

Computer Organization and Assembly Languages Final Project

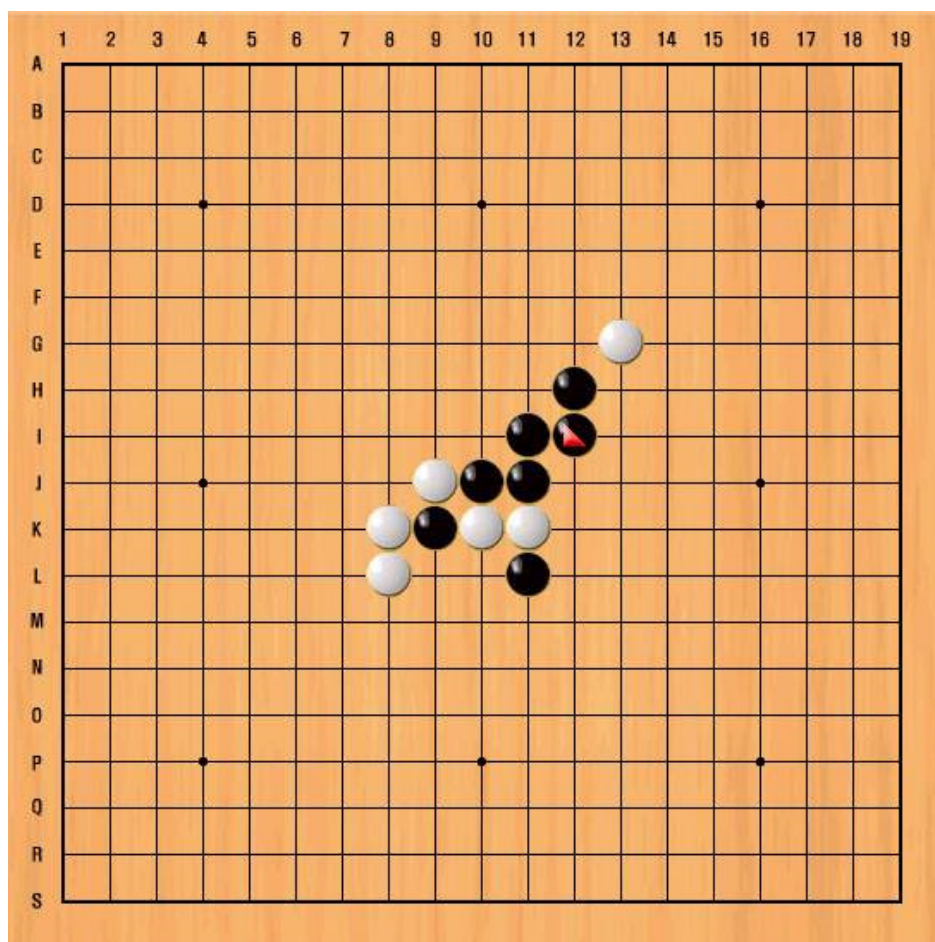
GO BANG

B96902025 資工二 張雅軒

B96902048 資工二 黃光世

B96902110 資工二 蘇緯倫

遊戲簡介



五子棋是起源於中國古代傳統黑白棋種之一，在中國有著悠久的歷史，又有良好的群眾基礎，幾乎人人都會下。相傳在「堯造圍棋」之前，在中國民間五子棋就很盛行了，就名稱而言，有「五子連」之說。《[山海經](#)》記載：「中次七經苦山之首，曰休輿之山。其上有石焉，名曰帝台之棋，五色而文其狀如鷓鴣卵」。這段話是說休輿之山上產有一種五色斑斕的石子，如鷓鴣蛋，叫做帝台之棋。可知道古代很早就用石頭當棋子，帝台之棋可能是五子棋或圍棋。其主要規則簡要的說即是雙方互持雙色棋，較對方先連成五子一線即獲勝。

