



3º ANO – MIEIC 2020/2021

Inteligência Artificial

Grupo 58

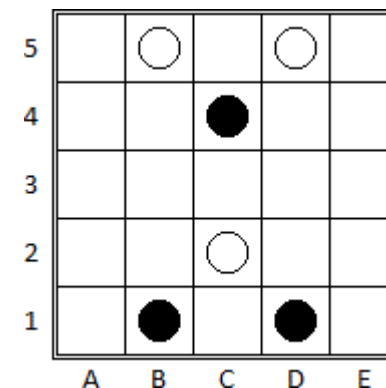
NEUTREEKO

Pedro Coelho, up201806802@fe.up.pt

Ricardo Carvalho, up201806791@fe.up.pt

Xavier Pisco, up201806134@fe.up.pt

Especificação



Neste trabalho vamos criar o jogo neutreeko e resolver o problema de pesquisa com adversários a ele associado.

Este jogo é jogado num tabuleiro com 5 linhas e 5 colunas em que cada um dos dois jogadores tem 3 peças que começam distribuídas conforme a imagem ao lado.

As peças são movimentadas ortogonalmente ou diagonalmente numa linha reta até chegarem ao fim do tabuleiro ou a uma casa que esteja ocupada.

Para ganhar, um jogador tem de formar uma linha ortogonal ou diagonal constituída pelas 3 peças do jogador antes do adversário.

Caso no decorrer do jogo se repita 3 vezes seguidas a mesma posição o jogo acaba em empate.

Formulação do problema

Representação do estado

- Peças pretas -> Matriz de tamanho 3 com as posições das peças
- Peças brancas -> Matriz de tamanho 3 com as posições das peças
- Jogador -> 1 (jogador com peças pretas) ou 2 (jogador com peças brancas)

Estado inicial

| | | |
|-------|-------|-------|
| (1,0) | (3,0) | (2,3) |
| (1,4) | (3,4) | (2,1) |

Jogador = 1

Teste objetivo

Tendo as coordenadas da última jogada, verificar se existe alguma linha de 3 (composta por peças da mesma cor) que seja composta pela peça da última jogada, ortogonalmente ou diagonalmente

