蛇蛇世界

Group30 Final Project Document

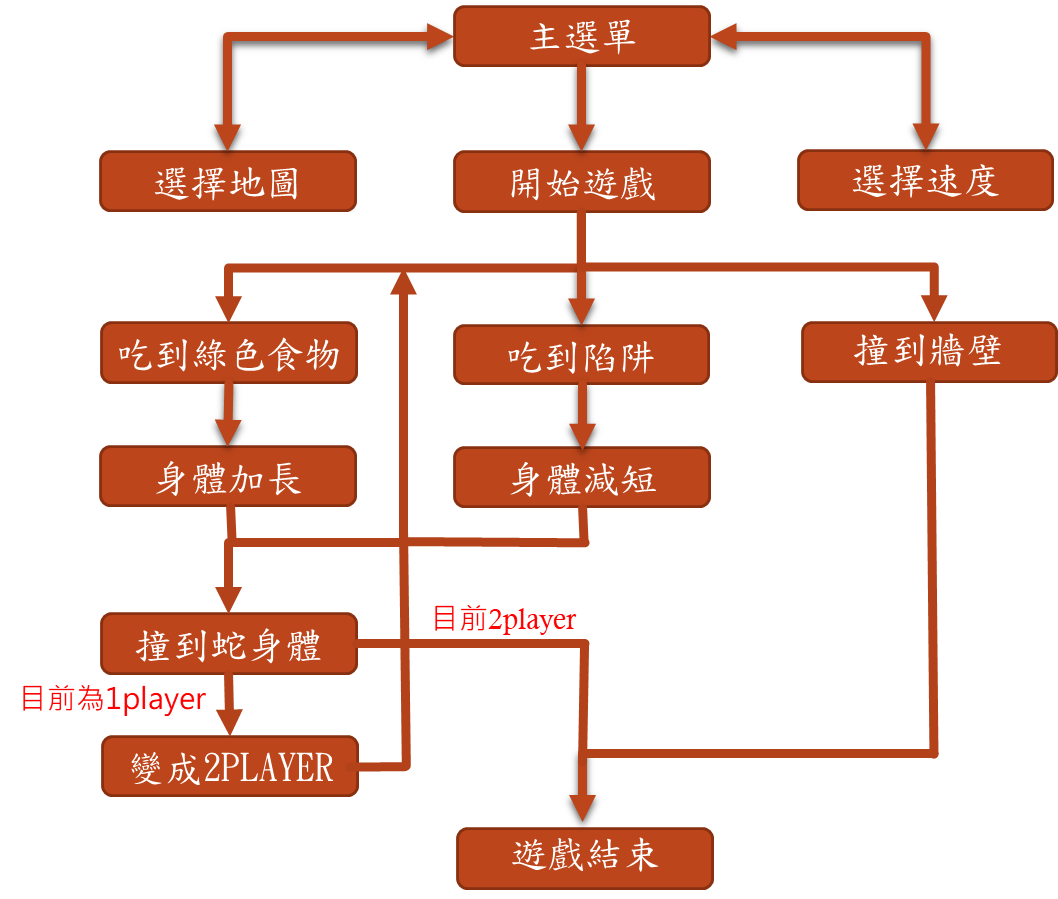
資工2A 陳皇宇 107502540

資工2A 何育賢 107502513

資工2B 王昱承 107402018

**簡要**

Group30致敬組合語言在程式語言發展中不可或缺的地位，感謝前人在此語言的努力才有現今如此方便、直覺與可讀性高的高階語言造福後人，Group30在final project的遊戲中目標帶出前人種樹後人乘涼的飲水思源概念，讓玩家中可以有不同的感受使我們的遊戲不僅僅是遊戲。

**遊戲流程**

**操作說明**

1. 進入遊戲前可以先選擇速度及地圖
2. 1player 透過 WASD 控制 上左下右
3. 2player 透過 方向箭頭控制相對應方向

**遊戲特點與功能**

Group30蛇蛇世界功能如下(橘色為特殊功能)：

* 選擇地圖
* 選擇蛇移動速度
* 吃到綠色(food)蛇身體加長
* 吃到紅色(trap)蛇身體縮短
* 若當前為1player撞到自己(需要至少保留兩截，因為要頭+尾巴才能組成一條蛇)則會斷為兩節，斷掉的部分會變成新的一條蛇，轉為2player(最多就2player)，原本尾巴會變為2player的頭
* 撞到牆Game Over
* 若當前已經是2player狀態，則無論是1p or 2p撞到自己或對方也是Game Over

