

Faculdade de Engenharia da Universidade Do Porto

Mestrado Integrado em Engenharia Informática e Computação

Programação em Lógica

Relatório Intercalar

Alunos:

João Miguel Vaz Tello da Gama Amaral

João Nuno Rodrigues Ferreira

16 de Outubro, 2018



Índice:

● O Jogo: Knight Line.....	3
● Início do jogo.....	4
● Objetivo.....	4
● Desenvolvimento.....	4
● Final.....	5
● Representação do Estado do Jogo.....	5
● Visualização do Tabuleiro.....	7
● Referências.....	9

O Jogo: Knight Line

Knight Line é um jogo abstracto de estratégia e de conexão minimalista para dois jogadores. Cada jogo tem uma duração média de 10 minutos. A origem do jogo é desconhecida.

No início do jogo, cada jogador tem 20 azulejos quadrados de cor preta ou branca. Estas peças são colocadas em duas pilhas adjacentes.

Cada jogada consiste em mover uma parte da pilha (deve ser deixado para trás, no mínimo, um tijolo) para um espaço vazio fazendo um “knight’s move” (movimento baseado na peça cavalo do xadrez). Todas as peças têm de estar conectadas pelo menos diagonalmente durante o decorrer do jogo. O primeiro jogador a fazer uma linha de 4 ortogonalmente ou na diagonal ganha.

Ao contrário de jogos como xadrez, Knight Line não é jogado num tabuleiro, mas sim numa grelha imaginária e cujos limites não estão estabelecidos. O tamanho da grelha na qual devemos pensar é, portanto, o espaço ocupado pelas peças, juntamente com as casas adjacentes, nas quais ainda não há peças, mas que são possíveis jogadas.

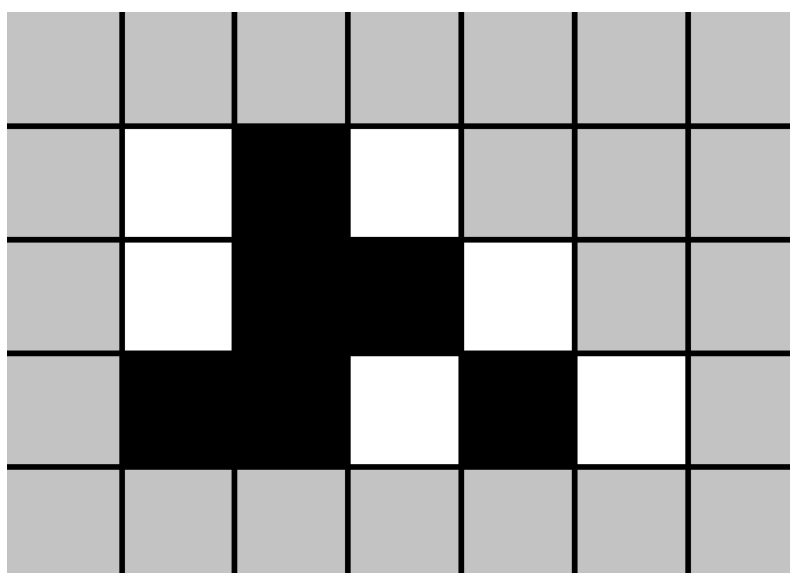


Fig. 1 – Exemplo de uma situação intermédia do jogo.

Início do jogo: Para o jogo começar devem ser colocadas as duas pilhas de 20 azulejos em duas posições adjacentes, horizontalmente. Tendo em conta que o jogador que possui as peças pretas será o primeiro a jogar, o que lhe dá alguma vantagem sobre o adversário, apenas poderá mover 1 peça para a casa na qual a pretende colocar.

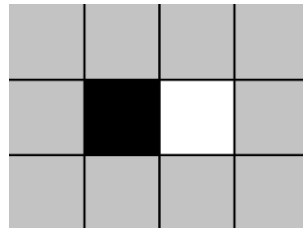


Fig. 2 – Situação inicial do jogo.

Objetivo: O objetivo é conseguir fazer uma linha de 4, podendo cada elemento da linha ser uma única peça ou uma pilha com mais de uma.

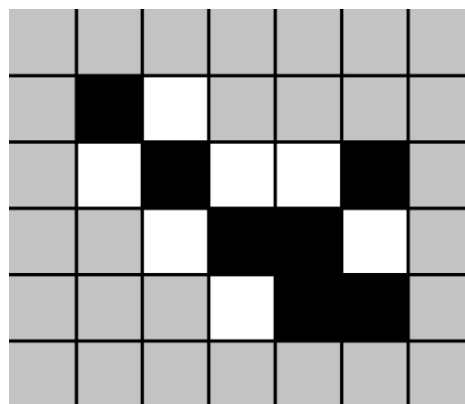


Fig. 3 – Situação final do jogo (4 pretas em linha na diagonal).

Desenvolvimento: Em cada jogada, cada jogador pode movimentar uma porção da pilha inicial, deixando na posição inicial, pelo menos, uma peça. As jogadas têm de respeitar o movimento “knight’s move”, porém a jogada só pode ser feita para uma posição adjacente a outra peça, quer seja uma peça do jogador ou do adversário.

