
PROJETO 3

Maria Eduarda de Jesus Padovan
cc22143@g.unicamp.br

Vitor Silveira de Lucena
cc22154@g.unicamp.br

1 Introdução

O projeto busca aprofundar os conhecimentos sobre a resolução de problemas de decisão em jogos de tabuleiro, com foco no jogo da velha (*Tic-Tac-Toe*), ao implementar uma solução onde o computador joga contra um ser humano. O desafio é garantir que o computador sempre tome a melhor decisão possível, visando vencer ou empatar o jogo, independentemente das jogadas do oponente. Para isso, utiliza-se o algoritmo *Minimax*, que explora todas as possíveis jogadas futuras e seus resultados, atribuindo valores a cada uma e escolhendo a jogada que minimiza as perdas e maximiza os ganhos. [1]

2 Algoritmo Minimax

Para compreender o algoritmo Minimax, é fundamental primeiro entender o funcionamento básico do jogo da velha, já que esse conhecimento é essencial para programar o computador para jogar de forma estratégica, como se fosse um oponente humano real. No jogo da velha, há três possíveis resultados:

vitória para o jogador 'X', vitória para o jogador 'O' ou empate, quando nenhum dos dois vence.

O princípio do algoritmo Minimax é que, ao jogar como 'X', o jogador tentará **maximizar** suas chances de vitória, analisando as possíveis jogadas futuras e antecipando as ações do oponente. Por outro lado, o jogador adversário, ao jogar como 'O', tentará **minimizar** as chances de vitória de 'X'. Assim, o Minimax simula essa interação, com cada jogador buscando a melhor estratégia para maximizar suas próprias chances de sucesso enquanto minimiza as do oponente. [3]

No exemplo abaixo, 'O' escolheria qualquer uma das opções que resultam em -10 (1ª e 3ª opções), pois isso minimizaria o ganho do 'X':

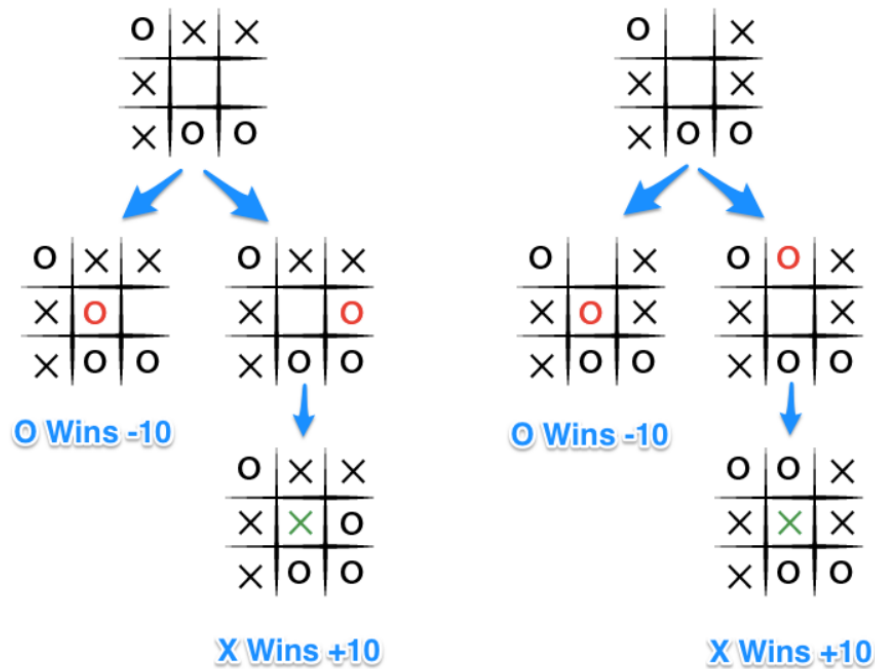


Figura 1: Exemplificação do Algoritmo Minimax [2]

O algoritmo Minimax pode ser explicado de maneira simples como um jogo de "imaginar o que o outro vai fazer" e escolher a melhor jogada, seguindo sempre: o jogador 'X' escolhe o movimento que lhe dá a maior pontuação. O jogador 'O' escolhe o movimento que dá a menor pontuação para 'X'.

