# Fines em Prolog

**Relatório Intercalar**


Mestrado Integrado em Engenharia Informática e Computação

Programação em Lógica

**Grupo Fines 2:**
Pedro José Leal de Sousa - 201205016
Vítor Filipe Oliveira Teixeira - 201208256

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

12 de Outubro de 2014

# 1 O Jogo Fines

Fines é um jogo de tática de tabuleiro para dois jogadores criado por Dieter Stein em 2014. O jogo é composto por um tabuleiro 7x7, 7 peças de cor branca, 7 de cor vermelha e cercas num número considerável (aprox. 50).
Cada jogador tem como objetivo chegar ao final do jogo com a maior área fechada pertencente às suas peças. Uma área fechada é um espaço delimitado por cercas em que no interior apenas se encontra uma peça de qualquer um dos jogadores.

jogadas são feitas alternadamente, cada jogador pode realizar uma das seguintes opções:

1. Mover uma peça e colocar uma cerca.
2. Colocar uma peça nova.

Uma peça só pode ser movida na horizontal e na vertical quantas casas o jogador quiser, aquando do movimento o jogador poderá virar a direção da peça para a direita uma vez e continuar a move-la até encontrar uma cerca ou outra peça. Após o movimento o jogador terá de construir uma cerca, esta não poderá fechar áreas vazias ou fechar áreas que contenham mais do que uma peça. Caso o posicionamento da cerca não seja possivel, a jogada também o será. No caso da adição de uma peça esta só poderá ser adicionada à distância de um movimento de uma das peças do jogador.

O jogo acaba quando apenas existirem regiões fechadas no tabuleiro, e ganha o jogador que conseguir uma maior área. No caso de existirem jogadores com diferentes experiências, o jogador com maior experiência deverá jogar com 5 ou 6 peças dependendo da diferença.

# 2 Representação do Estado do Jogo

Sendo Fines um jogo de tabuleiro, representa-mos numa lista de listas.
Cada sublista representará uma linha do tabuleiro guardando a informação necessária para a análise do jogo, isto é, a ocupação das células e a existência de paredes, seguindo o código:

- 0 – Célula neutra não ocupada
- 1 – Célula com peão do jogador 1 sem parede por baixo
- 2 – Célula com peão do jogador 2 sem parede por baixo
- 3 – Célula com peão do jogador 1 com parede por baixo
- 4 – Célula com peão do jogador 2 com parede por baixo
- ‘ ‘ – Espaço reservado para uma possível parede
- ‘-‘ – Parede

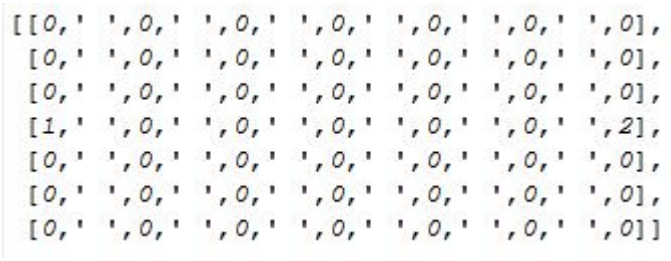


Figura 1: Lista Inicial

# 3 Visualização do Tabuleiro

Em modo de texto, a visualização do tabuleiro é feita através da chamada da função drawBoard(B), em que B é um tabuleiro.
Código de implementação:

printline([]).

printline([H—T]):- write(H), printline(T).

draw([H—T]):- printline(H), nl, draw(T).

drawBoard(B):- board(B), draw(B).

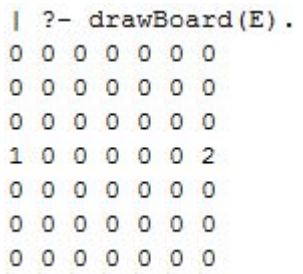


Figura 2: DrawBoard(B)

# 4 Movimentos

A cada jogada o jogador poderá optar por duas opções, colocar um novo peão no tabuleiro ou mover um peão já presente e colocar uma cerca.
Para colocar um novo peão deverá ser fornecido as variáveis, Tabuleiro, linha e coluna em que o pretende colocar.

AddPiece(Board,Col,Line).
Para mover um peão deverá ser fornecido as variáveis, Tabuleiro, linha e coluna iniciais e finais e em que sentido pretende colocar uma parede.

MovePiece(Board,Col$_i$, $Line_i$, $Col_f$, $Line_f$, $Side$).