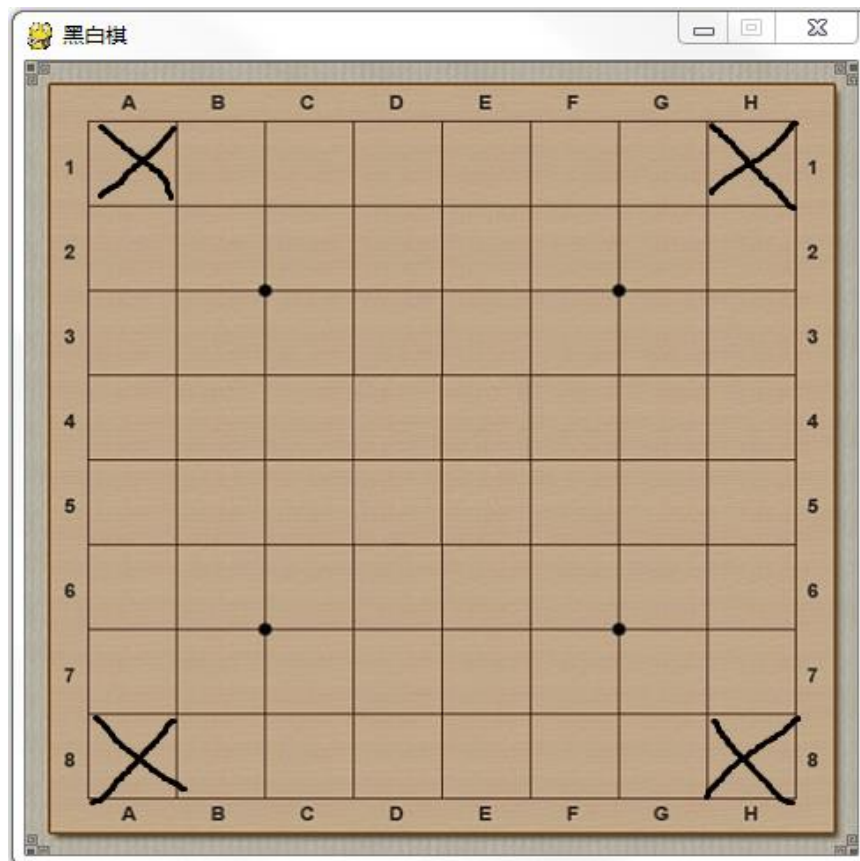


小組 黑白棋 Game AI Player 實作

Rule 規則:

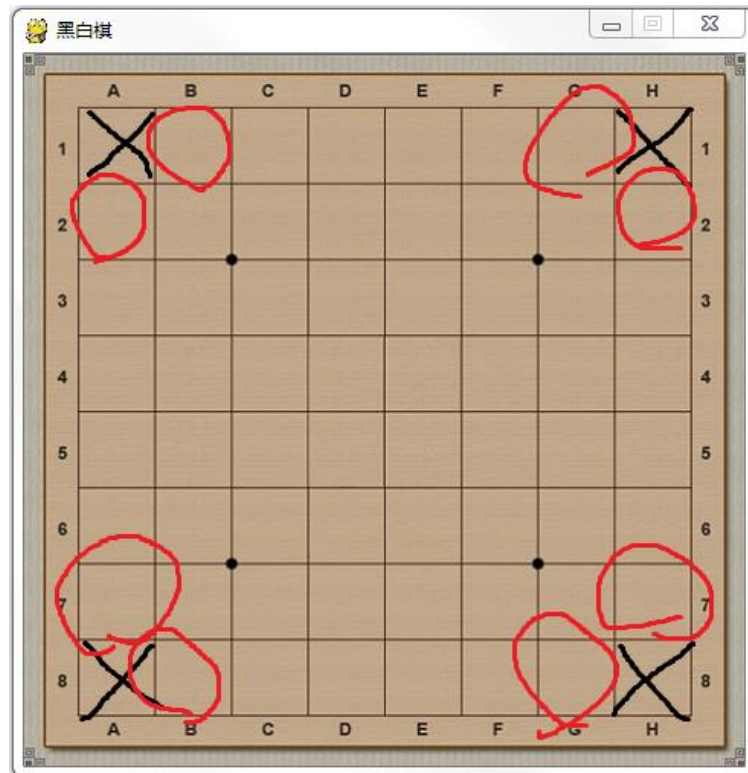
- 跟傳統的黑白棋中間 2x2 已有黑白相間的旗子不同，棋盤一開始是空的 (棋盤的圖片)



