資料結構期末專案

第九組

黑白棋

(Reversi)

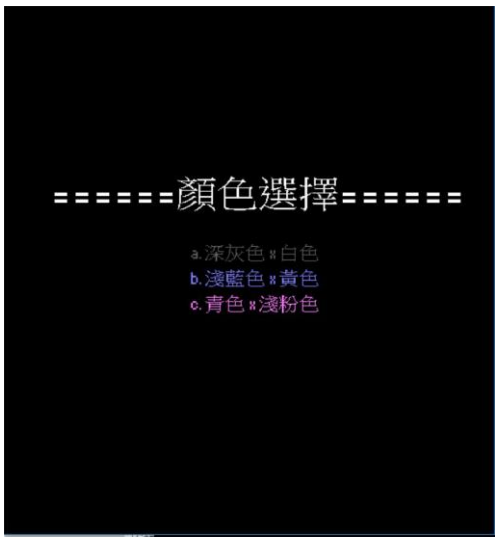
黃靖涵、黃欣惠、陳妍君

李欣倪、盧乙嫻

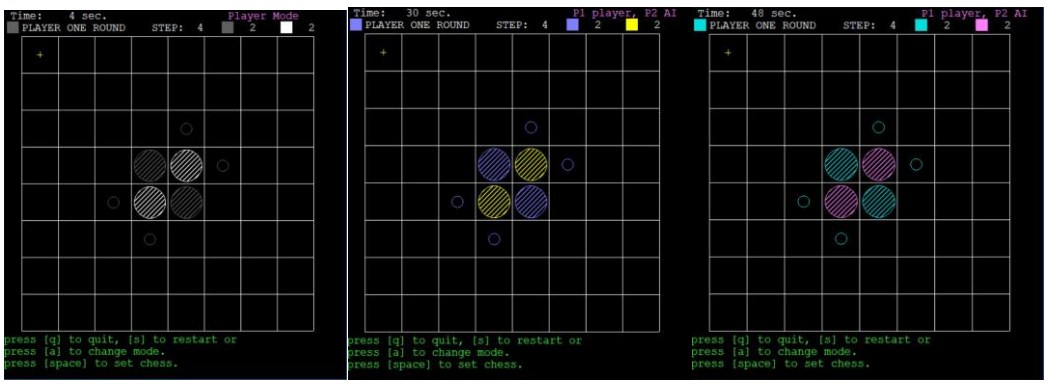
遊戲功能介紹

1. 主畫面：

2. 進入遊戲：



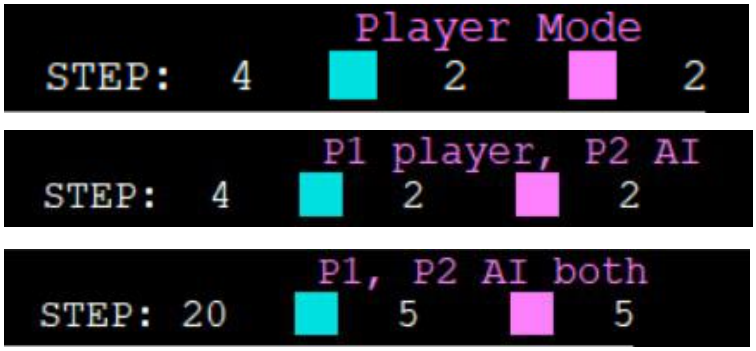
(a)進入顏色選擇頁面，其中有三種顏色組合可供使用者選擇，讓使用者選擇自己喜歡的顏色搭配。



