

Arthur Vinicius Cunha Camargo - 00291335 - turma A
Diego Souza de Mello Affonso - 00278050 - turma B
João Maieron Martins - 00228907 - turma B

Trabalho 1 - Parte 2

Poda alfa-beta em Othello/Reversi

Função de avaliação:

Nosso jogador considera as seguintes **heurísticas** para decidir o próximo movimento:

#1 - Número de moedas: Foca em conquistar peças para si e diminuir o número de peças do adversário.

$(\text{Peças MAX} - \text{Peças MIN}) / (\text{Total de peças})$

#2 - Número de movimentos possíveis: Foca em fazer o movimento que gerará o maior número de movimentos possíveis para si e o menor possível para o adversário.

$(\text{Movimentos MAX} - \text{Movimentos MIN}) / (\text{Total de movimentos possíveis})$

#3 - Número de cantos: Foca em conquistar cantos pois são as posições mais importantes do tabuleiro.

$(\text{Cantos MAX} - \text{Cantos MIN}) / (\text{Total de cantos conquistados})$

#4 - Tabela: Utilizamos a tabela abaixo para dar um valor para cada posição, onde o nosso jogador tentará conquistar as posições de maior valor.

	a	b	c	d	e	f	g	h
1	100	-20	10	5	5	10	-20	100
2	-20	-50	-2	-2	-2	-2	-50	-20
3	10	-2	-1	-1	-1	-1	-2	10
4	5	-2	-1	-1	-1	-1	-2	5
5	5	-2	-1	-1	-1	-1	-2	5
6	10	-2	-1	-1	-1	-1	-2	10
7	-20	-50	-2	-2	-2	-2	-50	-20
8	100	-20	10	5	5	10	-20	100

Nossa heurística terá como base de cálculo o seguinte:

$\#1 * 0,025 + \#2 * 0,05 + \#3 * 1 + \#4 * 0,25$

Estratégia de parada:

O jogador explora a árvore de estados do jogo até uma profundidade fixa, retornando a sua decisão após atingir essa profundidade.

Dificuldades encontradas:

Houve dificuldade em retornar para os estados anteriores (“subir” de volta na árvore) após avaliar os movimentos disponíveis. Isso aconteceu porque nosso primeiro modelo do algoritmo alterava o estado da board quando analisava os passos.

Outra dificuldade encontrada foi com a implementação de um limite de tempo como estratégia de parada, para que o nosso jogador fizesse o melhor movimento encontrado dentro desse limite. Foi difícil estipular um limite bom, e fazer com que retornasse o estado ideal encontrado nesse limite - já que o jogador acabava realizando movimentos aparentemente aleatórios. Por fim desistimos dessa estratégia e optamos por uma profundidade fixa na árvore.

Eventuais melhorias:

A utilização de uma estratégia de parada diferente, que não necessite sempre explorar uma mesma profundidade da árvore.

Bibliografia:

<https://kartikkukreja.wordpress.com/2013/03/30/heuristic-function-for-reversiothello/>
[https://github.com/kartikkukreja/blog-codes/blob/master/src/Heuristic%20Function%20for%20Reversi%20\(Othello\).cpp](https://github.com/kartikkukreja/blog-codes/blob/master/src/Heuristic%20Function%20for%20Reversi%20(Othello).cpp)
<https://github.com/JRChow/Reversi>
<https://github.com/pchampion/othello-prolog/blob/master/ai/heuristic.pl>
https://courses.cs.washington.edu/courses/cse573/04au/Project/mini1/RUSSIA/Final_Paper.pdf
<http://play-othello.appspot.com/files/Othello.pdf>