O jogo satisfaz o seguinte enunciado:

Algoritmo de Busca Competitiva

Algoritmos de busca competitiva são técnicas usadas em inteligência articial (IA) para encontrar a melhor solução em um ambiente competitivo onde diferentes agentes estão tentando otimizar seus próprios resultados. Esses algoritmos são comumente aplicados em jogos de dois jogadores, como Xadrez, Damas ou Connect4 (objetivo do nosso trabalho), onde os jogadores estão em competição direta um contra o outro. A ideia básica é avaliar as diferentes jogadas possíveis para determinar a melhor ação a ser tomada, considerando as possíveis respostas do oponente. Isso geralmente é feito através da construção de uma árvore de busca, onde cada nó representa um estado do jogo e as arestas representam as possíveis jogadas. Os algoritmos de busca competitiva, como Min-Max e Alfa-Beta, são então usados para explorar essa árvore e determinar a melhor sequência de jogadas para alcançar o objetivo desejado, geralmente a vitória no jogo. Min-Max: Este é um algoritmo de busca em árvore usado em jogos de dois jogadores com informação perfeita. O algoritmo alterna entre os papéis de maximizar e minimizar os valores dos nós na árvore de jogo. Ele também calcula recursivamente os valores dos nós da árvore, assumindo que o oponente fará a melhor jogada possível. Por um, retorna a melhor jogada possível para o jogador maximizador. Alfa-Beta: Melhoria do algoritmo Min-Max para reduzir o número de ramos avaliados. Usa limites alfa (α) e beta (β) para podar ramos desnecessários da árvore de jogo. Evita a exploração de ramos que não podem melhorar a melhor escolha atual. Efetivamente diminui o tempo de computação em comparação com o Min-Max padrão. Outro ponto importante é conhecer os níveis de dificuldade dos algoritmos de Inteligências Arti-ficiais por trás desses jogos. Por exemplo, em contexto de jogos, ply se refere a uma única jogada completa de um jogador. Em outras palavras, uma ply consiste em um movimento feito por um jogador, seguido pela resposta do oponente, se houver. Por exemplo, em um jogo de Connect4, um movimento completo seria uma ply. Se as X fazem uma jogada e, em seguida, as O fazem uma jogada em resposta, isso representa duas plies(ou meias jogadas), uma para cada jogador. Portanto, uma plies é uma única ação ou jogada em um jogo. No contexto dos algoritmos de busca em árvores de jogos, como o Min-Max ou Alfa-Beta, o termo ply é usado para denotar a profundidade da busca na árvore de jogos. Por exemplo, se um algoritmo Min-Max alcançou uma profundidade de 3 plies, signica que ele considerou três jogadas completas alternadas entre os jogadores. Neste trabalho, vamos explorar o jogo Connect 4 (Conectar 4) é um jogo de tabuleiro de estratégia simples, mas desafiador. O objetivo é ser o primeiro a alinhar 4 peças de sua cor em uma linha, seja horizontal, vertical ou diagonal. Este trabalho propõe a implementação dos algoritmos de busca Minimax e Poda Alfa-Beta para resolver o jogo Connect 4 de forma automatizada e a comparação do tempo de processamento desses algoritmos em diferentes profundidades de busca, chamadas de ply. Ou seja, nesse caso, teremos homem x máquina (Inteligência Articial). Vamos simular o jogo de Connect 4 com a analogia de dois almirantes que comandam esquadras em alto-mar, onde o objetivo é disparar tiros certeiros e destruir as embarcações inimigas, e as embarcações são representadas pelas peças dos jogadores no tabuleiro.

Objetivo do Trabalho Implementar o Jogo Connect 4 para dois jogadores usando a lógica de estratégias baseadas em inteligência articial. Implementar os algoritmos Minimax e Poda Alfa-Beta para busca da melhor jogada. Congurar o ply de forma manual (1, 2, 3, 4). Comparar os tempos de processamento entre os algoritmos Minimax e Poda Alfa-Beta para diferentes profundidades de busca.

Metodologia 1. Implementação do Jogo Connect 4 Tabuleiro: O tabuleiro é uma matriz 7 × 8, onde 7 são as linhas e 8 são as colunas. Regras do Jogo: Os jogadores alternam turnos para inserir suas peças (representadas por dois tipos de símbolos: por exemplo, X e O ou ainda cores diferentes) nas colunas. A primeira linha de uma coluna sempre será preenchida, e, à medida que o jogo avança, as peças vão caindo para a linha mais baixa disponível. O jogo termina quando um jogador alinha 4 peças ou quando o tabuleiro está cheio, resultando em um empate.

