

Computer Organization and Assembly Languages, Fall 2007

Final Project-黃金傳說

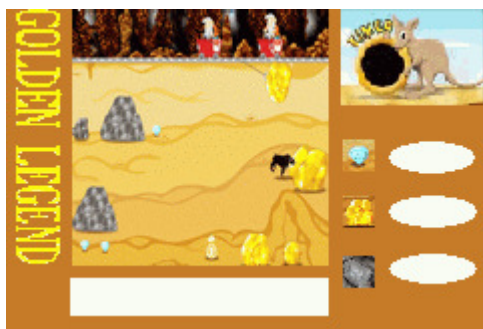


資工二 B96902013 邱柏睿
資工二 B96902023 游舜翔
資工二 B96902041 吳承恩

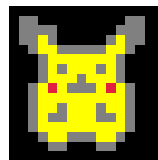
前言-遊戲介紹

從一開始對組合語言完全沒有有概念，經過一個學期對組合語言的訓練，漸漸對組合語言有了一些瞭解，甚至可以在G B A模擬器上作一些小程式，這次我們決定在G B A的平台上實作一個小遊戲「黃金傳說」，這個小遊戲並沒有複雜的A I，而是經由簡單的規則來產生樂趣，這遊戲是利用挖寶的方式，在一個8 x 8的地圖中每一格都有黃金、鑽石、鐵礦等等，每項的金錢都不同，一開始地圖是覆蓋住的，G a m e r要去挖掘未知的地區，並限制可以挖的次數，以挖到高分為目標。為了增加遊戲的刺激性，除了金礦之外，甚至還會有敵人，G a m e r此時必須要回答問題，回答了之後可能會發生好的事件也可能會發生壞的事件。

而在 code 方面，我們在整個螢幕上先放上一個background<如圖一>，每個地圖上覆蓋住的未知區域都是一個s p r i t e，主角人物也是一個s p r i t e<圖二>，經由Q u e r y _ s e l e c t來決定上下左右的移動，並利用r a n d o m的方式來決定挖到的區域產生甚麼，如果碰到敵人時，也是亂數決定要問甚麼問題！（較為細部的部分會在稍後提到）



(圖一)



(圖三)

操作介面

A	對前方區域開墾
B	選擇選單用
L	無
R	無
上	往上走 o r 面向上方
下	往下走 o r 面向下方
左	往左走 o r 面向左方
右	往右走 o r 面向右方
START	選擇問題的 A
SELECT	選擇問題的 B

起始介面我們分成兩段，第一個介面(圖三)在按下 start 後，會進入 select 畫面(圖四)，此時可以有三個選擇：分別是 start the game；review the high scores；quit the game (如圖四)，選擇 start the game 會進入遊戲畫面，選擇 review the high scores 可以觀看目前為止的最高分記錄，選擇 quit 則會進入 end 畫面。



(圖三)



(圖四)

進入遊戲畫面時，首先我們的主角皮卡丘出現在左上角(0,0)的位置，地圖上每一格都是16x16的pixel，我們利用上下左右來選擇皮卡丘前進的方向，右上角標示剩餘可以開墾的次數(用LIFE表示)當皮卡丘面對未知區域要開墾時，Gamer可以按下A鈕，這時前方的未知區域就會出現鐵礦、金礦、鑽石，出現的東西會在右方顯示。地圖中的驚嘆號是特殊的格子，player挖掘到那格時會有特別的事件發生！

如果遇到enemy時，player就必須回答問題時，如果選擇對的選項，則可以得到獎賞，如果選擇錯的選項，則會倒扣life(挖掘次數)。

地圖下方的對話框會顯示出對話跟player互動，回答問題時也是由對話框來顯示。當life剩餘數為0時，遊戲就會結束。進入結束畫面(圖五)





(圖五)

實作內容

地圖部分：

整個地圖由 8 x 8 個格子構成，每個格子都是 16 x 16 pixels 構成的，並且我們用開了二維陣列 `stone_x[][]` 和 `stone_y[][]` 來記錄格子的位置，左上角第一格的位置為 `[33][0]`，從左上角往右數第一格是 `[33+16][0]`，左上角往下數第一格是 `[33][0+16]` 以此類推。

我們設置一個 array：`stone_exist[][]` 則用來記錄區域是否挖掘過，一開始所有的 `stone_exist[][]` 值都為 1（未挖掘過），我們用 `calcute()` 這個自定涵式來判斷可否行走，當 `stone_exist[][] = 1` 時，人物的 `position` 就不能移動到那個座標，當挖掘過後，`stone_exist[][]` 會變成 0，

程式部分：

程式主要由一個 **main function** 和幾個涵式所構成，其中包括 **VBL function**、**query_buttons()**、**update_player_pos()**、**calcute()**、**query_select**、**random_happen()**、**random_questions()**、**s_random_happen()**、**setstartbg()**、**high_score()**、**setendbg()** 等

在 main 中我們將基本的變數設置好，顯示模式我們是用 `model 0` 也就是 `T i l e` 來做，我們固定將文字放在圖層

0，而BG則是放在圖層1，如此一來文字就可以印在背景圖片上，整個流程如下

初始化 H A M l i b 和 s p i r t e 的調色盤 → s t a r t :
初始化文字格式並且 s e t 開始介面 → 初始化所有的變數並進入 s e l e c t 畫面 → s e l e c t 中有三個選項：第一個選項會開始遊戲，第二個選項會跳到最高記錄的畫面，第三個選項會跳到 q u i t 畫面 → (I F 選擇第一個選項) 設定遊戲畫面和所有 s p r i t e 的調色盤 → 執行 V B L f u n c t i o n → 當 L I F E = 0 時 Reset 所有變數並跳進 e n d 畫面 → 在 e n d 畫面有兩個選項，選擇 r e s t a r t 會跳到一開始 s t a r t 的標籤，選擇 e x i t 則會進入結束化面並重新跑一次程式。