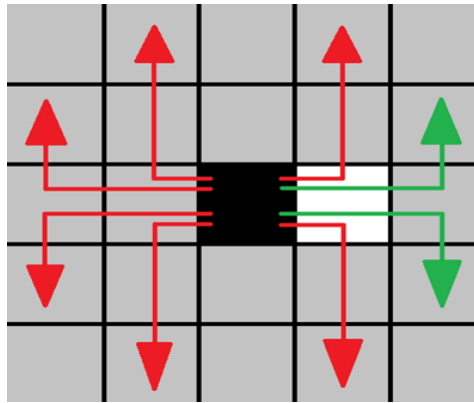


Fig. 4 – Possíveis jogadas iniciais da peça preta a verde e não possíveis a vermelho.

Final: O fim do jogo acontece quando um jogador conseguir fazer uma linha de 4 com as suas respetivas peças, respeitando sempre o “knight’s move” e nunca deixando espaços vazios em sítios onde já jogou.

Representação do Estado do Jogo

```
initialBoard([
[[empty, 0], [empty, 0], [empty, 0], [empty, 0]],
[[empty, 0], [black, 20], [white, 20], [empty, 0]],
[[empty, 0], [empty, 0], [empty, 0], [empty, 0]]
]).
```

////	0	1	2	3
A	e,0	e,0	e,0	e,0
B	e,0	b,20	w,20	e,0
C	e,0	e,0	e,0	e,0

Fig. 5 – Situação inicial do jogo vista na consola do SICStus.

midBoard(
[[empty,0],[empty,0],[empty,0],[empty,0],[empty,0],[empty,0],[empty,0]],
[[empty,0],[white,4],[black,1],[white,4],[empty,0],[empty,0],[empty,0]],
[[empty,0],[white,2],[black,6],[black,3],[white,4],[empty,0],[empty,0]],
[[empty,0],[black,2],[black,5],[white,5],[black,3],[white,1],[empty,0]],
[[empty,0],[empty,0],[empty,0],[empty,0],[empty,0],[empty,0],[empty,0]]
]).

////	0	1	2	3	4	5	6
A	e,0	e,0	e,0	e,0	e,0	e,0	e,0
B	e,0	w,4	b,1	w,4	e,0	e,0	e,0
C	e,0	w,2	b,6	b,3	w,4	e,0	e,0
D	e,0	b,2	b,5	w,5	b,3	w,1	e,0
E	e,0	e,0	e,0	e,0	e,0	e,0	e,0

Fig. 6 – Exemplo de uma situação intermédia do jogo vista na consola do SICStus.

finalBoard(
[[empty,0],[empty,0],[empty,0],[empty,0],[empty,0],[empty,0],[empty,0]],
[[empty,0],[black,1],[white,3],[empty,0],[empty,0],[empty,0],[empty,0]],
[[empty,0],[white,2],[black,4],[white,6],[white,2],[black,5],[empty,0]],
[[empty,0],[empty,0],[white,1],[black,2],[black,3],[white,1],[empty,0]],
[[empty,0],[empty,0],[empty,0],[white,5],[black,2],[black,3],[empty,0]],
[[empty,0],[empty,0],[empty,0],[empty,0],[empty,0],[empty,0],[empty,0]]
]).

//////	0	1	2	3	4	5	6
A	e,0	e,0	e,0	e,0	e,0	e,0	e,0
B	e,0	b,1	w,3	e,0	e,0	e,0	e,0
C	e,0	w,2	b,4	w,6	w,2	b,5	e,0
D	e,0	e,0	w,1	b,2	b,3	w,1	e,0
E	e,0	e,0	e,0	w,5	b,2	b,3	e,0
F	e,0	e,0	e,0	e,0	e,0	e,0	e,0

Fig. 7 – Situação final do jogo (4 brancas em linha na diagonal).

Visualização do Tabuleiro: Demonstração do código que será utilizado para mostrar o tabuleiro na consola.

Símbolos:

symbol(empty, S) :- S='e'.

symbol(white, S) :- S='w'.

symbol(black, S) :- S='b'.

letter(1, S) :- S='A'.

...

letter(26, S) :- S='Z'.

Tabuleiro:

display_game([H|T], Player) :-

arrayLength(H, Columns),

nl, write('|'), separation(Columns), write('|'), nl,

write('////////| '), tableTop(Columns, 0), nl,

write('|'), separation(Columns), write('|'),

printBoard([H|T], Columns, 0, 1), nl, nl.

printBoard([], _, _, _).

printBoard([H|T], C, R, N) :-

 nl,

 N1 is N + 1,

 R1 is R + 1,

 letter(R1, S),

 write('| '), write(S), write(' | '), printLine(H), nl,

 write('|'), separation(C), write('|'),

 printBoard(T, C, R1, N1).

Linha:

printLine([]).

printLine([H|T]) :-

 printPair(H),

 printLine(T).

Par de átomos:

printPair([]).

printPair([H|T]) :-

 symbol(H, S),

 write(S),

 printNumber(T).

Referências:

- https://nestorgames.com/#knightline_detail
- <https://boardgamegeek.com/boardgame/146989/knight-line>
- <https://boardgamegeek.com/thread/1594753/clever-little-game-which-deserves-more-attention>