

CPP1002 Coding Assignment 3

彩磚問題

電子檔 2011/03/30 09:00 前上傳至 moodle
書面檔 2011/03/30 09:15 前於課堂繳交

事前準備

本作業跟一個小遊戲Color Tiles相關，其玩法相當簡單，請先花十分鐘到遊戲網站體驗一下 (http://en.gamesaijen.com/game/color_tiles/)，記住！十分鐘，不要逗留過久。

該遊戲的玩法如下：在各空格的上下左右四個方位中，最近且非空白的方塊稱為其neighboring tiles，玩家得點選某空格以消除其四個neighboring tiles中相同顏色的方塊。原遊戲以2分鐘內消除的方塊數來定勝負，若點錯空格（即點選到其neighboring tile沒有重複顏色的空格），則時間會被扣掉10秒鐘，所以不能亂點。

問題描述

本作業旨在設計一個程式來玩這個遊戲，該程式將忽略原有的2分鐘時間限制，而是以玩到沒有方塊可再被消掉為目標。欲達到沒有方塊可再被消掉的作法有很多種，其中，測試所有可能消掉方塊的暴力 (brute force) 法即為一種直覺的作法。簡言之，此遊戲之暴力法乃判斷所有空格的neighboring tiles是否有相同的色塊；若有，則點選該方塊，消掉其上下左右同顏色的neighboring tiles。重複此程序直至沒有任何空格可再被點選，亦即達到所有該消掉的方塊皆已被消掉的結果。

在開始撰寫作業前，請參考題目給的 hw3_sample.cpp 檔案。本題提供了一組設計好的方格題目 (map.txt)，其中包含了空格 (0) 與九種顏色 (1~9) 的方塊。由於本作業可能應用到三個目前課程尚未詳細教授到的技巧：(a)讀取 map.txt 檔的資料，(b)使用動態二維陣列，以及(c)印出 map 內容。以下將簡略介紹此實作三技巧之部分程式碼，同學可先看過程式碼再來對照以下說明。

(a).讀取 map.txt 檔案(主檔為[10 × 20]例子，另有兩測試檔：map1.txt [5 × 5], map2.txt [25 × 15])

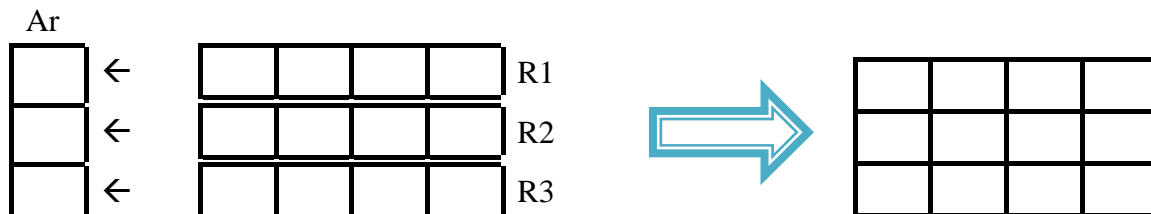
根據 map.txt 檔第一列定義的寬與高，動態配置了一個10 × 20的integer陣列，接下來就只要將這200格的顏色整數（將空格設為顏色0）一一讀入。讀取檔案時可先用 ifstream 開啟檔案，如程式碼第38行所示：ifstream inFile("map.txt"); 觀察 map.txt 檔可發現，每筆資料是用空白分隔開來，這時可利用 cin 讀取每一筆資料，因為 cin 會將兩個白字元(如空格、換行)間所夾的東西視為一筆資料，例如：1 2 3 這串字會被 cin 視為有三筆資料。將 cin 應用在迴圈當中，即可很容易地將資料讀取到二維陣列中，如程式碼第45行。

(b).動態陣列配置

依維基百科的定義，陣列(array)為一 systematic arrangement of objects，也就是對一群物件有系統的排列，這物件可以是 integer, double, string, ... 等等有特定結構的資料。以 integer 來說明，integer 的一維陣列就是由一系列 integers 組成，類似數學中的數列；而二維陣列可想像成由諸多的一維陣列疊起來組成，也就像是數學中的二維矩陣。

