

組合語言與系統程式

Final Project

資電二 109504502 陳紓嫻

資電二 109504503 林柔君

資電二 109504515 陳庭煥 第 33 組

題目：阿掌養成計畫 🌵

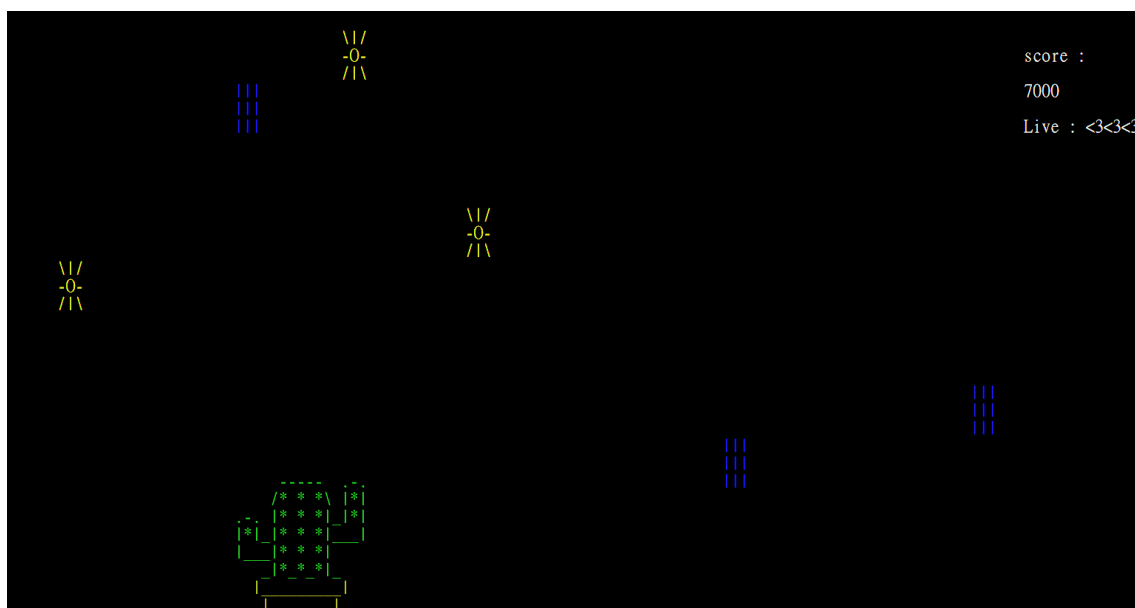
✓ 遊戲玩法

!!!!!!要記得將視窗大小調整為 145*35，不然會 gg!!!!!!

太陽和雨水會從天上隨機掉落，玩家使用左右方向鍵(←、→)操控阿掌的左右移動，因為阿掌非常喜歡照太陽☀️，所以當阿掌碰到太陽☀️時，將會獲得能量⚡️，相對的，因為阿掌極度厭惡下雨天☁️，因此如果當阿掌碰到雨水💧時，將會失去一條生命💔，達到一定的能量後，遊戲將會進入下一關，與此同時，我們可愛的阿掌將會依照關卡長大，每關都能看到不一樣的阿掌喔，而太陽和雨水的數量，將會依照關卡難度提升📈，而越來越多，除了使用左右鍵移動阿掌以外，按著 Ctrl 加左右鍵將會提升阿掌的移動速度喔(建議先按著 Ctrl 再按左右鍵才能加速)。

分數設定：第一關 3000 分→第二關 7000 分→第三關 15000 分→勝利

✓ 遊戲畫面



組合語言與系統程式

Final Project

函式介紹

阿掌設定

- `ClearBottom` ; 移除阿掌足跡
- `CreateCactus` ; 顯示阿掌第一階段的型態
- `CreateCactus2` ; 顯示阿掌第二三階段的型態
- `CactusMovement` ; 阿掌移動的判定

移除阿掌的足跡 `ClearBottom`

為了要增加阿掌行走的速度，我們多了加速這個功能，但因為加速後移動的距離會超出自身所覆寫的範圍，原本用空白加寬阿掌自身的寬度，但為了碰撞判定能夠更準確，因此採用此種方式，將會經過到的範圍的重新清空。

