

2009年1月20日

COMPUTER ORGANIZATION AND ASSEMBLY LANGUAGES FINAL PROJECT



資工二

B96902044 彭嫻晨

B96902060 張力兒

B96902083 陳亭汝

動機

在知道組合語言期末報告可以自己寫小遊戲之後，大家都在討論著到底做什麼遊戲最有趣，而且可讓玩家快速明瞭遊戲內容且仍可達到娛樂效果。

也許是都有玩過打地鼠的經驗，經過熱烈討論後，決定以打地鼠為基底，製作和打地鼠相似的小遊戲，但介面和圖形都由自己繪出，希望能達到輕鬆、易懂、娛樂的效果。

遊戲介紹

"MOLE HITTING !!"是一款和傳統打地鼠相似的遊戲，但"MOLE HITTING !!"裡的"地鼠"將不再只是"地鼠"，有可愛的兔子、乖巧的綿羊、淘氣的青蛙，每種動物還以兩種不同的面貌出現，增添遊戲趣味。

有別於以往地鼠只在固定的洞出現，我們的"地鼠"會在不定點冒出，每個洞都是會移動的，且某些特定的"地鼠"會因玩家的控制鍵而移動，使得玩家好像永遠都追不上"地鼠"。而某些"地鼠"打擊後會得到分數，某些則會失去分數。此外，遊戲中會不定時出現一隻蝴蝶飛過，若玩家的槌子被蝴蝶觸碰到，槌子就會被蝴蝶帶走，玩家就輸了。

遊戲規則

1. 玩家的目標是打擊奇怪顏色的動物，打到可以加分，若打到正常的動物則會扣分。
2. 若玩家的分數低於零分，或槌子被蝴蝶帶走，玩家就輸了。
3. 若分數到達20分，玩家就贏了。

控制鍵

上	up	左	left
下	down	右	right
打擊	z		



原理

· 選單

在開啟視窗之後，會出現一個遊戲的歡迎畫面，由void welcome()實做，當中有一個無限迴圈，玩家按下ENTER鍵後才會終止：

```
void welcome(){
    :
    :
    :
    while(1){
        //讓圖片顯示出來
        //玩家若按下ENTER鍵 則會跳
        出整個函式
        if(F_CTRLINPUT_START_PRESSED){
            ham_DeInitBg(0); //將背景刪除
            return;
        }
    }
}
```

跳出函式後則回到主程式，進入選單，玩家會看到一張選單，可以選擇HOW TO PLAY看遊戲規則，或是直接玩遊戲，由於選單可以上下選擇，所以選單主要是由兩個函式控制，select()和select2()，遊戲規則頁面則是由instr()控制，並且可以回到選單，所以整個選單的控制也是由一個無限迴圈包住：

```
while(1){
    select();
    if(mode == 1){
        mode = 0;
        select2();
    }
    if(go != 0) //如果玩家選擇進行遊戲，則跳出此迴圈進入遊戲
        break;
    if(mode == 2){
        mode = 0;
        instr();
    }
}
```



原理

· 地圖

在遊戲中，我們所用的背景採地圖模式，就是圖片較大（512*512）會超過整個螢幕，而玩家可以利用up, down, right或是left控制地圖，由於地圖移動時sprite，也就是地鼠，並不會跟著移動，所以當地圖移動一像素時，我們設定sprite也會向相反的方向移動一格：

```

if(F_CTRLINPUT_DOWN_PRESSED){ //按向下鍵
    if(map_y < (512-160)){ //如果還沒頂到地圖下端
        ham_SetBgXY(1,map_x,++map_y); //移動地圖
        plane_y--; //往反方向移動sprite
        frog_y--;
        sheep_y--;
        planebad_y--;
        frogbad_y--;
        sheepbad_y--;
        :
        :
        : //更新sprite位置
    }
}

```

雖然這樣的寫法會產生一個問題，就是sprite的位置會被減到變負的，或是被加到超過邊界時，但是對我們來說，這反而是一個很方便的點，因為當sprite的位置超過邊界到一定的程度之後，會像捲軸一樣又回到畫面中，所以不管玩家移動到地圖的哪個位置，都會有sprite出現，因此我們不用創造出非常多sprite來填滿地圖，就可以營造出到處都有地鼠可以打的情境。



原理

· 地鼠跳動與移動

我們的地鼠並不只會待在原地，有的地鼠會因玩家某些特定方向的按鍵而向相同的方向移動，所以看起來玩家似乎會追不到地鼠：

```
if (F_CTRLINPUT_LEFT_PRESSED && !F_CTRLINPUT_UP_PRESSED
&& !F_CTRLINPUT_DOWN_PRESSED){
    if (hammer_x > 0 ) hammer_x--;
    dir_hammer = ANIM_UP;
    update_hammer_pos();
    update_hammer_gfx();
    sheep_x -= 3; //地鼠朝同方向移動
    sheep_y -= 3;
    update_sheep_pos();
}
```

