UNIVERSIDADE FEDERAL DE SANTA CATARINA

CENTRO TECNOLÓGICO

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

INE5430 - INTELIGÊNCIA ARTIFICAL

Relatório - Minimax

Eduardo Borges Siqueira

Rafael Moresco Vieira

Florianópolis

2023

Sumário

1. Introdução

1.1 Connect 4

O Connect 4, também conhecido como "Quatro em Linha" ou "Lig 4", é um popular jogo de estratégia para dois jogadores. Foi inventado por Howard Wexler em 1974 e lançado como um jogo de tabuleiro pela Milton Bradley Company (agora parte da Hasbro). Desde então, o jogo também foi adaptado para versões digitais e é amplamente jogado em todo o mundo.

O tabuleiro do Connect 4 é composto por uma grade vertical de 6 colunas e uma grade horizontal de 7 linhas, resultando em 42 espaços no total. Cada jogador tem um conjunto de fichas coloridas (geralmente uma cor é vermelha e a outra amarela), e o objetivo do jogo é ser o primeiro a conectar quatro de suas fichas da mesma cor em linha reta, seja na vertical, na horizontal ou na diagonal.

As regras são simples: os jogadores alternam, colocando uma ficha por vez em uma das colunas vazias. A ficha cai para a posição mais baixa possível na coluna escolhida. O jogo requer estratégia, pois os jogadores precisam planejar suas jogadas para bloquear o oponente e criar oportunidades para suas próprias combinações de quatro fichas.

Connect 4 é um jogo fácil de aprender, mas oferece desafios estratégicos significativos, especialmente quando jogado entre jogadores experientes. É um passatempo divertido para todas as idades e uma ótima maneira de desenvolver habilidades de pensamento lógico e planejamento tático. Além disso, sua simplicidade e portabilidade o tornam uma escolha popular para jogos de viagem e competições casuais entre amigos e familiares.

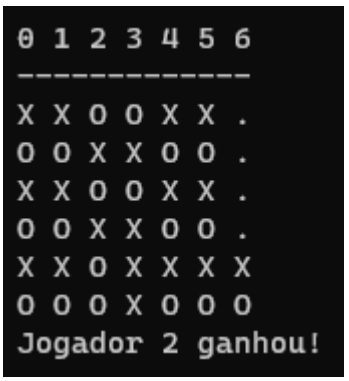
1.2 EasyAI

Para desenvolver o Connect 4, foi utilizado a linguagem de programação Python, com a biblioteca EasyAI. Projetada para facilitar a criação e implementação de algoritmos de IA em diversos contextos, a EasyAI oferece uma interface simples e intuitiva, tornando a construção de sistemas inteligentes mais acessível até mesmo para desenvolvedores com menos experiência. Com uma ampla gama de algoritmos e recursos para jogos, otimização, busca e muito mais, a EasyAI é uma escolha popular para aqueles que desejam explorar e experimentar com conceitos de IA em projetos Python, acelerando o processo de desenvolvimento e aprendizado no campo da inteligência artificial.

2. Heurísticas e Comparações

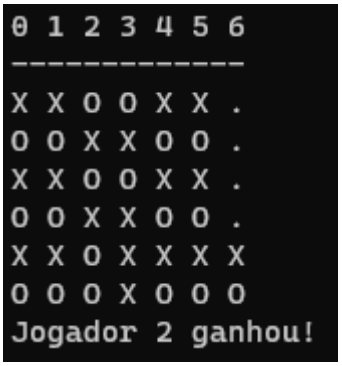
A primeira heurística implementada é a mais básica o possível, um valor extremamente baixo representando a derrota no jogo. Com essa estratégia, a IA tenta impedir que o jogador ganhe, mas nunca faz nada para tentar ganhar.

A imagem a seguir representa o final de um jogo onde as IAs apenas jogam na coluna mais à esquerda, porém caso enxerguem uma condição de vitória do oponente, elas jogam para anular essa condição.



A mudança de comportamento ocorre na jogada referente a jogada do jogador 1 na coluna 2, que levaria para uma condição de vitória. Visto isso, o jogador 2 (X), anula essa condição com um X na coluna 3.

