本次作業共設計了兩版，最後決定使用第一版。

第一版:

策略邏輯:資訊使用:本次程式只讀取BlokusBoard.txt 和LeftPattern\_Color.txt。

策略決定(1代表最優先):

(1) 先找能放的位置(指的是相鄰無自己顏色且對角有自己顏色或放第一次的座標)(從左而右再從上到下)

(2) 將所有能放的位置和磚塊都放上去一遍看看，看怎樣能擋住對方的路最多個(利用上述所存的敵人能放之位置)則優先擺放。

(3) 選磚塊從21號依序遞減選擇。(會做旋轉和翻轉)

策略架構:1.先讀取BlokusBoard.txt 和LeftPattern\_Color.txt。

2.判斷整張BlokusBoard.txt哪裡有角我可以放，順便存取敵人能放的角。

3.開始選擇磚塊，從LeftPattern\_Color.txt看這磚塊有無被使用，從編號21開始抓取，依序遞減選擇。

4.選出這磚塊其中一格對應第2步驟能放的角中，並將整塊磚塊放進去看可不可行，若不行，則選這磚塊其他格對應第2步驟能放格子中，以此類推。

5.若第4步驟無法放置，嘗試旋轉此磚塊，並重複執行第4步驟，若不行放置，再旋轉一次，以此類推，共旋轉4次，之後翻轉1次，再旋轉4次。

6. 會重複4到5步驟，且在第4步驟中會判斷放的位置有無擋住敵人能放的位置(利用步驟3存的位置)，擋越多優先排放，之後將這些資訊(選甚麼磚塊、磚塊大小、放的位置、擋人個數、翻轉幾次、旋轉幾次)存下來。

7.，最後將存儲的資訊用來寫檔，程式執行結束。

優點:瘋狂阻擋別人，讓人無法有效延伸自己棋步。

缺點:策略較少、較缺少防禦方式。

第二版:

策略:

擋越多優先排放，之後判斷放了這磚塊後多了多少路可以下，越多路則優先放置，最後判斷周圍有無鄰近敵人磚塊，鄰近越多敵人磚塊則優先排放。

優點:阻擋別人有利於讓自己有較多空間能放置。之後判斷下這步能接著延伸出多少空間，以免被別人擋死所有能放的位置。只有擋別人還是不夠，為了能想辦法鑽出敵人包圍住我的磚塊，以貼近別人下去實現這一策略，雖然無明顯相關，但透過貼著別人下，我所能延伸的活路也較可能鑽出敵人的阻擋，間接將包圍網擊碎。阻擋增加攻擊性，活路增加生存性，而貼近則會產生未知性帶來完全不同的局勢。

缺點:當遇到預判棋步的對手時可能無法應對，且可能會貼著牆下而這不是以人腦思考的最佳方式下法，磚塊編號1、2最實用的兩塊可能無法做出最大貢獻而只會放在毫無意義的地方。

第1版與第2版差異:雖然按照上述的邏輯介紹感覺2版比1版強，且在測試時不管先後手都是2版贏1版，但丟在平台上時，1版排名約為10，然而2版排名約為27(應該還會更低)，勝率甚至只達到28%。可能是1版讓電腦自己排的優先順序優於我自己想的2版策略，雖然2版設計的時間遠大於1版，但經過反覆對比後仍決定交出1版的程式當作最後結果。

Hw3和Hw4差異:Hw3中只能從左而右再從上到下，之後從編號21依序遞減選擇下的位置，毫無策略，Hw4則加了1種策略，阻擋增加攻擊性，使別人無法有效延伸自己磚塊面積。