

Branch: master ▾

Find file

Copy path

[LeeByoungmo\\_CIS\\_17c\\_47698](#) / [Proj](#) / [texasHoldem\\_17C\\_Ver1.3](#) / Dealer.cpp libmo No commit message

024972d 1 hour ago

[0 contributors](#)

Raw Blame History



423 lines (354 sloc) 12.4 KB

```
1  /*
2   * File: Dealer.cpp
3   * Author: Byoung Mo Lee
4   * Created on May 17, 2019 00:19 AM
5   * Purpose: Texas Holdem
6   *
7   */
8  #include <string>
9  #include <iostream>
10 #include <cstdlib>
11 #include "Dealer.h"
12
13 using namespace std;
14 Dealer::Dealer(int num){
15     int menu=1;
16     this->setPlayers(num);
17     this->DeckOfCards();
18     do{
19         this->setBlind();
20         this->setIniCont(50);
21         this->resetPot();
22
23         this->shuffle();
24         this->preflop();
25
26         int amount=bettingPrompt1();
27         cout << "bettingPrompt1 is Done" << endl;
28         // for(int i=0;i<this->numPlayers;i++){
29         //     cout << this->players[i].getBalance() << endl;
30         // }
31         // cout << this->potAmount << endl;
32
33
34         int cnt=1;
35         int nP=this->numPlayers;
36         int x=1;
37
38         do{
39
40             amount=this->bettingPrompt2((cnt+this->smallBlind)%nP,amount);
41             if((cnt+this->bigBlind)%nP==this->bigBlind) x=0;
42             if(this->Dealer::getNumAct()==1) {
43                 cout << "find the winner and terminate the round" << endl;
44                 x=0;
45             }
46             cnt++;
47         }while(amount-this->players[(cnt+this->smallBlind)%nP].putInThePot()||x);
48         cout << "bettingPrompt2 is Done" << endl;
49
50         this->flop();
51         cnt=0;
```

```

52     x=1;
53     while((amount->this->players[(cnt+this->bigBlind)%nP].putInThePot())||x){
54         int aP=this->Dealer::getNumAct();
55         amount=this->bettingPrompt3((cnt+this->bigBlind)%nP,amount);
56         cnt++;
57         if((cnt+this->bigBlind)%nP==this->bigBlind) x=0;
58         if(this->Dealer::getNumAct()==1) {
59             cout << "find the winner" << endl;
60             x=0;
61         }
62     }
63 }
64 cout << "bettingPrompt3 is Done" << endl;
65
66 this->turn();
67
68 cnt=0;
69 x=1;
70 while((amount->this->players[(cnt+this->bigBlind)%nP].putInThePot())||x){
71     int aP=this->Dealer::getNumAct();
72     // cout << "aP=" << aP << endl;
73     amount=this->bettingPrompt3((cnt+this->bigBlind)%nP,amount);
74     cnt++;
75     if((cnt+this->bigBlind)%nP==this->bigBlind) x=0;
76     if(this->Dealer::getNumAct()==1) {
77         cout << "find the winner" << endl;
78         x=0;
79     }
80 }
81 cout << "bettingPrompt3 is Done" << endl;
82
83 this->river();
84 cnt=0;
85 x=1;
86 while((amount->this->players[(cnt+this->bigBlind)%nP].putInThePot())||x){
87     int aP=this->Dealer::getNumAct();
88     amount=this->bettingPrompt3((cnt+this->bigBlind)%nP,amount);
89     cnt++;
90     if((cnt+this->bigBlind)%nP==this->bigBlind) x=0;
91     if(this->Dealer::getNumAct()==1) {
92         cout << "find the winner" << endl;
93         x=0;
94     }
95 }
96 }
97 cout << "bettingPrompt3 is Done" << endl;
98
99
100
101
102 for(int i=0;i<this->numPlayers;i++){
103     int order=(i+this->bigBlind)%this->numPlayers;
104     cout << "Player" << order << ": " << this->players[order].getHandsName() << endl;
105
106     // this->players[order].checkStraight();
107     // this->players[order].setFaces();
108     // this->players[order].setSuits();
109     // this->Dealer::displayPlayersInfo(order);
110 }
111 this->calBal();
112 for(int i=0;i<this->numPlayers;i++){
113     int order=(i+this->bigBlind)%this->numPlayers;
114     this->Dealer::displayPlayersInfo(order);
115 }
116
117 menu=this->resetGame();

