

Implementação minimax

Aluno: Rafael Begnini de Castilhos (20205642)

Introdução:

O código apresentado é uma implementação do jogo Dots Game, usando a biblioteca easyAI, que permite a criação de jogos de lógica simples com o suporte de jogadores humanos e jogadores de inteligência artificial. O jogo Dots Game envolve dois jogadores que alternam para conectar pontos em um tabuleiro quadrado de tamanho variável ($N \times N$). O objetivo é criar um quadrado completo, tomando pontos adjacentes. O jogador que completa o quadrado ganha pontos. O código também inclui uma opção para usar a pesquisa com aprofundamento iterativo para resolver o jogo automaticamente.

Passo a passo:

Para instalação das bibliotecas necessárias, é necessário executar:

```
pip3 install easyAI colorama
```

Para inicializar o jogo, é necessário executar:

```
python3 game.py
```

Opções de jogo:

- 1) Jogar com adversário a IA:

O adversário IA é definido como o Jogador 1, e você usuário como Jogador 2. O tabuleiro é definido da seguinte maneira:

Tabuleiro	Posições
...	00 01 02
...	03 04 05
...	06 07 08

E para realizar ações, basta digitar a posição 1 e posição 2, que serão conectadas, da seguinte forma: (0, 3) irá conectar a posição 0 com a posição 3, e assim por diante, quando finalizar as ações, basta ver no terminal quem é o vencedor.

- 2) Observar duas IAs jogando uma contra a outra:

Basta observar no terminal o vencedor, e os movimentos de cada jogador. O jogador IA, pensa 5 movimentos com antecedência e usa o algoritmo Negamax, que é uma variante da pesquisa minimax, que se baseia na propriedade de soma zero de um jogo para dois jogadores.

3) Resolução com aprofundamento iterativo:

A documentação do easyAI provém exemplos claros e objetivos para entendimento dessa estratégia de pesquisa, no caso desse jogo, temos $r=1$, o que significa que se ambos os jogadores jogarem perfeitamente, o primeiro jogador a jogar pode sempre ganhar (-1 significaria sempre perder). Para uma execução com o tabuleiro de tamanho 3, temos o seguinte resultado:

```
d:2, a:0, m:(0, 1)
d:3, a:0, m:(0, 1)
d:4, a:0, m:(0, 1)
d:5, a:0, m:(0, 1)
d:6, a:0, m:(0, 1)
d:7, a:0, m:(0, 1)
d:8, a:0, m:(0, 1)
d:9, a:0, m:(0, 1)
d:10, a:0, m:(0, 1)
d:11, a:0, m:(0, 1)
d:12, a:0, m:(0, 1)
d:13, a:0, m:(0, 1)
d:14, a:0, m:(0, 1)
d:15, a:0, m:(0, 1)
d:16, a:0, m:(0, 1)
d:17, a:0, m:(0, 1)
d:18, a:0, m:(0, 1)
d:19, a:0, m:(0, 1)
r: 0
d: 19
m: (0, 1)
```

Conclusão:

O código fornece uma implementação simples do Dots Game usando a biblioteca easyAI, permitindo a interação entre jogadores humanos e IA. O código também oferece a opção de resolver o jogo automaticamente usando aprofundamento iterativo, o que pode ser útil para fins de teste e depuração.