//Create_str 會將一排字串顯示在螢幕上

```
545 ;清掉足跡(for角色)=====
546 ClearBottom PROC USES eax ebx ecx edx
547     INVOKE Create_str,OFFSET ClearStr,0,27,color_white
548     INVOKE Create_str,OFFSET ClearStr,0,28,color_white
549     INVOKE Create_str,OFFSET ClearStr,0,29,color_white
550     INVOKE Create_str,OFFSET ClearStr,0,30,color_white
551     INVOKE Create_str,OFFSET ClearStr,0,31,color_white
552     INVOKE Create_str,OFFSET ClearStr,0,32,color_white
553     INVOKE Create_str,OFFSET ClearStr,0,33,color_white
554     INVOKE Create_str,OFFSET ClearStr,0,34,color_white
555     ret
556 ClearBottom ENDP
```

顯示阿掌第一/二三階段的型態 `CreateCactus` / `CreateCactus2`

為了版面的整潔，且因為阿掌型態不同會有不同的身高、顏色、位置，所以我們將要製作阿掌的程式碼整合成了兩種 function，程式裡都使用 `Create_str` 來呈現。

```
;畫仙人掌=====
CreateCactus PROC USES eax ebx ecx edx,
    pStr1: PTR BYTE, pStr2: PTR BYTE, pStr3: PTR BYTE,
    pStr4: PTR BYTE, pStr5: PTR BYTE, pStr6: PTR BYTE,;字串的POINTER
    Xcoord: BYTE, ;座標X
    color: DWORD ;顏色

    INVOKE Create_str ,pStr1,Xcoord,29,color
    INVOKE Create_str ,pStr2,Xcoord,30,color
    INVOKE Create_str ,pStr3,Xcoord,31,color
    INVOKE Create_str ,pStr4,Xcoord,32,color
    INVOKE Create_str ,pStr5,Xcoord,33,color_brown
    INVOKE Create_str ,pStr6,Xcoord,34,color_brown
    ret
CreateCactus ENDP
```

組合語言與系統程式

Final Project

阿掌移動的判定 CactusMovement

此程式碼中，我們使用 `readkey` 來讀取鍵盤輸入，`dx` 為判定左右鍵盤，`ah` 為判定是否有偵測到 `Ctrl`，628-632 行 `code` 為如果碰撞到雨水，阿掌身體將會短暫變成紅色的程式碼，其中使用 `color_red` 和 `color_green` 來作為顏色代碼，賦予 `eax` 顏色的值，下面使用 `.IF` 來判定 `level` 的值偵測進入到第幾關，來決定阿掌的型態，因為程式碼過長，因此擷取部分片段，625 行 `code` 為判斷是否偵測到 `ctrl + →` 的輸入狀況，635 行為印出仙人掌的部分。

```
618 ;移動仙人掌=====
619 CactusMovement PROC USES eax
620     call ReadKey
621     .IF dx == 25h && ah == 73h ;if left&ctrl are pressed
622         .IF CactusCoordinateX > 2 ;移動四個單位
623             dec CactusCoordinateX
624             dec CactusCoordinateX
625             dec CactusCoordinateX
626             dec CactusCoordinateX
627         .ENDIF
628
629         .IF CactusLoseLive==1
630             mov eax,color_red
631             mov CactusLoseLive,0
632         .ELSE
633             mov eax,color_green
634         .ENDIF
635         ;印出左仙人掌
636         .IF level == 1
637             INVOKE CreateCactus, OFFSET Cactus_left_1_1,OFF
638             OFFSET Cactus_left_1_5,OFFSET Cactus_left_1_6,C
639         .ELSEIF level == 2
640             INVOKE CreateCactus2, OFFSET Cactus_left_2_1,OF
641             OFFSET Cactus_left_2_5,OFFSET Cactus_left_2_6,0
642         .ELSE
643             INVOKE CreateCactus2, OFFSET Cactus_left_3_1,OF
644             OFFSET Cactus_left_3_5,OFFSET Cactus_left_3_6,0
645         .ENDIF
```

