



UNIVERSIDADE NOVE DE JULHO - UNINOVE
Projeto Em Sistemas Inteligente

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

Jogo da velha em sistema inteligente

São Paulo
2024

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

JOGO DA VELHA COM IA

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

Prof. Orientador: Edson Melo de Souza, Dr.

São Paulo
2024

SUMÁRIO

- **Introdução;**
- **Contexto e motivação;**
- **Objetivo do projeto;**
- **Visão geral do jogo;**
- **Regras do jogo;**
- **Objetivos do jogo;**
- **Metodologia**
- **Fundamentação teórica;**
- **Tecnologias utilizadas;**
- **Algoritmos;**
- **Lógica de implementação;**
- **Código;**
- **Elementos visuais;**
- **Testes;**
- **Resultados;**
- **Referências;**
- **Referências bibliográficas**
- **Conclusão;**

INTRODUÇÃO

O jogo da velha, também conhecido como “Tic-tac-toe” é um jogo simples atemporal, tradicionalmente jogado utilizando papel e caneta, agora ganha vida com aplicação de técnicas de inteligência artificial.

CONTEXTO E MOTIVAÇÃO.

A simplicidade e clareza do jogo da velha o tornam ideal para explorar algoritmos e estratégias de tomadas de decisão, com o avanço da inteligência artificial, é possível criar um ambiente interativo e desafiante, até mesmo para os jogadores mais experientes.

Este projeto nasceu baseando-se em nosso desejo de juntar nostalgia e tecnologia, nosso objetivo é proporcionar uma experiência inovadora, desafiadora e divertida deste jogo tão icônico. Em um contexto simples, entretanto elaborado, convidamos a todos para interagir com a nossa IA.

OBJETIVOS DO PROJETO

- Desenvolver um jogo da velha onde o jogador tenha como oponente a nossa IA.
- Implementar algoritmos de IA que permitam ao computador tomar decisões estratégicas durante o jogo.
- Oferecer uma interface intuitiva e amigável para uma experiência de jogo imersiva.
- Explorar e demonstrar as possibilidades e limitações da IA em um ambiente de jogo clássico.

VISÃO GERAL DO JOGO

O jogo da velha é um jogo de estratégia simples e clássico jogado em uma grade 3x3. O jogo é disputado por 2 jogadores, que alternam entre marcar espaços vazios na grade com seus símbolos designados, geralmente “X” ou “O”. O objetivo baseia-se em formar uma linha vertical, diagonal ou horizontal com três símbolos idênticos, enquanto se previne para que o adversário não o faça. O jogo termina quando um dos jogadores atingir essa condição ou quando todos os espaçamentos forem preenchidos, resultando em um empate.

REGRAS DO JOGO

1. Os jogadores alternam entre fazer movimentos, começando com o primeiro jogador.
2. Cada jogador pode colocar seu símbolo em qualquer espaço vazio na grade.
3. O jogo continua até que um jogador vença ao formar uma linha horizontal, vertical ou diagonal, com três dos seus símbolos, ou até que todos os espaços da grade sejam preenchidos, resultando em um empate.
4. Um jogador não pode jogar em um espaço já preenchido por um símbolo.

OBJETIVOS DO JOGO

- Para o jogador humano: Vencer o jogo ao formar uma linha com três símbolos, ou evitar perder até que o jogo termine em empate.
- Para a inteligência artificial: Vencer o jogo ao prever e bloquear os movimentos do jogador humano, ou garantir que o jogo termine em empate, caso a vitória não for possível.

METODOLOGIA

1. Projeto do Jogo da Velha

- Definição de Requisitos: Identificação dos requisitos funcionais e não funcionais do jogo, incluindo regras do jogo, interface do usuário e lógica de jogo.
- Prototipagem da Interface: Criação de protótipos da interface do jogo para planejar a disposição dos elementos visuais e interações do usuário.

2. Implementação da Lógica do Jogo

- Representação do Tabuleiro: Desenvolvimento da estrutura de dados para representar o tabuleiro do jogo e controlar o estado das jogadas.
- Implementação das Regras do Jogo: Codificação das regras do jogo da velha, incluindo verificação de vitória, detecção de empate e validação de movimentos.

