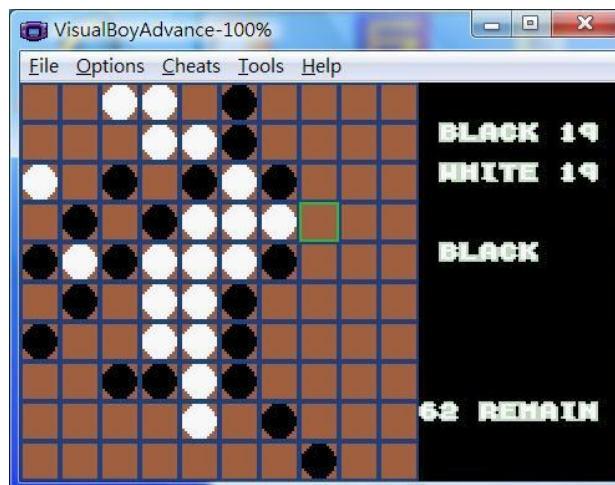


Computer Organization and Assembly Language

Final Project

★Chessman X9★



B96902011 林博智

B96902012 林士恩

B96902016 白哲璋

B96902086 謝宗潛

Index !

壹、遊戲製作動機	P2
貳、遊戲操作與介面	P3
參、實作內容	P4
肆、參考資料、軟體資源	P7

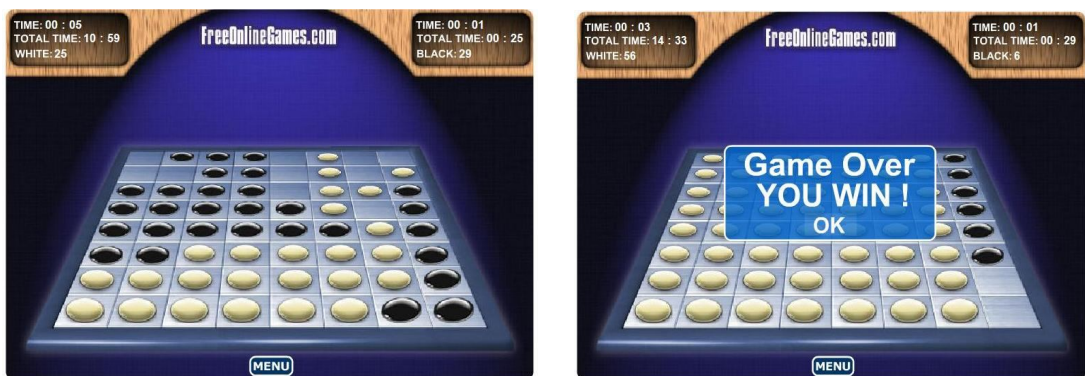
壹、遊戲製作動機

我們製作這個遊戲的靈感，是小時候相當流行的小遊戲黑白棋。黑白棋在各個電子辭典中常常是不可或缺的遊戲。雖然看似簡單容易上手的遊戲，但是往往讓人一玩及愛不釋手。一場十分鐘左右的遊戲，可是一玩可能就玩了一整個下午而渾然不知。實在是老少咸宜的小遊戲。



【3D 黑白棋遊戲畫面】

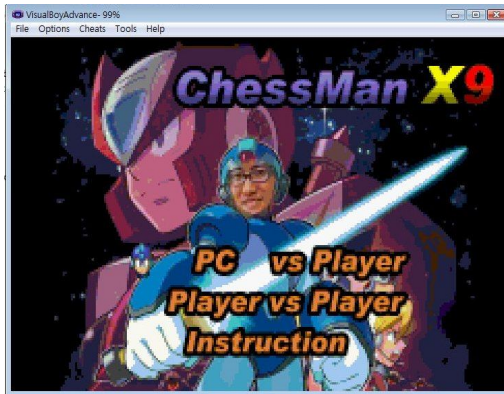
在遊戲中，玩家可以控制黑棋或是白棋。遊戲剛開始時，黑棋白棋各兩顆排列如同上圖所示。玩家可選擇控制白棋或是黑棋。當玩家下棋使得對方棋子被自己的棋子兩邊夾住時，即可將對方棋子的顏色變成自己的。當兩方都不能下的時候還有棋盤下滿，或者是有一方全被滅的時候遊戲就結束。



