

Computer Organization and Assembly Language

Final Project

RACING

B96902072 陳祥文

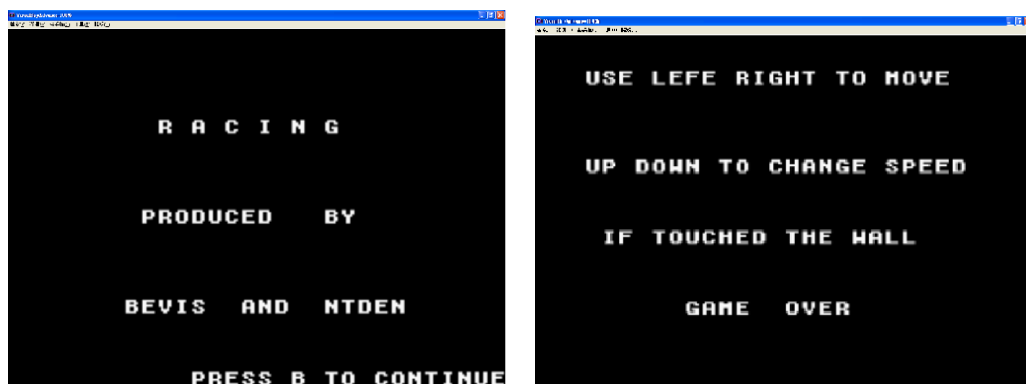
B96902096 鄧祐承

遊戲簡介

在童年時期，放學後或者閒暇時間，常常會呼朋引伴一起玩遊戲，除了玩紙上的遊戲、大地遊戲，也曾經玩過不少的電玩；而其中，就包含了賽車遊戲，不論是多人PK搶第一名的，或者是玩家與電腦PK搶第一的，或者玩家一直往前衝，追求累積積分更高的，總是讓我們熱血沸騰，心情隨著被超前而緊張，一路領先而興奮；而賽車遊戲，也一直不斷的在翻新，從以前的單機版，到現在的跑跑卡丁車可以讓玩家線上對戰，可說仍然在蓬勃的發展。

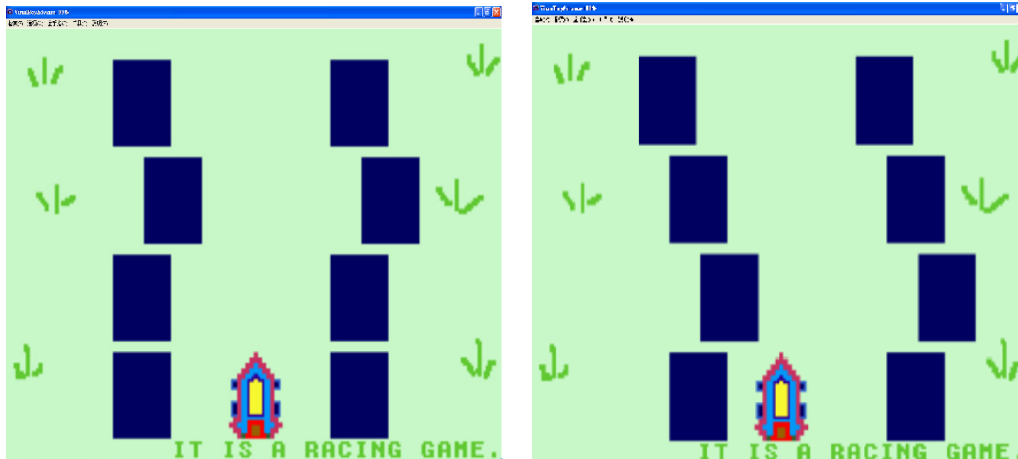
設計概述

本遊戲的的實作配置並不是玩家之間PK，也不是玩家與電腦爭奪第一名，而是偏向於自我挑戰，在兩側不斷變化的軌道之中，靠著敏銳的反應神經以及動作迅速的手指，避開迎頭撞上兩側牆壁，繼續前進。

