

計算機組織及組合語言期末專題

The Project of Computer Organization and Assembly Language

The Vent F22

◎Crew and Worklist :

CSIE SUFFERmore(sophomore)

B95902008	蔡博倫	Visual Art, Programming, BUG-detecting
B95902032	魏志強	Programming, BUG-detecting, Designing,Leader
B95902066	黃雨喬	Designing, Idea providing
B95902081	王裕博	Programming, BUG-detecting, Designing

Thanks to	翁健庭	for his help in Digital Sound Processing
	曾冠諭	for trouble-shooting

◎How to :

Brief Introduction:

We design and make the game which is entirely using the method and definition inside TONC. The game is built in DevkitPro under the environment of Windows XP. All function and all data, included graphics, are hand-made. Because the function inside the TONC is not so familiar as other kind of the references, we have rewritten the core section twice while coding. Its function needs a lot of addresses of registers, and then we use the concept and technique learned from assembly. Due to the conceivable features, we implement the general structure by C language and some register defined in the TONC tutorial.



Ideas beneath the code :

這次的原本所要設計的，是仿 **Front Line**，但後來我們決定改作“超時空要塞”。至於如何從零到現在所看到的這份成果，這是經過了以下充滿心酸血淚的過程。

Coding : F-22 戰鬥機是我們邁向成品的第一步，一切的一切，都是從這開始。我們仿造 **TONC reference** 裡的 **code**，印出了我們的主角 **F-22 戰鬥機**。因為 **TONC** 是屬於比較基本較貼近 **GBA** 底層的語言，所以爲了要印出一個 **Sprite**，就要設定一些 **memory** 以及 **register**，才能順利的印出來，而且因爲是使用 **tile**，所以在處理圖片方面，花了很多心思，常常是圖片出來了顏色卻不小心錯了，或是顯示出來很奇怪，都可能是 **memory** 的問題，所以花了很多時間在 **de** 這方面的 **Bug**。不管是圖片的出現或是消失，都要設定 **memory** 中 **tile** 的 **attribute**。

另外因爲出現在螢幕上的東西，都是一個 **Sprite**，而總共的 **Sprite** 個數有上限，所以我們用了 **Array** 來儲存子彈以及各個人物，再想辦法用“回收”的方式，達到在有限的個數裡，可以在螢幕上表現很多的東西。

一開始原本我們只有一個敵人(**Hil**)，不過花了很多時間將函式都 **General** 之後，就可以產生很多敵人(**Pangfeng**、**CYY**)，所以在這方面也下了苦功。而子彈和敵人的移動，一開始也是只會往 **X** 軸方向移動，後來設計成可以追蹤，增加遊戲性。

另外敵人的出現和開火，都是用隨機的方式來處理的(**TONC** 中的 **pseudo random**) 音樂和文字的處理又是另外的問題，將 **TONC** 的 **reference** 研究了一下，並且經過很多的 **try and error**，才成功的印出文字及放出音效。

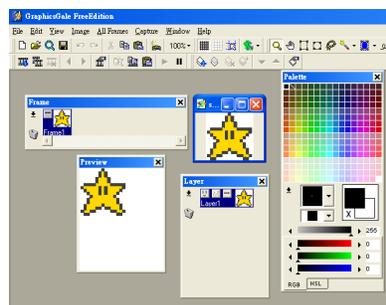
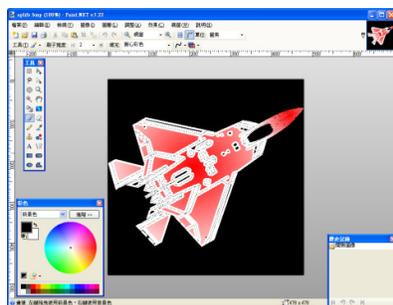
Under the screen :

關於視覺方面，由於 **Sprite** 的個數最多 **128** 個，而只有 **4bpp** 及 **8bpp** 兩種選擇，我們這組採用 **4bpp** 的規格去設計以節省調色盤空間。圖片大小有從子彈物件(其實是 **struct** 模擬)的 **8x8pixels** 到 **F22** 及教授們的 **16x16p** 使用 **16bit** 的調色盤(**Palette**)去實作。

背景部分，我們用 **tilemap** 的形式選擇 **8bpp-256** 色去製作。

所有的圖片皆非原本樣貌，處理如下：

1. 將畫面中想要的某些物件透過 **Paint.NET 3.22** 去剪裁、縮放、過濾並調整色彩和立體旋轉以符合視覺感受。



- 2.初步處理的圖片再經 Graphics Gale 去一格一格校正調色盤色彩，接著減少 color depth 以符合 16 色限制，然後精緻壓縮成 16x16 的大小。
- 3.最後一步是使用 TONC 作者的另一個成品 Usenti，利用其軟體可將圖片轉換成 C 的 HEADER 及 SOURCE CODE，並可在調色盤上做交換色彩以透明化想要的部分(類似 HAM 的 gfx2gba)。
- 4.而開頭動畫的部分使用 tilemap 的 background 中的兩層切換隨後緩衝去 copy tile sets and screen entry 進入記憶體中以營造效果。

How to play :

Controll :

↑ ↓ ← →	-----	方向鍵
A	-----	無
B	-----	發射子彈
L	-----	回復原本狀態(非金剛不壞之身)
R	-----	無
START	-----	加一條命
SELECT	-----	變身成浴血鳳凰(金剛不壞之身)

Rules :

一開始有六條命，人被子彈或是敵人擊中，都會少一條命，直到六條命用光為止。遊戲的目標是，殺暴敵人，往更高的里程數邁進！

Content :

本遊戲設計，是由三位教授扮演魔王的角色，鎮守第一關的魔王，就是令人聞之色變的呂學一教授，在零到一百英哩鎮守著，而他的武器也實他的拿手專門強項----- Θ (one of the Asymptotic Notations)。而當你好運過了第一關後，第二關是由使徒的頭頭，也就是劉邦鋒教授把關，在此關（一百至兩百英哩），魔王的速度與第一關有著不小的差異，而其所發射的子彈(使徒)也非第一關所能比擬，只能說一關比一關還難過。

而駐守在第三關的是，令人又愛又怕的 CYY 教授了，他使用了“OR Gate”當成他的武器。前面兩關的速度，跟這關比起來，簡直就是小巫見大巫！莊永裕教授移動的速度，讓你“目不暇給”，而其子彈的速度直比光速呀！不偷偷的使用密技，就連骨灰級的玩家，都要一敗塗地了。除了我們之外，要過這關的或然率-----零。

隨後的四、五、六關，魔王分別和之前第一、二、三關的魔王是一樣的，但截然不同的，這次，他們挾著第一次被你運氣好做掉的羞辱，將其轉換成子彈，源源不絕的奔來，非要置你於死地。因為我們堅信，沒有人可以過到最後，因此，

沒有設計所謂的最後大魔王，如果你沒靠密技而玩得到最後，請聯絡小強（02-23345678），將會寫一個究極的版本給你！

◎Conclusion :

Team work 是個很好的合作經驗，也因為這次的 Project 更了解一些硬體以及軟體內部的設計，獲益良多。

◎You can download the game here :

<http://w.csie.org/~b95032/vent.gba>

If you have any problem, please contact b95032@csie.ntu.edu.tw