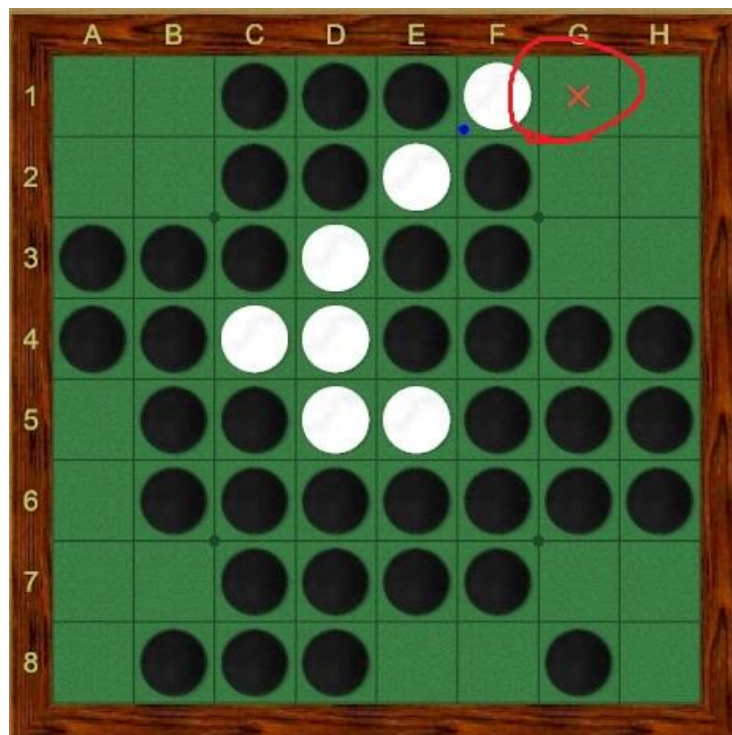
- 黑棋先動
- 4 個角落不能下棋(被排除) -(估計可以減少很多判斷)
- 合法的動作：
 - 下棋的地點必須可以使得現在下棋的顏色可以吃掉對方的棋子，
 - 可以下在中間 6x6 任何一個空白處
- 每一步運算時間限制 5 秒鐘
- 當我們的程式判斷沒有可以下的走法時，伺服器會檢查是不是沒有可以下的走法，如果還有可以下的話會使用可以下的走法
- 如果雙方沒有任何可以走的走法，則遊戲結束
- 分數是棋盤上比較多的那方顏色的棋子數量減去比較少的那方棋子數量，遊戲可能會平局

發現的事

我們發現**八個角落**(如下圖)，因為這八個位置無論如何都不會被吃掉，而且比較容易往內吃掉對手的棋子。除了八個角落外，最外面的邊邊也是比較不容易被吃掉的，所以盡量要在能下最外圈的時候先盡快佔得先機。