**Code 介紹及實作解說**

**函式庫**

Irvine32.inc

**使用到的內建function**

1.Clrscr

2.WriteString

3.WriteChar

4.ReadChar

5.SetTextColor

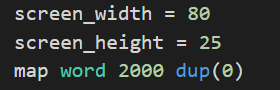
6.GoToXY

7.Delay

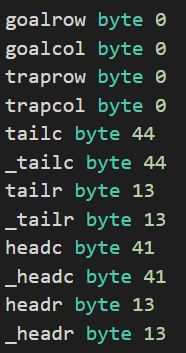
8.ReadKey

**Function與程式介紹**

蛇蛇世界的地圖透過一維陣列儲存80(width)x25(height)=2000



但在無論是紀錄蛇的頭還是尾巴或是goal或 trap皆適用row 跟 column的方式儲存，原因是這樣比較好做移動跟搜尋，因此需要有將row 跟 column轉為相對應在一維上的位置的function，取名為



(底線為2player)

**get\_index**:將row column轉為一維

get\_index PROC USES ebx eax edx

Row\*80 + column ，但是map是用word來存所以要乘上2

shl si,1 ; word

才可以找到對應map的位置

**get\_map、save\_map**:取得map、儲存map值

call 他們的要先把row 跟 column存在 dh dl 值無論是要存進去還是要return都存在bx

先call get\_index取得在map上的位置在取值或存值

**create\_goal:**產生goal，並畫出

透過RandomRange設定亂數範圍，來決定產生位置

**create\_trap:**產生trap，並畫出

透過RandomRange設定亂數範圍，來決定產生位置

**generate\_map:**根據mode變數(剛剛menu選的地圖會改變mode變數，預設為1，Box)

透過邏輯用LOOP產生想要的牆，將是牆的地方CALL save\_map，將0AAAAh存進去

**draw\_map:**將map畫出來(只畫牆)

透過兩個loop跑過所有map

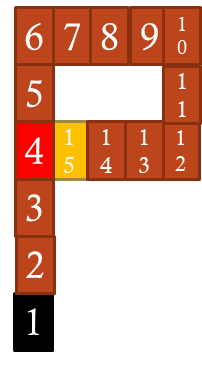
預先將牆設為0AAAAh所以若map值為0AAAAh則畫藍色(牆為藍色)

**clear\_map:**清空map

用雙層loop 跑過所有map，call save\_map把0存進去

**init\_snake:**初始化蛇的位置

**search\_segment:**搜尋蛇的下一節，如果傳進來是節3則會回傳節2的row 跟column

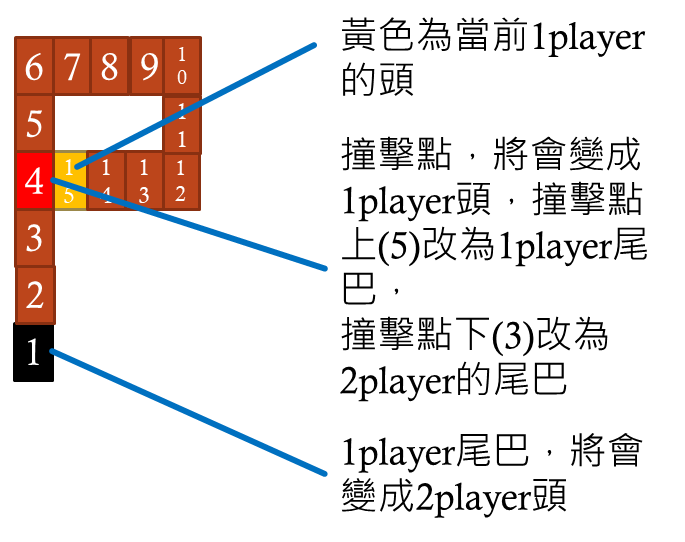
****

蛇是透過一個連續整數儲存的，尾巴固定為1一直到頭持續累加，因此移動時需要搜尋整個蛇，因此寫這個function，只要給某一節的值(bx)及row(dh)跟column(dls)就會回傳他的下一節(value-1)在哪

實作方法:

先將bx-1根據分別做dh dl 的加一減一找出這個row column的上下左右有沒有 bx-1這個值，找到存在dh dl

**create\_newplayer**:



然後將twoplayer變數改為 1

move\_snake:主要做移動及碰撞偵測

實作說明:

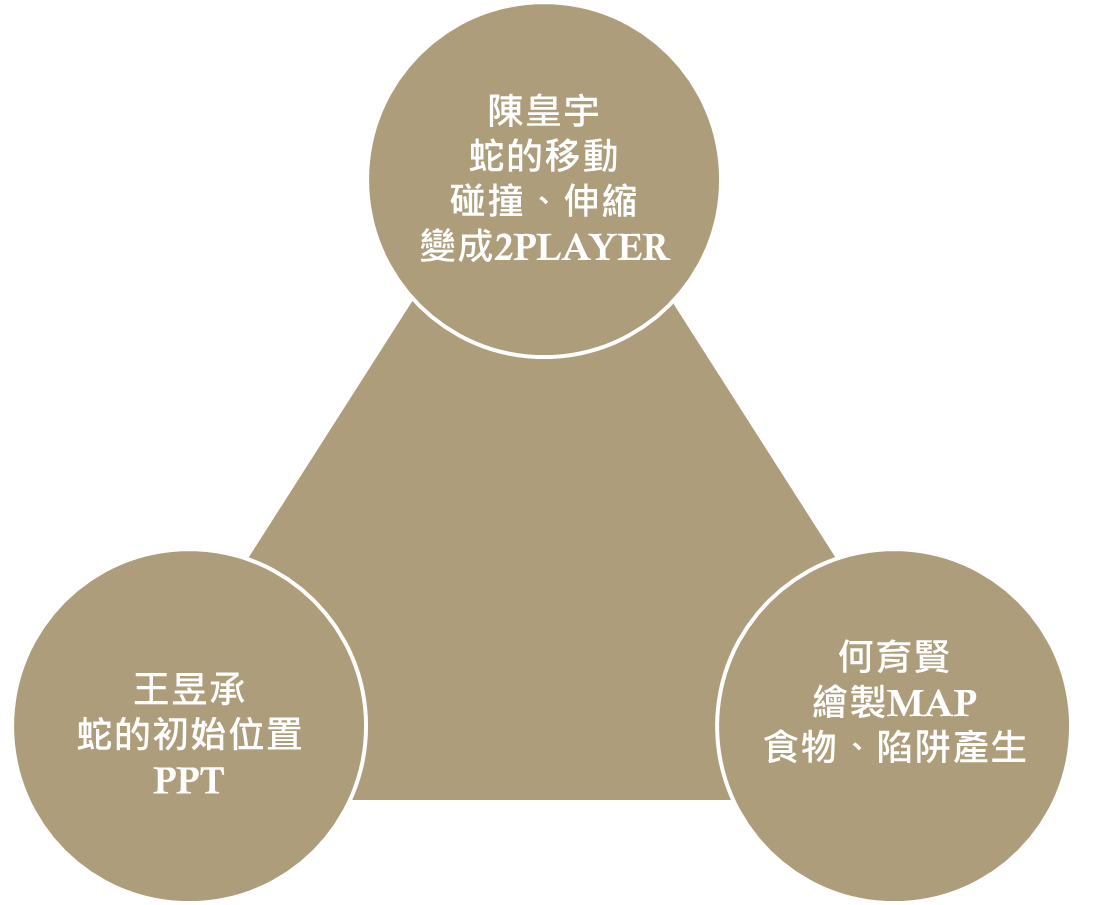
根據方向(snake\_d)決定上要上下左右，比對那個位置是否為goal的位置若是，eat設為1下次就不會跑讓尾巴消失的區塊erase(這樣就是變常因為頭還會繼續往前)，並且call create\_goal產生下一個，若撞到trap將eat\_t設為2下一次move 會將 eat\_t丟進ecx讓erase跑兩遍(這樣就會變短)，若撞到身體(get\_map後值不為0且不為0AAAAh)查看是否已經有2player有就近end\_game沒有就call creaete\_player，移動部分就是從頭開始(值為n有n節)，然後一直減一再call search\_segment 將原本的值-1(4會變3)直到跑到2

**其他\_開頭**:皆為2PLAYER的，功能一樣只是變數不一樣

**start\_game:**鍵盤事件，即跳會MENU

透過READKEY讀取鍵盤

**分工**

****