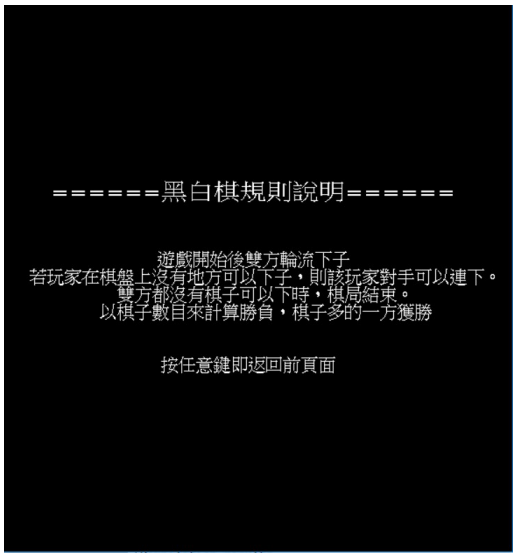
(b)選擇完顏色之後，即直接進入遊戲。

3. 選擇模式：

(a)在遊戲進行中，使用者可以切換三種不同模式，分別為玩家 vs 玩家、玩家 vs AI、AI vs AI。



4. 遊戲介紹：



5. 操作說明：

6. 遊戲結果：(a) 按 q 則可離開遊戲頁面。

(b) 按 s 則會回到選擇顏色的頁面，繼續遊戲。

主頁面

```
101  int main( )
102  {
103      openWindow();
104
105      while(TRUE){
106          Menu();
107      }
108  }
109
110  void startgame(){
111          speed = INIT_SPEED;
112
113          //設定遊戲場
114          int field[GRID_NUM][GRID_NUM] = {0};
115
116          playGame(field); //進行遊戲
117          if (key == 'q' || key == 'Q')
118              closeGame(); //如果玩家輸入'q'離開遊戲
119          else if (key == 's' || key == 'S')
120              Color();  //如果玩家輸入's' 繼續遊戲
121
122  }
```

函式 Menu、startgame、Color 執行順序：

1. 開啟視窗後，會先進入 Menu()函式。
2. 在 Menu()函式執行完之後，接著再到 startgame()函式。
3. 開始遊戲直到遊戲結束後，輸入 q 關閉遊戲，輸入 s 則返回選擇 Color()函式，繼續遊戲。

```
142     for(; ; j++){
143         if(kbhit()) {
144             choice1 = getch();
145             if (choice1 == 'a' || choice1 == 'A') {
146                 Color();
147             }
148             cleardevice();
149             if(choice1 == 'b' || choice1 == 'B'){
150                 ReversiRule();
151             }
152             cleardevice();
153             if(choice1 == 'c' || choice1 == 'C'){
154                 ReversiIntro();
155             }
156             cleardevice();
157             if(choice1 == 'q' || choice1 == 'Q'){
158                 closeGame();
159             }
160             cleardevice();
161         }
162     }
```

函式 Menu：

1. 若使用者輸入 a/A，利用 getch()讀入電腦，則連到 Color()，選擇完 Color 之後，直接進入遊戲。
2. 若使用者輸入 b/B，利用 getch()讀入電腦，即進入遊戲說明頁面。
3. 若使用者輸入 c/C，利用 getch()讀入電腦，即進入操作說明頁面。
4. 若使用者輸入 q/Q，利用 getch()讀入電腦，則離開遊戲。

選擇顏色頁面

```
27  int PLAYONE_COLOR; //設定玩家一顏色
28  int PLAYTWO_COLOR; //設定玩家二顏色
```

```
196     for(; ; r++){
197         if(kbhit()) {
198             choice2 = getch();
199             if (choice2 == 'a' || choice2 == 'A') {
200                 PLAYONE_COLOR = 8;
201                 PLAYTWO_COLOR = 15;
202                 startgame();
203             }
204             cleardevice();
205             if(choice2 == 'b' || choice2 == 'B'){
206                 PLAYONE_COLOR = 9;
207                 PLAYTWO_COLOR = 14;
208                 startgame();
209             }
210             cleardevice();
211             if(choice2 == 'c' || choice2 == 'C'){
212                 PLAYONE_COLOR = 3;
213                 PLAYTWO_COLOR = 13;
214                 startgame();
215             }
216             cleardevice();
217             if(choice2 == 'q' || choice2 == 'Q'){
218                 Menu();
219
220             }
221             cleardevice();
222         }
223     }
```

