



| 일반 소스 공유 | >

(공유) 포커[7포커] v1.2 by seori : 완성!

 **seori** 챗봇 마스터  1:1 채팅

2020.02.11. 20:10 조회 191

댓글 2 URL 복사

첨부파일 1

기본적인 틀이 모두 갖춰졌습니다.
레이즈가 안 되던 버그, 종료가 안 되던 버그 등 오류가 수정되었습니다.
기본 시드의 최저값이 기본 칩의 0.1%로 낮춰졌습니다.
일부 명령어의 입력 가능한 상황이 제한되었습니다.

4구룰, 5구룰, 6구룰을 설정할 수 있게 되었습니다.
일반룰(콜, 폴드, 레이즈), 하우스룰(콜, 폴드, 뺑, 따당, 쿼터, 하프)을 선택할 수 있게 되었습니다.

기본 설정은 일반 5구룰입니다.
하우스룰로 설정할 경우 기본 칩 10000개, 시드 20개로 초기 설정됩니다.

업데이트 예정:
(먼 훗날이겠지만) 서버화..

이외 만들 예정인 게임:
인디언포커/로우바둑이/하이로우/원카드/도둑잡기 등 (아예 카드게임봇으로 전략.. 할 예정 ㅎㅎ)

각종 오류 지적, 업데이트 아이디어 제공 너무 감사하겠습니다.

도움말
/설정: 설정 보기. 기본 설정과 현재 설정이 나옵니다.
/설정 일반: 베팅룰을 일반룰로 바꿉니다.
/설정 하우스룰: 베팅룰을 하우스룰로 바꿉니다.
/설정 4구룰: 4구룰 게임으로 바꿉니다.
/설정 5구룰: 5구룰 게임으로 바꿉니다.
/설정 6구룰: 6구룰 게임으로 바꿉니다.
*구룰이란? N구룰: 각자 카드 N장을 가진 상태에서 첫 베팅을 시작합니다.

/7포커: 도움말을 보여줍니다.
/7포커 생성: 방을 만듭니다. 단체톡방에서만 실행됩니다.
/7포커 참가: 봇의 토크로만 실행됩니다. 게임에 참가합니다.
/7포커 시작: 게임을 시작합니다.
!a b: 핸드 카드를 받았을 때 버릴 카드와 공개 카드를 결정할 때 이용됩니다.
a번째 카드를 버리고 b번째 카드를 공개합니다.
/7포커 종료: 게임을 종료합니다. 투표제입니다.
/찬성: /7포커 종료 명령어가 실행되었을 시, 5초 이내에 과반수 이상이 /찬성 을 입력하면 게임이 종료됩니다.

/나가기: 게임에서 나갑니다. 게임 진행 중에는 폴드하거나 한 게임이 끝날 때 자동으로 나가집니다.
/나가기취소: 나가기에 예약된 경우 취소합니다.

첫 게임을 시작하기 전
/칩 (숫자): 처음에 갖는 기본 칩 개수를 설정합니다.
/시드 (숫자): 시드금액을 설정합니다. 기본 칩의 0.1%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

(공통)

/체크: 첫 베팅자 한정, 0개를 베팅합니다.

/콜: 바로 전 사람이 베팅한 만큼 베팅합니다.

/폴드: 이번 게임을 포기합니다.

(일반물)

/레이즈 (숫자): (숫자)만큼 베팅금액을 올립니다.

N구를	최대 레이즈 횟수
4구를	1-2-2-2
5구를	1-2-2
6구를	2-2

(하우스룰)

/뺑: 첫 베팅자 한정. 시드와 같은 금액을 베팅합니다.

/따당: 바로 전 사람이 베팅한 금액의 2배를 베팅합니다.

/쿼터: 현재 판돈의 25%를 베팅합니다.

