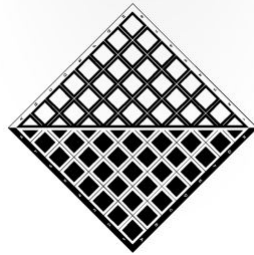


Figura 1: Tabuleiro de jogo 8x8

1.4 Peças do Jogo

As peças do jogo são as apresentadas na seguinte tabela:

Postos	Pretas		Brancas		Nº de Peças
General					<i>x 1</i>
Coronel					<i>x 2</i>
Capitão					<i>x 3</i>
Sargento					<i>x 4</i>
Soldado					<i>x 6</i>

Peças

Figura 2: Tabel das peças do jogo

Estas possuem uma forma em losango, inspirada nas estrelas usadas pelos soldados em batalha. Estas simbolizam a hierarquia militar em pirâmide.



Figura 3: Peça do jogo Trench

As peças seguem a seguinte hierarquia (estando no topo o de mais alto nível):

1. General
2. Coronel
3. Capitão
4. Sargento
5. Soldado

1.4.1 Disposição das Peças

As peças são dispostas no tabuleiro de forma a simular a formação de um exército Romano, em diamante, como se poder ver na Fig. 5. O general, a peça com maior ranking na hierarquia, fica no topo da metade aliada do tabuleiro, atrás de todo o exército. Na linha da frente ficam os 6 soldados, menor ranking na hierarquia.

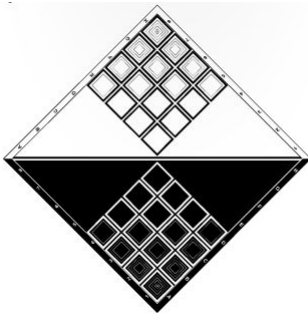


Figura 4: Disposição das peças do jogo

1.4.2 Movimento das Peças

- **Soldado:** 1 casa (na diagonal em qualquer direção);
- **Sargento:** 2 casas (na diagonal em qualquer direção e para a frente);
- **Capitão:** 3 casas (na diagonal em qualquer direção e tanto para a frente como para trás);
- **Coronel:** 4 casas (na diagonal em qualquer direção, para frente, esquerda e direita - mas não para trás);
- **General:** 5 casas (em todas as direções);








		Soldado
		Sargento
		Capitão
		Coronel
		General

Figura 5: Representação do movimento das peças do jogo

1.4.3 Condições do Movimento das Peças

- O jogador é sempre obrigado a movimentar uma peça, na sua vez.
- Nenhuma das peças de jogo pode avançar sobre outra peça (seja do seu ou do exército adversário);
- Para capturar uma peça inimiga, a peça do jogador passa a ocupar a parcela quadrada da peça do adversário (terminando o seu movimento imediatamente).
- Nenhuma peça é obrigada a percorrer a totalidade das casas que pode percorrer. Isto é, por exemplo, o General pode se mover apenas 1 casa ou 5 casas, conforme o jogador quiser.

1.5 A Trincheira

A trincheira é representada pela linha horizontal no centro do tabuleiro, como referido anteriormente. As peças nesta linha usufruem de um conjunto de vantagens e desvantagens:

- **1ª Vantagem:** Uma peça na trincheira não pode ser atacada por uma peça adversária.
- **2ª Vantagem:** Uma peça na trincheira não é obrigada a parar quando ataca uma peça adversária. Portanto, essas podem continuar o seu movimento ou até atacar outras peças, até concluir a sua totalidade de casas a movimentar, ou o jogador decidir parar.
- **1ª Restrição:** Uma peça em trincheira não pode capturar peças adversárias que se encontrem no território aliado (retaguarda). Mas podem ser atacadas por peças adversárias que se encontrem no território aliado.
- **2ª Restrição:** O Coronel e o General podem se movimentar ao longo de toda a linha da trincheira, mas não podem capturar nenhuma peça adversária que aí se encontre, ficando com os seus movimentos limitados.

1.6 Fim do Jogo

O jogo termina ao fim de duas partidas. Depois de cada partida os jogadores trocam de lado, para que cada um jogue uma vez com as brancas e outra com as pretas. Uma partida termina quando algum jogador capturar todas as peças adversárias. No entanto, se após 50 jogadas ninguém capturar todas as peças adversárias o jogo termina e procede-se a contagem de pontos, segundo a seguinte tabela:










  Soldado	 1 Estrela = 2 Pontos
  Sargento	 2 Estrelas = 4 Pontos
  Capitão	 3 Estrelas = 6 Pontos
  Coronel	 4 Estrelas = 8 Pontos
  General	 5 Estrelas = 10 Pontos

Figura 6: Valor de cada peça do jogo

No final das duas partidas somam-se os pontos feitos por cada jogador em cada partida e ganha o jogo o jogador com mais pontos. Em caso de empate joga-se outra partida e ganha o jogo o jogador que conseguir capturar 40 pontos primeiro.

