

| 일반 소스 공유 | >

(공유) 포커[텍사스 홀덤] v2.0 by seori



seori 챗봇 마스터 🏆 1:1 채팅

2020.02.07. 23:09 조회 59

댓글 31 URL 복사

📁 첨부파일 1

치명적인 오류들을 몇가지 수정하였습니다.

직접 테스트를 계속 시도하여 업데이트한 것이니
이제 안심하고 사용하셔도 될 듯합니다.

```

1 const scriptName="홀덤.js"
2 var cards=["s 14", "s 2", "s 3", "s 4", "s 5", "s 6", "s 7", "s 8", "s 9", "s 10", "s 11", "s 12", "s 13", "d 14", "d 2", "d 3", "d 4", "d 5", "d 6", "d 7", "d 8", "d 9", "d 10", "d 11", "d 12", "d 13", "h 14", "h 2", "h 3", "h 4", "h 5", "h 6", "h 7", "h 8", "h 9", "h 10", "h 11", "h 12", "h 13", "c 14", "c 2", "c 3", "c 4", "c 5", "c 6", "c 7", "c 8", "c 9", "c 10", "c 11", "c 12", "c 13"];
3 //카드들입니다. s: 스페이드, d: 다이아몬드, h: 하트, c: 클로버. 숫자는 A가 등급이 가장 높으므로 14를 1로 나타냅니다.
4 var poker=false; var Gstart=false;
5 var player=[]; var coin=[]; var pot=[]; //선수의 게임 진행 정보
6 var player2=[]; var chips=[]; var pot2=[]; //폴드한 선수의 정보 저장
7 var card1; var card2; var cardlist=[]; var common=[]; //카드관련
8 var basicchips=100; var basicseed=1; //시드관련
9 var isBet=false; var beting=[]; var isCheck=0; var status=0; var money=0; var call=0; var m=0;
10 var pc="";
11 var x="";
12 var R;
13 var isFirstgame=true;
14 var allsee="\u200b".repeat(500);
15 var nagagi=[];
16 var preChat=[];
17 var turnoff=false; var onoff=[]; //종료관련
18 var allin=[]; var side=[]; //올인관련
19
20 function shuffle(a) { //배열을 섞는 함수. 카드 셔플에 이용.
21     var j, x, i;
22     for (i = a.length - 1; i > 0; i--) {
23         j = Math.floor(Math.random() * (i + 1));
24         x = a[i]; a[i] = a[j]; a[j] = x;
25     }
26     return a;
27 }
28
29 function seol(moyang){ //게임이 끝나고 '카드모양'에 등급을 매길 때 이용됩니다.
30     return moyang.replace(/s/g,3).replace(/d/g,2).replace(/h/g,1).replace(/c/g,0);
31 }
32
33 function lee(sutja){ //족보 등급 계산을 위한 '숫자' 등급을 매기는 함수입니다.
34     var isStraight=0; var sut=[]; var ja=[];
35
36     for (var i=0;i<sutja.length;i++){
37         sutja[i]=Number(sutja[i]);
38         sut.push(Number(sutja[i]));
39         ja.push(Number(sutja[i]));
40     }
41     for (var i=0;i<sutja.filter(e => e == 14).length;i++){
42         sutja.push(1); sut.push(1); ja.push(1);
43     } //추후 백스트레이트 계산을 위해 배열에 A(14)가 포함된 경우 1을 추가해줍니다.
44
45     for (var i=14;i>1;i--) if (sutja.filter(e => e == i).length==4) return [9, i]; //포카드
46
47     for (var i=14;i>1;i--){
48         if (sutja.filter(e => e == i).length==3){
49             for (k=0;k<3;k++){
50                 sut.splice(sut.indexOf(i), 1);
51                 ja.splice(ja.indexOf(i), 1);
52             }
53             return [8, i];
54         }
55     }
56     return [7, 14];
57 }
58
59 function isFull(sutja){
60     var full=0;
61     for (var i=0;i<sutja.length;i++){
62         if (Number(sutja[i])>10){
63             full++;
64         }
65     }
66     return full>2;
67 }
68
69 function isFlush(sutja){
70     var flush=0;
71     for (var i=0;i<sutja.length;i++){
72         if (sutja[i].indexOf('s')>0){
73             flush++;
74         }
75     }
76     return flush==5;
77 }
78
79 function isPair(sutja){
80     var pair=0;
81     for (var i=0;i<sutja.length;i++){
82         for (var j=i+1;j<sutja.length;j++){
83             if (Number(sutja[i])==Number(sutja[j])){
84                 pair++;
85             }
86         }
87     }
88     return pair>1;
89 }
90
91 function isThree(sutja){
92     var three=0;
93     for (var i=0;i<sutja.length;i++){
94         for (var j=i+1;j<sutja.length;j++){
95             for (var k=j+1;k<sutja.length;k++){
96                 if (Number(sutja[i])==Number(sutja[j])&&Number(sutja[j])==Number(sutja[k])){
97                     three++;
98                 }
99             }
100         }
101     }
102     return three>1;
103 }
104
105 function isFour(sutja){
106     var four=0;
107     for (var i=0;i<sutja.length;i++){
108         for (var j=i+1;j<sutja.length;j++){
109             for (var k=j+1;k<sutja.length;k++){
110                 for (var l=k+1;l<sutja.length;l++){
111                     if (Number(sutja[i])==Number(sutja[j])&&Number(sutja[j])==Number(sutja[k])&&Number(sutja[k])==Number(sutja[l])){
112                         four++;
113                     }
114                 }
115             }
116         }
117     }
118     return four>1;
119 }
120
121 function isStraightFlush(sutja){
122     var straightFlush=0;
123     for (var i=0;i<sutja.length;i++){
124         for (var j=i+1;j<sutja.length;j++){
125             for (var k=j+1;k<sutja.length;k++){
126                 for (var l=k+1;l<sutja.length;l++){
127                     for (var m=l+1;m<sutja.length;m++){
128                         if (Number(sutja[i])==Number(sutja[j])&&Number(sutja[j])==Number(sutja[k])&&Number(sutja[k])==Number(sutja[l])&&Number(sutja[l])==Number(sutja[m])&&sutja[i].indexOf('s')>0){
129                             straightFlush++;
130                         }
131                     }
132                 }
133             }
134         }
135     }
136     return straightFlush>1;
137 }
138
139 function isRoyal(sutja){
140     var royal=0;
141     for (var i=0;i<sutja.length;i++){
142         for (var j=i+1;j<sutja.length;j++){
143             for (var k=j+1;k<sutja.length;k++){
144                 for (var l=k+1;l<sutja.length;l++){
145                     for (var m=l+1;m<sutja.length;m++){
146                         if (Number(sutja[i])==Number(sutja[j])&&Number(sutja[j])==Number(sutja[k])&&Number(sutja[k])==Number(sutja[l])&&Number(sutja[l])==Number(sutja[m])&&sutja[i].indexOf('s')>0){
147                             royal++;
148                         }
149                     }
150                 }
151             }
152         }
153     }
154     return royal>1;
155 }
156
157 function isFullHouse(sutja){
158     var fullHouse=0;
159     for (var i=0;i<sutja.length;i++){
160         for (var j=i+1;j<sutja.length;j++){
161             for (var k=j+1;k<sutja.length;k++){
162                 for (var l=k+1;l<sutja.length;l++){
163                     for (var m=l+1;m<sutja.length;m++){
