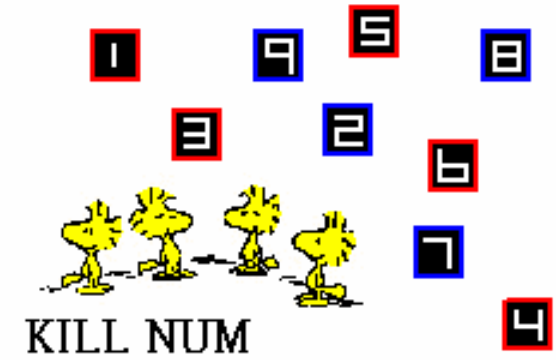
Computer Organization and Assembly Language

Final Project

益智遊戲—消數字



B95902057 薛琇文
B95902059 林琬瑜
B95902093 蔡長蓉

目錄

壹、動機

從小或多或少玩過一些掌上型 GBA 遊戲，對卡匣遊戲的開發，以及遊戲與 GBA 間的運作機制一直很感興趣，這學期修習組合語言，教授於課程中對 GBA 作了很完整的介紹，投影片上秀了許多歷代的 GBA 機型，讓人倍感懷念，期末 project 便因此決定，要寫一個可以在 GBA 上跑的小遊戲。

到了學期後半，開始逐步構思，自生活中尋覓、蒐羅靈感。一來，適前陣子到微風影城看了電影黃金羅盤，想像著...遊戲可以有個能選擇的盤面；二來，常在 217 實驗室，和同學們一起唸書、寫作業，作課業上的討論，平時部分同學的休閒踩地雷——許多顯示附近地雷數的數字...於是，數字盤面入袋！還有另一個也相當風行的魔法氣泡遊戲，常聽同學們喊著：「 X 消！... Y 消！」靈感忽地萌現——消數字！又，加法是最直覺、直觀的想法，便以在盤面上搜尋任意數目的數字，加總後為要求之總和，則能將已選數字全數消去的構想，來實作這次的期末 project。

由於和 gba 相關的作業三用的是 ARM，gba simulator — HAM 又提供許多實用而可供觀摩的 examples，仔細考量後便決定採用此兩者。

貳、遊戲介紹與操作介面

一、遊戲規則

消數字這個遊戲是以前曾在網站上玩過的一個小遊戲，因為不知道它正確的名稱，我們決定依照這個遊戲的目標，取名為「消

數字」(KILL NUM)遊戲。

消數字遊戲的規則如下，給定一個數字棋盤和 sum 值，另外再給一個時間當破關條件。玩家要做的事，就是每次選擇棋盤上的幾個數字，使他們相加等於 sum 值，而這幾個數字就會被消掉，當棋盤上已經沒有數字可以相加得到 sum 值，這個關卡結束。最後經過一個計分公式算出玩家分數(詳見三、記分方式)，若達過關條件則順利過關。

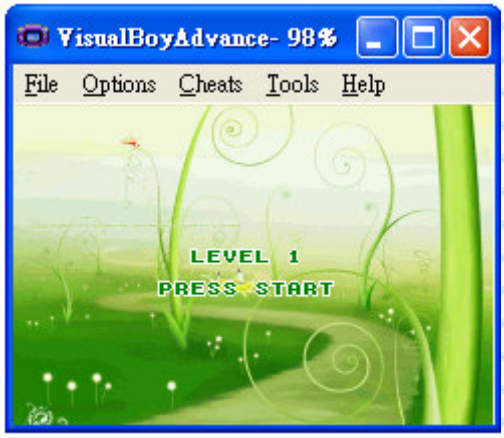
二、操作介面

遊戲一開始，畫面顯示現在是第幾關，按下 start 鍵進入這個關卡。(如圖一)

關卡進行時，畫面最上方會顯示本關的 sum 值和破關條件，底下是一個計時器顯示本關到目前為止所花的時間。畫面中央是這關的數字棋盤，紅色方框表示目前可圈選的數字，用 up、down、left、right 四個鍵移動位置，按下 start 鍵選擇紅色方框所在的數字，該數字會以藍色方框標記。(如圖二)