太陽和雨水設定

`ClearGround` ; 移除掉落物足跡

`CreateFalling` ; 生成掉落物

`refresh_fall` ; 刷新掉落物

`collision` ; 阿掌和掉落物碰撞的判定

移除掉落物足跡 ClearGround

當掉落物掉到螢幕下方時，使用字串覆蓋的方法，用空白字串將掉落物清除掉，不清掉的話會導致遊戲畫面出現問題。

```
530 ;清空底部畫面(for掉落物)=====
531 ClearGround PROC USES eax ebx ecx edx
532     INVOKE Create_str,OFFSET ClearStr,0,24,color_white
533     INVOKE Create_str,OFFSET ClearStr,0,25,color_white
534     INVOKE Create_str,OFFSET ClearStr,0,26,color_white
535     INVOKE Create_str,OFFSET ClearStr,0,27,color_white
536     INVOKE Create_str,OFFSET ClearStr,0,28,color_white
537     INVOKE Create_str,OFFSET ClearStr,0,29,color_white
538     INVOKE Create_str,OFFSET ClearStr,0,30,color_white
539     INVOKE Create_str,OFFSET ClearStr,0,31,color_white
540     INVOKE Create_str,OFFSET ClearStr,0,32,color_white
541     INVOKE Create_str,OFFSET ClearStr,0,33,color_white
542     ret
543 ClearGround ENDP
```

組合語言與系統程式

Final Project

生成掉落物 CreateFalling

我們採取 array 的方式，將掉落物的儲存在裡面，使用 Count_Fall 來決定陣列長度，就是掉落物的多寡(關卡設定那邊會決定掉落數量)，並偵測 randomVal 來決定要生成太陽或式雨水，如果掉落物 y 軸小於 0 的話，將會跳到 not_create，不生成掉落物，且 edi 加一，繼續判斷陣列的其他值，直到 ecx 為 0 跳出迴圈。

```
498 ;生成掉落物=====
499 CreateFalling PROC USES eax ebx ecx edx
500     movzx ecx,Count_Fall
501     mov edi,0
502 create_start:
503     mov bl,FallCoordY[edi]
504     cmp bl,0
505     jl not_create
506     .IF randVal[edi]==0
507         INVOKE Create_str ,OFFSET sun0,FallCoordX[edi],bl,color_yellow
508         inc bl
509         INVOKE Create_str ,OFFSET sun1,FallCoordX[edi],bl,color_yellow
510         inc bl
511         INVOKE Create_str ,OFFSET sun2,FallCoordX[edi],bl,color_yellow
512         inc bl
513         INVOKE Create_str ,OFFSET sun3,FallCoordX[edi],bl,color_yellow
514     .ELSE
515         INVOKE Create_str ,OFFSET rain0,FallCoordX[edi],bl,color_blue
516         inc bl
517         INVOKE Create_str ,OFFSET rain1,FallCoordX[edi],bl,color_blue
518         inc bl
519         INVOKE Create_str ,OFFSET rain2,FallCoordX[edi],bl,color_blue
520         inc bl
521         INVOKE Create_str ,OFFSET rain3,FallCoordX[edi],bl,color_blue
522     .ENDIF
523 not_create:
524     inc edi
525     dec ecx
526     jne create_start
527     ret
528 CreateFalling ENDP
```

刷新掉落物 refresh_fall

在太陽和雨水掉落到畫面底部時，就會呼叫此程式，將掉落物的 y 軸從新設定至最頂端，設定 eax=124 決定 RandomRange 的大小(掉落物會出現的寬度範圍)，將 al 的值設為掉落物的 x 軸，隨機決定掉落物的位置，之後呼叫 Random，隨機決定掉落物的型態。

```
833 ;刷新掉落物=====
834 refresh_fall PROC USES eax ebx ecx edx
835     mov FallCoordY[edi],-3
836     mov eax,124 ;get random 0 to 123
837     call RandomRange
838     inc eax ;make range 1 to 124
839     mov FallCoordX[edi],al
840     call ClearGround
841     call Random32
842     and al,1b ;只取個位數，隨機生成太陽or雨
843     mov randVal[edi],al
844     ret
845 refresh_fall ENDP
```