/하프: 현재 판돈의 50%를 베팅합니다.

```

1 const scriptName="7포커.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", "s 14", "s 15", "s 16", "s 17", "s 18", "s 19", "s 20", "s 21", "s 22", "s 23", "s 24", "s 25", "s 26", "s 27", "s 28", "s 29", "s 30", "s 31", "s 32", "s 33", "s 34", "s 35", "s 36", "s 37", "s 38", "s 39", "s 40", "s 41", "s 42", "s 43", "s 44", "s 45", "s 46", "s 47", "s 48", "s 49", "s 50", "s 51", "s 52", "s 53", "s 54", "s 55", "s 56", "s 57", "s 58", "s 59", "s 60", "s 61", "s 62", "s 63", "s 64", "s 65", "s 66", "s 67", "s 68", "s 69", "s 70", "s 71", "s 72", "s 73", "s 74", "s 75", "s 76", "s 77", "s 78", "s 79", "s 80", "s 81", "s 82", "s 83", "s 84", "s 85", "s 86", "s 87", "s 88", "s 89", "s 90", "s 91", "s 92", "s 93", "s 94", "s 95", "s 96", "s 97", "s 98", "s 99", "s 100", "s 101", "s 102", "s 103", "s 104", "s 105", "s 106", "s 107", "s 108", "s 109", "s 110", "s 111", "s 112", "s 113", "s 114", "s 115", "s 116", "s 117", "s 118", "s 119", "s 120", "s 121", "s 122", "s 123", "s 124", "s 125", "s 126", "s 127", "s 128", "s 129", "s 130", "s 131", "s 132", "s 133", "s 134", "s 135", "s 136", "s 137", "s 138", "s 139", "s 140", "s 141", "s 142", "s 143", "s 144", "s 145", "s 146", "s 147", "s 148", "s 149", "s 150", "s 151", "s 152", "s 153", "s 154", "s 155", "s 156", "s 157", "s 158", "s 159", "s 160", "s 161", "s 162", "s 163", "s 164", "s 165", "s 166", "s 167", "s 168", "s 169", "s 170", "s 171", "s 172", "s 173", "s 174", "s 175", "s 176", "s 177", "s 178", "s 179", "s 180", "s 181", "s 182", "s 183", "s 184", "s 185", "s 186", "s 187", "s 188", "s 189", "s 190", "s 191", "s 192", "s 193", "s 194", "s 195", "s 196", "s 197", "s 198", "s 199", "s 200", "s 201", "s 202", "s 203", "s 204", "s 205", "s 206", "s 207", "s 208", "s 209", "s 210", "s 211", "s 212", "s 213", "s 214", "s 215", "s 216", "s 217", "s 218", "s 219", "s 220", "s 221", "s 222", "s 223", "s 224", "s 225", "s 226", "s 227", "s 228", "s 229", "s 230", "s 231", "s 232", "s 233", "s 234", "s 235", "s 236", "s 237", "s 238", "s 239", "s 240", "s 241", "s 242", "s 243", "s 244", "s 245", "s 246", "s 247", "s 248", "s 249", "s 250", "s 251", "s 252", "s 253", "s 254", "s 255", "s 256", "s 257", "s 258", "s 259", "s 260", "s 261", "s 262", "s 263", "s 264", "s 265", "s 266", "s 267", "s 268", "s 269", "s 270", "s 271", "s 272", "s 273", "s 274", "s 275", "s 276", "s 277", "s 278", "s 279", "s 280", "s 281", "s 282", "s 283", "s 284", "s 285", "s 286", "s 287", "s 288", "s 289", "s 290", "s 291", "s 292", "s 293", "s 294", "s 295", "s 296", "s 297", "s 298", "s 299", "s 300", "s 301", "s 302", "s 303", "s 304", "s 305", "s 306", "s 307", "s 308", "s 309", "s 310", "s 311", "s 312", "s 313", "s 314", "s 315", "s 316", "s 317", "s 318", "s 319", "s 320", "s 321", "s 322", "s 323", "s 324", "s 325", "s 326", "s 327", "s 328", "s 329", "s 330", "s 331", "s 332", "s 333", "s 334", "s 335", "s 336", "s 337", "s 338", "s 339", "s 340", "s 341", "s 342", "s 343", "s 344", "s 345", "s 346", "s 347", "s 348", "s 349", "s 350", "s 351", "s 352", "s 353", "s 354", "s 355", "s 356", "s 357", "s 358", "s 359", "s 360", "s 361", "s 362", "s 363", "s 364", "s 365", "s 366", "s 367", "s 368", "s 369", "s 370", "s 371", "s 372", "s 373", "s 374", "s 375", "s 376", "s 377", "s 378", "s 379", "s 380", "s 381", "s 382", "s 383", "s 384", "s 385", "s 386", "s 387", "s 388", "s 389", "s 390", "s 391", "s 392", "s 393", "s 394", "s 395", "s 396", "s 397", "s 398", "s 399", "s 400", "s 401", "s 402", "s 403", "s 404", "s 405", "s 406", "s 407", "s 408", "s 409", "s 410", "s 411", "s 412", "s 413", "s 414", "s 415", "s 416", "s 417", "s 418", "s 419", "s 420", "s 421", "s 422", "s 423", "s 424", "s 425", "s 426", "s 427", "s 428", "s 429", "s 430", "s 431", "s 432", "s 433", "s 434", "s 435", "s 436", "s 437", "s 438", "s 439", "s 440", "s 441", "s 442", "s 443", "s 444", "s 445", "s 446", "s 447", "s 448", "s 449", "s 450", "s 451", "s 452", "s 453", "s 454", "s 455", "s 456", "s 457", "s 458", "s 459", "s 460", "s 461", "s 462", "s 463", "s 464", "s 465", "s 466", "s 467", "s 468", "s 469", "s 470", "s 471", "s 472", "s 473", "s 474", "s 475", "s 476", "s 477", "s 478", "s 479", "s 480", "s 481", "s 482", "s 483", "s 484", "s 485", "s 486", "s 487", "s 488", "s 489", "s 490", "s 491", "s 492", "s 493", "s 494", "s 495", "s 496", "s 497", "s 498", "s 499", "s 500", "s 501", "s 502", "s 503", "s 504", "s 505", "s 506", "s 507", "s 508", "s 509", "s 510", "s 511", "s 512", "s 513", "s 514", "s 515", "s 516", "s 517", "s 518", "s 519", "s 520", "s 521", "s 522", "s 523", "s 524", "s 525", "s 526", "s 527", "s 528", "s 529", "s 530", "s 531", "s 532", "s 533", "s 534", "s 535", "s 536", "s 537", "s 538", "s 539", "s 540", "s 541", "s 542", "s 543", "s 544", "s 545", "s 546", "s 547", "s 548", "s 549", "s 550", "s 551", "s 552", "s 553", "s 554", "s 555", "s 556", "s 557", "s 558", "s 559", "s 560", "s 561", "s 562", "s 563", "s 564", "s 565", "s 566", "s 567", "s 568", "s 569", "s 570", "s 571", "s 572", "s 573", "s 574", "s 575", "s 576", "s 577", "s 578", "s 579", "s 580", "s 581", "s 582", "s 583", "s 584", "s 585", "s 586", "s 587", "s 588", "s 589", "s 590", "s 591", "s 592", "s 593", "s 594", "s 595", "s 596", "s 597", "s 598",
```

```

50     return a;
51 }
52
53 function seol(moyang){ //게임이 끝나고 '카드모양'에 등급을 매길 때 이용됩니다.
54     return moyang.replace(/s/g,3).replace(/d/g,2).replace(/h/g,1).replace(/c/g,0);
55 }
56
57 function lee(sutja){ //죽보 등급 계산을 위한 '숫자' 등급을 매기는 함수입니다.
58     var isStraight=0; var sut=[]; var ja=[];
59
60     for (var i=0;i<sutja.length;i++){
61         sutja[i]=Number(sutja[i]);
62         sut.push(Number(sutja[i]));
63         ja.push(Number(sutja[i]));
64     }
65     for (var i=0;i<sutja.filter(e => e == 14).length;i++){
66         sutja.push(1); sut.push(1); ja.push(1);
67     } //추후 백스트레이트 계산을 위해 배열에 A(14)가 포함된 경우 1을 추가해줍니다.
68
69     for (var i=14;i>1;i--) if (sutja.filter(e => e == i).length==4) return [9, i]; //포카드
70
71     for (var i=14;i>1;i--){
72         if (sutja.filter(e => e == i).length==3){
73             for (k=0;k<3;k++) sut.splice(sut.indexOf(i),1);
74             for (var j=14;j>1;j--) if (sut.filter(e => e == j).length>=2) return [8,i+0.01*j];
75         }
76     } //플하우스
77
78     for (var i=14;i>9;i--){
79         if (sutja.filter(e => e == i).length>=1){
80             isStraight+=1;
81             if (isStraight==5) return [6,0];
82         } else {
83             isStraight=0;
84             break;
85         }
86     } //마운틴
87
88     for (var i=5;i>0;i--){
89         if (sutja.filter(e => e == i).length>=1){
90             isStraight+=1;
91             if (isStraight==5) return [5,0];
92         } else {
93             isStraight=0;
94             break;
95         }
96     } //백스트레이트
97
98     for (var p=13; p>5; p--){
99         for (var i=0;i<5;i++){
100             if (sutja.filter(e => e == (p-i)).length >=1 ){
101                 isStraight+=1;
102                 if (isStraight==5) return [4,p];
103             } else break;
104         }isStraight=0;
105     } //스트레이트
106
107     for (var i=14;i>1;i--){ if (sutja.filter(e => e == i).length==3) return [3,i]; } //트리플
108
109     for (var i=14;i>1;i--){
110         if (sutja.filter(e => e == i).length==2){
111             for (k=0;k<2;k++) ja.splice(ja.indexOf(i),1);
112             for (var j=i-1;j>1;j--) if (ja.filter(e => e == j).length==2) return [2,i+0.01*j];
113             return [1,i]; //원 페어
114         }
115     }
116
117     var ans=[0,Math.max.apply(null, sutja)];
118     return ans; //타일 판인