164                         if (Number(sutja[i])==Number(sutja[j])&&Number(sutja[j])==Number(sutja[k])&&Number(sutja[k])==Number(sutja[l])&&Number(sutja[l])==Number(sutja[m])&&(Number(sutja[i])>10||Number(sutja[j])>10||Number(sutja[k])>10||Number(sutja[l])>10||Number(sutja[m])>10)){
165                             fullHouse++;
166                         }
167                     }
168                 }
169             }
170         }
171     }
172     return fullHouse>1;
173 }
174
175 function isTwoPair(sutja){
176     var twoPair=0;
177     for (var i=0;i<sutja.length;i++){
178         for (var j=i+1;j<sutja.length;j++){
179             for (var k=j+1;k<sutja.length;k++){
180                 for (var l=k+1;l<sutja.length;l++){
181                     for (var m=l+1;m<sutja.length;m++){
182                         if (Number(sutja[i])==Number(sutja[j])&&Number(sutja[j])==Number(sutja[k])&&Number(sutja[k])==Number(sutja[l])&&Number(sutja[l])==Number(sutja[m])&&(Number(sutja[i])>10||Number(sutja[j])>10||Number(sutja[k])>10||Number(sutja[l])>10||Number(sutja[m])>10)){
183                             twoPair++;
184                         }
185                     }
186                 }
187             }
188         }
189     }
190     return twoPair>1;
191 }
192
193 function isOnePair(sutja){
194     var onePair=0;
195     for (var i=0;i<sutja.length;i++){
196         for (var j=i+1;j<sutja.length;j++){
197             for (var k=j+1;k<sutja.length;k++){
198                 for (var l=k+1;l<sutja.length;l++){
199                     for (var m=l+1;m<sutja.length;m++){
200                         if (Number(sutja[i])==Number(sutja[j])&&Number(sutja[j])==Number(sutja[k])&&Number(sutja[k])==Number(sutja[l])&&Number(sutja[l])==Number(sutja[m])&&(Number(sutja[i])>10||Number(sutja[j])>10||Number(sutja[k])>10||Number(sutja[l])>10||Number(sutja[m])>10)){
201                             onePair++;
202                         }
203                     }
204                 }
205             }
206         }
207     }
208     return onePair>1;
209 }
210
211 function isHighCard(sutja){
212     var highCard=0;
213     for (var i=0;i<sutja.length;i++){
214         for (var j=i+1;j<sutja.length;j++){
215             for (var k=j+1;k<sutja.length;k++){
216                 for (var l=k+1;l<sutja.length;l++){
217                     for (var m=l+1;m<sutja.length;m++){
218                         if (Number(sutja[i])>Number(sutja[j])&&Number(sutja[j])>Number(sutja[k])&&Number(sutja[k])>Number(sutja[l])&&Number(sutja[l])>Number(sutja[m])&&sutja[i].indexOf('s')>0){
219                             highCard++;
220                         }
221                     }
222                 }
223             }
224         }
225     }
226     return highCard>1;
227 }
228
229 function isLowCard(sutja){
230     var lowCard=0;
231     for (var i=0;i<sutja.length;i++){
232         for (var j=i+1;j<sutja.length;j++){
233             for (var k=j+1;k<sutja.length;k++){
234                 for (var l=k+1;l<sutja.length;l++){
235                     for (var m=l+1;m<sutja.length;m++){
236                         if (Number(sutja[i])<Number(sutja[j])&&Number(sutja[j])<Number(sutja[k])&&Number(sutja[k])<Number(sutja[l])&&Number(sutja[l])<Number(sutja[m])&&sutja[i].indexOf('s')>0){
237                             lowCard++;
238                         }
239                     }
240                 }
241             }
242         }
243     }
244     return lowCard>1;
245 }
246
247 function isStraightFlushLow(sutja){
248     var straightFlushLow=0;
249     for (var i=0;i<sutja.length;i++){
250         for (var j=i+1;j<sutja.length;j++){
251             for (var k=j+1;k<sutja.length;k++){
252                 for (var l=k+1;l<sutja.length;l++){
253                     for (var m=l+1;m<sutja.length;m++){
254                         if (Number(sutja[i])<Number(sutja[j])&&Number(sutja[j])<Number(sutja[k])&&Number(sutja[k])<Number(sutja[l])&&Number
```

```

49         for (k=0;k<5;k++) sut.splice(sut.indexOf(1),1);
50         for (var j=14;j>1;j--) if (sut.filter(e => e == j).length>=2) return [8,i+0.01*j];
51     }
52 } //폴하우스
53
54 for (var i=14;i>9;i--){
55     if (sutja.filter(e => e == i).length>=1){
56         isStraight+=1;
57         if (isStraight==5) return [6,0];
58     } else {
59         isStraight=0;
60         break;
61     }
62 } //마운틴
63
64 for (var i=5;i>0;i--){
65     if (sutja.filter(e => e == i).length>=1){
66         isStraight+=1;
67         if (isStraight==5) return [5,0];
68     } else {
69         isStraight=0;
70         break;
71     }
72 } //백스트레이트
73
74 for (var p=13; p>5; p--){
75     for (var i=0;i<5;i++){
76         if (sutja.filter(e => e == (p-i)).length >=1 ){
77             isStraight+=1;
78             if (isStraight==5) return [4,p];
79         } else break;
80     }isStraight=0;
81 } //스트레이트
82
83 for (var i=14;i>1;i--){ if (sutja.filter(e => e == i).length==3) return [3,i]; } //트리플
84
85 for (var i=14;i>1;i--){
86     if (sutja.filter(e => e == i).length==2){
87         for (k=0;k<2;k++) ja.splice(ja.indexOf(i),1);
88         for (var j=i-1;j>1;j--) if (ja.filter(e => e == j).length==2) return [2,i+0.01*j];
89         return [1,i]; //원페어
90     }
91 }
92
93 var ans=[0,Math.max.apply(null, sutja)];
94 return ans; //탑 (노페어)
95 }
96
97 function grade(CARD){ //모양 등급인 '플러시'의 경우를 생각하여 족보의 진짜 등급을 계산합니다.
98     var Flushshape; var shape=[]; var isFlush=false;
99     var num=[]; var card=[]; var Realgrade=[];
100
101     for (var i=0; i<CARD.length; i++){
102         shape.push(seol(CARD[i].split(" ")[0]));
103         num.push(CARD[i].split(" ")[1]);
104         card.push(Number(num[i])*10+Number(shape[i]));
105     }
106
107     for (var i=0; i<4; i++){
108         if (shape.filter(e => e == i).length>=5){
109             isFlush=true;
110             Flushshape=i;
111             break;
112         }
113     } //족보가 플러시인지 아닌지 판별합니다.
114
115     if (isFlush==true){ //플러시일 경우
116         for (var i=0;i<shape.length; i++){
117             if (shape[i]!=Flushshape){

