



UNIVERSIDADE NOVE DE JULHO - UNINOVE
PROJETO EM SISTEMAS INTELIGENTES (VIDE CLASSROOM)

EXCLUÍDOS OS DADOS SOBRE OS AUTORES EM ATENDIMENTO A
LGPD - LEI GERAL DE PROTEÇÃO DE DADOS

A Aplicação Do método Minimax de Simulação Monte Carlo em jogo da
velha

São Paulo
2024

**EXCLUÍDOS OS DADOS SOBRE OS AUTORES EM ATENDIMENTO A
LGPD - LEI GERAL DE PROTEÇÃO DE DADOS**

**A Aplicação Do método Minimax de Simulação Monte Carlo em jogo da
velha**

Projeto apresentado a Universidade Nove de
Julho - UNINOVE, como parte dos requisi-
tos obrigatórios para obtenção do título de
Ciência da Computação.

Prof. Orientador: Edson Melo de Souza, Dr.

**São Paulo
2024**

RESUMO

Contexto: Veremos como o Jogo da velha pode ser operado por uma IA treinada por algoritmo para vencer consecutivamente. **Objetivo:** Neste documento exploraremos a utilização de um algoritmo de muito usado em Teoria dos jogos chamado de Minimax e sua análise dentro de um Simulação monte carlo. **Método:** Operando apenas com a Lógica de GameTree e Minimax o algoritmo percorre sua árvore de decisões, onde cada jogada é analisada e salva em nós na qual ele verá +1 para jogadas que trarão vitória, -1 para jogada que ocasionam em derrotas e 0 para jogadas das quais ocorrem empates decidindo então qual próxima jogada será a melhor para um resultado otimizado. **Resultados:** Vemos que um algortimo bem treinado e que recursivamente observa suas próprias jogadas e as de seu oponente se sairá melhor do que aquele que opera apenas por uma lógica aleatória. **Conclusão:** Por fim dentro de sua capacidade de 1000 jogos analisados a Simulação de Monte carlo trará a comparação de performance entre os dois jogadores e destacará a vitória daquele que foi treinado pelo minimax e subsequentemente demonstrará em um gráfico tal diferença.

Palavras-chave: Minimax, GameTree, Simulação, Jogo, Algoritmo, Lógica.

ABSTRACT

Contextualization: We'll observe that the game of Tic-tac-toe can be operated by an Algorithm-based-AI to win consecutively. **Objective:** In this document we shall explore the utilization of the algorithmic game theory called Minimax and its analysis inside of a Monte Carlo Simulation. **Method:** Operating only through GameTree logic and Minimax the algorithm runs through its decision tree, where each play is analyzed and saved in nodes which will check +1 for plays that occurs in victory -1 for plays that occur in defeat and 0 for the plays that occur on ties then deciding so what next play shall be the best for an optimal result. **Results:** We see that on a well trained algorithm that recursively observes its own plays and those of his opponent's shall have the better outcome than that which operates on random logic. **Conclusion:** Lastly inside its capacity for 1000 analyzed games the Monte Carlo Simulation shall compare the performance between the two players and highlight the victory of that which was trained by the MM and subsequently display the difference in a plotted graphic.

Keywords: Minimax, GameTree, Simulation, Game, Algorithm, Logic.

SUMÁRIO

Lista de Ilustrações	6
Lista de Abreviaturas	7
1 Introdução	8
1.1 Inclusão de Figura	8
2 Fundamentação Teórica	10
2.1 Visão Geral	10
2.2 Composição	10
3 Metodologia	11
3.1 Visão Geral	11
3.2 Analisando MinMax	11
4 Análise dos Resultados	12
5 Conclusões	13
Referências Bibliográficas	14