3. Desenvolvimento da Inteligência Artificial

- Escolha do Algoritmo: Seleção do algoritmo de busca e tomada de decisão, como o algoritmo minimax com poda alfa-beta para a IA.
- Implementação da IA: Desenvolvimento do código que permite à IA fazer jogadas estratégicas com base na análise do tabuleiro.

4. Integração e Testes

- Integração de Componentes: Integração da lógica do jogo, interface do usuário e IA para criar uma aplicação funcional.
- Desenvolvimento de Testes: Criação de testes unitários, de integração e de performance para validar o funcionamento correto do jogo e da IA.

5. Melhorias e Otimizações

- Otimização da IA: Ajustes no algoritmo e na heurística utilizados pela IA para melhorar a eficiência e precisão das jogadas.
- Refatoração de Código: Revisão e aprimoramento do código-fonte para garantir clareza, manutenibilidade e desempenho adequado.

6. Documentação e Publicação

- Documentação Completa: Elaboração de documentação abrangente que descreve o funcionamento do jogo, a lógica da IA, os testes realizados e os

resultados obtidos.

- Publicação do Jogo: Disponibilização do jogo da velha com IA para uso público ou para fins de demonstração.

FUNDAMENTAÇÃO TEÓRICA

Estrutura HTML: Criamos a estrutura básica do jogo da velha em HTML. Utilizamos uma tabela `<table>` para representar o tabuleiro do jogo, com células `<td>` para cada posição no tabuleiro. Adicionamos eventos de clique em cada célula para permitir que os jogadores façam suas jogadas.

Estilização com CSS: Utilizamos CSS para estilizar o tabuleiro do jogo da velha e torná-lo visualmente atraente. Definimos estilos para as células do tabuleiro, incluindo bordas, cores e tamanho. Também adicionamos estilos para destacar as jogadas dos jogadores e a célula ativa.

Interação com JavaScript: Implementamos a interatividade do jogo usando JavaScript. Desenvolvemos a lógica para alternar entre os jogadores (X e O), verificar jogadas válidas, verificar condições de vitória ou empate e atualizar dinamicamente a interface do usuário com as jogadas feitas.

Lógica do Jogo e IA: Implementamos a lógica do jogo da velha em JavaScript, incluindo a detecção de vitória, empate e a interação com a inteligência artificial. Para a IA, criamos o algoritmo minimax para tomar decisões estratégicas de jogo e fazer jogadas contra o jogador humano.

Comunicação com PHP: Para funcionalidades como salvar o histórico de jogos, registrar pontuações ou realizar jogadas da IA no servidor, utilizamos PHP para lidar com essas operações de backend. Por exemplo, implementamos o armazenamento e recuperação de dados do jogo em um banco de dados.

Responsividade e Usabilidade: Garantimos que o jogo seja responsivo, adaptando-se a diferentes tamanhos de tela e dispositivos. Testamos a usabilidade da interface para garantir que seja intuitiva e fácil de usar, tanto para jogadores

humanos quanto para a IA.

TECNOLOGIAS UTILIZADAS

O projeto foi desenvolvido em HTML e JAVASCRIPT, e PHP.

ALGORITMOS

Foi utilizado o Minimax, como método de tomada de decisão.

LÓGICA DE IMPLEMENTAÇÃO

- A implementação do algoritmo Minimax no jogo da velha requer a representação do tabuleiro como uma estrutura de dados, geralmente uma matriz.
- O algoritmo é implementado recursivamente, explorando todas as jogadas possíveis até atingir uma condição de parada, como um estado de jogo terminal (vitória, derrota ou empate) ou uma profundidade máxima alcançada.
- Para cada nó do diagrama (estado do jogo), são calculados os valores minimax usando a lógica do algoritmo Minimax.
- A melhor jogada é determinada selecionando a jogada com o maior valor minimax para o jogador ou o menor valor minimax para o oponente.

CÓDIGOS

Estrutura HTML

Desenvolvemos a estrutura HTML do jogo da velha com inteligência artificial utilizando marcações padrão como `<html>`, `<head>` e `<body>`. Incorporamos referências a arquivos externos de CSS e PHP para garantir uma organização modular e aprimorar a manutenção do código.

Inclusão de PHP

Para modularizar o código e promover a reutilização de componentes, incluímos arquivos PHP com `include("cabecalho.php")` e `include("rodape.php")`. Isso contribui para uma organização eficiente e a manutenção de um código limpo e legível.