VBL_FUNCTION

中我們主要用來控制銀幕刷新的速度和顯示物件，在每一次刷新時間到時，會跑一個程序裡面包含了 query_buttons()、update_player_gfx()、update_player_pos()、query_select()、四個 function，並在跑完這些程序後，印出更新後的數值，這些數值包括 s c o r e、l i f e、n__d (挖到的鑽石數目)、n__i (鐵礦數目)、n__g (金礦數目)。

SETSTARTBG()

這個 function 是設定 start 畫面的調色盤並顯示在銀幕上。

HIGH_SCORE ()

這個函式是在 select 畫面時選擇 High scores 時，讀入這個函式，在銀幕上顯示出最高記錄的畫面，並且在判斷玩者輸入 start 鍵時，跳回到 select 畫面。

QUERY_BUTTONS()

主要是判斷移動的 function，在這裡我們用到 caculate function 來判斷是否有障礙物，這裡我們判斷使用者按下的上下左右按鍵，如果經過 caculate 判斷使用者案的方向上沒有障礙物，則更新 player 的位置和方向。

QUERY_SELECT()

則是判斷挖礦的 function，如果面對的方向有未開發區域並且接收到 A 按鈕的指令，則會開啟 random_happen 的函式，並且將 stone_exist[][] 設為 0，Life（剩餘挖掘次數），並且更新障礙物的圖像

RANDOM_HAPPEN()

總共有 8 種情況會 random 出現，有挖到金礦、鐵礦、鑽石、甚麼都沒有等等，其中最後一種狀況會碰到 enemy，此時會呼叫 random_questions 的 function。

S_RANDOM_HAPPEN()

這是遇到特殊事件時的 function，有兩種事件，第一種會得到大量的黃金，第二種則會遇到炸彈，此時會創造一個炸彈的物件，並利用一個 bomb_time 來記錄時間，當 bomb_time 到達 1 時炸彈變會爆炸。

RANDOM_QUESTIONS()

會隨機產生問題，使用者可以按下 start 和 select 來選擇答案，選擇完之後會出現加分或扣 life 等不同的結果，並在選擇完後顯示新的 score 並跳出問問題的模式

SETENDBG()

載入這個 function 時，會進入 end 畫面。

遇到的困難和問題

筆電數不足：我們這組共有 3 個人但是只有兩台筆電，因此當我們想聚在一起寫 code 或繪圖時，總會少一台電腦，也因此常常有閒置人力，造成時間的浪費，因此後來我們利用 219 電腦教室的電腦來補足電腦數的不足。

記憶體結構不了解：一開始的時候不太了解 GBA 的存取記憶體方式，留著以前寫 c 語言的習慣，使用了一堆變數而沒去清空，導致記憶體不足，在換背景時出現錯誤，之後就要用時才設定，用完立刻清空才解決這個問題。

錯誤使用調色盤：原先以為調色盤就是一般常見的 256 色，gfx2gba 指令只是去判斷圖片中哪裡要用甚麼顏色，導致色彩出現錯誤，圖形整個變得很奇怪，後來把每個背景用一個調色盤，所有物件編成一個調色盤後，圖形顯示就變的正常，順利解決問題。

亂數的使用：本來以為可以跟 C 一樣用 `srand(time(NULL))` 去擷取時間作為亂數種子，但後來發現沒辦法，可能是 GBA 沒有儲存時間，後來利用 `vbl` 這個函式再跑的時候順便加一個數字，把此數字當成亂數種子就可順利解決。

學習心得

b96902023 資工二 游舜翔

在整個 coding 中，大家從無到有慢慢得從學習 Ham 語法開始，繪出屬於自己遊戲的插圖，看著黑黑的銀幕開始變得有顏色，銀幕上的物件開始移動，到最後甚至有了開始畫面和選單，成為一個具有完整性的小遊戲，這其中不但花費了大量心力，也得到許多滿足感，當解決困擾已久的問題時，那種迸發出來的喜悅實在難以言喻，除此之外，這次接觸 Ham 的經驗非常特別，有別於以往編寫高階程式語言，這次我們要直接跟硬體作連結，並且對於圖片和編碼也有進一步的了解，可以說是實際了解了程式語言的實際操作，是一次非常好的學習經驗。

B96902013 資工二 邱柏睿

沒想到寫個小遊戲就要這麼麻煩，所有東西都要從頭弄起，當 code 越來越長時，要找出 bug 就越困難，過程中各式各樣的 bug 要去解決，才是最困難的問題。這次的報告，學習到許多程式實際應用，以及跟其他人合作完成報告。

B96902041 資工二 吳承恩

寫了好久的程式，和別人一起合作的感覺真得很不一樣，彼此間不同的寫作習慣真的有點麻煩，所幸經過溝通討論後都能順利執行。對於 gba 的結構有了更多的了解，也重新復習了許久未用的 c 語言，畫圖也滿好玩的，看到圖片順利出現就很開心，HAM 中很多指令都會牽扯到硬體，以前都沒有使用倒，這次報告真的收穫很豐富，最重要的就是將有限的資源盡可能最佳的去利用。

感謝名單：

Gfx2gba 轉圖檔軟體

Ham 教學網址 **L**<http://www.aaronrogers.com/ham/>

Ham 教程翻義：<http://angeljicu.googlepages.com/ham>

特別感謝：資工系館