```

```
118     }while(menu!=0);
119 }
120
121
122 void Dealer::setPlayers(int num){
123     this->numPlayers=num;
124     this->players=new Hands[this->numPlayers];
125
126     for(int i=0;i<this->numPlayers;i++) {
127         this->players[i].setPlayerName("player#" + to_string(i));
128     }
129 }
130
131 void Dealer::setBlind(){
132     this->bigBlind=this->getRound()%(this->getNumPlayers()-1);
133     this->smallBlind=this->bigBlind+1;
134 }
135
136
137 void Dealer::displayPlayersInfo(int num){
138     cout << this->Dealer::players[num].getName() << endl;
139     if(num==this->Dealer::getBigBlind()) {cout << "BB" << endl;}
140     else if (num==this->Dealer::getSmallBlind()) {cout << "SB" << endl;}
141     cout << "Balance: $" <<this->Dealer::players[num].getBalance() << endl << endl;
142     for(auto& it:this->players[num].getMyCards()){
143         it.toString();
144     }
145     cout << endl;
146 }
147
148 void Dealer::DeckOfCards() {
149     for(int i=0;i<52;i++){
150         this->deck[i]=new Card(i);
151         //cards[i]->toString();
152     }
153 }
154
155
156
157 void Dealer::shuffle(){
158     // Initialize seed randomly
159     srand(time(0));
160
161     for (int i=0; i<52 ;i++)
162     {
163         // Random for remaining positions.
164         int r = i + (rand() % (52 -i));
165
166         swap(this->deck[i], this->deck[r]);
167     }
168 }
169
170
171 void Dealer::preflop(){
172     int nCards=2;
173     int order;
174     int nP=this->numPlayers;
175     int bB=this->bigBlind;
176     for(int i=0;i<nCards;i++){
177         for(int j=0;j<nP;j++){
178             order=(j+bB)%nP;
179             this->players[order].addMyCards(*this->deck[j+nP*i]);
180
181             //         this->deck[j+i+nP*i]->toString();
182             //         this->Dealer::displayPlayersInfo(order);
183
```

```

184     }
185 }
186 }
187
188 void Dealer::flop(){
189     int nCards=3;
190     int nP=this->numPlayers;
191     int bB=this->bigBlind;
192     for(int i=nP*2+1;i<nP*2+1+nCards;i++){
193         // this->deck[i]->toString();
194         for(int j=0;j<nP;j++){
195             int order=(j+bB)%nP;
196             this->players[order].addMyCards(*this->deck[i]);
197         }
198     }
199 }
200
201 void Dealer::turn(){
202     int nCards=1;
203     int nP=this->numPlayers;
204     int bB=this->bigBlind;
205     for(int i=nP*2+1+3+1;i<nP*2+1+3+1+nCards;i++){
206         // this->deck[i]->toString();
207         for(int j=0;j<nP;j++){
208             int order=(j+bB)%nP;
209             this->players[order].addMyCards(*this->deck[i]);
210         }
211     }
212 }
213
214 void Dealer::river(){
215     int nCards=1;
216     int nP=this->numPlayers;
217     int bB=this->bigBlind;
218     for(int i=nP*2+1+3+1+1+1;i<nP*2+1+3+1+1+1+nCards;i++){
219         // this->deck[i]->toString();
220         for(int j=0;j<nP;j++){
221             int order=(j+bB)%nP;
222             this->players[order].addMyCards(*this->deck[i]);
223         }
224     }
225 }
226
227 int Dealer::bettingPrompt1(){
228     int input;
229     int raise;
230     int diff;
231     int nP=this->numPlayers;
232     int amount;
233     this->players[bigBlind].bet(this->initialContributeAmount);
234     this->potAmount+=this->initialContributeAmount;
235     cout << "Player" << this->bigBlind << ": $" << this->initialContributeAmount << " into the Pot" << endl;
236     cout << "Pot: $" << this->getPotAmount() << endl;
237     amount=this->players[bigBlind].putInThePot();
238
239     this->players[smallBlind].bet(this->initialContributeAmount/2);
240     this->potAmount+=this->initialContributeAmount/2;
241     diff=amount-this->initialContributeAmount/2;
242
243
244     cout << "Player" << this->smallBlind << ": $" << this->initialContributeAmount/2 << " into the Pot" << endl;
245     cout << "Pot: $" << this->getPotAmount() << endl;
246
247
248     do{
249         cout << "To call you have to put $" << diff << endl;

