



國立台灣科技大學

微算機概論與應用

指導教授：陸敬互 教授

---

## 微算機概論實習報告

### 期末報告

班 級       ： 四電二甲  
學 生       ： 魏趨棣 李秉和  
學 號       ： B10607025 B10607055  
建檔日期   ： 2019/01/03

(中文字型:標楷體 英文字型:Times New Roman)

## 一、學習成果(程式功能說明)

```
include macro.h
```

```
.model small
```

```
.stack
```

```
.data
```

```
string1 db "Game Over$";遊戲結束之字串定義
```

```
string2 db "Best Score:  $"
```

```
string3 db "Score:  $"
```

```
string4 db "press Enter to play again $"
```

```
string5 db "press Esc to leave $"
```

```
string6 db "Game Start(press Enter to play)$" ;遊戲結束之  
字串定義
```

```
string7 db "Exit(press Esc to leave)  $"
```

```
string8 db "    FROGGY    $"
```

```
xend dw 340;青蛙x座標的底線
```

```
x dw 300;青蛙x座標
```

```
y dw 440;青蛙y座標
```

```
virusy dw 480;下方毒氣y座標
```

```
counter dw 40 ;青蛙counter
```

```
counter0 dw 10 ;背景counter
```

```
counter1 dw 0 ;背景counter
```

```
viruscounter dw ? ;毒氣列印之counter
```

```
goviruscounter dw 0;開始施放毒氣之counter
```

```
carXcounter dw 45 ;車x座標counter
```

```
carYcounter dw 45
```

```
jumpcounter dw 5 ;青蛙可以跳的counter
```

```
car1xdw 595;車1的x座標
```

```
car2xdw 0
```

```
car3xdw 595
```

```

car4xdw 0
car5xdw 595
car6xdw 0
car7xdw 595
cary dw 55
car1speed dw 12;車1的速度
car2speed dw 7
car3speed dw 10
car4speed dw 15
car5speed dw 8
car6speed dw 6
car7speed dw 4
virusspeed dw 1
score dw 0 ;分數
bestscore dw 0 ;歷史分數
fail dw 0 ;失敗counter
nocardw 0 ;毒氣經過 無車車的counter
color1 db 1010b;青蛙顏色
carcolor db 1001b ;車顏色
.code
main proc
    mov ax,@data
    mov ds,ax
startpicture:
    SetMode 12h ;遊戲開始畫面
    Setcolor 1
    SetCursor 10,25 ;遊戲名稱
    PrintStr string8
    SetCursor 23,25 ;start位置
    PrintStr string6
    SetCursor 26,25 ;exit位置
    PrintStr string7
press:

```

```

mov ah,10h
int 16h
cmp al,1bh;按esc結束
jz over
cmp al,0dh;按enter進入遊戲
jz gamestart0
jmp press ;迴圈
gamestart0: ;新遊戲分數歸0
    mov     score,0
gamestart: ;遊戲開始 or 青蛙掉到隊岸 初始化設定
    mov     x,300
    mov     y,440
    mov     virusy,480
    mov     nocar,0
    mov     car1x,595
    mov     car2x,0
    mov     car3x,595
    mov     car4x,0
    mov     car5x,595
    mov     car6x,0
    mov     car7x,595
    mov     cary,595
    mov     fail,0
    mov     goviruscounter,0
    mov     jumpcounter,5
    mov     xend,340
    SetMode 12h
    SetColor 0111b ;背景灰色
    call Setback ;畫白色分隔島
    call drawfrog ;畫青蛙
    call drawcar ;畫不會動的7台車子

game: ;遊戲迴圈

```

```

callscan ;掃描按鍵，青蛙上下左右移動
cmp al,1bh;按esc 直接結束
je over
cmp al,38h
je up
cmp al,32h
je down
cmp al,34h
je left
cmp al,36h
je right
calldrawcarmove ;7台車子移動
cmp fail,1 ;碰到車子gameover
je gameover
cmp goviruscounter,60 ;車子移動20次後底部開始放毒
ja virus
jmp game

```

```

virus: ;放毒
    calldrawvirus ;畫毒氣
    cmp fail,1 ;碰到毒gameover
    je gameover
    jmp game

```

```

up: ;青蛙向上跳
    cmp jumpcounter,5 ;車子移動5次之後才可以跳，不能
    壓著按鍵一直跳
    jae canjump
    jmp game
canjump:
    subcolor1,0011b
    calldrawfrog ;畫背景顏色，清掉原本的
    addcolor1,0011b