```

```

118         card.splice(1,1);
119         num.splice(i,1);
120     }
121 }
122 if (lee(num)[0]==6) Realgrade[0].push(12); //로티플
123 else if (lee(num)[0]==5) Realgrade[0].push(11); //백티플
124 else if (lee(num)[0]==4) Realgrade[0].push(10); //스티플
125 else Realgrade[0].push(7); //플러시
126 } else Realgrade[0].push(lee(num)[0]); //플러시가 아닐 경우 숫자 등급을 그대로 반환합니다.
127
128 num.sort((a,b)=>b-a);
129 var nnum=0;
130 card.sort((a,b)=>b-a);
131 var sshape=0;
132 for (var i=0;i<num.length;i++){
133     nnum+=Math.pow(0.01,i)*num[i];
134     sshape+=Math.pow(0.001,i)*card[i];
135 }
136 Realgrade[0].push(lee(num)[1]);
137 Realgrade.push(nnum);
138 Realgrade.push(sshape);
139 return Realgrade;
140 /* [[족보 등급, 등급에 해당하는 숫자], 하이 카드, 하이 카드에 따른 모양 등급]으로 전체 등급을 반환
141 등급이 무승부가 날 시 우승자를 판별하기 위함입니다.*/
142 }
143
144 function sh(shs){ //내용 출력을 위한 함수입니다.
145     return shs.replace(/s /g,"♠").replace(/d /g,"♦").replace(/h /g,"♥").replace(/c /g,"♣").
146 }
147
148 function outgrade(put){ //족보 등급 출력을 위한 함수입니다.
149     var first=Number(put[1]).toFixed(0);
150     var second=Math.round((Number(put[1])-first)*100);
151     var third=Number(put[0]).toFixed(0).replace("12", "로티플").replace("11", "백티플").replace("10", "스티플").replace("7", "플러시");
152
153     var ans=first+", "+second+" "+third;
154
155     if (first==0) return third;
156     else if (second==0) return first+" "+third;
157     else return ans;
158 }
159
160 function response(room, msg, sender, isGroupChat, replier, ImageDB, packageName, threadId){
161
162     if (preChat[room]==msg) return 0; //도배(따닥챗)방지
163     preChat[room]=msg;
164
165     var cmd=msg.split(" ")[0];
166     var data=msg.replace(cmd+" ", "");
167     if (msg=="/홀덤") replier.reply("텍사스 홀덤 도움말\n(명령어, 룰 설명)+allsee+\n\n[명령어]\n");
168     if (msg=="/홀덤 생성" && isGroupChat==true){
169         if (poker==true) replier.reply("이미 생성된 방이 있습니다.");
170         else {
171             poker=true;
172             R=room;
173             replier.reply("방이 생성되었습니다.\n저에게 랜톡으로 '/홀덤 참가'라고 보내주시면 참가가능합니다.");
174         }
175     } //방을 만듭니다.
176
177     if (Gstart==false && poker==true && cmd=="/칩" && isNaN(data)==false){
178         if ((data-Math.floor(data))==0){
179             if (50<=data && data<=1000){
180                 basicchips=Number(data);
181                 replier.reply("기본 칩이 "+data+"개로 설정되었습니다.");
182                 coin.fill(basicchips);
183                 chips.fill(basicchips);
184                 if (basicseed/basicchips<0.002){
185                     basicseed=Math.cell(basicchips*0.002);
186                     replier.reply("최저시드("+basicseed+"개)로 자동 설정되었습니다.");
187                 }
188             }
189         }
190     }
191 }

