40947018S 劉鎧葶

# 程式設計二 期末專題 – 聖胡安桌遊

## 簡介:

1. 根據該桌遊規則編成
2. 中文介面
3. 可挑選難度: 普通版(無時間限制)、燒腦版(操作時間限制 30 秒、不含選擇職業)//beta
4. 你可以跟 3 個機器人(隨機選擇)對戰

## 編譯:

make

## 執行:

./main

## Function Reference：

- welcome
  列印歡迎頁面，選擇執行動作
- void start(struct list_head *player_list_head_1, struct list_head *player_list_head_2, struct list_head *player_list_head_3, struct list_head *player_list_head_4, struct list_head *build_list_head_1, struct list_head *build_list_head_2, struct list_head *build_list_head_3, struct list_head *build_list_head_4)
  執行準備動作(發牌)
- _splayer_card *add_newcard(_sbuild *building)
  增加新卡
- _sbuild *draw_card()
  從牌堆中抽卡
- int32_t gameround(int32_t roundnum, const int32_t playernum, const int32_t tradecardnum, struct list_head *player_list_head_1, struct list_head *player_list_head_2, struct list_head *player_list_head_3, struct list_head *player_list_head_4, struct list_head *build_list_head_1, struct list_head

*build_list_head_2, struct list_head *build_list_head_3, struct list_head *build_list_head_4, const int32_t level)
執行每輪遊戲(換一次總督為一輪)

- void print_table(struct list_head *player_list_head_1, struct list_head *player_list_head_2, struct list_head *player_list_head_3, struct list_head *player_list_head_4,struct list_head *build_list_head_1, struct list_head *build_list_head_2, struct list_head *build_list_head_3, struct list_head *build_list_head_4, const int32_t player)
  印出牌桌
- int32_t print_handcard(struct list_head *player_list_head, const int32_t player)
  印出手牌並回傳手牌數量
- int32_t print_build(struct list_head *build_list_head)
  印出牌桌上建築並回傳建築數量
- uint32_t choose_role(uint32_t *choseptr, uint32_t player)
  選擇角色並回傳角色代表數字
- _splayer_card *choose_card(struct list_head *player_list_head, const int32_t player)
  選擇卡片並回傳被選卡片
- void lost_card(struct list_head *player_list_head, struct list_head *choosecard, const int32_t player)
  丟棄卡片
- void lost_commod(struct list_head *build_list_head, const int32_t player)
  丟棄貨物
- void Chapel(struct list_head *player_list_head, struct list_head *build_list_head, const int32_t player)
  執行禮拜堂功能
- void Tower(struct list_head *player_list_head, struct list_head *build_list_head, const int32_t player)
  執行塔樓並檢查手牌數是否超過 7 或 12 張

- 建築師功能與相關特殊卡: 功能請見遊戲規則
  1. void Builder_func(const int32_t sp, const int32_t player, struct list_head *player_list_head, struct list_head *build_list_head)
     主要函式: 檢視是否蓋有特殊建築、符合行動資格
  2. void *Builder_func2(void *arg)
     限制操作時間，執行 thread 所用
  3. int32_t normal_build(int32_t cardfee, _splayer_card *choosecard, struct

list_head *player_list_head, struct list_head *build_list_head, const int32_t player, int32_t vicpoint)
執行行動

4. void Smithy(int32_t *feeptr, const int32_t player)
5. void Poor_house(struct list_head *player_list_head, const int32_t player)
6. void Black_market(struct list_head *build_list_head, int32_t *feeptr, int32_t commodnum, const int32_t player)
7. int32_t Crane(int32_t cardfee, int32_t *feeptr, struct list_head *build_list_head, const int32_t player)
8. void Carpenter(struct list_head *player_list_head, const int32_t player)
9. void Quarry(int32_t *feeptr, const int32_t player)
10. void build_Library(int32_t *feeptr, const int32_t player)

■ 生產者功能與相關特殊卡: 功能請見遊戲規則

1. void Producer_func(const int32_t sp, const int32_t player, struct list_head *player_list_head, struct list_head *build_list_head, int32_t facnum)
   主要函式: 檢視是否蓋有特殊建築、符合行動資格
2. void * Producer_func2(void *arg)
   限制操作時間，執行 thread 所用
3. int32_t normal_produce(_splayer_card *choosecard, struct list_head *build_list_head,const int32_t player)
   執行行動
4. void Well(struct list_head *player_list_head, const int32_t player)
5. void Aqueduct(int32_t *comptr, const int32_t player)
6. void produce_Library(int32_t *comptr, int32_t facnum, const int32_t player)