LISTA DE ILUSTRAÇÕES

1.1	Vistos em resultados.	9
1.2	Vistos em resultados.	9

LISTA DE ABREVIATURAS

MM	Minimax
SMC	Simulação de Monte Carlo
J1	Jogador 1

1 INTRODUÇÃO

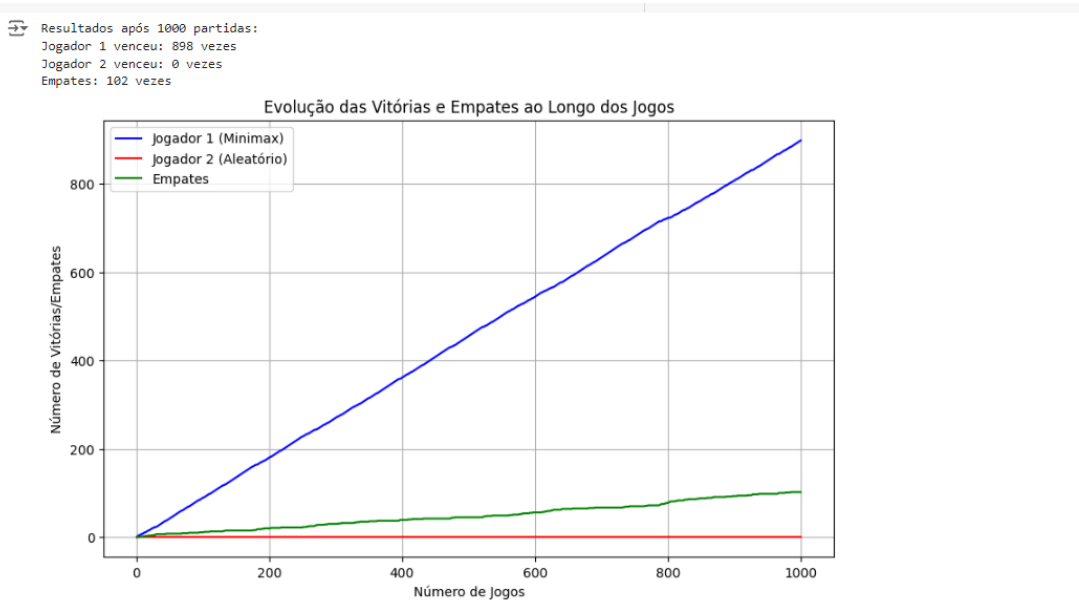
Resumo do capítulo

O Jogo da velha (Tic-Tac-Toe) é um jogo clássico que pode ser resolvido por meio de algoritmos de inteligência artificial, como o Minimax. No livro Artificial Intelligence: A Modern Approach, (NORVIG, 2019). "o algoritmo Minimax é uma técnica fundamental utilizada na teoria dos jogos para minimizar a perda máxima" Este trabalho busca apresentar a implementação do algoritmo Minimax para um dos jogadores e a execução de uma simulação de Monte Carlo para analisar o desempenho do algoritmo em diversas partidas contra um jogador que realiza movimentos aleatórios. Essa documentação aborda a metodologia utilizada, a implementação do algoritmo, a simulação e a análise dos resultados.

1.1 INCLUSÃO DE FIGURA

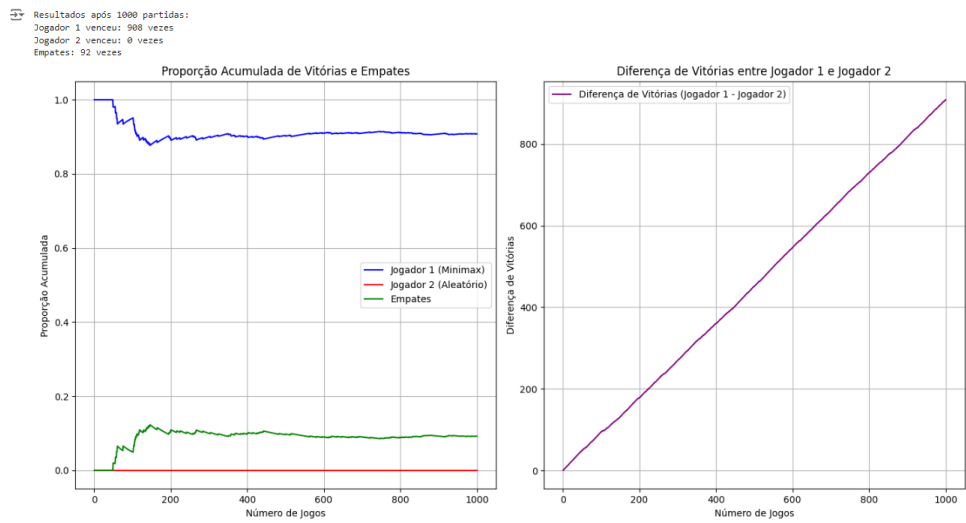
As Figuras 1.1 1.2, demonstram os resultados obtidos pelo algoritmo em forma de gráficos.

Figura 1.1 – *Vistos em resultados.*



Fonte: Google Colab

Figura 1.2 – *Vistos em resultados.*



Fonte: Google Colab

2 FUNDAMENTAÇÃO TEÓRICA

Resumo do capítulo

Aqui vai um pequeno resumo do capítulo.

