

1 Relatório Othello

1.1 Grupo:

1. Amanda Braga
2. João Curcio

1.2 Estratégias

Nosso grupo implementou duas estratégias baseadas no algoritmo minimax com corte alfa beta para jogar Othello:

1. O primeiro agente encontrado em `table_weights_player.py` usa uma tabela de pesos para decidir onde quer jogar. O tabuleiro foi dividido em 8 áreas diferentes e cada uma dessas recebeu um peso, como pode ser visto em

`m = { 'H':10 , 'P':-3 , 'I':5 , 'C':3 , 'M':3 , 'N':-1 , 'E':1 , 'A':0 }`

Os pesos foram definidos empiricamente, apenas jogando Reversi no celular e observando onde era bom ou ruim jogar. Quanto maior o peso de uma casa do tabuleiro, mais favorável é jogar ali.

Para calcular a jogada a ser feita, olhamos para os tabuleiros possíveis a partir do ponto que estamos no jogo e até a profundidade de busca que limitamos. O valor calculado pela heurística nas folhas é formado pela soma dos pesos das nossas peças no tabuleiro e a subtração dos pesos das peças adversárias.

2. O segundo agente `minimize_movements_player.py` usa como heurística minimizar o movimento do oponente, ou seja, ele sempre joga numa casa que vai deixar o oponente com o menor número de quadrados possíveis para ele jogar na próxima rodada.

1.3 Torneio

Para o torneio, resolvemos usar a estratégia `table_weights_player.py`. Esse jogador aplica o algoritmo minimax com profundidade 3 o que pode ser visto em `play` no arquivo do `table_weights_player`:

```
def play(self, board):  
    return self.minimax(self.color, board, 3, self.heuristic_a)[1]
```

Como ambas as nossas estratégias usam o minimax com corte alfa beta, implementamos um minimax separado no arquivo `minimax.py` que possui um método homônimo que implementa o minimax com a profundidade 3 passada como parâmetro em:

```
def minimax(self, player, board, depth, heuristic_fn, cut=None)
```

O método acima implementa o minimax com um corte alfa beta de profundidade limitada. O parâmetro `cut` desse método é usado para fazer o corte (que é feito ao final do método nas linhas 51 a 62).

1.4 Referências

1. <http://dhconnelly.com/paip-python/docs/paip/othello.html>
2. Livro do Luger