Final Project

40947071S 許証淵

**包含檔案:**

all.h

document.pdf

document.docx

game.c

game.o

main.c

Makefile

final Project.ppt

**Structure 結構:**

Each card has a structure.

**一張含有 文字 的圖片

自動產生的描述**

Also, each “player” has a structure

**一張含有 文字 的圖片

自動產生的描述**

-1-

**Functions (Before the Game Starts):**

一張含有 文字 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 室內, 螢幕 的圖片

自動產生的描述

**Functions (During the Game):**

void round\_start(sPlayer\* player,int32\_t numberOfPlayers,int32\_t governor);

//遊戲開始；透過governor找出是誰先需要有動作

void showCardsOnTable(sPlayer\* player,int32\_t numberOfPlayers);

//透過參數『玩家數量』把桌上的牌印出來

void discard(sPlayer\* player, int32\_t numberOfPlayer, sCard\* Card);

//透過 card(實際上是一個節點) 來丟牌

void cardOnHands(sPlayer\* player, int32\_t numberOfPlayer);

//印出手牌

void cardOnHands\_part(sPlayer\* player, int32\_t numberOfPlayer, sCard\* startCard); //印出部分手牌

void displayCardDescription(sPlayer\* player, int32\_t amountOfPlayers, int32\_t numberOfPlayer);

//印出手中卡牌敘述

void display\_TableCardDescription(sPlayer\* player, int32\_t amountOfPlayers, int32\_t numberOfPlayer);

//印出桌上卡牌敘述

-2-

sCard\* find\_handcard(sPlayer\* player, int32\_t numberOfPlayer, int32\_t card\_id); //透過id 找出特定卡牌

bool discardFromHand(sPlayer \*player, int32\_t amountOfPlayers, int32\_t numberOfPlayer, int32\_t amountOfDiscardCard);

//透過最後一項參數找出要丟的數量，並搭配節點找出其ID 與discard function 搭配

int32\_t chooseProfession(sPlayer\* player,int32\_t numberOfPlayers,int32\_t professionOfPlayer);

//選擇職業，透過節點把選到的職業代回去

void chapel\_MakeDecision(sPlayer\* player, int32\_t amountOfPlayer, int32\_t numberOfPlayer, int32\_t tableCardIndex);

//如果有教堂時，發動教堂功能，把卡排放到教堂下

bool office\_MakeDecision(sPlayer\* player, int32\_t amountOfPlayer, int32\_t numberOfPlayer, int32\_t tableCardIndex);

//如果有辦公樓，則發動辦公樓的功能

void produce\_product(sPlayer\* player, int32\_t numberOfPlayer, int32\_t tableCardIndex);

//丟掉貨物，透過節點與桌上卡牌的index於功能結束時紀錄新的狀態。

bool game\_end(sPlayer\* player, int32\_t amountOfPlayers);

//判別main function裡的無窮迴圈是否需要結束遊戲

**Function(職業設定):**

void pro\_Builder(sPlayer\* player,int32\_t amountOfPlayer,int32\_t professionOfPlayer,int32\_t numberOfPlayers);

void build(sPlayer\* player, int32\_t amountOfPlayer,int32\_t  professionOfPlayer,int32\_t numberOfPlayer);

//pro\_的意思是職業，當pro\_被呼叫時，表示啟動該職業並根據玩家選擇決定是否在此function呼叫該職業之特權。

//建築師的選擇與特權

void pro\_Councillor(sPlayer\* player,int32\_t amountOfPlayer,int32\_t professionOfPlayer,int32\_t numberOfPlayers);

void councillor(sPlayer\* player, int32\_t amountOfPlayer, int32\_t professionOfPlayer, int32\_t numberOfPlayer);

//議員的選擇與特權。 原理同上。

**-3-**

void pro\_Producer(sPlayer\* player,int32\_t amountOfPlayer,int32\_t professionOfPlayer,int32\_t numberOfPlayers);

//製造商的選擇與特權。

void pro\_Prospector(sPlayer\* player,int32\_t amountOfPlayer,int32\_t professionOfPlayer,int32\_t numberOfPlayers);

//礦工的選擇與特權。

void pro\_Trader(sPlayer\* player,int32\_t amountOfPlayer,int32\_t professionOfPlayer,int32\_t numberOfPlayers);

