

TCG homework 2 report

學生：莊博任

學號：R05922107

使用語言：C++

compile 方式：makefile

實作內容：

```
1. UCT tree: struct Node{
    int Board[BOUNDARYSIZE][BOUNDARYSIZE]; //node 盤面
    Node *child[HISTORYLENGTH];
    Node *parent;
    double ucb_score;
    double win;
    double sim_time; //該 node simulation 次數
    int num_legal_moves; //下一手總合法走步數
    int num_after_pruned; //pruned 過後的合法走步數
    int turn; //下一手輪誰下
    int GameRecord[MAXGAMELENGTH][BOUNDARYSIZE][BOUNDARYSIZE]; //
    歷史盤面紀錄
    int game_length; //雙方共走了幾手
    int move; //此盤面上一手下哪（主要用於 debug）
    bool expended; //是否已經被 expend 過
    bool pruned; //是否被 pruned，1 則代表此 node 不該再被當作合法
    盤面
}
```

2. MCTS algorithm function:

```
select()
expend()
simulate() //每一手 simulate，先用隨機位置找 30 次，都失敗才用
gen_legal_move() (也有優化檢查歷史盤面)
back_propagation()
sim_all_children() //simulate select node 的所有 child
init_root() //用於產生每步的 root
print_pv() //紀錄 pv path 於 PV_path.txt (只記錄最終 tree 的 select 過程，
```

用 ucb score 判斷，可能結果跟實際下的結果不同（實際下用 win rate 判斷）)

mcts_uct_pick_move() //將上述 function 包起來並判斷最終落子位置，還有決定落子位置之後 delete tree。

3. Progressive pruning function:

prunning() //實作結果不好

4. 優化原 template function:

count_liberty() //改 total_liberty > 1 即回傳

record() //使用 std::copy 複製 record, 並記錄盤面總共子數、總黑子數、總白子數

gen_legal_move() //檢查重複歷史盤面時，先檢查總子數是否相同，再檢查黑子數是否相同，都相同才用原本的線性檢查法。

gen_move() //將原本的記錄時間改用 chrono::system_clock::time_point，以記錄真實時間，以確保在 9 秒內做完。

最終成果：

我執黑跟我做的 AI 下，輸 9.5 目.....

心得：

幾乎是第一次認真寫 C++ code，以前都是寫 JAVA code，查了很多 c++ 的 function 和記憶體相關用法。花了很多時間在 debug select 和 simulate 的 turn 問題，一直搞錯勝率是己方還是對方的。優化的問題問了助教之後改了 count_liberty 和 simulate 先用隨機選取落子位置，但是查了很久都沒辦法實作出用 set 檢查歷史盤面，後來想到先檢查總子數的方法，總算把 10 秒內 simulate 三千次變成兩萬次左右。