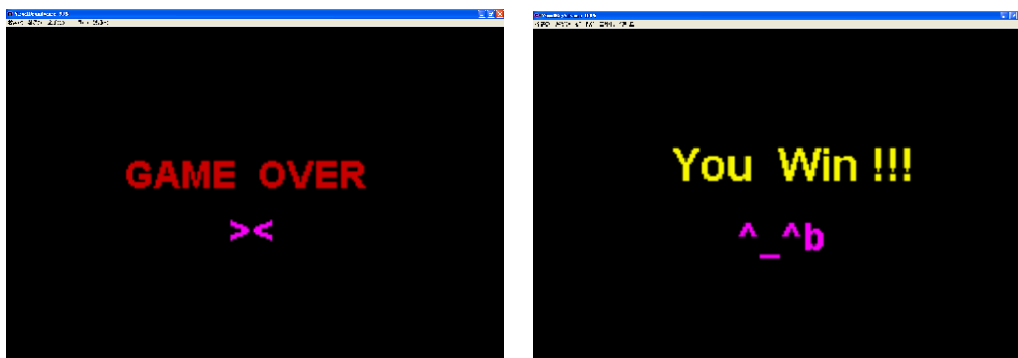


遊戲一開始，玩家的車會在畫面下方中間出現，而兩旁會有兩道牆

壁，一開始道路會是直線的，幾格之後，道路會開始左右的一次以一格隨機移動，若撞上牆壁，則遊戲就結束了。



而玩家在遊戲中，可以用方向鍵控制車體左右移動，閃躲牆壁；如果是初學者，可以按”下”減緩車體的前進速度；如果覺得普通模式太過簡單，也可以按”上”增加車體前進的速度，挑戰較高的難度。如果開始一段時間之後都沒撞上牆壁，則最終會顯示” You win . ”



實作內容

遊戲的架構部分採用了 HAM Library，而整個遊戲並不複雜，主要可

以分成三個部份，第一部份是開頭的文字以及遊戲的背景，以及最後面的 GAME OVER 以及 YOU WIN 這些較小的設定，文字的部分主要使用內建的函式 `ham_InitText()` 以及 `ham_DrawText(, , "")`; 而圖片則是使用點陣圖(.bmp)再經過 `gfx2gba` 這個程式進行編碼然後讀取。

第二部份是圖像的移動，這部分主要也有兩個物件，一個是車子的左右移動跟軌道的左右移動，一個是按住上下鍵時的加速跟減速，這部份需要對於上下左右按鍵做出回應，左右時候需要移動整個 `sprite`，上則是兩側軌道的 `sprite` 左右移動速度加快，做出車速加快的感覺，而下，則是兩側軌道移動減慢，做出減速的感覺。

第三部份則是是否過關的判段，對於車子以及兩側的軌道隨時進行判斷，如果車子的座標與軌道的不同，則表示安全，如果過於接近，則表示碰撞了，遊戲就結束了，並做一個計時器，當到達一定時間之前，遊戲都進行，若時間到了遊戲尚未終止，則會跳進獲勝的畫面。

一學期下來的組合語言(by 陳祥文)

我認為組合語言的優點真的很多,空間效率性,時間效率性,容易針對硬體做編寫等,和擁有可攜性,易維持性,易發展性的高階語言各有各的特色與存在的必要。

這學期也藉著組合語言和機器與嚴密不可分,接近逐字翻的特性,讓我們也對原本認為遙不可及的機器語言有片面的了解。從 Toy_machine, ARM到 Intel_IA32, 我雖然不是個成績很好的學生,卻也能從作業的練習裡吸收一些組合語言把C語言optimal的技巧,收獲匪淺。

期末開始大趕工的期末專題,最後以一個賽車遊戲呈現。我看過其他別組的同學的作品都很厲害,卻絲毫不會動搖我把作品完成時的那份喜悅。以前的我完全不相信自己有能力去寫遊戲,如今因為期末project的壓力以及開發HAM人員的強大,我成功的做出了我人生第一個視窗遊戲,真的很開心。

不過因為程式架構過於簡單,且時間因素不容許再增加一些其他東西,這次的作業沒有真正用到組合語言,十分得可惜,為來我還想利用空暇繼續改進這款遊戲,目前最想做的是把背景的圖可以隨著賽車的前進而移動或做改變,這部分也是我認為比較適合用組語的地方。

