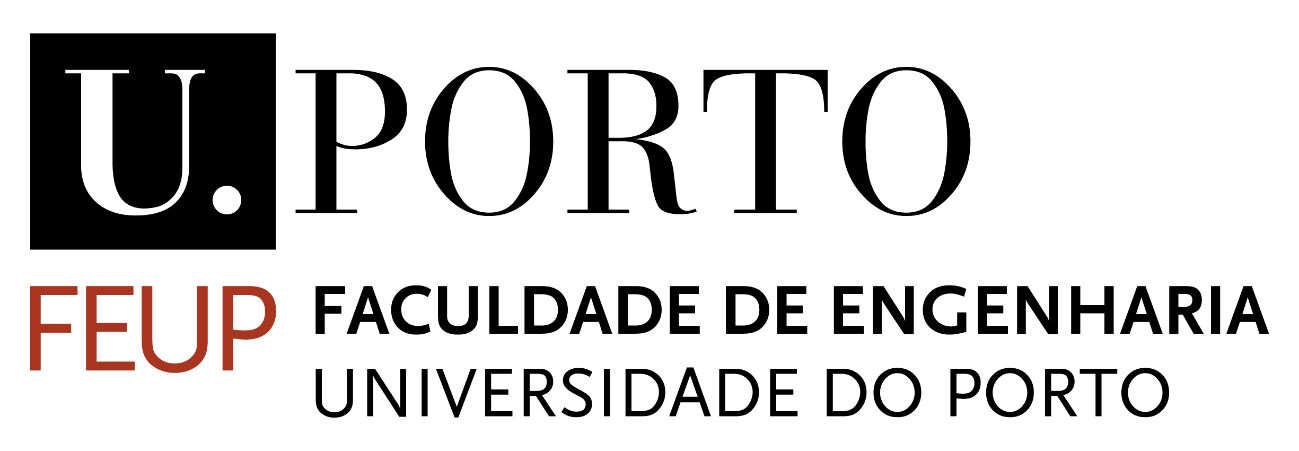
**Oshi**

Relatório Intercalar



Mestrado Integrado em Engenharia Informática e Computação

**Programação em Lógica**

Grupo: Oshi\_3

Bruno Marques up201405781

Diogo Silva up201405742

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

16 de Outubro de 2016

1. **Oshi, o jogo**

Oshi é um jogo Japonês inspirado numa lenda cujo significado do nome se traduz para algo como “Empurrar”. Foi agora convertido para um jogo de tabuleiro para dois jogadores publicado pela “Abysse Corp.” e “Wizkids”. É um jogo de pensamento estratégico cuja duração média é de 15 a 20 minutos.

1. **Objetivos**

O jogo desenrola-se através de um duelo entre dois jogadores moderado por um sistema de pontos, assim que um jogador empurrar o correspondente a 7 pontos (no valor das peças á frente descrito) do adversário para fora do tabuleiro este será o vencedor, desta forma nunca haverá empates.

1. **O tabuleiro e as peças**

O jogo é constituído por um tabuleiro 9x9 e 16 peças, 8 controladas por cada jogador de cores vermelha e branca.

Figura - Embalagem e tabuleiro de Oshi

Existem três tipos de peças, sendo que todas adotam a forma de uma habitação clássica japonesa existem 4 peças de um andar, 2 peças de dois andares e 2 peças de três andares. A sua mobilidade e valor pontual estão diretamente relacionados com este fator.

1. **Regras**

**4.1. Colocação das peças**

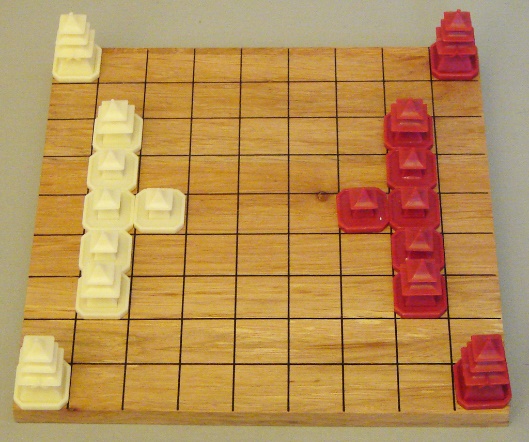


Figura - Peças colocadas no tabuleiro

As peças de jogo são colocadas por cor respetivamente em lados opostos do tabuleiro, as duas peças de 3 andares nos cantos do tabuleiro, os peças de 2 andares dispõem-se nas extremidades da formação uma linha á frente das peças de 3 andares e por fim as peças de 1 andar dispõem se em T invertido, no inferior centro do tabuleiro como podemos ver na imagem.

**4.2. Início do jogo**

O jogo inicia-se com um dos jogadores escolhido aleatoriamente a fazer a primeira jogada, podendo mover qualquer peça.