```

```

    sub y,55 ;青蛙往上一個道路
    call drawfrog
    mov    jumpcounter,0 ;可以跳得counter歸0
    cmp    dx,0 ;判斷到最上面時，增加遊戲難度
    je     gowin
    jmp     game
down:
    sub color1,0011b
    call drawfrog
    add color1,0011b
    add y,55
    cmp    y,440 ;在底部時不能再向下移動
    ja     bottom
    jmp     nothing
bottom:
    mov    y,440
nothing: ;在底部以上，可以向下
    call drawfrog
    jmp     game
left:
    sub color1,0011b
    call drawfrog
    add color1,0011b
    sub x,55 ;青蛙移動55格
    sub xend,55 ;青蛙x的底線也要一起移動55格
    call drawfrog
    jmp     game
right:
    sub color1,0011b
    call drawfrog
    add color1,0011b
    add x,55
    add xend,55
    call drawfrog

```

jmp game

gowin: ;增加遊戲難度

calldelay ;等待一下

callwin ;呼叫增加遊戲難度\*1次

jmp gamestart

gameover: ;碰到毒 or 車子 遊戲結束

mov ax,score ;判斷最佳歷史分數

cmp ax,bestscore

jae setscore

gameover1:

SetMode 12h

SetColor 0001b ;背景改藍色

SetCursor 8,27 ;設游標位置

PrintStr string1 ;印game over

SetCursor 14,27

PrintStr string2 ;印最佳分數

mov ax,bestscore ;ax為要處理後印的4位數

callprintscore ;處理4位數後印

SetCursor 16,27

PrintStr string3 ;印分數

mov ax,score

callprintscore

SetCursor 20,27

PrintStr string4 ;印在玩一次的方法

SetCursor 22,27

PrintStr string5 ;印離開的方法

GetChar

cmp al,1bh;按esc離開

je over

cmp al,0dh;按enter 再玩一次

```

    je gamestart0 ;再玩一次，分數設0 其餘初始化
    jmp    gameover1
setscore: ;當分數大於最佳分數時
    mov    bestscore,ax
    jmp    gameover1

over: ;離開遊戲
    SetMode 03h
    mov ax,4c00h
    int 21h
main endp

```

```

drawfrog proc    ;劃青蛙
    mov    cx,x
    mov    dx,y
L5:
    WtPixel cx,dx,color1
    inc dx    ;由上面向下畫
    dec counter;y軸畫40次
    cmp    counter,0
    je    L6
    jmp    L5
L6:
    mov    counter,40 ;y畫的次數重設為40
    sub dx,40 ;y從回上方
    inc cx ;X軸+1，再往右邊畫
    cmp    cx,xend    ;只能畫40次，不可超過
    je    L7
    jmp    L5
L7:
    ret

```



drawfrog endp

drawvirus proc ;畫毒

mov cx,0 ;從最左邊開始

mov dx,virusy

mov ax,viruspeed ;毒向上的速度

mov viruscounter,ax ;毒向上的格數

virusstart:

callcolortest ;取當前位置的顏色，是青蛙的顏色就失敗

cmp fail,1

je virusend

cmp al,1001b ;取當前位置的顏色，是車的顏色就設置車子不要動

je carstop

WrPixel cx,dx,0100b

inc cx ;由最右邊 0 畫到 640

cmp cx,641

je virus1

jmp virusstart

virus1:

mov cx,0

sub dx,1 ;向上一列

sub viruscounter,1

cmp viruscounter,0 ;判斷毒可以向上的格數，第一次只會向上一格

je virusend

jmp virusstart

carstop:

addnocar,1

virusend:

```
mov    virusy,dx
ret
drawvirus endp
```

```
drawcar proc;畫7台部會動的車車
    callcar1    ;畫車1的副涵式
    callcar2
    mov    car2x,0;車2 4 6為從最左邊出來
    callcar3
    callcar4
    mov    car4x,0
    callcar5
    callcar6
    mov    car6x,0
    callcar7
    ret
drawcar endp
```

```
drawcarmove proc ;7車移動
```

```
drawcar7set:;最下方的車子移動
    cmp    nocar,1;碰到毒
    je     drawcar6set0 ;nocar等於1，表示只清一次
    cmp    nocar,2;清完一次後，直接跳下一台車移動
    jae    drawcar6set
    mov    carcolor,0111b
    callcar7set;車子移動，先用灰色清除
    mov    carcolor,1001b
    callcar7set;車子移動

    jmp    drawcar6set
```

```
drawcar6set0: ;碰到毒，則全部用背景顏色(灰色)清掉
    mov    carcolor,0111b
    subcar7x,44 ;不會動的車子是從左上開始畫，所以X座
    標要減回去
    callcar7
    addnocar,1;清完一次後，nocar+1，下次可以直接跳下
    一台車
```

```
drawcar6set: ;車6移動
    cmp    nocar,3
    je     drawcar5set0
    cmp    nocar,4
    jae    drawcar5set
    mov    carcolor,0111b
    callcar6set
    mov    carcolor,1001b
    callcar6set
    jmp    drawcar5set
drawcar5set0:
    mov    carcolor,0111b
    callcar6
    addnocar,1
```

```
drawcar5set: ;車5移動
    cmp    nocar,5
    je     drawcar4set0
    cmp    nocar,6
    jae    drawcar4set
    mov    carcolor,0111b
    callcar5set
    mov    carcolor,1001b
    callcar5set
    jmp    drawcar4set
drawcar4set0:
```

```
mov    carcolor,0111b
subcar5x,44
callcar5
addnocar,1
```