- I. al 為 0 的話，掉落物為太陽。
- II. al 為 1 的話，掉落物為雨水。

組合語言與系統程式

Final Project

阿掌和掉落物碰撞的判定 collision

FallCollisionX1、

FallCollisionX2 為掉落物 x 軸的起始位置擊結束位置，CactusCollisionX1、CactusCollisionX2 為阿掌 x 軸的起始位置擊結束位置，以此類推；如果掉落物的 x 軸落在阿掌的寬度內，將會進入下一層判斷掉落物的 y 軸位置，使用 randVal 變數來判定碰到的掉落物為太陽或雨水

- I. 如果 randVal 為 0 則為太陽，分數會加 500 分，系統就會 call WriteScore 變更分數在螢幕上
- II. 相反的，如果 randVal 為 1，生命將會減一，call WriteLive 變更螢幕上的生命血量，並將 CactusLoseLive 設為 1，在 CactusMovement 的函式中，將會使阿掌的身體短暫變為紅色提醒玩家，

最後 call refresh_fall 和 ClearGround 重新設定及清理掉落物。

```
789 ;碰撞判定=====
790 collision PROC USES EAX EBX ECX EDX
791     movzx ecx,Count_Fall
792     mov edi,0
793     start:
794     mov dl,FallCollisionX1[edi]
795     mov dh,FallCollisionX2[edi]
796     .IF dl <= CactusCollisionX2 && dl >= CactusCollisionX1
797     mov dl, FallCollisionY2[edi]
798     .IF dl <= CactusCollisionY2 && dl >= CactusCollisionY1
799     .IF randVal[edi] == 0
800         add score,500
801         call WriteScore
802     .ELSEIF randVal[edi] == 1
803         dec LiveCount
804         call Writelive
805         mov CactusLoseLive,1
806     .ENDIF
807     call refresh_fall
808     call ClearGround
809     .ENDIF
810     .ELSEIF dh <= CactusCollisionX2 && dh >= CactusCollisionX1
811     mov dh, FallCollisionY2[edi]
812     .IF dh<= CactusCollisionY2 && dh >= CactusCollisionY1
813     .IF randVal[edi] == 0
814         add score,500
815         call WriteScore
816     .ELSEIF randVal[edi] == 1
817         dec LiveCount
818         call Writelive
819         mov CactusLoseLive,1
820     .ENDIF
821     call refresh_fall
822     call ClearGround
823     .ENDIF
824     .ENDIF
825     inc edi
826     dec ecx
827     jne start
828
829     ret
830 collision ENDP
```

組合語言與系統程式

Final Project

畫面設定

`start_Animate` ; 顯示開頭動畫
`cactus_Start` ; 印出開始畫面
`LevelUP1/ LevelUP2/ LevelUP3` ; 印出換關畫面
`win/ Lose` ; 印出結束畫面
`WriteScore` ; 顯示分數
`WriteLive` ; 顯示血量
`updateXY` ; 更新移動物座標
`shining` ; 血量剩一滴血閃閃發光

顯示開頭動畫 `start_Animate`

利用判斷 `b1` 的值來決定視窗起始畫面顯示的字串，每判斷完一次，都將賦予 `b1` 下一個值 (0-3 一直重複)，`cactus_Start` 會呼叫此程式。

印出開始畫面 `cactus_Start`

最一開始先設定視窗的起始畫面，之後呼叫 `Readkey` 決定玩家是否有按空白鍵，如果 `dh` 為 20，跳出此 function，反之，呼叫 `start_Animate` 顯示動畫，一直重複 `keycheck` 迴圈，直到偵測到空白鍵按下。

```
385 ;印出開始畫面=====
386 cactus_Start PROC USES eax ebx ecx edx
387     INVOKE Create_str ,OFFSET START0,30,6,color_green
388     INVOKE Create_str ,OFFSET START1,30,7,color_green
389     INVOKE Create_str ,OFFSET START2,30,8,color_green
390     INVOKE Create_str ,OFFSET START3,30,9,color_green
391     INVOKE Create_str ,OFFSET START4,30,10,color_green
392     INVOKE Create_str ,OFFSET START5,30,11,color_green
393     INVOKE Create_str ,OFFSET START6,30,12,color_green
394     INVOKE Create_str ,OFFSET START7,30,13,color_green
395     INVOKE Create_str ,OFFSET START0,30,14,color_green
396     INVOKE Create_str ,OFFSET PressToStart1,30,20,color_white
397     INVOKE Create_str ,OFFSET PressToStart2,30,21,color_white
398     INVOKE Create_str ,OFFSET PressToStart3,30,22,color_white
399     INVOKE Create_str ,OFFSET rule,40,25,color_yellow
400     INVOKE Create_str ,OFFSET rule2,40,26,color_yellow
401
402 keycheck:
403     call Readkey
404     .IF dx ==20h ;if space is pressed
405         call Clrscr
406         ret
407     .ENDIF
408     mov ecx , 4
409     mov eax, Start_Delay ;螢幕更新速率
410     call delay
411     call start_Animate ;更新螢幕動畫
412     loop keycheck
413     ret
414 cactus_Start ENDP
```

