

Straight 4

Relatório Intercalar
Programação em Lógica
20 de Outubro de 2019

Índice

Descrição.....	2
Regras.....	2
Representação interna do estado de jogo.....	4

Descrição

Straight 4 é um jogo de estratégia abstrata criado em 2019 por Binary Cocoa, feito para ser jogado por dois jogadores. O seu tempo de jogo é entre 5-10 minutos.

O material necessário para o jogo é um tabuleiro quadrado com 5x5 espaços e oito peças (4 de cor branca e 4 de cor preta).

Regras

Inicialmente o tabuleiro está vazio. Cada jogador tem uma cor associada (branca ou preta) e quatro pedras dessa mesma cor. Depois de decidido quem joga primeiro, cada jogador vai colocar, alternadamente, uma peça no tabuleiro.

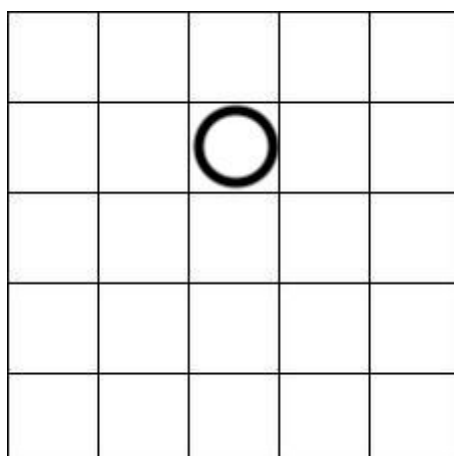


Figura 1: Tabuleiro do jogo com a peça do jogador inicial colocada (neste caso, o branco).

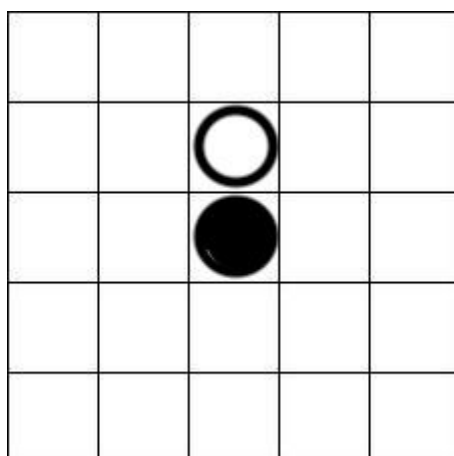


Figura 2: Tabuleiro do jogo após duas jogadas.

O objetivo do jogo é um jogador conseguir colocar as suas 4 peças numa linha reta (horizontal, vertical ou diagonal).

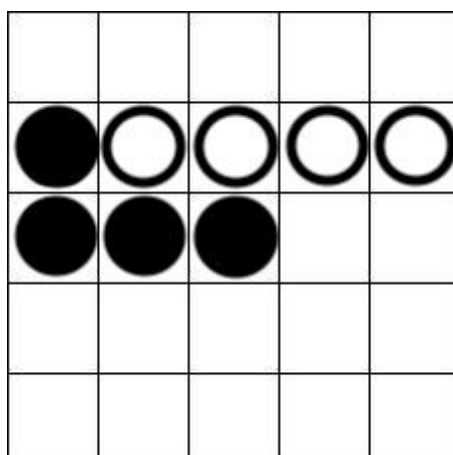


Figura 3: Exemplo de fim de jogo em que o jogador com as peças brancas ganha.

Se depois de colocadas todas as peças ninguém conseguir vencer, vai-se alterando a posição de uma peça de cada vez tentando colocar quatro peças em linha.

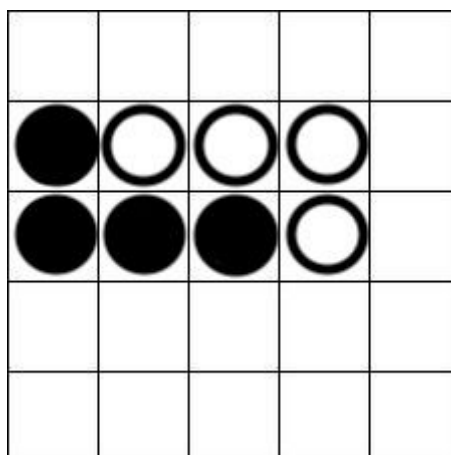


Figura 4: Exemplo com todas as peças colocadas e nenhum jogador ganha.

Vence o primeiro jogador a conseguir colocar 4 peças em linha

Referências: <https://www.boardgamegeek.com/boardgame/284636/straight-4>
<http://www.jbobsmoviereviews.com/wordpress/straight-4-review/>
<https://binarycocoa.com/portfolio/straight-4/>
<https://binarycocoa.com/2019/02/23/straight-4-now-available/>

Representação interna do estado de jogo

Estado Inicial

```
table([[0,0,0,0,0],
      [0,0,0,0,0],
      [0,0,0,0,0],
      [0,0,0,0,0],
      [0,0,0,0,0]
      ]).
```

	A	B	C	D	E	
1	-	-	-	-	-	1
2	-	-	-	-	-	2
3	-	-	-	-	-	3
4	-	-	-	-	-	4
5	-	-	-	-	-	5
	A	B	C	D	E	

Figura 5: Situação Inicial

Estado Intermédio

```
table([[0,0,0,0,0],
      [0,1,1,2,0],
      [0,2,2,1,0],
      [0,0,0,0,0],
      [0,0,0,0,0]
      ]).
```

	A	B	C	D	E	
1	-	-	-	-	-	1
2	-	●	●	○	-	2
3	-	○	●	-	-	3
4	-	-	-	-	-	4
5	-	-	-	-	-	5
	A	B	C	D	E	

Figura 6: Situação Intermédia

Estado Final

```
table([[0,0,1,0,0],
      [0,1,1,2,0],
      [0,2,1,2,0],
      [0,0,1,0,0],
      [0,0,2,0,0]
      ]).
```

	A	B	C	D	E	
1	-	-	-	-	-	1
2	-	○	○	○	-	2
3	●	●	●	●	○	3
4	-	-	-	-	-	4
5	-	-	-	-	-	5
	A	B	C	D	E	

Figura 7: Situação Final

Visualização do tabuleiro

Segue-se o código utilizado para a impressão do tabuleiro na consola:

```
print_table_header :-  
    write(' '),  
    write('A'), write(' '), write('|'), write(' '), write('B'), write(' '), write('|'), write(' '),  
    write('C'), write(' '), write('|'), write(' '), write('D'), write(' '), write('|'), write(' '), write('E'), nl.
```

```
print_cell(0):- put_code(173), !.  
print_cell(1):- put_code(9679), !.  
print_cell(2):- put_code(9675).
```

```
print_column([]):- !.  
print_column([H|T]) :-  
    print_cell(H),  
    write(' | '),  
    print_column(T).
```

```
print_board([], 6):- !.  
print_board([H|T], N) :-  
    print_format_number(N),  
    write('| '),  
    print_column(H),  
    print_format_number(N),  
    nl,  
    Next is (N + 1),  
    print_board(T, Next).
```

```
print_format_number(N) :-  
    !,  
    write(N),  
    write(' ').
```

```
initBoard(Board) :-  
    table(Board),  
    print_table_header,  
    print_board(Board, 1),  
    print_table_header.
```

```
show_player(Player) :-  
    Player = '1', nl, write('PLAYER: black'), nl, nl;  
    Player = '2', nl, write('PLAYER: white'), nl, nl.
```

```
display_game(Board, Player) :-
```

```
show_player(Player),  
initBoard(Board).
```

As três últimas figuras representam os outputs na consola produzidos pelo código acima representado.