```

```

250 cout << "Player" << this->smallBlind << ": Call - 1, Raise - 2, Fold -3" << endl;
251 cin >> input;
252 if(input==1) {
253     this->players[smallBlind].bet(diff);
254     this->potAmount+=diff;
255     cout << "Player#" << smallBlind << ": $" << diff << endl;
256     cout << "Pot: $" << this->getPotAmount() << endl;
257
258 }
259 else if(input==2){
260     cout << "Amount: " ;
261     do{
262         cin >> raise;
263         if(raise<=diff) cout << "Amount should be greater than call Amount\n";
264     }while(raise<=diff);
265     this->players[smallBlind].bet(raise+diff);
266     this->potAmount+=(raise+diff);
267     cout << "Player#" << smallBlind << ": $" << raise+diff << " into the Pot" << endl;
268     cout << "Pot: $" << this->getPotAmount() << endl;
269     amount = this->players[smallBlind].putInThePot();
270 }
271 else if(input==3){
272     this->players[smallBlind].setPlayerInact();
273     cout << "Player#" << smallBlind << ": folded" << endl;
274     cout << "Pot: $" << this->getPotAmount() << endl;
275 }
276 else {cout << "wrong input" << endl;}
277 }while(input<1||input>3);
278
279 return amount;
280 }
281
282 int Dealer::bettingPrompt2(int num, int amount){
283     int input;
284     do{
285         if(this->players[num].getActStatus()){
286             int nP=this->numPlayers;
287
288             int raise;
289             int diff=amount - this->players[num].putInThePot();
290             cout << "To call you have to put $" << diff << endl;
291             cout << "Player" << num << ": call - 1, Raise - 2, Fold -3" << endl;
292             cin >> input;
293             if(input==1) {
294                 this->players[num].bet(diff);
295                 this->potAmount+=(diff);
296                 cout << "Player#" << num << ": $" << diff << endl;
297                 cout << "Pot: $" << this->getPotAmount() << endl;
298             }
299             else if(input==2){
300                 cout << "Amount: " ;
301                 do{
302                     cin >> raise;
303                     if(raise<=diff) cout << "Amount should be greater than call Amount\n";
304                 }while(raise<=diff);
305
306                 this->players[num].bet(raise+diff);
307                 this->potAmount+=(raise+diff);
308                 cout << "Player#" << num << ": $" << raise+diff << endl;
309                 cout << "Pot: $" << this->getPotAmount() << endl;
310                 amount=this->players[num].putInThePot();
311             }
312             else if(input==3){
313                 this->players[num].setPlayerInact();
314                 cout << "Player#" << num << ": folded" << endl;
315                 cout << "Pot: $" << this->getPotAmount() << endl;

```

```

316         }
317         else {cout << "wrong input" << endl;}
318     }
319     }while(input<1||input>3);
320
321     return amount;
322 }
323
324 int Dealer::bettingPrompt3(int num, int amount){
325     int nP=this->numPlayers;
326     int input;
327     do{
328         if(this->players[num].getActStatus()){
329             int raise;
330             int diff=amount - this->players[num].putInThePot();
331             if(!diff){
332                 cout << "You can check" << endl;
333                 cout << "Player" << num << ": check - 1, Raise - 2, Fold -3" << endl;
334
335             }
336             else{
337                 cout << "To call you have to put $" << diff << endl;
338                 cout << "Player" << num << ": call - 1, Raise - 2, Fold -3" << endl;
339
340             }
341
342             cin >> input;
343             if(input==1) {
344
345                 this->players[num].bet(diff);
346                 this->potAmount+=(diff);
347                 cout << "Player#" << num << ": $" << diff << endl;
348                 cout << "Pot: $" << this->getPotAmount() << endl;
349             }
350             else if(input==2){
351                 cout << "Amount: " ;
352                 do{
353                     cin >> raise;
354                     if(raise<=diff) cout << "Amount should be greater than call Amount\n";
355                 }while(raise<=diff);
356
357                 this->players[num].bet(raise+diff);
358                 this->potAmount+=(raise+diff);
359                 cout << "Player#" << num << ": $" << raise+diff << endl;
360                 cout << "Pot: $" << this->getPotAmount() << endl;
361                 amount=this->players[num].putInThePot();
362             }
363             else if(input==3){
364                 this->players[num].setPlayerInact();
365                 cout << "Player#" << num << ": folded" << endl;
366                 cout << "Pot: $" << this->getPotAmount() << endl;
367             }
368             else {cout << "wrong input" << endl;}
369         }
370     }while(input<1||input>3);
371
372     return amount;
373 }
374
375 int Dealer::getNumAct(){
376     int num=0;
377     for(int i=0;i<this->numPlayers;i++){
378         if(this->players[i].getActStatus()) num++;
379     }
380
381     return num;

```

```
382 }
383
384 int Dealer::decideWinner(){
385     int max=0;
386     int winner=0;
387     int nP=this->getNumPlayers();
388
389     for(int i=0;i<nP;i++){
390         if(this->players[i].getActStatus()==1) {
391             if(this->players[i].getHands()>max) {
392                 max=this->players[i].getHands();
393                 winner=i;
394             }
395         }
396     }
397     // cout << "Max=" << max << endl;
398     return winner;
399 }
400 }
401
402 void Dealer::calBal(){
403     int winner=this->decideWinner();
404     cout << "Player" << winner << " won $" << this->getPotAmount() << "!" << endl;
405     this->players[winner].addBal(this->getPotAmount());
406     this->Dealer::resetPot();
407 }
408
409 int Dealer::resetGame(){
410     int input;
411     this->nextRound();
412
413     for(int i=0;i<this->getNumPlayers();i++){
414
415         this->players[i].resetHands();
416     }
417
418
419     cout << "if you want to exit press 0" << endl;
420     cin >> input;
421
422     return input;
423 }
```