* 1. **Jogadas**

A cada turno um jogador pode efetuar três tipos de jogadas:

* Movimento simples, quando uma peça se move até um numero de casas inferior ou igual ao numero de andares da mesma.
* Empurrar, quando uma peça colide com outra peça e faz com que a outra se mova.
* Empurrar para fora do tabuleiro, quando a colisão causa que uma ou mais peças caiam do tabuleiro obtendo os pontos respetivos. De notar que em certos casos este movimento pode também incluir a queda de uma peça amigável sendo assim os pontos deverão ser distribuídos de forma normal, atribuindo os pontos respetivos ao jogador no seu turno pela peça que empurrou e ao jogador que está a observar pela peça do seu opositor.

1. **Estados de Jogo**

A nível computacional o tabuleiro de jogo será representado por uma matriz, cujos elementos são outras matrizes que representarão as peças em questão. Haverá então três estados de jogo.

O estado de inicial, que consiste no tabuleiro sem peças, que se cria usando a função create\_board(B), em que B é o tabuleiro.

create\_board(H):-

length(H,9),

create\_rows(H).

create\_rows([]).

create\_rows([H|T]):-

length(H,9),

create\_pieces(H),

create\_rows(T).

create\_pieces([]).

create\_pieces([H|T]):-

piece\_empty(H),

create\_pieces(T).

O estado de jogo inicial com as peças colocadas nas posições respetivas.

O estado intermédio quando as peças já não estão na sua posição inicial e ocupam agora posições dependentes das jogadas efetuadas.

O estado final quando peças equivalentes a 7 pontos já tenham sido empurradas para o exterior do tabuleiro.

1. **Visualização do Tabuleiro**

A visualização do tabuleiro é feita usando a função display\_board(B), em que B é o tabuleiro.

Código usado para a função:

display\_board(H):-

display\_line(73),

display\_line(73),

display\_board\_aux(H).

displat\_board\_aux([]).

display\_board\_aux([H|T]):-

display\_board\_row(H,1),

display\_board\_aux(T).

display\_board\_row([],N).

display\_board\_row(H,N):-

N>3,

display\_line(73).

display\_board\_row(H,N):-

N < 4,

display\_board\_line(H,N),

write('|'),

nl,

N1 is N+1,

display\_board\_row(H,N1).

display\_board\_line([],N).

display\_board\_line([H|T],N):-

write('| '),

nth1(N,H,R),

display\_piece\_line(R),

write(' '),

display\_board\_line(T,N).

display\_piece\_line([]).

display\_piece\_line([H|T]):-

write(H),

display\_piece\_line(T).

display\_line(N):-

N<1,

nl.

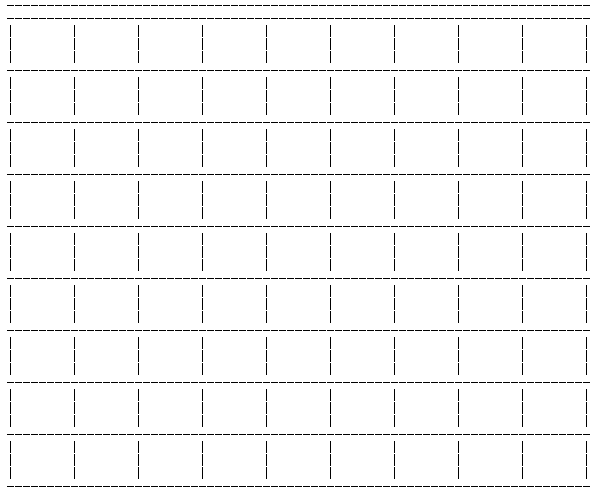
display\_line(N):-

N>0,

N1 is N-1,

write('-'),

display\_line(N1).

Resultado da impressão de um tabuleiro vazio:

1. **Movimentos**

Depois de colocadas as peças nas posições iniciais em conformidade com as regras do jogo, e selecionado o jogador que irá iniciar os movimentos podem se executar de acordo com a seguinte sintaxe:

* Movement (P, X, Y, V, D).

Legenda:



* P – Jogador (Player);
* X – Coordenada horizontal da peça a mover;
* Y – Coordenada vertical da peça a mover;
* V – Valor a deslocar;
* D – Direção da deslocar.

Esta direção varia ortogonalmente, ou seja, apenas pode ser executada para Norte, Sul, Este ou Oeste.

1. **Conclusão**

Depois de refletir sobre os vários aspetos deste projeto, esclarecer as regras a implementar e projetar os elementos mais básicos do jogo olhamos para este projeto como um desafio interessante, pois ao utilizar uma linguagem declarativa teremos perante nós uma tarefa algo diferente daquelas a que estamos habituados.

Esperamos portanto ansiosamente começar a desenvolver o código que permitirá implementar este jogo de forma computacional.