

# Final Project

---

姓名：謝皓青

學號：40947021S

## 目錄

---

- 檔案
- 結構
- 定義
- 設定函式
- 遊戲函式
- 職業函式

## 檔案

---

### 標頭檔

- setting.h
- role.h
- game.h

### C 檔

- main.c
- setting.c
- game.c
- builder.c
- councilor.c
- producer.c
- prospector.c
- trader.c

## 結構

---

### 卡片結構

```
1  struct _Card
2  {
3      char *name;           // 卡片名稱
4      uint8_t cost;         // 費用
5      uint8_t points;       // 得分
6      uint8_t color;        // 牌面顏色
7      bool is_special_struct; // 是否為特殊建築
8      bool have_good;       // 是否有貨物
9  } __attribute__((__packed__));
10 typedef struct _Card Card;
11
```

## 玩家結構

```
1 struct _Player
2 {
3     uint8_t id;           // 玩家編號
4     uint8_t cards;        // 手牌數量
5     uint8_t builds;       // 建築數量
6     uint32_t points;      // 積分
7     Card *board[12];      // 版面卡片
8     Card *hand[20];       // 手上卡片 (防止 overflow)
9     bool is_bot;          // 是否為電腦玩家
10 } __attribute__((__packed__));
11 typedef struct _Player Player;
12
```

## 定義

### 卡片顏色

```
1 #define BLUE 0
2 #define WHITE 1
3 #define BROWN 2
4 #define COFFEE 3
5 #define GRAY 4
6 #define PURPLE 5
```

### 清除輸入緩存

```
1 #define ERROR printf("\033[1;31mInvalid input\033[0m\n");
```

### 錯誤訊息

```
1 #define ERROR printf("\033[1;31mInvalid input\033[0m\n");
```

## 設定函式

遊戲前的各項設定，包含在 setting.h

### void menu()

輸出歡迎介面並選擇開始遊戲或進入注意事項畫面

### void setup()

設定開局 ex: 玩家資訊、牌堆

### Card \*deal(Card \*deck[])

從牌堆中發一張牌

## **void caution()**

輸出注意事項

## **void set\_player(Player \*player\_list[], Card \*deck[])**

初始化玩家資訊(編號、手牌 etc)並且每人發四張牌和蓋一個染坊

## **void set\_card(Card \*deck[])**

初始化 110 張牌到牌堆

## **遊戲函式**

---

除了職業函式以外函式或共用的函式，包含在 game.h

## **void game()**

遊戲進行中的主函式掌控每一回合的進行

## **uint8\_t select\_role(uint8\_t turn, uint8\_t role\_list[])**

讓玩家選擇職業，回傳職業代碼並記錄已選擇的職業使下一次選擇時不會重複選擇

## **void show\_hand(Player \*player)**

顯示玩家手牌

## **void check\_status()**

查看各玩家的狀態包括：版面、分數

## **uint8\_t update()**

更新玩家資訊基本上職責等同總督

## **void overflow(Player \*player)**

如玩家手牌超過上限則需丟棄

## **void draw(Player \*player, int8\_t take, int8\_t keep)**

抽取 take 張牌並且保留 keep 張牌

## **void chapel(Player \*player)**

有 Chapel 的玩家得以選擇一張手牌至於 Chapel 中

## **void show\_supply()**

顯示牌庫剩餘幾張牌

## **void recycle(Card \*card)**

將丟棄的牌回補至牌堆

## **void gameover()**

遊戲結束結算分數選出贏家

## **職業函式**

---

和職業相關的所有函式，包含在 role.h

## **void builder(uint8\_t turn)**

建築師職業的主函式，使每一位玩家都執行一次建築行動

## **int8\_t select\_card(Player \*player, uint8\_t smithy, uint8\_t quarry, uint8\_t cost\_discount)**

讓玩家選擇要興建甚麼建築，回傳 -1 代表不興建否則回傳卡片在手牌中的位置

## **int8\_t check\_card(Player \*player, int8\_t idx, uint8\_t cost\_discount, uint8\_t smithy, uint8\_t quarry)**

檢查所選的卡片消費減去各種情況的消費減免，是否可興建，回傳 -1 代表否 1 代表是

## **int8\_t discard(Player \*player, int8\_t idx, int8\_t card\_idx)**

將選定的手牌引數 idx 丟棄並且不得跟要興建的手牌引數 card\_idx 重複，回傳 1 代表成功 -1 代表失敗

## **uint8\_t blackMarket(Player \*player)**

讓玩家決定是否使用黑市，回傳支付的貨物數

## **void build(Player \*player, int8\_t card\_idx, uint8\_t cost\_discount)**

興建指定手牌引數 card\_idx 的卡片

## **void Crane(Player \*player, uint8\_t smithy, uint8\_t quarry, uint8\_t carpenter, uint8\_t cost\_discount)**

起重機的主程式，選擇要被加蓋的建築物

## **int8\_t craneCheck(Player \*player, int8\_t src, int8\_t target, uint8\_t smithy, uint8\_t quarry, uint8\_t cost\_discount)**

檢查欲建造建築的手牌引數 src 和被拆除的卡片引數 target，決定是否可以興建，回傳 -1 代表否不然回傳最終消費

## **void councillor(uint8\_t turn)**

議員的主程式

## **void Archive(Player \*player, int8\_t take, int8\_t keep)**

設計類似 draw() 但會先將抽的牌全部加到手牌再選擇拋棄

## **void producer(uint8\_t turn)**

生產者的主程式

## **void produce\_good(Player \*player, int8\_t n)**

生產貨物，如果可生產建築數 > 可生產貨物數玩家可選擇要用哪個建築生產，其餘情況系統自動生產  
如果為 level 2 電腦會選擇最高利潤的選項

## **void prospector(uint8\_t turn)**

淘金者的主程式，如果判定玩家有金礦則進入金礦函式

## **void goldMine(Player \*player)**

先抽 4 張牌再判斷是否有兩兩重複，如果沒有則挑選 1 張保留

## **void trader(uint8\_t turn)**

商人的主程式，會先隨機選擇一張價目表

## **void sell(Player \*player, uint8\_t n, uint8\_t market\_hall)**

選擇要販賣的貨物，系統會告知最多可販賣多少貨物，需自行決定要販賣哪些，如果沒有貨物則不會顯示  
桌面須直接輸入 0 結束  
如果為 level 2 電腦會選擇高價的賣出