函式 Color：

1. 先將 PLAYONE_COLOR、PLAYTWO_COLOR 設為 integer 的全域變數。
2. 當玩家選擇 a、b、c 任一顏色組合之後，利用 getch()讀入電腦，便會使玩家一與玩家二的棋子顏色皆呈現該顏色組合。
3. 定義完雙方玩家之顏色後，即進入 startgame 函式，開始遊戲。

遊戲介紹頁面

```
226 void ReversiRule(){
227     int i = 0;
228     char msg1[40] = "======黑白棋規則說明======";
229     char msg2[40] = "遊戲開始後雙方輪流下子";
230     char msg3[80] = "若玩家在棋盤上沒有地方可以下子，則該玩家對手可以連下。";
231     char msg4[140] = "雙方都沒有棋子可以下時，棋局結束。";
232     char msg5[140] = "以棋子數目來計算勝負，棋子多的一方獲勝";
233     char msg6[140] = "按任意鍵即返回前頁面";
234 
235     settextstyle(SMALL_FONT, HORIZ_DIR , 8);
236     outtextxy(SCREEN_WIDTH / 9, SCREEN_HEIGHT / 2 - 50, msg1);
237 
238     settextstyle(SMALL_FONT, HORIZ_DIR , 6);
239     outtextxy(SCREEN_WIDTH / 3 + 10, SCREEN_HEIGHT / 2 + 30, msg2);
240 
241     settextstyle(SMALL_FONT, HORIZ_DIR , 6);
242     outtextxy(SCREEN_WIDTH / 2 - 220, SCREEN_HEIGHT / 2 + 50, msg3);
243 
244     settextstyle(SMALL_FONT, HORIZ_DIR , 6);
245     outtextxy(SCREEN_WIDTH / 2 - 120, SCREEN_HEIGHT / 2 + 70, msg4);
246 
247     settextstyle(SMALL_FONT, HORIZ_DIR , 6);
248     outtextxy(SCREEN_WIDTH / 2 - 140, SCREEN_HEIGHT / 2 + 90, msg5);
249 
250     settextstyle(SMALL_FONT, HORIZ_DIR , 6);
251     outtextxy(SCREEN_WIDTH / 2 - 70, SCREEN_HEIGHT / 2 + 150, msg6);
252 
253     for(; ; i++){
254         if (getch()){
255             cleardevice();
256             Menu();
257         }
258         delay(100);
259     }
260 }
261
```

函式 ReversiRule：

1. 將欲呈現的字放進 char 裡面
2. 用 settextstyle 設定字體、字體方向及字體大小
3. 用 outtextty 設定字體位置以及要呈現的字
4. 最後，若 getch 到任一鍵則 return

遊戲操作說明頁面

```
262 void ReversiIntro(){
263     int k = 0;
264     char mssg1[40] = "======黑白棋操作說明======";
265     char mssg2[40] = "↑ :向上 1 格";
266     char mssg3[80] = "← :向左 1 格";
267     char mssg4[40] = "→ :向右 1 格";
268     char mssg5[40] = "↓ :向下 1 格";
269     char mssg6[40] = "空白鍵(Space):下子";
270     char mssg7[40] = "按任意鍵即可返回前頁面";
271 
272     settextstyle(SMALL_FONT, HORIZ_DIR , 8);
273     outtextxy(SCREEN_WIDTH / 9, SCREEN_HEIGHT / 2 - 50, mssg1);
274 
275     settextstyle(SMALL_FONT, HORIZ_DIR , 7);
276     outtextxy(SCREEN_WIDTH / 2 - 20, SCREEN_HEIGHT / 2 , mssg2);
277 
278     settextstyle(SMALL_FONT, HORIZ_DIR , 7);
279     outtextxy(SCREEN_WIDTH / 2 - 25, SCREEN_HEIGHT / 2 + 30, mssg3);
280 
281     settextstyle(SMALL_FONT, HORIZ_DIR , 7);
282     outtextxy(SCREEN_WIDTH / 2 - 25, SCREEN_HEIGHT / 2 + 60, mssg4);
283 
284     settextstyle(SMALL_FONT, HORIZ_DIR , 7);
285     outtextxy(SCREEN_WIDTH / 2 - 20, SCREEN_HEIGHT / 2 + 90, mssg5);
286 
287     settextstyle(SMALL_FONT, HORIZ_DIR , 7);
288     outtextxy(SCREEN_WIDTH / 2 - 65, SCREEN_HEIGHT / 2 + 120, mssg6);
289 
290     settextstyle(SMALL_FONT, HORIZ_DIR , 7);
291     outtextxy(SCREEN_WIDTH / 2 - 110, SCREEN_HEIGHT / 2 + 180, mssg7);
292 
293     for(; ; k++){
294         if (getch()){
295             cleardevice();
296             Menu();
297            }
298         delay(100);
299     }
300 }
```

