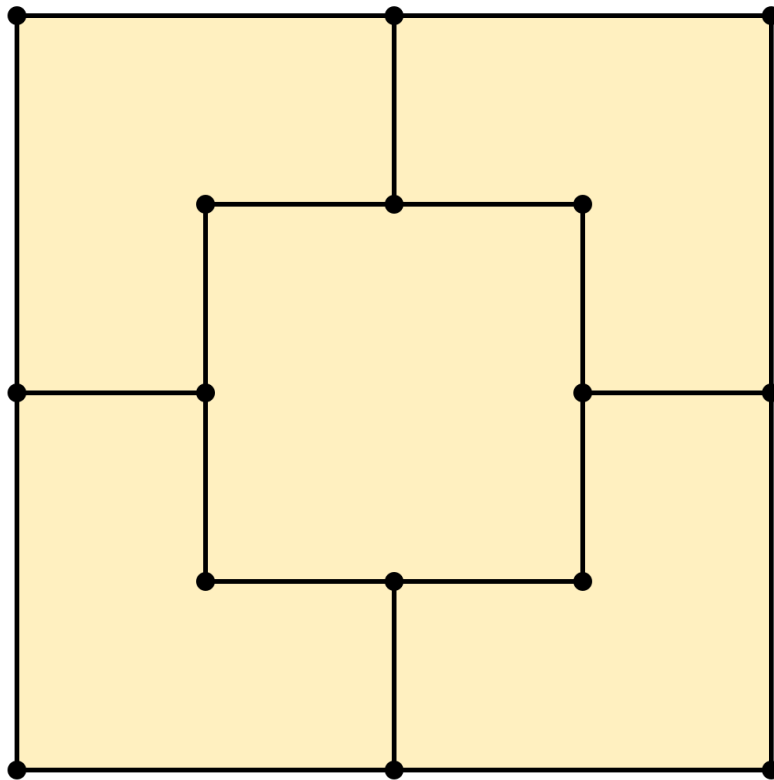


人工智慧 hw3

40647007s 朱健愷

1. 我設計的棋類遊戲人工智慧是莫里斯六子棋，他的盤面如下：



玩家分為白方與黑方，先手為白方，遊戲有四個主要規則。

一、 遊戲分成兩個階段：

- i. 第一階段為擺放棋子，由白方開始與黑方玩家輪流擺放棋子至棋盤上未擺放棋子的黑點空位。

- ii. 第二階段是移動棋子，由白方開始與黑方玩家輪流選擇己方棋子移動到沿著黑線相鄰，無棋子擺放於上的黑點空位。

二、 當己方透過擺放或移動棋子使與該棋子沿同一黑直線的另兩顆己方棋子連成一直線，則形成磨坊。當己方形成磨坊時，己方玩家可以移除敵方玩家的棋子，且該棋子必須是除敵方玩家用以形成磨坊的棋子以外的棋子。

三、 當某方在移動階段時所剩餘的棋子為三顆時，則該玩家可以不受第一條規則所限制，自由移動己方棋子至其上未有棋子的黑點空位。

四、 當某方在移動階段時所剩餘的棋子為兩顆時，則宣告另一方獲勝。

2. 硬體方面我使用 Intel Core i7-8750H CPU，記憶體為 32GB。

使用的作業系統為 Windows 10。

我的連絡電話是 0963674796，不過我更常用的是電子信箱，我的電子信箱是 jack111331@gmail.com。

開發軟體我使用 Sublime Title，因為介面很舒適，Code

Auto Complete 的功能也相當好用。

Python 的版本為 3.7.4，pip 的版本為 20.1.1。

程式因為運用到 pygame 1.9.6，因此需要先透過 pip 安裝
pygame

```
pip install pygame
```

接著鍵入

```
python Six_Men_Morris_GUI.py
```

進入遊戲，Single Player 的選項會讓玩家跟電腦人對打，

Two Player 的選項會讓玩家跟玩家對打。

3. 我參考了 [Alpha-Beta Pruning 的維基百科](#)，並把 Alpha-Beta Pruning 運用在刪去不可能成為最佳解的盤面樹，因為 Alpha-Beta Pruning 會在求可能盤面中的最高價值(我方)時紀錄左邊界、求可能盤面最低價值(敵方)時紀錄右邊界，因此只要遇到超出邊界的盤面就會立即捨棄掉，因為往這盤面繼續搜尋下去並不會有更佳的結果。我搭配的審局函數分為兩個判斷標準：