操作說明

1.開始歡迎畫面 按下 start 鍵即可進入遊戲



2.進入遊戲後，由黑子先下，以上下左右控制游標，

A 鍵決定於游標位置下一新棋

B 鍵顯示敵方有利位置

L.R 鍵可選擇棋盤顏色

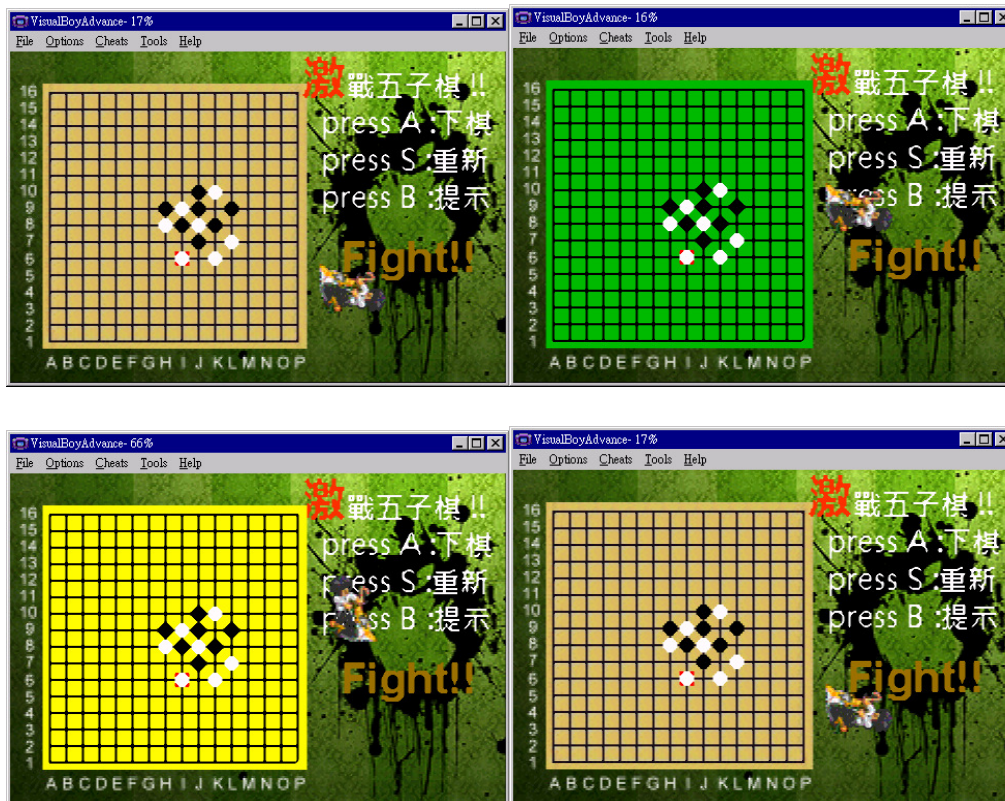
S 鍵可重新開啟新局



3.每當有一方五子連成一線，程式即會自動判斷遊戲輸贏，並顯示出勝利畫面



4. L.R 鍵可切換選擇棋盤顏色



開發環境

以 ham 為基礎利用其 lib 來完成大部分圖形介面的技術困難處，在主要程式判斷部份以組語完成，再以 function call 的形式，與處理圖形介面的 c 語言溝通。

函式說明

```
void chbcolor(u8* a,u8 b);
```

此函式用於改變旗盤的底色

```
void refresh(u8 *a,u8 b);
```

此函式用於將棋子貼上棋盤

原先我們是利用 HAM 語法中的 CreatObj，每下一顆棋便創造一個棋子的物件，讓他留在原地，但這種作法，棋子的個數會受限於 Object 的個數上限，128。因為棋盤是 16X16，棋子並不夠，且為了一個不會再動的棋子創造一個 Object，相當浪費記憶體，於是我們想到作業三所練習過的，利用組語做影像的處理。由於要貼上的棋子形狀固定，因此使用組語不需再判定外型，增加不必要的 bubble，減慢程式的運作。

```
int check_asm(u8 pos, u8 board[ ][ ], u8 color );
```

此函式用於判斷，在棋子落下後，是否獲勝。

運用組語最後一堂課時所講的平行式的 for 運算的概念，原先需用分開的八個 for 迴圈，從水平線、垂直線、左上斜線、左下斜線、判定是否有相連的五顆棋子。運用了這個概念後，只需一個 for 迴圈，去計算四個方向的棋子相連數，如此便可加快程式的運作。

```
Void AI (u8 input1,u8 input2)
```