2.1 VISÃO GERAL

O objetivo é demonstrar o algoritmo Minimax que influencia os movimentos da máquina ao jogar jogo da velha e analisa seu desempenho utilizando uma simulação de Monte Carlo. O jogador 1 (CPU) realiza seus movimentos baseado no algoritmo e seu adversário (também CPU) realiza movimentos aleatórios indicativos de uma jogada despreparada e impensada.

2.2 COMPOSIÇÃO

1. Algoritmo Game Tree. Dentro da lógica de jogos como Xadrez, Dama e o abordado Jogo da Velha, visualizamos que há uma árvore de escolhas a ser entendida pelo código com cada nó sendo uma jogada e em cada linha uma rodada de cada turno. Segundo [Becker \(2019\)](#), "No caso do Jogo da Velha, é possível processar computacionalmente a sua respectiva Game Tree, pois o número de 'nós' é igual a 255.168. Um jogo de Xadrez por exemplo, possui mais possibilidades de jogadas do que átomos no nosso universo observável!".
2. O Algoritmo Minimax Minimax, muito usado no campo de Teoria de Jogos para desenvolver a inteligência artificial de jogos competitivos, principalmente os chamados jogos de soma zero, onde se um jogador vence, o outro necessariamente perde.

Lacuna de Pesquisa 1. Há uma conversa entre o Game Tree e a resolução do Minimax onde o CPU consulta pela árvore dos nós da Game Tree e consulta se há a possibilidade de vencer em cada jogada e com ela, consegue identificar o melhor caminho a se prosseguir para vitória sem que seu oponente interfira. Os caminhos de diversos cenários são abordados pelo Minimax seu princípio é partir analisando desde a primeira jogada e descer até o Terminal (o resultado) identificando se o jogador irá perder, empatar ou vencer. Depois dessa análise o algoritmo sobe o nó identificando se é seu turno de jogar caso não seja ele guarda um valor menor de suas ramificações $\text{Min}()$, caso seja seu turno ele guarda o maior resultado de suas ramificações $\text{Max}()$.

3 METODOLOGIA

Resumo do capítulo

Aqui vai um pequeno resumo do capítulo.

3.1 VISÃO GERAL

Como a simulação de Monte Carlo se encaixa nisso tudo? Método Monte Carlo ou **Simulação Monte Carlo** é um algoritmo computacional de cálculo probabilístico que prevê um conjunto de resultados baseados nos valores que recebeu de entrada. O SMC então, recalcula recursivamente ajustando os números entre os mínimos e os máximos.

3.2 ANALISANDO MINMAX

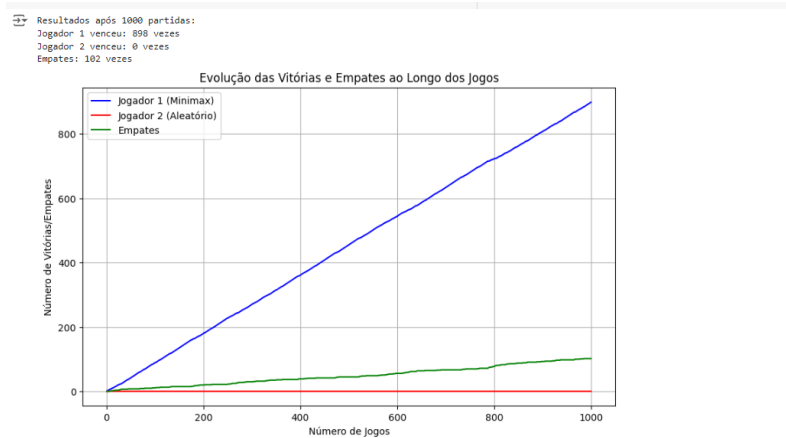
Vejamos como ele se comporta dentro dessa análise

