專題---推箱子

●專題簡介

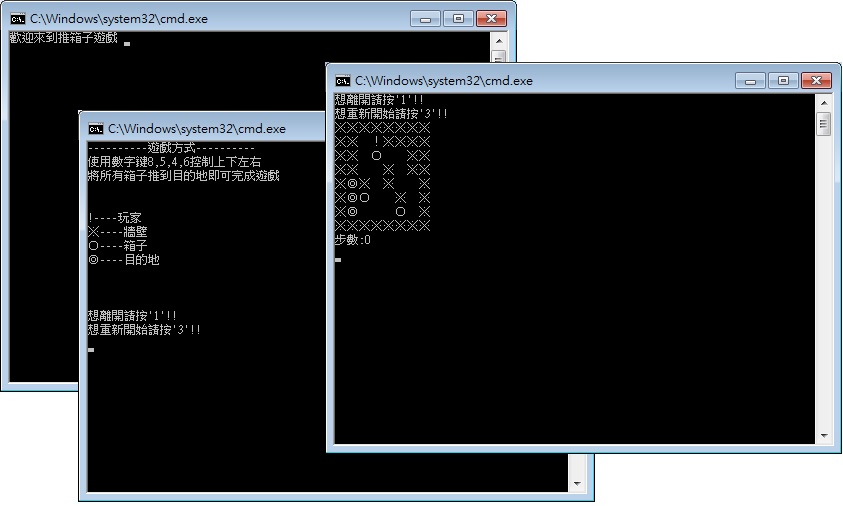
專題內容是實作一個推箱子的小遊戲，一開始有遊戲規則介紹的頁面，按下任意鍵即可開始遊戲，由數字鍵”8”，”5”，”4”，”6”來控制上下左右，玩家符號為”！”，需要把箱子”○”推到指定目標”◎”，而當箱子到達指定目標時會變成”●”，另外附有計算步數的功能，當所有箱子都到達目的地時，即結束遊戲，並顯示出走了多少步，若是在遊戲中發現失敗，可以按”3”重新開始遊戲，步數也重新開始計算，若是想離開，可以按下”1”，到了離開頁面，可以選擇重新開始或是直接離開遊戲。

●設計方式

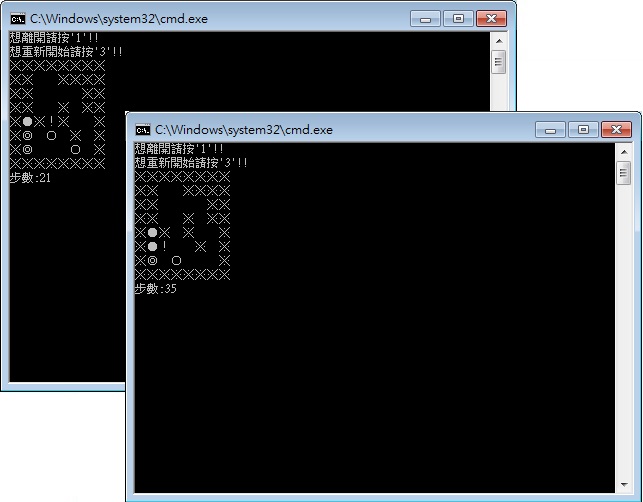
首先將需要使用到的文字敘述都先宣告在.data中，再來是地圖的宣告，由於推箱子有特定的地圖及解答，因此不能用隨機生成地圖的方式，所以參考了網路上的地圖，直接寫死在.data中，另外宣告了另外一個一模一樣的地圖，因為每次移動都直接改變地圖的資訊，因此第二個地圖是為了重新開始的時候可以初始化原本的地圖，地圖是用數字來存，然後用雙重迴圈掃描一次，看是數字幾，就印出相應的符號。再來是做上下左右的判斷，用cmp去比較輸入的值是4、5、6、8哪一個，並呼叫各自對應的PROC執行動作，最困難的地方在於判斷移動，因為首先要判斷旁邊是牆壁、道路、箱子、目的地、或是到達目的地的箱子這幾種可能，若是路或是牆壁即可直接判斷移動或是不動，但是若是箱子則需要再多判斷再旁邊一個是牆壁、道路、箱子、目的地、還是到達目的地的箱子，每個都有不同的對應動作，這部分是這個遊戲最重要的部分也是最需要思考地方，要是漏掉任何一個地方，遊戲就會無法執行。移動寫完之後就是要做結束的判定，因此看有幾個目的地，因為箱子移動到目的地之後會目的地會變成” 到達目的地的箱子”，這2個存在地圖中的數字是不同的，因此每到達一次，就把count目的地的值-1，直到0結束遊戲。做完之後發現走錯路就只能重新開始，因此加上重新開始以及離開遊戲的選項，重新開始就是把地2個地圖的值一個一個mov到第一個地圖，然後初始所有計算目的地、步數、玩家位置的值，即完成重新開始，離開則是跳到另一個畫面，再讓使用者輸入值，決定離開或是重新開始。