```

```

118     return ans; //답 (오피어)
119 }
120
121 function grade(CARD){ //모양 등급인 '플러시'의 경우를 생각하여 족보의 진짜 등급을 계산합니다.
122     var Flushshape; var shape=[]; var isFlush=false;
123     var num=[]; var card=[]; var Realgrade=[];
124
125     for (var i=0; i<CARD.length; i++){
126         shape.push(seol(CARD[i].split(" ")[0]));
127         num.push(CARD[i].split(" ")[1]);
128         card.push(Number(num[i])*10+Number(shape[i]));
129     }
130
131     for (var i=0; i<4; i++){
132         if (shape.filter(e => e == i).length>=5){
133             isFlush=true;
134             Flushshape=i;
135             break;
136         }
137     } //족보가 플러시인지 아닌지 판별합니다.
138
139     if (isFlush){ //플러시일 경우
140         for(var i=0;i<shape.length; i++){
141             if (shape[i]!=Flushshape){
142                 card.splice(i,1);
143                 num.splice(i,1);
144             }
145         }
146         if (lee(num)[0]==6) Realgrade[0].push(12); //로티플
147         else if (lee(num)[0]==5) Realgrade[0].push(11); //백티플
148         else if (lee(num)[0]==4) Realgrade[0].push(10); //스티플
149         else Realgrade[0].push(7); //플러시
150     } else Realgrade[0].push(lee(num)[0]); //플러시가 아닐 경우 숫자 등급을 그대로 반환합니다.
151
152     num.sort((a,b)=>b-a);
153     var nnum=0;
154     card.sort((a,b)=>b-a);
155     var sshape=0;
156     for (var i=0;i<num.length;i++){
157         nnum+=Math.pow(0.01,i)*num[i];
158         sshape+=Math.pow(0.001,i)*card[i];
159     }
160     Realgrade[0].push(lee(num)[1]);
161     Realgrade.push(nnum);
162     Realgrade.push(sshape);
163     return Realgrade;
164     /* [[족보 등급, 등급에 해당하는 숫자], 하이 카드, 하이 카드에 따른 모양 등급]으로 전체 등급을 반환
165     등급이 무승부가 날 시 우승자를 판별하기 위함입니다.*/
166 }
167
168 function sh(shs){ //내용 출력을 위한 함수입니다.
169     return shs.replace(/s /g,"♠").replace(/d /g,"♦").replace(/h /g,"♥").replace(/c /g,"♣").
170 }
171
172 function outgrade(put){ //족보 등급 출력을 위한 함수입니다.
173     var first=Number(put[1]).toFixed(0);
174     var second=Math.round((Number(put[1])-first)*100);
175     var third=Number(put[0]).toFixed(0).replace("12", "로티플").replace("11", "백티플").replace
176
177     var ans=first+", "+second+" "+third;
178
179     if (first==0) return third;
180     else if (second==0) return first+" "+third;
181     else return ans;
182 }
183
184 function response(room, msg, sender, isGroupChat, replier, ImageDB, packageName, threadId){
185
186     if (preChat[room]==msg) return 0; //도배(따닥챗)방지

