目標

1.使用SDL做成視窗遊戲

2.回合、賭注、財產

3.使用github開發

需解決問題

**長期:**

1.Visual studio 裝上SDL函式庫以及其擴充

2.紀錄github commit 紀錄以及分枝圖

3.繪製&設計遊戲視窗

4.回合數，加入Player中

5.Deck重新洗牌 ,設定牌之數量少到一定程度需重新洗牌，將Deck資料放入Player中，表示該Player正在用該Deck

**短期:**

1. Constructor 分成player以及deck, 將這兩者的enum 在main設成global

2. 處理Player內部Cards card\_holder問題

3.Player使用Card 須注意constructor的holder\_type

4.需注意Card 的exist成員

5.card\_num

Git branch

Project\_Card\_Game

\*\*注意事項

1.須將日期寫入commit資料中

程式架構

card.h

class Cards{

private:

int cards\_num;

struct Card{

enum suit{clubs,diamonds,hearts,spades}

int card\_face;

suit card\_suit;

bool exist; //在Cards中為1

bool compare\_card(const Card\* const that);

//先比數字在比花色,this比that大傳回1

char card\_face\_TurnToChar();

void print\_card(); //用來輔助 Cards::print\_cards()

}

Card cards[cards\_num];

Cards(card\_holder holder\_type);

//constructor, 用holder\_type判斷為玩家或是Deck

~Cards(); //destructor

public:

void creat\_deck(); //僅用於創造整份排組

void set\_player\_cards(Cards\* Deck); //僅用於創造player手排

void print\_cards();//每輸出5個換一行

int get\_cards\_num();

friend Player;

}

player.h

class Player{

private:

char\* player\_name;

int money\_left;

int round; //回合數

struct hand\_type{

enum type{ straight\_flush=9, four\_of\_a\_kind=8, full\_house=7, flush=6, straight=5; three\_of\_a\_kind=4, two\_pair=3; one\_pair=2, high\_card=1;}

type Type;

card dominant\_card; //Player為Cards的friend,可用card

}

Hand\_type Handtype;

Cards player\_cards;

Player(char\* player\_name,int money\_left, Cards\* Deck); //constructor

~Player(); //destructor

public:

//void set\_player\_cards(); //發牌給player

//發牌函數改到Cards

void judge\_hand\_type(); //初始化Handtype

void sort\_cards(); //先比數字在比花色，使用card\_compare

bool compare(const Player\* const that); //this贏that傳回1

void show\_hand\_type();

//void print\_cards(); //移到Cards

char\* get\_name() const;

int get\_round();

int get\_money\_left();

}

main.cpp

enum cards\_holder{Player,Deck}

Main(){

/\*

1.先創建排組

2.創建腳色

windows\_display.h