星位(相鄰著角的格子)是絕對**不能去佔**的！因為會使**對手很容易搶得角落**

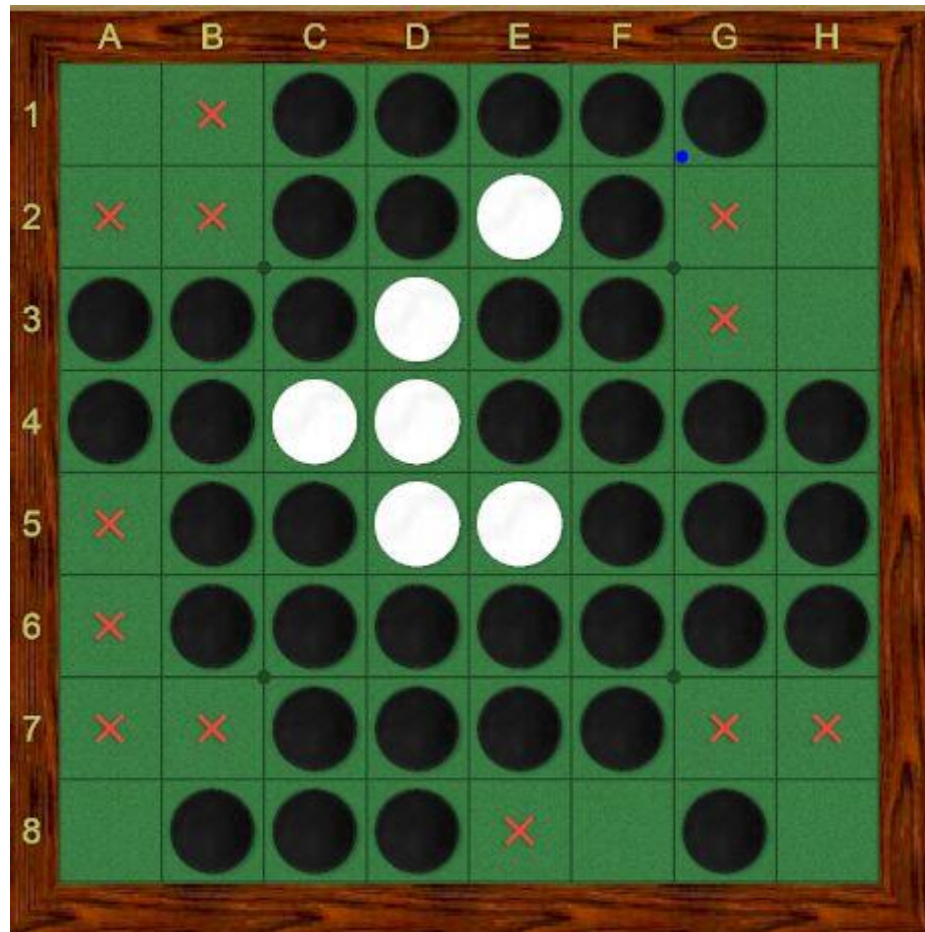


大部分人關注敵我子數來判斷趨勢，這是根本性的錯誤觀念。

在沒占到角之前，子數的比較毫無意義。

不比子數，那比的是什麼？

是行動力（也就是“X”的個數）：



在尾盤之前，對黑白棋指標的判斷，應根據雙方行動力的多少，而不是子數。偏激一點說，黑白棋的爭奪，就是對行動力的爭奪。

根據這點我們可以得出一個結論，把棋子盡量落在棋盤中央位置，因為這樣我們就佔據了主動的位置，我們往外可以隨意的侵占一整條，還可以巧妙避免星位，導致對手最後只能被迫下在星位，最後被我們搶角位吃掉。

總結一下，黑白棋的走法是：搶行動力 > 搶角 > 搶邊 > 子數，這些是有一個因果關係的，搶行動力使得我們更好的搶角，而搶角基本=搶到邊，搶到邊就可以容易搭配中間的棋子搶到一條線，當然最後子數就增加囉。

實作方法

先判斷 **八個角落 > 棋盤邊界 > 中間 6x6 的地區(越中間越好)**來選擇下棋
再判斷有沒有符合規則的一步，先檢查有沒有可以吃掉對方棋子的步數，在判斷哪一步可以吃掉最多的棋子。

如果沒有可以吃掉棋子的走法，則要選擇一個比較安全的走法，先選擇下一步不會被吃掉的一手，也就是周圍沒有其他另一個顏色棋子的地方。

最後如果都沒有的話，則再棋盤中 6x6 範圍內選擇一個位置下。

```
round 1: black:[team 37] white:[team 37]
-----START GAME-----
it's player [black]'s turn
movement: (5, 1)
~ 0 1 2 3 4 5 6 7
0  - - - - - - -
1  - - - - - - -
2  - - - - - - -
3  - - - - - - -
4  - - - - - - -
5  - X - - - - -
6  - - - - - - -
7  - - - - - - -

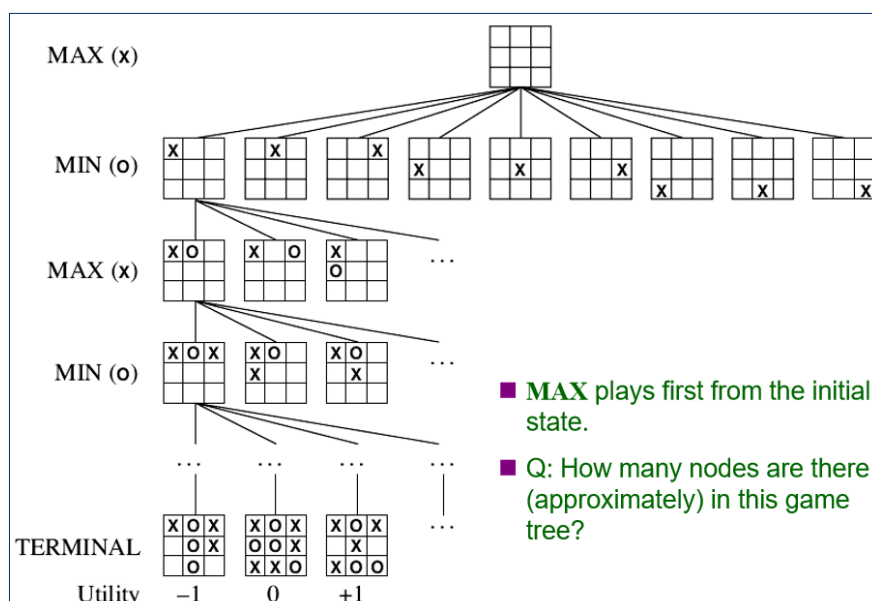
it's player [white]'s turn
movement: (7, 4)
~ 0 1 2 3 4 5 6 7
0  X X X X X X X
1  O X X X X X X O
2  O O X X X X X O
3  O X X X X X X O
4  O O X X X X X X
5  O O O O O O O X
6  O O X O O O X O
7  X X X O X X

-----END GAME-----
Winner is [Black stones], Score is [37 : 23]
```

