**Universidade Federal do Rio Grande do Sul** **Inteligência Artificial**
**Instituto de Informática** **Prof. Anderson Rocha Tavares**
**Departamento de Informática Aplicada**

Gabriel Tamusiunas Schumacker: 285648 - Turma B
Luís Dias Ferreira Soares 00287660 - Turma A
Bruna dos santos Gonzaga: 00252743 - Turma A

# Trabalho 1 – Parte 2
# Poda alfa-beta em Othello/Reversi

Nenhuma biblioteca adicional fora as padrões do Python3 são necessárias.

Caso ocorra um erro de permissão ao executar algum script, tente o comando *chmod +x <nome_script>*

Para executar os arquivos .sh a pasta do projeto deve ser acessada chamando o script da seguinte forma:

- *./prepara.sh*
- *./launch.sh*

## Função de avaliação

O algoritmo *minimax* com poda alfa-beta foi usado para determinar qual o movimento ideal, dada a função de avaliação. A Poda alfa-beta Computa o valor Minimax de um estado sem precisar examinar necessariamente todos os estados. Possibilita assim a poda dos ramos improdutivos da árvore de busca.

Em relação a estratégia de parada, a profundidade foi definida como 15. Com base nos testes que fizemos, uma profundidade maior não garante necessariamente um resultado vencedor pois para um oponente aleatório, nunca podemos prever o que ele jogaria na próxima rodada.

Decidimos não usar outras técnicas como profundidade variada pois essa busca até a profundidade 1, então até 2, e assim sucessivamente até que o tempo disponível seja esgotado. Como estamos preocupados em retornar o melhor resultado no menor tempo possível, a profundidade fixa nos pareceu oferecer mais vantagens.

A heurística é bastante simples e uma combinação das heurísticas de pontuação e heurística de captura de cantos e quinas. A ideia principal de unir as duas é maximizar suas próprias posições valiosas (capturando os cantos) enquanto minimiza as posições valiosas do oponente. A heurística também pontua negativamente as posições possíveis para a jogada seguinte, de forma que reduzir as possibilidades do adversário conta a favor da heurística.

Utilizar somente a heurística de pontos não é bom, pois mesmo que uma jogada gere grande quantidade de pontos o oponente pode reverter o jogo. Porém quando combinada com a heurística de captura de cantos, cada quina têm uma importância especial porque, uma vez capturados, não podem ser ocupados pelo oponente. Eles também permitem que um jogador capture *moedas* ao seu redor e forneçam estabilidade a elas. Se você possui dois cantos adjacentes, provavelmente terá toda a conexão.

Os quadrados dos cantos valem mais porque são os quadrados mais importantes do jogo. Os quadrados imediatamente adjacentes (incluindo diagonalmente) aos quadrados dos cantos valem pontos negativos porque ajudam seu oponente a pegar os quadrados das quinas. Esse decréscimo deixa de existir, quando a quina é conquistada.
Os quadrados a dois espaços dos cantos valem uma boa quantidade de pontos porque ajudam a obter os quadrados dos cantos.
Para o resto do tabuleiro, geralmente, os quadrados próximos ao centro valem alguns pontos e os que estão um pouco mais distantes valem alguns pontos negativos. Isso é baseado na estratégia básica de Othello.

A seguir temos a tabela de pontos conforme cada quadrado do tabuleiro.

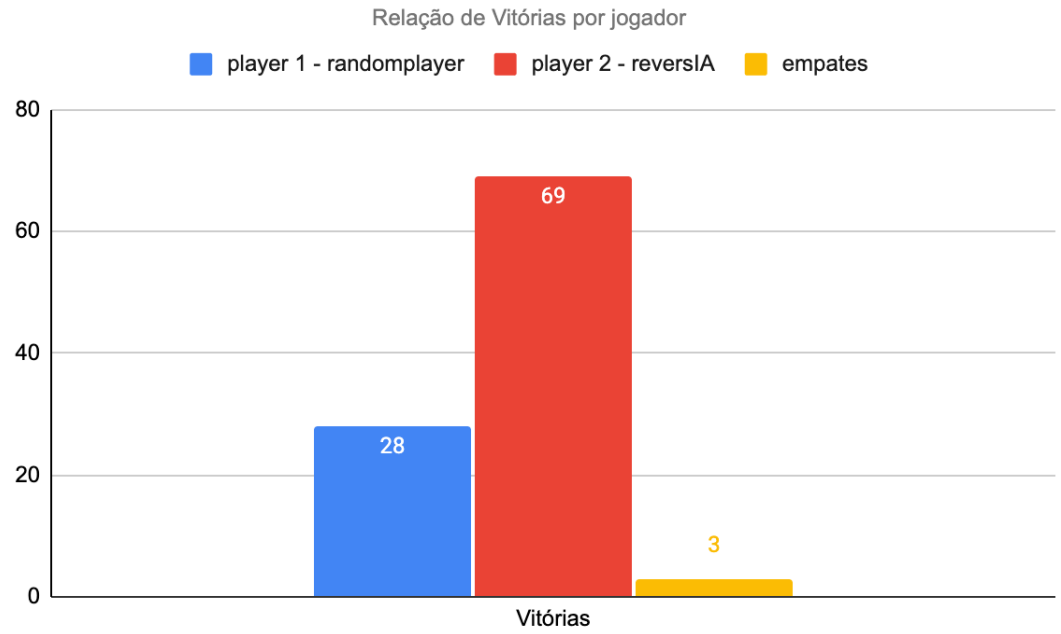
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8\n |
|---|---|---|---|---|---|---|---|---|
| 1 | 150 | -30 | 10 | 10 | 10 | 10 | -30 | 150 |
| 2 | -30 | -30 | | -5 | -5 | | -30 | -30 |
| 3 | 10 | | | | | | | 10 |
| 4 | 10 | -5 | | | | | -5 | 10 |
| 5 | 10 | -5 | | | | | -5 | 10 |
| 6 | 10 | | | | | | | 10 |
| 7 | -30 | -30 | | -5 | -5 | | -30 | -30 |
| 8 | 150 | -30 | 10 | 10 | 10 | 10 | -30 | 150 |