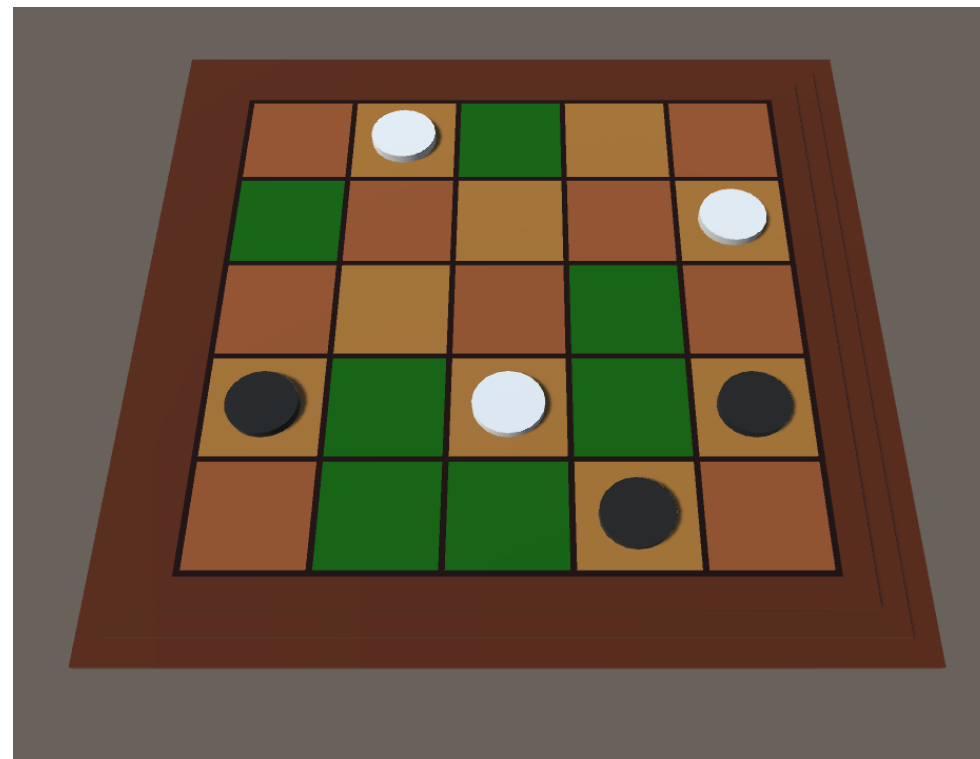
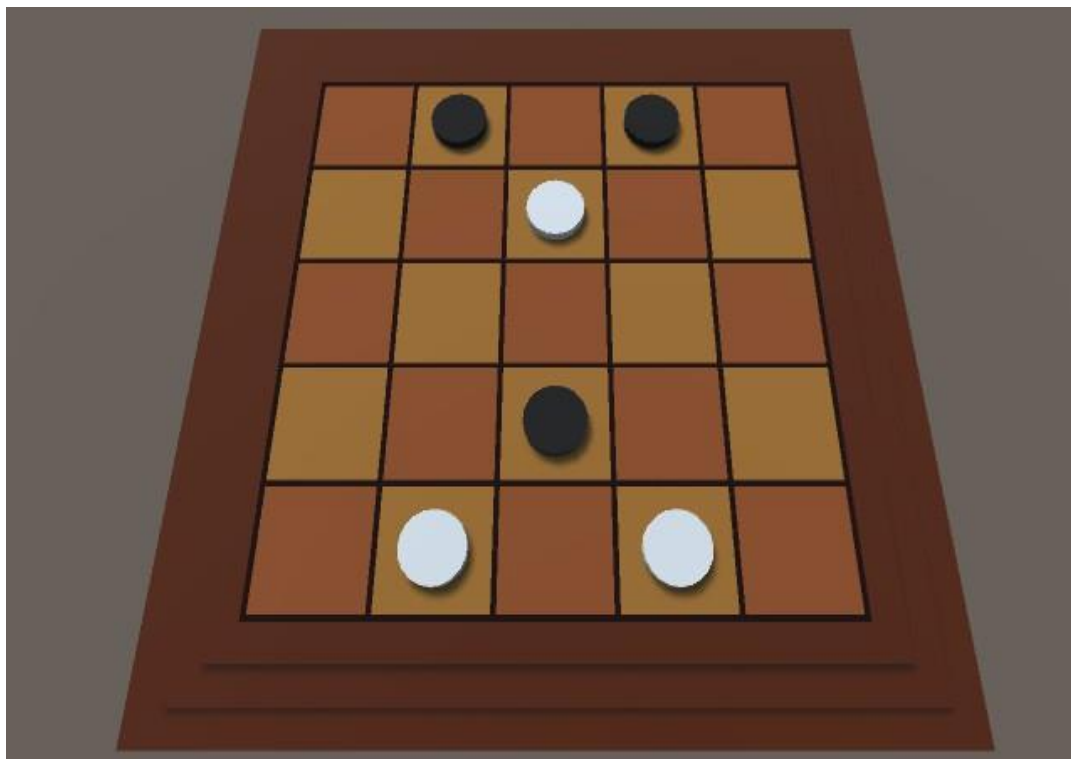
Operadores

| Nome | Pré-condições | Efeito | Custo |
|----------|---|---|-------|
| Cima | $Y_{pe\grave{c}a} > 0$ $\forall pe\grave{c}a2, Posi\grave{c}ao\ pe\grave{c}a2 \neq (X_{pe\grave{c}a}, Y_{pe\grave{c}a} - 1)$ | Posi\c{c}ao pe\c{c}a = (X_{final}, Y_{final}) | 1 |
| Baixo | $Y_{pe\grave{c}a} < 4$ $\forall pe\grave{c}a2, Posi\grave{c}ao\ pe\grave{c}a2 \neq (X_{pe\grave{c}a}, Y_{pe\grave{c}a} + 1)$ | Posi\c{c}ao pe\c{c}a = (X_{final}, Y_{final}) | 1 |
| Direita | $X_{pe\grave{c}a} < 4$ $\forall pe\grave{c}a2, Posi\grave{c}ao\ pe\grave{c}a2 \neq (X_{pe\grave{c}a} + 1, Y_{pe\grave{c}a})$ | Posi\c{c}ao pe\c{c}a = (X_{final}, Y_{final}) | 1 |
| Esquerda | $X_{pe\grave{c}a} > 0$ $\forall pe\grave{c}a2, Posi\grave{c}ao\ pe\grave{c}a2 \neq (X_{pe\grave{c}a} - 1, Y_{pe\grave{c}a})$ | Posi\c{c}ao pe\c{c}a = (X_{final}, Y_{final}) | 1 |

