

Faculdade de Engenharia da Universidade do Porto



FUSE: Análise do Jogo

Programação em Lógica 2019/2020 - MIEIC:

Grupo FUSE_1:

André Filipe Magalhães Rocha, up201706462@fe.up.pt

Manuel Monge dos Santos Pereira Coutinho, up201704211@fe.up.pt

1 - Introdução	3
2 - FUSE	3
2.1 - História	3
2.2 - Regras	3
2.2.1 - Preparação do Tabuleiro	3
2.2.2 - Desenvolvimento do Jogo	4
2.2.3 - Fim do Jogo	4
2.2.4 - Vencedor	5
3 - Modelação do Jogo	5
3.1.1 - Estado Inicial	5
3.1.2 - Estados Intermédios	6
3.1.3 - Estado final	6

1 - Introdução

Na Unidade Curricular de Programação em Lógica, foi-nos proposta a escrita de um relatório preliminar relativamente ao desenvolvimento de um Jogo de Tabuleiro em Prolog. A escolha deste foi feita com base numa lista pré determinada, sendo que o grupo selecionou o **FUSE**.

Durante o presente documento, será apresentado uma pequena história da criação do jogo, assim como das suas regras. Será ainda feita uma demonstração em Prolog de como o jogo será modelado, apresentando diversos estados do mesmo e a respetiva apresentação no terminal.

É de notar que todo o código apresentado durante este trabalho foi escrito utilizando [SWI-Prolog](#), utilizado maioritariamente pelas suas capacidades de escrever caracteres com formatação com **atributos ANSI**.

2 - FUSE

2.1 - História

Néstor Romeral Andrés é um criador de jogos de tabuleiro com elevada experiência. Com **mais de 120** jogos de tabuleiro já criados, FUSE inspira-se noutra criação sua, [Feed the Ducks](#) do tipo 'push/attract', revelando uma crescente maturidade em relação ao seus jogos passados.

2.2 - Regras

Durante um jogo de FUSE, ambos os jogadores introduzem, alternadamente, discos dentro do tabuleiro, onde poderão eventualmente empurrar outros discos no caminho, de modo a criarem o grupo da sua cor de maior tamanho.

2.2.1 - Preparação do Tabuleiro

São usados 12 discos brancos e 12 discos pretos, espalhados aleatoriamente pelo tabuleiro, de modo a que não haja mais do que dois discos da mesma cor colocados consecutivamente. Uma possível configuração inicial é apresentada na Figura 1.

No caso de ambos os jogadores discordarem da disposição inicial das peças, esta pode ser descartada e será gerada uma nova.

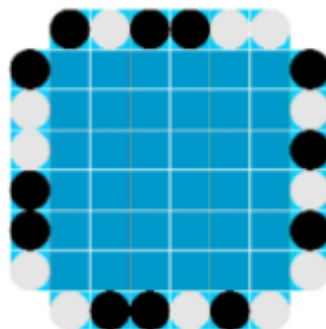


Figura 1 - Exemplo de uma possível disposição inicial do tabuleiro

2.2.2 - Desenvolvimento do Jogo

O jogo é iniciado pelo jogador que escolheu os discos brancos, lançando-se alternadamente um novo disco para dentro da área 6x6 interior do tabuleiro.

Os jogadores devem lançar um novo disco para dentro do tabuleiro se possível. Caso contrário, o jogador deverá passar a jogada. Os discos podem-se mover ao longo da linha ou coluna onde começaram, sem qualquer tipo de restrições relativamente ao número de casas a avançar, empurrando os discos à sua frente como resultado.

Qualquer disco já em jogo, deverá manter-se sempre na área 6x6 do tabuleiro, uma vez que não é possível puxar o disco para a periferia ou fora do mesmo.

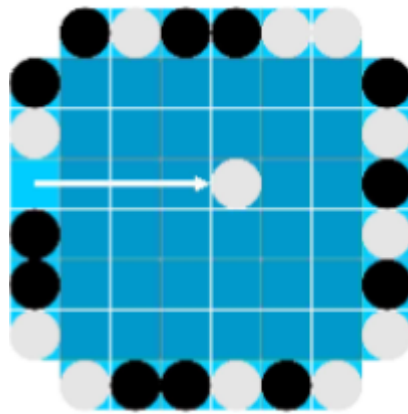


Figura 2 - Exemplo de uma primeira jogada

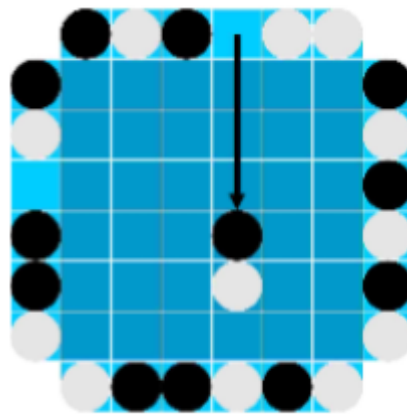


Figura 3 - Exemplo de uma resposta válida

2.2.3 - Fim do Jogo

O jogo termina quando ambos os jogadores passam a vez consecutivamente, ou seja, nenhum dos dois consegue trazer uma nova peça da área inicial do tabuleiro de jogo para o quadrado 6x6 respeitando todas as regras previamente explanadas.