```

```

187         } else if (basicseed/basicchips>0.05){
188             basicseed=Math.floor(basicchips*0.05);
189             replier.reply("최고시드("+basicseed+"개)로 자동 설정되었습니다.");
190         }
191     } else replier.reply("50에서 1000 사이의 자연수를 입력해주세요.");
192 } else replier.reply("자연수를 입력해주세요.");
193 } //게임 시작전에 기본적으로 가지게 되는 칩 개수를 설정합니다.
194
195 if (Gstart == false && poker==true && cmd=="/시드" && isNaN(data)==false){
196     if ((data-Math.floor(data))==0){
197         basicseed=Number(data);
198         if (basicseed/basicchips<0.002){
199             basicseed=Math.ceil(basicchips*0.002);
200             replier.reply("최저시드("+basicseed+"개)로 자동 설정되었습니다.");
201         } else if (basicseed/basicchips>0.05){
202             basicseed=Math.floor(basicchips*0.05);
203             replier.reply("최고시드("+basicseed+"개)로 자동 설정되었습니다.");
204         } else replier.reply("시드가 "+data+"개로 설정되었습니다.");
205     } else replier.reply("자연수를 입력해주세요.");
206 } //시드금액을 설정합니다. 방장이 자동 SB가 됩니다.
207
208 if (msg=="/홀덤 참가"){
209     if (Gstart==false){
210         if (poker==true){
211             if (room==sender){
212                 if (player.length<5){
213                     if (player.indexOf(sender)==-1){
214                         player.push(sender);
215                         player2.push(sender);
216                         coin.push(basicchips);
217                         chips.push(basicchips);
218
219                         pot.push(0);
220                         pot2.push(0);
221                         cardlist.push([]);
222                         replier.reply("정상 참가 처리 되었습니다.");
223                         replier.reply(R, "현재 참가자: "+player);
224                     } else replier.reply(sender + "님은 이미 참가하셨습니다.");
225                 } else replier.reply("최대 인원은 5명입니다.");
226             } else replier.reply("관적으로 보내주셔야 참가가 인정됩니다.");
227         } else replier.reply("아직 방이 만들어지지 않았습니다.");
228     } else replier.reply("이미 게임이 진행중입니다.");
229 } //게임에 참가합니다.
230
231 if (msg=="/홀덤 시작" && room==R && player.indexOf(sender)!=-1){ // 게임을 시작합니다.
232     if (player.length>=2){
233         if (poker==true){
234             if (Gstart==false){
235                 Gstart=true;
236                 if (isFirstgame==false && basicseed<Math.floor(0.05*basicchips)){
237                     replier.reply("시드를 올리시겠습니까?\n/업 (숫자) 또는 /킵 으로 설정 가능");
238                     while (isFirstgame==false){}
239                 } /*두 번째 이상의 게임부터 시드를 올릴 지 말 지 선택에 대한 질문입니다.
240                 이미 최고시드일 경우에는 묻지 않습니다.*/
241                 shuffle(cards); //카드 섞기
242                 replier.reply("게임을 시작합니다.");
243                 for(var i=0; i<=player.length-1; i++){
244                     card1=cards[cards.length-1];
245                     cards.pop();
246                     card2=cards[cards.length-1];
247                     cards.pop();
248                     cardlist[i].push(card1);
249                     cardlist[i].push(card2);
250                     var ans="핸드 카드를 받았습니다.\n"+cardlist[i]+"현재 등급: "+outgrade;
251                     replier.reply(player[i],sh(ans));
252                 } //처음에 받게 되는 핸드 카드를 받습니다.
253                 java.lang.Thread.sleep(1000);
254                 replier.reply("처음 기본 칩은 "+basicchips+"개입니다. "+basicseed+"/"+basics
255                 coin[0]-=basicseed;

```

```

256         coin[1]-=basicseed*2;
257         chips[0]-=basicseed;
258         chips[1]-=basicseed*2;
259         money=basicseed*3;
260         pot[0]=basicseed;
261         pot2[0]=basicseed;
262         pot[1]=basicseed*2;
263         pot2[1]=basicseed*2;
264         //SB와 BB는 시드금액을 지불합니다.
265
266         for (var i=0;i<2*player.length;i++) beting.push(player[i%player.length]);
267         beting.push(player[0]);
268         beting.push(player[1]);
269         beting.splice(0,2);
270         //첫 베팅의 순서:SB의 좌측부터 시작
271         pc="";
272         for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+": "+pot[k]+"개 베팅/
273         replier.reply(beting[m]+"님부터 베팅을 시작합니다.\n/콜, /폴드, /레이즈 (숫자
274         isBet=true; //베팅시작
275         } else replier.reply("이미 게임이 시작되었습니다.");
276         } else replier.reply("아직 방이 만들어지지 않았습니다.");
277     } else replier.reply("최소 참가자는 2명입니다.");
278 }
279
280 if (isFirstgame==false && cmd=="/업" && isNaN(data)==false){ //시드 올릴까
281     data=Number(data);
282     basicseed+=Math.floor(data);
283     if (basicseed/basicchips>0.05){
284         basicseed=Math.floor(basicchips*0.05);
285         replier.reply("최고시드("+basicseed+"개)로 자동 설정되었습니다.");
286         isFirstgame=true;
287     } else {
288         replier.reply("시드가 "+data+"개 올라 "+basicseed+"개가 되었습니다.");
289         isFirstgame=true;
290     }
291 } else if (isFirstgame==false && msg=="/킵") isFirstgame=true; //올리지 말까
292
293 if (isBet==true){ //베팅 스크립트
294     if (msg == "/체크"){ //체크
295         if (isCheck==0 && sender==beting[m] && room==R && status>=1){
296             replier.reply("체크! 다음 "+beting[m+1]+"님 베팅해주세요.\n/콜, /폴드, /레이즈 (
297             m+=1;
298             isCheck+=1;
299         }
300     } else if (msg=="/콜" && sender==beting[m] && room==R){ //콜
301         if (status==0 || isCheck!=0){
302
303             if (allin.indexOf(sender)==-1 && (Math.max.apply(null, pot)-pot[player.indexOf(
304             var whole=coin[player.indexOf(sender)];
305             coin[player.indexOf(sender)]=0;
306             chips[player2.indexOf(sender)]=0;
307             money += whole;
308             pot[player.indexOf(sender)]+=whole;
309             pot2[player2.indexOf(sender)]+=whole;
310             side.push(money);
311             allin.push(sender);
312             replier.reply(sender+"님 올인! 사이드베팅이 적용됩니다."); //콜 금액이 보유금
313         } else if (allin.indexOf(sender)!=-1) replier.reply(sender+"님은 올인하셨으므로
314         else {
315             coin[player.indexOf(sender)]=coin[player.indexOf(sender)]-Math.max.apply(nu
316             chips[player2.indexOf(sender)]=chips[player2.indexOf(sender)]-Math.max.app
317             money = money+ Math.max.apply(null, pot)-pot[player.indexOf(sender)];
318             pot[player.indexOf(sender)]=Math.max.apply(null, pot);
319             pot2[player2.indexOf(sender)]=Math.max.apply(null, pot2);
320         }
321         pc="";
322         for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+": "+pot[k]+"개 베팅/"
323         pc=pc.trim();
324         call +=1; //콜 횟수