drawcar4set: ;車4移動

```
cmp    nocar,7
je     drawcar3set0
cmp    nocar,8
jae    drawcar3set
mov    carcolor,0111b
callcar4set
mov    carcolor,1001b
callcar4set
jmp    drawcar3set
```

drawcar3set0:

```
mov    carcolor,0111b
callcar4
addnocar,1
```

drawcar3set: ;車3移動

```
cmp    nocar,9
je     drawcar2set0
cmp    nocar,10
jae    drawcar2set
mov    carcolor,0111b
callcar3set
mov    carcolor,1001b
callcar3set
jmp    drawcar2set
```

drawcar2set0:

```
mov    carcolor,0111b
subcar3x,44
callcar3
```

```
addnocar,1
```

```
drawcar2set: ;車2移動
```

```
    cmp    nocar,11  
    je     drawcar1set0  
    cmp    nocar,12  
    jae    drawcar1set  
    mov    carcolor,0111b  
    call   car2set  
    mov    carcolor,1001b  
    call   car2set  
    jmp    drawcar1set
```

```
drawcar1set0: ;車1移動
```

```
    mov    carcolor,0111b  
    call   car2  
    addnocar,1
```

```
drawcar1set:
```

```
    cmp    nocar,13  
    jae    drawcarmoveend  
    mov    carcolor,0111b  
    call   car1set  
    mov    carcolor,1001b  
    call   car1set
```

```
drawcarmoveend:
```

```
    inc    goviruscounter  
    inc    jumpcounter  
    ret
```

```
drawcarmove endp
```

car1set proc ;車1移動涵式，為從右上往左下開始畫，X軸只畫速度排，而非全畫

```

    jmp    start
Lcarset: ;車子碰到邊界時，先清除原本位置的車子，再
從原始位置開始畫。
    mov    carcolor,0111b
    mov    car1x,0;從邊界(0)開始往右下清
    callcar1 ;用畫原始車子的涵式清除台車
    mov    carcolor,1001b
    mov    car1x,595 ;車子初始位置
    callcar1
    mov    carcolor,0111b
    callcar1set;車子回初始位置後再移動一次
    jmp    setlend
start:
    cmp    carcolor,0111b    ;比較是要清除還是移動
    je     cover1
    jne    add1
cover1:
    mov    cx,car1x    ;車1的X座標再最右邊，不用動即可劃
    jmp    draw1
add1:;移動增加
    cmp    car1speed,45    ;當速度超過車子大小時，用畫原
始車子的涵式移動
    ja     bigsize
    mov    ax,car1x
    sub    ax,45    ;再最右邊的X座標需減45，變到最右邊再開
始往左增加
    mov    cx,ax
    mov    ax,car1speed
    sub    car1x,ax    ;車子的X座標(再最右邊)扣掉速度，改為
移動後在最右邊的座標。
    jmp    draw1
bigsize:
    cmp    cx,45    ;比較車子是否碰到邊界

```

```

    jb  Lcarset
    sub cx,45 ;X座標需減45，因為畫原始車子的涵式是從
最左邊開始畫，所以cover完後的x座標需再-45再畫
    mov  car1x,cx
    call car1
    jmp  setlend

```

draw1:

```

    mov  cary,55;車7的y座標
    mov  dx,cary
    mov  carYcounter,45 ;車子的大小(45*45)
    mov  ax,car1speed
    mov  carXcounter,ax ;車子速度，及x軸一次可以增
加多少

```

Lcar0set:

```

    call colortest ;判斷是否碰到青蛙
    WtPixel cx,dx,carcolor
    inc dx ;向下畫
    dec carYcounter
    cmp  carYcounter,0
    je  Lcar1set
    jmp  Lcar0set

```

Lcar1set:

```

    mov  carYcounter,45
    sub dx,45
    dec cx ;向左畫
    cmp  cx,0 ;碰到邊界
    je  Lcarset
    dec carXcounter
    cmp  carXcounter,0 ;一次只畫 or 清 數排，而非全畫
    je  setlend
    jmp  Lcar0set

```

```

set1end:
    ret
    car1set endp

```

car2set proc ;車2移動涵式，為從左上往右下開始畫，X軸只畫速度排，而非全畫

```

    jmp    start2
Lcar2set:;車2碰到邊界
    mov    carcolor,0111b
    mov    car2x,595
    callcar2
    mov    carcolor,1001b
    mov    car2x,0
    callcar2
    subcx,45
    mov    car2x,cx
    jmp    set2end