1. Avaliação de Eficácia em múltiplas partidas
2. Performance Estatística

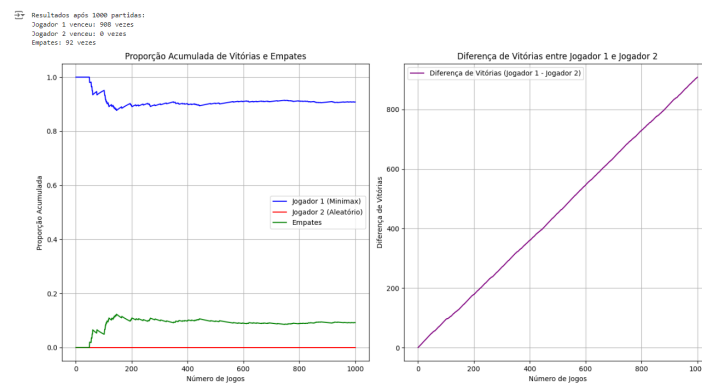
Lacuna de Pesquisa 2. Realiza-se em um código gerado por ChatGPT e Google Colab o corpo de testes para essas simulações. Onde operamos 1000 jogos da velha com método SMC, para refinar e destacar, o dado resultado de cada partida que é gravada ao exibir as tentativas dos dois oponentes, um com sua estratégia aleatória o outro operando com nosso tão utilizado **algoritmo MinMax**, em que o j1 (CPU) obtém suas jogadas através de sua análise recursiva de sua própria Game Tree observando seus erros, e desafiando suas convenções para cada jogada ser a melhor possível, seus movimentos e os de seu adversário, são registrados nas variáveis que irão para a comparação e validação ao final de cada partida, nessa avaliação podemos perfilar a melhora ou piora da estratégia e calcular estatisticamente seu resultado.

4 ANÁLISE DOS RESULTADOS

Teremos então um algoritmo que operará com máximas vitórias de todas as partidas e mínimos empates, sem nenhuma derrota sequer. Essa abordagem fará com que toda jogada realizada pelo nosso J1, seja recursivamente estratégica comparada com seu último movimento, e o seu oponente que será incapaz de acompanhar seu nível de raciocínio já que este não opera sobre nenhuma estratégia, e sim está apenas capacitado a jogar em um quadrado aleatório que não lhe trará vitória alguma, tendo apenas uma pequena margem de empates sobre seu rival, por pura sorte já que ele ainda estaria aprendendo a melhorar suas intenções de jogada para lhe trazer apenas vitórias. Sucessivamente dentro destes mil jogos se dará o resultado com Jogador 1 sendo predominantemente vitorioso com **898 Vitórias e apenas 102 empates**, aprendendo desde sua primeira intenção até a última refinada abordagem para o tabuleiro do jogo. E então teremos todos nossos resultados unidos pelo Sistema de Monte Carlo que analisará cada MinMax dessa GameTree e os finais de cada partida guardando os resultados em suas variáveis e validando estatisticamente o desempenho dessa "IA". A Imagem a seguir compara resultados e performance do código rodado pelo Google COlab.



(a) Evolução de vitórias empates e derrotas



(b) Proporção de vitórias acumuladas

5 CONCLUSÕES

Concluindo. Ao estudarmos a performance de nossa IA notamos que instintivamente há um viés crescente este sendo o de jogar no meio do tabuleiro assim que possível tomando dominância de 3 eixos ao mesmo tempo, e possibilita os melhores resultados essa abordagem quando é efetuada ao jogador 1 começar primeiro, o resultado para o Jogador 2 já não é muito promissor já que este só está operando em escolhas aleatórias.

Não há diferenças quanto a escolha de X ou O ambas trazem os mesmos resultados.

O Algoritmo MinMax melhora significativamente o desempenho do Jogador 1, e opera em sua melhor performance após cerca de 100 jogos onde ele recursivamente está operando na maior GameTree e consultando diversas jogadas já efetuadas por ele e seu oponente é aí que ele começa a ganhar sucessivamente e mantém-se assim até o fim da simulação dos 1000 jogos,

O Método Monte Carlo opera sua capacidade de 1000 jogos consultando as jogadas de MinMax do Jogador 1 e comparando com as aleatórias do Jogador 2, obviamente que as da IA estão muito mais performáticas de que seu oponente então as comparações tendem a ser em favor do Jogador 1, no final da simulação são exibidos resultados e a amostragem é então convertida em Gráficos para melhor visualização sendo estes de proporções acumuladas e um mais linear que demonstra apenas a diferenças de vitórias.

REFERÊNCIAS BIBLIOGRÁFICAS

BECKER, L. **Algoritmo Minimax - Introdução à Inteligência Artificial.** *Organica Digital*, 2019. doi:. Citado na pág. 10.

NORVIG, R. ***Artificial Intelligence: A Modern Approach***. Prentice Hall, 2019. Disponível em: <<https://dl.ebooksworld.ir/books/Artificial.Intelligence.A.Modern.Approach.4th.Edition.Peter.Norvig.%20Stuart.Russell.Pearson.9780134610993.EBooksWorld.ir.pdf>>. Citado na pág. 8.