Branch: master ▾

Find file | Copy path

LeeByoungmo_CIS_17c_47698 / Proj / texasHoldem_17C_Ver1.3 / **Hands.cpp**

libmo No commit message

024972d 1 hour ago

0 contributors

Raw | Blame | History

186 lines (169 sloc) 5.27 KB

```
/*
 * File:   Hands.cpp
 * Author: Byoung Mo Lee
 * Created on May 18, 2019 20:43 PM
 * Purpose: Hands Class for Texas Holdem
 *
 */
#include <iostream>
#include <iomanip>
#include <set>
#include <list>
#include <algorithm>
#include <iterator>
#include <map>
#include <string>
#include "Hands.h"

using namespace std;

Hands::Hands():Player(){
    this->Hands::setPlayerAct();
    this->Hands::setInThePot();
}

string Hands::getHandsName(){
    switch (this->Hands::getHands()) {
            case 0: this->handsName = "Highcard";break;
            case 1: this->handsName = "One Pair";break;
            case 2: this->handsName = "Two Pair";break;
            case 3: this->handsName = "Three of a card";break;
            case 4: this->handsName = "Straight";break;
            case 5: this->handsName = "Flush";break;
            case 6: this->handsName = "Full house";break;
            case 7: this->handsName = "Four of a card";break;
            case 8: this->handsName = "Straight Flush";break;
            default: this->handsName = "Bad Value";
        }
    return this->handsName;
}
int Hands::getHands(){
    this->checkStraight();
    this->Hands::setSuits();
    this->Hands::setFaces();

    if(this->isStraight!=0&&this->isFlush!=0){
        this->isStFl=1;
        this->hands=8;
        set<int,greater<int>> stFl;
        for(auto& it: this->Hands::getMyCards()){
            if(it.getSuit()==this->isFlush) {stFl.insert(it.getFace());}
        }
        if(this->isStraight==15){
            for(int i=14;i>9;i--){
                if(stFl.find(i)==stFl.end()) this->isStFl=0;
            }
        }
        else if(this->isStraight==15){
            for(int i=5;i>1;i--){
                if(stFl.find(i)==stFl.end()) this->isStFl=0;
            }
        }
        else{
            for(int i=this->isStraight;i>this->isStraight-5;i--){
                if(stFl.find(i)==stFl.end()) this->isStFl=0;
            }
        }
        if(this->isStFl==0){
            this->hands=5;
        }
    }
    else if(this->Hands::isPair==7) {this->hands=7;}
    else if(this->isPair==6) {this->hands=6;}
    else if(this->isFlush!=0) {this->hands=5;}
    else if(this->isStraight!=0) {this->hands=4;}
    else if(this->isPair==3) {this->hands=3;}
    else if(this->isPair==2) {this->hands=2;}
    else if(this->isPair==1) {this->hands=1;}
    else {this->hands=0;}
//    cout << endl;
//    for(auto& it: this->Hands::getMyCards()){
//            it.toString();
//    }
//    cout << endl << "Hands=" << this->Hands::getHandsName() << endl;
    return this->hands;
}
void Hands::checkStraight(){
    this->isStraight=0;
    set<int, greater<int>> faceSet;
    int cnt=0;
    for(auto& it: this->Hands::getMyCards()){
        faceSet.insert(it.getFace());
    }
//    cout << endl;

    set<int>::iterator it=faceSet.begin();

    while(it != faceSet.end()&&cnt!=4){
        if(*it-*faceSet.upper_bound(*it)==1) {cnt++;}
        else{cnt=0;}
//        cout << *it << ' ';
        ++it;
    }
    if(cnt==4&&*it+4!=14) {this->isStraight=*it+4;}
    else if(cnt==4&&*it+4==14) {this->isStraight=15;}
    if(*it==14){
        while(it != faceSet.end()&&cnt!=3){
        if(*it-*faceSet.upper_bound(*it)==1) {cnt++;}
        else{cnt=0;}
//        cout << *it << ' ';
        ++it;
        }
    }
    if(cnt==3&&*it+3==5) {this->isStraight=14;}
}


void Hands::setSuits(){
```

```cpp
    this->isFlush=0;
    multiset<int> suitSet;
    for(auto& it: this->Hands::getMyCards()){
        suitSet.insert(it.getSuit());
    }
//    for(int elem:suitSet){
//        cout << elem << ' ';
//    }

    for(int i=0;i<4;i++){
        if(suitSet.count(i)>4) {
            this->suits.insert(pair<int,int>(suitSet.count(i),i));
            this->isFlush=1;
        }
    }
//    cout << endl;
    map<int,int>::iterator pos;
//    for(pos=this->suits.begin();pos!=this->suits.end();++pos){
//        cout << "num: " <<  pos->first << ' '
//             << "suits: " << pos->second << endl;
//    }
//
//    cout << endl;


}
void Hands::setFaces(){
    this->isPair=0;
    multiset<int> faceSet;
    for(auto& it: this->Hands::getMyCards()){
        faceSet.insert(it.getFace());
    }
//    for(int elem:faceSet){
//        cout << elem << ' ';
//    }
    for(int i=2;i<15;i++){
        if(faceSet.count(i)>1) this->faces.insert(pair<int,int>(faceSet.count(i),i));
    }
//    cout << endl;
    map<int,int>::iterator pos;
//    for(pos=this->faces.begin();pos!=this->faces.end();++pos){
//        cout << "pairs: " <<  pos->first << ' '
//             << "faces: " << pos->second << endl;
//    }
//    cout << endl;
//    cout << endl << "faces.empty? " << faces.empty() << endl;
//    cout << endl << "faces.size()? " << faces.size() << endl;

    if(this->faces.empty()) {this->isPair=0;}
    else if(this->faces.size()==1){
        pos=this->faces.begin();
//        cout << "pos->first: " << pos->first << endl;
            if(pos->first==2) this->isPair=1;
            else if(pos->first==3) this->isPair=3;
            else if(pos->first==4) this->isPair=7;

    }
    else if(this->faces.size()>1){
        pos=this->faces.begin();

//        cout << "pos2->first: " << pos->first << endl;
        if(pos->first==2) this->isPair=2;
        else if(pos->first==3) this->isPair=6;
    }


```

```
//    cout << "isPair: " << this->isPair << endl;

}
```