Final Project

電機工程學系電機己碩一 邱偉城

策略邏輯：

1. 資訊的使用：本策略程式使用了chessColor.txt、chessLayer.txt、chessType.txt等三個文件。

B.策略的決定：

各棋子種類剩餘的棋子數越少權重越大，棋子高度越高權重越大，

//height權重

heightvalue[17]={10,10,40,60,80,90,90,90,90,60,50,45,40,30,20,15,15};

//各種類棋子剩餘數量權重

countvalue[17]={130,130,90,60,40,30,20,10,9,8,7,6,5,4,3,2};

權重大小是根據自身遊玩的經歷與自我腦補得出，各棋子種類剩得越少，權重肯定越高，棋子高度亦是如此，但是棋子高度跌得越高，相反得也會限制到他的可移動性，所以到達一定高度後，權重會下降。

Evaluation function為heightvalue\*countvalue，每個步驟主要以這兩個因素為判斷依據來算出evaluation value，每個步驟動evaluation value最高的那個棋子。

每一回合有兩個步驟要做，第一步驟必吃子，第二步驟可以選擇吃子、併子或不動，以長遠的遊戲來看，第二步驟做併子比較好，因為高度越高可以比較好的防守與攻擊，所以本程式只有在連吃後能贏得勝利才作連吃，其餘情況必做併子，若無法併子則不動。

* 第一步驟：

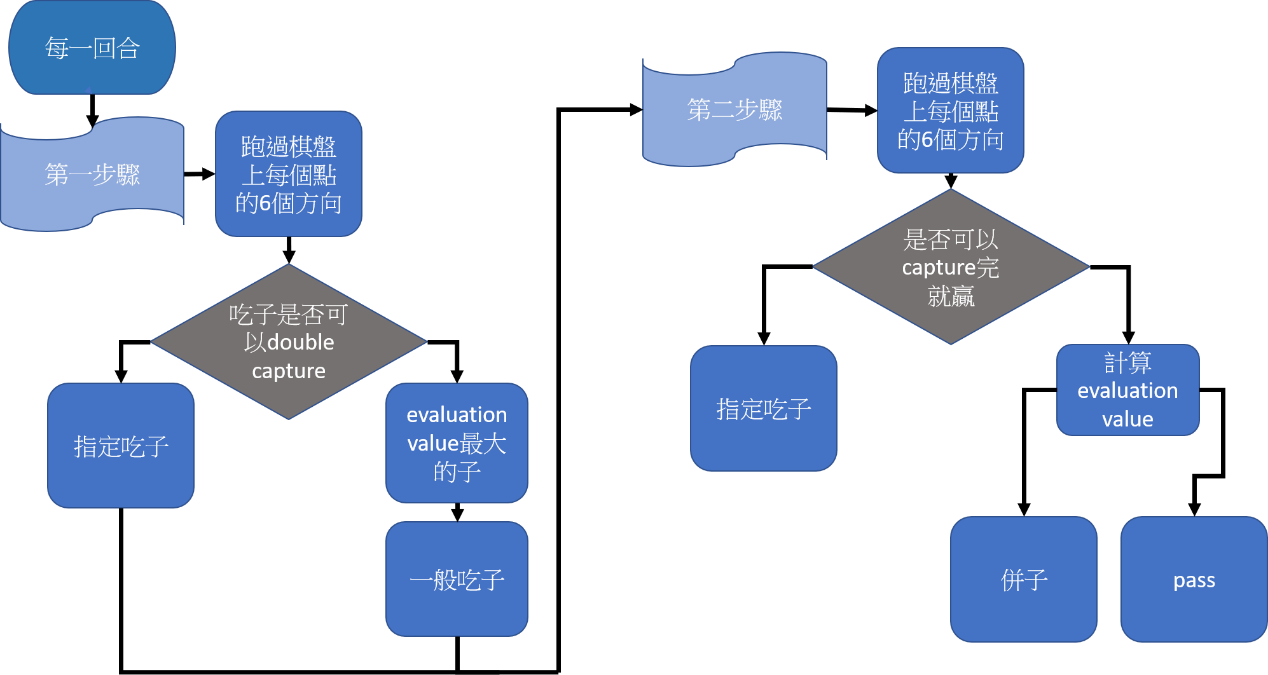
跑過棋盤上每個點求evaluation value，先判斷吃子後再吃子，是否可以使對方某個棋子種類歸零，若是則直接吃子，若否則跑完後吃evaluation value值最大的那個子，若有evaluation value相同的，則動吃過去後較不危險的那個子。

* 第二步驟：

跑過棋盤上每個點求evaluation value，先判斷吃子後是否可以使對方某

個棋子種類歸零，若是則吃子贏得勝利，若否則跑完後併evaluation value值最大的那個子，若有evaluation value相同的，則動合併後lost最少evaluation value的子。

策略架構：



測試：

與boardspace.net的weakbot不管待在白方或黑方都能贏