| | | | |
|----------------|---|---|---|
| Cima Direita | $Y_{pe\grave{c}a} > 0, X_{pe\grave{c}a} < 4$ $\forall pe\grave{c}a2, Posi\grave{c}ao\ pe\grave{c}a2 \neq (X_{pe\grave{c}a} + 1, Y_{pe\grave{c}a} - 1)$ | Posi\c{c}ao pe\c{c}a = (X_{final}, Y_{final}) | 1 |
| Cima Esquerda | $Y_{pe\grave{c}a} > 0, X_{pe\grave{c}a} > 0$ $\forall pe\grave{c}a2, Posi\grave{c}ao\ pe\grave{c}a2 \neq (X_{pe\grave{c}a} - 1, Y_{pe\grave{c}a} - 1)$ | Posi\c{c}ao pe\c{c}a = (X_{final}, Y_{final}) | 1 |
| Baixo Direita | $Y_{pe\grave{c}a} < 4, X_{pe\grave{c}a} < 4$ $\forall pe\grave{c}a2, Posi\grave{c}ao\ pe\grave{c}a2 \neq (X_{pe\grave{c}a} + 1, Y_{pe\grave{c}a} + 1)$ | Posi\c{c}ao pe\c{c}a = (X_{final}, Y_{final}) | 1 |
| Baixo Esquerda | $Y_{pe\grave{c}a} < 4, X_{pe\grave{c}a} > 0$ $\forall pe\grave{c}a2, Posi\grave{c}ao\ pe\grave{c}a2 \neq (X_{pe\grave{c}a} - 1, Y_{pe\grave{c}a} + 1)$ | Posi\c{c}ao pe\c{c}a = (X_{final}, Y_{final}) | 1 |

Estas opera\c{c}o\~es aplicam-se a qualquer uma das 3 pe\c{c}as do jogador que seja selecionada
 pe\c{c}a -> pe\c{c}a que foi selecionada para mover
 pe\c{c}a2 -> pe\c{c}a para comparar posi\c{c}ao com a pe\c{c}a selecionada

Estado de implementação



Heurísticas

H1 - Atribuímos pontos a cada par de peças juntas e a cada par de peças com um espaço vazio no meio.

Notação: H1(10, 5) - 10 pontos para peças juntas e 5 pontos para peças com espaço vazio no meio

H2 – Atribuímos pontos a cada par de peças juntas e a cada par de peças com um espaço vazio no meio, bem como a 3 peças com 1 espaço vazio entre elas

Notação: H2(10, 5, 10) – 10 pontos para peças juntas, 5 pontos para peças com espaço vazio no meio e 10 pontos para 3 peças com um espaço vazio entre elas

H3 – Atribuímos pontos a cada par de peças juntas, a cada par de peças com um espaço vazio no meio e a peças que estejam em linha reta com um espaço vazio

Notação: H3(10, 5, 10) – 10 pontos para peças juntas, 5 pontos para peças com espaço vazio no meio e 2 pontos para cada peça que esteja em linha reta com o espaço vazio

H4 – Atribuímos pontos a cada par de peças juntas, a cada par de peças com espaço vazio no meio e mais pontos consoante maior seja a distância ao centro

Notação: H4(10, 5, 2) – 10 pontos para peças juntas e 5 pontos para peças com espaço vazio no meio

H5 – Atribuímos pontos a cada par de peças juntas

Notação: H5(10) – 10 pontos para cada par

H6 – Atribuímos pontos a cada par de peças juntas, a cada par de peças com um espaço vazio no meio e a 3 peças na mesma linha sem peças do adversário no meio

Notação: H6(10, 5, 10) – 10 pontos para peças juntas, 5 pontos para peças com espaço vazio no meio e 10 pontos para 3 peças na mesma linha

Algoritmos

```
function minimax(node, depth) is
  if (player == 0)
    value :=  $-\infty$ 
  else value :=  $+\infty$ ;
  if (depth == 0 or node is a terminal node) then
    return the heuristic value of node;
  for each child of node do
    if (player == 0) then
      value := max(value, minimax(child, depth - 1));
    else (* player == 1 *)
      value := min(value, minimax(child, depth - 1));
  return value;
```

Algoritmo 1: Minimax

```
function alphabeta(node, depth,  $\alpha$ ,  $\beta$ ) is
  if (player == 0) value :=  $-\infty$ ;
  else value :=  $+\infty$ ;
  if (depth == 0 or node is a terminal node) then
    return the heuristic value of node;
  for each child of node do
    if (player == 0)
      value := max(value, alphabeta(child, depth - 1,  $\alpha$ ,  $\beta$ ));
       $\alpha$  := max( $\alpha$ , value);
      if ( $\alpha \geq \beta$ ) then
        break (*  $\beta$  cutoff );
    else
      value := min(value, alphabeta(child, depth - 1,  $\alpha$ ,  $\beta$ ));
       $\beta$  := min( $\beta$ , value);
      if ( $\beta \leq \alpha$ ) then
        break (*  $\alpha$  cutoff *);
  return value;
```

Algoritmo 2: Minimax com cortes alfa-beta

Resultados

Analizando as diversas Fig. Do Anexo 2, concluímos que o Minimax com cortes Alfa-Beta é mais eficiente do que o Minimax.