```

```

325
326         if (call==player.length-1){ //콜을 마지막으로 베팅 완료
327             replier.reply("베팅이 완료되었습니다.\n\nPOT: "+money+"개\n\n"+pc.trim());
328             isBet=false;
329             status+=1;
330             call=0;
331             isCheck=0;
332         } else { //베팅이 완료되지 않았을 시 다음 주자로 넘어갑니다.
333             replier.reply("다음 "+beting[m+1]+"님 베팅해주세요.\n/콜, /폴드, /레이즈 (숫자)");
334             m+=1;
335             isCheck+=1;
336         }
337     }
338 } else if (msg=="/폴드" && sender==beting[m] && room==R){ //폴드
339     if (player.length != 2) { //플레이어가 2인 이상일때
340         coin.splice(player.indexOf(sender),1);
341         pot.splice(player.indexOf(sender),1);
342         cardlist.splice(player.indexOf(sender),1);
343         player.splice(player.indexOf(sender),1);
344         if (allin.indexOf(sender)!=-1){
345             side.splice(allin.indexOf(sender),1);
346             allin.splice(allin.indexOf(sender),1);
347         }
348         for (var i=0;i<=2;i++) beting.splice(beting.indexOf(sender),1);
349         pc="";
350         for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+": "+pot[k]+"개 베팅/"+coin[k]+"개";
351         replier.reply(sender+"님 폴드(다이)하셨습니다.");
352
353         if (nagagi.indexOf(sender)!=-1){
354             chips.splice(player2.indexOf(sender),1);
355             pot2.splice(player2.indexOf(sender),1);
356             player2.splice(player2.indexOf(sender),1);
357
358             replier.reply(sender+"님이 나가셨습니다.\n남은 참가자: "+player2);
359             nagagi.splice(nagagi.indexOf(sender),1);
360         } //폴드 시 게임에서 완전 나가집니다.
361
362         if (call==player.length-1){ //폴드를 마지막으로 베팅이 완료될 경우
363             replier.reply("베팅이 완료되었습니다.\n\nPOT: "+money+"개\n\n"+pc.trim());
364             isBet=false;
365             status+=1;
366             call=0;
367             isCheck=0;
368         } else { //베팅이 완료되지 않았을 경우
369             m=m-Math.floor(m/(player.length+1));
370             if (isCheck==0) replier.reply("다음 "+beting[m]+"님 베팅해주세요.\n/체크, /폴드, /레이즈 (숫자)");
371             //첫 주자가 폴드할 경우
372             else replier.reply("다음 "+beting[m]+"님 베팅해주세요.\n/콜, /폴드, /레이즈 (숫자)");
373             //아닐 경우
374         }
375     } else { //플레이어가 2명이었을 경우. 폴드 시 남은 1명이 자동 승리하게 됩니다.
376         player.splice(player.indexOf(sender),1);
377         chips[player2.indexOf(player[0])]+=money;
378         pc="현재 칩 보유량";
379         for (var k=0; k<=player2.length-1;k++) pc=pc+"\n"+player2[k]+": "+chips[k]+"개";
380         replier.reply("나머지 선수가 모두 폴드하여 "+player+"님 승리입니다!\n\n"+pc);
381         if (nagagi.length!=0){
382             for (i=0;i<nagagi.length;i++){
383                 player2.splice(player2.indexOf(nagagi[i]),1);
384                 chips.splice(player2.indexOf(nagagi[i]),1);
385             }
386             replier.reply(nagagi+"님이 나가셨습니다.\n남은 참가자: "+player2);
387         }
388         Gstart=false;
389         player=[]; coin=[]; pot=[]; pot2=[];
390         cardlist=[]; common=[];
391         allin=[]; side=[];
392         player2.unshift(player2[player2.length-1]);
393         chips.unshift(chips[chips.length-1]);