組合語言與系統程式

Final Project

顯示分數 WriteScore

呼叫 Create_str 將分數以及分數名稱印到視窗上，每次都會清除上次的分數，以免發生覆蓋不成功的狀況。

顯示血量 WriteLive

顯示血量前，先清除上次的血量，以防覆蓋不成功發生，將 ecx 的值設為 LiveCount，決定迴圈次數(顯示血量的多寡)，如果 ecx 為 0，即跳到 no_Live 不顯示任何東西，即代表死亡，反之重複 write_all 迴圈，直到 ecx 為 0。

更新移動物座標 updateXY

一開始先設定 ecx 的值為 Count_Fall (array 大小)，將每個掉落物的高度都下降 1，如果掉落物的 y 軸的值比遊戲畫面大，即呼叫 refresh_fall，清除掉落物，將掉落物的碰撞框設為目前掉落物 x、y 軸的值，並加上掉落物的長寬，設成另外一邊的碰撞框，如果 level 為 2，將更新阿掌碰撞框的高度，level 為 3，將更新阿掌的寬度(因為阿掌會隨著 level 有不同形態，長寬就會不同)，接著設定阿掌的碰撞框。

血量剩一滴血閃閃發光 shining

如果 LiveCount 為 1 即代表血量剩最後一滴，會將畫面上血量的地方，加上 color_red_magenta 的背景色，提醒玩家剩最後一滴血，之後將 LiveShine 設為 1 (LiveCount 為 1 時，會設成 0)，之後再重複呼叫此 function 的時候，就會移除背景色(LiveCount 為 0)，就會在畫面上有閃爍的效果，進而有提醒玩家的功能。

```
741 ;更新移動物座標=====
742 updateXY PROC USES eax ebx ecx edx esi edi
743     movzx ecx, Count_Fall
744     mov edi, 0
745     start:
746         inc FallCoordY[edi] ;往下掉1
747         mov bl, FallCoordY[edi]
748         .IF bl > BRow ;掉落物掉到邊界，刷新掉落物
749             call refresh_fall
750         .ENDIF
751
752         ;設定掉落物碰撞框
753         mov dl, FallCoordX[edi]
754         mov FallCollisionX1[edi], dl
755         mov FallCollisionX2[edi], dl
756         mov dl, Fall_width
757         add FallCollisionX2[edi], dl
758
759         mov dl, FallCoordY[edi]
760         mov FallCollisionY1[edi], dl
761         inc FallCollisionY1[edi]
762         mov FallCollisionY2[edi], dl
763         mov dl, Fall_height
764         add FallCollisionY2[edi], dl
765
766         inc edi
767         dec ecx
768         jne start
769         .IF level == 2
770             mov CactusCollisionY1, 27
771         .ENDIF
772         .IF level == 3
773             mov Cactus_width, 16
774         .ENDIF
775
776         ;設定仙人掌碰撞框
777         mov dl, CactusCoordinateX
778         add dl, 2
779         mov CactusCollisionX1, dl
780         add dl, Cactus_width
781         mov CactusCollisionX2, dl
782         ret
783     updateXY ENDP
```


