POKER GAME

TOAN DO

CSC5

SUMMER 2016

45277

Introduction

Poker is a very common card game, especially in western countries. This game not only depends on players' luck, but also requires many skills and solid mind from them. The rule of poker is quite simple and many players can play together.

Rule of Poker

Each player will be delivered two game cards from a set of 52 cards. After all players have their cards, they will decide how many they want to bet, call or just check. They also could fold as well. Then, three game cards from set left will be issued. Players again decide which step they move next (bet,call, fold or check). Two remained cards will be sent on the board respectively. Players combines all 5 cards on the board and their own two cards to win the game. There are levels of strength players coud have in order:

- One pair
- Two pairs
- Three of a Kind
- Straight
- Flush
- Full House
- Four of a Kind
- Royal Flush

Due to limitation of time and skill, my program only provide a room for two players. Couple function, such as fold and check sometimes do not work logically as expected. I hope next version will be own bigger room as well as works more efficiently.

The Logic Of Program

Pseudocode

Random 2 cards for two players (rand()%52)

```
Player1Card1=Card[rand()%52];
Player2Card1=Card[rand()%52];
Player1Card2=Card[rand()%52];
Player2Card2=Card[rand()%52];
```

Write each two cards into each player's file for confidential

```
myfile1.open("Player1.txt");
myfile1<<Player[0][0]<<" and "<< Player[0][1];
myfile2.open("Player2.txt");
myfile2<<Player[1][0]<<" and "<< Player[1][1];
```

Use Start() function to write the name of the cards

```
Player[0][0]=Start(Player1Card1);
Player[0][1]=Start(Player1Card2);
Player[1][0]=Start(Player2Card1);
Player[1][1]=Start(Player2Card2);
```

Continue random three more cards from set of cards left

```
CardBoard[0]=Card[rand()%52];
CardBoard[1]=Card[rand()%52];
CardBoard[2]=Card[rand()%52];
```

Display in screen options player could have (bet,check,call,fold)

After players decides at all, random one more cards from the set of cards and display on the screen

```
   CardBoard[3]=Card[rand()%52+1];
}while(CardBoard[3]==CardBoard[0]||CardBoard[3]==CardBoard[1]||
CardBoard[3]==CardBoard[2]);
cout<<"Next card: "<<endl;
Start(CardBoard[3]);
```

Ask players about their move, after that generate the last card

Use function() of levels of strength to decide who is the winner

Libraries

***Iostream**:* input output data

***cstdlib** :* Randoom seeds

***cmath**:* math operator

***string**:* string types, character straits

fstream: write to file

***iomanip**:* output formatted

Number of user defined function: 19

Number of variable**:** 32

Line of code: 630

References

1. Dr Lehr's Lecture and Lab
2. “Starting out with C++ : From Control Structures Through Objects” Gaddis, Tony. 8th Edition. (Textbook)
3. www.cplusplus.com

