Mestrado Integrado em Engenharia Informática e Computação

PLOG

Lear



Trabalho Prático 1 - Relatório Final

**Grupo Lear\_4**

Leonor Mendes de Freitas 201207603

Marisa Oliveira 201308594

**Resumo**

No âmbito da unidade curricular de programação em lógica enquadrada no 3ºano do plano de estudos do mestrado integrado em engenharia informática e de computação foi proposto aos alunos o desenvolvimento de um jogo de tabuleiro em Prolog.

// Resumo sucinto do trabalho com 150 a 250 palavras (problema abordado, objetivo, como foi o problema resolvido/abordado, principais resultados e conclusões).

**ÍNDICE**

Contents

Introdução 2

O JOGO LEAR 3

História 3

Constituição 3

Regras 3

Movimentos 3

LÓGICA DE JOGO 4

REPRESENTAÇÃO DO ESTADO DE JOGO 4

VISUALIZAÇÃO DO TABULEIRO 6

Lista de jogadas válidas 8

Execução de jogadas 8

Avaliação do tabuleiro 8

Final do jogo 8

Jogada do Computador 8

Interface com o utilizador 8

Conclusões 8

Bibliografia 8

Anexos 8

# Introdução

//Descrever os objetivos e motivação do trabalho. Descrever num parágrafo breve a estrutura do resto do relatório.

O objetivo do desenvolvimento deste trabalho foi implementar, em linguagem Prolog, um jogo de tabuleiro para dois jogadores com diferentes jogadas e com condições de terminação como derrota, vitória ou empate. O jogo permite três modos de utilização: humano versus humano, humano versus computador e computador versus computador – com níveis de jogo para o computador.

(Nota: verificar se tudo foi implementado como aqui escrito)

A nossa escolha recaiu no jogo Lear porque para além de nos evidenciar ser um jogo interessante também nos pareceu um trabalho bastante apropriado para desenvolver em Prolog.

O relatório está subdividido em várias partes:

//TODO

# 

# O JOGO LEAR

O Lear é um jogo de território, num tabuleiro quadrado com dimensões entre 7x7 e 10x10, para dois jogadores. Os jogadores podem jogar com peças pretas (Black) ou com peças brancas (White). A dimensão escolhida, neste caso, foi 8x8.

O tabuleiro está inicialmente vazio e os jogadores colocam à vez uma peça num ponto no tabuleiro, sendo que Black começa primeiro.

Na sua vez, o jogador deve colocar uma peça da sua cor num ponto vazio. Se o ponto escolhido faz com que a peça passe a fazer parte de uma linha ortogonal contínua de peças, composta por duas peças amigáveis e uma linha contínua de peças inimigas, estas peças inimigas são substituídas por peças da cor oposta.

As duas peças amigáveis podem estar ou as duas juntas numa das pontas da linha ou uma em cada ponta.

x+oo ... +xoo ... +ooox ... +xxooo

Nos primeiros três diagramas acima, jogar um x no local + substitui as peças o por x. No quarto diagrama, tal já não se verifica.

O jogo termina quando o tabuleiro está completamente preenchido. O vencedor é o jogador com a pontuação mais alta. A pontuação de um jogador é igual ao número de peças da sua cor no tabuleiro mais, quando apropriado, um komi.

O komi são pontos que são adicionados à pontuação do jogador que não faz a última jogada. Para evitar empates, o komi deve ser par se a dimensão do tabuleiro for ímpar e vice-versa. Nos tabuleiros ímpares o komi deve ser atribuído ao White e nos tabuleiros pares o komi deve ser atribuído ao Black.

Antes do jogo começar, o primeiro jogador define qual o valor do komi e o segundo jogador escolhe qual a cor com que quer jogar, sendo que o Black é o primeiro a colocar uma peça no tabuleiro.

