AI project report

組名 : 嘻嘻羊 (Smile Sheep)

組員: 9817059李揚 、9817074吳晞浩、9817088李懷哲

我們的主要是利用min-max search演算法，搭配一個evaluate function作為我們的程式架構。

我們這組認為，這次project最複雜且最重要的部分就是架設evaluate function，並找出一個最好的。如何找出最好的呢?我們這組的大致流程如下(以先手為例): 1.先想出幾個可能會用到的判斷含式(如對手(自己)有幾子、對手(自己)有幾個活二、雙方子的質心、以幾質心密度等等)🡪2.經過自己組員內互下累積經驗🡪3.就由這些經驗去對這些函數做線性組合產生14組各有不同特色的AI🡪4.根據我們互相下子的經驗，認為先手只要考慮周全基本上是立於不敗之地的。所以我們認為我們的先手AI必須同時打敗這14個AI並且經過人為測試後也是得勝，為此我們自己撰寫可以讓AI打完這14組AI並顯示共贏多少子的程式，以此標準架構先手AI🡪贏過這14組AI後，再用認為測試，若通過就會變成我們候選人🡪最後在從眾多候選人中討論出最好的那一個。

先手基本上是不容易輸的，只要抱持著和Greedy的下法即可(凡是有三就先下，那不然就下活二…)，到最後，很容易地先手的子就可以形成一個正方形，或是L型的聚集情況，根據我們的研究，這樣子就可以確定贏棋了。

不過經過幾百局的測試後，我們發現下面兩招破解此Greedy演算法的後手下法。以下是此後手下法的會造成的I字棋型，以及其手序。 利用此法可以讓先手不再這麼有利，此兩棋型都給後手很高贏的機率，尤其是棋型二。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  | X |  |  |
|  |  | O | O | O |  |
|  |  | X | O | X |  |
|  |  | O | O | O | X |
|  |  |  |  |  |  |

棋型一

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  | 6 |  |  |
|  |  | 1 | 3 | 5(eat4) |  |
|  |  | 2,8 | 4,7 | 6,12 |  |
|  |  | 13(eat6) | 9(eat2) | 11 | 10 |
|  |  |  |  |  |  |

棋型一手序

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  | X |  | X |  |
|  |  | O | X | O |  |
|  |  | O | O | O |  |
|  |  | O | X | O |  |
|  |  |  |  |  |  |

棋型二

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  | 14 |  | 8 |  |
|  |  | 2,11 | 4,10,16 | 7 |  |
|  |  | 1 | 3 | 5(eat2) |  |
|  |  | 13(eat4) | 6,12 | 9(eat4) |  |
|  |  |  |  |  |  |

棋型二手序

比起其他的狀況，此I字形的下法對後手來講，雖然犧牲了一些自己的攻擊機會，但卻讓先手自己把自己給鎖死。

根據我們做得實驗，這樣子先手還是會贏，但先手必須算得十分準確才可以。也可以這麼說，這種棋型是所有Case裡最複雜的一個。我們曾經做過這樣的測試，我們用我們之前寫得程式跟我們自己下，然後我們是先用一個棋盤擺好下過所有的可能後再放子，在這樣的情形下，我們的AI並無法贏得了我們。

既然沒辦法很順利得Handle這種情形，我們就讓我們AI避免這樣的狀態，這也是我們先手把握的一個很重要的原則。

後手原則:

我們後手的原則則較為簡單，利用之前做的眾多的先手AI全部比過，選擇積分最高的(贏最多輸最少的)。為什麼沒盡量下成I字形呢?主要是因為我們發現這是一種不是慘敗就是險勝的下法，而根據我們的測試，我們後手在考慮這種狀況時還是會有點小差錯，故我們決定不做這個冒險。

現在來講解一下我們Evaluate function有考慮到的變因:

1. 對手子數和自己子數
2. 質心密度

把自己的質心密度算出來(把每顆子對質心距離做平均)，我們認為先手下密一點會較佳，所以在做evaluate function時，會考慮這一點。

1. 質心距離

雙方質心的距離，後手的時候我們認為質心距離下越近越好，這樣可以去擋住先手，並增加棋面的複雜度，從亂中取勝。先手則是越遠越好，降低複雜度，讓勝負更加明確，增加自己的贏面。

1. 活二
2. 牆面

有三的話盡量下三，造成自己有很多密集的牆面。

2,3,4,5都有考慮到對手的因素放進evaluate function