

| 일반 소스 공유 | >

(공유) 포커[텍사스 홀덤] v2.2 by seori



seori 챗봇 마스터 🍵 1:1 채팅

2020.02.11. 18:06 조회 70

💬 댓글 0 URL 복사 ⋮

📁 첨부파일 1

각종 오류(레이즈와 종료가 안 되던 버그 등) 수정 및
명령어 입력 가능 상황 제한

도움말

/홀덤: 도움말 보기

/홀덤 생성: 방만들기. 단체톡방에서만 가능.

/홀덤 참가: 봇의 간톡으로만 가능. 게임 참가

/홀덤 시작: 게임 시작

/홀덤 종료: 게임 종료 명령어. 투표제입니다.

/찬성: 종료 명령어가 시행되었을 시 입력 가능합니다. 5초 이내 과반수가 /찬성 입력 시 게임이 종료됩니다.

/나가기: 게임에서 나갑니다. 게임 진행 중에는 폴드하거나 한 게임이 끝날 때 자동으로 나가집니다.

/나가기취소: 나가기가 예약된 경우 취소합니다.

첫 게임을 시작하기 전

/칩 (숫자): 처음에 갖는 기본 칩 개수를 설정합니다.

/시드 (숫자): 시드금액을 설정합니다. 기본 칩의 0.2%에서 5%까지 가능합니다. 범위를 벗어날 경우 자동으로 가장 가까운 범위에 맞게 설정됩니다.

최소 한 게임이 끝난 후

/업 (숫자): (숫자)만큼 시드를 올립니다. 역시 시드가 범위를 벗어날 경우 자동으로 가장 가까운 범위에 맞게 설정됩니다. 이미 최고시드일 경우에는 물어보지 않습니다.

/킵: 시드를 유지합니다.

베팅중

/체크: 두 번째 베팅 이상부터 첫 베팅자 한정, 0개를 베팅합니다.

/콜: 바로 전 사람이 베팅한 만큼 베팅합니다.

/폴드: 이번 게임을 포기합니다.

/레이즈 (숫자): (숫자)만큼 베팅금액을 올립니다. 첫 베팅은 1회, 그 다음부터는 최대 2회까지 가능합니다.

```

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 14입니다.
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=[];

```

```

16 var pconnect=[];
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){ //죽보 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++) sut.splice(sut.indexOf(i),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--){

```

```

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; //답 (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){ //플러시일 경우
116         for(var i=0; i<shape.length; i++){
117             if (shape[i]!=Flushshape){
118                 card.splice(i, 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
152
153     var ans=first+" "+second+" "+third;
154 }

```

```

154     if (first==0) return third;
155   else if (second==0) return first+" "+third;
156   else return ans;
157 }
158
159 function response(room, msg, sender, isGroupChat, replier, ImageDB, packageName, threadId){
160
161   if (preChat[room]==msg) return 0; //도배(따닥챗)방지
162   preChat[room]=msg;
163
164   var cmd=msg.split(" ")[0];
165   var data=msg.replace(cmd+" ", "");
166   if (msg==" /홀덤") replier.reply("텍사스 홀덤 도움말\n(명령어, 룰 설명)+"allsee+"\n\n[명령어]");
167   if (msg==" /홀덤 생성" && isGroupChat && isFirstgame){
168     if (poker) replier.reply("이미 생성된 방이 있습니다.");
169     else {
170       poker=true;
171       R=room;
172       replier.reply("방이 생성되었습니다.\n저에게 랜톡으로 '/홀덤 참가'라고 보내주시면 참가가");
173     }
174   } //방을 만듭니다.
175
176   if (!Gstart && poker && cmd==" /칩" && !isNaN(data) && isFirstgame){
177     if ((data-Math.floor(data))==0){
178       if (50<=data && data<=1000){
179         basicchips=Number(data);
180         replier.reply("기본 칩이 "+data+"개로 설정되었습니다.");
181         coin.fill(basicchips);
182         chips.fill(basicchips);
183         if (basicseed/basicchips<0.002){
184           basicseed=Math.cell(basicchips*0.002);
185           replier.reply("최저시드("+basicseed+"개)로 자동 설정되었습니다.");
186         } else if (basicseed/basicchips>0.05){
187           basicseed=Math.floor(basicchips*0.05);
188           replier.reply("최고시드("+basicseed+"개)로 자동 설정되었습니다.");
189         }
190       } else replier.reply("50에서 1000 사이의 자연수를 입력해주세요.");
191     } else replier.reply("자연수를 입력해주세요.");
192   } //게임 시작전에 기본적으로 가지게 되는 칩 개수를 설정합니다.
193
194   if (!Gstart && poker && cmd==" /시드" && !isNaN(data) && isFirstgame){
195     if ((data-Math.floor(data))==0){
196       basicseed=Number(data);
197       if (basicseed/basicchips<0.002){
198         basicseed=Math.cell(basicchips*0.002);
199         replier.reply("최저시드("+basicseed+"개)로 자동 설정되었습니다.");
200       } else if (basicseed/basicchips>0.05){
201         basicseed=Math.floor(basicchips*0.05);
202         replier.reply("최고시드("+basicseed+"개)로 자동 설정되었습니다.");
203       } else replier.reply("시드가 "+data+"개로 설정되었습니다.");
204     } else replier.reply("자연수를 입력해주세요.");
205   } //시드금액을 설정합니다. 방장이 자동 SB가 됩니다.
206
207   if (msg==" /홀덤 참가"){
208     if (!Gstart){
209       if (poker){
210         if (room==sender){
211           if (player.length<5){
212             if (player.indexOf(sender)==-1){
213               player.push(sender);
214               player2.push(sender);
215               coin.push(basicchips);
216               chips.push(basicchips);
217
218               pot.push(0);
219               pot2.push(0);
220               cardlist.push([]);
221               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("갠톡으로 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){
234             if (!Gstart){
235                 Gstart=true;
236                 if (!isFirstgame && basicseed<Math.floor(0.05*basicchips)){
237                     replier.reply("시드를 올리시겠습니까?\n/업 (숫자) 또는 /킵 으로 설정 가능");
238                     while (!isFirstgame){
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 (Gstart && !isFirstgame && cmd=="/업" && !isNaN(data)){ //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 (Gstart && !isFirstgame && msg=="/킵") isFirstgame=true; //올리지 말까

