人工智慧 Project1

0856096 江陳建鏵

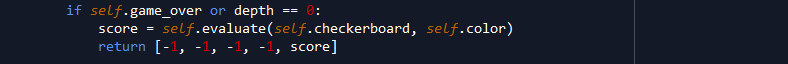
1. **用甚麼方法**

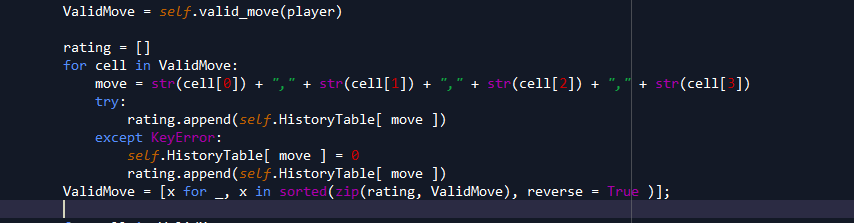
主要：MiniMax

優化：Alpha-Beta pruning + 使用dictionary紀錄最後一步

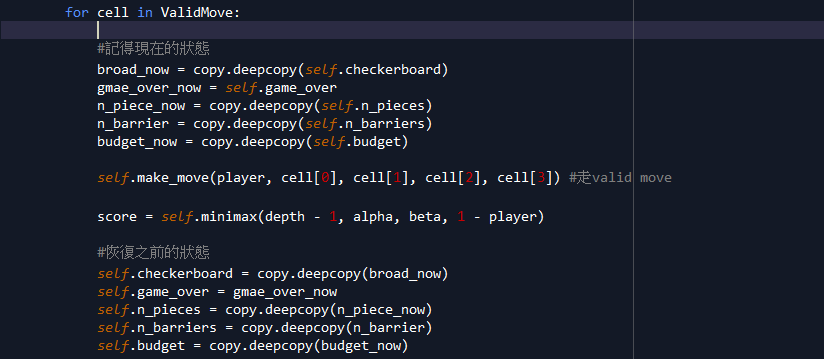
1. **如何寫的**

MiniMax終止遞迴的條件: 如果遊戲結束或是樹的深度等於0的時候，回傳evaluation value

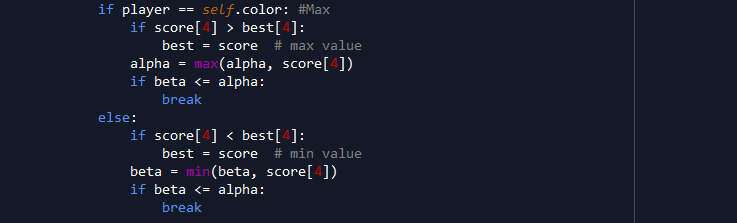


找到目前棋盤上自己的棋子可以走的步伐，並依照History Table的值將所有可以走的步伐排序(遞減)，這個方法是我去網路查要怎麼優化alpha-beta minimax查到的

走出那些步伐後遞迴找下一步另一方會怎麼走，在遞迴之前會先記錄目前的狀態，遞迴後再恢復狀態



若遞迴有傳值回來後就執行alpha-beta pruning，看可不可以不用再展開其他的child，以節省時間。



1. **我的發現**

在做這個作業的時候，我參考網路上的教學影片很快地就可以寫出minimax還有alpha-beta pruning的演算法，花比較多的時間都在優化AI整體的速度跟思考evaluation function的設計。

優化AI速度: 因為每一顆棋子的每一條直線最多有8個位子可以走，這會導致search tree展開太多的children，所以我簡化成每一條直線只能走棋子上下左右前一步跟上下左右最遠的地方，因此從最多24種可能縮減到8種可能，雖然這會減少AI的聰明程度但可以讓search tree更深。

Evaluation function設計: 我一開始用棋盤上有多少個敵人的棋子還有多少自己的棋子，然後在跟AI玩的過程中發現我們可以給不同的權重，像是旗子那條column還有旗子左右兩邊的column的權重給得比敲掉磚塊的權重還高，代表棋子走到中間那三條column(3, 4, 5)比敲掉磚塊還重要。我花很多時間在尋找棋子走到甚麼地方或做甚麼事情會比較強還有給不同的權重會造成甚麼結果。我在想這些參數可能就是機器學習在train的參數吧。

1. **如何執行**

* 需要的package

numpy, math, copy, time

* 打開command line並輸入” python 0856096.py” 即可執行