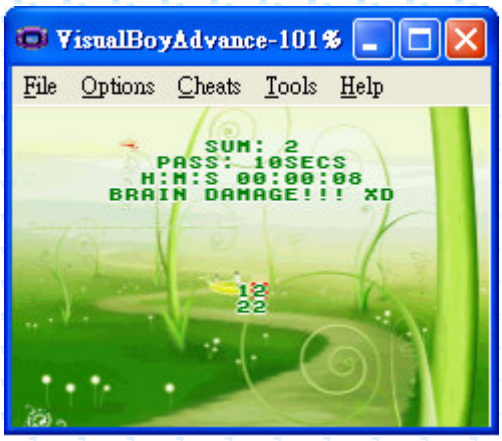
為了增加遊戲的趣味性，我們有設計處罰機制。如果玩家選擇的數字合大於 sum 值，棋盤上方會印出”BRAIN DAMAGED!!! XD”的字樣(如圖三)。又為了增加遊戲的神祕性，我們設計了密技，按下 select 鍵即可跳關。

每關結束後畫面都會顯示”GAMEOVER!!!”的字樣，及”COST TIME”(這一關所花的時間)、”PENALTY”(處罰機制，詳見三、記分方式)、”TOTAL”(經過計分公式算出的結果，詳見三、記分方式)。若順利過關，畫面顯示"press start to next level"，按下 start 鍵進入下一個關卡；否則，畫面顯示"You lose! Please start again."，按下 start 鍵由這一關重新開始。(如圖四和圖五)



(圖一)

(圖二)



(圖三)

(圖四)(成功)

(圖五)(失敗)

三、記分方式

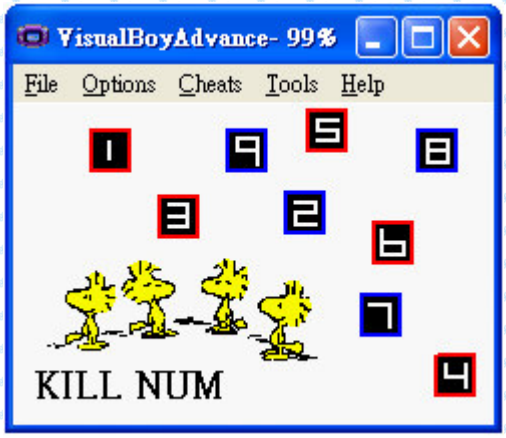
Penalty：每當玩家選擇的數字合大於 sum 值，啟動處罰機制，penalty 加 10 sec。如果最後盤面上的數字全部消完，penalty 為 max(0, penalty-10)。

Total： 本關所花時間(cost time)加上(剩下數字合)*(剩下數字個數)/10 再加上 penalty。

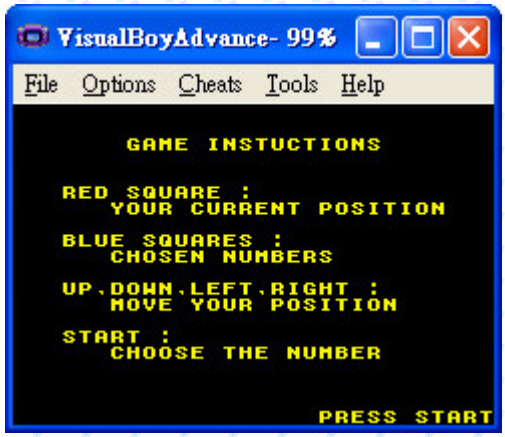
四、關卡流程

一開始顯示遊戲開始畫面(如圖六)，當玩家按下 start 鍵進入遊戲說明畫面(如圖七)，按下 start 鍵即進入第一關的準備畫面，再按下 start 鍵開始第一關。