```

```

start2:
    cmp    carcolor,0111b
    je     cover2
    jne    add2

```

cover2:  
 mov cx,car2x ;車2的X座標再最左邊，不用動即可從左邊開始清

```

    jmp    draw2

```

```

add2:
    cmp    car2speed,45 ;車速大於45
    ja     bigsize2
    mov    ax,45
    addax,car2x ;車2的X座標清完後，需加45才能從右邊開始往右畫，移動。

```

```

    mov    cx,ax
    mov    ax,car2speed
    addcar2x,ax ;車子清完 畫完後把X座標重新設在最左

```



邊。

```
    jmp    draw2
```

bigsize2:

```
    cmp    cx,595 ;碰到邊界
```

```
    ja     Lcar2set
```

```
    mov    car2x,cx
```

```
    call   car2
```

sub cx,45 ;X座標需減45，因為畫完後X再最右邊，需減45回到最左邊

```
    mov    car2x,cx
```

```
    jmp    set2end
```

draw2:

```
    mov    cary,110 ;車2的y座標
```

```
    mov    dx,cary
```

```
    mov    carYcounter,45
```

```
    mov    ax,car2speed
```

```
    mov    carXcounter,ax
```

Lcar2set1:

```
    call   colortest
```

```
    WrPixel cx,dx,carcolor
```

```
    inc dx ;向下畫
```

```
    dec carYcounter
```

```
    cmp    carYcounter,0
```

```
    je     Lcar2set2
```

```
    jmp    Lcar2set1
```

Lcar2set2:

```
    mov    carYcounter,45
```

```
    sub dx,45
```

```
    inc cx ;向右畫
```

```
    cmp    cx,640 ;碰到邊界
```

```
    je     Lcar2set
```

```
    dec carXcounter
```

```
    cmp    carXcounter,0
```

```

    je set2end
    jmp Lcar2set1
set2end:
    ret
car2set endp

```

car3set proc ;同車1，只改y座標跟速度

```

    jmp start3
Lcar3set:
    mov carcolor,0111b
    mov car3x,0
    call car3
    mov carcolor,1001b
    mov car3x,595
    call car3
    mov carcolor,0111b
    call car3set
    jmp set3end
start3:
    cmp carcolor,0111b
    je cover3
    jne add3
cover3:
    mov cx,car3x
    jmp draw3
add3:
    cmp car3speed,45
    ja bigsize3
    mov ax,car3x
    sub ax,45
    mov cx,ax
    mov ax,car3speed
    sub car3x,ax
    jmp draw3

```

bigsize3:

```
    cmp    cx,45
    jb     Lcar3set
    sub cx,45
    mov     car3x,cx
    call car3
    jmp     set3end
```

draw3:

```
    mov     cary,165
    mov     dx,cary
    mov     carYcounter,45
    mov     ax,car3speed
    mov     carXcounter,ax
```

Lcar3set1:

```
    call colortest
    WrPixel cx,dx,carcolor
    inc dx
    dec carYcounter
    cmp     carYcounter,0
    je      Lcar3set2
    jmp     Lcar3set1
```

Lcar3set2:

```
    mov     carYcounter,45
    sub dx,45
    dec cx
    cmp     cx,0
    je      Lcar3set
    dec carXcounter
    cmp     carXcounter,0
    je      set3end
    jmp     Lcar3set1
```

set3end:

```
    ret
```

```
car3set endp
```

```
car4set proc ;同車2，只改y座標跟速度
```

```
    jmp    start4
```

```
Lcar4set:
```

```
    mov    carcolor,0111b
```

```
    mov    car4x,595
```

```
    call car4
```

```
    mov    carcolor,1001b
```

```
    mov    car4x,0
```

```
    call car4
```

```
    sub cx,45
```

```
    mov    car4x,cx
```

```
    jmp    set4end
```

```
start4:
```

```
    cmp    carcolor,0111b
```

```
    je     cover4
```

```
    jne    add4
```

```
cover4:
```

```
    mov    cx,car4x
```

```
    jmp    draw4
```

```
add4:
```

```
    cmp    car4speed,45
```

```
    ja     bigsize4
```

```
    mov    ax,45
```

```
    add ax,car4x
```

```
    mov    cx,ax
```

```
    mov    ax,car4speed
```

```
    add car4x,ax
```

```
    jmp    draw4
```

```
bigsize4:
```

```
    cmp    cx,595
```

```
    ja     Lcar4set
```

```

mov    car4x,cx
call car4
sub cx,45
mov    car4x,cx
jmp    set4end

```

draw4:

```

mov    cary,220
mov    dx,cary
mov    carYcounter,45
mov    ax,car4speed
mov    carXcounter,ax

```

Lcar4set1:

```

call colortest
WrPixel cx,dx,carcolor
inc dx
dec carYcounter
cmp    carYcounter,0
je     Lcar4set2
jmp    Lcar4set1

```

Lcar4set2:

```

mov    carYcounter,45
sub dx,45
inc cx
cmp    cx,640
je     Lcar4set
dec carXcounter
cmp    carXcounter,0
je     set4end
jmp    Lcar4set1

