

組合語言與系統程式

期末 *project* 書面報告

主題：*Zombie Hunter*

組員：

109502517 洪啓賢

109502555 蔡明翰

109502518 陳洛鈞

*From*



## 組員分工：

### ➤ 蔡明翰：

程式架構、倒數計時

### ➤ 洪啓賢：

殭屍顏色變化、細節修改

景整造型與程式碼、計算分數動畫

### ➤ 陳洛鈞：

設計殭屍造型、開始畫面

倒數計時畫面、遊戲結束畫面

### ➤ 共同部份：

ppt 與書面報告製作

## 使用函式庫：

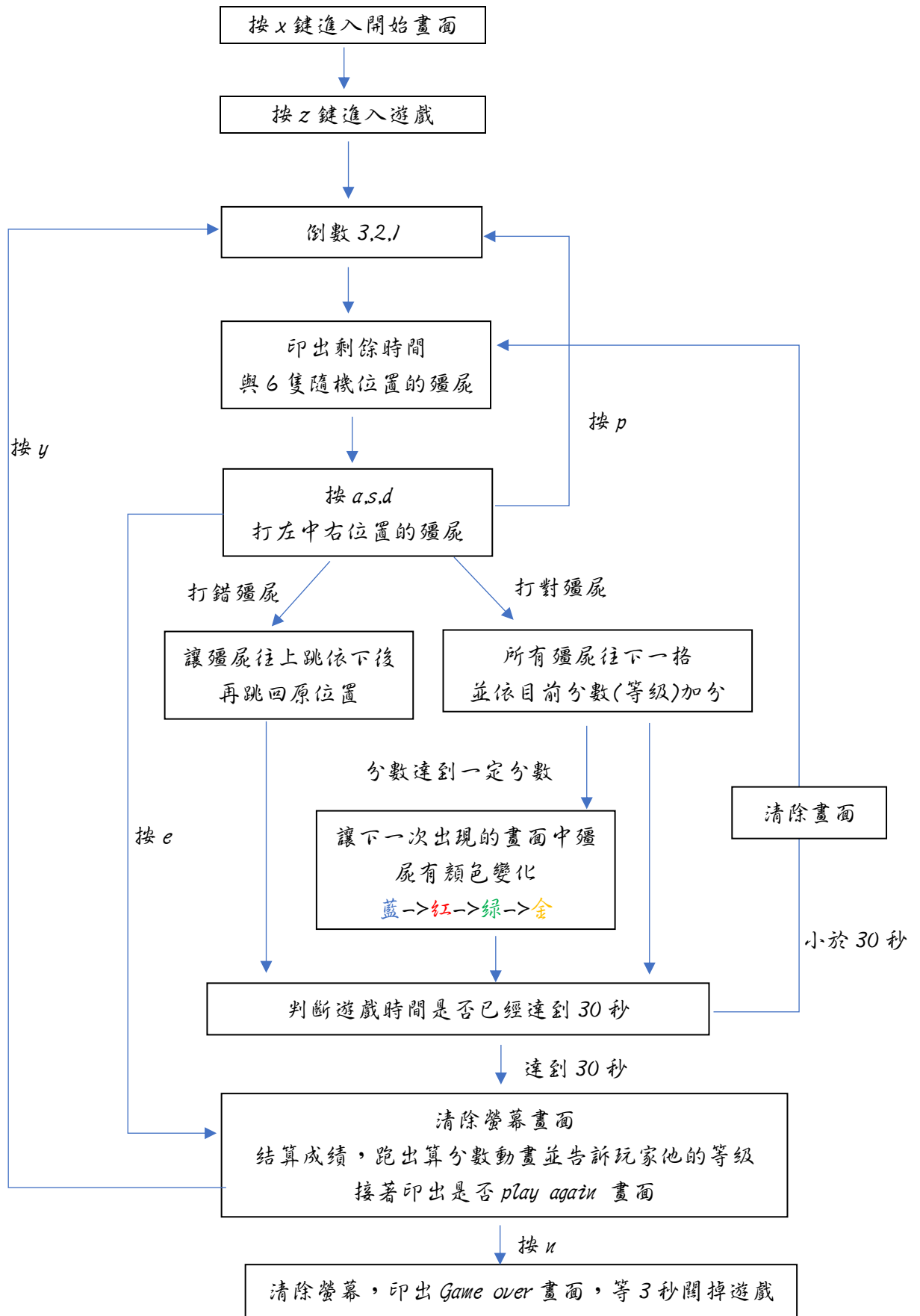
*Irvine32.inc*



## 遊戲規則：

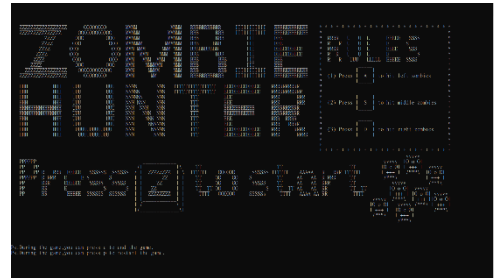
1. 先按 *alt+enter* 進入全螢幕，按 *z* 進入遊戲畫面
2. 每次遊戲時間為 30 秒。
3. 過程中以 *a,s,d* 為左中右三個按鍵打擊由上往下掉的殭屍。
4. 打錯殭屍會跳一下，打對就打下一個殭屍。
5. 如果分數達 1 萬多、4 萬多、9 萬多時，調整殭屍顏色(越高級分數越高)。
6. 如果想要結束遊戲，請按 *e*。
7. 如果想要重來，請按 *p*。
8. 30 秒過後遊戲結束，會自動計算得分。
9. 按 *y* 可以重新開始遊戲，按 *n* 會關掉遊戲。

程式流程：



關於程式的詳細說明：

```
main PROC
firstl:      ;叫玩家換成全螢幕
mov dl,0
mov dh,0
call gotoxy
mov edx,offset beginsmes
call writestring
call readchar
cmp al,"x"
jne firstl
begin:      ;遊戲開場及規則
call clrscr
mov success,0
mov dl,0
mov dh,0
call Gotoxy ;游標移動到(0,0)
call crlf
call crlf
call crlf
call crlf
mov edx,offset ms1
call writestring
call crlf
mov edx,offset ms2
call writestring
call crlf
mov edx,offset ms3
call writestring
call crlf
```



先顯示字串讓玩家自己開啓全螢幕後按x鍵進行下一步。

秀出華麗開場以及規則，按z後開始遊戲(使用 writestring 印出)

```
call crlf
mov edx,offset ms34
call writestring
call crlf
mov edx,offset ms35