組合語言與系統程式

Final Project

主程式

Init ;開始遊戲

此 label 裡面主要從遊戲開始畫面 (cactus_Start) 呈現，當按下 space 後，即會跳出 cactus_Start，並清空畫面 (Clrscr)，宣告 delay 可以使我們轉變畫面時更有遊戲體驗，之後進入 level 1 的進入畫面 (LevelUP1)，當 delay 完後，進入主要遊戲畫面。

GameStart ;進入遊戲

一開始，我們先呼叫 updateXY，更新掉落物的位置，以及物品的碰撞框，每次進入 GameStart 的時候，都會判定阿掌和掉落物是否有接觸到 (collision)，之後將阿掌和掉落物印到畫面上，如果有碰撞，shining 即會讓阿掌短暫變紅色，增加遊戲體驗感。

- I. 如果 LiveCount 為 0，代表生命歸零遊戲結束，即會跳到 GameOver，印出 Gameover 的畫面，並結束遊戲。
- II. 如果遊戲分數達到 3000，即會跳到 LevelChange1，進入第二關遊戲畫面。
- III. 如果分數達到 7000，即會跳到 LevelChange2，進入第三關遊戲畫面。
- IV. 如果分數達到 15000，就會跳到 WinGame，印出 You Win 的畫面，結束遊戲。

之後持續跳回 GameStart，重複循環。

```
864 ;主程式=====
865 main PROC
866     call Randomize
867     Init:                ;開始遊戲
868     mov bl,0
869     call cactus_Start
870     call Clrscr
871     mov eax, Delay_FPS    ;螢幕更新速率
872     call delay
873     call LevelUP1
874     mov eax, LevelUP_FPS
875     call delay
876     call Clrscr
877     call WriteScore
878     call Writelive
879     GameStart:           ;進入遊戲
880     call updateXY
881     call collision
882     call ClearBottom
883     call CreateFalling
884     call CactusMovement
885     call shining
886     cmp LiveCount,0
887     je GameOver
888     cmp score,3000        ;level up
889     je LevelChange1
890     cmp score,7000
891     je LevelChange2
892     cmp score,15000
893     je WinGame            ;win
894
895     back_to_game:
896     mov eax, Delay_FPS    ;螢幕更新速率
897     call delay
898     jmp GameStart
```

```
944  GameOver:
945     mov eax,1000
946     call delay
947     mov eax, color_white
948     call Clrscr
949     call lose
950     mov eax,3000
951     call delay
952     jmp EndGame
953
954  WinGame:
955     mov eax,1000
956     call delay
957     mov eax, color_white
958     call Clrscr
959     mov eax, color_white
960     call win
961     mov eax,3000
962     call delay
963     jmp EndGame
964
965  EndGame:
966     exit
967  main ENDP
968  END main
```


組合語言與系統程式

Final Project

LevelChange1/LevelChange2 ;轉換關卡

一開始，先比較 level 大小，如果與要進入的關卡等即一樣的話，即代表已經到下一關了，因此跳回 back_to_game，反之，將 level 加 1，並加 Count_Fall 加二，增加掉落物數量，再呼叫需要的換官畫面(LevelUP2/3)，跳到 trytry。

Trytry

如果直接沒有進行迴圈，直接回到 GameStart (941 行執行完之後)，會導致遊戲的 delay 失效直到再次呼叫 trytry 的 delay(原因不明)，因此需要透過迴圈重新執行一次，因此使用 b1 作為記錄執行次數之用，當 b1 為 2 時，跳出迴圈，並清空畫面，印出分數和生命。

Reset_position ;將掉落物重置

經過此 label 前，一定會先經換轉換關卡的部分，因此當轉換完關卡的時候，都需要將掉落物移回最高點(mov FallCoordY[edi],al)，才不會導致移進入新關卡時，掉落物已經在畫面中了，之後將 al 減 10，是為了讓每個掉落物的高度都不相同，防止以同一個高度一起掉落，執行迴圈直到 array 裡每個掉落物都重新設定完，最後跳回 back_to_game，再跳回 GameStart。

```
900  LevelChange1:
901      cmp level,2
902      je back_to_game
903      inc level
904      add Count_Fall,2
905      mov bl,0
906      mov eax,color_white
907      call SetTextColor
908      call Clrscr ;清空畫面
909      call LevelUP2
910      jmp trytry
911
912  LevelChange2:
913      cmp level,3
914      je back_to_game
915      inc level
916      add Count_Fall,2
917      mov bl,0
918      mov eax,color_white
919      call SetTextColor
920      call Clrscr ;清空畫面
921      call LevelUP3
922      jmp trytry
923  trytry:
924      inc bl
925      mov eax,LevelUP_FPS
926      call delay
927      cmp bl,2
928      jne trytry
929      call Clrscr
930      call WriteScore
931      call WriteLive
932
933      mov edi,0
934      mov al,0
935      movzx ecx,Count_Fall
936  reset_position:
937      mov FallCoordY[edi],al ;把掉落物移回最高點
938      inc edi
939      sub al,10
940      loop reset_position
941      jmp back_to_game
942
```