另外，當玩家打到地鼠時，我們也設定地鼠會立刻跳開：

```
if( (hammer_x >= plane_x - 32 && hammer_x <= plane_x + 64) &&
(hammer_y >= plane_y - 32 && hammer_y <= plane_y + 10) &&
dir_plane == ANIM_RIGHT){
    dir_plane = ANIM_LEFT; //地鼠顯示死掉圖片
    point++; //加分
    ham_DrawText(20,2,"Point: %d",point); //顯示分數
    jump(&plane_x, &plane_y); //地鼠跳開
    update_plane_gfx(); //更新地鼠圖片
    update_plane_pos(); //更新地鼠位置
}
```



原理

· 蝴蝶移動

遊戲進行中，會有蝴蝶不定時的出現，蝴蝶的路徑由其X坐標決定其下一步的前進方向：

```

if(bfly_x>=0&&bfly_x <=20){
    bfly_x ++;      //蝴蝶前進
    bfly_y ++;
    update_bfly_pos();
    if(flag==1){   //如果蝴蝶抓到槌子
        hammer_x = bfly_x;
        hammer_y = bfly_y;
        update_hammer_pos();
        return;
    }
}
    
```

蝴蝶前進的路徑因此看起來是隨機的，另外，蝴蝶抓到槌子時，槌子會跟蝴蝶一起前進。



遇到的困難

製作動畫時，連續圖的變化速度需在人眼可辨識的範圍內，原先使用for迴圈增圖片與圖片之間的變換時間，但for迴圈使用太多時，遊戲會超出負荷而無法繼續運作。最後利用了ham_StartIntHandler每六十分之一秒呼叫一次函式的功能，設定一個flag，每呼叫一次即加一，當flag為特定數字時再執行變換圖片，藉此控制圖片變換的速度。

參考資料

HAM Documentation

<http://www.aaronrogers.com/ham/>

心得

還在討論階段，我們決定要用GBA寫小遊戲，因為我們從來沒接觸過GBA的函式，感覺好像很困難。我們就這樣一邊摸索一邊寫程式，也慢慢在誤打誤撞中學會了如何做出介面，控制圖片等等功能，然後再一步一步做出各種各樣我們想要的功能，漸漸也有了信心。

等到最後從開頭的選單到最後的結束畫面都能完整呈現的時候，真的覺得很有成就感，尤其是看到別人是玩我們的遊戲的時候，因為我們遊戲中的功能或小惡作劇驚呼一聲時，就覺得熬夜的痛苦和寫不出來的挫折都是值得的！大概以後我們如果遇到朋友，都會請他們玩我們的遊戲吧！

最後要感謝教授這一個學期下來指導了我們這麼多東西，期末別出心裁的出了一個小遊戲的作業，讓我們在寫程式無可避免的爆肝中，多了一點小樂趣，還有要感謝在系館一起同甘共苦同學們，在我們為一個問題卡住時，耐心的提供解答，節省許多摸索的時間，還有還有，最後感謝Aaron Rogers，雖然我們不曾碰面，但是他在網路上提供的資訊，卻讓我們受益無窮，感謝大家！