```

set4end:

```

ret
car4set endp

```

car5set proc ;同車1，只改y座標跟速度

```

    jmp    start5
Lcar5set:
    mov    carcolor,0111b
    mov    car5x,0
    call car5
    mov    carcolor,1001b
    mov    car5x,595
    call car5
    mov    carcolor,0111b
    call car5set
    jmp    set5end
start5:
    cmp    carcolor,0111b
    je     cover5
    jne    add5
cover5:
    mov    cx,car5x
    jmp    draw5
add5:
    cmp    car5speed,45
    ja     bigsize5
    mov    ax,car5x
    sub    ax,45
    mov    cx,ax
    mov    ax,car5speed
    sub    car5x,ax
    jmp    draw5
bigsize5:
    cmp    cx,45
    jb     Lcar5set
    sub    cx,45
    mov    car5x,cx
    call car5
    jmp    set5end
draw5:

```

```

    mov    cary,275
    mov    dx,cary
    mov    carYcounter,45
    mov    ax,car5speed
    mov    carXcounter,ax
Lcar5set1:
    call colortest
    WrPixel cx,dx,carcolor
    inc dx
    dec carYcounter
    cmp    carYcounter,0
    je     Lcar5set2
    jmp    Lcar5set1
Lcar5set2:
    mov    carYcounter,45
    sub dx,45
    dec cx
    cmp    cx,0
    je     Lcar5set
    dec carXcounter
    cmp    carXcounter,0
    je     set5end
    jmp    Lcar5set1
set5end:
    ret
    car5set endp

```

car6set proc ;同車2，只改y座標跟速度

```

    jmp    start6
Lcar6set:
    mov    carcolor,0111b
    mov    car6x,595
    call car6
    mov    carcolor,1001b

```

```

    mov    car6x,0
    call car6
    sub cx,45
    mov    car6x,cx
    jmp    set6end
start6:
    cmp    carcolor,0111b
    je     cover6
    jne    add6
cover6:
    mov    cx,car6x
    jmp    draw6
add6:
    cmp    car6speed,45
    ja     bigsize6
    mov    ax,45
    add ax,car6x
    mov    cx,ax
    mov    ax,car6speed
    add car6x,ax
    jmp    draw6
bigsize6:
    cmp    cx,595
    ja     Lcar6set
    mov    car6x,cx
    call car6
    sub cx,45
    mov    car6x,cx
    jmp    set6end

draw6:
    mov    cary,330
    mov    dx,cary
    mov    carYcounter,45
    mov    ax,car6speed

```



```

    mov    carXcounter,ax
Lcar6set1:
    call colortest
    WrPixel cx,dx,carcolor
    inc dx
    dec carYcounter
    cmp    carYcounter,0
    je     Lcar6set2
    jmp    Lcar6set1
Lcar6set2:
    mov    carYcounter,45
    sub dx,45
    inc cx
    cmp    cx,640
    je     Lcar6set
    dec carXcounter
    cmp    carXcounter,0
    je     set6end
    jmp    Lcar6set1
set6end:
    ret
    car6set endp

car7set proc ;同車1，只改y座標跟速度
    jmp    start7
Lcar7set:
    mov    carcolor,0111b
    mov    car7x,0
    call car7
    mov    carcolor,1001b
    mov    car7x,595
    call car7
    mov    carcolor,0111b
    call car7set
    jmp    set7end

```

```

start7:
    cmp    carcolor,0111b
    je     cover7
    jne    add7
cover7:
    mov     cx,car7x
    jmp     draw7
add7:
    cmp     car7speed,45
    ja     bigsize7
    mov     ax,car7x
    sub     ax,45
    mov     cx,ax
    mov     ax,car7speed
    sub     car7x,ax
    jmp     draw7
bigsize7:
    cmp     cx,45
    jb     Lcar7set
    sub     cx,45
    mov     car7x,cx
    call    car7
    jmp     set7end

draw7:
    mov     cary,385
    mov     dx,cary
    mov     carYcounter,45
    mov     ax,car7speed
    mov     carXcounter,ax
Lcar7set1:
    call    colortest
    WrPixel cx,dx,carcolor
    inc     dx

```

```

    dec carYcounter
    cmp  carYcounter,0
    je  Lcar7set2
    jmp  Lcar7set1
Lcar7set2:
    mov  carYcounter,45
    sub dx,45
    dec cx
    cmp  cx,0
    je  Lcar7set
    dec carXcounter
    cmp  carXcounter,0
    je  set7end
    jmp  Lcar7set1
set7end:
    ret
    car7set endp

```

car1 proc;畫初始的車子，從最左上角往右下角畫

```

    mov  cx,car1x
    mov  dx,55
    mov  carYcounter,45    ;車子大小為45*45
    mov  carXcounter,45
Lcar0:
    call colortest ;碰到青蛙
    WrPixel cx,dx,carcolor
    inc dx    ;向下畫
    dec carYcounter
    cmp  carYcounter,0
    je  Lcar1
    jmp  Lcar0
Lcar1:

```

```

mov    carYcounter,45
sub dx,45
inc cx ;向右畫
dec carXcounter
cmp    carXcounter,0
je     Lcar2
jmp     Lcar0

```

Lcar2:

```

mov    car1x,cx ;將x座標改為最右邊，因為車1 3 5 7
的移動是從右上往左下畫
ret
car1 endp

```

car2 proc;同車1，只改y座標

```

mov    cx,car2x
mov    dx,110
mov    carYcounter,45
mov    carXcounter,45

```

Lcar20:

```

call colortest
WrPixel cx,dx,carcolor
inc dx
dec carYcounter
cmp    carYcounter,0
je     Lcar21
jmp     Lcar20

```

Lcar21:

```

mov    carYcounter,45
sub dx,45
inc cx
dec carXcounter

```

```
    cmp    carXcounter,0
    je     Lcar22
    jmp     Lcar20
```

```
Lcar22:
    mov     car2x,cx
    ret
car2 endp
```

car3 proc;同車1，只改y座標

```
    mov     cx,car3x
    mov     dx,165
    mov     carYcounter,45
    mov     carXcounter,45
```

```
Lcar30:
    call colortest
    WriPixel cx,dx,carcolor
    inc dx
    dec carYcounter
    cmp     carYcounter,0
    je      Lcar31
    jmp     Lcar30
```

```
Lcar31:
    mov     carYcounter,45
    sub dx,45
    inc cx
    dec carXcounter
    cmp     carXcounter,0
    je      Lcar32
    jmp     Lcar30
```

```
Lcar32:
    mov     car3x,cx
```

```
ret
car3 endp
```

car4 proc;同車1，只改y座標

```
mov    cx,car4x
mov    dx,220
mov    carYcounter,45
mov    carXcounter,45
```

Lcar40:

```
call colortest
WrPixel cx,dx,carcolor
inc dx
dec carYcounter
cmp    carYcounter,0
je     Lcar41
jmp    Lcar40
```

Lcar41:

```
mov    carYcounter,45
sub dx,45
inc cx
dec carXcounter
cmp    carXcounter,0
je     Lcar42
jmp    Lcar40
```

Lcar42:

```
mov    car4x,cx
ret
car4 endp
```

car5 proc;同車1，只改y座標

```
mov    cx,car5x
```

```

    mov     dx,275
    mov     carYcounter,45
    mov     carXcounter,45
Lcar50:
    call colortest
    WrPixel cx,dx,carcolor
    inc dx
    dec carYcounter
    cmp     carYcounter,0
    je      Lcar51
    jmp     Lcar50
Lcar51:
    mov     carYcounter,45
    sub dx,45
    inc cx
    dec carXcounter
    cmp     carXcounter,0
    je      Lcar52
    jmp     Lcar50
Lcar52:
    mov     car5x,cx
    ret
    car5 endp

```

car6 proc;同車1，只改y座標

```

    mov     cx,car6x
    mov     dx,330
    mov     carYcounter,45
    mov     carXcounter,45
Lcar60:
    call colortest
    WrPixel cx,dx,carcolor
    inc dx
    dec carYcounter

```

```

    cmp    carYcounter,0
    je     Lcar61
    jmp     Lcar60
Lcar61:
    mov     carYcounter,45
    sub dx,45
    inc cx
    dec carXcounter
    cmp     carXcounter,0
    je     Lcar62
    jmp     Lcar60

Lcar62:
    mov     car6x,cx
    ret
    car6 endp

```

car7 proc;同車1，只改y座標

```

    mov     cx,car7x
    mov     dx,385
    mov     carYcounter,45
    mov     carXcounter,45
Lcar70:
    call colortest
    WrPixel cx,dx,carcolor
    inc dx
    dec carYcounter
    cmp     carYcounter,0
    je     Lcar71
    jmp     Lcar70
Lcar71:
    mov     carYcounter,45
    sub dx,45
    inc cx
    dec carXcounter

```



```

    cmp    carXcounter,0
    je     Lcar72
    jmp     Lcar70

```

```

Lcar72:
    mov     car7x,cx
    ret
car7 endp

```

win proc ;達到最上層時，增加遊戲難度，且重設青蛙位置

```

    mov     ax,440
    sub ax,y
    addscore,ax ;分數增加440
    addcar1speed,1 ;車速+1
    addcar2speed,1
    addcar3speed,1
    addcar4speed,1
    addcar5speed,1
    addcar6speed,1
    addcar7speed,1
    addvirusspeed,1 ;毒氣速度+1
    call delay
    ret
win endp

```

printscore proc ;分數的4位數10進制轉16進制得以印出

```

    mov     dx,0h
    mov     bx,0ah
    div bx
    mov     cl,dl

    mov     dx,0h
    mov     bx,0ah