- 一、 擺放棋子階段時，己方存活的棋子越多價值越高，敵方存活的棋子越多價值越低，會如此設計是因為在擺放旗子階段時需要盡可能確保己方有多餘

的棋子可以形成磨坊或甚至是阻隔敵方，降低敵方形成磨坊的機會。

二、 移動棋子階段時，一樣以己方存活的棋子越多價值越高，敵方存活的棋子越多價值越低作為審局函數，不過價值倍率比上一個價值判斷標準更極端一點。會如此設計是因為在移動棋子階段時需要以攻擊代替防禦，將敵方的棋子快速擊滅能減少敵方形成磨坊的機會。而且由於己方剩餘三顆棋子時能任意移動至任何空位，因此己方存活的棋子多寡並無太大關係，但仍是希望己方盡量保存棋子。

4. 在發展程式的時候，最令我困擾的是審局函數的設計，因為雖然 Alpha-Beta Pruning 可以大幅減少搜索盤面的時間，但若審局函數挑的不好除了有可能增加許多搜索時間外，也有可能讓 AI 表現得相當笨拙，因此最初的審局函數我原本是採用根據當下處於甚麼階段，根據敵方與我方形成磨坊的成本高低，己方所占位置之價值，殺敵數等判斷盤面價值高低。後來仔細想想，可以簡化審局函數為根據己方或敵方存活棋子判斷盤面價值高低，因為最初的審局函數所判斷的盤面價值最終都會影響到己

方與敵方存活棋子的數量，因此最終採用上述提到的己方與敵方存活棋子數判斷盤面價值。

5. 至於表示棋盤的方式，因為盤面只有 16 個點，因此可以簡單使用陣列來表示，但因為這 16 個點之間並無相連規則，因此必須以另一個陣列表示 16 個點之間的鄰接關係。走步的是以該盤面的狀態搭配擺放或移動、移除棋子的位置產生，該走步會記錄經過上述動作後形成的盤面，形成一個節點。會如此做的原因是若要設計成只記錄動作位置的話，那在設計盤面要回溯到先前盤面時，編程複雜度會變得相當高，再者盤面只有 16 個點，並不會消耗太大的記憶體空間，於是採用每個節點紀錄盤面的完整資訊的做法。

我設計的 Alpha-Beta Pruning 他執行的深度最多只會執行到 8 層，在可動作位置相當多的遊戲初期仍可在可接受的時間內給出相當不錯的答案。

6. 在擺放棋子階段時，AI 除了會努力阻斷玩家形成磨坊的機會外，還會積極規劃形成磨坊，這部分使得玩家難以喘息，不過因為前期搜索的可能相當多，審局函數判斷出在前幾步的價值都差不多，Alpha-Beta Pruning 可能發

揮不出太大的效用，不過仍是在可接受的時間內可以給出合理的解答，讓 AI 可以擺放旗子。而在移動階段的 8 步內能形成磨坊的情況下，AI 的表現通常會給玩家很大的壓力，但若需要超過 8 步才能形成磨坊則 AI 的表現會感覺略為笨拙，不過在這階段 Alpha-Beta Pruning 就發揮它的功用，極大的減少了搜索的可能。

我設計的人工智慧優點主要在他在審局函數給出的生成盤面價值有一些差異時即可以快速找出最佳行動點。而圖形介面的部分是可以行動的棋子與位置都有相當明顯的提示。

7. 標題介面的圖是我根據刺客教條 3 遊戲中出現的莫里斯六子棋印象，花了一點時間畫出來的。這隻程式是從[這個人寫的五子棋](#)經過大量修改寫成的，基本上架構是差不多，不過內容已經被我完整重寫過一遍，因為遊戲完全不一樣，因此在介面操作與美化上花了相當大的工夫去根據原本的五子棋程式優化。