此函式用於單機版的電腦下棋判斷，透過紀錄目前盤面棋子的陣列，開發出簡易的演算法，讓電腦自行判斷應如何下棋以達成有利於電腦的最佳盤面。

演算法最基本想法如下(若實作須稍加修改，而防守，進攻的演算法為之延伸、大約相同)：

使用者下棋後，掃描使用者所下之棋(假設在(i,j))的上下(左右、左上右下、左下右上)各 4 顆棋，若遇到空白便紀錄上個空白到此空白有幾個黑子，記錄在陣列 A 中，並記錄此空白之位址紀錄在 B 陣列中，掃描完後，若 $A[k]+A[k+1]=4$

則電腦在(B[k],j)處必定需要下棋。

由於是先在 dev C++ 上實作，在轉移至 HAM 時，遇到的最大 bug 是宣告 int 時不小心宣告成 u8，以至於出現許多問題，在這部分花的時間最多，也給我們學到了一個 unsigned integer 與 integer 用法的寶貴經驗。

遇到困難

在圖形介面上由於 lib 的不熟悉，花去許多時間在此方面上，相對的壓縮在組語方面的時間分配。但整體來說還是相當值得的，因為若不以 ham 提供的 lib 來實做圖形介面，想必會花更多時間來完成圖形介面的部份。並且也會失去實際運作組語與 c 這樣的高階語言溝通的機會。

另外一方面，在一開始挑選題目方面似乎並沒有選擇到一個很適合組語發揮的題目，以至於在圖形介面的問題解決之後，組語並沒有真正發揮其強大的效能。

預期目標

以 c 搭配組語來完成遊戲的內容。遊戲內容主體希望能由組語完成，在搭配 c 來實做圖形的使用者介面。希望能夠達到五子棋的主要對戰功能以及提供使用者基本的對戰提示等。

完成進度

在 ham 所提供的 lib 中，大致上已能熟悉背景、物件等的創立以及刷新所需之相關函式及其硬體相關知識。甚至使物件產生動畫，使物件依路徑移動等技巧也都大致能掌握。基本上一些圖形介面的應用以無大問題。

在組語方面，我們抽出主要判斷勝負與下子等 function，成功的以組語實做出來，並藉由 function 與 c 成功溝通。

另外在判斷對戰提示方面，在最後由於已實作出對戰提示，故順水推舟的以類似方式發展出一簡易 ai。基本上利用了簡單的 greedy algorithm，實作出一具有高度防守能力的 ai。

心得

在設計此遊戲過程中，我們學習到許多 c 語言與組語之間溝通的方式，並藉由此種方式，來達到以組語完成各個 function 的實做。並且在處理遊戲圖形介面時，同時瞭解到 gba 在圖形介面上的各種硬體實做方式。透過 ham 提供的強力 function 省去許多在圖形顯示上的困難，而 gba 的 arm 環境也讓我們更容易了解 arm 的架構。

雖然整個遊戲是以組語與 c 語言交錯著產生而得，但正因如此，讓我們更能夠體會平時 c 等高階語言實際在系統運作時所扮演的角色。以 arm 為開發平台

讓我們更能夠輕鬆的在這兩個語言之中溝通，加上 ham 各方面充足的 lib，讓我們更輕鬆的完成這個遊戲的設計。

雖然我們在這次題目中並沒有選擇到一個能讓組語發揮其效能的題目，但這更讓我們能夠去思考組語與高階語言之間的取捨，加上之前幾個作業下來的心得，讓我們更能了解組語這把利刃，究竟該在何時使用較為恰當，也比較能夠理解為何組語的應用目前都較集中在多媒體方面的原因。

參考資料

[http://en.wikipedia.org/wiki/Five_in_a_Row_\(game\)](http://en.wikipedia.org/wiki/Five_in_a_Row_(game))

<http://www.gbadev.org/>

http://theharbourfamily.com/jonathan/?page_id=89

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