```

```

187 preChat[room]=msg;
188
189 var cmd=msg.split(" ")[0];
190 var data=msg.replace(cmd+" ", "");
191 if (msg=="/설정") {
192     replier.reply("기본 설정: 일반 5구를\n현재 설정: "+rule+" "+N+"구를");
193     replier.reply("[설정 변경]\n/설정 일반\n/설정 하우스를\n\n/설정 4구를\n/설정 5구를\n설정 (
194 }
195 if (cmd=="/설정" && !Gstart && isFirstgame) {
196     if (data=="일반" || data=="하우스를") {rule=data; replier.reply("배팅률이 "+data+"(으)로
197     if (data=="4구를") {N=4; replier.reply("4구로 변경되었습니다.");}
198     if (data=="5구를") {N=5; replier.reply("5구로 변경되었습니다.");}
199     if (data=="6구를") {N=6; replier.reply("6구로 변경되었습니다.");}
200 }
201 if (msg=="/7포커") replier.reply("세븐 카드 포커 도움말\n(명령어, 룰 설명)+"allsee+"\n\n[명령
202 if (msg=="/7포커 생성" && isGroupChat && isFirstgame){
203     if (poker) replier.reply("이미 생성된 방이 있습니다.");
204     else {
205         poker=true;
206         basicchips=[100,10000][["일반", "하우스를"].indexOf(rule)];
207         st=basicchips;
208         basicseed=[1, 20][["일반", "하우스를"].indexOf(rule)];
209         btr1=btr[["일반", "하우스를"].indexOf(rule)][0];
210         btr2=btr[["일반", "하우스를"].indexOf(rule)][1];
211         R=room;
212         replier.reply("방이 생성되었습니다.\n저에게 관톡으로 '/7포커 참가'라고 보내주시면 참가
213     }
214 } //방을 만듭니다.
215
216 if (!Gstart && poker && cmd=="/칩" && !isNaN(data) && isFirstgame){
217     if ((data-Math.floor(data))==0){
218         if (st/2<=data && data<=st*10){
219             basicchips=Number(data);
220             replier.reply("기본 칩이 "+data+"개로 설정되었습니다.");
221             coin.fill(basicchips);
222             chips.fill(basicchips);
223             if (basicseed/basicchips<0.001){
224                 basicseed=Math.cell(basicchips*0.001);
225                 replier.reply("최저시드("+basicseed+"개)로 자동 설정되었습니다.");
226             } else if (basicseed/basicchips>0.05){
227                 basicseed=Math.floor(basicchips*0.05);
228                 replier.reply("최고시드("+basicseed+"개)로 자동 설정되었습니다.");
229             }
230         } else replier.reply(st/2+"에서"+st*10+"사이의 자연수를 입력해주세요.");
231     } else replier.reply("자연수를 입력해주세요.");
232 } //게임 시작전에 기본적으로 가지게 되는 칩 개수를 설정합니다.
233
234 if (!Gstart && poker && cmd=="/시드" && !isNaN(data) && isFirstgame){
235     if ((data-Math.floor(data))==0){
236         basicseed=Number(data);
237         if (basicseed/basicchips<0.001){
238             basicseed=Math.cell(basicchips*0.001);
239             replier.reply("최저시드("+basicseed+"개)로 자동 설정되었습니다.");
240         } else if (basicseed/basicchips>0.05){
241             basicseed=Math.floor(basicchips*0.05);
242             replier.reply("최고시드("+basicseed+"개)로 자동 설정되었습니다.");
243         } else replier.reply("시드가 "+data+"개로 설정되었습니다.");
244     } else replier.reply("자연수를 입력해주세요.");
245 } //시드금액을 설정합니다.
246
247 if (msg=="/7포커 참가"){
248     if (!Gstart){
249         if (poker){
250             if (room==sender){
251                 if (player.length<5){
252                     if (player.indexOf(sender)==-1){
253                         player.push(sender);
254                         player2.push(sender);
255                         coin.push(basicchips);

```

```

256         chips.push(basicchips);
257
258         pot.push(0);
259         pot2.push(0);
260         cardlist.push([]);
261         openlist.push([]);
262         replier.reply("정상 참가 처리 되었습니다.");
263         replier.reply(R, "현재 참가자: "+player);
264     } else replier.reply(sender + "님은 이미 참가하셨습니다.");
265     } else replier.reply("최대 인원은 5명입니다.");
266     } else replier.reply("랜독으로 보내주셔야 참가가 인정됩니다.");
267     } else replier.reply("아직 방이 만들어지지 않았습니다.");
268     } else replier.reply("이미 게임이 진행중입니다.");
269 } //게임에 참가합니다.
270
271 if (msg=="/7포커 시작" && room==R && player.indexOf(sender)!=-1){ // 게임을 시작합니다.
272     if (player.length>=2){
273         if (poker){
274             if (!Gstart){
275                 Gstart=true;
276                 if (!isFirstgame && basicseed<Math.floor(0.05*basicchips)){
277                     replier.reply("시드를 올리시겠습니까?\n업 (숫자) 또는 /킵 으로 설정 가능");
278                     while (!isFirstgame){}
279                 } /*두 번째 이상의 게임부터 시드를 올릴 지 말 지 선택에 대한 질문입니다.
280                    이미 최고시드일 경우에는 묻지 않습니다.*/
281                 shuffle(cards); //카드 섞기
282                 replier.reply("게임을 시작합니다.");
283                 replier.reply("랜독으로 핸드 카드를 전달드렸습니다.\n각자 버릴 카드와 공개할 카드를 선택합니다.");
284                 for(var i=0; i<player.length-1; i++){
285                     for (var j=1; j<5; j++){
286                         cardlist[i].push(cards[cards.length-1]);
287                         cards.pop();
288                     }
289                     var ans="핸드 카드를 받았습니다.\n"+cardlist[i]+" \n버릴 카드와 공개할 카드를 선택합니다.";
290                     replier.reply(player[i],sh(ans));
291                 } //처음에 받게 되는 핸드 카드를 받습니다.
292                 for (var k=0; k<player.length; k++) ab.push(false);
293                 while (ab.filter(e=>!e).length>0) {}
294                 var pc="공개 카드";
295                 for (var i=0; i<player.length; i++){
296                     pc=pc+"\n"+player[i]+": "+openlist[i];
297                 }
298                 replier.reply(sh(pc));
299                 for (var i=0; i<player.length; i++){
300                     for (var j=1; j<N-2; j++){
301                         cardlist[i].push(cards[cards.length-1]);
302                         openlist[i].push(cards[cards.length-1]);
303                         cards.pop();
304                     }
305                 }
306                 //모든 플레이어가 카드를 한 장씩 공개하면 추가 공개 카드를 배분합니다.
307                 var pc=(N-3)+"장씩 추가 공개 카드를 배분합니다.";
308                 for (var i=0; i<player.length; i++){
309                     pc=pc+"\n"+player[i]+": "+openlist[i];
310                     var ans="현재 나의 카드\n"+cardlist[i]+" \n현재 등급: "+outgrade(grade(cards[i]));
311                     replier.reply(player[i],sh(ans));
312                 }
313                 replier.reply(R,sh(pc));
314                 java.lang.Thread.sleep(1000);
315                 replier.reply("처음 기본 칩은 "+basicchips+"개입니다. \n시드는 "+basicseed+"$");
316                 for (var i=0; i<player.length; i++){
317                     coin[i]-=basicseed;
318                     chips[i]-=basicseed;
319                     pot[i]+=basicseed;
320                     pot2[i]+=basicseed;
321                     money+=basicseed;
322                 }
323                 //공개 카드를 모두 배분하고 나면 베팅이 시작됩니다.
324                 pc="";

