

San Juan Code Documentation

main.c

Struct

宣告

```
1  uint32_t ACT_cards[10]={0};
2  uint32_t Trader_List[6][6]={{0,0,0,0,0,0},{0,1,2,2,3,3},{0,1,1,2,2,3},{0,1,2,2,2,2},
3  int32_t LT,pg_flag=0,match_flag=0;
```

1. uint32_t ACT_cards[10]={0};

宣告行動卡

2. uint32_t Trader_List[6][6]={{0,0,0,0,0,0},{0,1,2,2,3,3},{0,1,1,2,2,3},{0,1,2,2,2,3}}

建立商品價目表

3. int32_t LT,pg_flag=0,match_flag=0;

LT: 商品價目表的指針。

其他: 利用模運算判定輪、回合、Governor的交換的指針。

函式

```
1  uint32_t pick_one_card(void)
2  uint32_t pick_one_trader_list(void)
3  void print_SANJUAN(void)
4  void init(void)
5  void pick_action_card(void)
6  void print_number_list_all_cards(void)
7  void print_hand_card(void)
8  void print_self_building(void)
9  void print_table(void)
10 void choose_number(void)
11 void abandon_card(int32_t num)
12 void Carpenter(int32_t n)
13 void Poor_House(int32_t n)
14 int32_t Trading_post(int32_t n)
15 void Market_Hall(int32_t n)
16 void Gold_Mine(int32_t n)
17 void computer_act(int32_t p, int32_t num)
18 void computer_act_pri(int32_t p, int32_t num)
19 uint32_t computer_pick_action_card(void)
20 void new_match(uint32_t pg)
```

1. uint32_t pick_one_card(void)

抽一張卡，如果牌堆是空的，就重新洗其牌堆的牌。

2. uint32_t pick_one_trader_list(void)

有玩家當前行動式 Trader，就要抽取桌子中央的商品價目表。

3. void print_SANJUAN(void)

印出由 / _ 組成的 SAN JUAN 字體。

4. void init(void)

將玩家的資訊、vp、手牌、桌面上的牌初始化

5. void pick_action_card(void)

選擇職位，如果當輪選到重覆的會有錯誤訊息，並要求你重新選擇

6. void print_number_list_all_cards(void)

輸出一個清單，含有數字所對應的卡牌名字

7. void print_hand_card(void)

印出自己的手牌

8. void print_self_building(void)

印出自己的所蓋的建築

9. void print_table(void)

印出桌面上其他玩家所蓋的建築

10. void choose_number(void)

印出選單，讓你選擇選單上你欲執行的事

11. void abandon_card(int32_t num)

將排放入棄牌堆

12. void Carpenter(int32_t n)

Carpenter卡牌效果

13. void Poor_House(int32_t n)

Poor_Huouse卡牌效果

14. int32_t Trading_post(int32_t n)

Trading_post卡牌效果

15. void Market_Hall(int32_t n)

Market_Hall卡牌效果

16. void Gold_Mine(int32_t n)

Gold_Mine卡牌效果，內含有玩家自己選擇與電腦玩家的自動選擇。

17. void computer_act(int32_t p, int32_t num)

電腦玩家在 normal action 時的所有操作。p 是第 p 位玩家，num 是當前拿著行動卡的職位是什麼。

18. void computer_act_pri(int32_t p, int32_t num)

電腦玩家在拿著行動卡時的所有操作。p 是第 p 位玩家，num 是當前拿著行動卡的職位是什麼。

```
19. uint32_t computer_pick_action_card(void)
```

電腦玩家抽取行動卡

```
20. void new_match(uint32_t pg)
```

每一輪、每一回合、每一次換 Governor，進行遊戲抽卡、判定卡牌技能、選取選單執行... 的所有操作

```
21. int32_t calculate_vp(int32_t n)
```

遊戲結束時計算總分



卡牌技能

因為在 San Juan 中有提到本支程式比較高耦合，因此只有部分卡牌技能是分開成獨立函式進行，所以其他卡牌的技能都黏在 `computer_act` `computer_act_pri` `calculate_vp` `new_match` 裡

零星介紹

- 627~638行: 利用模運算判定輪、回合、Governor的交換。
- 641~709行: 換 Governor 時要檢查每位玩家手牌不超過 7 張，如果有 Tower 就是不超過 12 張
- 1821~1888行: 計算哪一位玩家獲勝

basic.h

- 定義顏色

```
1  #define RED_BOLD "\x1b[;31;1m"
2  #define BLU_BOLD "\x1b[;34;1m"
3  #define YEL_BOLD "\x1b[;33;1m"
4  #define GRN_BOLD "\x1b[;32;1m"
5  #define PURPLE "\e[1;35m"
6  #define CYAN_BOLD_ITALIC "\x1b[;36;1;3m"
7  #define RESET "\x1b[0;m"
```

- 玩家與卡牌的結構

```

1  typedef struct _Cards {
2      uint32_t status; //0: in the card pile  1: discard  2: in the abandoned
3      uint32_t cost;
4      uint32_t vp;
5      uint32_t w_building; // 1: produce  2: special
6      uint32_t cargo;
7  } __attribute__((__packed__)) Cards;
8
9  typedef struct _player {
10     uint32_t Governor;// 0: No 1: Yes
11     uint32_t action_card;//0: No 1: Builder 2: Producer 3: Trader 4: Council
12     uint32_t cnt_have_building;
13     uint32_t cnt_hand_card;
14     Cards building_cards[120];// building on the table
15     // 1:have this card  0: ~1  all: 110 building cards
16     uint32_t hand_cards[120];// your hand card
17     // 1:have this card  0: ~1  all: 110 building cards
18     uint32_t Chapel_vp;
19 } __attribute__((__packed__)) player;
20
21 player Player[5];
22 Cards All_Cards[120];

```

- Cards:
 - status:
 - 0: 在卡牌堆裡
 - 1: 在某位玩家手裡
 - 2: 在棄牌堆
 - cost: 建造這張牌的價格
 - vp: 這張牌的價值
 - w_building:
 - 1: 生產型建築
 - 2: 特殊形建築
 - cargo
 - Chapel: 記錄教堂下面牌數
 - 生產型建築: 生產出的貨物
- player:
 - Governor: 是否為 Governor
 - action_card: 持有行動卡的人執業式什麼?
 - cnt_have_building: 玩家有多少棟建築
 - cnt_hand_card: 玩家有多少張手牌
 - building_cards: 玩家桌前已經蓋了什麼建築
 - hand_cards: 手牌有哪些

comp.c

```
1 void print_player_card(int32_t p);  
2 void print_player_building(int32_t p);
```

1. void print_player_card(int32_t p)

印出其他玩家有幾張手牌，不會顯示他的牌是什麼。

2. void print_player_building(int32_t p)\

顯示其他玩家蓋了多少棟建築、建築是什麼、有多少貨物。