2. Algoritmo Minimax: conforme já explicado anteriormente, o Minimax é um algoritmo de busca de decisão utilizado em jogos de soma zero, como o Connect 4. Ele explora todas as possibilidades de jogadas e escolhe a melhor, assumindo que o adversário também jogará da melhor maneira possível. A árvore de decisões é gerada recursivamente, com o algoritmo alternando entre o jogador maximizador (tentando maximizar o valor) e o jogador minimizador (tentando minimizar o valor). O algoritmo de Minimax vai até uma profundidade especificada, que é o ply. O valor de cada nó é calculado com base no número de movimentos possíveis até a vitória ou derrota de cada jogador. 3. Poda Alfa-Beta: a Poda Alfa-Beta é uma otimização do algoritmo Minimax que permite evitar a exploração de ramos da árvore de busca que não vão alterar a decisão final. Isso acelera significativamente o processo de busca, pois elimina a necessidade de avaliar todos os nós da árvore de decisões. A poda alfa-beta mantém dois valores durante a busca:

Alfa: O valor mais alto encontrado até o momento no lado do jogador maximizador. Beta: O valor mais baixo encontrado até o momento no lado do jogador minimizador. Quando um ramo da árvore de decisão já é pior do que uma decisão já tomada, ele é podado e não é explorado.

4. Configuração do Ply: O ply representa a profundidade da árvore de busca, ou seja, quantos movimentos à frente o algoritmo irá considerar. A profundidade influencia diretamente o tempo de processamento e a qualidade da decisão. Neste trabalho, o usuário configurará manualmente o ply, testando as profundidades de 1, 2, 3 e 4 para comparar o tempo de processamento entre os algoritmos Minimax e Poda Alfa-Beta.

Implementação: Passos Detalhados Implementação do Jogo Connect 4. Criar um tabuleiro 7 × 8. Implementar as funções de verificação de vitória (linhas, colunas, diagonais). Implementar a função de jogada: adicionar uma peça na coluna escolhida e verificar se a jogada resultou em vitória. Algoritmo Minimax: Função recursiva de Minimax que explora todos os movimentos possíveis até a profundidade dada (ply). Para cada nível da árvore de busca, alternar entre maximizar e minimizar, de acordo com o jogador. Implementar uma função de avaliação para atribuir valores aos estados finais do jogo. Algoritmo de Poda Alfa-Beta: Modificar a função de Minimax para incluir as condições de poda alfa-beta. Implementar a poda durante a exploração da árvore, interrompendo a busca de ramos que não contribuem para a decisão final. Configuração do Ply: Permitir ao usuário definir a profundidade da árvore de busca (ply) através de entrada manual. Executar os dois algoritmos (Minimax e Poda Alfa-Beta) para diferentes profundidades de ply (1, 2, 3, 4 - se possível). Medição de Tempo: Medir e registrar o tempo de execução para cada configuração de ply usando a função time() em Python ou equivalente em outra linguagem. Comparar os tempos de execução entre os dois algoritmos para cada profundidade. Experimentos e Resultados.Experimentos de Tempo de Execução: Teste com ply = 1, ply = 2, ply = 3 e ply = 4. Comparar os tempos de execução do Minimax e Poda Alfa-Beta para cada profundidade. Resultados Esperados: O tempo de execução do Minimax aumentará exponencialmente à medida que a profundidade aumenta. O tempo de execução da Poda Alfa-Beta deve ser signicativamente menor que o do Minimax, especialmente para profundidades maiores, devido à eliminação de ramos inúteis. Este trabalho visa analisar a diferença de desempenho entre dois algoritmos de busca clássicos para o jogo Connect 4. O Minimax, por ser uma abordagem de busca completa, tende a ser mais lento à medida que a profundidade aumenta. Já a Poda Alfa-Beta é uma otimização signicativa, especialmente para jogos mais complexos, onde a busca em profundidades maiores pode ser impraticável sem essa poda. Ao final, espera-se que se tenha uma compreensão prática da aplicação de algoritmos de busca em jogos e da importância de otimizações como a poda alfa-beta para melhorar a performance de algoritmos em jogos com grande espaço de busca.

Critério de Avaliação O jogo deve ter uma interface gráfica de usuário 3D ou 2D, que deve ser intuitiva e com boa usabilidade. O jogador deve escolher entre Jogar com a IA com Busca Competitiva por Min-Max ou IA com Busca Competitiva por Alfa-Beta. O jogador também deve escolher entre o nível fácil, médio, difícil e muito difícil (ply).