//貿易商的選擇與特權。

**Function(機器人與雜項):**

bool bot\_decision(int32\_t chance);

//透過chance比大小回傳要不要出牌

int32\_t findCard(sPlayer\* player, int32\_t numberOfPlayer, int32\_t cardID);

//透過ID找出是否存在在手牌或桌上

**Function(卡牌效果):**

所有的function都會先判斷是否是玩家，若非玩家則有機器人模式，隨機決定是否出牌。

void office(sPlayer\* player, int32\_t amountOfPlayers, int32\_t numberOfPlayer, int32\_t tableCardIndex);

void bank(sPlayer\* player, int32\_t amountOfPlayers, int32\_t numberOfPlayer, int32\_t tableCardIndex);

void chapel(sPlayer\* player, int32\_t amountOfPlayers, int32\_t numberOfPlayer, int32\_t tableCardIndex);

//處理將牌放到教堂底下的行動，還有紀錄分數

void carpenter(sPlayer\* player, int32\_t amountOfPlayers, int32\_t numberOfPlayer, sCard\* card);

//判別是否發動木工坊的條件，如果符合就多抽一張牌

void smithy(sPlayer\* player, int32\_t numberOfPlayer,sCard\* card, int32\_t\* fee);

void quarry(sPlayer\* player, int32\_t numberOfPlayer,sCard\* card, int32\_t\* fee);

void black\_market(sPlayer\* player, int32\_t amountOfPlayers, int32\_t numberOfPlayer,int32\_t\* fee);

void poor\_house(sPlayer\* player,int32\_t amountOfPlayer, int32\_t numberOfPlayer); //判別是否發動濟貧院的條件，如果符合就多抽一張牌

void library(sPlayer\* player, int32\_t numberOfPlayer, int32\_t\* fee, int32\_t profession);

int32\_t crane(sPlayer\* player, int32\_t amountOfPlayers, int32\_t numberOfPlayer, sCard\* card);

void archive(int32\_t numberOfPlayer, int32\_t\* original\_NumberOfHandcard);

void prefecture(int32\_t numberOfPlayer, int32\_t amountOfCard, int32\_t\* amountOfDiscard);

void customs\_office(sPlayer\* player ,int32\_t amountOfPlayer, int32\_t numberOfPlayer, int32\_t tablecardIdx);

void aquaduct(int32\_t numberOfPlayer,int32\_t\* amountProduct);

void well(sPlayer\* player,int32\_t amountOfPlayers,int32\_t numberOfPlayer,int32\_t amountProduct);

//判別是否符合發動條件，如果符合就多抽一張牌

void goldsmith(sPlayer\* player, int32\_t amountOfPlayers, int32\_t numberOfPlayer);

void goldmine(sPlayer\* player, int32\_t amountOfPlayers, int32\_t numberOfPlayer);// //判別是否符合發動條件，如果符合就發動並處理玩家手牌增減

void trading\_post(int32\_t numberOfPlayer, int32\_t\* mostProduct);

void market\_stand(sPlayer\* player, int32\_t amountOfPlayers, int32\_t numberOfPlayer, int32\_t amountProduct);

//判別是否符合發動條件，如果符合就多抽一張牌

void            market\_hall(sPlayer\* player, int32\_t amountOfPlayers, int32\_t numberOfPlayer, int32\_t amountProduct);

//判別是否符合發動條件，如果符合就多抽一張牌

void            harbor(sPlayer\* player, int32\_t amountOfPlayers, int32\_t numberOfPlayer, int32\_t tableCardIndex);

void            tavern(sPlayer\* player, int32\_t amountOfPlayers, int32\_t numberOfPlayer);

void            cottage(sPlayer\* player, int32\_t amountOfPlayers, int32\_t numberOfPlayer);

void            guild\_hall(sPlayer\* player, int32\_t numberOfPlayer);

void            city\_hall(sPlayer\* player, int32\_t numberOfPlayer);

void            triumphal\_arch(sPlayer\* player, int32\_t numberOfPlayer);

void            palace(sPlayer\* player, int32\_t numberOfPlayer);

void            residence(sPlayer\* player, int32\_t numberOfPlayer);

//上面五個為處理總分結算是否有額外加分的function