

期末報告 Term project report (II)

科目名稱：C01001-計算機概論 I

授課老師：唐之瑋

組別編號：第 18 組

組員：蔡嘉倫 (106503008)、馮致齊 (106503009)

專題題目：五子棋

一、方案設計

1. 文字描述

初始化棋盤 15x15

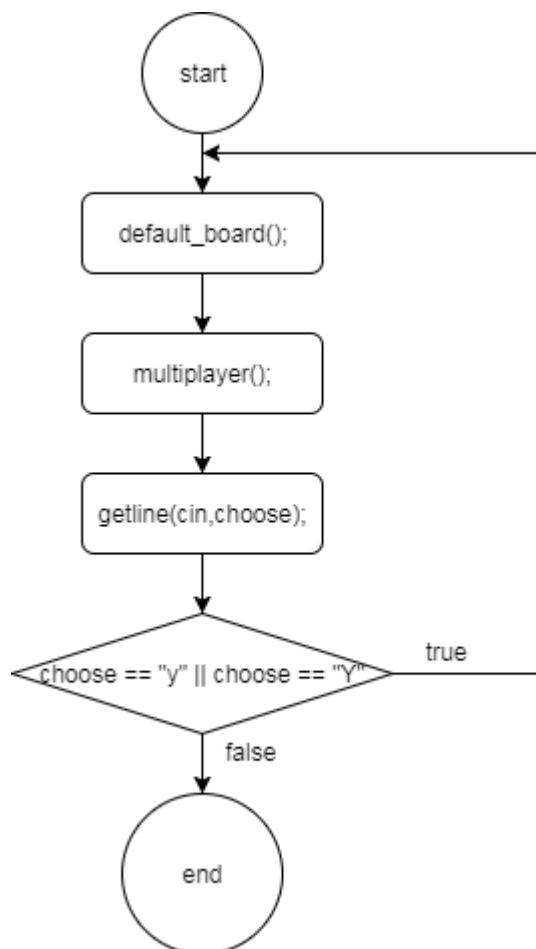
顯示棋盤 15x15

下棋+判斷有無棋子

勝負判斷

兩個玩家，黑棋先下

2. 流程圖



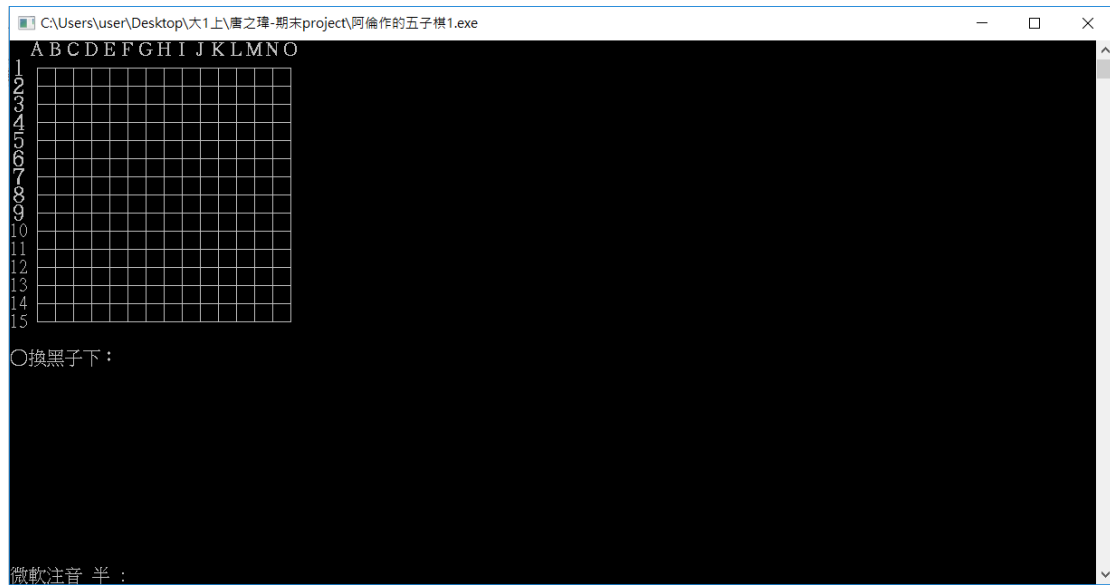
二、實驗結果

1. 文字描述

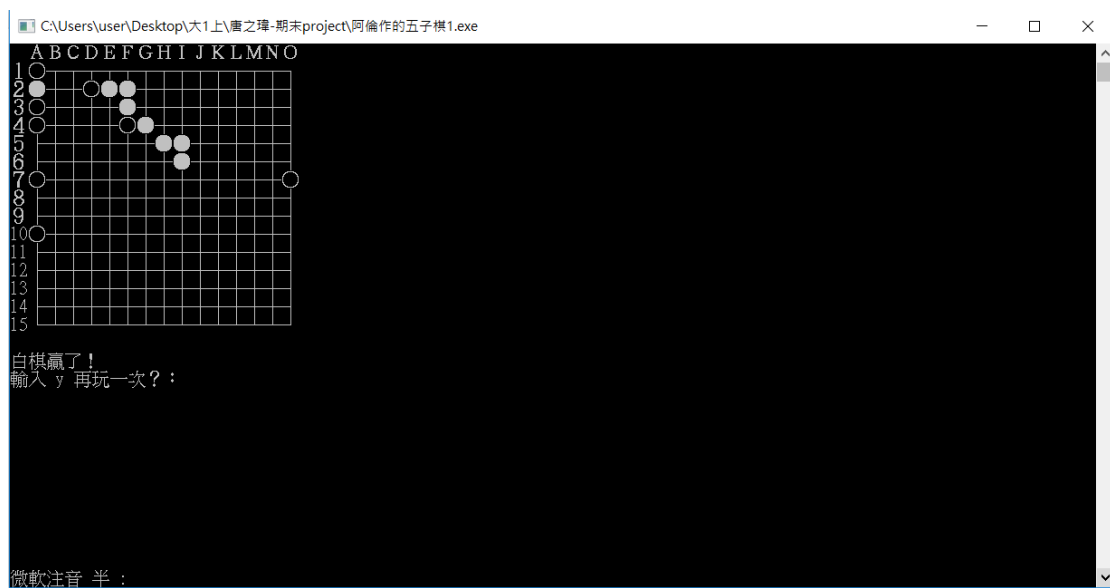
此為五子棋之程式，當程式執行時，必須輸入 A-0，與數字 1-15 之一，用以輸入所想下棋的位置。並且當五顆棋子連線成功時，系統便會告訴使用者是否想繼續下一局，若輸入 y 便能再次執行下一局，並且在輸入位置時若是超出其範圍，系統會自動告訴使用者：棋的範圍超出格子外！並且如果在重複的位置下棋的話，系統亦能告訴使用者：此位置已經有棋子了！若是使用了其他不是英文或是數字的字元便會顯示出：輸入了不合法的字元！最後一個重點即是此遊戲是由黑棋先下的。