Nas Fig. 7 e 8 foi usado um sentido diferente para procurar movimentos possíveis para as diferentes peças (sentido do ponteiro dos relógios). Testando esse sentido, concluímos que os resultados foram muito semelhantes relativamente ao sentido que usávamos antes (cima, baixo, direita, esquerda, e depois diagonais). Os resultados finais (vitória/empate/derrota) foram os mesmos usando esse novo sentido.

Conclusões

A realização deste projeto foi bem-sucedido, tendo sido cumpridos todos os objetivos propostos. Conseguiu-se ainda implementar uma interface gráfica apropriada que facilita tanto a visualização como a jogabilidade. Os integrantes do grupo sentem-se satisfeitos com o trabalho realizado e destacam a evolução do seu conhecimento desde o início do semestre no que à área da Inteligência Artificial diz respeito. Para tal, contribuiu largamente a aplicação de conceitos neste trabalho prático, que aumentou ainda o interesse pela área. Foi, portanto, um projeto satisfatório de executar, o que motivou um maior esforço e dedicação.

Por fim, espera-se que os conhecimentos adquiridos e aplicados possam vir a ser úteis ao longo do nosso percurso enquanto engenheiros informáticos e que sirvam de base à aprendizagem de conhecimentos futuros.

Referências

- <https://en.boardgamearena.com/gamepanel?game=neutreeko>
- <https://www.neutreeko.net/neutreeko.htm>

Anexo 1

| Classificação | Heurística |
|---------------|------------|
| 1º | H4 |
| 2º | H2/H6 |
| 3º | H1/H3 |
| 4º | H5 |

Anexo 2

| | Minimax | | Alfa-Beta | |
|---------------|-------------|------------|-------------|----------|
| Depth-3 | H1 - B | H2 - W | H1 - B | H2 - W |
| | 0.021967 | 0.021966 | 0.006616 | 0.008224 |
| | 0.018858 | 0.025777 | 0.002755 | 0.003555 |
| | 0.025867 | 0.02571 | 0.007856 | 0.003637 |
| | 0.017806 | 0.16136 | 0.003506 | 0.006556 |
| | | | | |
| Average | 0.0211245 | 0.05870325 | 0.00518325 | 0.005493 |
| Total average | 0.039913875 | | 0.005338125 | |

Fig1. Comparação dos dois algoritmos (Minimax e Minimax com cortes alfa-beta) entre as heurísticas H1 e H2, usando profundidade 3

Anexo 2

| | Minimax | | Alfa-Beta | |
|---------------|----------|---------|-----------|----------|
| Depth-6 | H1 - B | H2 - W | H1 - B | H2 - W |
| | 39.95 | 39.84 | 0.4036 | 0.6464 |
| | 29.76 | 28.04 | 0.6974 | 0.7252 |
| | 21.16 | 21.26 | 0.2277 | 0.6881 |
| | 29.65 | 28.47 | 0.6881 | 0.735 |
| | | | | |
| Average | 30.13 | 29.4025 | 0.5042 | 0.698675 |
| Total average | 29.76625 | | 0.6014375 | |

Fig2. Comparação dos dois algoritmos (Minimax e Minimax com cortes alfa-beta) entre as heurísticas H1 e H2, usando profundidade 6

Anexo 2

| | Minimax | | Alfa-Beta | |
|---------------|-----------|-----------|------------|------------|
| Depth-3 | H3 - B | H4 - W | H3 - B | H4 - W |
| | 0.02622 | 0.02116 | 0.009857 | 0.01131 |
| | 0.02326 | 0.02637 | 0.003283 | 0.01016 |
| | 0.02725 | 0.0231 | 0.005303 | 0.009376 |
| | 0.01874 | 0.01712 | 0.01126 | 0.002505 |
| | | | | |
| Average | 0.0238675 | 0.0219375 | 0.00742575 | 0.00833775 |
| Total average | 0.0229025 | | 0.00788175 | |

Fig3. Comparação dos dois algoritmos (Minimax e Minimax com cortes alfa-beta) entre as heurísticas H3 e H4, usando profundidade 3