在 C++ 中宣告 integer 陣列最簡單的方式如下：例如要宣告一個含有三個 integers 的一維陣列可寫成 int array[] = {1, 2, 3}，或是要宣告一個 3 × 4 的二維陣列，可寫成 int array[3][4]，其概念可視為將 3 列含有 4 個 integer 的一維陣列疊起來。

陣列的定義雖然簡單易懂，但其大小必須事先知道，如此才能於宣告時同時定義，**無法**等待使用者輸入其大小（譬如n）之後再以 `int array[n]` 的方式來動態宣告之。然而，實務上我們卻常需要根據使用者輸入的數字來定義陣列大小，此時就有必要學會如何根據輸入的數字來動態地配置記憶體給陣列。以本作業為例，每個題目(map.txt)可視為一個顏色矩陣，各儲存格則存取其方塊顏色；當一開始所定義的維度不同時（譬如 5×5 、 10×20 或 25×15 等），程式應該要有辦法根據所讀到的維度來動態設定出 5×5 、 10×20 或 25×15 等二維陣列。而動態配置記憶體給二維陣列也有許多作法，以下我們提供一種較常見的作法：其概念是先宣告一個一維（column）指標（pointer）陣列Ar，然後在Ar陣列的每一格中又分別宣告一維（row）陣列R1~R3，如下圖所示：



動態陣列配置將應用到指標(pointer)，這也會在後續課程中詳細介紹，在此可先將 hw3_sample.cpp 檔案中第 41 行的程式碼 `int **map = new int*[map_height+1];` 視為如上頁所說的，先建立一個一維陣列 Ar，然後再以迴圈在 Ar 中別建立一維陣列 R1~R3，如程式碼第 45 行所示：`map[i] = new int[map_width+1];`。

(c).印出 map 內容

在 hw3_sample.cpp 程式碼中提供了兩個印出 map 內容的函式，分別為 `print_map(int**,int,int)` 與 `color_print_map(int**,int,int)`，後者與前者的差異在於者可以印出有顏色的方塊。然而，由於後者有利用到 `<windows.h>`，在非 windows 作業系統可能無法使用。因此，在非 windows 作業系統下，請刪除程式碼第 28, 34 與 77 到 89 行。此二函式之使用方式相同，只要呼叫此函式，給予 map 矩陣與 map 的高及寬，即可印出 map，例如：`print_map(map, map_height, map_width);`。

解題提示

要寫出一個程式來玩這遊戲，方法不只一種，在這提示最簡單的暴力法應該如何應用在此作業中，暴力法即測試所有可能的組合，在此則是判斷所有空格是否可被點選，我們可以依循以下步驟來完成：

Step1.找出一個空格的 neighboring tiles



(圖片引用自 http://en.gamesaien.com/game/color_tiles/)

我們可以先宣告一個 4×3 矩陣 NT 來儲存四個 neighboring tiles 的顏色與位置，以上圖為例，

座標(x,y)中 x 表示列，y 表示欄，以方格圖的最左上方塊為(1,1)，假設我們現在點了一空格(3,4)，我們要從這空格向四個方位尋找非空白的方塊，如向上 (Direction=(0,-1)) 應該可以找到褐色，向右 (Direction=(1,0)) 應該可以找到綠色。這個尋找的過程可以用一個 while 迴圈來做，例如要找左邊的 neighboring tile：

```
Direction ← left           //←為 assign 之意，可視為=; left=(-1,0)
Index ← (3,4) + Direction  //Index 移到 (2,4)
While (Index is not a space and margin of map not reached)
    Index ← Index + Direction //Index 將持續移到一個有色方塊或邊界
Save Index to NT
```

以上圖為何，在四個方向都探索後，NT 將存有四個 neighboring tiles 的顏色與位置，假設棕、紅、綠色分別以顏色代碼 1、2、3 表示，則 NT 如下表所示：

	Color	NT_Row	NT_Column
Up	1	1	4
Down	2	5	4
Left	1	3	1
Right	3	3	6

Step2.消掉重複顏色的 neighboring tiles

接下來可以另外宣告一個陣列 SUM 來統計 NT 陣列中各種顏色的數目，以上圖為例，在 NT 中有 2 棕、1 紅、1 綠，假設有 9 種顏色，各以 1~9 表示 (空格之顏色 0 不能消掉故不用累計)：

