

## Portfólio 2 - Projeto de busca complexa

### 1 Introdução

Este documento tem como foco apresentar o problema a ser resolvido, a implementação e os resultados de um algoritmo de Busca em um ambiente complexo, desenvolvido na linguagem Python, para a disciplina de Inteligência Artificial.

Foram utilizados a IDE pycharm para implementar o algoritmo de poda alfa-beta.

### 2 Problema

O Jogo da Velha é determinístico e jogado em um tabuleiro 3x3. Dois jogadores, 'X' e 'O', alternam entre si para marcar células vazias, com o objetivo de formar uma linha com 3 marcações seguidas coluna ou diagonal. O jogo empata se o tabuleiro for preenchido sem um vencedor.

O que o torna um ambiente de estudo perfeito é sua natureza adversarial cada jogada deve antecipar a melhor resposta possível do oponente. Isso o qualifica como o problema fundamental para implementar e testar a eficiência de algoritmos de busca complexos..

### 3 Algoritmo desenvolvido de Busca Complexa: Poda Alfa-Beta

A Poda Alfa-Beta é uma otimização do Minimax que reduz drasticamente o número de nós que precisam ser explorados na árvore. Ela alcança o mesmo resultado exato do Minimax, mas de forma muito mais rápida, "podando" ramos inteiros da árvore que ela determina serem irrelevantes.

#### 3.1 Exemplo de uso

A função que busca o melhor movimento testa todas as jogadas possíveis no tabuleiro, simula cada uma e usa a função minimax para prever o resultado final, considerando que o oponente jogará perfeitamente. Após avaliar todas as opções, escolhe a jogada com a melhor pontuação.

A função minimax explora recursivamente todas as possibilidades de jogo buscando a maior pontuação na jogada que irá fazer, já no turno do humano, busca a menor. Se o jogo terminou, retorna a pontuação final (10, -10 ou 0).

A poda alfa-beta otimiza esse processo mantendo dois limites alpha (melhor resultado da máquina) e beta (pior resultado do humano) quando é detectado um caminho que não pode gerar um resultado melhor do que outro já encontrado, interrompe sua análise, eliminando simulações desnecessárias e acelerando o cálculo da jogada ideal.

```

0 0|X|
-----
1 |X|
-----
2 X|0|0

Digite a linha (0, 1, ou 2): 0
Digite a coluna (0, 1, ou 2): 2

    0 1 2
0 0|X|0
-----
1 |X|
-----
2 X|0|0

IA ('X') está pensando...
IA jogou na posição (1, 2)

    0 1 2
0 0|X|0
-----
1 |X|X
-----
2 X|0|0

Digite a linha (0, 1, ou 2): 1
Digite a coluna (0, 1, ou 2): 0

    0 1 2
0 0|X|0
-----
1 0|X|X
-----
2 X|0|0

0 jogo terminou em EMPATE!

```

Figura 1: Execução da Poda Alfa-Beta