2 Representação do Estado do Jogo

A representação do estado do jogo irá ser feita através de uma lista de listas. A matriz (lista de listas) irá ser composta por uma lista contendo 15 listas de diferentes tamanhos. As listas mais internas serão de diferentes tamanhos para se poder representar a forma em diamante do tabuleiro. Portanto, a primeira e a última lista seriam compostas por apenas 1 elemento, enquanto que a oitava lista, a do meio, correspondente à trincheira, seria composta por 8 elementos.

Cada peça do jogo será representado pelo seguinte átomo:

- **Soldado:** So
- **Sargento:** Sa
- **Capitão:** Ca
- **Coronel:** Co
- **General:** G
- **Espaço Vazio:** E

A cada peças destas, excepto as vazias, será acrescentado um sufixo com o número do jogador. Por exemplo, para o jogador 1, um soldado será representado por: So1.

Para distinguir a metade do tabuleiro branca da preta, assumimos sempre que as listas de 0 a 6 pertencem ao branco e as listas de 8 a 14 pertencem ao preto. A lista 7 (oitava lista) é a linha da trincheira, como já referido anteriormente.

Em Prolog a representação será a seguinte:

Estado inicial

```
gameList( [ [g1], [co1, co1], [ca1, ca1, ca1], [sa1, sa1, sa1, sa1], [e, so1, so1, so1, e], [e, e, so1, so1, e, e], [e, e, e, so1, e, e, e], [e, e, e, so2, e, e, e], [e, e, so2, so2, e, e], [e, so2, so2, so2, e], [sa2, sa2, sa2, sa2], [ca2, ca2, ca2], [co2, co2], [g2] ]).
```

Estado final com todas as peças conquistadas (exemplo)

```
gameList( [ [e], [co1, co1], [ca1, ca1, e], [sa1, e, sa1, sa1], [e, so1, e, so1, e], [e, e, so1, so1, e, e], [e, ca1, e, e, e, e, e], [e, e, e, e, g1, e, e], [e, e, sa1, e, e, e], [e, e, e, e, e], [e, e, e, so1], [e, e, e], [e, e], [e] ]).
```

Estado intermédio (exemplo)

```
gameList( [ [e], [co1, co1], [ca1, ca1, e], [sa1, e, sa1, sa1], [e, so1, e, so1, e], [e, e, so1, so1, e, e], [e, ca1, e, e, e, e, e], [e, so2, e, e, g1, e, e], [e, e, sa1, e, e, e], [ca2, e, e, e, e], [e, e, e, so1], [e, g2, e], [e, e], [e] ]).
```


3 Visualização do Tabuleiro

A representação do jogo na consola, em Prolog, está implementada da seguinte forma: (Versão ainda temporária)

```
printGameState([], _).

// Parameters: List to print, Starting line
printGameState([H|T], I) :-
    S is abs(7 - I),
    printLineSpaces(S);
    printGameLine(H),
    nl,
    I1 is I + 1,
    printGameState(T, I1).

// Prints empty lines
// Parameters: S: number of line to print.
// Restrictions: S should be bigger than 0 (zero).
printLineSpaces(S) :-
    S > 0,
    write(' '),
    S1 is S - 1,
    printLineSpaces(S1).

// Prints the actual game line
printGameLine([]).

// Parameters: Game line to print (list)
printGameLine([H|T]) :-
    getSymbol(H, S),
    write(S),
    write(' '),
    printGameLine(T).
```

A função de `getSymbol()` retorna o símbolo correspondente a cada átomo.

```
gameList(X), printGameState(X, 0).
G1
Co1 Co1
Ca1 Ca1 Ca1
Sa1 Sa1 Sa1 Sa1
_ So1 So1 So1 _
_ _ So1 So1 _ _
_ _ _ So1 _ _ _
_ _ _ _ So2 _ _ _
_ _ _ _ So2 So2 _ _
_ _ _ _ So2 So2 So2 _
_ Sa2 Sa2 Sa2 Sa2
Ca2 Ca2 Ca2
Co2 Co2
G2
```

Figura 7: Output para a consola em Prolog

4 Movimentos

Versão temporária e ainda não implementada dos predicados que poderão ser aplicados na mecânica do jogo:

```
// =====  
// Predicates , not implemented yet  
// =====  
  
// Checks if players piece can move to a given direction  
canMove(s1, northeast).  
canMove(s1, northwest).  
canMove(s1, southeast).  
canMove(s1, southwest).  
  
// etc for all the other pieces...  
//  
// Updates the list according a Y movement of a X piece.  
// update(X, Y).  
  
// Moves the piece X to the direction Y... Checks if the movement is possible  
// if it's possible, updates the list  
// move(X, Y) :- canMove(X, Y), update(X, Y).  
  
// Other predictates:  
  
// canAttack(X, Y). -> Checks if the piece X can attack the piece Y  
  
// attack(X, Y) :- canAttack(X, Y), updateAttk(X, Y).  
  
// win(X) - checks if player X has won the game  
  
// =====
```