

Trabalho Prático de Inteligência Artificial

Jogo "Ouri"



Universidade de Évora
22 de Junho de 2020

Miguel Azevedo nº 36975 e Vasco Crespo nº 37913

Algoritmos

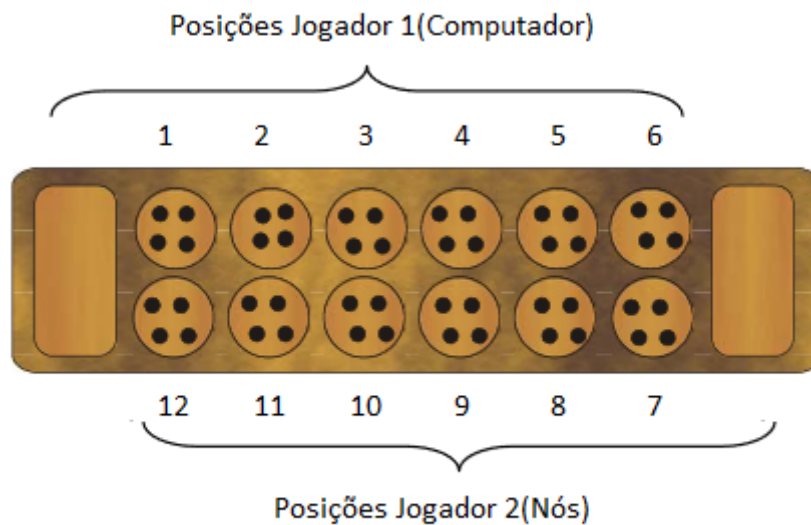
Para a implementação deste jogo decidimos utilizar o algoritmo minimax para a primeira parte do trabalho e o algoritmo minimax com corte alfa-beta para a segunda. Ambos estes algoritmos são muito bons para este jogo de dois jogadores pois não escolhem apenas a melhor jogada possível no momento mas sim tendo em conta todas as possíveis jogadas do adversário, tentando otimizar a sua jogada e minimizar a jogada do adversário.

Primeira parte do trabalho

Para esta primeira parte decidimos utilizar o algoritmo minimax implementado pelo professor e apenas fizemos algumas alterações:

- Para a parametrização de forma a dar resposta em três tempos diferentes utilizámos a profundidade da árvore criado em cada jogada, ou seja, limitámos a quantidade de jogadas que são tidas em conta:
 - Para a primeira parametrização a árvore chegará à profundidade 2, ou seja o algoritmo apenas considerará a melhor jogada tendo em conta a sua própria jogada e uma possível jogada do adversário.
 - Para a segunda parametrização a árvore chegará à profundidade 4 e o algoritmo considerará a melhor jogada tendo em conta 2 jogadas tanto suas como do adversário.
 - Para a terceira parametrização a árvore chegará à profundidade 7 e o algoritmo considerará a melhor jogada tendo em conta 4 jogadas suas e 3 do adversário.
 - Estes valores foram pensados para simularem diferentes dificuldades, sendo a primeira o modo "fácil", a segunda o "intermédio" e a terceira o modo "difícil".
- Alterámos também as variáveis passadas na lista criada nos findalls para serem passadas o estado novo, a posição onde foi efetuada a jogada e ambos os pontos do jogador 1 e 2 para no fim da aplicação do algoritmo nessa jogada seja calculada a melhor jogada tendo em conta os pontos ganhos tanto pelo programa como pelos possíveis pontos ganhos pelo adversário.

Em relação à maneira como tratamos o tabuleiro, nós decidimos tratá-lo como uma lista apenas e que tem a seguinte representação perante o tabuleiro original:



De início tratamos o tabuleiro desta forma mas para a jogada em si primeiro invertemos a lista para podermos percorrer a lista linearmente e ao passar para a fase de verificação das capturas voltamos a inverter a nossa lista(o tabuleiro) para novamente fazermos as verificações que necessitamos linearmente na lista.

O nosso estado inicial é então esta lista com o estado atual, os pontos do primeiro jogador e os pontos do segundo jogador. E o nosso estado terminal(predicado terminal) também tem estes argumentos.

Segunda parte do trabalho

Para a segunda parte do trabalho(a extensão), decidimos implementar o mesmo programa mas com o algoritmo minimax com corte alfa-beta, que tem as mesmas características do algoritmo minimax normal mas é mais rápido e ocupa menos espaço do que este pois retira alguns ramos da árvore que não são necessários.

As alterações à implementação do professor foram novamente apenas para corresponder à nossa implementação do programa e foram idênticas às alterações feitas na primeira parte do trabalho.

O tabuleiro foi tratado exatamente da mesma forma do que na primeira parte do trabalho e as operações e os estados, tanto o inicial como o final, também são idênticos.

Questões

Primeira parte do trabalho

- 1) Ficheiro com o nome "minimax1.pl"
 - 1.a) Modo 1
 - 1.b) Modo 2
 - 1.c) Modo 3
- 2) Ficheiro com o nome "minimax2.pl"
 - 2.a) Modo 1
 - 2.b) Modo 2
 - 2.c) Modo 3

Extensão do trabalho

- 1) Ficheiros com o nome "alfaBeta1.pl" e "alfaBeta2.pl" com os mesmos 3 modos
- 2) Ficheiros como os nomes "novoAlfabeta.pl" e "novoMinimax.pl"
- 3)
 - Modo 1
 - * Minimax começa: Minimax: 7 pontos, Alfabeta: 30 pontos
 - * Alfabeta começa: Minimax: 25 pontos, Alfabeta: 2 pontos
 - Modo 2
 - * Minimax começa: Minimax: 7 pontos, Alfabeta: 26 pontos
 - * Alfabeta começa: Minimax: 4 pontos, Alfabeta: 25 pontos
 - Modo 3
 - * Minimax começa: Minimax: 16 pontos, Alfabeta: 26 pontos
 - * Alfabeta começa: Minimax: 3 pontos, Alfabeta: 26 pontos

Conclusão e escolha do algoritmo para o torneio

Os resultados obtidos justificam-se devido ao número de jogadas que ambos os algoritmos verificam, onde se pode comprovar que quando são vistos poucas jogadas à frente os algoritmos se comportam razoavelmente da mesma maneira e verifica-se que o segundo jogador a começar a jogar ganha. No entanto quando os algoritmos começam a verificar uma profundidade maior nota-se claramente numa discrepância dos resultados, onde se pode ver que o minimax com corte alfa-beta escolhe a próxima jogada não só mais rápido que o minimax, mas que ocupa menos espaço devido aos cortes e que ainda a sua escolha é melhor e mais eficiente.

Por fim, decidimos escolher o programa com o nome **”novoAlfabetapl”** para representar o nosso grupo no “I Torneio de Ouri deIA@LEI@UÉ”.