

algorithm_TA 架構說明

1. algorithm_B : random version

隨機位置下棋

2. algorithm_C : noLook version

在 algorithm_C 中會找 min(capacity-orbNum)來下棋

3. algorithm_D

- algorithm_D is heithoff version bot，可藉由 gameTree 預測四步後的局勢

Game Tree 介紹

- Feature :

1. (Simple)BoardEvaluator 計算方式：Board 上玩家棋子數量減掉對手棋子數量(orbPlayer-orbOpponent)
2. 在 gameTree::chooseOrb 中，TA 藉由列出各種下棋的可能來建立出 4-level gameTree。

下一步的棋勢紀錄為 firstLevelNode

下兩步的棋勢紀錄為 secondLevelNode

.....以此類推

3. 在 chooseOrb 之中，TA 藉由計算 gameTree 中哪一個棋步所得分數最高來決定下一步要下哪一個位置的棋子
4. algorithm_D 中的設計原則是愈前面的步數(EX:第一步及第二步)比分愈高，愈後面的步數(EX:第三步及第四步)比分愈低

4. algorithm_E

1. algorithm_E is rlawrenc version，比起 algorithm_D 具有更強的 BoardEvaluator
2. algorithm_E 中 BoardEvaluator 計算方式：除了計算棋盤中玩家棋子數量減掉對手棋子數量外，還加上判斷下一步棋子周圍對手的棋子數量來計算分數

