Data Structure Project 1 – Tetris

107062318 李俊逸

1. Project Description
   * Program Flow Chart

NO

END?

NO

YES

YES

* + 1-2) Detailed Description

**概述：**

用一個整數二維陣列及多個能夠修改此二維陣列的function，輔以一些判斷式及輸出二維陣列的functuon，完成此次實做。

**詳述：**

輸入有效的row跟column，開啟一個二維陣列，稱之為map，將元素初始化為0。將代號所指之方塊寫入一個string，如果是End的話則直接印出執行結果；若不是，則判斷由哪一個function做將方塊填入的動作。呼叫function會傳入左下角參考點應該落在哪條column以及整張map，以便對map做修改。

Function判斷方塊放在哪的方法是：用for迴圈從地圖頂端的上面幾個row往下看(有在地圖頂端另開了一些空間(int extend就是多開幾行的意思)，以便處理消除後落下方塊的case)，觀察方塊底端的下一個位置是否已經有方塊佔據了，若有則把方塊放在那個地方，若沒有就繼續往下找位置放，直到跑到地圖的最底端都沒有阻礙的話，就把方塊放在地圖的最底端。

Function放好方塊之後，檢查是否有需要消除的row，若有的話就呼叫eliminate\_line。 eliminate\_line這個function會幫我從map頂端把全是1的row消除，row以上的方塊掉下來。結束這個function後，會繼續檢查相同的row是否有全是1的情況，若沒有，再繼續往下檢查。

該消的row都消乾淨之後，檢查map以上一個row有沒有沒消掉的方塊，有的話就輸出地圖的結果，並結束程式。沒有的話就繼續吃下一個block代號，重複做放方塊、消行、檢查是否遊戲結束。

1. Test case Design

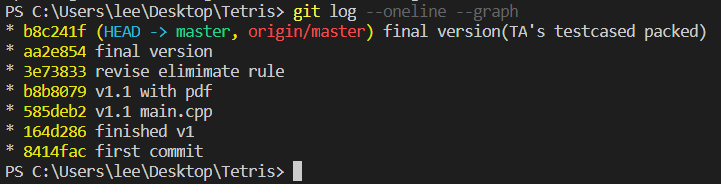
Row = 1, Col = 1, (I1 1)999個, End.

* 2-1) Detailed Description of the Test case

我覺得Project1沒辦法出刁鑽的測資。原因之一是invalid case不是由程式判斷，而是由出測資的人判斷的。

但有個較不直觀的點值得測試，也就是消除一列，除了要把上面的列放下來外，還要再次檢查同一列是否是滿的，否則就錯了，故我用row,col 都等於1，加上999個I1 1來測試這個問題。999是看會不會有些人的程式有沒消乾淨的可能，整個遊戲畫面以上的部分若寫得不夠大(Ex只有兩列)的話，跑我的測資可能會出錯。

1. Git log紀錄



Github網址：<https://github.com/karta1502545/Tetris>