姓名：陳聲發

系級：資工109

電腦對局理論與實作

之

期末專題——黑白棋報告

棋盤資料結構

Bitmap， 用兩個unsigned long long int分別儲存棋盤黑白子分佈的訊息，例如負責儲存黑子的位圖中，第n bits為1表示棋盤坐標（floor(n/8), n mod 8）為黑子。

演算法

Negamax search 及 alpha-beta pruning

深度

最開始為7層，中局以後慢慢升至10層，最後20步會暴力搜索所有結果，若某步結果必勝則直接回傳該步為最好步，否則選擇最終棋子最多的。

改進：覺得若必敗應該要選擇對方最有可能失誤的位置，也就是對方下了之後就非必敗的位置比例最高的一步。

審局函數

雙方位置之權重【角 >> 邊 > 中間 > C位(角的兩邊) >> 星位(角的對角)】 以及 己方具權重之行動力。

Hash function

白子旋轉90度後與黑子做xor，再將這64bits的結構前後32bits做xor。由於hash table只能建到2^26個，因此再向右shift 6 bits後與原本的做一次xor，取後面26bits為結果。collision機率很常大於10%，但效果還是比沒有來的好，尤其最後暴力搜索20步後，在往下每一步基本上就能直接得出結果了。

改進：設計collision更少的hash function。

程式碼：

Github 網址<https://github.com/siahuat0727/othello>

比賽過程：

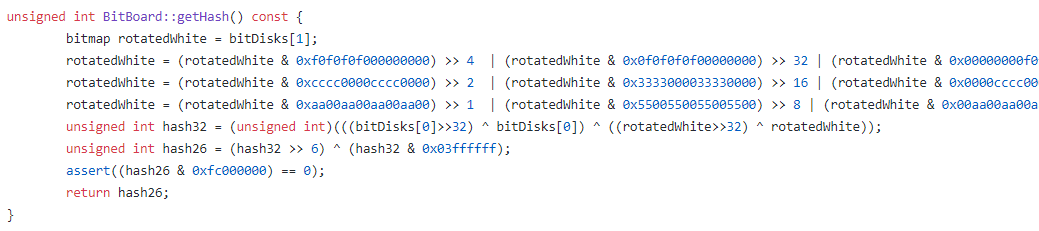
第一次比賽開始前，當時想說反正行動力會最明顯的優勢，於是把行動力調高1倍，結果第一場整個下的超奇怪，一直讓角什麼的。

之後把它改回來後，就一直正常了。

直到和第一名相比的第一場，終局時看起來還蠻勢均力敵的，結果到暴力搜的第一步就說必勝，那之後下的步看起來有點怪怪的，但我原本還是堅信他會險勝。結果倒數幾步時，突然改口回傳必敗了……不太清楚是哪裡的bug，之前都沒遇過啊……

後來第二場就蠻順利的，回傳必勝之後一開始下的步有點怪怪的，但後來真的就險勝了。（因為必勝之後就直接不再搜了，所以可能下得非常危險，但基本上不出bug就會贏）

現在想想，要是第一局沒心血來潮改一改參數，就可以並列第一的呢。再加上要是最後沒這莫名其妙的bug，是可以贏目前的第一名的啊啊，真可惜！



Hash function



審局函數