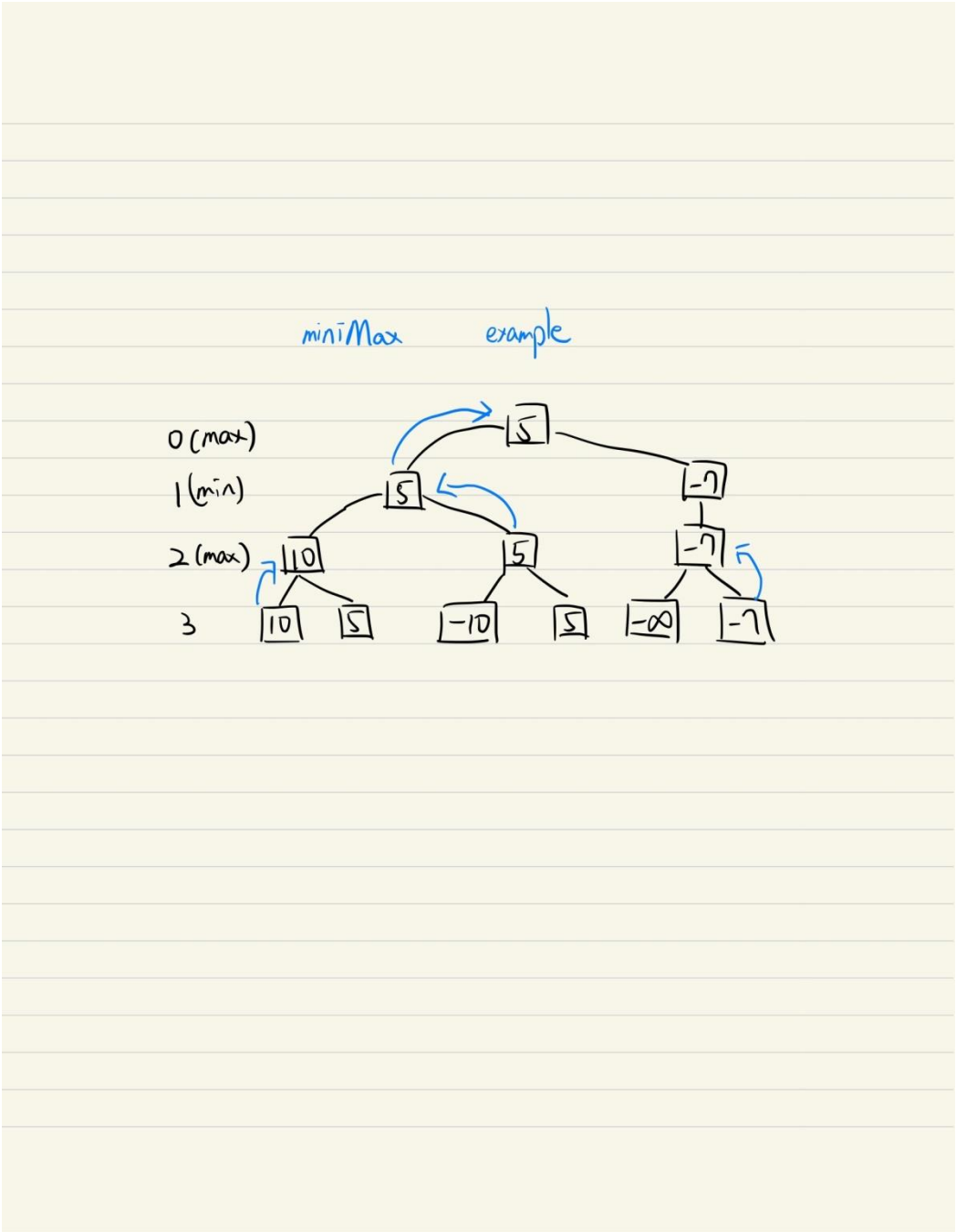
策略邏輯：

A 此程式使用了 chessColor.txt 與 chessStructure.txt 資訊來做出決策

B 目前此程式只有使用一種策略來判斷，是實作 minimax 演算法，算出四層(三回合)中的組合，其中在以評分最高的去做移動，最大值設為 10000，最小值設為 -10000，其中要是達成目標則會獲得相應的分數，例如附近 8 格有第三層就+150…

