一・主題介紹

《小精靈》（Pac-Man）是電子遊戲歷史上的經典街機遊戲，由 Namco 公司的岩谷徹設計並由 Midway Games 在 1980 年發行。缺了一角的薄餅是岩谷徹創作此遊戲的靈感來源。

遊戲的目的就是控制遊戲的主角小精靈吃掉藏在迷宮內所有的豆子，並且不能被鬼魂抓到。

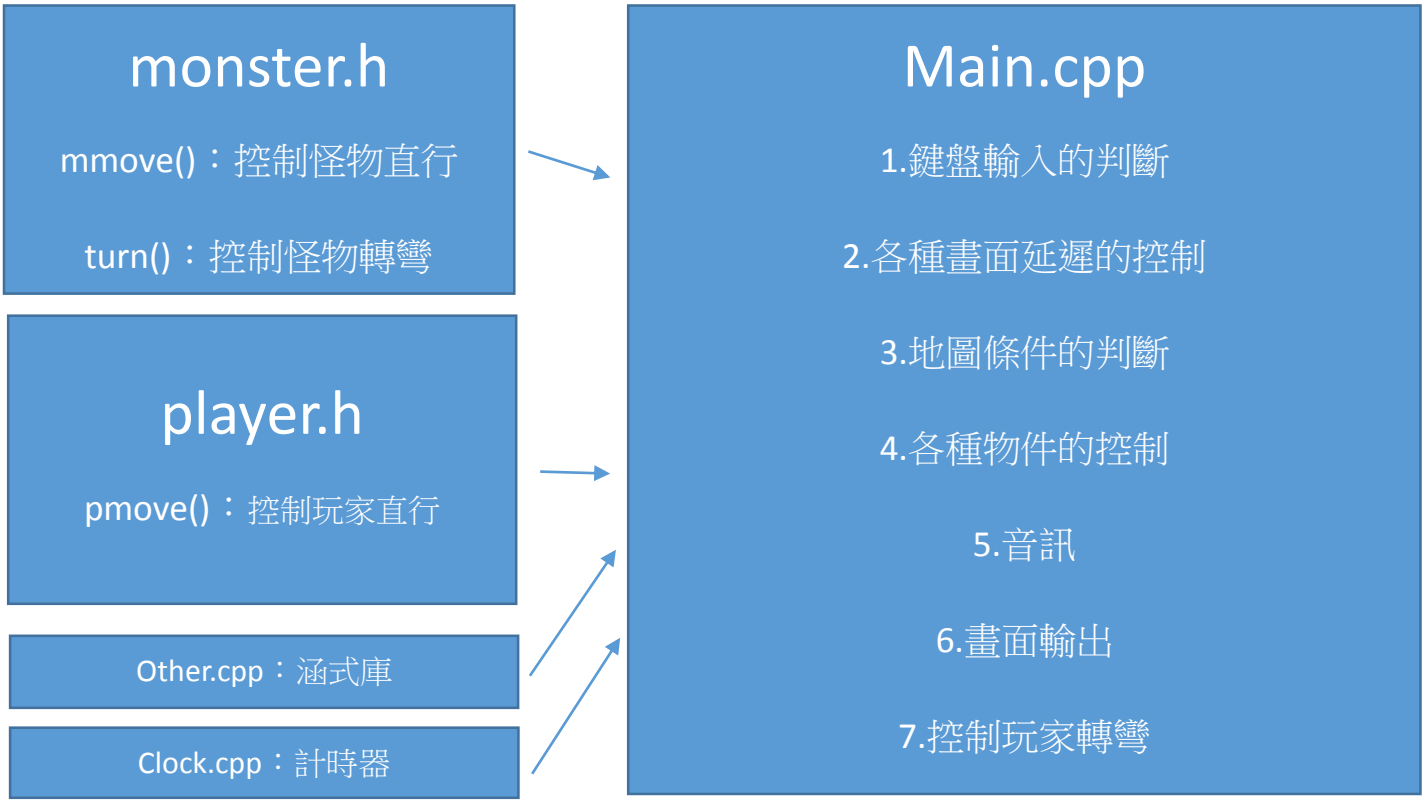
迷宮的四個角落有大的閃爍點稱為大力丸，提供小精靈一小段時間，可以反過來吃掉鬼魂。鬼魂在這段時間內會變成深藍色，會往反方向逃逸，而且比平常的移動速度慢。有趣的是，當鬼魂被吃掉時，它的眼睛還在，會飛回鬼屋，並再生而回復正常的顏色。當深藍色的鬼魂反白閃動時，表示大力丸的效力即將消失，而隨著遊戲的進展，大力丸的有效時間會越來越短。在遊戲的後段，大力丸被吃掉時，鬼魂不會改變顏色，但仍會往相反方向逃。

