計算機程式設計 HW7report

103061223 李俊穎

1. Introduction of the problem and your solution

The homework 7 is a card game called Royal Casino. We need to take in cards with more total scores. During the game, we have to design each player try to play cards from the hand to match cards in the layout. However, this week we just have to complete the shuffle parts and deal parts in this game, and we should make two versions: the code doesn't use struct function and the code uses struct function.

Because this time the instructor has given us most of codes, so we just need to complete the functions and call them correctly. At shuffle stage I use the swap and random to randomize the sequence of the cards, and at deal stage I call the Deal function with for loop to solve this problem.

2. Implementation details ,Additional features of your program (data structure, flows and algorithms)

Because we were required to use the specific function to complete the shuffle and deal. In order to store the correct cards into the arrays, I set the three arrays in Suit data type, and Rank data type do,too. The following is the part of my code:

```
void Shuffle(Suit s[], Rank r[])
{
srand(time(NULL));

for(int i=0;i<=1000;i++){
    int random1=rand()%52;
    int random2=rand()%52;
    Suit temp=s[random1];
    s[random1]=s[random2];
    s[random2]=temp;
    Rank change=r[random1];
    r[random1]=r[random2];
    r[random2]=change;
}
}
```

The above is the shuffle function's code. Because we need to randomize them, I use the random and pass them in the array. After that, I can randomly choose two arrays to exchange their value.

Besides, I use the for loop to do this thing 1000 times to make it more random.

```
void Deal(Suit& s1, Rank& r1, Suit s[], Rank r[], int& n){
//Hand out the next card on top
n--;
s1=s[n];
r1=r[n];
}
```

How the deal function run is that we initialize the cards number is 52. When the number is passed into the function, it will be minus 1 and pass to call the arrays. 52,51,50……, so we can deal the cards from the top of the deck in the sequence.

```
Suit player[4],dealer[4],table[4];
Rank player1[4],dealer1[4],table1[4];
for(int i=0;i<2;i++)
{
        Deal(player[i],player1[i],my_suit,my_rank,num);
}
for(int i=0;i<2;i++)
{
        Deal(table[i],table1[i],my_suit,my_rank,num);
}
for(int i=0;i<2;i++)
{
        Deal(dealer[i],dealer1[i],my_suit,my_rank,num);
 }
```

The next procedure is store the cards value to the players at array. This isn't the big problem because it just need to set three for loops, and then this problem is solved. The previous Deal function is written, so what we need to do is pass the value to the function.

The struct version is similar to nostruct, but the problem of struct is that the variables in it are not easy to understand how the value store in it and how to call the value and function. After I discuss with classmates, I finally realize how to use them. The following is my code:

```
Card player[4],dealer[4],table[4];
```

```
for(int i=0;i<2;i++)
{
    player[i]=Deal(dl) ;

}
for(int i=0;i<2;i++)
{
    table[i]=Deal(dl) ;
}
for(int i=0;i<2;i++)
{
   dealer[i]=Deal(dl) ;
}
```

The concept of the shuffle function is the same as nostruct version, so I don't show them in this report. The Deal function in struct version will run automatically and return the value in the Card struct arrays and store in it's struct variable. Then the player, dealer and layout can get the cards from the top of the deck.