心得(by 鄧祐承)：

這次的 FINAL PROJECT 讓我對於程式設計有了不同的體驗，以往在寫 C 跟 JAVA 的時候，總是輸入數字，輸出答案，玩全沒有接觸到關於圖像的部份，而這次的 PROJECT，從我們經常使用到的 GBA 模擬器下手，讓我們實地了解他的設計，也讓我對於使用圖檔有了比較詳細的認知；而這次做出來的東西，雖然很陽春，可是卻讓我充滿成就感，雖然對於強者來說，可能動動幾根手指頭就完成了，可是對我來說，卻是從無到有，從 0 到現在的，整個一系列的學習，也讓我對於程式語言有了比較多的了解。

參考資料：

[http://www.csie.ntu.edu.tw/~cyy/courses/assembly/08fall/assignments/fin
al/reports/](http://www.csie.ntu.edu.tw/~cyy/courses/assembly/08fall/assignments/final/reports/)

<http://www.aaronrogers.com/ham/Day6/day6.php>

附件(原始碼):

```
#include <mygba.h>

#include <stdlib.h>

#include "gfx/gr.raw.c"

#include "gfx/car.pal.c"

#include "gfx/gamebg.pal.c"

#include "gfx/gameover.raw.c"

#include "gfx/gameover.pal.c"

#include "gfx/gameover.map.c"

#include "gfx/youwin.raw.c"

#include "gfx/youwin.pal.c"

#include "gfx/youwin.map.c"

#include "gfx/car.raw.c"

#include "gfx/gamebg.raw.c"

#include "gfx/gamebg.map.c"

#include "gfx/block/block1_L.raw.c"

#include "gfx/block/block2_L.raw.c"

#include "gfx/block/block3_L.raw.c"

#include "gfx/block/block4_L.raw.c"

#include "gfx/block/block1_R.raw.c"

#include "gfx/block/block2_R.raw.c"
```

```
#include "gfx/block/block3_R.raw.c"

#include "gfx/block/block4_R.raw.c"

#include "gfx/block/block.pal.c"

void gameover();

void youwin();

u8 thecar;

u8 car_x = 105 ;

// car X 座標

u8 car_y = 120 ;

// car Y 座標

int vbl_count;

int i;

int speed=30;

u8 block1_L; u8 block2_L; u8 block3_L; u8 block4_L;

u8 block1_R; u8 block2_R; u8 block3_R; u8 block4_R;

u8 block1_L_X=50; u8 block1_R_X=155;

u8 block1_L_Y=12; u8 block1_R_Y=12;

u8 block2_L_X=50; u8 block2_R_X=155;

u8 block2_L_Y=48; u8 block2_R_Y=48;

u8 block3_L_X=50; u8 block3_R_X=155;
```



```
u8 block3_L_Y=84; u8 block3_R_Y=84;

u8 block4_L_X=50; u8 block4_R_X=155;

u8 block4_L_Y=120; u8 block4_R_Y=120;

void vbl_func();

// VBL function

void query_buttons();

// 檢測輸入

void update_car();

// 更新座標

void vbl_func();

int main(void){

    ham_Init();

    ham_InitText(2);

    int kl=0; //temp

    ham_DrawText( 9 , 5 , "R A C I N G" );

    while(kl<1000000) kl++;
```

```

ham_DrawText( 8 , 10 , "PRODUCED BY" );

kl=0;

while(kl<1000000) kl++;

ham_DrawText( 7 , 15 , "BEVIS AND BTDEN" );

kl=0;

while(kl<1000000) kl++;

ham_DrawText( 11 , 19 , "PRESS B TO CONTINUE" );

while ( 1 ) {

    if (F_CTRLINPUT_B_PRESSED) break;

}

ham_DrawText( 9 , 5 , "
");
ham_DrawText( 8 , 10 , "
");
ham_DrawText( 7 , 15 , "
");
ham_DrawText( 11 , 19 , "
");

ham_DrawText( 3 , 2 , "USE LEFE RIGHT TO MOVE" );

ham_DrawText( 3 , 7 , "UP DOWN TO CHANGE SPEED" );

ham_DrawText( 4 , 11 , "IF TOUCHED THE WALL" );

kl=0;

while(kl<1000000) kl++;

ham_DrawText( 9 , 15 , "GAME OVER " );