本 project 即是依照上述原則寫成並加以衍生之 Pac-man。

二・特色

1.完整精密的重現，多種狀況的考慮
2.無重大 BUG
3.可以讓程式設計者或是玩家自行設定敵人數
4.建立於 3 特色，衍生出許多新模式

三・設計架構



四・重點設計討論

一・玩家設計

玩家有自己的方向與 xy 座標，當 main 判斷玩家可以直行時 pmove()就會被執行，將玩家移向自己方向的下一個位置，但是不能繼續直行時，玩家就會停在轉彎處等待使用者的輸入，假想有三種情形：

1.玩家在轉彎處等待輸入，這時玩家鍵入一個方向

→判斷該方向是否可行，如可行則改變方向元素並往該方向移動一步。

2.玩家在直行時輸入同方向之按鍵。

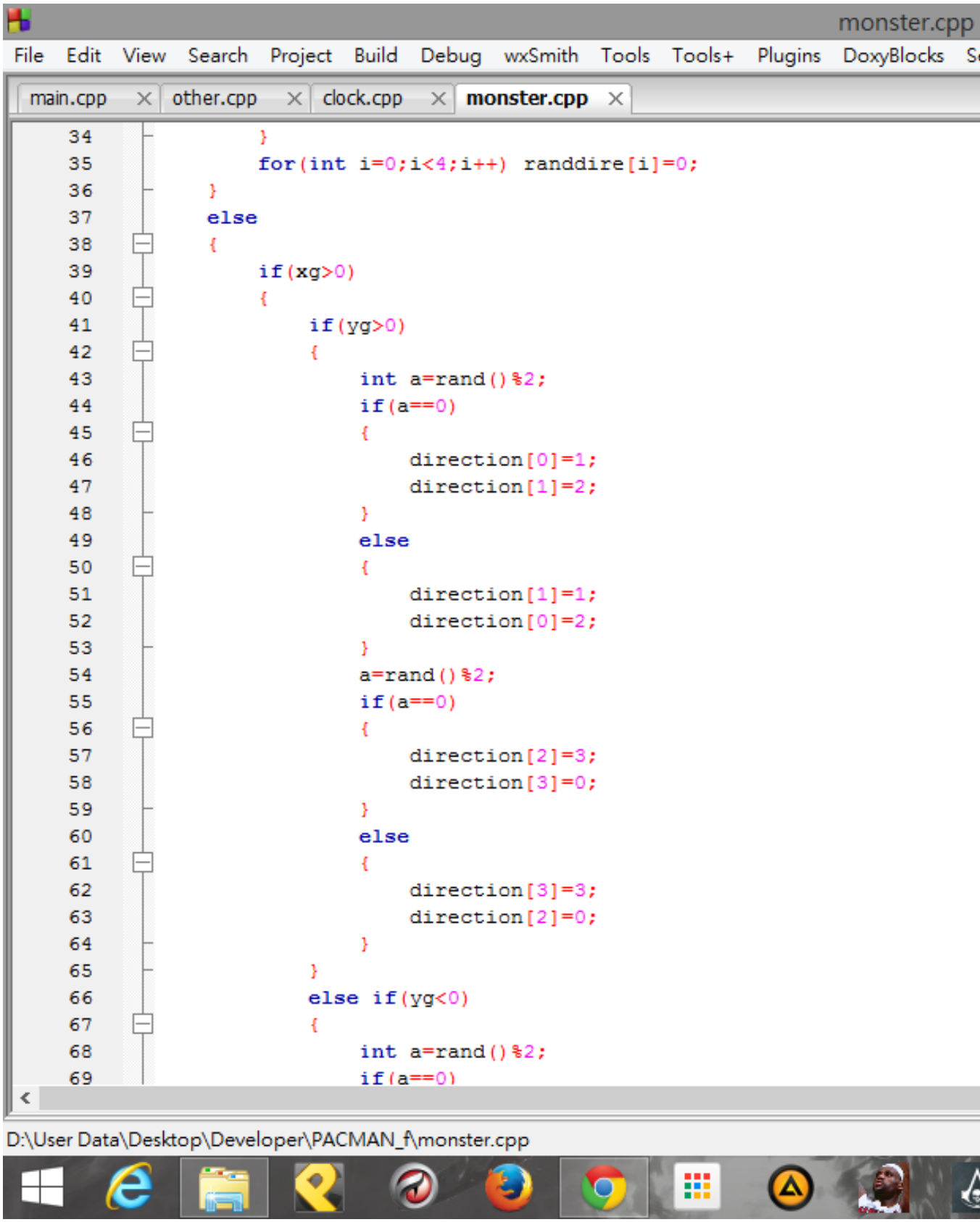
→因為設計為同時改變方向元素並往該方向移動一步，如果玩家一直鍵入該方向則會產生加速現象，避免這種情況發生，則不讓玩家在同方向時有任何改變。