```

```

393     chips.push(chips[sender]);
394     player2.pop();
395     chips.pop();
396     //게임이 끝날 때마다 자리가 한칸씩 밀려나 SB와 BB가 바뀝니다.
397     for (var i=0; i<=player2.length-1;i++){
398         coin.push(chips[i]);
399         player.push(player2[i]);
400     }
401     for(var j=0; j<=player.length-1; j++){
402         pot.push(0);
403         pot2.push(0);
404         cardlist.push([]);
405     }
406     beting=[]; money=0; call=0; m=0; isBet=false; status=0;
407     x="";
408     isFirstgame=false;
409     nagagi=[];
410     isCheck=0;
411     replier.reply("새로운 참가자가 저에게 갠톡으로 '/홀덤 참가'를 보내주시면 참가 처리
412 }
413 } else if (cmd=="/레이즈" && sender==beting[m] && room==R){ //레이즈
414     if (isNaN(data) == false){
415         if ((data-Math.floor(data))==0){
416             if ((Math.max.apply(null, pot)-pot[player.indexOf(sender)])<data && data<=
417                 if (m<beting.length-player.length){
418                     if(allin.indexOf(sender)==-1){
419                         data=Number(data);
420                         if (data>=coin[player.indexOf(sender)]) { //레이즈 금액이 보유금
421                             var whole=coin[player.indexOf(sender)];
422                             coin[player.indexOf(sender)]=0;
423                             chips[player2.indexOf(sender)]=0;
424                             money += whole;
425                             pot[player.indexOf(sender)]+=whole;
426                             pot2[player2.indexOf(sender)]+=whole;
427                             side.push(money);
428                             allin.push(sender);
429                             replier.reply(sender+"님 올인! 사이드베팅이 적용됩니다.");
430                         } else { //아닐 경우
431                             pot[player.indexOf(sender)]+=data;
432                             pot2[player2.indexOf(sender)]+=data;
433                             coin[player.indexOf(sender)]-=data;
434                             chips[player2.indexOf(sender)]-=data;
435                             money +=data;
436                         }
437                         call=0; //레이즈 시 콜 횟수가 초기화됩니다.
438                         pc="";
439                         for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+": "+pot[
440                             replier.reply("다음 "+beting[m+1]+"님 베팅해주세요.\n/콜, /폴드,
441                             m+=1;
442                             isCheck+=1;
443                         } else replier.reply("올인한 플레이어는 /콜, /폴드 만 가능합니다.");
444                     } else replier.reply("레이즈 횟수 최대치 달성으로 /콜, /폴드 만 가능합니
445                         //레이즈는 최대 2번(콜과 폴드를 하지 않아야) 가능합니다.
446                     } else replier.reply("베팅금액이 콜 이하이거나 레이즈 최대치("+2*Math.max.ap
447                     } else replier.reply("자연수만 입력해주세요.");
448                 } else replier.reply("숫자만 입력해주세요.");
449             }
450         }
451     }
452     if (status==1 && isBet==false && common.length==0){ //첫 베팅이 끝났을 때
453         m=0;
454         beting=[];
455         for (var i=0; i<3*player.length; i++) beting.push(player[i%player.length]);
456         for (var i=0; i<3; i++){
457             common.push(cards[cards.length-1]);
458             cards.pop();
459         }
460         for (var i=0;i<player.length;i++) for (var j=0;j<common.length;j++) cardlist[i].push(cc
461         //두번째 이상의베팅에서는 SB부터 베팅을 하도록 순서가 변경됩니다.
462         var ans="고토 카드 교환 개개합니다\n\n"+common; //고토 카드 교환

```



```

462     var ans="공동 카드 1장을 공개합니다.\n" +common; //공동 카드 공개
463     replier.reply(sh(ans));
464     for (var i=0; i<player.length; i++){
465         var ans="현재 나의 카드\n"+cardlist[i]+"현재 등급: "+outgrade(grade(cardlist[i]))[0];
466         replier.reply(player[i],sh(ans));
467     } //공동 카드가 공개될 때마다 현재 나의 카드와 등급이 공개됩니다.
468     replier.reply(beting[m]+"님부터 베팅을 시작합니다.\n/체크, /폴드, /레이즈 (숫자) 로 베팅이
469     isBet=true; //베팅 다시 시작!
470 }
471
472 if (status>=2 && status<=3 && isBet==false && common.length==status+1){ //두번째, 세번째 베팅
473     m=0;
474     if (status==3 ) x="마지막 ";
475     common.push(cards[cards.length-1]);
476     cards.pop();
477     for (var j=0;j<player.length;j++) cardlist[j].push(common[status+1]);
478     var ans=x+"추가 공통 카드 1장을 공개합니다: "+common[status+1]+"\n공통 카드: "+common;
479     replier.reply(sh(ans));
480     for (var i=0; i<player.length; i++){
481         var ans="현재 나의 카드\n"+cardlist[i]+"현재 등급: "+outgrade(grade(cardlist[i]))[0];
482         replier.reply(player[i],sh(ans));
483     }
484     replier.reply(beting[m]+"님부터 "+x+"베팅을 시작합니다.\n/체크, /폴드, /레이즈 (숫자) 로
485     isBet=true; //베팅 다시 시작!
486 }
487
488 if (status==4 && isBet==false){ //마지막 베팅이 끝나고 결과를 계산합니다.
489     var gl=[]; //플레이어별 족보 등급 리스트
490     var prize=[]; //플레이어별 획득할 수 있는 칩 개수
491     var winners=[]; //플레이어를 1등부터 순위대로 나열
492     var ult="공통 카드: "+common+"\n\n핸드 카드를 공개합니다."; //각자의 카드 모두 공개
493     var Res="";
494     replier.reply("마지막 베팅이 종료되었습니다.");
495     for (var i=0; i<player.length; i++){
496         var gg=player[i]+": "+outgrade(grade(cardlist[i]))[0])+"\n";
497         Res=Res+gg;
498         gg=player[i]+": "+cardlist[i][0]+", "+cardlist[i][1];
499         ult=ult+"\n"+gg;
500     } //족보 등급 출력
501     for (var i=0;i<player.length;i++) gl.push([grade(cardlist[i]),player[i]]);
502     gl.sort((a,b)=>b[0][2]-a[0][2]);
503     gl.sort((a,b)=>b[0][1]-a[0][1]);
504     gl.sort((a,b)=>b[0][0][1]-a[0][0][1]);
505     gl.sort((a,b)=>b[0][0][0]-a[0][0][0]); //등급을 순위대로 나열
506     for (var i=0;i<gl.length;i++) winners.push(gl[i][1]); //순위에 맞는 플레이어 나열
507     for (var i=0; i<winners.length; i++) prize.push(money);
508     if (allin.length>0) for (var i=0; i<allin.length; i++) prize[winners.indexOf(allin[i])]
509     //플레이어별 획득가능 칩 배열. 올인한 플레이어의 획득가능 칩은 사이드금액으로 저장됩니다.
510     for (var i=0;i<prize.length-1;i++){
511         if (prize[i]>prize[i+1]) {
512             winners.splice(i+1,prize.length-i-1);
513             prize.splice(i+1,prize.length-i-1);
514             break;
515         }
516     } //상금을 받을 수 있는 플레이어들만 가려냅니다.
517     replier.reply(sh(ult));
518     replier.reply(sh(Res.trim()));
519
520     for (var i=0;i<winners.length;i++){
521         chips[player2.indexOf(winners[i])]+=prize[i];
522         for (var j=i;j<prize.length-1;j++) prize[j+1]-=prize[i];
523     } //상금 분배
524     pc="현재 칩 보유량";
525     for (var k=0; k<=player2.length-1;k++) pc=pc+"\n"+player2[k]+": "+chips[k]+"개";
526     replier.reply("승리자는 "+winners[0]+"님입니다!\n\n"+pc);
527     //결과 출력
528     if (nagagi.length!=0){
529         for (i=0;i<nagagi.length;i++){
530             player2.splice(player2.indexOf(nagagi[i]),1);