Elementos e Estilização

O jogo apresenta elementos como o título "JOGO DA VELHA" e o tabuleiro representado por divs identificadas por IDs específicos. Utilizamos um arquivo externo de CSS para estilizar os elementos, garantindo uma interface visual atraente e intuitiva para os jogadores.

Interação com JavaScript

Implementamos interatividade por meio do JavaScript, com eventos de clique nas células do tabuleiro para que os jogadores façam suas jogadas. Essa interação é fundamental para a dinâmica do jogo e proporciona uma experiência de usuário envolvente.

Lógica do Jogo e Inteligência Artificial

Desenvolvemos a lógica do jogo, incluindo a verificação de vitória, empate e

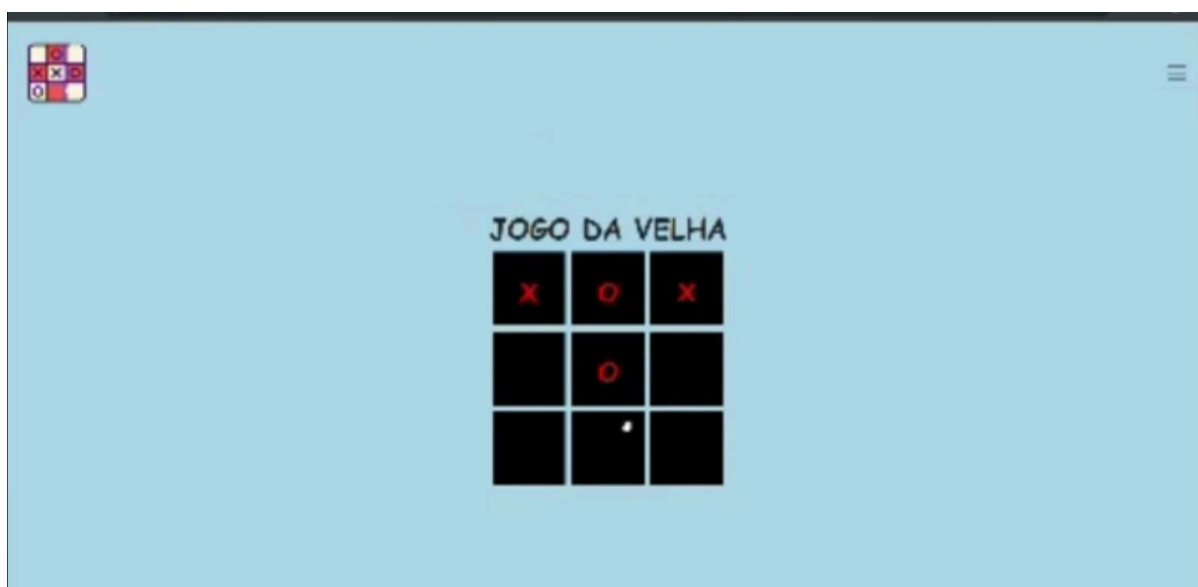
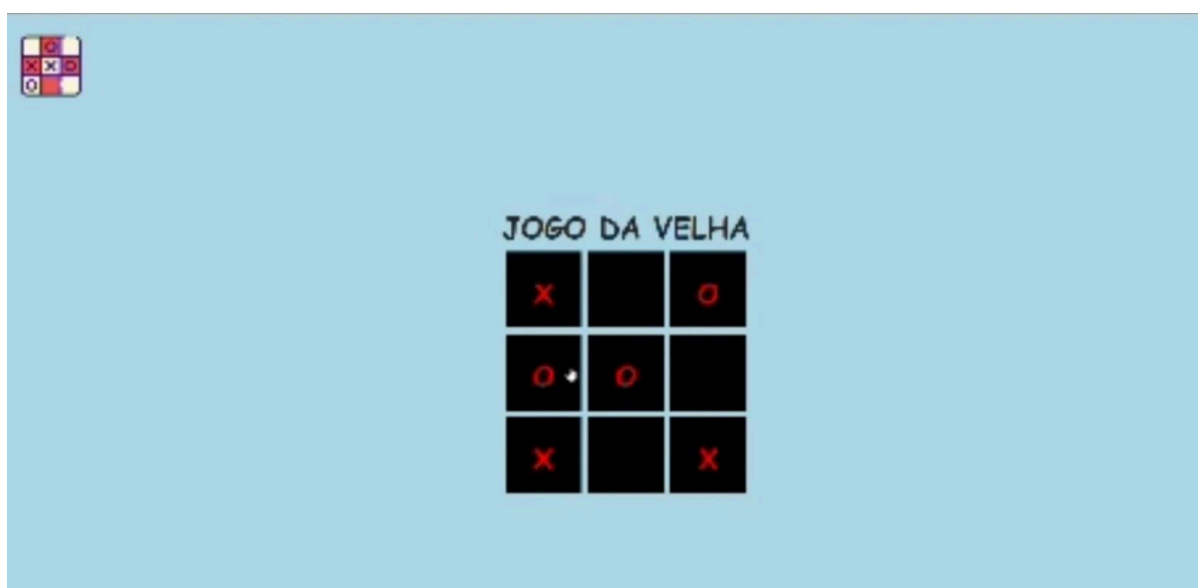
interação com a inteligência artificial, tudo em JavaScript. Utilizamos o algoritmo minimax para a IA, permitindo jogadas estratégicas que desafiam e engajam os jogadores.

