

人工智慧-hw03 縱橫殺棋對抗賽

40823117L 資工系113級 方國丞

1.請說明如何執行你的程式,並詳細說明你所使用之機器軟硬體規格、作業系統、開發軟體版本、相關資訊以及你為何選擇這樣的規格。另外請提供你的連絡電話,以便不時之需

機器：MacBook Pro 2023
CPU/GPU：M2
OS：Mac OS
Python version：Python 3.10.10
電話：0978271809

- 原因: python比較好寫

2. 原始碼中應加註解,請說明如何執行這個程式。

1. 使用 input_generator 來生input1.txt ~ input5.txt

python input_generator.py 2. 執行下棋程式 python chess.py 4. 輸出結果為output.txt，每個input的output使用-隔開

3. 首先你先設法製作測試用的輸入檔至少5個。盤面尺寸需有大有小,解題難度也不同,由你決定。請說明你如何製作這些測試用的輸入檔。(約佔10%的分數。提示:可考慮用亂數產生)

生成測試檔案的程式碼在 input_generator.py

這段程式碼用於生成測試用的輸入檔案，每個檔案包含一個由 0 和 1 組成的矩陣。程式碼中的 generate_test_file() 函數接受兩個整數，代表矩陣的行數和列數。它也可以接受一個可選的檔案名稱參數，預設為 "input.txt"。

執行時，會將行數和列數寫入檔案的第一行，以空格分隔。接下來生成一個 n 行 m 列的矩陣，每個元素都是 0 或 1。每一行的元素會以空格分隔，使用 random.randint(0, 1) 函數隨機生成的

4. 說明程式中你使用甚麼方法、甚麼資料結構、甚麼技術(操練要項)來解決這個問題，並請你說明你測試一些盤面時的表現如何。請你用一些例子輔助說明，並說明你的程式具有那些功能及優點。

方法、資料結構和技術

1. **Alpha-Beta pruning algorithm**：這是一種在遊戲樹中進行搜索的演算法，用於找出最佳的遊戲策略。Alpha-Beta的主要優點是它可以在不影響結果的情況下，減少在尋找最佳走法時需要評估的節點數量，從而提高搜索效率。(前提是這兩位玩家都會做出最佳的選擇)
2. **Numpy**：使用 Numpy 陣列來表示棋盤的狀態。Numpy 是一種高效的多維陣列對象，提供了大量的便利的矩陣運算功能。
3. **遞迴**：在 Alpha-Beta 演算法中，使用遞迴來進行深度優先搜索。
4. **元組(Tuple)**: 使用元組來表示移動，其中第一個元素是行或列的標記，第二個元素是行或列的編號
5. **時間複雜度**: 取決於棋盤的大小和棋子的分佈。worse case下，alpha-beta演算法的時間複雜度是 $O(b^d)$ ，b是棋盤的 branch factor (也就是每一步可能的移動數量)，d是搜索的深度。但是由於剪枝的效果，實際的時間複雜度通常會低於這個值。

主要功能和優點

1. 找出最佳的遊戲策略：程式可以根據當前的棋盤狀態，找出最佳的遊戲策略，即第一個玩家應該移除哪一行或哪一列的棋子。
2. 計算遊戲結果：程式可以計算出如果兩個玩家都採取最佳策略，第一個玩家最終將贏或輸多少分。
3. 計算執行時間：程式可以計算出進行 Alpha-Beta 剪枝搜索所需的時間。

5. 說明你的參考文獻及網站及參考了那些部份用於實作中

- Wikipedia: https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning
- Alpha-Beta pruning algorithm: <https://fu-sheng-wang.blogspot.com/2017/02/ai-16-alpha-beta-pruning.html>
- Stack Overflow
- GPT-4

6. 請說明你發展此程式所碰到的一些狀況及困難

- 1.Alpha-Beta pruning algorithm的實作，要先充分理解這個演算法的理論，才能正確剪枝和評估節點。
- 2.搜索深度問題，在實作剪枝演算法後，有時候當搜索深度加大時，算法無法在有限的時間內找到最佳解，結果反而變得不精確，與預期的深度加大精度變高相反。

7. 額外加分:如果你能利用一些現有的或自己想出來的進階技術,來得出一些成果,則另有加分
