

Data Structure Assignment 3
日本將棋對弈程式 v2.0 (Linked list)

通訊二 110503518 李秉宸

November 10, 2022

1 Version 版本紀錄

GitHub連結：https://github.com/BingChenLee/DSA_Assignment_2

v1.0 以陣列型態完成基本規則判定、持駒、持駒打入、升變、悔棋、棋譜紀錄、棋譜讀取與基本計時器。

v1.1 修復攻方可移動敵方部分棋子的問題。

v2.0 將對局過程改以Linked list存取。

2 Compile 編譯結果

本程式使用Makefile編譯程式。

```
bingchen@LAPTOP-34JN212B:~/110503518_assignment_2_new/Data_Structures_Assignment2_new$ make
gcc main.c -g -pthread -lncurses -lm -o main
cc -c -o main.o main.c
```

Figure 1: Makefile

3 Run 執行結果

```
bingchen@LAPTOP-34JN212B:~/110503518_assignment_2_new/Data_Structures_Assignment2_new$ ./main -n -s shogi.txt
```

Figure 2: 開新局

```
bingchen@LAPTOP-34JN212B:~/110503518_assignment_2_new/Data_Structures_Assignment2_new$ ./main -l shogi.txt
```

Figure 3: 讀取棋譜

輸入./main -n -s NEW_GAME_FILE_NAME.txt，將會開新局

輸入./main -l OLD_GAME_FILE_NAME.txt，將會讀取棋譜檔案。

遊戲移動輸入規則與Assignment 2相同。

存檔後的檔名為開新局前使用者自訂的NEW_GAME_FILE_NAME.txt。

複盤模式目前在輸入”b”（切換至上一手）仍存在一些問題待解決，亦尚無法顯示原始對局時間、累積花費時間，亦不可在任一手中做為新局繼續行棋。

4 Usage 程式解釋與使用教學

4.1 Link list使用位置

為了達到可以在回放時使用”f”以及”b”變更前後手，我使用Double linked list存取對局過程，同時在讀檔時可減免Reverse的過程。唯目前在輸入”b”時，發生預期外的部分資料失問題（如存取的座標自動變為0，但其餘資料正常存取），尚在尋找問題。

```
if(top >= STACKSIZE){
    printf("儲存空間已滿\n");
}
else{
    top++;

    if((tmp=(NODE*)malloc(sizeof(NODE))) != NULL){
        tmp->initial_row_memory = initial_row;
        tmp->initial_column_memory = initial_column;
        tmp->initial_chess_memory = initial_chess;
        tmp->goal_row_memory = goal_row;
        tmp->goal_column_memory = goal_column;
        tmp->goal_chess_memory = goal_chess;
        tmp->left = NULL;
        if(stack != NULL){
            tmp->right = stack;
            stack->left = tmp;
        }
        stack = tmp;
    }

    for(i=0; i<20; i++){
        X_chess_memory[top][i] = X_chess[i];
        Y_chess_memory[top][i] = Y_chess[i];
    }
}
```

Figure 4: Double Linked List(Push)

```

if(stack == NULL){
    printf("無法再悔棋\n");
    move_chess();
}
else{
    fseek(new_shogi_game, -18, SEEK_CUR);

    tmp = stack;
    stack = stack->right;
    checkerboard[tmp->initial_row_memory][tmp->initial_column_memory] = tmp->initial_chess_memory;
    checkerboard[tmp->goal_row_memory][tmp->goal_column_memory] = tmp->goal_chess_memory;
    free(tmp);

    for(i=0; i<20; i++){
        X.chess[i] = X_chess_memory[top][i];
        Y.chess[i] = Y_chess_memory[top][i];
        if(X.chess[i] != 0){
            Xmax++;
        }
        if(Y.chess[i] != 0){
            Ymax++;
        }
    }
    X_chess_top = Xmax;
    Y_chess_top = Ymax;
    top--;
    return TRUE;
}

```

Figure 5: Double Linked List(Pop)

4.2 計時器

本版本計時器仍使用time.h函式庫製作

- 1、每一手開始後，攻方玩家紀錄開始時間，輸入完畢後記錄完成時間。
- 2、將完成時間減去開始時間並累加至攻方玩家的計時器中。
- 3、示意圖可參考Assignment 2。

4.3 寫入與讀取棋譜

新局模式中，所有輸入方法與寫檔規則皆與Assignment 2相同。

複盤模式中，程式依下列順序執行棋譜讀取：

- 1、開檔案，檔名為使用者在getopt函式中自定義的名稱。（須為已存在且格式正確的txt檔）
- 2、程式在背景將每一手的移動方式完成並存於link list中（對局過程），與stack中（駒台）。
- 3、程式回到第一手顯示棋盤，使用者可輸入”f”移動至下一手或”s”結束程式。（英文字母大小寫皆可）
- 4、當使用者輸入”f”時，link list 的stack移動為stack指向right，駒台的stack top會+1並pop出原本的資料。
- 5、示意圖可參考Assignment 2。

5 Todo 待改善問題

5.1 王手放置

在基礎將棋規則中，若任一方已經王手（下一手可將死），則一方應優先拯救王將離險境，可使用的�方法包括但不限於移動王將、移動我方其他棋子以阻擋路綫、打入駒台上的棋子等。也就是，受威脅的那方不得任王將在下一手被將死，除非任何方式都無法拯救王將，此時的情況稱為”詰”，代表遊戲結束。

5.2 Libev 計時器與棋譜時間問題

本作業原先要求使用Libev函式庫製作計時器，但因目前尚未能完全理解其使用方法，故本次作業先使用time.h來完成計時器，目前計時器僅會顯示個人累積的行棋時間。

讀取棋盤時，螢幕所顯示的開始時間亦不是真正對局的時間，而是當下的系統時間，且不會顯示當時每一手所花費的時間，整體尚待改進。

5.3 棋譜格式問題

預計在未來將棋譜新增對局開始與結束時間，每一手的攻方，每一手行棋所花費的時間與總累積時間等資料。

5.4 細節顯示問題

因程式中使用system(“clear”)函式，導致部分訊息顯示瞬間即被清除，此部分仍在尋找解決方法。

5.5 Double linked list問題

目前未知原因，讀取棋譜後輸入”b”，會導致部分資料失，已確認push資料過程中，linked list連結沒問題。

6 Reference 參考資料

- [1] 日本將棋走法玩法 <https://shogi.hk/Gameplay-of-Japanese-Chess-Shogi/>