函式 ReversiIntro：

1. 將欲呈現的字放進 char 裡面
2. 用 settextstyle 設定字體、字體方向及字體大小
3. 用 outtextty 設定字體位置以及要呈現的字
4. 最後，若 getch 到任一鍵則 return

```
693   //繪製棋子
694   void drawChess(int row, int col, int color) {
695       int squareHeight = SCREEN_HEIGHT / GRID_NUM;
696       int SquareWidth = SCREEN_WIDTH / GRID_NUM;
697       int left = 0, right = 0, bottom = 0, top = 0;
698       left = LEFT_MARGINE + col * SquareWidth + 3;
699       top =  TOP_MARGINE + row * squareHeight + 3;
700       right = left + SquareWidth - 3;
701       bottom = top + squareHeight - 3;
702       int lr_center = (left + right)/2;
703       int td_center = (top + bottom)/2;
704
705       int r = 27;
706       setcolor(color);
707       setfillstyle(SLASH_FILL,color); //設定繪製物件的為實心和顏色
708       circle(lr_center,td_center,r);
709       floodfill(lr_center,td_center,color);
710   }
```

函式 drawChess：

1.用 setfillstyle 宣告顏色以及填色的方式

2.接著用 circle 函數，在座標(lr_center,td_center)畫圓形（棋子），設半徑為 r

3.用 floodfill 在座標(lr_center,td_center)填色，顏色為 color。

AI

```
843   //電腦玩家，請在此function實作AI功能
844   Location* PLAYONE_AI(int field[][GRID_NUM],  Location *focusPtr, NodePointer validated_locs){
845       // MAX_loc: 會翻最多棋子的location
846       // current_loc: 目前location
847       Location MAX_loc,current_loc;
848       int max_num=0, row, col;
849       // 看field中每個點，把每個點的location用reverse算會翻轉的棋子的個數
850       // 如果reverse return的值 > max_num 就把current_loc存到 MAX_loc 中，並把 max_num存成 MAX_loc 的 reverse 值
851       for(row=0; row < GRID_NUM; row++){
852           for(col=0; col < GRID_NUM; col++){
853               if(field[row][col] != EMPTY)
854                   continue;
855               current_loc.col=col;
856               current_loc.row=row;
857               printf("max location: [%d][%d]\tmax num: %d\n", current_loc.row, current_loc.col, max_num);
858               if(reverse(field, current_loc, true) > max_num){
859                   MAX_loc.col=current_loc.col;
860                   MAX_loc.row=current_loc.row;
861                   max_num = reverse(field, current_loc, true);
862               }
863           }
864       }
865       printf("\nmax location: [%d][%d]\nmax num: %d\n", MAX_loc.row, MAX_loc.col, max_num);
866       focusPtr->row = MAX_loc.row;
867       focusPtr->col = MAX_loc.col;
868       return focusPtr;
869   }
```