call writestring
call crlf
mov edx,offset ms36
call writestring
call crlf
mov edx,offset ms37
call writestring
call crlf
mov eax,1000
call delay
mov ecx,42
mov dl,0
mov dh,0
call Gotoxy
L42:
mov edx,offset ms38
call writestring
call crlf
Loop L42
mov dl,0
mov dh,0
call Gotoxy
mov ecx,4
LLLLL:
mov edx,offset ms39
call writestring
```

倒數 321，印出3後，使用 call delay 將 EAX 輸入 1000，延遲出現一秒的效果，2 跟 1 也是依此類推印出。

```

invoke gettickcount
mov systime,eax ;紀錄開始時間
mov dl,0
mov dh,6
call Gotoxy
mov ecx,6
mov esi,0
mov edx,offset times ;印出剩幾秒
call writestring
invoke gettickcount ;倒數計時30秒
sub eax,systime
mov edx,0
mov ebx,1000
div ebx
sub counting,eax
mov eax,counting
call writedec
call crlf
mov eax,16 ;初始殭屍顏色
mov eax,lightblue+(black*16)
call settextcolor
L: ;印出殭屍
mov eax,3
call randomrange
mov randnum,eax
cmp randnum,0
je L1
cmp randnum,1
je L2
cmp randnum,2
je L3
L1:
mov edx,offset a1
call writestring
call crlf
mov edx,offset a2
call writestring
call crlf
mov edx,offset a3
call writestring
call crlf
mov edx,offset a4
call writestring
call crlf
mov edx,offset clrms
call writestring
call crlf
mov ans[esi], "a"
inc esi
jmp exitl
L2:
mov edx,offset s1
call writestring
call crlf
mov edx,offset s2
call writestring
call crlf