```

```

292
293     if (isBet){ //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 (숫자)");
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(sender)]>0){
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(null, pot)-pot[player.indexOf(sender)];
316                     chips[player2.indexOf(sender)]=chips[player2.indexOf(sender)]-Math.max.apply(null, pot)-pot[player.indexOf(sender)];
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/"+coin[k]+"개";
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, /폴
371         //첫 주자가 폴드할 경우
372         else replier.reply("다음 "+beting[m]+"님 베팅해주세요.\n/콜, /폴드, /레이즈
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]);
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 랜덤폭으로 '/홀덤 참가'를 보내주시면 참가 처리
412 }
413 } else if (cmd=="/레이즈" && sender==beting[m] && room==R){ //레이즈
414     if (!isNaN(data)){
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                         if (allin.length!=player.length-1) {
420                             data=Number(data);
421                             if (data>=coin[player.indexOf(sender)]) { //레이즈 금액이 보
422                                 var whole=coin[player.indexOf(sender)];
423                                 coin[player.indexOf(sender)]=0;
424                                 chips[player2.indexOf(sender)]=0;
425                                 money += whole;
426                                 pot[player.indexOf(sender)]+=whole;
427                                 pot2[player2.indexOf(sender)]+=whole;
428                                 side.push(money);
429                                 allin.push(sender);