Color	1(棕)	2(紅)	3(綠)	4(藍)	5(紫)	6(灰)	7(黃)	8(橙)	9(黑)
Number	2	1	1	0	0	0	0	0	0

然後再以一個迴圈去掃描此 SUM 陣列，判斷如果有哪個顏色的個數是大於等於 2，則到 NT 中尋找有此顏色的色塊，將其在 map 位置上的顏色改成 0(表示空白)，動作如下所述：

```
For each color Clr in SUM
    If SUM[Clr]>=2 then
        For each row in NT
            If color of one neighboring tile is Clr then
                map[NT_Row][NT_Column] ← 0
```

Step3.嘗試所有可能

以上的 Step1 與 Step2，只是對於某一空格來找出其 neighboring tiles，並消掉同顏色的方塊，但暴力法要試過所有空格，故在以上步驟之外須加上迴圈，在所有空格都要做以上兩步驟的動作。如下所示：

```
For each row in map
    For each column in map
        If map[row][column] is a space, then
            Find neighboring tiles (Step1)
            Eliminate tiles which have duplicated color (Step2)
```

僅掃描過 map 中所有空格一次是不夠的，因為消掉一些方塊後，那一個空格的 NT 值會有更動，因此必須要反覆一輪(即掃過所有格子)又一輪地掃描 map 中所有的空格，直到某一輪的掃描中發現不再有任何方塊可被消掉才可停止。而這個持續輪迴的掃描動作，可在以上程式碼的外層再加上一層 while 來判斷，若是都沒有任何方塊可以被消掉時才停止該 while 迴圈。

最後完整的演算法虛擬碼(pseudo code)如下：

While still some tiles can be eliminated

For each row in map

For each column in map

If map[row][column] is a space, then

Find up neighboring tiles, and save to NT (Step1)

Find down neighboring tiles, and save to NT (Step1)

Find left neighboring tiles, and save to NT (Step1)

Find right neighboring tiles, and save to NT (Step1)

Count number of each color, and save to SUM (Step2)

Eliminate tiles which have duplicated color (Step2)

作業要求

請使用暴力法設計一C++程式處理map.txt，map1.txt及map2.txt三個map檔([下載相關檔案](#)：包括此三map檔、hw3_sample.cpp以及map產生器create_map.cpp[參考用])，書面報告中必須：

- 需使用預先設計好的函式 print_map() 印出程式處理各 map 檔的最後 map 圖(結果的 map 圖不一定所有色塊都可以被消掉)
- 印出程式針對各 map 檔所消掉的方塊個數

另外，此作業雖要求使用暴力法來解決這遊戲，但若你想到其他方法可以解決此問題，請在書面報告的後頭說明你的想法，助教將視方法的可行性、執行效率酌以加分(最多 20 分)，注意！請務必詳細清楚明瞭地說明你的想法。

***作業繳交應注意事項 (!!特別注意要記得標上你的組別編號!!)**

1. 作業需要繳交電子檔以及書面。
2. 電子檔請於作業繳交截止時間以前上傳至<http://moodle.ncku.edu.tw>
 - 2.1 請同學先建立一個資料夾，資料夾名稱為“**學號_組別_hw3**”，例如學號為 H12345678，組別為第一組，則資料夾名稱則為 **H12345678_g1_hw3**
 - 2.2 將程式檔案名稱存為“**hw3.cpp**”，並將此程式檔案存於上述設立之**學號_組別_hw3** 資料夾中
 - 2.3 最後將整個**學號_組別_hw3** 資料夾壓縮成zip檔(**學號_組別_hw3.zip**)，再上傳至moodle系統(! 注意!：**請勿**將cpp檔copy/paste至word檔而上傳之)
3. 書面作業請於 **2011/03/30 上課 5 分鐘內(09:15 前)**繳交至講台，其中需要註明程式是否能被編譯與執行、撰寫人、程式之目的、如何編譯及執行等資料(詳見http://ilin.iim.ncku.edu.tw/ilin/course/_CPP1002/programming.html)。