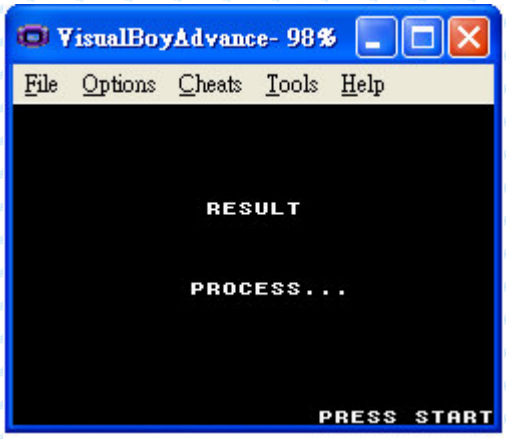
遊戲總共有五個關卡，棋盤大小會隨著關卡越來越大，每一關邊長增加兩個數字。每個關卡有不同的破關條件，若五關都通過了，則進入 result process 畫面(如圖八)，按下 start 鍵進入成績畫面(如圖九)，按下 start 鍵進入感謝名單(如圖十)，再按 start 鍵進入破關畫面(如圖十一)。



(圖六)



(圖七)



(圖八)

(圖九)

(圖十)

(圖十一)

參、實作內容

一、重要函式

1. void input(void)

跟據玩家按的不同按鍵，做出該有的反應。up、down、left、right 分別是紅色方框上、下、左、右移動，當超出邊界時會從另一邊的邊界回來，例如在最左邊按 left，會跑到棋盤

同列最右邊那行去。按下 start 鍵(表示選取數字)則呼叫函式 check() (請參照函式 3.)。

2. void OutputTime(void)

用來顯示玩家每一關到目前為止所花費的時間。

3. void check()

在 input() (請參照函式 1.)中，當玩家按了 start 鍵(選擇數字)，就執行 check()這個函式。用一個變數 total 記錄目前所選擇的數字總和，將新選擇的這個數字加進 total。接著檢查 total 值，若 total 值小於 sum 值，繼續遊戲；若 total 值與 sum 值相同，消掉被選擇的那些數字，並將 total 歸零；若 total 值比 sum 值大，啟動處罰機制，將所有原本選擇的數字取消標記，把 total 歸零，penalty 加 10 並印出"BRAIN DAMAGED!!!　XD"的字樣。

4. void checkgameover()

呼叫 recursion() (請參照函式 5.)，檢查棋盤上是否還有可以相加得到 sum 值的數字。若沒有，結束這個關卡。並統計棋盤上剩下的數字個數以及剩下的數字合，以便計分。

5. void recursion()

用遞迴的方式檢查棋盤上是否還有可以相加得到 sum 值的數字。

二、特殊處理

為了防止出現不可能消掉的數字，例如 sum 為 3 可是盤面上有 9，每一關都是先選出 sum 然後把 max(sum, 9)當作盤面數字的上限。

三、特效

遊戲的目的是娛樂，而聲光效果對遊戲的趣味性有相當程度的影響，所以在遊戲主程式寫完後，我們就開始研究如何製作絢麗的聲光效果。其中又可分為圖片、聲音和動畫三大部分。

圖片部分可分為背景和插圖兩種。插圖是用 sprite 顯示，先找到合適的插圖圖案，用小畫家調成標準的 sprite 大小，如 64*64，然後存成 bmp 檔。背景也是先找到合適的圖片，然後視情況如果有需要的話用小畫家加工，再用小畫家或 PhotoImpact 調成標準的 background 大小，即圖片部分為 240*160，補白邊補成 256*256。最後再用 gfx2gba 轉成.c 檔，sprite 會有一個 palette 檔和一個 tile 檔，background 則是再加一個 map 檔，然後用 function 填到合適的記憶體位置和設好記憶體中對應位置的值，就可以顯示出來了。