2. 執行結果畫面截圖

a.) 遊戲過程之畫面



b.) 該局遊戲結束之結算畫面：



3. 部份重點 code 截圖

```

void default_board()
{
    for(int i=0 ;i<=16 ;i++){
        for(int j=0 ;j<=16 ;j++){
            if(i==0 && j==0){
                board[i][j] = 0;
            }else if(i==16 || j==16){
                board[i][j] = 0;
            }else if(i==0){
                board[i][j] = j+30;
            }else if(j==0){
                board[i][j] = i+10;
            }else if(i==1 && j==1){
                board[i][j] = -7;
            }else if(i==1 && j==15){
                board[i][j] = -9;
            }else if(i==15 && j==1){
                board[i][j] = -1;
            }else if(i==15 && j==15){
                board[i][j] = -3;
            }else if(i==1){
                board[i][j] = -8;
            }else if(j==1){
                board[i][j] = -4;
            }else if(j==15){
                board[i][j] = -6;
            }else if(i==15){
                board[i][j] = -2;
            }else{
                board[i][j] = -5;
            }
        }
    }
}

void put_chess(short i,short j,short whosturn){
    if(i == 0 || i >= 16 || j == 0 || j >= 16){
        cout <<"下棋的範圍超出格子外!"<< endl;
        endtry = 0;
        system("pause");
    }else if(board[i][j] == 1 || board[i][j] == 2){
        cout <<"此位置已經有棋子了!"<< endl;
        endtry = 0;
        system("pause");
    }else{
        board[i][j] = whosturn;
    }
}

```

```

bool winner(short i,short j,short whosturn){
    short count,tempi,tempj;
    bool win = 0;
    count = 1;
    tempi = i;
    while(board[--tempi][j] == whosturn){count++;}
    tempi = i;
    while(board[++tempi][j] == whosturn){count++;}
    if(count >= 5){win = 1; goto winner_check_end;}
    count = 1;
    tempj = j;
    while(board[i][--tempj] == whosturn){count++;}
    tempj = j;
    while(board[i][++tempj] == whosturn){count++;}
    if(count >= 5){win = 1; goto winner_check_end;}
    count = 1;
    tempi = i,tempj = j;
    while(board[--tempi][--tempj] == whosturn){count++;}
    tempi = i,tempj = j;
    while(board[++tempi][++tempj] == whosturn){count++;}
    if(count >= 5){win = 1; goto winner_check_end;}
    count = 1;
    tempi = i,tempj = j;
    while(board[++tempi][--tempj] == whosturn){count++;}
    tempi = i,tempj = j;
    while(board[--tempi][++tempj] == whosturn){count++;}
    if(count >= 5){win = 1; goto winner_check_end;}
winner_check_end:
    return win;
}

```

```

nextstep:
    I = 0;
    J = 0;
    endtry = 1;
    system("cls");
    show_board();
    if(whosturn == 1){
        cout <<"o 換黑子下：";
    }else{
        cout <<"• 換白子下：";
    }
    getline(cin,keyin);
    for(int n=0 ;n<keyin.length() ;n++){
        if(keyin[n]>='a' && keyin[n]<='o'){
            J = keyin[n] - 'a' + 1;
        }else if(keyin[n]>='A' && keyin[n]<='O'){
            J = keyin[n] - 'A' + 1;
        }else if(keyin[n]>='0' && keyin[n]<='9'){
            I = I*10 + keyin[n] - '0';
        }else if(keyin[n] == ' '){
            continue;
        }else{
            cout <<"輸入了不合法的字元！" << endl;
            system("pause");
            goto nextstep;
        }
    }
}

```

三、 實驗結果討論分析

此次程式設計，我們組參考了網路上現有的五子棋範本，學習他們的 function 的設計方式，包括如何達成 16*16 棋盤，並新增了些許功能，例如設計用輸入而得的棋子輸入法。最困難的部分在於如何判定遊戲結束的部分，如何利用程式將五顆棋子連成一條線作為判斷條件是相當困難的，最終經過討論我們仍得出了相當優質的改良並設計出程式內容。

四、 方案優缺點分析

優點：以此方式設計，能讓使用者一目瞭然。

缺點：可能缺少與 AI 遊戲的可能性，下棋方式有待改進。

五、 參考資料

- ▶ <http://monkeycoding.com/?p=908>
- ▶ <http://www.tcgs.tc.edu.tw/~sagit/cpp/q3.htm>