```



```

531         chips.splice(player2.indexOf(nagagi[1]),1);
532     }
533     replier.reply(nagagi+"님이 나가셨습니다.\n남은 참가자: "+player2);
534 } //나가기
535 Gstart=false;
536 coin=[]; player=[]; pot=[]; pot2=[];
537 cardlist=[]; common=[];
538 player2.unshift(player2[player2.length-1]);
539 chips.unshift(chips[chips.length-1]);
540 player2.pop();
541 chips.pop(); //자리가 바뀝니다
542 for ( var i=0; i<=player2.length-1;i++){
543     coin.push(chips[i]);
544     player.push(player2[i]);
545 }
546 for(var j=0; j<=player.length-1; j++){
547     pot.push(0);
548     pot2.push(0);
549     cardlist.push([]);
550 }
551 beting=[]; money=0; call=0; m=0; isBet=false; status=0;
552 x="";
553 isFirstgame=false;
554 nagagi=[];
555 isCheck=0;
556 allin=[]; side=[];
557 replier.reply("새로운 참가자가 저에게 랜톡으로 '/홀덤 참가'를 보내주시면 참가 처리됩니다.\n");
558 }
559
560 if (msg=="/나가기" && player2.indexOf(sender)!=-1 && nagagi.indexOf(sender)==-1){
561     if (Gstart==true){ //게임이 진행중인 경우 나가기가 예약됨
562         nagagi.push(sender);
563         replier.reply(R, sender+"님의 나가기가 예약되었습니다. 폴드 시 혹은 한 게임이 끝날 시
564     } else { //게임이 진행 중인 경우 바로 나가짐
565         player.splice(player.indexOf(sender),1);
566         player2.splice(player2.indexOf(sender),1);
567         replier.reply(R, sender+"님이 나가셨습니다.\n남은 참가자: "+player2);
568     }
569 }
570
571 if (msg=="/나가기취소" && player2.indexOf(sender)!=-1 && nagagi.indexOf(sender)!=-1){
572     nagagi.splice(nagagi.indexOf(sender),1);
573     replier.reply(R, sender+"님 나가기 예약 취소");
574 } //나가기 예약이 되어 있었을 경우 취소
575
576 if (Gstart==true && player2.length==1){
577     replier.reply("참가자가 1명밖에 남지 않아 게임이 자동 종료됩니다.");
578     Api.compile("홀덤.js");
579 } //모두 나가고 1명밖에 남지 않았을때 자동 게임 종료
580
581 if (msg=="/홀덤 종료" && player2.indexOf(sender)!=-1){
582     turnoff=true;
583     onoff.push(sender);
584     replier.reply("/찬성 으로 종료 찬성이 가능합니다. 5초 이내 과반수 이상 찬성 시 게임이 종료");
585     java.lang.Thread.sleep(5000);
586     if (onoff.length>=player2.length/2){
587         replier.reply(onoff.length+"명의 찬성으로 게임이 종료됩니다.");
588         Api.compile("홀덤.js");
589     } else {
590         replier.reply("과반수가 넘지 않아 게임이 계속 진행됩니다.");
591         onoff=[];
592         turnoff=false;
593     }
594 } //게임 종료: 투표제. 5초 이내 과반수 이상 찬성 시 자동 종료. 종료 명령어를 친 사람은 자동 찬성
595
596 if (turnoff=true && onoff.indexOf(sender)==-1 && msg=="/찬성"){
597     onoff.push(sender);
598     replier.reply(sender+"님 찬성");
599 } //찬성

```



seori님의 게시글 더보기 >

❤ 좋아요 0 💬 댓글 31

🔗 공유 | 신고

댓글 등록순 최신순 ↻

댓글알림 ☐

미니

왁ㅋ 계속 업뎃되네음

2020.02.07. 23:10 답글쓰기



seori 작성자

친구랑 직접 플레이해보니 볼 수 없었던 다양한 오류들이 보이네여 ㅎㅎ

2020.02.07. 23:14 답글쓰기



마른얼음

한 수 배우고 싶습니다 스승님
어떻게 저렇게 코드를 깔끔하게 짤 수가 있지

2020.02.07. 23:12 답글쓰기



seori 작성자

고인얼음님 저 약올리세요?

2020.02.07. 23:13 답글쓰기



마른얼음

seori 진심으로 솔직하게 말하는거예요
저보다 코드 짜는 실력이 훨씬 뛰어나셔요

2020.02.07. 23:14 답글쓰기



미니

seori 그럼 저를 제자로 받아주세요

2020.02.07. 23:14 답글쓰기



seori 작성자

마른얼음 거의 대학생이 초딩한테 공부잘한다고 하는거 아니예여? ㅠ

2020.02.07. 23:16 답글쓰기



마른얼음

seori 아니 진짜 잘하는건데 왜그러세요 ㅠㅠㅠ
알고리즘 부분은 제가 조금 약해서 그런건지도 모르겠는데...

2020.02.07. 23:20 답글쓰기



마른얼음

미니 띠용? 제 톡방으로 오세요

2020.02.07. 23:20 답글쓰기



미니

마른얼음 링크 부탁요

2020.02.07. 23:21 답글쓰기



seori 작성자

마른얼음 <https://cafe.naver.com/nameeyee/11098> ??