聲音是用 KRAWALL 系列的 function 處理，因為 KRAWALL 系列只接受 s3m 檔，所以先上網找 s3m 檔的音樂，選出合適的加入遊戲中。

背景和聲音都是一關一個，其中背景還有進遊戲畫面、感謝名單和結束畫面，聲音是還有結束音樂。

動畫主要原理就是用 timer 控制顯示與否，例如進遊戲畫面中右下角會閃動的”press enter”。

四、組語實做

主要目標是把適合組語做的部分用組語實作，所以我們這組決定把方向定為和硬體溝通的部分。

第一個實作的 function 為 input()，也就是控制紅色方框移動的 function。先找到 Pad 對應的記憶體，用 register 把值 load 出來，看是哪一個 bit 被設為 1 就知道現在是哪一個鍵被按下去，然後

再做出適當回應。

第二個實作的 function 為 OutputTime()，也就是把時間顯示出來的 function。先 load 出秒、分、小時現在的值，把秒數加 1 後判斷是否大於 59，如有進位再判斷分的部分，又有進位再判斷小時的部分，然後在畫面上印出最後處理好的值。

第三個實作的 function 為 myLoadObjPal()，也就是把 sprite 的 palette 檔存到合適的記憶體位置。這部分原本是用 HAM 的 ham_LoadObjPal()這個 function。

第四個實作的 function 為 myLoadBg()，也就是把 sprite 的 palette 檔、tile 檔、map 檔存到合適的記憶體位置，再把記憶體中相關的設定設好。這部分原本是用 HAM 的 ham_LoadBGPal()、ham_InitTileSet()、ham_InitMapSet()、ham_InitBg()這四個 function。

第五個實作的 function 為 freqFunc()。這個是用來在動畫中配合 timer 決定顯示與否的 function。

第六個實作的 function 為 vblFunc()。這個是用來在控制聲音時配合 KRAWALL 系列的 function。

五、困難和解決方法

第一個遇到的困難是 random，因為每一關的 sum 和數字盤面都是亂數產生的，主要是用 c 的 rand 和 srand 實作。其中 srand 的參數我們原本不知道要給什麼，後來想到有進遊戲畫面，所以就用一個變數在遊戲一開始就從 0 開始加，直到玩家按下進遊戲畫面的第一個 start，把這個變數當作 srand 的參數。

第二個遇到的困難是背景圖，其中又分為圖片解析度問題和記憶體問題。記憶體問題就是記憶體太小，所以 gba 其實只能容納一張 background 圖，解決方法為每次要 load 新圖前先 reset 背景部份的記憶體，可是這麼做會造成字型也遺失，導致無法在背

景上印出文字，所以 text system 也要重新 initial。圖片解析度問題是因為我們一開始是用小畫家轉成 256 色的 bmp 檔，這會造成色彩有誤差，後來用 PhotoImpact 色彩誤差就小很多，顯示出來的圖也變的非常漂亮。

第三個遇到的問題為聲音，HAM 處理聲音有兩種方式，Direct Sound Control functions 和 Krawall functions。我們一開始是用前者，要先把 wav 檔轉成.c 檔，可是轉出來的檔案太大，不只聲音放不出來，連在聲音檔後面的圖片檔也全部壞掉，只好放棄。後來用 Krawall functions 一開始也是無法用，後來發現 makefile 也要改才成功。

肆、參考資料、軟體資源與感謝名單

一、參考資料

1. HAM online documentation
2. 音效
   http://mccormick.cx/viewcvs/looper/Attic/krawall.h?rev=1.4
3. mygba.h
   http://emulinks.de/svn/enjin/trunk/toham/mygba.h
4. tonc
   http://www.coranac.com/tonc/text/

二、軟體資源

1. HAM
2. gfx2gba
3. 小畫家
4. PhotoImpact

三、感謝名單

B93902102 侯昆邦 (Favonia)

組語助教　李根逸 (Feis)

B95902108 馮俊菘 (ferng)

B95902106 溫在宇 (Joseph)

B95901207 陳柏龍 (pP)

B95902049 陳耀男 (sas)

B95902028 蔡明亨 (scan)

B95902064 蔡青樺 (zhichin)