## História

## Constituição

## Regras

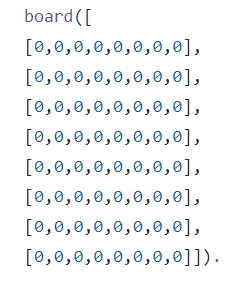
## Movimentos

# LÓGICA DE JOGO

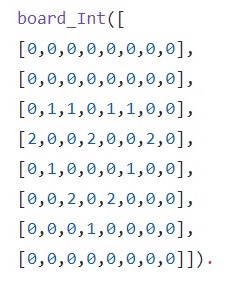
## REPRESENTAÇÃO DO ESTADO DE JOGO

A representação do *Lear* em Prolog consiste numa lista de listas, sendo que cada uma destas representa uma linha do tabuleiro. Cada linha é composta por símbolos que correspondem a cada ponto do tabuleiro, sendo que o valor do mesmo determina o estado daquela célula (vazia, peça preta ou peça branca). A primeira lista (da lista de listas) corresponde à primeira linha e o primeiro elemento de cada lista corresponde ao ponto mais à esquerda dessa linha.

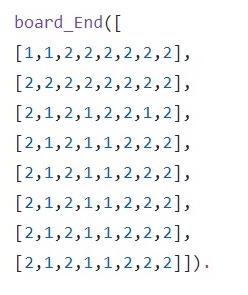
**REPRESENTAÇÃO INICIAL**



**REPRESENTAÇÃO INTERMÉDIA**



**REPRESENTAÇÃO FINAL**



## VISUALIZAÇÃO DO TABULEIRO

A visualização do tabuleiro irá ser feita através de caracteres ASCII impressos na consola, de forma a que cada caracter corresponda a uma célula vazia ou a uma célula ou com uma peça preta ou com uma peça branca. A visualização inicial representa um tabuleiro vazio.

Nas imagens abaixo estão exemplos da visualização do tabuleiro em vários estados, bem como alguns exemplos de código.

**VIZUALIZAÇÃO DO TABULEIRO**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DISPLAY BOARD

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

% imprime as coordenadas das

colunas e respetivas barras

view(Board):-

nl,nl, write(' 1 2 3 4 5 6 7 8'), nl,

lines(1,Board),nl,!.

% imprime recursivamente

as linhas do tabuleiro

lines(\_,[]).

lines(N,[L|Ls]):-

write(N), write('. ') ,show\_line(L), nl,

N1 is N+1,

lines(N1, Ls).

% imprime recursivamente os caracteres de cada linha do tabuleiro

show\_line([]).

show\_line([El|Es]):-

translate(El),

show\_line(Es).

% imprime o tabuleiro

printBoard(Board):-

board(Board),

view(Board).

% imprime o tabuleiro

printBoardInit(Board):-

board\_Int(Board),

view(Board).

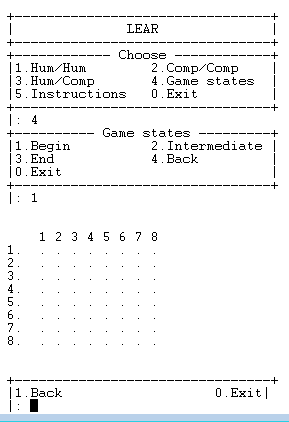
% imprime o tabuleiro

printBoardEnd(Board):-

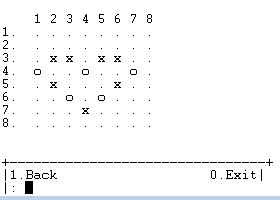
board\_End(Board),

view(Board).

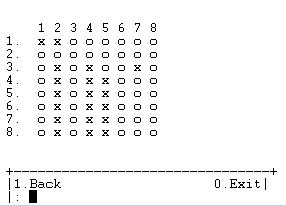
**MENU E VISUALIZAÇÃO INICIAL**



**VISUALIZAÇÃO INTERMÉDIA**



**VISUALIZAÇÃO FINAL**



## Lista de jogadas válidas

## Execução de jogadas

## Avaliação do tabuleiro

## Final do jogo

## Jogada do Computador

# Interface com o utilizador

# Conclusões

# Bibliografia

# Anexos