Program

```
#include <cstdlib>//random seed
#include<iostream>//input output library
#include<cmath>//math library
#include<string>//string library
#include<fstream>// read write file library
#include<iomanip>// output format

using namespace std;

/*
 *
 */
//Function Prototypes
string Start(int i);//output name of cards
int Difference(int a,int b,int c);//difference function
void Sort(int a[],int );// sort set of cards

int Pair(int [],int );//determine player has only one pair
int twoPair(int [],int );//determine player has only two pairs
int ThreeOfAKind(int [],int );//determine player three of a kind
int Straight(int [],int);//determine player has straight
int Flush(int [],int );//determine player has flush
int FullHouse(int [],int );//determine player has full house
int FourOfAKind(int [],int);//determine player has four of a kind
int RoyalFlush(int [],int);//determine player has royal flush
void Player1bet( int &,int &);//store value of player1's bets
void Player1call(int & ,int &);//store value of player1's call
void Player2bet( int &,int &);//store value of player2's bets
void Player2call(int &,int ,int &);//store value of player1's call
int fold();
void ask1(int & ,int & ,int &);//display which move player 1 makes
void ask2(int & ,int & ,int &  );//display which move player 2 makes
```

```
void check();//check


int main(int argc, char** argv) {
    int
Card[52]={11,21,31,41,51,61,71,81,91,101,111,121,131,12,22,32,42,52,62,72,82,92,102
,112,122,132,

13,23,33,43,53,63,73,83,93,103,113,123,133,14,24,34,44,54,64,74,84,94,104,114,124,1
34};

    int Player1Card1,Player1Card2,Player2Card1,Player2Card2;
    int sizeBoard=5;
    int CardBoard[sizeBoard];
    int size=7;
    int Player1[size],Player2[size];
    int money1=50000;
    int money2=50000;
    int betmoney=0,callmoney=0;
    char choice;
    int turn;
    int potsize;
    int numPlayer=2;
    int numCard=7;
    int flag1=0,flag2=0;

    string Player[numPlayer][numPlayer];

    ofstream myfile1,myfile2;
     srand(time(NULL));
    for(int i=1;i<=2;i++)
    {
       do
       {

          Player1Card1=Card[rand()%52];
          Player2Card1=Card[rand()%52];
          Player1Card2=Card[rand()%52];
          Player2Card2=Card[rand()%52];

       }while( Player1Card1== Player1Card2|| Player1Card1== Player2Card1||
Player1Card1== Player2Card2|| Player1Card2== Player2Card1|| Player1Card2==
```

```
Player2Card2|| Player2Card1== Player2Card2);

    }
     Player[0][0]=Start(Player1Card1);
     Player[0][1]=Start(Player1Card2);
     Player[1][0]=Start(Player2Card1);
     Player[1][1]=Start(Player2Card2);
     myfile1.open("Player1.txt");
     myfile1<<Player[0][0]<<" and "<< Player[0][1];
     myfile1.close();
     myfile2.open("Player2.txt");
     myfile2<<Player[1][0]<<" and "<< Player[1][1];
     myfile2.close();
     cout<<endl;

    for(int i=0;i<52;i++)
    {
       if(Card[i]==Player1Card1||Card[i]==Player1Card2||Card[i]==Player2Card1||
Card[i]==Player2Card2)
           Card[i]=0;
    }

    do
    {

    CardBoard[0]=Card[rand()%52];
    CardBoard[1]=Card[rand()%52];
    CardBoard[2]=Card[rand()%52];
    }while (Difference(CardBoard[0],CardBoard[1],CardBoard[2])!=1);
    cout<<" The Flush is : "<<endl;
    Start(CardBoard[0]);
    cout<<", ";
    Start(CardBoard[1]);
    cout<<" and ";
    Start(CardBoard[2]);
    turn=rand()%2;
    cout<<endl;
    if(turn==0||turn==1)
    {

       ask1(betmoney,callmoney,money1);
       ask2(betmoney,callmoney,money2);
```

```
        potsize=100000-money1-money2;
         do
         {
            CardBoard[3]=Card[rand()%52+1];
         }while(CardBoard[3]==CardBoard[0]||CardBoard[3]==CardBoard[1]||
CardBoard[3]==CardBoard[2]);
         cout<<"Next card: "<<endl;
         Start(CardBoard[3]);
         cout<<endl;
         cout<<"Pot size: "<<potsize<<endl;
         ask1(betmoney,callmoney,money1);
         ask2(betmoney,callmoney,money2);
         potsize=100000-money1-money2;
         cout<<"Pot size: "<<potsize<<endl;
         do
         {
            CardBoard[4]=Card[rand()%52+1];
         }while(CardBoard[4]==CardBoard[0]||CardBoard[4]==CardBoard[1]||
CardBoard[4]==CardBoard[2]||CardBoard[4]==CardBoard[3]);
         cout<<"Next card: "<<endl;
         Start(CardBoard[4]);
         cout<<endl;
         ask1(betmoney,callmoney,money1);
         ask2(betmoney,callmoney,money2);
         potsize=100000-money1-money2;
         cout<<"Pot size: "<<potsize<<endl;
        Player1[0]=Player1Card1;
        Player1[1]=Player1Card2;
        for(int i=0;i<size-2;i++)
           Player1[i+2]=CardBoard[i];