●設計方式

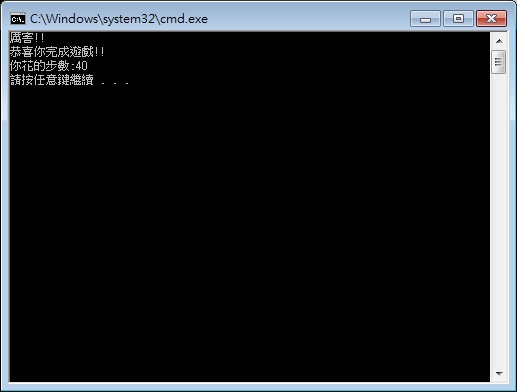
遊戲開始畫面



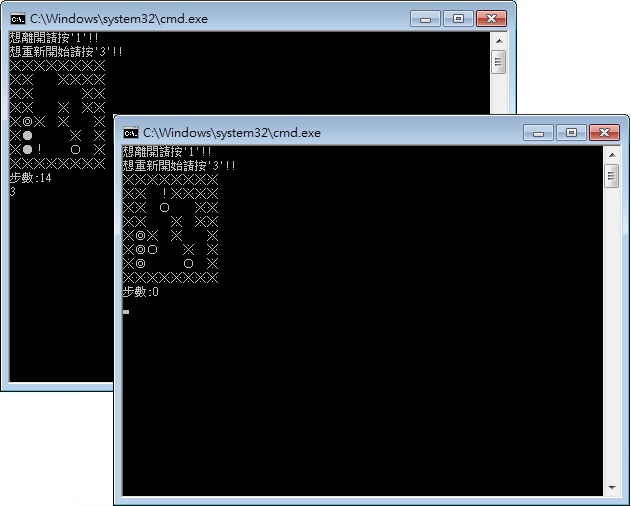
遊戲畫面



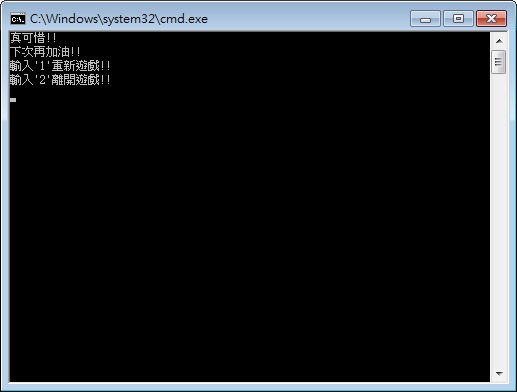
勝利畫面



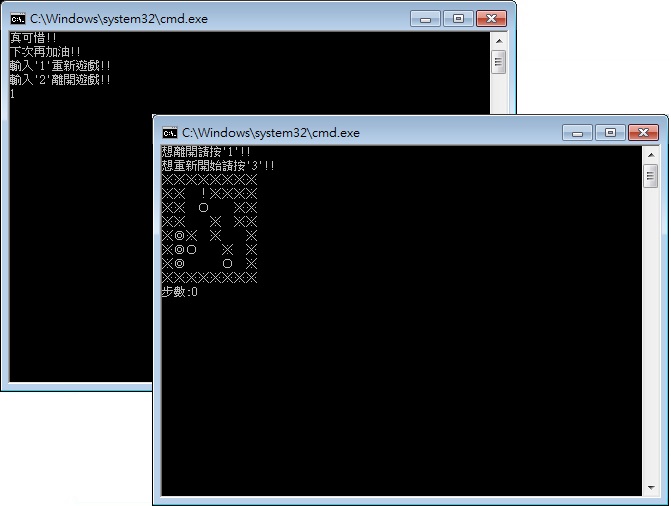
重新開始畫面



離開畫面



離開後重新開始畫面



●優缺點

●(缺)原本想說由於製作地圖是寫死的，其他的執行上都是一模一樣的，因此想說只是在.data中多一個宣告而已，但是之後認為還是有點單調，應該多訂其他地圖，並在一開始選單中加上選項。

●(缺)原本想用方向鍵控制，但是上網查了一下覺得執行上有點麻煩，也想用類似get()的方式讀取，但是最後也是沒有找到方法，因此只好用數字鍵，並需要多按enter才能移動。

●(缺)原本想用上一步的功能，但是這樣就需要判斷上一個動作是上下作左右哪一個才能做出返回的動作，因此就沒有加上這個功能，或是有其他方式判斷，那時候沒有想到所以就只能直接離開或是重新開始。

●(優)在遊戲中若是發現失敗了可以重新開始或是直接離開，離開後也有選單選擇是要離開或是再重新開始。

●心得

原本以為很簡單，想說只要做出地圖然後判斷旁邊那一格就可以了，但是實際做的時候發現，除了旁邊那一格之外，還有再旁邊一個也要判斷，這樣原本只有5種可能，就變成快20種可能要判斷了，尤其是一開始沒有想到在目的地的箱子，之後做完所有可能後，又發現另一個難題，就是在移動到目的地之後，要是離開的話，目的地就變成一般的路，因此還要多設一個變數，在踩到目的地的時候定為1，然後在每個移動的時候要先考慮是否站在目的地上，如果是就要在離開目的地的時候，將原本那格還原成目的地，這個部分花了我很多時間，一直產生bug。還有做if else的時候，常常沒有做好jump的設定，因此重複執行到同一個指令，所以要多思考幾次，才能知道有沒有重複jump到。原本的符號是”\*”、”^”、”#”、”&”…等的符號，要用writechar來印出，但是非常不容易辨識出各個特徵，因此才改成將符號(“●”、”○”、”※”)存在.data中，再用writestring輸出。