

Theory of Computer Games (Fall 2018)

Homework #2

資工所碩二 劉又萱 R06922041

- Implementation

- How to **compile and run**

- ◆ 編譯:我提供了一個 Makefile，可以透過 **make** 指令來編譯，並生成 r06922041 的可執行檔。

- ◆ Makefile:

```
OP=-std=c++11 -O2 -Wall -Wno-unused-result -fopenmp
all:
    g++ $(OP) einstein.cc -c
    g++ $(OP) r06922041.cc einstein.o -o r06922041
    rm -f *.o
clean:
    rm -f r06922041
```

- ◆ 執行: **./game -n 0 otherAI r06922041**

```
Usage: game [-n np agents...] [-r round] [-s seed] [-g] [-l logfile]

np: number of human players (0-2), 2 by default
agents...: the (2-np) AIs
round: number of rounds, 8/∞ when np=0/np≠0 by default, and can only be specified if np=0
seed: random seed for the random part, std::random_device() by default
-g: enable the GUI; can only be specified if np=0
logfile: the file to record the game
```

(助教那邊需要自行編譯出 game 檔案，如下：

g++ \$(OP) game.cc agent.cc getch.cc einstein.o -o game)

- What algorithm and heuristic you implement

- ◆ UCB 分數 $UCB_i = \frac{W_i}{N_i} + c\sqrt{\frac{\log N}{N_i}}$

- Exploitation: 目前得分

1. 勝率: 該節點(含子節點)總共獲勝次數

→ 但後來發現格子容易移到被敵方吃掉的地方，而且與 Greedy 打的勝率大約只有 75%；random 則是全勝。

2. 平均得分: 採用最終盤面平均會吃掉敵方多少格子來計算

→ 計算方式(零和遊戲): (-6~+6)

- 贏家: 6 - 敵方(輸方)剩餘格子

- 輸家: -6 + 我方(輸方)剩餘格子

→ 棋風就會趨向於把敵方格子通通吃掉的目標來贏，勝率對於 Greedy 為 98%；random 則是全勝。

- Exploration: 未來潛力

→ 由於此項目的 C 是一個很重要的參數，由於 Exploitation 為 $(-6 \sim +6)$ ，故 C 會比採用「勝率」的時候 $(C=2.78)$ 大，目前選定 $C=8$

- ◆ Progressive pruning

- 目標: 找出 statistically inferior，把他 hard pruning 掉
- 方法: 每個節點紀錄分數總合以及分數平方和，如此一來要求標準差的時候就可以利用這些該點已經記錄好的資訊來算即可
- 參數: (經由反覆與別人 PK 調整參數)
 1. r_d 通常越大效果越好，但是時間越久，我選擇 $r_d=1$
 2. sig_e 通常越小效果越好，但是時間久，我選擇 $sig_e=1.5$
 3. 由於改成 PP 的方式，每次每個子節點的 stimulate 次數我從原本的 5000 次降為 750 次

- Testing Method

- ◆ 這次遊戲為多人遊戲，當我還沒寫 Progressive pruning 的時候，打 Greedy 的勝率為 98%，實在是無法驗證 PP 的可用性，因此我與朋友們互相 PK，來得到該棋風較準確的勝率，便於精進遊戲演算法，最終打 Greedy 的勝率為 99%。

- Reference

- 整次作業與 R06922025 林家緯、R06922034 王皓正、R06922049 徐誌鴻，共同討論與競爭比賽的結果