```

```
div bx
mov    ch,dl
```

```
mov    ah,0h
mov    bl,0ah
div bl
mov    bh,ah
```

```
add al,30h
add bh,30h
add ch,30h
add cl,30h
```

```
PrintChar al
PrintChar bh
PrintChar ch
PrintChar cl
```

```
ret
printscore endp
```

```
scan    proc
mov     ah,06h
mov     dl,0ffh
int 21h
ret
scan    endp
```

```
Setback proc    ;分隔島劃法
mov     cx,0
mov     dx,45
addcounter1,8 ;共8個分隔島
```

L0:

```

    WrPixel cx,dx,1111b
    inc dx
    dec counter0 ;分隔島寬為10
    cmp counter0,0
    je L1
    jmp L0
L1:
    mov counter0,10
    sub dx,10
    inc cx
    cmp cx,641 ;分隔島從0畫到640
    je L3
    jmp L0
L3:
    mov cx,0
    add dx,55 ;下一列的分隔島
    dec counter1
    cmp counter1,0
    je L4
    jmp L0
L4:
    ret
Setback endp

```

```

colortest proc ;比較是否碰到青蛙
    mov ah,0dh
    mov bh,00h
    int 10h ;取當前位置的顏色
    cmp al,1010b ;碰到青蛙則為綠色
    je failset