O número de peças aliadas (peças que você já capturou) e reduzir as opções de jogada do seu oponente também entra para o somatório de pontos.

## Performance

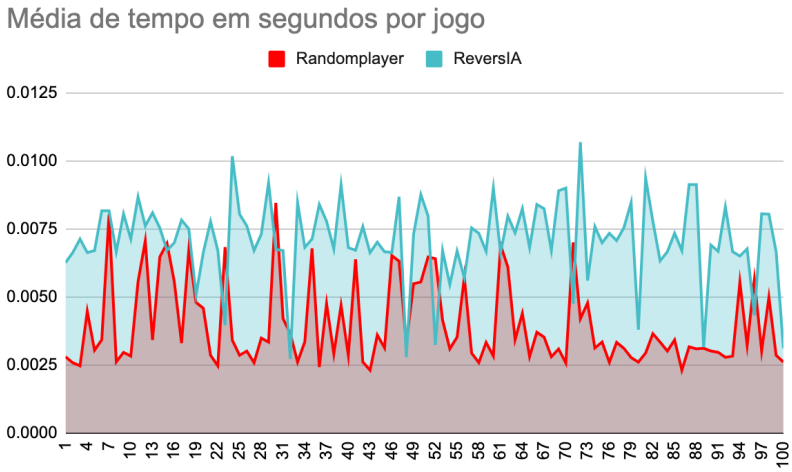
Para fins de testes foi criado um script que simula inúmeras partidas entre a IA criada pelo grupo *reversIA* e a IA disponibilizada pelo professor *randomplayer*.
Foram simuladas 100 partidas completas e os resultados seguem a seguir.

**Vitórias por jogador:** Podemos perceber que a IA que apresentava as heurísticas teve um desempenho muito melhor que o jogador random.




**Movimentos Ilegais:** Como pudemos perceber, o jogador 2 apresentou um numero insignificante de jogas ilegais enquanto o jogador random tentou fazer pelo menos 1 jogada ilegal por jogo.


E quando analisamos a média de tempo por jogo, podemos perceber que na maior parte dos jogo o randomplayer tomava decisões muito mais rápido.




## Conclusão

Após a implementação e testes conseguimos observar que ao inserir mais heurísticas aumentamos as chances de vitória. Observamos também que aumentarmos a profundidade de busca da árvore nem sempre garante um desempenho melhor.

Em relação às dificuldades, o primeiro desafio foi entender como o jogo funcionava. Após entendermos a dinâmica dos cantos e que garantindo os cantos aumentamos as nossas chances de vitória conseguimos pensar nas heurísticas necessárias para a implementação. Encontrar uma profundidade adequada e definir uma tabela de pontos também foi desafiador. Iniciamos chutando alguns valores na profundidade e depois foi preciso verificar como isso refletia no desempenho do programa.

No geral, foi interessante observar as melhorias conforme íamos modificando e testando o programa.

## Bibliografia


1. Parekh, Parth, et al. “Othello/Reversi using Game Theory techniques.” - http://play-othello.appspot.com/files/Othello.pdf.
2. Meng Tran . “Reversi” - https://www.seas.upenn.edu/~cse400/CSE400_2006_2007/MengTran/tranm_senior_project.pdf. Paul G. Allen Center,
3. Department of Computer Science and Engineering, University of Washington, and Muthukaruppan Annamalai. “An Analysis of Heuristics in Othello.” - [Seattle] https://courses.cs.washington.edu/courses/cse573/04au/Project/mini1/RUSSIA/Final_Paper.pdf.
4. "Minimax with Alpha-Beta Pruning in Python" - https://stackabuse.com/minimax-and-alpha-beta-pruning-in-python/
5. "Heuristic/Evaluation Function for Reversi/Othello"- https://kartikkukreja.wordpress.com/2013/03/30/heuristic-function-for-reversiothello/