策略架構：

此程式的架構是以 minimax 演算法為基礎，優點是會討論所有(三回合內)可能的步數，選出最佳解(由自己另的評分系統)來走，缺點則是需要了解走哪個策略的分數比較高來做修改，因此分數的設定就十分重要。



程式概述

A.因為 santorini.c 程式太長所以把函式宣告跟用到的 struct 放在 santorini.h(以合併)

B.各 struct 的用途

i.Board 盤面

1. player1 & player2 用來儲存玩家當前的狀態
2. Cell[][]用來儲存每個 Cell 的狀態
3. isMove 設定目前盤面是移動
4. isPlace 設定目前盤面是建築
5. isP1Turn 保留

i.Cell 盤面格

1. x , y 座標
2. level 當前格的建築樓層 0,1,2,3,4
3. player 當前格的玩家 0:無 , 1:player1, 2:player2

i.Player 玩家

1. pid 玩家編號 1: , 2:
2. Worker w1,w2 這個玩家的兩個工人
3. isAI 你自己的 player 就是 AI

如你是 player1=>isAI=true , player2=> isAI=false

i.Worker 工人

1.pid 所屬的 player , 1:player1 ,2:player2
2.wid 工人編號 1 or 2
3.x,y 目前位置

i.Location

1. x,y 用一個 location struct 變數紀錄 x,y

C.main 主程式

argc=8 , 從 argv[1]~ argv[7]個別代表

argv[1] , pcPlayer 帶進來 1:就是標示自己是 player1 , 2: 就是標示自己是 player2

argv[2], god 自己天神卡

argv[3], opponentgod 對手天神卡

argv[4], placeplayer 是否一開始放置工人(y : isPlacePlayer=0)或是遊戲(n :

isPlacePlayer=1)

argv[5], fcolor 目前玩家位置

argv[6], fstructure 目前建築物位置&樓層

argv[7], fsteplog steplog 檔程式沒用到,但還是要帶進變數

readBoardStream() : 把 fcolor 跟 fstructure 整合到 Board

showcell() : 顯示 Board 內 player , 建築 level 的狀態

initPlayer(Board* board) : isPlacePlayer=0 放置工人函式

pcplay(Board* board) : isPlacePlayer=1 跑你的策略程式

D.pcplay(Board* board)策略程式

MiniMax()函式 : 用來評估 depth 得到最好的走法 ,此處 depth =3

evaluate()函式 :估計 depth 後的評分

MiniMax:

1.一定以自己的 player 也就是 isAI=true 的 player 開始

2.depth=0 or isGameover return 評分(evaluate()函式) :自己

3.如否,取得自己 player 的 worker w1,w2 可以移動的位置(八個方向)

4.移動後(move)判斷是否 gameover,計算評分(evaluate()函式)

5.如沒 gameover 判斷能否建築(place)

6.繼續 MiniMax ,depth -1 ,此時為對手的估計

7.depth=0 or isGameover return 評分(evaluate()函式):對手階段

8.如否,取得對手 player 的 worker w1,w2 可以移動的位置(八個方向)

9.移動後(move)判斷是否 gameover,計算評分(evaluate()函式)

10.如沒 gameover 判斷能否建築(place)

11.繼續 MiniMax ,depth -1 ,此時又為自己的估計 turn 步驟 2

所以會如同下列 depth =3

取 max(自己) maxmin 的 max

取 min (對手) maxmin 的 min

取 max(自己) maxmin 的 max

取 min (對手) 會跑 evaluate()

E.函式有用到指標參數都是參數需要被更新賦值