```

```
kl=0;

while(kl<1000000) kl++;/*

ham_DrawText( 14 , 19 , "PRESS A TO START" );

while (1){

    if (F_CTRLINPUT_A_PRESSED) break;

}*/

ham_DrawText( 3 , 2 , "                " );

ham_DrawText( 3 , 7 , "                " );

ham_DrawText( 4 , 11 , "                " );

ham_DrawText( 9 , 15 , "                " );

ham_DrawText( 14 , 19 , "                " );
```

```
// 變數
```

```
map_fragment_info_ptr bg_gamebg;
```

```
// BG pointer
```

```
// 設置背景模式
```

```
ham_SetBgMode( 1 );

// 初始化背景調色板

ham_LoadBGPal(( void *)gamebg_Palette, 256 );

// 為圖片設置 tileset

ham_bg[ 0 ].ti = ham_InitTileSet(( void *)gamebg_Tiles,
SIZEOF_16BIT(gamebg_Tiles), 1 , 1 );

// 為圖片設置地圖

ham_bg[ 0 ].mi = ham_InitMapEmptySet( 3 , 0 );
bg_gamebg = ham_InitMapFragment(( void *)gamebg_Map,
30 , 20 , 0 , 0 , 30 , 20 , 0 );
ham_InsertMapFragment(bg_gamebg, 0 , 0 , 0 );

// 顯示背景

ham_InitBg( 0 , 1 , 1 , 0 );

ham_DrawText( 10 , 19 , "It is a racing game." );

ham_LoadObjPal(( void *)car_Palette, 256 );
thecar=ham_CreateObj(( void *)car_Bitmap, OBJ_SIZE_32X32 ,
OBJ_MODE_NORMAL, 1 , 0 , 0 , 0 , 0 , 0 , 105 , 120 );
```

```
//ham_LoadObjPal(( void *)block_Palette, 256 );  
  
block1_L=ham_CreateObj(( void *)block1_L_Bitmap,  
OBJ_SIZE_32X32 ,  
  
OBJ_MODE_NORMAL, 1 , 0 , 0 , 0 , 0 , 0 , 0 , block1_L_X ,  
block1_L_Y );
```

```
//ham_LoadObjPal(( void *)block_Palette, 256 );  
  
block2_L=ham_CreateObj(( void *)block2_L_Bitmap,  
OBJ_SIZE_32X32 ,  
  
OBJ_MODE_NORMAL, 1 , 0 , 0 , 0 , 0 , 0 , 0 , block2_L_X ,  
block2_L_Y );
```

```
//ham_LoadObjPal(( void *)block_Palette, 256 );  
  
block3_L=ham_CreateObj(( void *)block3_L_Bitmap,  
OBJ_SIZE_32X32 ,  
  
OBJ_MODE_NORMAL, 1 , 0 , 0 , 0 , 0 , 0 , 0 , block3_L_X ,  
block3_L_Y );
```

```
//ham_LoadObjPal(( void *)block_Palette, 256 );  
  
block4_L=ham_CreateObj(( void *)block4_L_Bitmap,  
OBJ_SIZE_32X32 ,  
  
OBJ_MODE_NORMAL, 1 , 0 , 0 , 0 , 0 , 0 , 0 , block4_L_X ,  
block4_L_Y );
```

```
//ham_LoadObjPal(( void *)block_Palette, 256 );  
  
block1_R=ham_CreateObj(( void *)block1_R_Bitmap,  
OBJ_SIZE_32X32 ,  
  
OBJ_MODE_NORMAL, 1 , 0 , 0 , 0 , 0 , 0 , 0 , block1_R_X ,  
block1_R_Y );
```

```
//ham_LoadObjPal(( void *)block_Palette, 256 );  
  
block2_R=ham_CreateObj(( void *)block2_R_Bitmap,  
OBJ_SIZE_32X32 ,  
  
OBJ_MODE_NORMAL, 1 , 0 , 0 , 0 , 0 , 0 , 0 , block2_R_X ,  
block2_R_Y );
```

```
//ham_LoadObjPal(( void *)block_Palette, 256 );  
  
block3_R=ham_CreateObj(( void *)block3_R_Bitmap,  
OBJ_SIZE_32X32 ,  
  
OBJ_MODE_NORMAL, 1 , 0 , 0 , 0 , 0 , 0 , 0 , block3_R_X ,  
block3_R_Y );
```

```
//ham_LoadObjPal(( void *)block_Palette, 256 );  
  
block4_R=ham_CreateObj(( void *)block4_R_Bitmap,  
OBJ_SIZE_32X32 ,  
  
OBJ_MODE_NORMAL, 1 , 0 , 0 , 0 , 0 , 0 , 0 , block4_R_X ,  