3.玩家在直行時輸入反方向之按鍵

→特例，可行，立即改變方向元素並往該方向移動一步。

二・怪物設計

玩家有自己的方向與 xy 座標，當 main 判斷玩家可以直行時 mmove()就會被執行，將玩家移向自己方向的下一個位置，但是不能繼續直行時，turn()就會執行，turn()的主要設計為：有一陣列 direction[4]儲存的是怪物方向的志願，當玩家不在怪物判定區之內時，以 rand()方式將 0(上)、1(右)、2(下)、3(左)，存入該陣列，在 main 中，會從 direction 第一個元素開始判斷，假設該方向下一個位置為不能走之牆壁時，則再往 direction 下一個元素判斷，但是當玩家出現在怪物的判定區中，假設為第一象限，則 0 與 1 將被優先填入 direction[0]與 direction[1]中，填入方式也是 rand()，剩下兩個也已 rand()方式填入，討論四個象限。


monster.cpp
File Edit View Search Project Build Debug wxSmith Tools Tools+ Plugins DoxyBlocks
main.cpp
other.cpp
clock.cpp
monster.cpp
34 }
35 for(int i=0;i<4;i++) randdire[i]=0;
36 }
37 else
38 {
39 if(xg>0)
40 {
41 if(yg>0)
42 {
43 int a=rand()%2;
44 if(a==0)
45 {
46 direction[0]=1;
47 direction[1]=2;
48 }
49 else
50 {
51 direction[1]=1;
52 direction[0]=2;
53 }
54 a=rand()%2;
55 if(a==0)
56 {
57 direction[2]=3;
58 direction[3]=0;
59 }
60 else
61 {
62 direction[3]=3;
63 direction[2]=0;
64 }
65 }
66 else if(yg<0)
67 {
68 int a=rand()%2;
69 if(a==0)
D:\User Data\Desktop\Developer\PACMAN_f\monster.cpp

三・延遲設計

在 survivla 模式中有個相當特殊的設計，當一個大力丸被吃掉後，過了一段時間大力丸就會在原地自動生成，這邊採用的是一種環形結構的設計，地圖中有 10 個大力丸，則建立一個 surv[10][2]陣列，存放的是大力丸的座標，初始值皆為 0。如果 surv 裡的元素為 0，則代表沒有大力丸需要生成，在系統運行中，每隔一段時間會自動 load 不同的 surv 一為元素，舉例來說，系統一開始 load surv[0]，過了一段時間後，系統便會卸載 surv[0]轉而 load surv[1]…卸載 surv[10]轉而卸載 surv[0]如此環形的下去，又假設現在 load surv[0]，當玩家吃到大力丸時系統便會將 surv[0][0]、surv[0][1]分別存入大力丸的 xy 值，並卸載 surv[0] load surv[1]，而當 surv[1]裡有元素不是 0 時，代表該陣列元素中存放的 xy 值所指的位置上的大力丸在前一輪環狀循環時曾經被吃掉且目前為消除狀態，因此利用 xy 值讓大力丸重新產生並將 sur[1]重新賦值為{0.0}。

四・自由控制怪物數量

由於怪物係以 class 方式寫成，故使用陣列建立許多怪物並用 for 針對每個怪物做判斷。

五・心得感想

能把一個可以跑的程式做出來的感覺真的很棒，從提案，實驗實作一直到完全做完甚至 demo 呈現於大家的面前，感覺很有開公司推出新產品的感覺，從開始做時就已經讓人覺得躍躍欲試了，時間關係沒辦法使用圖形化 library，因此將來的願景是利用 openGL，以這次的小精靈為 base 做出「3D 板可翻轉的小精靈」，這已經變成我階段大學生涯的小目標。