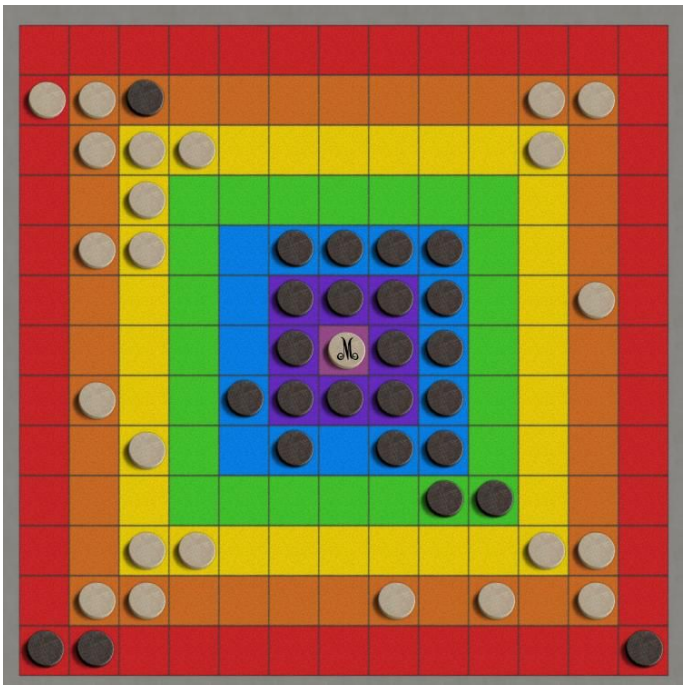
U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Pesquisa com Adversários - *Morelli*

Relatório Intercalar

Inteligência Artificial

3º ano do Mestrado Integrado em Engenharia Informática e Computação



Elementos do Grupo:

Bernardo José Coelho Leite – 201404464 – up201404464 @gcloud.fe.up.pt
Francisco José Sousa Silva – 201502860 – francisjssilva@gmail.com
Francisco Tomé Macedo Martins Santos Moreira - 201607929 - up201607929@fe.up.pt

8 de abril de 2018

Objetivo

No âmbito da Unidade Curricular de Inteligência Artificial, o nosso grupo apresenta este relatório com o objetivo de descrever o nosso trabalho.

O **objetivo** do tema escolhido é incidir sobre os tópicos da **Pesquisa Adversarial** com a utilização de **regras de decisão** no contexto da **teoria da decisão**, **teoria do jogo**, **estatística** e **técnicas de minimização de possíveis perdas** para cenários pessimistas (perda máxima). A nossa implementação conta, por isso, com a aplicação do algoritmo ***Minimax*** e derivados para concretizar decisões na presença de incertezas.

Com vista a cumprir os objetivos acima descritos, o grupo escolheu implementar o jogo de Tabuleiro - ***Morelli***. Veja-se agora uma breve descrição das suas regras:

- É possível jogar-se *Morelli* num tabuleiro composto por 13x13 células quadradas que são coloridas em **faixas concêntricas** através das cores do arco-íris;

- Na faixa mais afastada existem 48 quadrados de cor vermelha seguidos pelos da cor laranja, amarela, verde-claro, verde-escuro e roxo. O seguimento destas faixas aponta para a célula central, o **Trono**;

- Cada um dos jogadores começa com 24 peças, dispostas inicialmente na faixa vermelha e em cada uma das jogadas move-se uma peça;

- As peças usufruem da **liberdade das Rainhas no Xadrez** exceto que em cada jogada, estas peças terão de se aproximar para uma faixa mais próxima do trono;

- Uma peça cercada por duas peças adversárias (ortogonalmente ou diagonalmente) é **convertida** numa peça do outro jogador;

- Um jogador que contenha **4 peças dispostas em simetria radial** em relação ao centro, **captura-o**;

- A finalidade do jogo é ter **controlo do centro** quando já não houver mais jogadas possíveis.

Descrição

- **Estados**

Para representar cada estado do jogo iremos utilizar a estrutura de *int[][] array*, segundo a linguagem de programação Java. Para representar as nossas peças em modo texto adotamos o seguinte formato: *0 - posição vazia; 1 - Jogador com Peça de cor Branca; 2 - Jogador com Peça de cor Preta*. Seguem-se os estados de jogo possíveis (estado **inicial**, **intermédio** e **final**):

```
1 1 1 2 1 2 2 1 2 2 1 1 2
2 0 0 0 0 0 0 0 0 0 0 0 1
2 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 0 0 0 2
2 0 0 0 0 0 0 0 0 0 0 0 2
1 0 0 0 0 0 0 0 0 0 0 0 2
1 0 0 0 0 0 0 0 0 0 0 0 1
2 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 0 0 0 2
2 0 0 0 0 0 0 0 0 0 0 0 1
2 0 0 0 0 0 0 0 0 0 0 0 2
1 2 2 1 1 1 2 2 2 1 2 1 2
```

