# 通訊三 109503509 林俐嫺 資料結構 Assignment2 分析報告

#### 一、編譯結果

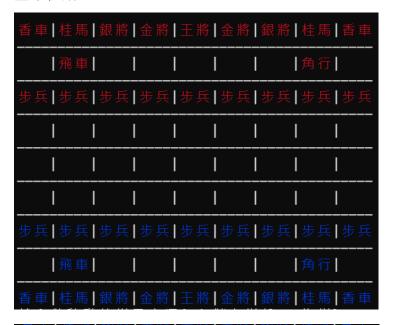
a28896165@LAPTOP-7JFN2AP5:/mnt/c/Users/user/Desktop/DSA\_assignments/Assignment2\$ make gcc ./inc/Assignment\_2.h ./src/Assignment\_2.c -I ./inc -c gcc ./build/Assignment\_2.o -o ./bin/Assignment\_2

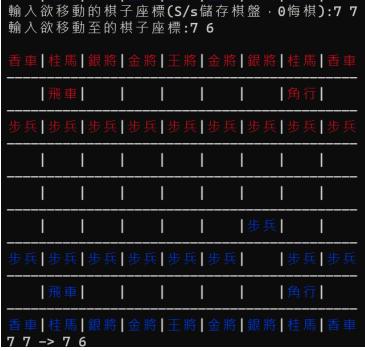
#### 二、執行結果

#### 1. 開新局

'Desktop/DSA\_assignments/Assignment2\$ ./bin/Assignment\_2 -n -s game1.txt

a. 基本移動





## b. 悔棋

香車 桂馬 銀將	金將   王將	金將 銀將	桂馬 香車
飛車	l I	I I	角行
步兵 步兵 步兵	l I	歩兵 歩兵	步兵 步兵
1 1	歩兵 歩兵	 	I I
1 1	l I		I I
1 1	l I	I I	  步兵
步兵 步兵 步兵	歩兵 歩兵	步兵	
	l I	I I	 <b> </b> 角行
香車 桂馬 銀將  76->75	金將 王將	  金將 銀將	  桂馬 香車
輸入欲移動的棋	子座標 <b>(</b> S/s <sup>/</sup>	儲存棋盤,	0悔棋):0

香車 桂原	馬 銀將	金將	王將	金將	銀將	桂馬	香車
飛耳	<b>≢</b>	I	l	I	I	角行	<del>-</del> [
步兵 步身	兵 步兵		I	步兵	步兵	步兵	步兵
I	ı	步兵	步兵		I	I	I
I	I	I	I	I	I	I	I
	ı	l		ı	步兵	步兵	
步兵 步身	兵 步兵	步兵	步兵	步兵	[	1	步兵
飛車	<b>‡</b>   	I	I	I		角行	1
香車 桂馬 銀將 金將 王將 金將 銀將 桂馬 香車 退回上一步,換Player1輸入。							

## c. 儲存棋盤

輸入欲移動的棋子座標(S/s儲存棋盤,0悔棋):s 儲存棋盤 遊戲結束

```
a28896165@LAPTOP-7JFN2AP5:/mnt/c/Users/user/Desktop/DSA_assignments/Assignment2$ ls bin build doc game1.txt inc makefile README.txt Resultfile.txt res
```

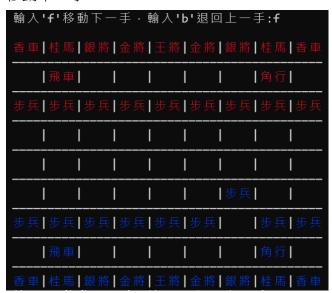
7 7 7 6 5 3 5 4 8 7 8 6 4 3 4 4 (game1.txt 的内容)

## 2. 開舊局

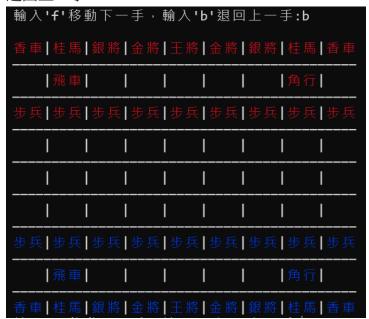
/Assignment2\$ ./bin/Assignment\_2 -l game1.txt

香車 桂馬 銀將	金將	王將 金牌	將┃銀將	桂馬	香車
飛車	l l	ı	ı	角行	 
步兵 步兵 步兵	步兵	步兵 步!	兵 步兵	步兵	步兵
1 1	l l	I	I	I	
1 1	I I	ı	ı	I	 
1 1	l l	ı	ı	I	I
步兵 步兵 步兵	歩兵	步兵 步	兵 步兵	步兵	步兵
飛車	1 1	ı	I	角行	
香車   桂馬   銀將   金將   王將   金將   銀將   桂馬   香車 輸入 'f'移動下一手,輸入 'b' 退回上一手:					

a. 移動下一手



#### b. 退回上一手



### 三、 問題檢討與心得

目前還沒有完成計時器的功能,關於悔棋、升變等等進階的機制也還沒有寫得非常完整,因此這個程式還只能說是非常基本的將棋移動程式。

在這一次的作業裡,我是將別人的開源程式做為參考,自己從 0 開始編寫。比起使用陣列儲存 structure,我覺得用 Linked list 會更容易實現儲存移動座標這個部分,所以我是先寫 Assignment3(linked list)之後才往回改的,遇到比較大的問題反而是在改陣列的時候,函式宣告也需要做更改,我花了很多時間在尋找 double pointer 和 pointer to array 的對照,剛好也能複習期中考的範圍。

希望未來除了將功能補得更加完整以外,也能將程式精簡化(如流程圖所示,我在第一次輸入座標及第二次輸入座標時,過程幾乎都一樣,只是因為儲存的變數不同而必須拆開),在寒假期間可以多想想要怎麼將這個部分改掉。