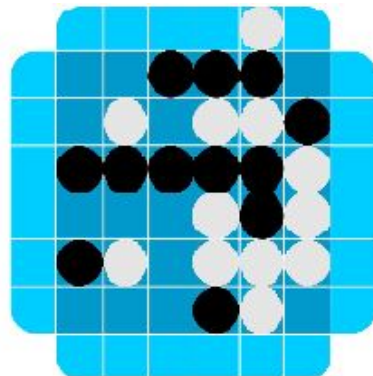


Figura 4 - Exemplo de um final de jogo

2.2.4 - Vencedor

Existem várias formas de pontuar o jogo e decidir o vencedor. O caso mais simples baseia-se na contagem do maior grupo de peças conectadas ortogonalmente (na imagem acima ganham as brancas 7 vs 6). Um modo mais avançado consiste em multiplicar o tamanho de todos os grupos (ganham as pretas pois $6 \times 3 \times 1 \times 1 \times 1 = 18$ vs $7 \times 2 \times 1 \times 1 \times 1 = 14$). Existe ainda o modo ‘Kings’ onde cada rei é assinalado com uma peça azul por cima e vale 2 pontos na contagem final (o número de réis a que cada jogador tem direito é acordado no início).

3 - Modelação do Jogo

Uma vez que o nosso tabuleiro 6x6, possui adjacente a ele em cada um dos lados uma entrada de 5 quadrados onde se encontram os discos, foi necessário procurar uma solução para lidar com estes espaços que mais tarde não poderiam ser ocupados. Deste modo foi criado o átomo **null**, cuja função será preencher as posições do tabuleiro que **não poderão ser mais ocupados**.

```
wt:- ansi_format([bold, fg(white)], '~c', [9679]). % Disco branco
bl:- ansi_format([bold, fg(black)], '~c', [9679]). % Disco preto
null:- write(' '). % Disco nulo
corner:- write(' '). % Canto do Tabuleiro
empty:- ansi_format([bold, bg(cyan)], '~s', [' ']). % Lugar vazio
```

Figura 1 - Traduções para o terminal

3.1.1 - Estado Inicial

Uma vez que o estado do tabuleiro não será sempre o mesmo em cada início de um jogo de FUSE, apenas é possível mostra um possível exemplo de como seria o estado do tabuleiro caso este fosse aceite. Um tabuleiro apenas é aceite caso este não possua mais que duas peças da mesma consecutivamente.

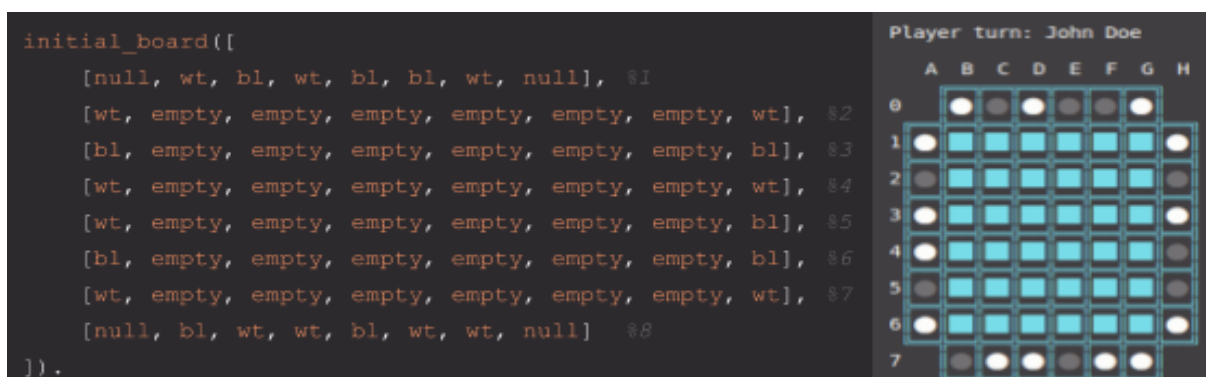


Figura 2 - Exemplo um tabuleiro no seu estado inicial

3.1.2 - Estados Intermédios

A cada jogada, cada um dos jogadores coloca um disco em jogo, pelo o que apenas irá abdicar de tal direito caso não seja possível colocar nenhum dos seus discos em jogo.

