

HW2 README

本程式以物件導向方式設計，以下將解釋各個物件的功能。

- **class tile**

- 紀錄各個磁磚的X,Y座標，以及被蟑螂走過的次數
- 建構式：輸入X,Y座標，並將次數歸零。
- `int Count()` 函式回傳被蟑螂走過的次數。
- `int operator ++(int)` 重載運算子++，方便計算蟑螂走過的次數。

- **class Bug**

- 行走各塊磁磚，並記錄自己所在位置的X,Y座標以及行走的步數。
- 建構式：輸入開始的X,Y座標，並將步數歸零，以及紀錄題目給定的 `imove, jmove`。
- `int Random()` 函式回傳亂數介於0~8之間(不包含8)以便自己行走。
- `int XPosition()` 函式回傳自己的X座標。
- `int YPosition()` 函式回傳自己的Y座標。
- `int Steps()` 函式回傳走過的步數。
- `void Walk(vector<tile *>&square)` 傳入的是可以行走的磁磚的pointer的陣列，在此函數中會由亂數產生下一步的座標，並判斷傳入的磁磚是否可以走這一步如果是則走這一步，並將這塊磁磚++，不是則再產生下一組座標。

- **class Game**

- 紀錄磁磚的Row與Col，並產生遊戲地圖，以及進行遊戲。
- 建構式：輸入Col與Row，以此產生Row*Col個tile物件，並給予座標。(Row*Col個tile物件是以二維陣列方式儲存)
- `bool IsOver()` Scan過所有的磁磚，如果有磁磚是0則回傳False 否則回傳True
- `void Print()` 會把所有的磁磚裡的Count輸出
- `void Play(Bug bug)` 需要傳入的參數是要玩這個遊戲地圖的bug。在函數裡，會根據bug目前的位置將bug可以行走的磁磚的pointer傳入bug.Walk中。

main :

- 宣告Game物件，並給予遊戲區域的Row和Col。

- 宣告Bug物件，並給予bug的起點座標。
- `game.Play(bug)`開始進行遊戲。