```

```

325         for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+": "+pot[k]+"개 베팅/
326         beting=Beting(openlist));
327         replier.reply(Boss(openlist)+"님부터 베팅을 시작합니다.\n"+btr1+"로 베팅이 기
328         isBet=1; //베팅시작
329         } else replier.reply("이미 게임이 시작되었습니다.");
330         } else replier.reply("아직 방이 만들어지지 않았습니다.");
331     } else replier.reply("최소 참가자는 2명입니다.");
332 }
333
334 if (Gstart && !isFirstgame && cmd=="/업" && !isNaN(data)){ //시드 올릴까
335     data=Number(data);
336     basicseed+=Math.floor(data);
337     if (basicseed/basicchips>0.05){
338         basicseed=Math.floor(basicchips*0.05);
339         replier.reply("최고시드("+basicseed+"개)로 자동 설정되었습니다.");
340         isFirstgame=true;
341     } else {
342         replier.reply("시드가 "+data+"개 올라 "+basicseed+"개가 되었습니다.");
343         isFirstgame=true;
344     }
345 } else if (Gstart && !isFirstgame && msg=="/립") isFirstgame=true; //올리지 말까
346
347 if (!ab[player.indexOf(sender)] && Gstart && room==sender && msg.indexOf("!")==0){ //버릴 카
348     msg=msg.replace("! ", "").replace("!", "");
349     var a=msg.split(" ")[0];
350     var b=msg.split(" ")[1];
351     if (!isNaN(a) && !isNaN(b)) {
352         a=Number(a); b=Number(b);
353         if (a==Math.floor(a) && b==Math.floor(b)){
354             if (a>=1 && b>=1 && a<=4 && b<=4 && a!=b){
355                 var xx=cardlist[player.indexOf(sender)];
356                 replier.reply(sh(a+"번째 카드 "+xx[a-1]+"을(를) 버리고 \n"+b+"번째 카드 " +x
357                 var ac=xx[a-1];
358                 var bc=xx[b-1];
359                 openlist[player.indexOf(sender)].push(bc);
360                 xx.splice(xx.indexOf(ac),1);
361                 xx.splice(xx.indexOf(bc),1);
362                 xx.push(bc);
363                 replier.reply(R, sender+"님 결정 완료");
364                 ab[player.indexOf(sender)]=true;
365             } else replier.reply(sender, "a와 b는 1 이상 4 이하의 서로 다른 자연수여야 합니다
366         } else replier.reply(sender, "a와 b는 1 이상 4 이하의 서로 다른 자연수여야 합니다.");
367     }
368 }
369
370 if (isBet==1){ //베팅 스크립트
371     if (msg == "/체크"){ //체크
372         if (isCheck==0 && sender==betting[m] && room==R ){
373             var pc="";
374             for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+": "+pot[k]+"개 베팅/"+co
375             replier.reply("체크! 다음 "+betting[m+1]+"님 베팅해주세요.\n"+btr2+"로 베팅이 가능
376             m+=1;
377             isCheck+=1;
378         }
379     } else if (msg=="/콜" && sender==betting[m] && room==R && isCheck!=0){ //콜
380         if (allin.indexOf(sender)==-1 && (Math.max.apply(null, pot)-pot[player.indexOf(
381             var whole=coin[player.indexOf(sender)];
382             just=whole;
383             coin[player.indexOf(sender)]=0;
384             chips[player2.indexOf(sender)]=0;
385             money += whole;
386             pot[player.indexOf(sender)]+=whole;
387             pot2[player2.indexOf(sender)]+=whole;
388             side.push(money);
389             allin.push(sender);
390             replier.reply(sender+"님 올인! 사이드베팅이 적용됩니다."); //콜 금액이 보유금
391         } else if (allin.indexOf(sender)!=-1) replier.reply(sender+"님은 올인하셨으므로
392         else {
393             just=Math.max.apply(null, pot)-pot[player.indexOf(sender)];