        Player2[0]=Player2Card1;
        Player2[1]=Player2Card2;
        for(int i=0;i<size-2;i++)

           Player2[i+2]=CardBoard[i];
        for(int i=0;i<size;i++)

              cout<<Player1[i]<<" ";
        cout<<endl;
        for(int i=0;i<size;i++)

              cout<<Player2[i]<<" ";
```

```
}
cout<<endl;
if(Pair(Player1,7)==1)
   flag1=1;
if(twoPair(Player1,7)==1)
   flag1=2;
if(ThreeOfAKind(Player1,7)==1)
   flag1=3;
if(Straight(Player1,7)==1)
   flag1=4;
if(Flush(Player1,7)==1)
   flag1=5;
if(FullHouse(Player1,7)==1)
   flag1=6;
if(RoyalFlush(Player1,7)==1)
   flag1=7;
 if(Pair(Player2,7)==1)
   flag2=1;
if(twoPair(Player2,7)==1)
   flag2=2;
if(ThreeOfAKind(Player2,7)==1)
   flag2=3;
if(Straight(Player2,7)==1)
   flag2=4;
if(Flush(Player2,7)==1)
   flag2=5;
if(FullHouse(Player2,7)==1)
   flag2=6;
if(RoyalFlush(Player2,7)==1)
   flag2=7;
if(flag1>flag2)
{
   cout<<"Player 1 wins "<<endl;
   money1=money1+potsize;
   money2=money2;
   cout<<" Player1's Budget so far: "<<money1<<endl;
   cout<<" Player2's Budget so far: "<<money2<<endl;

}
 if(flag1<flag2)
{
```

```
        cout<<"Player 2 wins "<<endl;
        money1=money1;
        money2=money2+potsize;
        cout<<" Player1's Budget so far: "<<money1<<endl;
        cout<<" Player2's Budget so far: "<<money2<<endl;

    }
     if(flag1==flag2)
    {
        cout<<"Tied match "<<endl;
        money1=money1+potsize/2;
        money2=money2+potsize/2;

        cout<<" Player1's Budget so far: "<<money1<<endl;
        cout<<" Player2's Budget so far: "<<money2<<endl;

    }
    cout<<"flag1 "<<flag1<<endl;
    cout<<flag2<<endl;

    return 0;
}

string Start(int i)
{   string output;
     switch(i)
    {
        case 11:
            output="Spades Ace";
            cout<<output;
        break;
        case 21:
            output="Spades Two";
            cout<<output;
        break;
        case 31:
            output="Spades Three";
            cout<<output;
        break;
        case 41:
            output="Spades Four";
            cout<<output;
```

```
break;
case 51:
   output="Spades Five";
   cout<<output;
break;
case 61:
   output="Spades Six";
   cout<<output;
break;
case 71:
   output="Spades Seven";
   cout<<output;
break;
case 81:
   output="Spades Eight";
   cout<<output;
break;
case 91:
   output="Spades Nine";
   cout<<output;
break;
case 101:output="Spades Ten";
   cout<<output;
   break;
case 111: output="Spades Jack";
   cout<<output;
break;
case 121: output="Spades Queen";
   cout<<output;
break;
case 131: output="Spades King";
   cout<<output;
break;
case 12:
   output="Club Ace";
   cout<<output;
break;
case 22:  output="Club Two";
   cout<<output;
break;
case 32: output="Club Three";
   cout<<output;
```

```
break;
case 42:  output="Club Four";
   cout<<output;
break;
case 52:  output="Club Five";
   cout<<output;
break;
case 62:  output="Club Six";
   cout<<output;
break;
case 72:  output="Club Seven";
   cout<<output;
break;
case 82:  output="Club Eight";
   cout<<output;
break;
case 92:  output="Club Nine";
   cout<<output;
break;
case 102:  output="Club Ten";
   cout<<output;
break;
case 112:  output="Club Jack";
   cout<<output;
break;
case 122: output="Club Queen";
   cout<<output;
break;
case 132: output="Club King";
   cout<<output;
break;
case 13:
   output="Diamond Ace";
   cout<<output;
break;
case 23: output="Diamond Two";
break;
case 33: output="Diamond Three";
   cout<<output;
break;
case 43: output="Diamond Four";
   cout<<output;
```

```
break;
case 53: output="Diamond Five";
   cout<<output;
break;
case 63:output="Diamond Six";
   cout<<output;
break;
case 73: output="Diamond Seven";
   cout<<output;
break;
case 83: output="Diamond Eight";
   cout<<output;
break;
case 93: output="Diamond Nine";
   cout<<output;;
break;
case 103: output="Diamond Ten";
   cout<<output;;
break;
case 113: output="Diamond Jack";
   cout<<output;
break;
case 123: output="Diamond Queen";
   cout<<output;
break;
case 133: output="Diamond King";
   cout<<output;
break;
case 14:
   output="Heart Ace";
   cout<<output;
break;
case 24: output="Heart Two";
   cout<<output;
break;
case 34: output="Heart Three";
   cout<<output;
break;
case 44: output="Heart Four";
   cout<<output;
break;
case 54: output="Heart Five";
```

```
        cout<<output;
      break;
      case 64: output="Heart Six";
        cout<<output;
      break;
      case 74: output="Heart Seven";
        cout<<output;
      break;
      case 84: output="Heart Eight";
        cout<<output;
      break;
      case 94: output="Heart Nine";
        cout<<output;
      break;