組合語言與系統程式

Final Project

引入函式庫

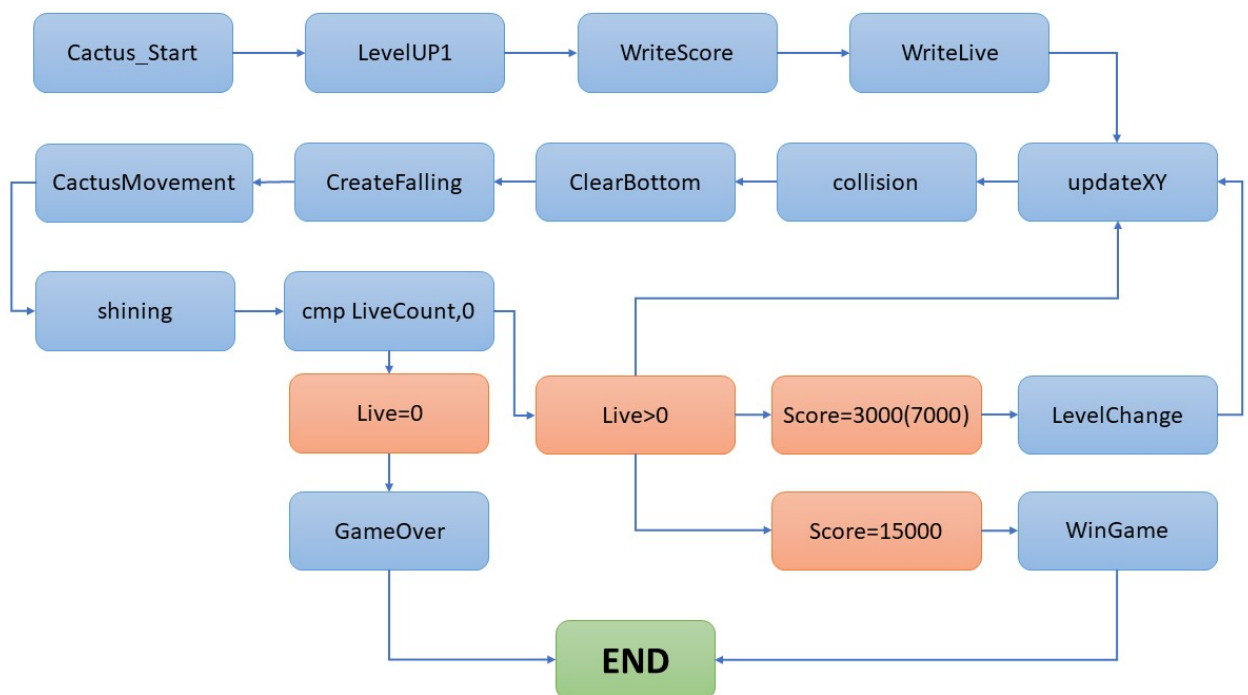
FROM Irvine Library

Name	Description
Clrscr	Clears the screen by writing blanks to all positions.
Delay	Delays (pauses) the current process for a given number of milliseconds.
ReadKey	Performs a no-wait keyboard check and reads a single character if available.
Readchar	Reads a single character from standard input and returns the character in the AL register.
SetTextColor	Changes the color of all subsequent text output.
Gotoxy	Locates the cursor at a given row and column in the screen's console buffer. (DL = column DH = row)
WriteString	Writes a null-terminated string to standard output.
WriteDec	Writes an unsigned 32-bit decimal number to standard output in decimal format with no leading zeros.
Randomize	Re-seeds the random number generator with the current time in hundredths of seconds.