```



```

394     coin[player.indexOf(sender)]-=just;
395     chips[player2.indexOf(sender)]-=just;
396     money+=just;
397     pot[player.indexOf(sender)]+=just;
398     pot2[player2.indexOf(sender)]+=just;
399 }
400 pc="";
401 for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+": "+pot[k]+"개 베팅/"+co
402 pc=pc.trim();
403 call +=1; //콜 횟수
404
405 if (call==player.length-1){ //콜을 마지막으로 베팅 완료
406     replier.reply("베팅이 완료되었습니다.\n\nPOT: "+money+"개\n\n"+pc.trim());
407     isBet=2;
408     status+=1;
409     call=0;
410     isChecked=0;
411 } else { //베팅이 완료되지 않았을 시 다음 주자로 넘어갑니다.
412     replier.reply("다음 "+beting[m+1]+"님 베팅해주세요.\n"+btr2+"로 베팅이 가능함
413 m+=1;
414     isChecked+=1;
415 }
416 } else if (msg=="/폴드" && sender==beting[m] && room==R){ //폴드
417     if (player.length != 2) { //플레이어가 2인 이상일때
418         coin.splice(player.indexOf(sender),1);
419         pot.splice(player.indexOf(sender),1);
420         cardlist.splice(player.indexOf(sender),1);
421         player.splice(player.indexOf(sender),1);
422         if (allin.indexOf(sender)!=-1){
423             side.splice(allin.indexOf(sender),1);
424             allin.splice(allin.indexOf(sender),1);
425         }
426         for (var i=0;i<=2;i++) beting.splice(beting.indexOf(sender),1);
427         pc="";
428         for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+": "+pot[k]+"개 베팅/"+co
429         replier.reply(sender+"님 폴드(다이)하셨습니다.");
430
431         if (nagagi.indexOf(sender)!=-1){
432             chips.splice(player2.indexOf(sender),1);
433             pot2.splice(player2.indexOf(sender),1);
434             player2.splice(player2.indexOf(sender),1);
435
436             replier.reply(sender+"님이 나가셨습니다.\n남은 참가자: "+player2);
437             nagagi.splice(nagagi.indexOf(sender),1);
438         } //폴드 시 게임에서 완전 나가집니다.
439
440         if (call==player.length-1){ //폴드를 마지막으로 베팅이 완료될 경우
441             replier.reply("베팅이 완료되었습니다.\n\nPOT: "+money+"개\n\n"+pc.trim());
442             isBet=2;
443             status+=1;
444             call=0;
445             isChecked=0;
446         } else { //베팅이 완료되지 않았을 경우
447             m=m-Math.floor(m/(player.length+1));
448             if (isChecked==0) replier.reply("다음 "+beting[m]+"님 베팅해주세요.\n"+btr1+"로
449             //첫 주자가 폴드할 경우
450             else replier.reply("다음 "+beting[m]+"님 베팅해주세요.\n"+btr2+"로 베팅이 가
451             //아닐 경우
452         }
453     } else { //플레이어가 2명이었을 경우. 폴드 시 남은 1명이 자동 승리하게 됩니다.
454         player.splice(player.indexOf(sender),1);
455         chips[player2.indexOf(player[0])]+=money;
456         pc="현재 칩 보유량";
457         for (var k=0; k<=player2.length-1;k++) pc=pc+"\n"+player2[k]+": "+chips[k]+"개"
458         replier.reply("나머지 선수가 모두 폴드하여 "+player+"님 승리입니다!\n\n"+pc);
459         if (nagagi.length!=0){
460             for (i=0;i<nagagi.length;i++){
461                 player2.splice(player2.indexOf(nagagi[i]),1);
462                 chips.splice(player2.indexOf(nagagi[i]),1);

```



```

463     }
464     replier.reply(nagagi+"님이 나가셨습니다.\n남은 참가자: "+player2);
465 }
466 Gstart=false;
467 player=[]; coin=[]; pot=[]; pot2=[];
468 cardlist=[]; openlist=[];
469 allin=[]; side=[];
470 for ( var i=0; i<=player2.length-1;i++){
471     coin.push(chips[i]);
472     player.push(player2[i]);
473 }
474 for(var j=0; j<=player.length-1; j++){
475     pot.push(0);
476     pot2.push(0);
477     cardlist.push([]);
478 }
479 beting=[]; money=0; call=0; m=0; isBet=0; status=0;
480 x="";
481 ab=[];
482 isFirstgame=false;
483 nagagi=[];
484 isCheck=0;
485 replier.reply("새로운 참가자가 저에게 관적으로 '/7포커 참가'를 보내주시면 참가 처리
486 }
487 } else if (rule=="일반"){ //일반 베팅률
488     if (cmd=="/레이즈" && sender==beting[m] && room==R){ //레이즈
489         if (!isNaN(data)){
490             if ((data-Math.floor(data))==0){
491                 if ((Math.max.apply(null, pot)-pot[player.indexOf(sender)])<data && dat
492                     if (m<beting.length-player.length){
493                         if(allin.indexOf(sender)==-1){
494                             if (allin.length!=player.length-1){
495                                 data=Number(data);
496                                 if (data>=coin[player.indexOf(sender)]) { //레이즈 금액
497                                     var whole=coin[player.indexOf(sender)];
498                                     coin[player.indexOf(sender)]=0;
499                                     chips[player2.indexOf(sender)]=0;
500                                     money += whole;
501                                     pot[player.indexOf(sender)]+=whole;
502                                     pot2[player2.indexOf(sender)]+=whole;
503                                     side.push(money);
504                                     allin.push(sender);
505                                     replier.reply(sender+"님 올인! 사이드베팅이 적용됩니
506                                 } else { //아닐 경우
507                                     pot[player.indexOf(sender)]+=data;
508                                     pot2[player2.indexOf(sender)]+=data;
509                                     coin[player.indexOf(sender)]-=data;
510                                     chips[player2.indexOf(sender)]-=data;
511                                     money +=data;
512                                 }
513                                 call=0; //레이즈 시 콜 횟수가 초기화됩니다.
514                                 pc="";
515                                 for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+'
516                                 replier.reply("다음 "+beting[m+1]+"님 베팅해주세요.\n/콜
517                                 m+=1;
518                                 isCheck+=1;
519                             } else replier.reply("한 명을 제외한 모든 플레이어가 올인하여
520                             } else replier.reply("올인한 플레이어는 /체크, /콜, /폴드 만 가능
521                             } else replier.reply("레이즈 횟수 최대치 달성으로 /콜, /폴드 만 가능
522                                 //레이즈는 최대 2번(콜과 폴드를 하지 않아야) 가능합니다.
523                             } else replier.reply("베팅금액이 콜 이하이거나 POT MONEY를 초과하여 레이즈
524                             } else replier.reply("자연수만 입력해주세요.");
525                             } else replier.reply("숫자만 입력해주세요.");
526                         }
527     } else if (rule=="하우스룰" && sender==beting[m] && room==R){ //하우스룰
528         if (msg=="/뺑" && isCheck==0){ //뺑: 시드금액 베팅
529             if(allin.indexOf(sender)==-1){
530                 if (basicseed>=coin[player.indexOf(sender)]) { //레이즈 금액이 보유금액보다
531                     var whole=coin[player.indexOf(sender)];