```

遊戲開始：

1. 寫一個打殭屍正確答案的陣列(使用 `call randomrange`)
  - a. `Call randomrange` 輸入變數 3，代表從隨機從 0~2
  - b. 如果是 0 就跳 L1，1 跳 L2，2 跳 L3。每個 label 都會印出隨機位置的殭屍
  - c. 每次殭屍印完後，在 `ans` 陣列輸入需要該按鈕名稱，如左邊就是要打 a，中間要打 s，右邊要打 d
  - d. L1~3 是印出一開始的殭屍，L9~11 是印出之後出現在陣列的殭屍。

```

mov eax,16
mov eax,gray+(black*16)
call settextcolor
mov edx,offset pal
call writestring
call crlf
mov edx,offset pa2
call writestring
call crlf
mov edx,offset pa3
call writestring
call crlf
mov edx,offset pa4
call writestring
call crlf
mov edx,offset pa5
call writestring
call crlf
jmp hit

```



2. 印出按鈕 ASD

```

hit: ; 擊打
mov dl,0
mov dh,6
call Gotoxy
mov eax,16
mov eax,white+(black*16) ; 設定顏色
call settextcolor
mov edx,offset times
call writestring
invoke gettickcount ; 倒數計時30秒
sub eax,systime
mov edx,0
mov ebx,1000
div ebx
cmp eax,30
jae reallyexit
cmp eax,20
ja L12
sub counting,eax
mov eax,counting
mov counting,30
call writedec
call crlf
jmp L14

```

3. 印出剩下的時間(倒數 30 秒)以及 6 隻殭屍(時間是使用 *invoke gettickcount* 利用一開始的系統時間與現在的系統時間相減相加等運算得出剩餘時間。

- 如果是剩個位數秒，就多印一個 0，讓數字看起來還是兩位數。
- 如果是剩餘兩位數秒，就不用多印 0，

```

; wronghit, 1 ; 打錯僵屍時
mov esi,5
mov dl,0
mov dh,35
call Gotoxy
mov edx,offset times
cmp success,10120
jbe changeback_whitel
cmp success,40529
jbe changeback_blue1
cmp success,91345
jb changeback_green1
jmp changeback_yellow1
changeback_whitel:
mov eax,16
mov eax,lightblue+(black*16)
call settextcolor
jmp call_set2
changeback_blue1:
mov eax,16
mov eax,lightred+(black*16)
call settextcolor
jmp call_set2
changeback_green1:
mov eax,16
mov eax,green+(black*16)
call settextcolor
jmp call_set2
changeback_yellow1:
mov eax,16
mov eax,yellow+(black*16)
call settextcolor
; call writestring
call_set2:
mov edx,offset clmes
call writestring
mov dl,0
mov dh,31
call Gotoxy
jmp L8

```

```

; wronghit, 2
LLLL:
mov esi,5
mov wronghit,2
mov dl,0
mov dh,36
call Gotoxy
mov eax,300
call delay
mov dl,0
mov dh,31
call Gotoxy
mov edx,offset clmes
call writestring
call crlf
call_set1:
jmp L8

```

```

L8: ; 倒數
xor edx,edx
cmp ans[esi],"a"
je L9
cmp ans[esi],"s"
je L10
cmp ans[esi],"d"
je L11
jmp exit3
L9:
mov edx,offset a1
call writestring
call crlf
mov edx,offset a2
call writestring
call crlf
mov edx,offset a3
call writestring
call crlf
mov edx,offset a4
call writestring
call crlf
inc esi
cmp wronghit,1
je LLLL
jmp exit3
L10:
mov edx,offset s1
call writestring
call crlf
mov edx,offset s2
call writestring
call crlf
mov edx,offset s3
call writestring
call crlf
mov edx,offset s4
call writestring
call crlf
inc esi
cmp wronghit,1
je LLLL
jmp exit3

