

遊戲名稱：4096

組員：張巧如（0716326）、劉燕芬（0612542）

AI-G_AI玩家說明文件：

★ 描述AI玩家的設計理念：

目前暫定的情況是，讓電腦能根據每次的面板(2D array)去研判下一次移動的方向(利用case structure)，好讓遊戲有繼續下去的可能(比較不容易輸掉)，若是此構思成功了，接下來，可能還會做一些改變跟要求，像是讓最終結果能越高分越好，或是讓電腦的研判時間降到最低，或是將預測的步數增多。

★ 預定導入的方法：

使用兩種演算法: maxmin search和Alpha-beta剪枝的原理(根據網路上查詢)，再利用 plausible reasoning，將其導入2048遊戲裡的可能性調到最高。

→ maxmin search:

- 原理: 該算法是一個零總和算法，即一方要在可選的選項中選擇將其優勢最大化的選擇，另一方則選擇令對手優勢最小化的方法。而開始的時候總和為0。也就是說，必須考量讓自己得分最多，且讓對手得分最少。必須注意的是，當電腦要下下一個位子之前，會下讓自己得分最多的那一格，但是這很容易會落入對手的陷阱，因為得分最多的那一格很可能接下來失分更多。於是，一個合理的想法是將所有層次分為「敵我雙方」兩類，我方下的那層，自己得分越多越好(即MAX)，而對方下的那層，自己失分越少越好(即MIN)。
- 第一步，先往左下角進行移動(遊戲攻略:L行法)
- 每次在隨機位置出現2(已知條件)
- 紀錄每次的面板矩陣給AI(設計)
- 預測下一步:
 - 1)考慮2會出現在行的哪一個index和欄的哪一個index，去看該行或該列是否有相同數字，若有則進行2)和3)的步驟(即追求最大得分數 \leq 面板最大數字)，若無則滑向0最多的一邊(或一角)(追求最小空格數=目前空格數再+0或+1)，重複1)的步驟
 - 2)考慮行的四個index滑動左方後，看得分結果
 - 3)考慮欄的四個index滑動下方後，看得分結果
- 預測步數為兩步(暫定)
- 在某個方向可移動的情況下，追求最大得分數，比較面板滑動某個方向的sum結果
- 在某個方向不可移動的情況下，追求最小空格數，比較面板滑動某個方向的0個數結果

→ Alpha-beta剪枝:

- ❖ 原理:在搜尋的過程中，若發現無論如何都無法改變對方目前的最佳分數時，就可以提早放棄，不必浪費時間搜尋其樹枝後的下法。又或者，Max方(自己)從自己的利益出發，努力尋找盡可能好的棋步，但是一旦他的棋步好過頭了，極有可能同一層的選擇中有極壞的棋步，畢竟Min方(對手)不可能那樣選擇(讓Max方鐵定贏的局面)，因此當前局面的搜索工作就瞬間變成是多餘的。主要是在 Minimax 當中加入了 α 與 β 兩個紀錄值，用來做為是否要修剪的參考標準。兩個參數以交錯的方式傳遞給下層的子樹。意即:
~~~在最大層(即自己下的那一層)，目前下的步數結果中，取最大值(自己得分)的時候，若發現 $\alpha$ 大於等於 $\beta$ 的值，就不用再對其它分枝進行搜尋(這會往自己鐵定贏的局面)，這就是所謂的  $\beta$  剪枝。

~~~在最小層(即對手下的一層)取最小值(自己失分)的時候, 發現了一個 $\beta$ 小於等於 $\alpha$ 的值, 也不用再對其它分枝進行搜尋(這會往自己鐵定輸的局面), 這就是所謂的 $\alpha$ 剪枝。

- ❖ 用 α 定義為追求最大得分數
- ❖ 用 β 定義為追求最小空格數(反面意思是:得分數=0或2)
- ❖ 將最大層定義為AI玩家滑動後所得的sum結果
- ❖ 將最小層定義為在隨機位置出現的2後, 所能追求的最大空格數
- ❖ 在預測下一步的時候:

~~~在最大層:去尋求此次滑動方向所得的最大sum結果, 接著在預測下一步即果中, 去比較可能有的最大空格數, 如果此最小空格數 $\beta$ 的得分結果大於現在的 $\alpha$ , 則直接放棄

~~~在最小層:去尋求出現此2的位置的行或列, 並追求此次滑動方向所得的最小空格數的結果, 若最大層在滑動後可空出的格數小於現在的 $\beta$ , 則直接放棄

★ 相關的參考資料 :

- ✓ <http://programmermagazine.github.io/201407/htm/focus3.html>
- ✓ <https://itw01.com/D3MJEWV.html>
- ✓ <http://blog.pfan.cn/rickone/16930.html>
- ✓ http://wordstohealbyheart.blogspot.com/2014/04/2048_6.html
- ✓ <https://github.com/ehds/AI-2048-note/blob/master/ai.js>
- ✓ <http://blog.codinglabs.org/articles/2048-ai-analysis.html>
- ✓ <http://mropengate.blogspot.com/2015/04/ai-ch4-minimax-alpha-beta-pruning.html>