■ 商人功能與相關特殊卡: 功能請見遊戲規則

1. void Trader_func(const int32_t sp, const int32_t player, const int32_t tradecardnum, struct list_head *player_list_head, struct list_head *build_list_head, int32_t commodnum)
   主要函式: 檢視是否蓋有特殊建築、符合行動資格
2. void *Trader_func2(void *arg)
   限制操作時間，執行 thread 所用
3. int32_t normal_trade(_splayer_card *choosecard, struct list_head *player_list_head,const int32_t player, const int32_t tradecardnum)
   執行行動
4. void Market_stand(struct list_head *player_list_head, const int32_t player)

5. void Market_hall(struct list_head *player_list_head, const int32_t player)
6. void Trading_post(int32_t *soldptr, const int32_t player)
7. void trade_Library(int32_t *soldptr, int32_t commodnum, const int32_t player)

- 市長功能與相關特殊卡: 功能請見遊戲規則
  1. void Councilor_func(const int32_t sp, const int32_t player, struct list_head *player_list_head, struct list_head *build_list_head)
     主要函式: 檢視是否蓋有特殊建築、符合行動資格
  2. void *Councilor_func2(void *arg)
     限制操作時間，執行 thread 所用
  3. int32_t normal_council(const int32_t chosenum, const int32_t drawnum, struct list_head *player_list_head, const int32_t player, const int32_t ar)
     執行行動
  4. int32_t Archive(const int32_t player)
  5. void Prefecture(int32_t *choseptr, const int32_t player)
  6. void council_Library(int32_t *drawptr, const int32_t player)
- 淘金者功能與相關特殊卡: 功能請見遊戲規則
  1. void Prospector_func(const int32_t player, struct list_head *player_list_head, struct list_head *build_list_head)
     主要函式: 檢視是否蓋有特殊建築、符合行動資格
  2. void *Prospector_func2(void *arg)
     限制操作時間，執行 thread 所用
  3. int32_t normal_prospect(int32_t drawnum, struct list_head *player_list_head)
     執行行動
  4. void Gold_mine(struct list_head *player_list_head, const int32_t player)
  5. void prospect_Library(int32_t *drawptr, struct list_head *player_list_head, const int32_t player)

- int32_t end_game(struct list_head *player_list_head_1, struct list_head *player_list_head_2, struct list_head *player_list_head_3, struct list_head *player_list_head_4)
  判斷是否有玩家建築擁有 12 棟以上，決定是否終止遊戲

- 計算分數功能與相關特殊卡: 功能請見遊戲規則

1. int32_t score_count(struct list_head *build_list_head, const int32_t player)
   主要函式: 檢視是否蓋有特殊建築、進行各分數加總
2. int32_t commodcount(struct list_head *build_list_head)

計算含有貨物數量

3. void Guild_hall(int32_t *sctmp, struct list_head *build_list_head, const int32_t player)
4. void City_hall(int32_t *sctmp, struct list_head *build_list_head, const int32_t player)
5. void Triumhal_arch(int32_t *sctmp, struct list_head *build_list_head, const int32_t player)

- void delAllplayercard(struct list_head *player_list_head)
  遊戲結束後，將各 linked list 清空

- 時間倒數器
  1. void *timer()
  2. void handler()

## Structure Set:

- 玩家持有卡片

```
typedef struct Player_card
{
    int32_t id;
    char name[128];
    int32_t fee;
    int32_t score;
    char tip[5096];
    int32_t commodity;
    int32_t vicpoint;//禮拜堂積點

    struct list_head list;

}__attribute__((packed)) _splayer_card;
```

- 建築設定卡片

```
typedef struct Building
{
    int32_t id;
    char name[128];
  int32_t fee;
```

```
	int32_t score;
	int32_t num;
	char tip[5096];

}__attribute__((packed)) _sbuild;
```

- 價物卡

```
typedef struct _cost_card
{
	int32_t value[5];
}cost_card;
```

- 職業設定(enum)

```
enum role_card
{
	Builder,
	Producer,
	Trader,
  Councilor,
	Prospector,
};
```

- Threads argument

```
typedef struct lv2{
	int32_t sp;
	int32_t player;
	struct list_head *player_list_head;
	struct list_head *build_list_head;
	int32_t facnum;
	int32_t commodnum;
	int32_t tradercardnum;
}lv2;
```