3 Experimentos computacionais

Todos os experimentos computacionais foram realizados em um computador macOS Sonoma 14.4.1, com 16GB de RAM, desenvolvido durante os horários vagos e/ou durante a aula de Inteligência Artificial do curso de Informática. O código foi implementado em Python 3.12.4 e resolvido no Visual Studio Code 2024.

3.1 Instâncias de testes

O algoritmo Minimax foi testado a partir do código disponibilizado pelo professor no ambiente de ensino aberto, o qual foi complementado pelos alunos. A implementação foi avaliada com base em diferentes cenários de jogo, garantindo que o computador pudesse maximizar suas chances de vitória ou, no mínimo, garantir um empate, independentemente das jogadas do oponente humano.

3.2 Resultados

Os resultados obtidos confirmaram a teoria inicial: um computador que aplica o algoritmo de minimax sempre irá, no mínimo, empatar com o o jogador, se não ganhar. Pois ele sempre joga da melhor maneira possível e assume que o

oponente fará o mesmo, sendo impossível ocorrer uma derrota para seu lado por conta disso.

Quando o jogador inicia o jogo, ele pode escolher jogar como 'X' ou como 'O' - importante lembrar que, no caso desse jogo da velha, o 'X' sempre será o primeiro a jogar, portanto estará sempre maximizando, enquanto o 'O' estará minimizando. Caso decida jogar como 'X', será o primeiro a jogar no tabuleiro, caso jogue como 'O', o computador é quem vai começar.



Figura 2: Exemplo de início do jogo

Uma observação, é que nem sempre o algoritmo ganhava da forma mais eficiente possível, em sua primeira implementação ele "sempre fazia a melhor jogada pensando no futuro". Para explicar melhor será dado um exemplo:

Nesse exemplo, o jogador está controlando o "O". E caso ele joga na casa (3,1).

Em vez do Minimax ganhar o jogo jogando na casa (1,1). Ele faz sua

jogada na casa (3,2).

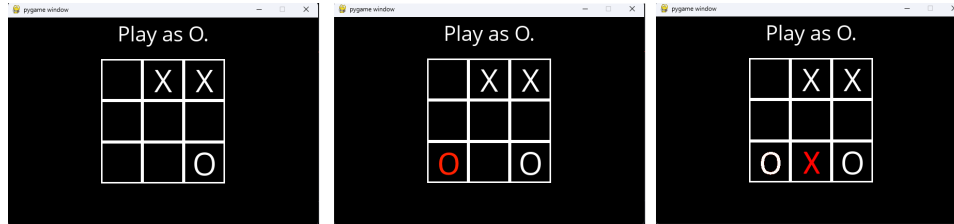


Figura 3: Exemplificação da situação descrita

Isso acontece, pois o algoritmo estava ignorando a possibilidade de ganhar imediatamente por uma situação melhor no futuro. Pois com o tabuleiro da maneira acima, o Minimax pode ganhar de duas maneiras, seja na casa (1,1) ou na casa (2,2), enquanto que, a ação do jogador não consegue parar as duas ao mesmo tempo.

Isso é algo que decidimos mudar diretamente, e agora o algoritmo checa se existe uma jogada que dá uma vitória nesse mesmo movimento, e sempre a prioriza.

Referências

- [1] DA SILVA, L., LYRA, R., CORRÊA, V. J., AND HENRIQUE, A. S. Minimax no jogo da velha. *Anais do Computer on the Beach 14* (2023), 539–543. Accessed: 2024-10-16.
- [2] FOX, J. Tic tac toe: Understanding the minimax algorithm, 2013. Accessed: 2024-10-16.
- [3] YOUTUBE. Understanding the minimax algorithm — a visual explanation, 2023. Accessed: 2024-10-16.