DS Assignment3 Shogi Report

電機 4B 108501537 蔡雨蓁

1. Build Guide
   - Linked-list version
     - `gcc -o main main.c -lev`
2. Execute Guide
   - Linked-list version
     - Play:

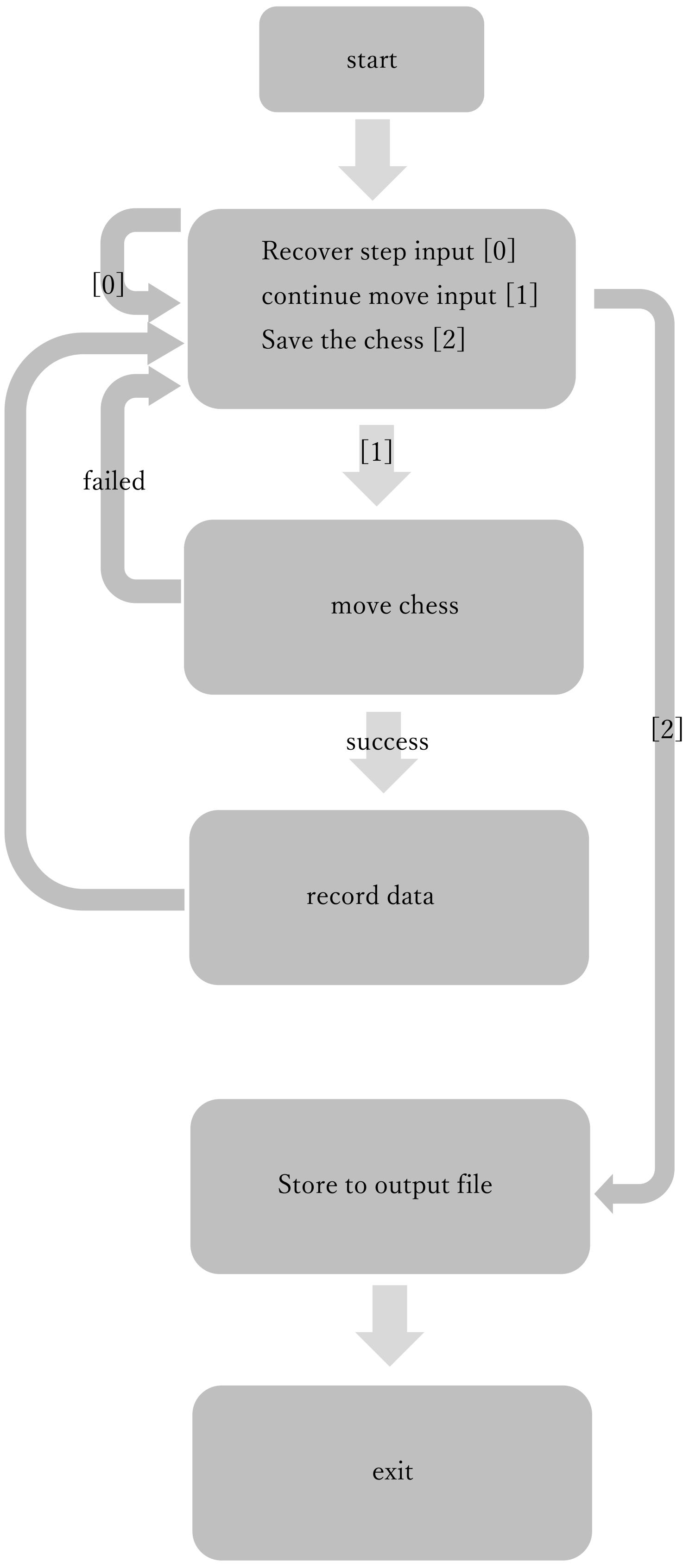
       ```
       ./main -n -s new_game_file_name
       ```
     - Load manual:

       ```
       ./main –l old_game_file_name
       ```

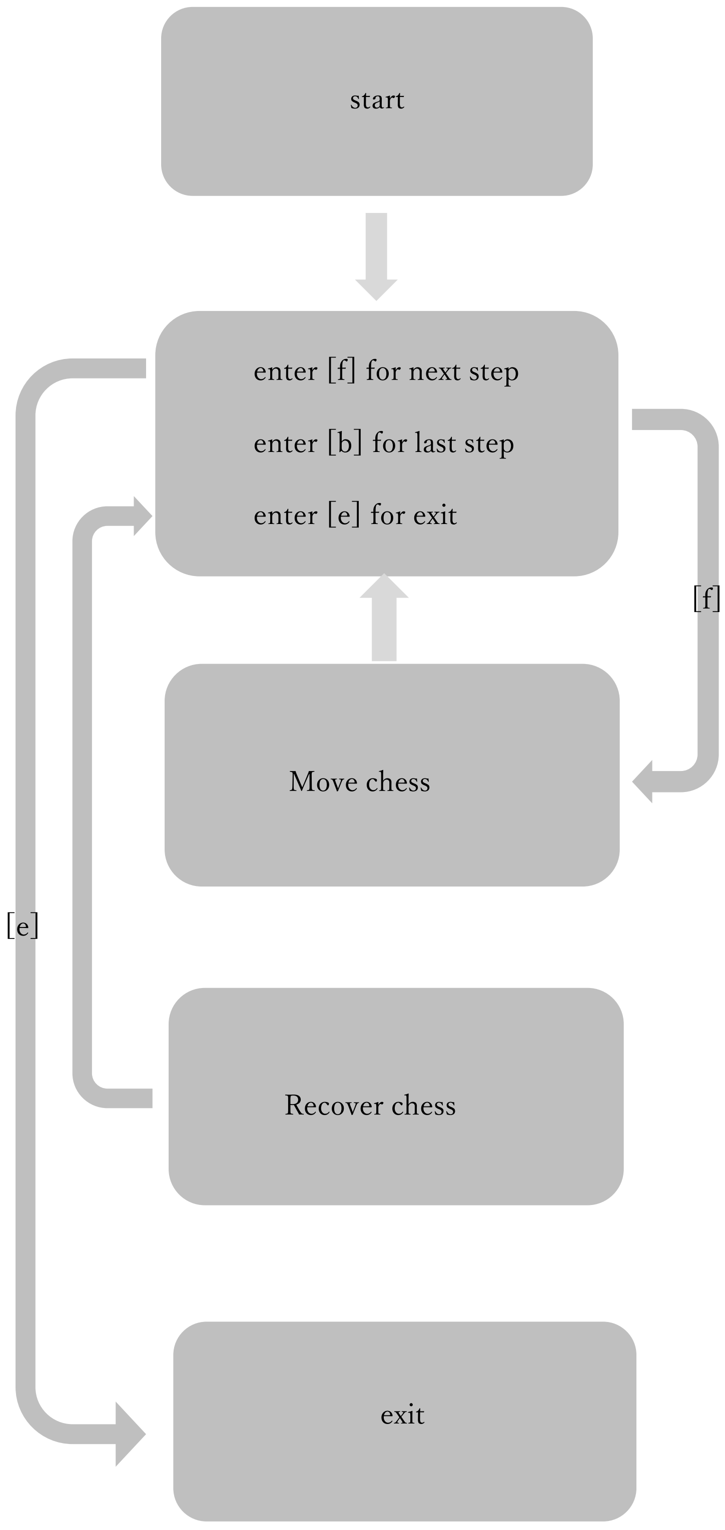
(flow chart 太大放到下一頁)

3. Flowchart

a. Play a new game



b. Load manual



4. System Architecture

- 2D array which element is “struct piece” to store each grid on the board

Struct	…	Struct
Struct	…	…
…	…	…
Struct	…	Struct

- The struct consist of:

```
struct piece {
    char name[NAMESIZE];
    char controller[STRLENGTH];
};
```

name -> chess type
controller -> chess color

```
struct MOVE_Linked_List{

    int MOVE_Linked_List[MOVESIZE] ;
    struct MOVE_Linked_List *next ;

};
```

MOVE_Linked_List -> record the step (x1,y1) to (x2,y2)

> MOVESIZE = 4

- In Linked List version, each player step will be record in a stack which built in linked-list

5. Function Introduction

```
void board_initial(struct piece board[][BHEIGHT])
```

>初始化棋盤

```
void board_show(struct piece (*board)[BHEIGHT])
```

>顯示棋盤

```
void swap(struct piece *A,struct piece *B)
```

>交換棋子

```
int eat(struct piece chess[][BHEIGHT] ,int before_x ,int before_y, int
after_x, int after_y)
```
>吃棋子

```
int move_chess(struct piece chess[][BHEIGHT] ,int before_x ,int
before_y, int after_x, int after_y)
```
>移動棋子

```
int legal_position(struct piece chess[][BHEIGHT] ,int before_x ,int
before_y, int after_x, int after_y, int attacker)
```
>判斷移動是否為合法位置

```
int Push(struct stack *all_chess_stack_Ptr,struct piece
chess[][BHEIGHT])
```
>HW2 之 Stack Push

```
int Pop(struct stack *all_chess_stack_Ptr,struct piece
chess[][BHEIGHT])
```
>HW2 之 Stack Pop

```
MOVE_Linked_List *Push_move(MOVE_Linked_List *chess_move_Ptr, int
before_x, int before_y, int after_x, int after_y)
```
>HW3 之 linked list 之 Push (x1,y1) 到 (x2,y2)

```
MOVE_Linked_List *Pop_move(MOVE_Linked_List *chess_move_Ptr)
```
>HW3 之 linked list 之 Pop

```
void Write(FILE *fptr , MOVE_Linked_List * chess_move_Ptr, struct stack
*all_chess_stack_Ptr )
```
>寫檔

```
int get_move(MOVE_Linked_List *read_move_Ptr,int *read_x_b, int
*read_y_b,int *read_x_a,int *read_y_a,int take)
```
>讀檔

```
int return_move(MOVE_Linked_List *read_move_Ptr, int take)
```
>回復上一動