# AIC Group Program 1

Team 4 segmentation fault
0616027 陳昱銘 0616003 徐敏恆 0712254 張均聖

- How the game AI works

  α-β pruning

  當AI取得board後，用generate_moves函數生成所有可能的下子位置，並計算這些走法在將來所產生的效益。依照Minimax的概念，選擇獲益最大、損失最小的走法。

- Experiments

  play with Sample1

  round 1:  black:[team 4]  white:[team 1]

  Winner is [Black pieces],  Score is [321 : 33]

  round 2:  black:[team 1]  white:[team 4]

  Winner is [White pieces],  Score is [36 : 308]

  play with Sample2

  round 1:  black:[team 4]  white:[team 2]

  Winner is [Black pieces],  Score is [305 : 69]

  round 2:  black:[team 2]  white:[team 4]

  Winner is [White pieces],  Score is [69 : 329]

從上面的實驗結果可以看出，我們的AI(team4)可以輕鬆勝過助教提供的Sample(先後手皆是)，另外我們有讓自己的AI互打(舊版和改良過)，結果如下可看出，即使後手也能贏下來。
play with old AI(black:old AI, white:new AI)

round 1:  black:[team 4]  white:[team 4]

Winner is [White pieces],  Score is [126 : 272]

round 2:  black:[team 1]  white:[team 4]

Winner is [Black pieces],  Score is [289 : 73]

- Experiences

  我們的AI在計算best move時，是根據evaluation score，也就是下該棋子之後，計算整個board的可能得分(稱之為potential，有潛力得高分之意)，potentia方式計算如下，以4個連線位置為一個window，計算該window內有幾個我方棋子、敵方棋子及未下的棋子(empty)，我方棋子越多則該手所得到之potential越高，e.g.有某四格(window)，我方棋子有三顆，另一個是空的，則該window的得分會相對較高(極有可能連線)，之後AI越有可能下這一手。

  最終計算evaluation會考慮到的點為該board情況下雙方分數及雙方可能得分(potential)，選擇會讓我方得分及potential最高，敵方最低的走法。

  後來我們有對AI進行優化，在產生所有可能的走法時，優先下在附近棋子較多的地方(可以較快找到高分的點)，順便加深搜尋的深度以加強AI，上面的結果證明，該方法確實較優。

- What we learn

  這次專題實作α-β pruning的遊戲策略讓我們更加理解AI的想法，並思考如何優化策略。在撰寫程式時也對強大功能的numpy模組有更多細部了解與實驗。

- Future investigation

  若有時間，希望可以改用MCTS測試是否能獲得更佳的效率；或是用更精確的Evaluation function。

- Contributions
  陳昱銘：板子結構及score計算程式撰寫，實驗結果試驗及整理，找bug，報告書寫

  徐敏恆：evaluation function及計算AI可能走法之程式撰寫，找bug

  張均聖：minimax及search程式撰寫，找bug，實驗報告書寫