```



```

600         pot2[player2.indexOf(sender)]+=just;
601         coin[player.indexOf(sender)]-=just;
602         chips[player2.indexOf(sender)]-=just;
603         money +=just;
604     }
605     call=0; //레이즈 시 콜 횟수가 초기화됩니다.
606     pc="";
607     for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+": "+pot[k]+'
608     replier.reply("다음 "+beting[m+1]+"님 베팅해주세요.\n"+btr2+"로 베팅
609     m+=1;
610     isCheck+=1;
611 }
612 else if (msg=="하프"){ //하프: 판돈의 절반 베팅
613     if (Math.round(money/2)>=coin[player.indexOf(sender)]) { //레이즈 금액
614         var whole=coin[player.indexOf(sender)];
615         just=whole;
616         coin[player.indexOf(sender)]=0;
617         chips[player2.indexOf(sender)]=0;
618         money += whole;
619         pot[player.indexOf(sender)]+=whole;
620         pot2[player2.indexOf(sender)]+=whole;
621         side.push(money);
622         allin.push(sender);
623         replier.reply(sender+"님 올인! 사이드베팅이 적용됩니다.");
624     } else { //아닐 경우
625         just=Math.round(money/2);
626         pot[player.indexOf(sender)]+=just;
627         pot2[player2.indexOf(sender)]+=just;
628         coin[player.indexOf(sender)]-=just;
629         chips[player2.indexOf(sender)]-=just;
630         money +=just;
631     }
632     call=0; //레이즈 시 콜 횟수가 초기화됩니다.
633     pc="";
634     for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+": "+pot[k]+'
635     replier.reply("다음 "+beting[m+1]+"님 베팅해주세요.\n"+btr2+"로 베팅
636     m+=1;
637     isCheck+=1;
638 }
639 } else replier.reply("한 명을 제외한 모든 플레이어가 올인하여 레이즈가 불가능
640 } else replier.reply("올인한 플레이어는 /체크, /콜, /폴드 만 가능합니다.");
641 } else replier.reply("레이즈 횟수 최대치 달성으로 /콜, /폴드 만 가능합니다.");
642 }
643 }
644
645 if (status<=6-N && isBet==2 && openlist[0].length==status+N-3){ //오픈 카드 수령
646     m=0;
647     for (var i=0;i<player.length; i++){
648         cardlist[i].push(cards[cards.length-1]);
649         openlist[i].push(cards[cards.length-1]);
650         cards.pop();
651     } // 공개카드 한 장씩 더 배분
652     var pc="추가 공개 카드를 배분합니다.";
653     for (var i=0;i<player.length;i++){
654         pc=pc+"\n"+player[i]+": "+openlist[i];
655         var ans="현재 나의 카드\n"+cardlist[i]+"\n현재 등급: "+outgrade(grade(cardlist[i]))[0];
656         replier.reply(player[i],sh(ans));
657     }
658     replier.reply(R,sh(pc));
659     var pc="";
660     for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+": "+pot[k]+"개 베팅/"+coin[k]+"개
661     beting=Beting(Boss(openlist));
662     replier.reply(Boss(openlist)+"님부터 베팅을 시작합니다.\n"+btr1+"로 베팅이 가능합니다.\n\
663     isBet=1; //베팅 다시 시작!
664 }
665
666 if (status==7-N && isBet==2 && openlist[0].length==4){ //히든 카드 수령
667     m=0;
668     for (var i=0;i<player.length; i++){

```

```

669         cardlist[i].push(cards[cards.length-1]);
670         cards.pop();
671     } // 히든카드 배분
672     replier.reply("마지막 히든 카드를 랜덤으로 배분합니다.");
673     var pc="현재 공개 카드"
674     for (var i=0;i<player.length;i++){
675         pc=pc+"\n"+player[i]+": "+openlist[i];
676         var ans="현재 나의 카드\n"+cardlist[i]+\n현재 등급: "+outgrade(grade(cardlist[i]))[0]
677         replier.reply(player[i],"히든 카드: "+sh(cardlist[i][cardlist[i].length-1]));
678         replier.reply(player[i],sh(ans));
679     }
680     replier.reply(R,sh(pc));
681     var pc="";
682     for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+": "+pot[k]+"개 베팅/"+coin[k]+"개"
683     replier.reply(Boss(openlist)+"님부터 베팅을 시작합니다.\n"+btr1+"로 베팅이 가능합니다.\n"
684     isBet=1; //베팅 다시 시작!
685 }
686
687 if (status==8-N && isBet==2){ //마지막 베팅이 끝나고 결과를 계산합니다.
688     var gl=[]; //플레이어별 족보 등급 리스트
689     var prize=[]; //플레이어별 획득할 수 있는 칩 개수
690     var winners=[]; //플레이어를 1등부터 순위대로 나열
691     var ult="각자의 카드를 공개합니다."; //각자의 카드 모두 공개
692     var Res="";
693     replier.reply("마지막 베팅이 종료되었습니다.");
694     for (var i=0; i<player.length; i++){
695         var gg=player[i]+": "+outgrade(grade(cardlist[i]))[0])+"\n";
696         Res=Res+gg;
697         gg=player[i]+":\n"+cardlist[i];
698         ult=ult+"\n"+gg;
699     } //족보 등급 출력
700     for (var i=0;i<player.length;i++) gl.push([grade(cardlist[i]),player[i]]);
701     gl.sort((a,b)=>b[0][2]-a[0][2]);
702     gl.sort((a,b)=>b[0][1]-a[0][1]);
703     gl.sort((a,b)=>b[0][0][1]-a[0][0][1]);
704     gl.sort((a,b)=>b[0][0][0]-a[0][0][0]); //등급을 순위대로 나열
705     for (var i=0;i<gl.length;i++) winners.push(gl[i][1]); //순위에 맞는 플레이어 나열
706     for (var i=0; i<winners.length; i++) prize.push(money);
707     if (allin.length>0) for (var i=0; i<allin.length; i++) prize[winners.indexOf(allin[i])]
708     //플레이어별 획득가능 칩 배열. 올인한 플레이어의 획득가능 칩은 사이드금액으로 저장됩니다.
709     for (var i=0;i<prize.length-1;i++){
710         if (prize[i]>=prize[i+1]) {
711             winners.splice(i+1,prize.length-i-1);
712             prize.splice(i+1,prize.length-i-1);
713             break;
714         }
715     } //상금을 받을 수 있는 플레이어들만 가려냅니다.
716     replier.reply(sh(ult));
717     replier.reply(sh(Res.trim()));
718
719     for (var i=0;i<winners.length;i++){
720         chips[player2.indexOf(winners[i])]+=prize[i];
721         for (var j=i;j<prize.length-1;j++) prize[j+1]-=prize[i];
722     } //상금 분배
723     pc="현재 칩 보유량";
724     for (var k=0; k<=player2.length-1;k++) pc=pc+"\n"+player2[k]+": "+chips[k]+"개";
725     replier.reply("승리자는 "+winners[0]+"님입니다!\n\n"+pc);
726     //결과 출력
727     if (nagagi.length!=0){
728         for (i=0;i<nagagi.length;i++){
729             player2.splice(player2.indexOf(nagagi[i]),1);
730             chips.splice(player2.indexOf(nagagi[i]),1);
731         }
732         replier.reply(nagagi+"님이 나가셨습니다.\n남은 참가자: "+player2);
733     } //나가기
734     Gstart=false;
735     coin=[]; player=[]; pot=[]; pot2=[];
736     cardlist=[]; openlist=[];
737     for ( var i=0; i<=player2.length-1;i++){