```

4. 打錯殭屍時，用 *call gotoxy* 在上面一小格印出一樣的殭屍 *delay* 0.3 秒後恢復原裝，就可以製造出殭屍跳一下的效果

- 因為 *settextcolor* 的函式會連續印出所有出現在螢幕上的文字的顏色，因此連打錯的時候都要重新判斷要上什麼顏色。
- 我們設一個 *jmp L8* 讓他跳到 L9~11 其中一個印出看起來往上跳的殭屍，再用 *LLLL* 讓他跳到可以 *call delay* 延遲 300 毫秒的地方。

```

L4: ;將正確答案的陣列由往下移一個
mov bl,ans[esi-1]
mov ans[esi],bl
sub esi,1
loop L4

mov eax,3 ;將新答案放進ans[0]
call randomrange
mov randnum,eax
cmp randnum,0
je L5
cmp randnum,1
je L6
cmp randnum,2
je L7

L5:
mov ans[0],"a"
jmp exit2
L6:
mov ans[0],"s"
jmp exit2
L7:
mov ans[0],"d"
exit2:
mov esi,0
mov ecx,6
mov di,0
mov dh,7
call Gotoxy

L8: ;印出殭屍
xor edx,edx

```

5. 打對殭屍時，把陣列中正確答案依序往下移一格，再補上一個新答案進去，再重新印出所有殭屍及倒數時間。

```

yes: ;打正確殭屍時
cmp success,10120
jb L50p
cmp success,40529
jb L150p
cmp success,91345
jb L300p
jmp L300up
L50p:
mov eax,16
mov eax,lightblue+(black*16)
call settexitcolor
add success,184
jmp endoflp
L150p:
mov eax,16
mov eax,lightred+(black*16)
call settexitcolor
add success,647
jmp endoflp
L300p:
mov eax,16
mov eax,green+(black*16)
call settexitcolor
add success,1588
jmp endoflp
L300up:
mov eax,16
mov eax,yellow+(black*16)
call settexitcolor
add success,3669
jmp endoflp
endoflp:

```

6. 分數達1萬多、4萬多、9萬多時會調整殭屍顏色，越高級殭屍分數越高。

```

cmp al,"e" ;按e結束
je reallyexit1
cmp al,"p" ;按p重來
je gamestart

```

7. 按e會跳到結算分數，按p會回到3.2.1畫面。

```

hit: ; 擊打
mov dl,0
mov dh,6
call Gotoxy
mov eax,16
mov eax,white+(black*16) ; 設定顏色
call settextcolor
mov edx,offset timemes
call writestring
invoke gettickcount ; 倒數計時30秒
sub eax,systime
mov edx,0
mov ebx,1000
div ebx
cmp eax,30
jae reallyexit
cmp eax,20
ja L12
sub counting,ecx
mov ecx,counting
mov counting,30
call writedec
call crlf
jmp L14

LL184: ; 分數結算動畫
mov ecx,104
mov ebx,0
Loopop:
mov edx,offset score
call writestring
mov eax,ebx
call writedec
call crlf
mov dl,0
mov dh,4
call Gotoxy
mov eax,1
call delay
add ebx,1
Loop Loopop
mov edx,offset score
call writestring
mov eax,success
call writedec
call crlf
; 10000

cmp success,10000
jb L50pl
cmp success,30000
jb L150pl
cmp success,70000
jb L300pl
cmp success,150000
jb L400pl
jmp L400up
L50pl:
mov edx,offset donemes1
call writestring
call crlf
jmp endofLpl
L150pl:
mov edx,offset donemes2
call writestring
call crlf
jmp endofLpl
L300pl:
mov edx,offset donemes3
call writestring
call crlf
jmp endofLpl
L400pl:
mov edx,offset donemes4
call writestring
call crlf
jmp endofLpl
L400up:
mov edx,offset donemes5
call writestring
call crlf

```

8. 時間到 30 秒，用 `jmp reallyexit` 跳出去迴圈，印出分數以及 `play again`(印出殭屍時，分數由 1、2、3、4 等等開始加上，每次印出 `delay 0.001` 秒，就可以製造出分數跳上去的感覺。

```

L13:
call readchar
cmp al,"n" ; 按n結束
je done
cmp al,"y" ; 按y重來
je gamestart
jmp L13
done: ; 印出game over
call clrscr
mov ecx,11