block4_R_Y );
```

```
ham_CopyObjToOAM();

vbl_count=0; //歸零

ham_StartIntHandler(INT_TYPE_VBL,( void *)&vbl_func);
//ham_StartIntHandler(INT_TYPE_VBL,( void *)&vbl_func);

// 無限迴圈

//End:

while (1) {

if (F_CTRLINPUT_A_PRESSED) break;

}

return 0 ;

} // End of main()

void vbl_func(){

vbl_count++;

if(vbl_count==600)

    youwin();

ham_CopyObjToOAM();

query_buttons();

if(block4_L_X>(car_x-20)||block4_R_X<(car_x+20)){
```

```
    gameover();
}

update_car();

if (vbl_count%speed==1){

    block4_L_X=block3_L_X;

    block3_L_X=block2_L_X;

    block2_L_X=block1_L_X;

    block4_R_X=block3_R_X;

    block3_R_X=block2_R_X;

    block2_R_X=block1_R_X;

    i=block1_L_X+15*(rand()%3-1);

    while (i<10||i+105>200)

    i=block1_L_X+15*(rand()%3-1);

    block1_L_X=i; block1_R_X=(i+105);

    ham_SetObjX(block1_L,block1_L_X);

    ham_SetObjX(block1_R,block1_R_X);

    ham_SetObjX(block2_L,block2_L_X);

    ham_SetObjX(block2_R,block2_R_X);

    ham_SetObjX(block3_L,block3_L_X);

    ham_SetObjX(block3_R,block3_R_X);
```



```
        ham_SetObjX(block4_L,block4_L_X);
        ham_SetObjX(block4_R,block4_R_X);
    }

return ;
}

void query_buttons() {

    if (F_CTRLINPUT_LEFT_PRESSED){
        if (car_x > 0 )
            car_x--;
        ham_DrawText( 1 , 19 , "Left " );
    }
    if (F_CTRLINPUT_RIGHT_PRESSED){
        if (car_x < 208 )
            car_x++;
        ham_DrawText( 1 , 19 , "Right" );
    }
    if (F_CTRLINPUT_DOWN_PRESSED){
        speed=speed+3;
    }
}
```

```

if (F_CTRLINPUT_UP_PRESSED){
    if (speed>=6 )
        speed=speed-3;
    }
return ;
}

void update_car() {
    ham_SetObjX(thecar,car_x);
    ham_SetObjY(thecar,car_y);
    return ;
}

void youwin(){
    ham_StopIntHandler(INT_TYPE_VBL);
    ham_ResetObj();
    ham_ResetBg();
    map_fragment_info_ptr bg_youwin;
    ham_LoadBGPal(( void *)youwin_Palette, 256 );
    ham_bg[ 1 ].ti = ham_InitTileSet(( void *)youwin_Tiles,
    SIZEOF_16BIT(youwin_Tiles), 1 , 1 );

    ham_bg[ 1 ].mi = ham_InitMapEmptySet( 3 , 0 );
    bg_youwin = ham_InitMapFragment(( void *)youwin_Map,

```

```

    30 , 20 , 0 , 0 , 30 , 20 , 0 );

    ham_InsertMapFragment(bg_youwin, 1 , 0 , 0 );

    ham_InitBg( 1 , 1 , 0 , 0 );

}

void gameover(){

    ham_StopIntHandler(INT_TYPE_VBL);

    ham_ResetObj();

    ham_ResetBg();

    map_fragment_info_ptr bg_gameover;

    ham_LoadBGPal(( void *)gameover_Palette, 256 );

    ham_bg[ 1 ].ti = ham_InitTileSet(( void *)gameover_Tiles,
    SIZEOF_16BIT(gameover_Tiles), 1 , 1 );

    ham_bg[ 1 ].mi = ham_InitMapEmptySet( 3 , 0 );

    bg_gameover = ham_InitMapFragment(( void *)gameover_Map,
    30 , 20 , 0 , 0 , 30 , 20 , 0 );

    ham_InsertMapFragment(bg_gameover, 1 , 0 , 0 );

    ham_InitBg( 1 , 1 , 0 , 0 );

}

```