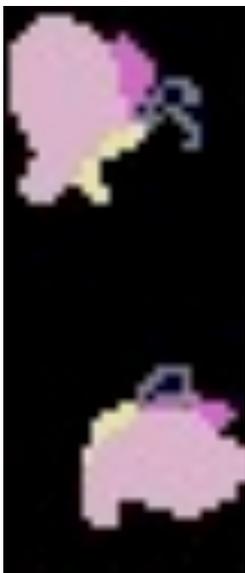
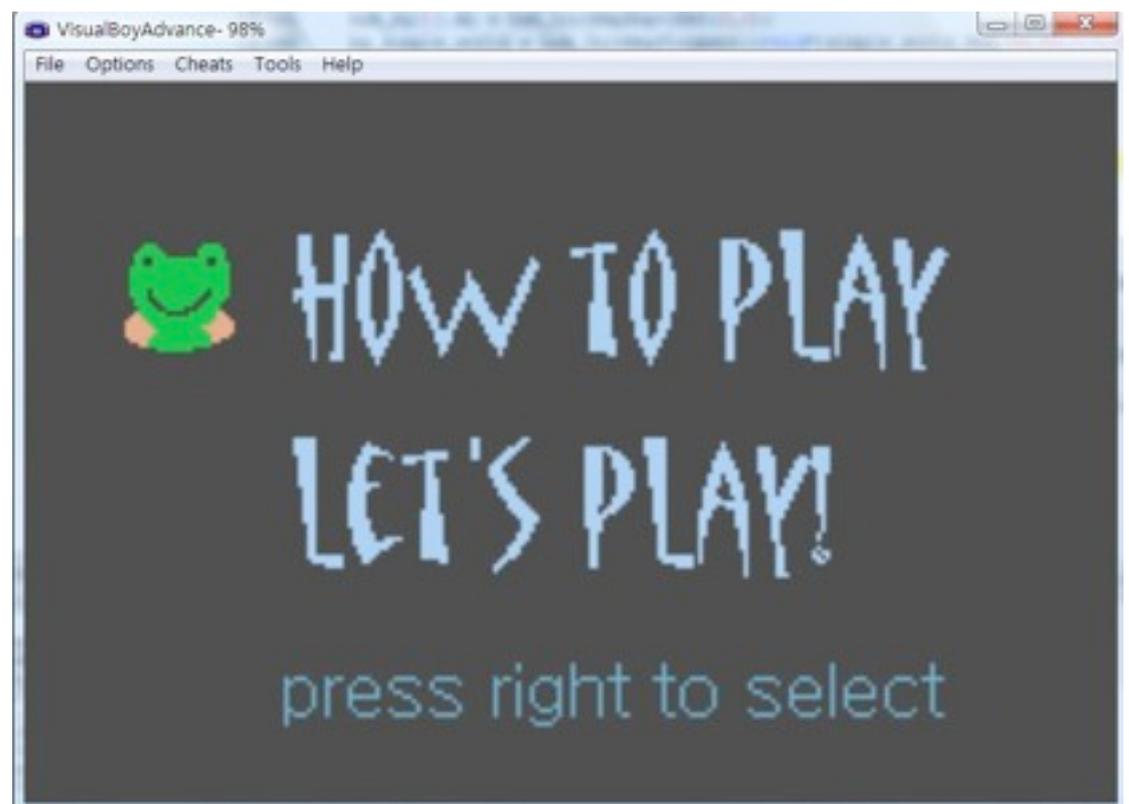


畫面擷取

開始畫面



遊戲選單

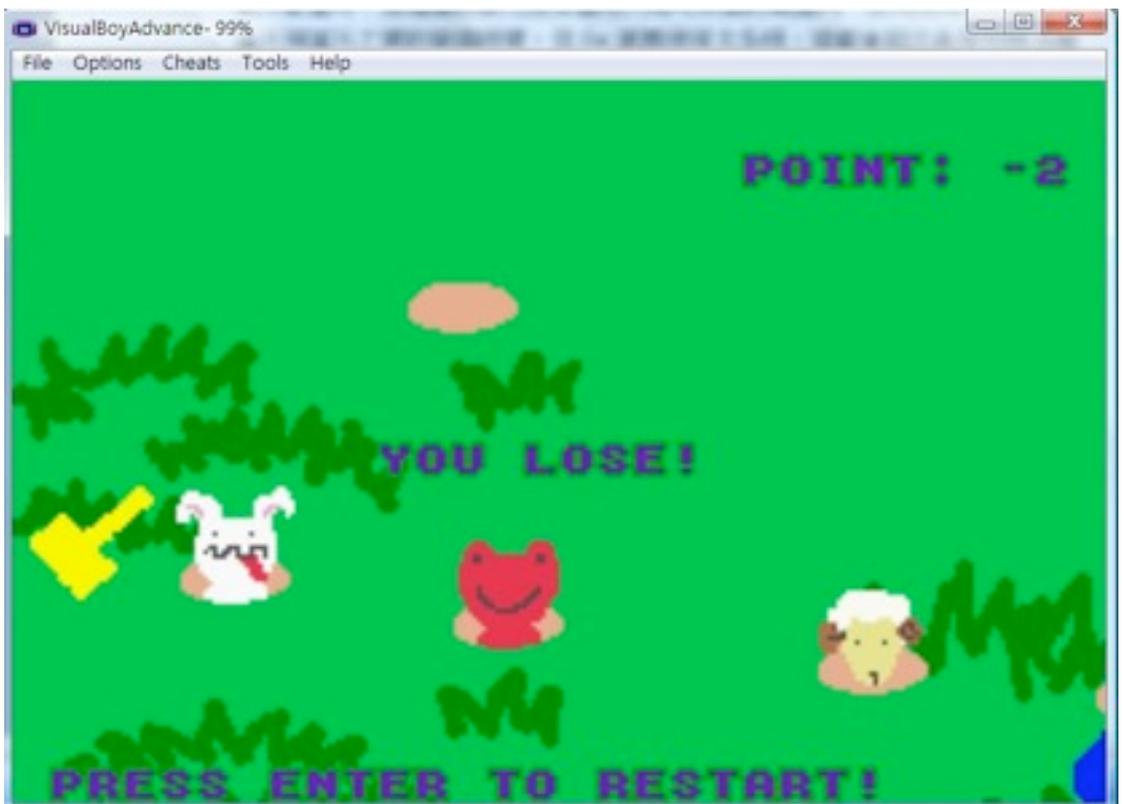


畫面擷取

你贏了！



你輸了！



ASM FINAL PROJECT