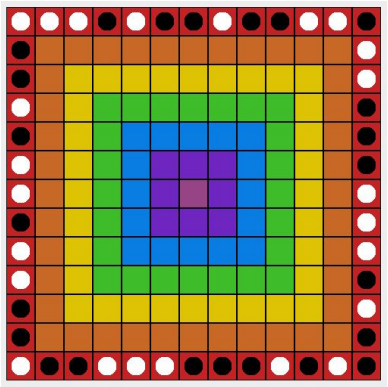


imagem 1 - Estado Inicial com a disposição inicial das peças.

```
0 1 1 2 2 1 0 0 0 0 1 2 0
0 0 0 2 0 0 0 0 0 0 0 0 2
0 0 0 0 0 0 0 2 0 0 0 0 2
1 0 0 0 0 0 0 0 0 0 0 0 2
1 0 0 0 1 0 0 0 1 0 0 0 1
2 0 0 0 0 0 0 0 0 0 0 0 2
1 0 0 0 0 0 1 0 0 0 0 0 1
2 0 2 0 0 0 0 0 0 0 0 0 1
2 0 0 0 1 0 0 0 1 0 0 0 1
1 0 0 0 0 0 0 0 0 2 0 0 2
2 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 0 0 0 1
2 1 1 1 2 1 2 2 2 1 2 2 0
```

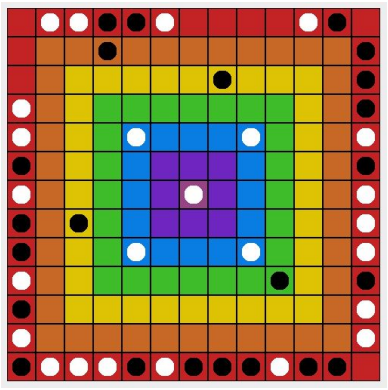


imagem 2 - Estado Intermédio numa situação Captura.

```
0 2 0 0 2 2 0 0 0 0 1 1 0
0 0 0 2 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 2 0 0 0 0 2
2 0 0 0 0 0 0 0 0 2 0 0 1
1 0 0 0 1 0 0 0 2 0 0 0 1
1 0 0 0 0 0 0 2 0 0 0 0 1
1 0 0 0 0 0 1 0 0 0 0 0 1
2 0 2 0 0 0 0 0 0 0 0 0 2
2 0 0 0 1 0 0 0 1 0 0 0 2
1 0 0 0 0 0 0 0 0 2 0 0 1
2 0 0 0 0 0 0 0 0 0 0 0 2
2 0 0 0 0 0 0 0 0 0 0 0 1
1 2 1 2 2 2 2 1 1 2 1 2 0
```

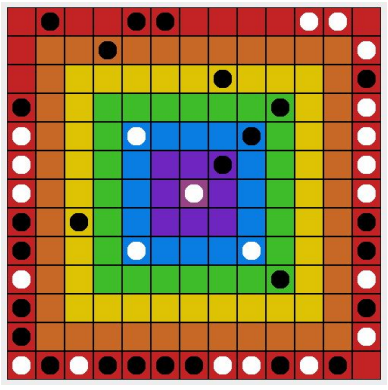


imagem 3 - Estado Intermédio na situação em que uma das Peças é cercada e convertida.

```
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 2 2 0 2 2 2 0 2 0 0 0
0 0 0 2 2 2 2 1 2 0 0 0 0
0 0 0 2 2 2 2 2 1 0 0 0 0
0 0 0 1 1 1 1 2 0 1 0 0 0
0 0 0 1 2 2 1 2 1 0 0 0 0
1 1 0 1 1 1 2 1 1 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
```

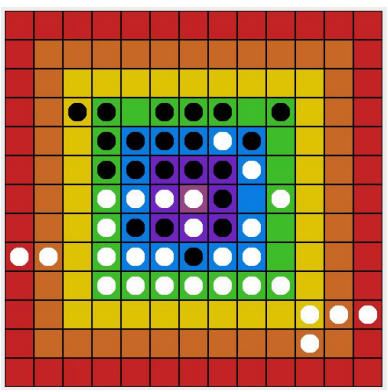


imagem 4 - Estado Final em que o Jogador com as peças brancas é o vencedor (centro capturado e não há mais movimentos possíveis).

- **Função de transição**

Considerar a seguinte máquina de estados finita dada por : $A = (\Sigma, S, So, \delta, F)$

Σ é o Alfabeto de Entrada;
S é o conjunto de estados finitos;
So é o estado inicial, elemento de S;
δ é a função de transição de estados em que $\delta: S \times \Sigma \rightarrow S$
F é o conjunto de estados Finais.