      case 104: output="Heart Ten";
        cout<<output;
      break;
      case 114: output="Heart Jack";
        cout<<output;
      break;
      case 124: output="Heart Queen";
        cout<<output;
      break;
      case 134: output="Heart King";
        cout<<output;
      break;
      return output;

}
}
int Difference(int a,int b,int c)
{
   if(a==b || a==c || b==c || a==0 || b==0 ||c==0)
      return 0;
   else return 1;

}
int Pair(int a[],int value)

{
   int pair=0;
   for(int i=0;i<value-1;i++)
```

```
    {
        for(int j=i+1;j<value;j++)
            if (a[i]/10==a[j]/10)
                pair++;
        }
    if (pair==1)
        return 1;
    else
        return 0;
}
int twoPair(int a[],int value)

{
    int pair=0;
    for(int i=0;i<value-1;i++)
        for(int j=i+1;j<value;j++)
        {
            if (a[i]/10==a[j]/10)
                pair++;
        }
    if (pair==2)
        return 1;
    else
        return 0;
}

int ThreeOfAKind(int a[],int value)
{
    int pair=0;
    for(int i=0;i<value-1;i++)
    {
        for(int j=i+1;j<value;j++)
        {
            if(a[i]/10==a[j]/10)
            pair++;
        }
        if(pair<2)
            pair=0;

    }
    if (pair==2)
        return 1;
```

```
    else return 0;
}
int FourOfAKind(int a[],int value)
{
    int pair=0;
    for(int i=0;i<value-1;i++)
    {
        for(int j=i+1;j<value;j++)
        {
            if(a[i]/10==a[j]/10)
            pair++;
        }
        if(pair<3)
            pair=0;

    }
    if (pair==3)
        return 1;
    else return 0;
}
void Sort(int a[],int value){
    for(int i=0;i<value-1;i++){
        for(int j=i+1;j<value;j++){
            if(a[i]>a[j])
            {
                a[i]=a[i]^a[j];
                a[j]=a[i]^a[j];
                a[i]=a[i]^a[j];
            }
        }
    }
}
int Straight(int a[],int value)
{
    int flag;
    int temp=0;
    for(int i=0;i<value;i++)
        a[i]=a[i]/10;
    Sort(a,value);
    for(int i=0;i<value-1;i++)
    {
        for(int j=i+1;j<i+5;j++)
```

```
        {
            temp=temp+a[j]-a[i];
        }
        if(temp==10)
            flag=1;
        temp=0;

    }
    if(flag==1)
        return 1;
    else
        return 0;

}
int Flush(int a[],int value)
{
    int count=0,flag;
    for(int i=0;i<value;i++)
        a[i]=a[i]%10;
    for(int i=0;i<value-1;i++)
    {
        for(int j=i+1;j<value;j++)
        {
            if(a[i]==a[j])
                count++;

        }
        if(count<5)
            count=0;

    }
    if(count>=5)
        {
            flag=1;

        }
    if(flag==1)
        return 1;
    else
        return 0;
}
```

```
int FullHouse(int a[],int value)
{
   if(ThreeOfAKind(a,value)==1&& Pair(a,value)==1&&FourOfAKind(a,value)==0)
      return 1;
   else
      return 0;

}
int RoyalFlush(int a[],int value)
{
   if(Straight(a,value)==1 && Flush(a,value)==1)
      return 1;
   else return 0;
}
void Player1bet(int &betmoney,int &money1)
{

   cout<<" How many you want to bet :";
   cin>>betmoney;
   money1=money1-betmoney;
   cout<<"Player1's budget: "<<money1;
   cout<<endl;

}
void Player2bet(int &betmoney,int &money2)
{

   cout<<" How many you want to bet :";
   cin>>betmoney;
   money2=money2-betmoney;
   cout<<"Player2's budget: "<<money2;
   cout<<endl;

}
void Player1call(int &betmoney,int &money1)
{
   money1=money1-betmoney;
    cout<<"Player1's budget: "<<money1;
    cout<<endl;
}
void Player2call(int &betmoney,int &money2)
{
```

```
    money2=money2-betmoney;
    cout<<"Player2's budget: "<<money2;
    cout<<endl;
}

void check()
{

}
int fold()
{
    return 1;
}
void ask1(int &betmoney,int &callmoney,int &money1)
{
    char choice;
    cout<<"Player1 's turn: "<<endl;
    cout<<"What is your step, h for check, c for call, b for bet, f for fold :";
    cin>>choice;
        switch(choice){
            case 'h': check();
            break;
            case 'c':Player1call(betmoney,money1);
            break;
            case 'b':Player1bet(betmoney,money1);

            break;
            case 'f':fold();
            break;
        }

}
void ask2(int &betmoney,int &callmoney,int &money2)
{
    char choice;
     cout<<"Player2 's turn: "<<endl;
    cout<<"What is your step, h for check, c for call, b for bet, f for fold :";
    cin>>choice;
        switch(choice){
            case 'h': check();
            break;
            case 'c':Player2call(betmoney,money2);
```

```
            break;
            case 'b':Player2bet(betmoney,money2);

            break;
            case 'f':fold();
            break;
        }

}
```