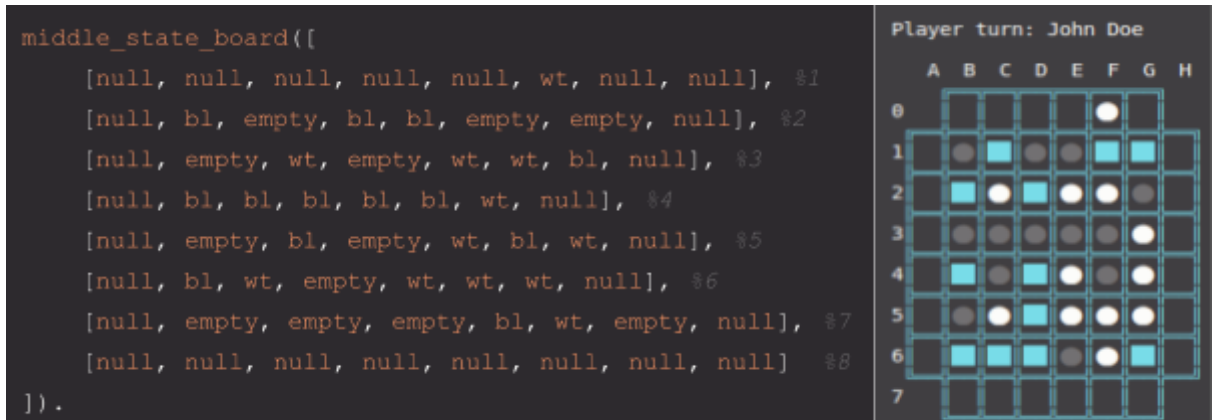


Figura 3 - Exemplo de Estado Intermédio

É de notar que aquando a colocação em jogo de um determinado disco, este nunca mais se poderá mover, excepto quando um disco adversário colidir com o mesmo.

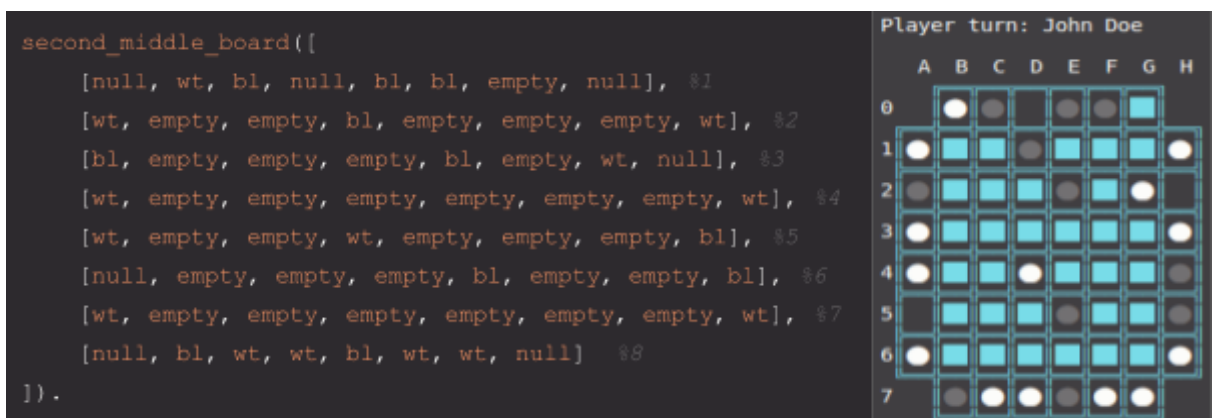


Figura 4 - Mais um possível estado intermédio do tabuleiro

3.1.3 - Estado final

Um jogo de FUSE termina quando nenhum movimento é possível, pelo que ambos os jogadores passam duas vezes. Um possível exemplo de um tabuleiro no seu estado final é apresentado em baixo.

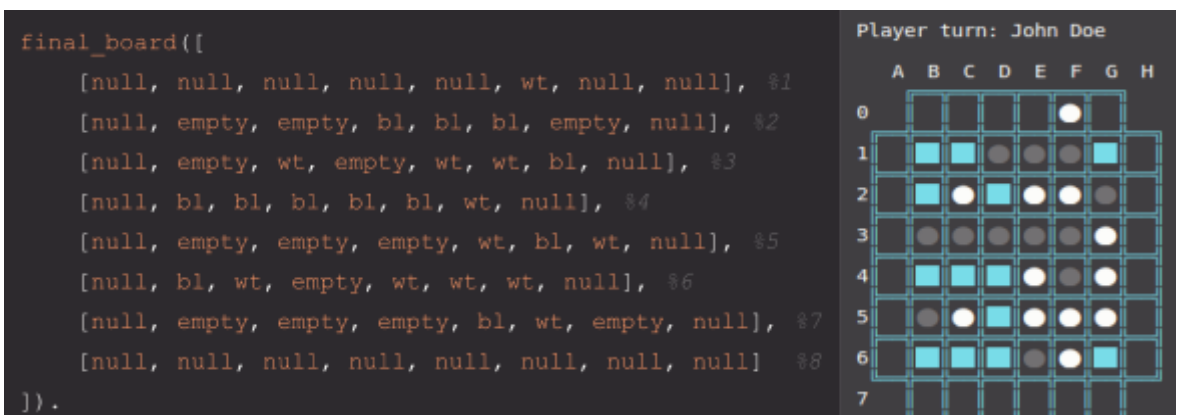


Figura 5 - Tabuleiro num possível estado final