Tendo em conta que :

Σ = {*capturar*, *bloqueio*, *mover*} e ainda que,

S0 : Estado Inicial com as Peças dispostas na Primeira Faixa (vermelha) do Tabuleiro;
S1 : Estado Intermédio após Movimento de Peça sem que haja qualquer Captura ou bloqueio;
S2 : Estado Intermédio após Captura de uma Peça;
S3 : Estado Intermédio após Bloqueio de uma Peça;
Qualquer um dos estados (S1,S2.S3) **pode ser um estado final** desde que não haja mais movimentos possíveis.

A **função de transição** define-se pelo seu conjunto de transições: ***δ(S0,mover) = S1, δ(S1, capturar) = S2, δ(S1, bloqueio) = S3, δ(S2, mover) = S1 e δ(S3, mover) = S1***.

- **Função de avaliação**

Apresentamos a seguinte função heurística de avaliação que pondera a soma de vários fatores através da sua influência no valor da posição.

heurística = 1 ***(para vitória)*** **e heurística = -1 (para derrota)**

Se Jogador estiver em modo Ofensivo:

heurística = Captura * 0.4 + Bloqueio de Captura * 0.3 + Cercar Peça * 0.15 + Bloquear Cerco de Peça * 0.15

Se Jogador estiver em modo Defensivo:

heurística = Bloqueio de Captura * 0.4 + Captura * 0.3 + Cercar Peça * 0.15 + Bloquear Cerco de Peça * 0.15

O modo **Ofensivo** é quando o jogador atual tem como prioridade Capturar o centro;
O modo **Defensivo** é quando o jogador atual tem como prioridade Bloquear a Captura do Centro;

Captura = $(\frac{1}{4} * nr\ peças\ já\ em\ simetria\ radial) * 0.7 + (\frac{1}{6} * distância\ radial) * 0.3$

1 <= Nr Peças já em Simetria radial <= 4 $\wedge$ ***1 <= distância radial <= 6***

Bloqueio de Captura = ($\frac{1}{3}$ * Nr de peças prestes a fazer Captura)

1 <= Nr de Peças prestes a fazer Captura <= 3

Cercar Peça = ($\frac{1}{8}$ * Nr de peças convertidas em resultado do Cerco)

1 <= Nr Peças Convertidas em Resultado do Cerco <= 8

Bloquear Cerco de Peça = ($\frac{1}{8}$ * Nr de peças convertidas em resultado do Cerco)

1 <= Nr Peças Convertidas em Resultado do Cerco <= 8

- **Algoritmos de pesquisa a aplicar**

Os algoritmos de pesquisa a utilizar são o ***Minimax*** e o ***Minimax*** **com Cortes Alfa-Beta**. Cada um deles irá utilizar na sua implementação a função heurística anterior para estimar o valor heurístico de um determinado nó.

Trabalho Efetuado

De acordo com a elaboração da planificação do nosso trabalho, apresentamos as seguintes fases e os estados das mesmas:

1. Implementação da Interface Gráfica - **Tabuleiro de Jogo**:
   Concluído
2. Implementação da Interface Gráfica - **Representação das Peças**:
   Concluído
3. Implementação da Interface Gráfica - **Movimento das Peças**:
   Em progresso
4. Implementação da Interface Gráfica - **Realçar os possíveis movimentos de uma peça** quando selecionada:
   A fazer
5. Implementação da Interface Gráfica - **Funcionalidades Avançadas** (Pausar Jogo, Reiniciar Jogo, Guardar Jogo, Sons, Animações, etc, …):
   A fazer
6. Implementação da Lógica - **Tabuleiro de Jogo**:
   Concluído
7. Implementação da Lógica - **Representação das Peças**:
   Concluído
8. Implementação da Lógica - **Representação dos possíveis movimentos** para uma peça:
   Concluído
9. Implementação da Lógica - Identificar a situação de "**Captura do Trono**":
   Concluído
10. Implementação da Lógica - Identificar a situação de "**Cercar uma Peça do Oponente**":
    Concluído
11. Implementação da Lógica - Identificar a situação de **Vitória**:
    Em progresso
12. Implementação da IA - Estabelecer uma **Função de Avaliação**:
    Concluído
13. Implementação da IA - Aplicação do **Algoritmo Minimax:**
    A fazer
14. Implementação da IA - Aplicação do Algoritmo **Minimax com Cortes Alfa-Beta:**
    A fazer
15. Implementação da IA - Aplicação do Algoritmo ***Monte Carlo Tree Search***:
    A fazer

