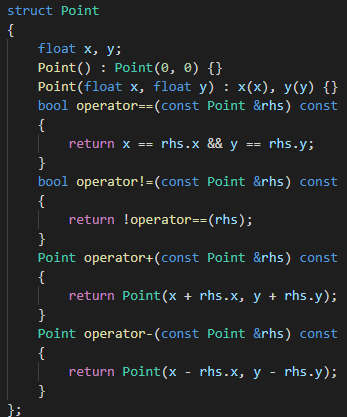
MiniProject3

在設計自己的AI之前，首先要先對黑白棋規則有一定的了解，隨後在瀏覽過一次main code之後發現裡面寫的Point class和OthelloBoard class已經為我準備了一個很好的工具供我在設計AI時使用。



一張含有 文字 的圖片

自動產生的描述

而其中最重要的莫屬於這個function

一張含有 文字 的圖片

自動產生的描述

這個function可以做翻棋子和更新棋盤，幾乎做完了下棋所需要的全部事情，所以我只需要再寫兩個建構子使得設計AI的時候可以實時更新棋盤屬性

一張含有 文字 的圖片

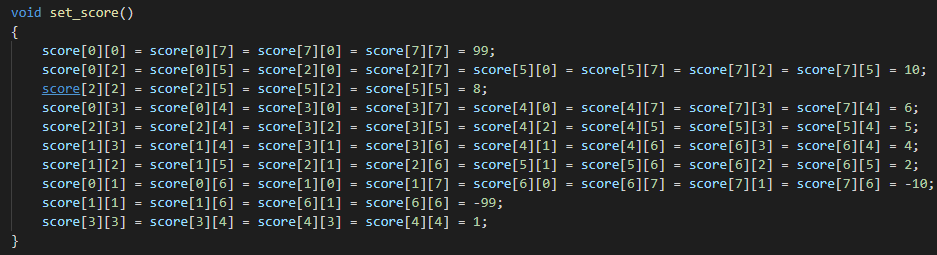
自動產生的描述

隨後以player\_random的code做基礎去修改，寫入和寫出文件已經做好，可以直接開始設計AI所需要的function。首先要實作的便是MinMax再加上alpha-beta-pruning，根據ppt所給出的psudocode，我只需要把內容修改上去並且設定搜尋深度即可一張含有 文字 的圖片

自動產生的描述

在寫完alpha-beta-pruning後便要來實作計算value的function才可以使這個函式可以完整運作。根據網上查找到的資料一張含有 白色 的圖片

自動產生的描述

我將棋盤格給定分數好讓AI計算每一步的收益，而分數高低則透過黑白棋的一些通用法則來設定，例如：角落為穩定子很重要，C-square為角落相鄰的兩格危險性較高不建議先下，X-square為角落斜相鄰的格危險性極高不能先下，除了這兩個square之外的邊緣優先度較高（爬邊）。所以得出以下棋盤分數

但單單給棋盤設置分值不足以應付千變萬化的棋盤，所以在網上查找黑白棋攻略的時候便又學到了幾手下棋方法可以提高獲勝幾率——行動力

一張含有 文字 的圖片

自動產生的描述

如果能限制對方能下點的數量並且提高我方自由度，那麼棋盤的局面會比較明朗。

而透過這些策略，我的AI已經基本可以打敗baseline前4，但最難的baseline5還需要一點小技巧。此時我便再添加一個角對角佔領策略一張含有 文字 的圖片

自動產生的描述

如果先佔領了角落，那麼其鄰近的X-square點會從危險點變成一個重要的戰略點，可以鞏固邊角使得爬邊戰術可以收益更大。

在做完角對角佔領之後，便可以輕鬆的擊敗baseline5，此次的project也告一段落。