Funções Auxiliares e Inicialização

Implementamos funções como `init()` para iniciar o jogo, `drawBoard()` para desenhar o tabuleiro, `isFull()` para verificar se o tabuleiro está completo, e `checkWinner()` para determinar o vencedor. Essas funções são cruciais para o funcionamento do jogo e proporcionam uma experiência consistente e fluida.

Em resumo, nosso grupo desenvolveu o jogo da velha com inteligência artificial, apresentando uma estrutura bem organizada, tecnologias adequadas e funcionalidades essenciais para proporcionar uma experiência de jogo interativa e desafiadora.

ELEMENTOS VISUAIS





JOGO DA VELHA

X		O
O		
X		X

TESTES

TESTES UNITÁRIOS:

Movimento:

- Teste de movimento válido para garantir que o jogador possa fazer uma jogada em uma posição vazia no tabuleiro.
- Teste de movimento inválido para verificar se o jogador não pode fazer uma jogada em uma posição já ocupada.

Verificação de Vitória:

- Teste de vitória horizontal, vertical e diagonal para garantir que o jogo reconheça corretamente quando um jogador vence.
- Teste de empate para verificar se o jogo é corretamente declarado como empate quando o tabuleiro está cheio e não há vencedores.

Algoritmos de IA:

- Teste do algoritmo minimax para garantir que a IA faça as melhores jogadas possíveis em diferentes cenários.
- Teste de poda alfa-beta para avaliar a eficiência do algoritmo de busca em árvores de jogo grandes.

TESTES DE INTEGRAÇÃO:

- Teste de integração entre a interface do usuário e a lógica do jogo para garantir que as interações do jogador sejam refletidas corretamente no tabuleiro.
- Teste de integração entre o algoritmo de IA e a lógica do jogo para garantir que a IA faça jogadas válidas e eficientes.

TESTES DE PERFORMANCE:

- Medição do tempo de resposta da IA para diferentes profundidades de busca no algoritmo minimax.
- Simulação de jogos repetidos para avaliar a capacidade da IA de tomar decisões consistentemente boas.

TESTES DE INTERFACE:

- Teste de usabilidade para verificar a facilidade de uso da interface do jogo.
- Teste de resposta da interface para garantir que a interação do jogador seja suave e responsiva.

TESTES DE CONFORMIDADE:

- Teste de conformidade com as regras padrão do jogo da velha para garantir que todas as regras sejam seguidas corretamente.
- Teste de validação dos movimentos para impedir movimentos inválidos de

jogadores.

TESTES DE REGRESSÃO:

- Reexecução de todos os testes após cada alteração no código para garantir que novos recursos não causem regressões.
- Teste de regressão específico para verificar se correções de bugs anteriores permanecem corrigidas.

RESULTADOS

TESTES UNITÁRIOS;

Movimento:

- Todos os testes de movimento foram bem-sucedidos, garantindo que os jogadores só possam fazer jogadas em posições válidas.