Resultados esperados e forma de avaliação

Para testar o funcionamento dos Algoritmos de Pesquisa implementados no trabalho, iremos submeter o nosso Programa a uma série de situações (forçadas) e observar o seu comportamento. Observe-se os seguintes cenários:

Nota: Considerar no Algoritmo Minimax o parâmetro Profundidade Máxima = 3

Situação 1: *Considere-se um cenário em que o jogador (Max) tem as seguintes opções:*

1. *Colocar a peça junto das 3 peças que perfazem uma captura;*
2. *Colocar a peça de forma a cercar (e converter) uma peça do oponente.*

Sabe-se ainda que:

1. *No seguimento da jogada - opção 1 o jogador (Min) tem a opção de fazer uma jogada que perfaz uma Captura desfazendo, por isso, a Captura anterior;*
2. *No seguimento da jogada - opção 2 o jogador (Min) não tem nenhuma jogada que capture o centro ou que cerque o adversário.*

Comportamento esperado: *De acordo com a Função de Avaliação e do Algoritmo Minimax espera-se que nesta situação o Jogador (Max) escolha a opção 2 e coloque a peça de forma a cercar a peça do jogador oponente (e assim convertê-la). O motivo para tal acontecer é pelo facto do jogador (Max) minimizar a possível perda máxima.*

Situação 2: *Considere-se um cenário em que o jogador (Max) tem as seguintes opções:*

1. *Colocar uma peça de forma a Capturar o Centro e, simultaneamente, cercar (e converter) 1 peça do adversário;*
2. *Colocar uma peça de forma a Capturar o Centro e, simultaneamente, cercar e (converter) 2 peças do adversário.*

Sabe-se ainda que:

1. *No seguimento da jogada - opção 1 o jogador (Min) não tem nenhuma jogada que capture o centro ou que cerque o adversário;*
2. *No seguimento da jogada - opção 2 o jogador (Min) tem a opção de cercar (e converter) três peças do jogador (Max).*

Comportamento esperado: *De acordo com a Função de Avaliação e com o Algoritmo Minimax espera-se que nesta situação o jogador (Max) escolha a opção 1. O motivo para tal acontecer é pelo facto do jogador (Max) minimizar a possível perda máxima.*

Situação 3: *Considere-se um cenário em que o jogador (Max) tem as seguintes opções:*

1. *Colocar uma peça de forma a Capturar o Centro e, simultâneamente, cercar (e converter) 2 peças do jogador (Min);*
2. *Colocar uma peça algures no tabuleiro sem efetuar Captura ou cercar (e converter) uma peça do jogador (Min).*

Sabe-se ainda que:

1. *No seguimento da jogada - opção 1 o jogador (Min) tem a opção de fazer uma jogada que perfaz uma Captura do Centro e para além disso, finaliza o Jogo;*
2. *No seguimento da jogada - opção 2 o jogador (Min) tem a opção de cercar (e converter) uma peça do jogador (Max).*

Comportamento esperado: *De acordo com a Função de Avaliação e com o Algoritmo Minimax espera-se que nesta situação o jogador (Max) escolha a opção 2. O motivo para tal acontecer é pelo facto do jogador (Max) minimizar a possível perda máxima.*

Situação 4: Utilizar o Algoritmo Minimax com Cortes Alfa Beta e compará-lo com o Minimax.

Comportamento esperado: É esperado que a utilização da memória seja reduzida para um fator de ramificação grande.

Conclusões

Ao finalizar a elaboração do Relatório Intercalar o grupo considera que cumpriu grande parte dos objetivos delineados no que diz respeito à tomada de decisões e implementação dessas mesmas no contexto da nossa aplicação. Temos também consciência de que nos falta percorrer um caminho substancial até que o nosso projeto esteja finalizado. Até à data, tivemos como principal objetivo a estruturação e implementação das funções necessárias para a aplicação dos Algoritmos de Pesquisa e, por isso, a partir deste momento, iremos concentrar os nossos esforços na integração da Inteligência Artificial com o nosso programa.

Recursos

Bibliografia

- *Apontamentos das aulas (ppt);*
- *"Artificial Intelligence: A modern approach". S.Russel, P.Norvig, Prentice-Hall, 3rd Edition, 2010;*
- *"Artificial Intelligence for Games". Ian Millington, John Funge, 2nd Edition, 2009;*
- Link*: http://www.boardspace.net/english/about_morelli.html*.

Software e Tecnologias

- Utilização da Linguagem de Programação *Java*;
- Utilização da *widget toolkit Swing*;
- Utilização do Ambiente de Desenvolvimento Integrado 1: *Eclipse*;
- Utilização do Ambiente de Desenvolvimento Integrado 2: *IntelliJ IDEA*.