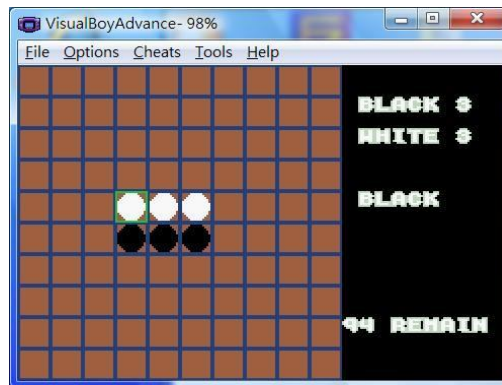
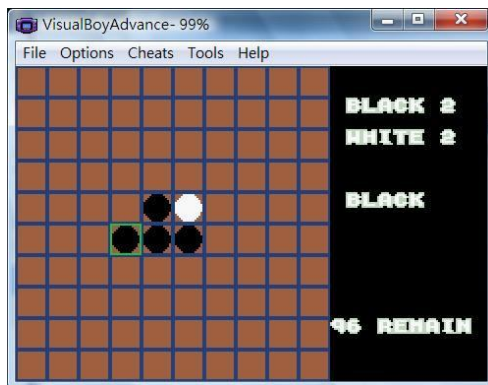
【遊戲畫面】

貳、遊戲介紹與及操作介面

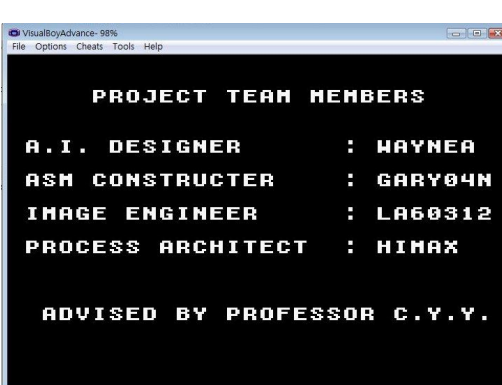
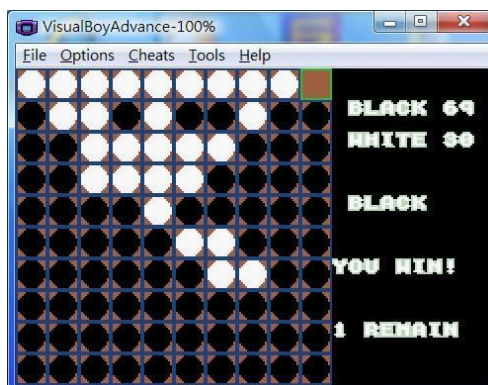
遊戲主選單中有三個選項，第一個是玩家對電腦，第二個是玩家對玩家，第三個是 instruction，遊戲操作介紹。



如果選擇跟電腦對戰，則玩家控制黑棋，在螢幕右方有黑棋白棋分別的統計數目，跟場上還剩下幾個空格可下。當有一方不能下時，另一方即可直接下棋。若找不到地方下時，按下 B 鍵可以獲得提示，螢幕會顯示提示要下的地方。當有一方被全部吃掉，或是兩方都沒地方下時遊戲即結束。遊戲中除了可以選擇跟電腦對戰外，也可選擇兩個玩家互相對戰。



【遊戲畫面】



【遊戲結束畫面】

參、遊戲實作

我們需要一開始的遊戲選單，只要做出合適的圖存成 **bmp** 檔，再把圖讀入就可以做出一個好的選單了。

接下來就是這個實作的核心，大致上可把它分成三部份：

一：操作

如何讓我們的棋子移動，其實很簡單，只要有了下面的對應表再利用一個 `Vbl_func()` 去控制移動的速度就可以了

	HAM 中對應值
GBA 按鈕	
UP	F_CTRLINPUT_UP_PRESSED
DOWN	F_CTRLINPUT_DOWN_PRESSED
LEFT	F_CTRLINPUT_LEFT_PRESSED
RIGHT	F_CTRLINPUT_RIGHT_PRESSED
START	F_CTRLINPUT_START_PRESSED
SELECT	F_CTRLINPUT_SELECT_PRESSED
B	F_CTRLINPUT_B_PRESSED
A	F_CTRLINPUT_A_PRESSED
L	F_CTRLINPUT_L_PRESSED
R	F_CTRLINPUT_R_PRESSED

二：資料結構

要如何將 GBA 上的圖或點去作操作或變換，這裡我們用了一個二維的陣列去對應 GBA 上的圖，初始值為 0，黑棋為 1，白棋為-1 這樣我們就可以簡單去對圖作操作了，也很簡單的把我們的陣列映在 GBA 的螢幕上了

三：如何吃棋和 PC 下棋

Q1:如何判斷這個位置可不可下勒?

