

# FUSE

## **Relatório Intercalar** Programação em Lógica

Eduardo João Santana Macedo  
Francisco José Paiva Gonçalves

*up201703658@fe.up.pt*  
*up201704790@fe.up.pt*

## Índice:

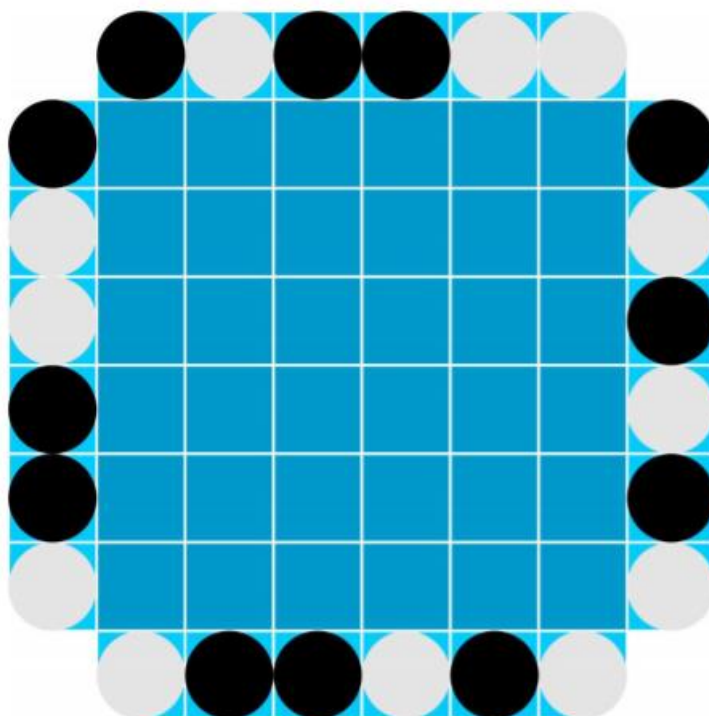
Sobre o Jogo	3
Preparação	3
Como jogar	4
Fim do Jogo	5
Representação	6
Output	8

## Sobre o Jogo:

FUSE é um jogo de tabuleiro espanhol criado apenas em 2019 que iremos desenvolver usando a linguagem de. Trata-se de um jogo em que cada jogador alterna entre jogadas para no final obter a melhor pontuação.

## Preparação do Jogo:

O jogo começa com as peças brancas e pretas distribuídas nas casas do perímetro aleatoriamente, porém nunca poderão estar mais que duas casas da mesma cor seguidas. Na imagem em baixo está representada uma possível configuração inicial do tabuleiro.



**Figura 1:** Exemplo do tabuleiro do jogo inicial

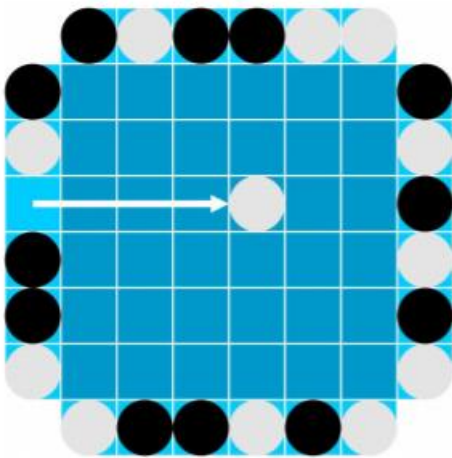
O material necessário para o jogo é um tabuleiro quadrado 7x7 (sendo que as casas dos cantos nunca serão usadas nem terão representatividade no jogo), 12 peças brancas, 12 peças pretas e, para um modo de jogo de jogo extra, duas peças azuis.

## Como jogar:

Começando com as peças brancas, os jogadores vão alternando jogadas, arrastando os discos do perímetro para a área do tabuleiro 6x6 (azul escura na imagem). O jogador é obrigado a usar um disco a menos que não tenha alternativa sendo que, nesse caso, passa a vez. Os discos apenas se podem mover na coluna ou linha em que começaram e podem se mover tantas casas sendo que:

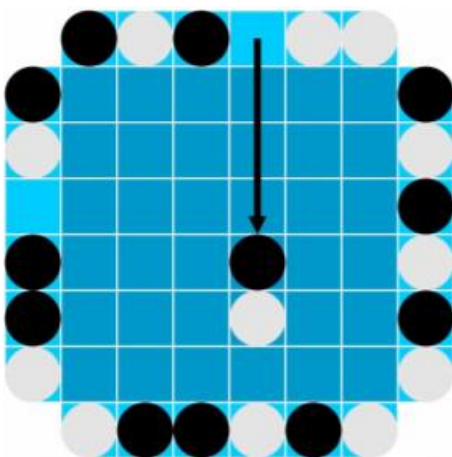
- Um ou mais discos que estejam no caminho podem ser empurrados como resultado da jogada

- Todos os discos no fim da jogada têm de estar dentro da área jogável (6x6), ou seja, é ilegal fazer uma jogada em que sejam empurrados discos para fora da área azul escura.