VERIFICAÇÃO DE VITÓRIA

- Os testes de vitória horizontal, vertical e diagonal estão funcionando corretamente, identificando corretamente quando um jogador vence.
- O teste de empate também está funcionando conforme o esperado, detectando empates quando o tabuleiro está cheio e não há vencedores.

ALGORITMOS DE IA

- O algoritmo minimax está produzindo resultados precisos, fazendo a IA tomar as melhores decisões possíveis em diferentes situações.

- O algoritmo de poda alfa-beta demonstrou uma melhoria significativa na eficiência de busca, reduzindo o número de nós avaliados durante a busca.

TESTES DE INTEGRAÇÃO

- A integração entre a interface do usuário e a lógica do jogo está funcionando sem problemas, refletindo corretamente as jogadas do jogador no tabuleiro.
- A integração entre o algoritmo de IA e a lógica do jogo permite que a IA jogue de forma eficiente e tome decisões estratégicas.

TESTES DE PERFORMANCE

- A IA responde rapidamente mesmo em profundidades de busca mais altas, garantindo uma experiência de jogo suave e sem atrasos perceptíveis.
- As simulações de jogos repetidos mostram que a IA mantém um desempenho consistente, demonstrando sua capacidade de tomar decisões eficazes de forma consistente.

TESTES DE INTERFACE

- Os testes de usabilidade indicam que a interface do jogo é intuitiva e fácil de usar para os jogadores.
- A interface responde rapidamente às interações do jogador, proporcionando uma experiência de jogo agradável e responsiva.

TESTES DE CONFORMIDADE

- O jogo cumpre todas as regras padrão do jogo da velha, garantindo que todas as jogadas e resultados estejam de acordo com as expectativas dos jogadores.
- Os testes de validação de movimentos impedem movimentos inválidos, mantendo a integridade do jogo.

TESTES DE REGRESSÃO

- Todos os testes de regressão foram bem-sucedidos, garantindo que as correções de bugs anteriores permaneçam corrigidas e novos recursos não introduzam regressões no código.
- As alterações recentes no código foram validadas através dos testes de regressão, garantindo a estabilidade e funcionalidade contínua do jogo.

REFERÊNCIAS

▶ **12 Beginner Python Projects - Coding Course**

▶ **Inteligência Artificial - Jogo da Velha c/ Procedimento Minimax**

REFERÊNCIAS BIBLIOGRÁFICAS

"Artificial Intelligence: A Modern Approach" (Russell & Norvig, 2022): Este livro é uma referência amplamente utilizada em cursos de inteligência artificial. Ele aborda vários tópicos relevantes, como algoritmos de busca (como minimax para jogos), representação do conhecimento e aprendizado de máquina, todos essenciais para criar uma IA para o jogo da velha.

"Artificial Intelligence: Structures and Strategies for Complex Problem Solving" (Luger & Stubblefield, 2017): Este livro também é bem abrangente e aborda estruturas de dados e algoritmos comuns em IA, incluindo representações de estado e algoritmos de busca como minimax com poda alfa-beta, que são fundamentais para jogos como o da velha.

"Artificial Intelligence: A New Synthesis" (Nilsson, 2014): Este livro oferece uma perspectiva ampla sobre a IA, incluindo temas como planejamento, raciocínio e aprendizado.

"Artificial Intelligence" (Rich & Knight, 1991): Embora seja uma edição mais antiga, este livro fornece uma base sólida em conceitos de IA, incluindo estruturas de representação de conhecimento e técnicas de busca que podem ser aplicadas na implementação de uma IA para jogos.

"Artificial Intelligence: Structures and Strategies for Complex Problem Solving" (Stuart & Russell, 2010): Esta edição do livro aborda estruturas e estratégias para resolver problemas complexos, como os encontrados em jogos. Algoritmos como minimax e suas variantes são discutidos em detalhes.

CONCLUSÃO

Desenvolvemos com sucesso uma inteligência artificial para jogar jogo da velha usando o algoritmo Minimax. Nossa IA demonstrou competência em jogar contra humanos e outras IA's, oferecendo insights valiosos sobre a aplicação prática de algoritmos de inteligência artificial em jogos. Para futuras iterações, consideramos implementar algoritmos de poda alfa-beta para melhorar o desempenho em jogos mais complexos. Este projeto foi uma experiência empolgante e educativa, fornecendo uma base sólida para explorações futuras no campo da inteligência artificial em jogos.