```

```

colortestend:
    ret

```

```

failset:
    mov    fail,1
    mov    ax,440
    sub ax,y    ;設定分數取底部440的差直
    addscore,ax    ;設定分數
    colortest endp

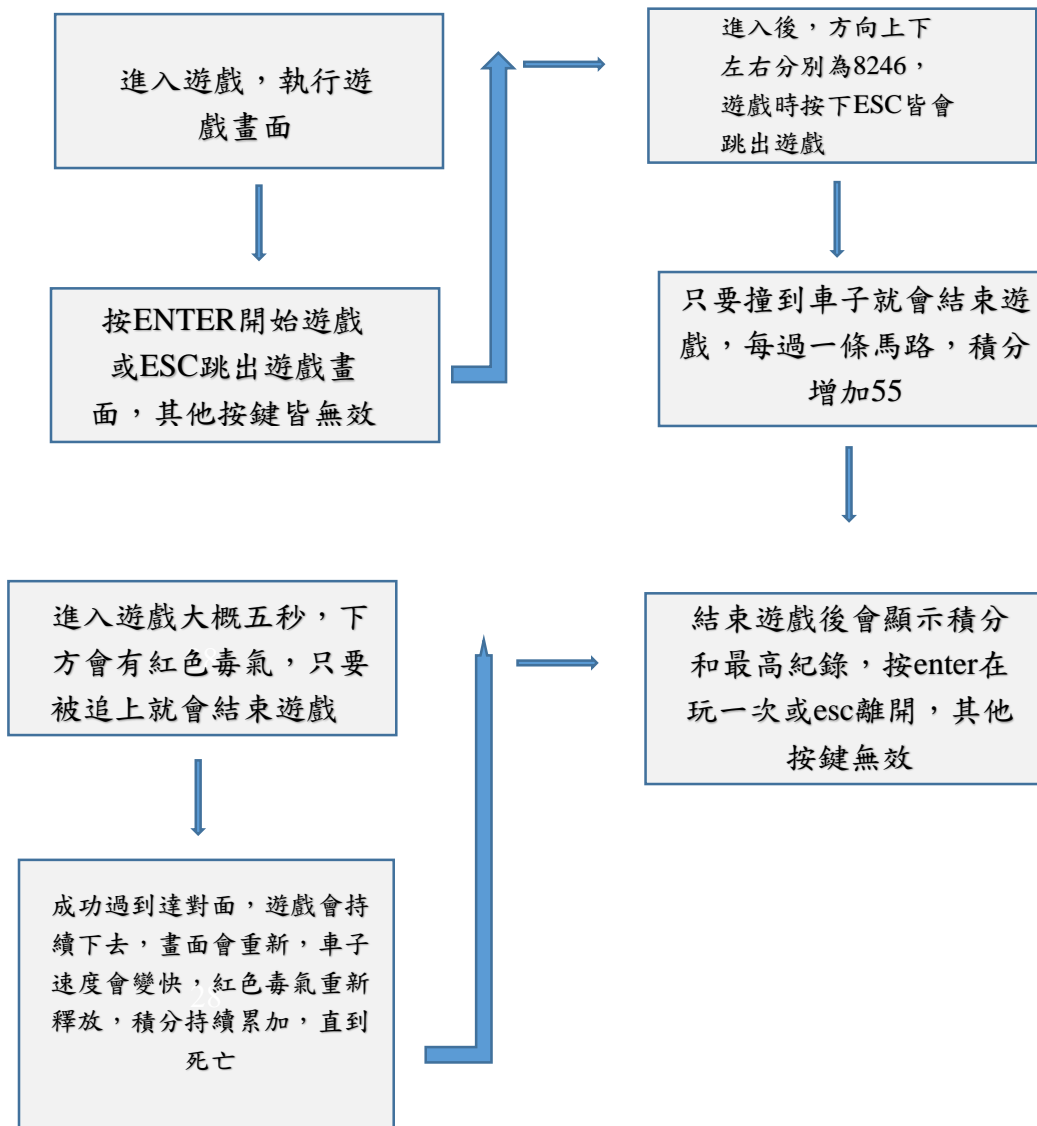
```

```

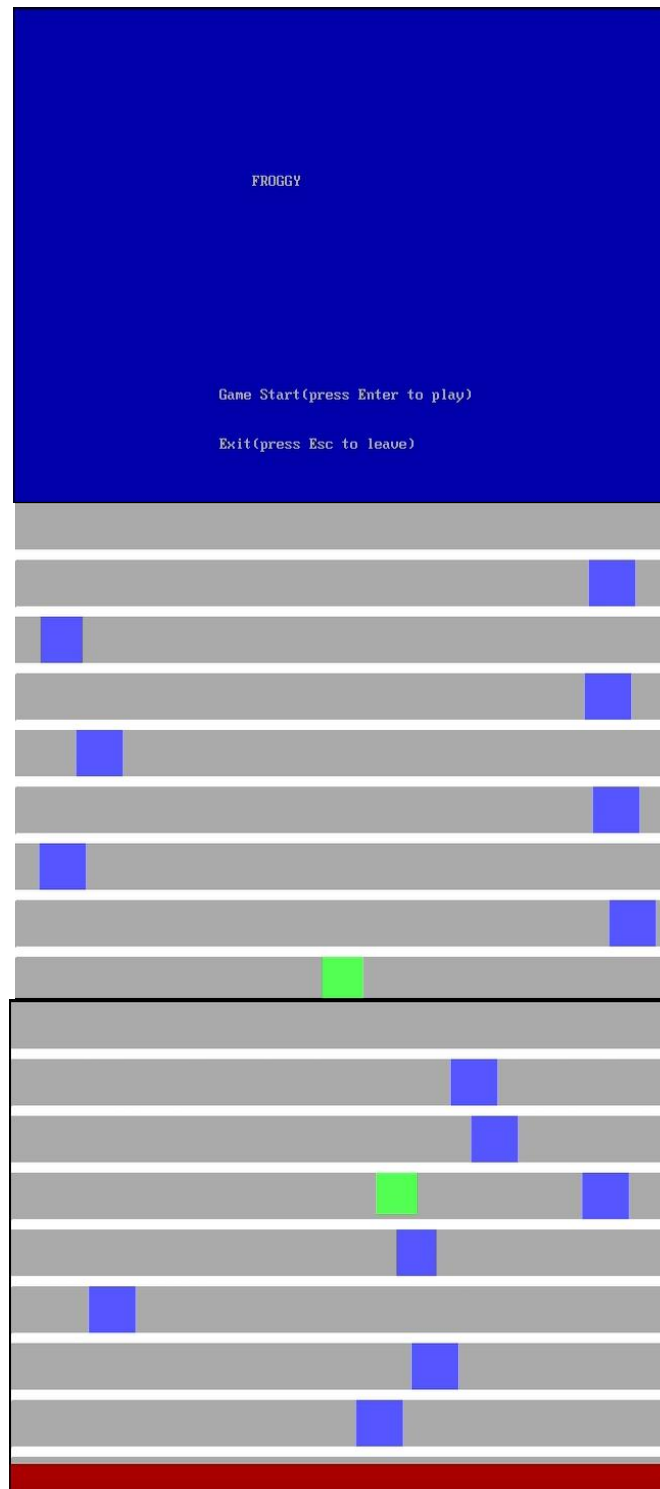
delayproc    ;delay副涵式
    mov    cx,1000h
L8:
    mov    bp,8000h
L9:
    dec bp
    cmp    bp,0
    jnz L9
    loop   L8
    ret
delay endp
end    main

```

## 二、 流程圖



### 三、實習結果





#### 四、心得

魏趨棣:

想想期初就說過期末要做出一個遊戲，結果現在也真的做出來了，組合語言真的讓我學會耐心和提升專注力，透過寫出各種指令然後讓電腦執行，真的是一件有趣的事。遊戲裡遇到的問題，做成一個遊戲的步驟，各種煩人迴圈，都讓成就感越高，也謝謝隊友大力幫忙，讓專題變得更好，也能有所進步。

李秉和:

在這次專題製作的過程中，讓我學到了按部就班的重要性，一步一步的完成目標。好比遊戲中，先畫出背景，再畫出不會動的青蛙跟車子，最後畫出可以移動的青蛙跟會動的車子與毒氣。此外在撰寫程式抵 Bug 時也讓人需要不停的思考，完成後非常的有成就感。整體遊戲的製作時，因為要反覆用到 Lable 跟 call function 和一些基礎指令，所以對基本的組合語言更加熟稔。