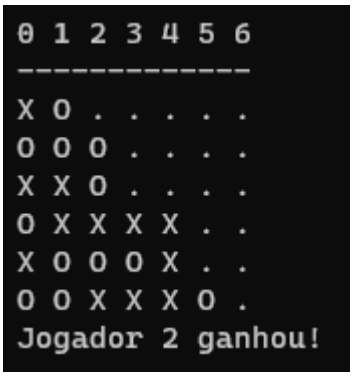
O passo lógico para melhorar a IA é adicionar além de uma condição de derrota, uma condição de vitória. Para um jogo com poucas possibilidades, ou com suficientes passos futuros analisados, uma estratégia [-1, 0, 1] pode ser o suficiente.



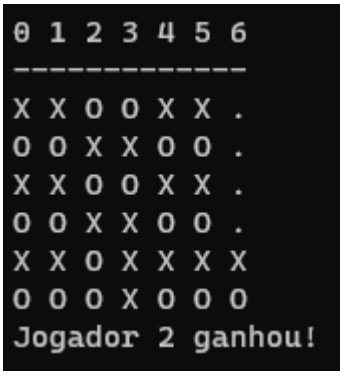
Rodando novamente, nota-se que o comportamento permanece idêntico.

Para tentar adicionar uma visão mais estratégica, é necessário avaliar a condição do tabuleiro. Podemos dar preferência para posições específicas, ou para formas específicas. Foi então colocado duas avaliações, com uma pontuação intermediária, para 3 peças em ordem. Conseguir 3 peças alinhadas vale [0.5], enquanto o adversário conseguir 3 peças alinhadas vale [-0.5].

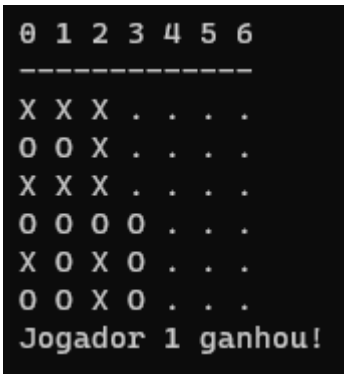
A seguir, temos uma imagem do resultado de um jogo onde a IA é incentivada a alinhar 3 peças.



Já aqui, temos o resultado com a IA tentando evitar que o adversário coloque 3 peças seguidas:



Agora, quando adicionamos as duas heurísticas, obtemos o seguinte resultado:



Com isso, observamos os seguintes comportamentos:

- Com apenas a heurística positiva, obteve-se um pensamento estratégico por parte das IAs, onde percebe-se que as jogadas pararam de ocorrer somente na coluna da esquerda.
- Com apenas a negativa, o comportamento foi igual ao comportamento enxergado nas situações de apenas vitória ou derrota.
- Com ambas, nota-se que o jogo foi muito mais curto, devido ao comportamento mais focado das IAs, que permitiu o jogador 1 (naturalmente com vantagem), finalmente ganhar.

Com isso, podemos notar a necessidade de comportamentos extras para esse jogo, porque percebe-se que com apenas 5 passos a IA não consegue se comportar corretamente.

3. Execução

Para executar o código, é necessário o Python 3.10 ou mais novo, além dos pacotes EasyAI e Numpy. Ambos podem ser adquiridos através do instalador de pacotes do python *pip*.

Com os pacotes instalados, é só executar o arquivo connect4.py com o Python. Primeiramente, o programa pede o modo de jogo, que deve ser selecionado dentre as seguintes opções:

- Jogador contra IA
- IA contra IA

Selecionado o modo, deve-se escolher a quantidade de jogadas que a IA irá considerar.

Com o jogo iniciado, o jogador digita em qual coluna ele quer jogar quando indicado pelo programa. As colunas são numeradas de 0 a 6. O primeiro a jogar é decidido aleatoriamente, mas o jogador humano, definido como jogador 1, possui as peças representadas por “O”, enquanto a IA é definida como jogador 2 e representada por “X”.

4. Referências

https://pt.wikipedia.org/wiki/Lig_4

http://en.wikipedia.org/wiki/Connect_Four

https://zulko.github.io/easyAI/