A:使用 check()這個函式去掃描這個位置，並且去判斷這個位置可以吃到多少顆子。假如回傳值是 0 就代表，這個位置是不可以下的!!!

Q2:要怎麼把子吃掉呢?

A:使用 change()這個函式，利用 recursion 方式。方式如下->

這個函式有四個參數，分別為棋子的 X 坐標 Y 座標 下這顆棋子的顏色的負數還有，一個方向(call 這函式這個值都是 0)，其後就往這個座標的 8 個方向去 recursion，遇到空白或是跟自己同顏色或是邊界就 return，並且在 return 的時候把子的顏色換掉。就可以達成吃子的目的囉

Q3: PC 是怎麼去下棋的阿?:

A: PC 在每次玩家下玩棋後，掃描一次陣列，基本上會先下目前可以吃最多的地方，但是我們加了許多判別式來增加他的能力，他會先從四個角落掃描，判斷是否可以下，如果可以下則優先下在角落。其次掃描四個邊，同樣去掃描不可下，並且去判斷下在這個位置會不會被對方吃掉角落，如果會則不下在這個位置，在去尋找其他合法的位置，如果最後只有上述的位置合法，就還是下在那個位置，最後在去掃描中間的部份，並且找到可以吃到最多子的位置，然後就去 call change()去把子吃掉，如果整個掃描下來沒有地方吃，就換玩家下。

有了上面的方法，我們就可以實作出酷炫 GBA Chesse man x9 的遊戲了

四、重要函式

`void change_index();`

藉由全域變數 `x,y` 對應回 `database` 上，來確認目前游標的位置上的應該顯是哪種圖片，並解顯示應該顯示的圖片。

`void counteat(int eatnumber,int color);`

傳入顏色(`color`)和吃了幾個棋子(`eatnumber`)，即改變全域變數中的黑白棋總數(`blacksum,whitesum`)。

`void vbl_func_menu();`

呼叫遊戲選單期間的平行動作

`void vbl_func_game();`

呼叫遊戲執行期間的平行動作

`void vbl_func_end();`

呼叫遊戲結束期間的平行動作

`void query_buttons();`

遊戲執行期間按鍵接收

`void change(int i,int j,int color,int dir);`

傳入新下的棋座標(`i,j`)、顏色(`color`)，即會改變矩陣中該其該吃掉的棋子

`int pc(int color);`

電腦運用 AI 幫(`color`)下棋

`int check(int x,int y,int color);`

查看此座標(`x,y`)對(`color`)可以吃幾個棋子

`int bestsol(int color);`

利用 AI 選出目前對(`color`)的最佳解

`int player(int color);`

玩家下棋

五、遇到的困難

調色盤的問題:

一開始還搞不懂調色盤的功用，常常調到後來的圖片變得很奇怪

背景和物件的轉換:

一開始想要換背景或換物件，但常常發生錯亂，或是沒換到或被覆蓋等等的問題

黑白棋的判斷問題:

有一些邊界的處理會有錯誤產生，還有一些棋子計算會有錯誤

物件的覆蓋問題

物件的優先順序，背景和物件的優先顯示問題，還有換背景的初始問題

組語的實行問題:

一開始輸入過多參數，後來才發現一般只能輸入 4 個參數，重要運算執行組語實行時發生了未知的錯誤

電腦 AI 的運算問題:

下棋的優先順序固定，還有很多要判斷的情形，一開始的 AI 易被玩家找到下棋模式，經過修改有比較好

肆、參考資料、軟體資源

一、參考資料

1. HAM Tutorial introduction

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

2. 3D 黑白棋

<http://www.gagameme.com/tw/play/3d-reversi.html>

二、軟體資源

1. HAM

2. gfx2gba

3 小畫家

4. PhotoImpact.