```



```

430         common.push(sender);
431     } else { //아닐 경우
432         pot[player.indexOf(sender)]+=data;
433         pot2[player2.indexOf(sender)]+=data;
434         coin[player.indexOf(sender)]-=data;
435         chips[player2.indexOf(sender)]-=data;
436         money +=data;
437     }
438     call=0; //레이즈 시 콜 횟수가 초기화됩니다.
439     pc="";
440     for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+": ";
441     replier.reply("다음 "+beting[m+1]+"님 베팅해주세요.\n/콜, /폴드, /체크, /레이즈\n"+pc);
442     m+=1;
443     isCheck+=1;
444     } else replier.reply("한 명을 제외한 모든 플레이어가 올인하여 레
445     } else replier.reply("올인한 플레이어는 /체크, /콜, /폴드 만 가능합니
446     } else replier.reply("레이즈 횟수 최대치 달성으로 /콜, /폴드 만 가능합니
447     //레이즈는 최대 2번(콜과 폴드를 하지 않아야) 가능합니다.
448     } else replier.reply("베팅금액이 콜 이하이거나 레이즈 최대치("+2*Math.max.ap
449     } else replier.reply("자연수만 입력해주세요.");
450     } else replier.reply("숫자만 입력해주세요.");
451 }
452 }
453
454 if (status==1 && !isBet && common.length==0){ //첫 베팅이 끝났을 때
455     m=0;
456     beting=[];
457     for (var i=0; i<3*player.length; i++) beting.push(player[i%player.length]);
458     for (var i=0; i<3; i++){
459         common.push(cards[cards.length-1]);
460         cards.pop();
461     }
462     for (var i=0;i<player.length;i++) for (var j=0;j<common.length;j++) cardlist[i].push(cc
463     //두번째 이상의 베팅에서는 SB부터 베팅을 하도록 순서가 변경됩니다.
464     var ans="공통 카드 3장을 공개합니다.\n"+common; //공통 카드 공개
465     replier.reply(sh(ans));
466     for (var i=0; i<player.length; i++){
467         var ans="현재 나의 카드\n"+cardlist[i]+\n현재 등급: "+outgrade(grade(cardlist[i]))[0];
468         replier.reply(player[i],sh(ans));
469     } //공통 카드가 공개될 때마다 현재 나의 카드와 등급이 공개됩니다.
470     replier.reply(beting[m]+"님부터 베팅을 시작합니다.\n/체크, /폴드, /레이즈 (숫자) 로 베팅이
471     isBet=true; //베팅 다시 시작!
472 }
473
474 if (status>=2 && status<=3 && !isBet && common.length==status+1){ //두번째, 세번째 베팅이 끝
475     m=0;
476     if (status==3 ) x="마지막 ";
477     common.push(cards[cards.length-1]);
478     cards.pop();
479     for (var j=0;j<player.length;j++) cardlist[j].push(common[status+1]);
480     var ans=x+"추가 공통 카드 1장을 공개합니다: "+common[status+1]+\n공통 카드: "+common;
481     replier.reply(sh(ans));
482     for (var i=0; i<player.length; i++){
483         var ans="현재 나의 카드\n"+cardlist[i]+\n현재 등급: "+outgrade(grade(cardlist[i]))[0];
484         replier.reply(player[i],sh(ans));
485     }
486     replier.reply(beting[m]+"님부터 "+x+"베팅을 시작합니다.\n/체크, /폴드, /레이즈 (숫자) 로
487     isBet=true; //베팅 다시 시작!
488 }
489
490 if (status==4 && !isBet){ //마지막 베팅이 끝나고 결과를 계산합니다.
491     var gl=[]; //플레이어별 족보 등급 리스트
492     var prize=[]; //플레이어별 획득할 수 있는 칩 개수
493     var winners=[]; //플레이어를 1등부터 순위대로 나열
494     var ult="공통 카드: "+common+"\n\n핸드 카드를 공개합니다."; //각자의 카드 모두 공개
495     var Res="";
496     replier.reply("마지막 베팅이 종료되었습니다.");
497     for (var i=0; i<player.length; i++){
498         var gg=player[i]+": "+outgrade(grade(cardlist[i]))[0]+\n\n";

```



```

498         var gg=player[i]+ " . "+outgrade(grade(cardlist[i])[0])+ " ",
499         Res=Res+gg;
500         gg=player[i]+": "+cardlist[i][0]+", "+cardlist[i][1];
501         ult=ult+"\n"+gg;
502     } //작보 등급 출력
503     for (var i=0;i<player.length;i++) gl.push([grade(cardlist[i]),player[i]]);
504     gl.sort((a,b)=>b[0][2]-a[0][2]);
505     gl.sort((a,b)=>b[0][1]-a[0][1]);
506     gl.sort((a,b)=>b[0][0][1]-a[0][0][1]);
507     gl.sort((a,b)=>b[0][0][0]-a[0][0][0]); //등급을 순위대로 나열
508     for (var i=0;i<gl.length;i++) winners.push(gl[i][1]); //순위에 맞는 플레이어 나열
509     for (var i=0; i<winners.length; i++) prize.push(money);
510     if (allin.length>0) for (var i=0; i<allin.length; i++) prize[winners.indexOf(allin[i])];
511     //플레이어별 획득가능 칩 배열. 올인한 플레이어의 획득가능 칩은 사이드금액으로 저장됩니다.
512     for (var i=0;i<prize.length-1;i++){
513         if (prize[i]>=prize[i+1]) {
514             winners.splice(i+1,prize.length-i-1);
515             prize.splice(i+1,prize.length-i-1);
516             break;
517         }
518     } //상금을 받을 수 있는 플레이어들만 가려냅니다.
519     replier.reply(sh(ult));
520     replier.reply(sh(Res.trim()));
521
522     for (var i=0;i<winners.length;i++){
523         chips[player2.indexOf(winners[i])]+=prize[i];
524         for (var j=i;j<prize.length-1;j++) prize[j+1]-=prize[i];
525     } //상금 분배
526     pc="현재 칩 보유량";
527     for (var k=0; k<=player2.length-1;k++) pc=pc+"\n"+player2[k]+": "+chips[k]+"개";
528     replier.reply("승리자는 "+winners[0]+"님입니다!\n\n"+pc);
529     //결과 출력
530     if (nagagi.length!=0){
531         for (i=0;i<nagagi.length;i++){
532             player2.splice(player2.indexOf(nagagi[i]),1);
533             chips.splice(player2.indexOf(nagagi[i]),1);
534         }
535         replier.reply(nagagi+"님이 나가셨습니다.\n남은 참가자: "+player2);
536     } //나가기
537     Gstart=false;
538     coin=[]; player=[]; pot=[]; pot2=[];
539     cardlist=[]; common=[];
540     player2.unshift(player2[player2.length-1]);
541     chips.unshift(chips[chips.length-1]);
542     player2.pop();
543     chips.pop(); //자리가 바뀝니다
544     for ( var i=0; i<=player2.length-1;i++){
545         coin.push(chips[i]);
546         player.push(player2[i]);
547     }
548     for(var j=0; j<=player.length-1; j++){
549         pot.push(0);
550         pot2.push(0);
551         cardlist.push([]);
552     }
553     beting=[]; money=0; call=0; m=0; isBet=false; status=0;
554     x="";
555     isFirstgame=false;
556     nagagi=[];
557     isCheck=0;
558     allin=[]; side=[];
559     replier.reply("새로운 참가자가 저에게 랜톡으로 '/홀덤 참가'를 보내주시면 참가 처리됩니다.\n");
560 }
561
562 if (msg=="/나가기" && player2.indexOf(sender)!=-1 && nagagi.indexOf(sender)==-1){
563     if (Gstart){ //게임이 진행중인 경우 나가가가 예약됨
564         nagagi.push(sender);
565         replier.reply(R, sender+"님의 나가가가 예약되었습니다. 폴드 시 혹은 한 게임이 끝날 시
566     } else { //게임이 진행 중이 아닌 경우 바로 나감