組合語言與系統程式

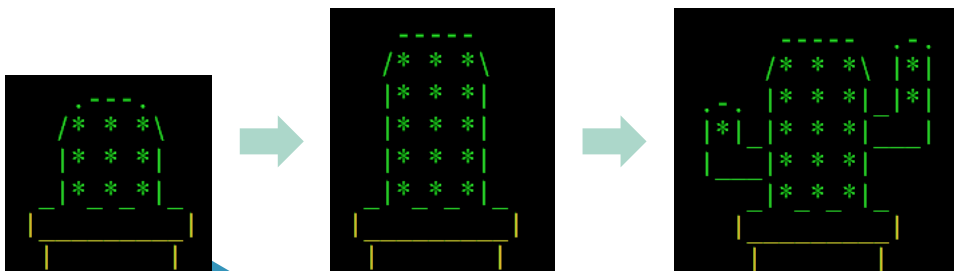
Final Project

流程圖

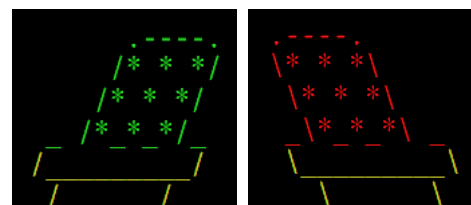


實際 DEMO 截圖

阿掌的成長歷史 🌵



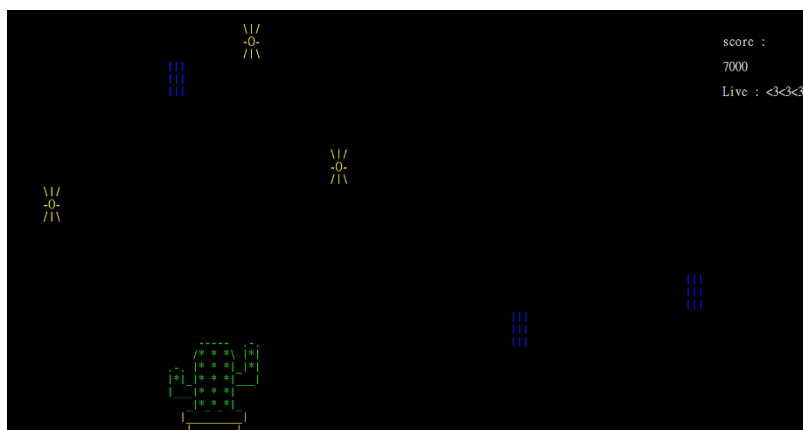
會左右搖擺的阿掌 aka 剛被擊中的它



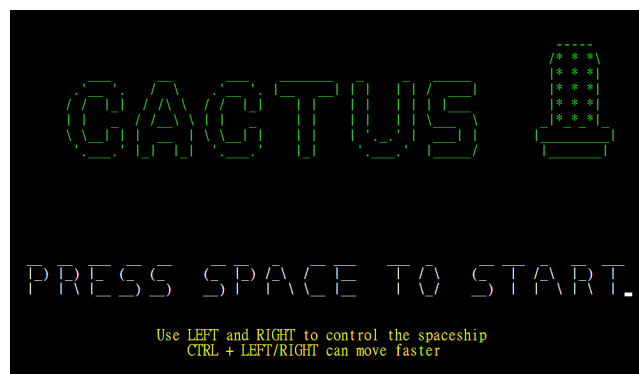
組合語言與系統程式

Final Project

實際遊玩畫面



遊戲開始畫面



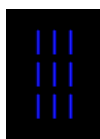
關卡切換畫面



遊戲結束(輸贏)畫面



這是雨水



這是太陽





組合語言與系統程式

Final Project

組員分工表

- ✧ 109504502 陳紓嫻：仙人掌設定、碰撞判定、debug、口頭報告、書面報告、整合
- ✧ 109504515 陳庭煥：掉落物設定、主程式、debug、血量、分數、整合
- ✧ 109504503 林柔君：開頭動畫、UI 設計、ppt 製作