**Figura 2:**

- Exemplo de uma possível jogada por parte do jogador com peças brancas



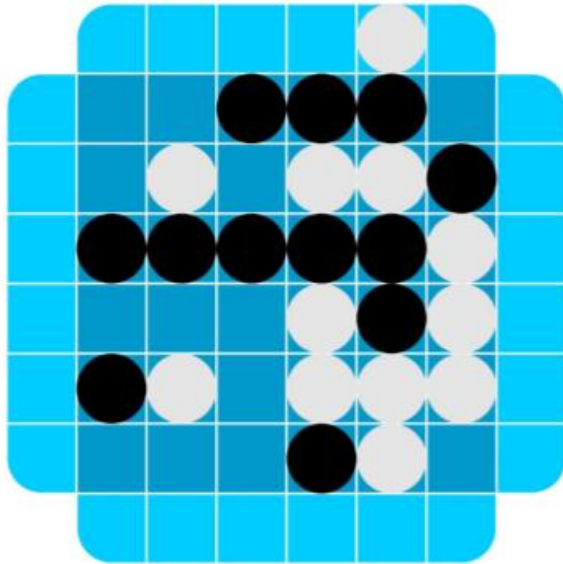
**Figura 3:**

- Exemplo de uma possível jogada de resposta do jogador das peças brancas.

- Note-se que a peça branca foi empurrada.

## Fim do Jogo:

- O jogo termina quando mais nenhum jogador já não tiver mais jogadas possíveis a fazer.
- O jogador que tiver mais peças conectadas ortogonalmente da da sua cor ganha sendo que empates são possíveis.
- Devido à rapidez do jogo, é recomendado este ser jogado à melhor de 3 ou de 5.



**Figura 4:**

- Exemplo de vitória do jogador de peças brancas (7 – 6)
- Note que o jogo terminou mesmo com a peça branca no perímetro pois já não são possíveis mais nenhuma jogada (jogar a peça branca implicaria empurrar a outra peça branca da extremidade para fora da área jogável)

## Representação do Jogo:

Para armazenar a informação do jogo foi utilizada uma lista de listas com diferentes átomos (white, black, null e empty) para representarem as peças no tabuleiro ou ausência destas mesmas.

Em baixo encontra-se a definição usada em código para dois exemplos de tabuleiros.

```
tabuleiroInicial([
    [null, black, white, black, black, white, white, null],
    [black, empty, empty, empty, empty, empty, empty, black],
    [white, empty, empty, empty, empty, empty, empty, white],
    [white, empty, empty, empty, empty, empty, empty, black],
    [black, empty, empty, empty, empty, empty, empty, white],
    [black, empty, empty, empty, empty, empty, empty, black],
    [white, empty, empty, empty, empty, empty, empty, white],
    [null, white, black, black, white, black, white, null]
]).

tabuleiroFinal([
    [null, empty, empty, empty, empty, white, empty, null],
    [empty, empty, empty, black, black, black, empty, empty],
    [empty, empty, white, empty, white, white, black, empty],
    [empty, black, black, black, black, black, white, empty],
    [empty, empty, empty, empty, white, black, white, empty],
    [empty, black, white, empty, white, white, white, empty],
    [empty, empty, empty, empty, black, white, empty, empty],
    [null, empty, empty, empty, empty, empty, empty, null]
]).
```

Com o objetivo de dar display do Tabuleiro foi elaborado o seguinte código provisório:

```
symbol(empty, '.').
symbol(null, ' ').
symbol(black, 'B').
symbol(white, 'W').

display_game(Board, Player):-
    tabuleiroFinal(Board),
    format('It\'s ~p\'s turn ~n', [Player]),
    imprimeTabuleiro(Board).

imprimeTabuleiro([Head|Tail]) :-
    write('|'),
    imprimeLinha(Head),
    nl,
    imprimeTabuleiro(Tail).
imprimeTabuleiro([ ]).

imprimeLinha([Head|Tail]) :-
    symbol(Head,S),
    format(' ~s |', [S]),
    imprimeLinha(Tail).
imprimeLinha([ ]).
```

-Para se alternar entre tabuleiros, deverá alterar-se a primeira linha de código no predicado `display_game` atribuindo à variável `Board` o tabuleiro desejado. (Código provisório)

## Output:

-Tabuleiro inicial e final apresentados nas figuras 1 e 4 nas paginas anteriores do relatorio

```
| ?- display_game(Board, francisco)
```

```
It's francisco's turn
```

	B	W	B	B	W	W	
B	.	.	.	.	.	.	B
W	.	.	.	.	.	.	W
W	.	.	.	.	.	.	B
B	.	.	.	.	.	.	W
B	.	.	.	.	.	.	B
W	.	.	.	.	.	.	W
	W	B	B	W	B	W	

```
| ?- display_game(Board, francisco).
```

```
It's francisco's turn
```

.	.	.	B	B	W	.	.
.	.	W	.	W	W	B	.
.	B	B	B	B	B	W	.
.	.	.	.	W	B	W	.
.	B	W	.	W	W	W	.
.	.	.	.	B	W	.	.
.	.	.	.	.	.	.	.