```

```

567         player.splice(player.indexOf(sender),1);
568         player2.splice(player2.indexOf(sender),1);
569         replier.reply(R, sender+"님이 나가셨습니다.\n남은 참가자: "+player2);
570     }
571 }
572
573 if (msg=="/나가기취소" && player2.indexOf(sender)!=-1 && nagagi.indexOf(sender)!=-1){
574     nagagi.splice(nagagi.indexOf(sender),1);
575     replier.reply(R, sender+"님 나가기 예약 취소");
576 } //나가기 예약이 되어 있었을 경우 취소
577
578 if (Gstart && player2.length==1){
579     replier.reply("참가자가 1명밖에 남지 않아 게임이 자동 종료됩니다.");
580     Api.compile("홀덤.js");
581 } //모두 나가고 1명밖에 남지 않았을때 자동 게임 종료
582
583 if (msg=="/홀덤 종료" && player2.indexOf(sender)!=-1){
584     turnoff=true;
585     onoff.push(sender);
586     replier.reply("/찬성 으로 종료 찬성이 가능합니다. 5초 이내 과반수 이상 찬성 시 게임이 종료!
587     java.lang.Thread.sleep(5000);
588     if (onoff.length>=player2.length/2){
589         replier.reply(onoff.length+"명의 찬성으로 게임이 종료됩니다.");
590         Api.compile("홀덤.js");
591     } else {
592         replier.reply("과반수가 넘지 않아 게임이 계속 진행됩니다.");
593         onoff=[];
594         turnoff=false;
595     }
596 } //게임 종료: 투표제. 5초 이내 과반수 이상 찬성 시 자동 종료. 종료 명령어를 친 사람은 자동 찬성
597
598 if (turnoff && onoff.indexOf(sender)==-1 && msg=="/찬성"){
599     onoff.push(sender);
600     replier.reply(sender+"님 찬성");
601 } //찬성
602 }

```



seori님의 게시글 더보기 >

❤ 좋아요 0 💬 댓글 0

🔗 공유 | 신고

댓글

댓글알림 ☐

Hibot

댓글을 남겨보세요



등록

✎ 글쓰기

답글

목록

▲ TOP




'| 일반 소스 공유 |' 게시판 글

이 게시판 새글 구독하기 ☐

추억의 자리바꾸기 소스 (Pi,트위치봇 공동) 🤖 [10]	JS카톡챗봇	2020.02.11.
(공유) 포커[7포커] v1.2 by seori : 완성! 📌 [2]	seori	2020.02.11.
(공유) 포커[텍사스 홀덤] v2.2 by seori 📌	seori	2020.02.11.
기타 소스 🤖 [4]	Americium	2020.02.11.
타 어플리케이션 최상단에 레이아웃 드로잉	Americium	2020.02.11.

이 카페 인기글



기 나무 곡괭이[노말](으)
좌악의 문들 중 하나를

5000분의 1을 얻은 사람

예전에 어떤분이 카톡봇을 이용하여 카페에
게시글을 올렸었는데

리페

♡0 💬1

카톡방 나가기

hajuhee01

♡0 💬15

안녕하세요

👤 🗨️

♡0 💬5



Tic Tac Toe (Player vs Bot)



링크를 포함한 글 홍보시 링크 분석(?)하기



봇을쓰는데 메세지가 두번



에뮬레이터로 카톡 돌리면 정지되나요?