Anexo 2

| | Minimax | | Alfa-Beta | |
|---------------|----------|--------|-----------|----------|
| Depth-6 | H3 - B | H4 - W | H3 - B | H4 - W |
| | 36.7 | 36.41 | 0.4036 | 0.6464 |
| | 27.5 | 25.91 | 0.6974 | 0.7252 |
| | 19.38 | 19.11 | 0.2277 | 0.6881 |
| | 27.39 | 25.79 | 0.6881 | 0.735 |
| | | | | |
| Average | 27.7425 | 26.805 | 0.5042 | 0.698675 |
| Total average | 27.27375 | | 0.6014375 | |

Fig4. Comparação dos dois algoritmos (Minimax e Minimax com cortes alfa-beta) entre as heurísticas H3 e H4, usando profundidade 6

Anexo 2

| | Minimax | | Alfa-Beta | |
|---------------|-----------|-----------|------------|----------|
| Depth-3 | H5 - B | H6 - W | H5 - B | H6 - W |
| | 0.02 | 0.0278 | 0.01654 | 0.01054 |
| | 0.03487 | 0.02519 | 0.01101 | 0.00464 |
| | 0.01974 | 0.0309 | 0.00385 | 0.00345 |
| | 0.01694 | 0.02148 | 0.00353 | 0.005186 |
| | | | | |
| Average | 0.0228875 | 0.0263425 | 0.0087325 | 0.005954 |
| Total average | 0.024615 | | 0.00734325 | |

Fig5. Comparação dos dois algoritmos (Minimax e Minimax com cortes alfa-beta) entre as heurísticas H5 e H6, usando profundidade 3

Anexo 2

| | Minimax | | Alfa-Beta | |
|---------------|----------|--------|-----------|----------|
| Depth-6 | H5 - B | H6 - W | H5 - B | H6 - W |
| | 36.7 | 36.41 | 0.4036 | 0.6464 |
| | 27.5 | 25.91 | 0.6974 | 0.7252 |
| | 19.38 | 19.11 | 0.2277 | 0.6881 |
| | 27.39 | 25.79 | 0.6881 | 0.735 |
| | | | | |
| Average | 27.7425 | 26.805 | 0.5042 | 0.698675 |
| Total average | 27.27375 | | 0.6014375 | |

Fig6. Comparação dos dois algoritmos (Minimax e Minimax com cortes alfa-beta) entre as heurísticas H5 e H6, usando profundidade 6

Anexo 2

| Clockwise | Minimax | | Alfa-Beta | |
|---------------|-------------|------------|--------------|------------|
| Depth-3 | H5 - B | H6 - W | H5 – B | H6 - W |
| | 0.01859 | 0.02565 | 0.01152 | 0.00735 |
| | 0.03816 | 0.02245 | 0.00456 | 0.00345 |
| | 0.02084 | 0.01818 | 0.01859 | 0.00567 |
| | 0.02474 | 0.01772 | 0.003297 | 0.006989 |
| | 0.01905 | 0.0208 | 0.009292 | 0.010819 |
| | 0.02389 | 0.02727 | 0.005097 | 0.008468 |
| | 0.01444 | 0.01948 | 0.008342 | 0.00868 |
| | 0.02015 | 0.01556 | 0.006689 | 0.00268 |
| | | | | |
| Average | 0.0224825 | 0.02088875 | 0.008423375 | 0.00676325 |
| Total average | 0.021685625 | | 0.0075933125 | |

Fig7. Comparação dos dois algoritmos (Minimax e Minimax com cortes alfa-beta) entre as heurísticas H5 e H6, usando profundidade 3 e sendo que os movimentos possíveis são encontrados usando o sentido do ponteiros do relógio

Anexo 2

| Clockwise | Minimax | | Alfa-Beta | |
|---------------|---------|--------|-----------|----------|
| Depth-6 | H5 - B | H6 - W | H5 - B | H6 - W |
| | 34.94 | 36.88 | 0.3471 | 0.5526 |
| | 25.41 | 26.09 | 0.5816 | 0.5265 |
| | 17.93 | 19.37 | 0.1399 | 0.5372 |
| | 25.32 | 26.04 | 0.5727 | 0.5214 |
| | | | | |
| Average | 25.9 | 27.095 | 0.410325 | 0.534425 |
| Total average | 26.4975 | | 0.472375 | |

Fig8. Comparação dos dois algoritmos (Minimax e Minimax com cortes alfa-beta) entre as heurísticas H5 e H6, usando profundidade 6 e sendo que os movimentos possíveis são encontrados usando o sentido dos ponteiros do relógio