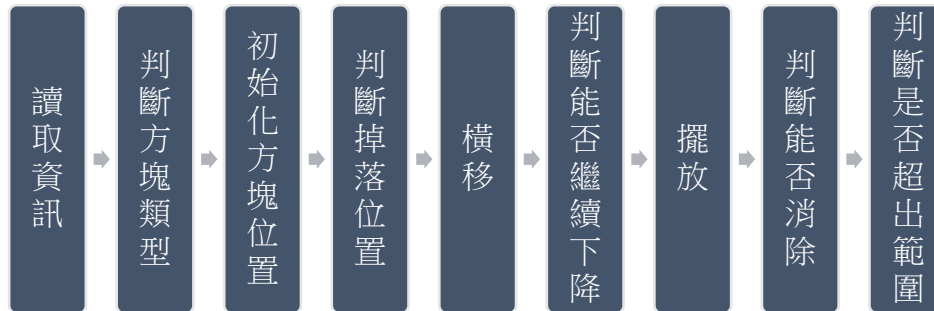


1) Project Description

1-1) Program Flow Chart



1-2) Detailed Description

0. 所有 matrix 裡的東西都是以 row-major 的方式記錄
1. 讀取資訊
 - ◆ 利用 argc argv 讀取 testcase 裡的資料，並依資訊建立一個大小相同的 matrix，做為整個遊戲的範圍
2. 判斷方塊類型
 - ◆ 根據不同字母、數字判斷方塊類型，並建立出一個形狀相同的 matrix，作為掉落的方塊
3. 初始化方塊位置
 - ◆ 初始化方塊掉落的 column，並判斷是否為 invalid (超出邊界)
 - ◆ 預設方塊最下層的最初的 row 為目前遊戲疊到最高的 row+1
4. 判斷掉落位置
 - ◆ 分別檢查掉落方塊裡四個位置分別能夠下降的格數 (從 top row 開始往下找有沒有和它同 column 的)，並取最短的那個距離，就等於方塊最後會下降的 row 數目
 - ◆ 這時還先不會把它加入主要的遊戲 matrix 裡
5. 橫移
 - ◆ 依照讀取進來的資訊對掉落方塊裡的每個位置進行橫移
 - ◆ 檢查是否為 invalid (判斷遊戲 matrix 裡那個位置是否已經有東西了)
6. 判斷能否繼續降落
 - ◆ 一樣是取掉落方塊裡每格能下降的格數的最小值
7. 擺放
 - ◆ 依照步驟 4 裡的掉落位置加上步驟 6 的降落格數，最後能確定方塊裡每個位置應該在哪一個 row 和 column
 - ◆ 擺放完畢後要更新現在遊戲裡的 top row
8. 判斷能否消除
 - ◆ 以判斷每個 row 裡面有東西的數量是否等於遊戲最初給定的 column 來判斷是否可以進行消除

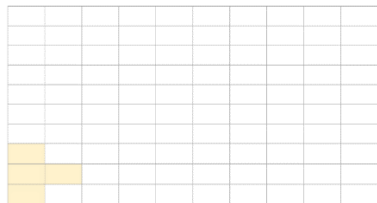
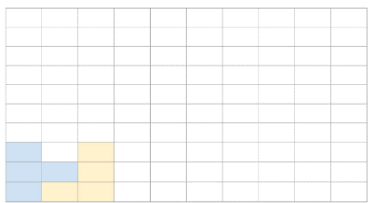
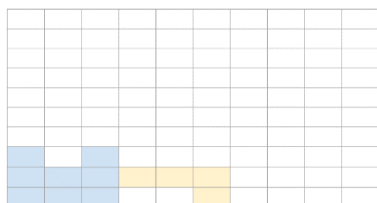
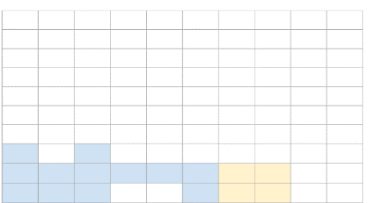
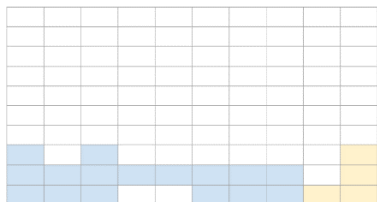
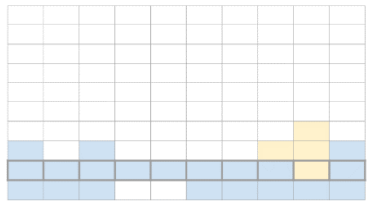
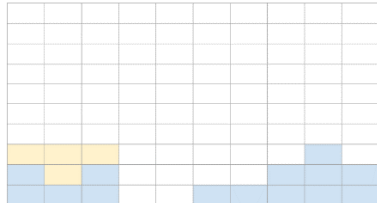
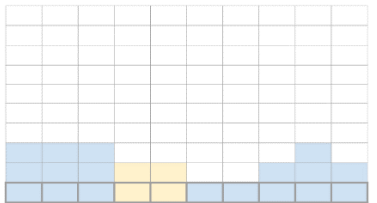
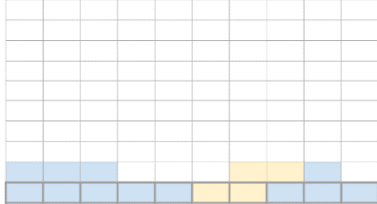
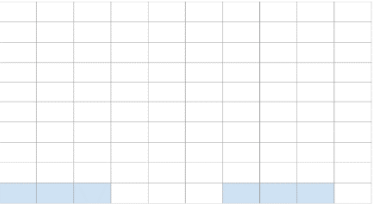
- ◆ 由上往下判斷，如果有可以消除的 row，就把它 clear，再把疊在它上面的所有 row 都往下一移一行，top row-1

9. 判斷是否超出範圍

- ◆ 看 top row 是否超出最初給定的 row 數目可以判斷是否超出範圍

2) Test case Design

2-1) Detailed Description of the Test case

testcase	result	
10 10	 step 1	 step 2
T4 1 0		
J1 3 -1	 step 3	 step 4
J4 3 1		
O 7 0	 step 5	 step 6
J1 9 0		
T2 7 1	 step 7	 step 8
T1 1 0		
O 2 2	 step 9	 End
S1 6 0		
End		