

網路程式設計 project#1

組別：第二組

名稱：黑白棋

組員：

410285001 410285002 410285019 410285034 410285050

蔡 典 甫 鄭 力 瑋 孔 啟 熙 蕭 勝 興 林 詔 瑩

1. 設計動機與專案目標

黑白棋又叫翻轉棋，雙方各代表一種顏色來進行，是個老少咸宜的活動。據統計，全世界有超過 2 億的黑白棋愛好者，其受人們喜好程度可見一斑。這次的 project 我們決定不要用牌類遊戲，而是用這款流行 300 多年的黑白棋。

而我們的 project 目標一開始先寫出遊戲本體，再來試著弄 sever 與 client 連線，最後達成 sever 與 client 連線對戰的目標。

2. 程式架構與原始碼架構

我們使用 1、2 當作黑子與白子，分別給 sever 與 client 來操作
規則：每一步都要一定有轉換到棋子，如果在任一位置落子都不能夾住對手的任一顆棋子，就要讓對手下子，直到可以下為止。當雙方皆不能下子時，遊戲就結束，子多的一方勝。

程式介紹：

我們是傳送一個 table，去取代對手的 table

```
47     char sendTable[MAXDATASIZE];  
48  
49     int size=6;
```

雙方要輸入座標去決定要下哪裡

```
107     while(printf("輸入橫排%c~%c : 輸入直排1~%d :", (char)(row+1), (char)(row+size), size))
108     {
109         char inRow;
110         int inColumn;
111         // printf("輸入橫排%c~%c : 輸入直排1~%d :", (char)(row+1), (char)(row+size), size);
112         scanf(" %c %d", &inRow, &inColumn);
113
114         if((int)inRow < 65 || (int)inRow > 64+size || inColumn < 1 || inColumn > size)
115         {
116             printf("輸入錯誤\n");
117         }
118         else
119         {
120             check=1;
121             printf("輸入:%c%d 確認嗎? Y:1 N:0 ", inRow, inColumn);
122             scanf(" %d", &check);
123             if(check!=0)
124             {
125                 check=1;
126                 change=0;
127                 tableR=inRow-64;
128                 tableC=inColumn;
129
130                 if(table[tableR][tableC]!=0)
131                 {
132                     printf("此步無法下\n");
133                 }
134                 else
135                 {
136                     checkTable();
137                     if(change!=1)
138                     {
139                         printf("此步無法下\n");
140                     }
141                     else
142                     {
143                         table[tableR][tableC]=chess;
144                         break;
145                     }
146                 }
147             }
148         }
```

我們有一個 function 去看看這個點能不能下

```
241 void canChess()
242 {
243     check=0;
244     change=0;
245     for(tableR=1; tableR <= size; tableR++)
246     {
247         for(tableC=1; tableC <= size; tableC++)
248         {
249             if(table[tableR][tableC]==0)
250             {
251                 checkTable();
252             }
253         }
254     }
255     if(change!=1)
256     {
257         if(chess==1)
258         {
259             chess=2;
260         }
261         else
262         {
263             chess=1;
264         }
265     }
266 }
267
268 }
```

這是傳送 table 去給對手

```
151     tableToSendTable();
152     if (send(new_fd, buf, MAXDATASIZE, 0) == -1)
153         perror("send");
154     if ((numbytes = recv(new_fd, buf, MAXDATASIZE, 0)) == -1) {
155         perror("recv");
156     }
157     for(I=0;I<numbytes;I++)
158     {
159         sendTable[I]=buf[I];
160     }
161
162     sendTableToTable();
```

最後是檢查 table 上還有沒有點能下的 function

```
466 void changeTable(int A)
467 {
```

中略....

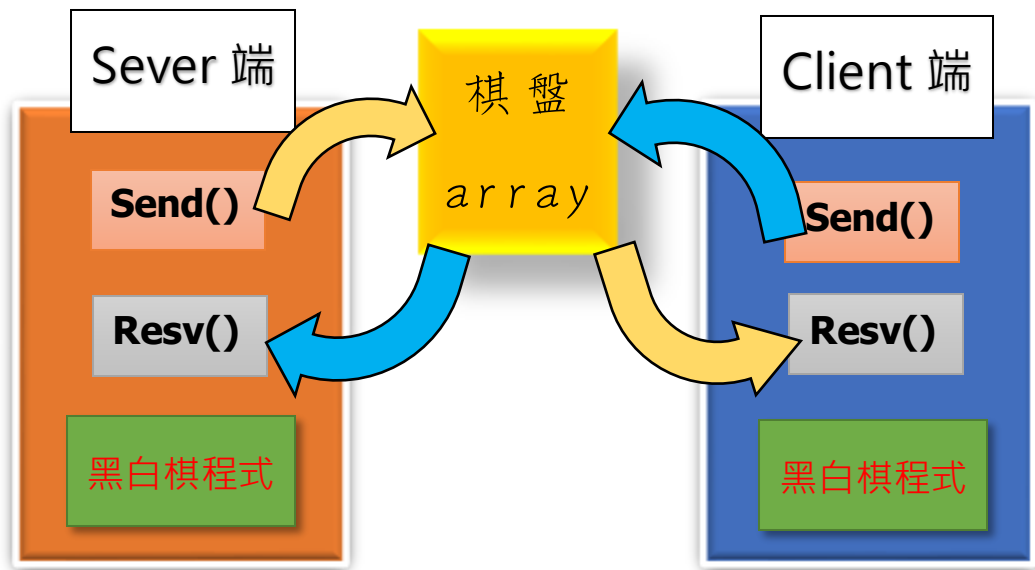
```
550     }
551 }
```

3. 測試方法 包含安裝和使用 並現場 demo

4. 所遭遇困難與克服方法

一開始我們的遊戲是用 c++ 的語法寫，到後來要連接 sever 與 client 的時候才發現要用 c 的語法寫，因此我們在從 c++ 轉換到 c 的語言中花了許多時間。再來是在 sever 與 client 互相 send 與 recv 的時候發現不能傳一個二維陣列(或者是可以，我們還沒參透)，所以我們將棋盤(table)裝進一個一維的陣列裡傳送，這裡也是我們找了許多相關書籍與資料，也詢問了很多 人，得到了這個結論，是沒有辦法中的辦法。

5. 程式穩定度與可用性



程式的穩定性應該不錯，我們都有防呆的 function 去防止玩家亂下或下錯。

程式的可用性可以寓教於樂,使兩位玩家在悠閒的時刻也能互相連線讓彼此進行一個黑白棋的對弈的動作。

6. 每位組員的分工和貢獻

程式：蕭勝興、鄭力瑋、蔡典甫

報告：蕭勝興、孔啟熙、林詔瑩

上台 demo：蕭勝興