函式 PLAYONE_AI：

1. 宣告位置 struct MAX_loc 以及 current_loc，MAX_loc 用來儲存當下搜尋到可翻轉棋子數最多的可下子位置資訊，而 current_loc 則用來儲存目前的位置資訊。
2. 另外宣告整數變數 max_num，用來儲存當下搜尋到可翻轉棋子數最多的數量（初始值為零），由於此 AI 策略為「選擇當下可下的步數中，可翻轉對手棋子最多的一個」，所以後面每當搜尋到比 max_num 大的數字，就會存進 max_num。
3. 用兩個 for 迴圈去檢查棋盤(field)中每個點，把每個點的 location 傳入 reverse，去算該點會翻轉對手棋子的個數。如果該個數結果比 max_num 大，就會存進 max_num，並用 MAX_loc 去紀錄該點的位置資訊。
4. 等 for 迴圈跑完，就將 MAX_loc 位置資訊複製到 focusPtr，並回傳出去。

```
388         //繪製合法座標標記
389         int i=drawValidatedLocs(validated_locs);
390         if (i==1) {
391             endPoint=2;
392             showGameWinMsg();
393             return;
394         }
395         printf("end point: %d\n", endPoint);
396 
397     }
398 }
```

←在函式 playGame 中

```
561 //標記出所有該玩家所有合法的下棋位置
562 int drawValidatedLocs(NodePointer list){
563 
564     if (list==NULL){
565         printf("validated location do not exist!!\n");
566         if(playMode == AI_BOTH) return 1;
567         else if(playMode == AI_SINGLE) return 1;
568         else return 0;
569     }
570 
571     printf("validated location: ");
572     while(list != NULL){
573         printf("(%d, %d) ",  list -> loc.row, list -> loc.col);
574 
575         if(currentPlayer == PLAY_ONE)
576             drawCircle(list -> loc.row, list -> loc.col, PLAYONE_COLOR);
577         else
578             drawCircle(list -> loc.row, list -> loc.col, PLAYTWO_COLOR);
579 
580         list = list -> next;
581     }
582     printf("\n");
583     return 0;
584 }
```

函式 playGame 及函式 drawValidatedLocs 判斷：

1. 原本程式碼執行後，無法顯示遊戲結果(不能跑進函式 showGameWinMsg() 中)，所以在函式 drawValidatedLocs 中新增判斷式(556~568 行)。
2. 並讓函式可以 return boolean(0 代表遊戲尚未結束，1 代表遊戲結束)。
3. 在函式 playGame 中將函式 drawValidatedLocs return 的值存入變數 i 中，如果 i=1(遊戲結束)，就將 endPoint=2 後顯示遊戲結果，並跳出函式 playGame。

心得感想

這次的專案中，我們除了學習到許多有關<graphics.h>的應用外，也學習到如何看別人的程式碼。以前上程式設計時候寫的專案是沒有使用 graphics 函式庫的，整個程式幾乎都是手刻，所以畫出來的介面不僅很難對齊，也很難找出問題的原因，但是透過 graphics 函式庫，可以輕易地畫出直線、圓圈，也可以更有系統的寫程式。另外，因為在寫新的函式的同時也要與現有的函式結合，除了學習到不一樣函式庫的應用外，也要學習看懂別人的程式碼，將一個範例的程式碼理解之後，再加上自己的思考邏輯，因此在後續的應用上就會相當有效率，整個專案下來，也比較願意耐住性子去看別人的程式。

另外，這次的過程中也很深刻的體會到團隊合作的重要性，初期剛拿到助教給的程式碼的時候因為才剛接觸新的函式庫所以撞牆很久，但大家都會耐心地搜尋資料並一直嘗試，因為大家互相幫忙、漸漸熟悉函式庫之後，最後才能做出一個完整的遊戲出來。除了透過不斷上網查詢別人的作法並從中學習外，也學習到如何將持續測試，找出問題所在，不斷嘗試修正直到問題解決，所以此次專案讓我們收穫許多，也學習到許多新的事物，非常開心能有此次的學習機會!

參考資料

https://www.geeksforgeeks.org/setcolor-function-c/