```

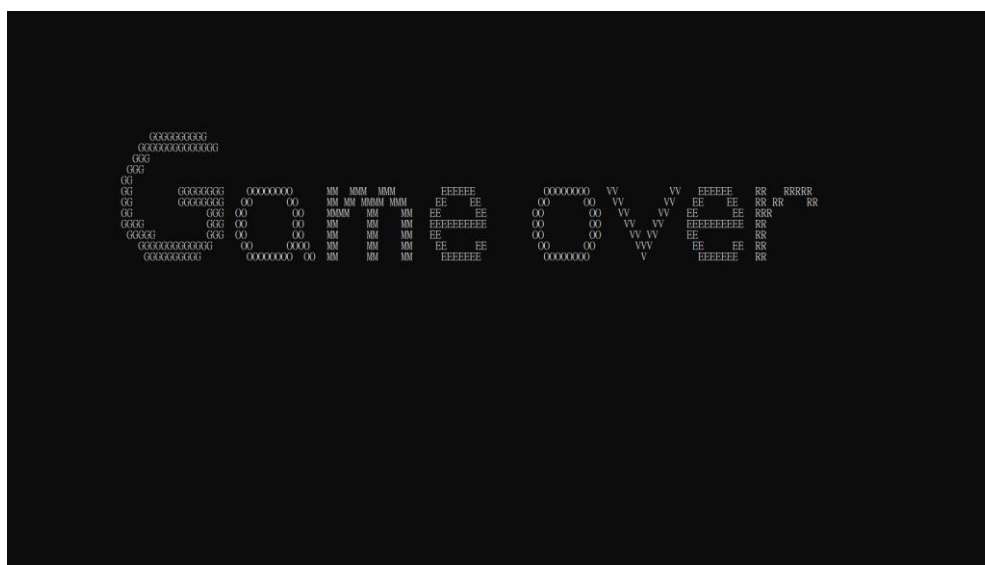
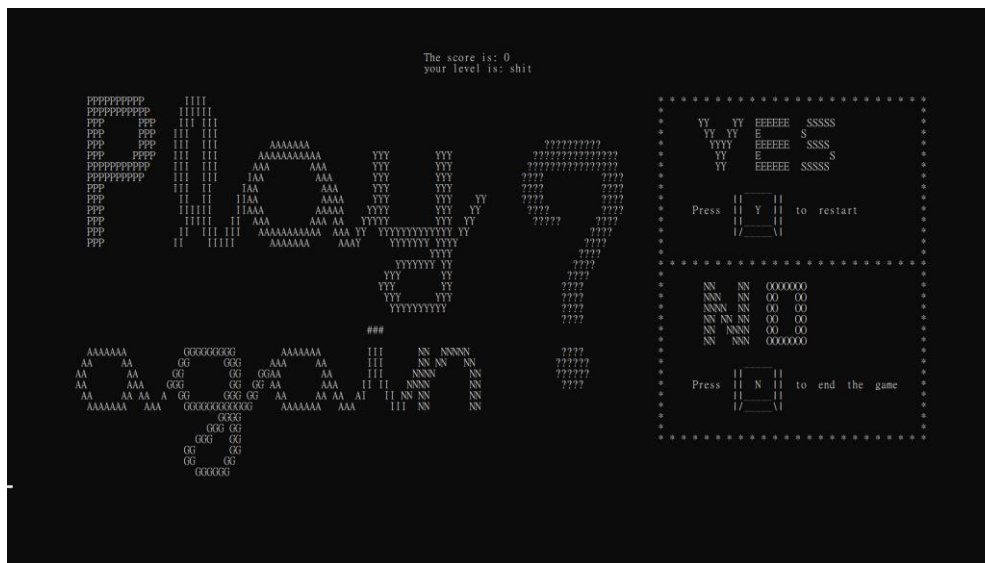
9. `readchar` 按 `y` 重新開始，按 `n` 印出 `game over` 然後結束程式。

Demo 截圖：









Demo 影片：

[影片網址](#)

Ps. Demo 的配樂為邊玩邊播放 youtube 歌曲掩蓋鍵盤吵雜聲。