2020.02.07. 23:40 답글쓰기



마른얼음

seori 아 스큐 개같은거

2020.02.08. 08:43 답글쓰기



seori 작성자



마른얼음 ㅋㅋㅋㅋㅋㅋㅋㅋ 전 아직 멀었어요 ㅜ
그래도 칭찬 정말 감사합니다

2020.02.08. 10:57 답글쓰기

삭제된 댓글입니다.



seori 작성자

?? 처음 들어보는 말이네요!

2020.02.07. 23:15 답글쓰기



카톡우앙

굿굿

2020.02.07. 23:23 답글쓰기



AlphaDo

저런 방식도 깔끔하군...예전 방식으로 갈아탈ㄱ

질문이 있어서 카페 개인챗으로 채팅 부탁드립니다 ㅇㅂㅇ

2020.02.07. 23:52 답글쓰기



seori 작성자

예뻐

2020.02.08. 10:57 답글쓰기



wemer12

/홀덤 생성 이 안되는데 어떻게 해결하나요?

컴파일은 했고 도움말도 잘나오는데 방 생성이 안되네요 ㅠ

2020.02.08. 15:50 답글쓰기



seori 작성자

단톡방에서만 실행되도록 설정해놓았어요~

2020.02.08. 15:51 답글쓰기



wemer12

seori 알고있어요 ㅠ 근데도 안되는건 ㅠ

2020.02.08. 15:52 답글쓰기



seori 작성자

wemer12 컴파일해도 안되냐여? 캡처 한번 해주시겠어요?

2020.02.08. 15:52 답글쓰기



wemer12

seori 분명 그룹채팅인데



2020.02.08. 15:55 답글쓰기



seori 작성자

wemer12 헬 뭐지 그럼 안될 이유가 없는데
메신저봇 내 디버깅룸 비워져있고 저 단톡방에 봇 있고 알림 잘 가는거 맞져?

2020.02.08. 15:57 답글쓰기



wemer12

seori 내 π 디버깅 룸에서는 오류 날까봐 방생성을 안했고
도움말만 잘나와요

2020.02.08. 15:58 답글쓰기



seori 작성자

wemer12 아예 읽음 처리도 안됐다는건 알림이 잘 안갔거나 연결 문제거나 하는거 아닌가요.. π 파일 그대로 사용하
셨나요?

2020.02.08. 15:59 답글쓰기



wemer12

seori 제가 뽕대가리라 파일을 열주 몰라서 $\pi\pi$
복불했어요
파일을 적용어케 해요?

2020.02.08. 16:02 답글쓰기



seori 작성자

wemer12 복불해도 잘 돼야할텐데..

2020.02.08. 16:03 답글쓰기



wemer12

seori 제가 아아어아아앙어앙 복불을 바보같이 한것 같아요

2020.02.08. 16:06 답글쓰기



seori 작성자

wemer12 지금 업데이트글 올릴건데 한번 다시 시도해보셔요

2020.02.08. 16:17 답글쓰기



seori 작성자

wemer12 해결되셨나요?

2020.02.08. 20:37 답글쓰기



wemer12

seori 흐규규
일부로 계속 일리리 물어보니까 실례인거 가타소
더 안말했어요 ㅜ
이번엔 1줄로 되서 먹통이다 ㅏ ㅏ ㅏ

2020.02.08. 20:39 답글쓰기

Hibot

댓글을 남겨보세요



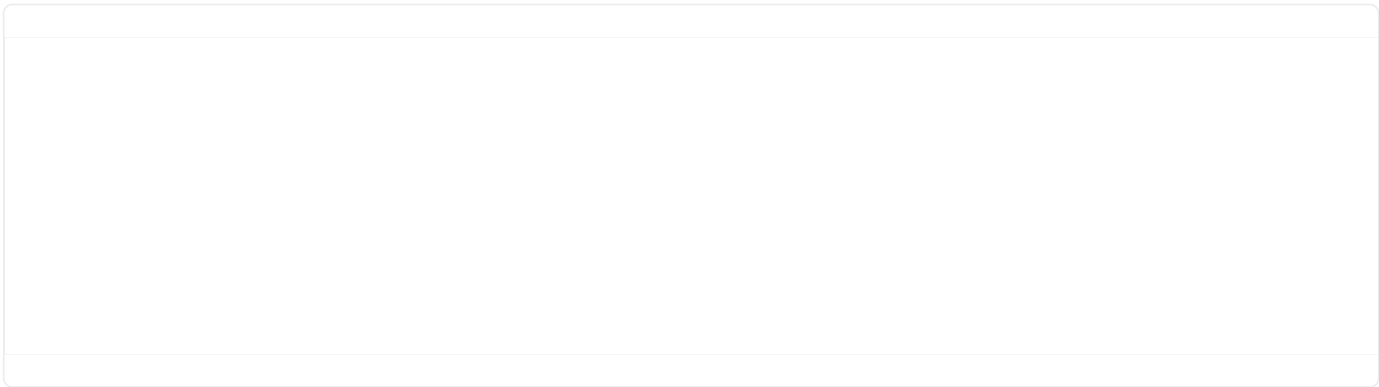
등록

글쓰기

답글

목록

▲ TOP



'| 일반 소스 공유 |' 게시판 글

이 게시판 새글 구독하기 ☐

최저시급 [1]	dowhd	2020.02.08.
시차 알려주는 소스 🐼 [5]	deadmau5	2020.02.08.
(공유) 포커[텍사스 홀덤] v2.0 by seori 🐼 [31]	seori	2020.02.07.
초급 인사봇 소스 [8]	soulout	2020.02.07.
실용적인 욕감지 🐼 🐼 [15]	알랑뽕까	2020.02.07.

1 2 3 전체보기

이 카페 인기글

자바스크립트

도유니
♡0 💬10

"시초가 중국발" 발언 관련 해명글 및 부탁의 글

틱택토 (Player vs Player)

마녀사냥 금지

AlphaDo
♡3 💬8

안녕하세요

안녕하세요

tomohong
♡1 💬3

자동학습 봇 특정 말만 무시하기?

헤히
♡0 💬4

태양, 달 정보 구현 완료

오른이발 괴롭히기(메봇R)

ERROR
♡0 💬7

1 2 3 4 5