理想方法

想法 1:

因為棋盤有限，而且不大，可以使用 **Min Max** 的方法，跑到底選擇一個最佳解，最算沒辦法 5 秒內跑到底，也可以根據現在運算到的地方，判斷哪個方法可能比較好，如果可以 min max 到底，就可以得到一個理解中最佳的一手。



想法 2:

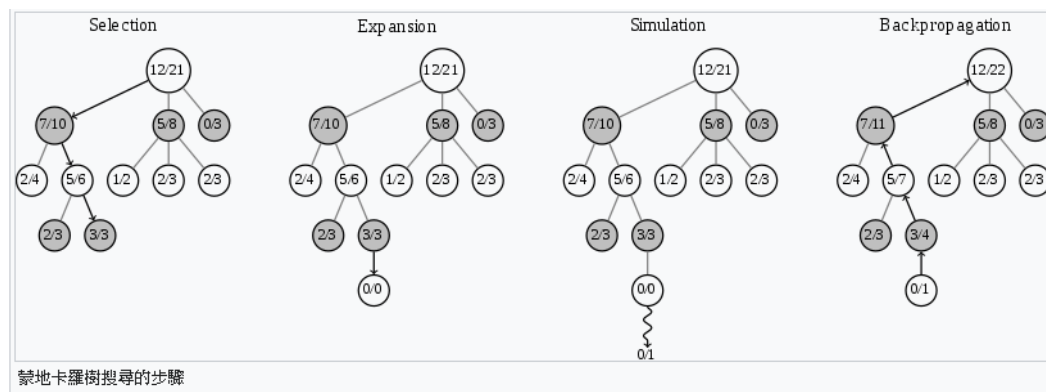
利用 **Neural Network** 讓他自己跟自己下棋，自主學習，來逐步調整運算參數，使他可以判斷出較好的一手

想法 3:

如果考慮 min max 沒辦法在 5 秒內得到最佳解，就要在規則上做更多的限制，在我們原本的下法上，假設是黑棋，下的一手可以吃掉最多棋子且白旗下一步不能吃掉黑棋下的那一步和剛剛吃掉的白棋，憑著**搶行動力 > 搶角 > 搶邊 > 子數**，在落子前應該要先計算行動力，並避免將棋子下在角落旁邊的格子，(因為我們的終極目標就是要搶角落)，這樣我們在尾盤下外圈的棋子的時候，就可以配合中間的棋子，一口氣把一整條全部吃下來

想法 4:

可以改用**蒙地卡羅樹 (Mcts)**的方法



心得

這次的 PROJECT 我們先參考了網路上黑白棋的寫法，原本一開始的想法是用 Min Max 跑完整的棋局做最佳解，後來寫一寫才看到有限制 running time 5s，結果只好砍掉重練，也是上網看了很多 AI 的演算法，但是後來寫不太出來，就又放棄了，再來不小心記錯 Deadline 日期，結果就是弄出了一個不是非常完整的 AI。不過因為助教有稍微更改原本黑白棋的玩法，所以我們也有想出一些新的，估計可以增強 AI 的想法，像是上面提到的 8 個角落>邊界>中間 6x6 的走法，也有上網查到的一些像一開始要搶中間，避免下在星位等等的技巧。這次作業讓我更加理解上課的內容，算是了解如何學以致用吧。

參考資料

黑白棋 python code

https://blog.csdn.net/guzhou_diaoke/article/details/8201542

黑白棋小技巧

<https://www.zhihu.com/question/25271618>

蒙地卡羅樹維基百科