```

```

738         coin.push(chips[i]);
739         player.push(player2[i]);
740     }
741     for(var j=0; j<=player.length-1; j++){
742         pot.push(0);
743         pot2.push(0);
744         cardlist.push([]);
745     }
746     beting=[]; money=0; call=0; m=0; isBet=0; status=0;
747     x="";
748     ab=[];
749     isFirstgame=false;
750     nagagi=[];
751     isCheck=0;
752     allin=[]; side=[];
753     replier.reply("새로운 참가자가 저에게 랜톡으로 '/7포커 참가'를 보내주시면 참가 처리됩니다.");
754 }
755
756 if (msg=="/나가기" && player2.indexOf(sender)!=-1 && nagagi.indexOf(sender)==-1){
757     if (Gstart){ //게임이 진행중인 경우 나가가 예약됨
758         nagagi.push(sender);
759         replier.reply(R, sender+"님의 나가가 예약되었습니다. 폴드 시 혹은 한 게임이 끝날 시
760     } else { //게임이 진행 중이 아닌 경우 바로 나가짐
761         player.splice(player.indexOf(sender),1);
762         player2.splice(player2.indexOf(sender),1);
763         replier.reply(R, sender+"님이 나가셨습니다.\n남은 참가자: "+player2);
764     }
765 }
766
767 if (msg=="/나가기취소" && player2.indexOf(sender)!=-1 && nagagi.indexOf(sender)!=-1){
768     nagagi.splice(nagagi.indexOf(sender),1);
769     replier.reply(R, sender+"님 나가기 예약 취소");
770 } //나가기 예약이 되어 있었을 경우 취소
771
772 if (Gstart && player2.length==1){
773     replier.reply("참가자가 1명밖에 남지 않아 게임이 자동 종료됩니다.");
774     Api.compile("7포커.js");
775 } //모두 나가고 1명밖에 남지 않았을때 자동 게임 종료
776
777 if (msg=="/7포커 종료" && player2.indexOf(sender)!=-1){
778     turnoff=true;
779     onoff.push(sender);
780     replier.reply("/찬성 으로 종료 찬성이 가능합니다. 5초 이내 과반수 이상 찬성 시 게임이 종료!
781     java.lang.Thread.sleep(5000);
782     if (onoff.length>=player2.length/2){
783         replier.reply(onoff.length+"명의 찬성으로 게임이 종료됩니다.");
784         Api.compile("7포커.js");
785     } else {
786         replier.reply("과반수가 넘지 않아 게임이 계속 진행됩니다.");
787         onoff=[];
788         turnoff=false;
789     }
790 } //게임 종료: 투표제. 5초 이내 과반수 이상 찬성 시 자동 종료. 종료 명령어를 친 사람은 자동 찬성
791
792 if (turnoff && onoff.indexOf(sender)==-1 && msg=="/찬성"){
793     onoff.push(sender);
794     replier.reply(sender+"님 찬성");
795 } //찬성
796 }

```


즐거




seori님의 게시글 더보기 >

댓글 등록순 최신순 🔄

댓글알림 ☐

 **AlphaDo**
아이고...고여버렸네
2020.02.11. 23:09 답글쓰기

 **seori** 작성자
증발하신분이..
2020.02.12. 01:25 답글쓰기

Hibot

댓글을 남겨보세요



등록

글쓰기

답글

목록

▲ TOP

'| 일반 소스 공유 |' 게시판 글


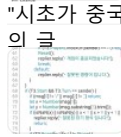
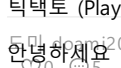
이 게시판 새글 구독하기 ☐

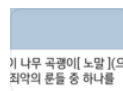
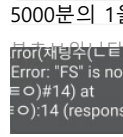
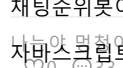
못하는 계산이 없는 올프람알파 파싱	seori	2020.02.12.
추억의 자리바꾸기 소스 (Pi,트위치봇 공동) 🤖 [10]	JS카톡챗봇	2020.02.11.
(공유) 포커[7포커] v1.2 by seori : 완성! 📎 [2]	seori	2020.02.11.
(공유) 포커[텍사스 홀덤] v2.2 by seori 📎	seori	2020.02.11.
기타 소스 🤖 [4]	Americium	2020.02.11.


1 2 3

전체보기

이 카페 인기글


"시초가 중국발" 발언 관련 해명글 및 부탁의 글

틱택토 (Player vs Player)

도만 dman 2005
안녕하세요
KHP
0 0 6


5000분의 1을 얻은 사람

채팅순위봇이 갑자기 안되네요..ㅠ

자바스크립트
도유니
0 0 10

자동학습 봇 특정 말만 무시하기?
해히
0 0 4
안녕하세요!
트위터봇
0 0 6

태양, 달 정보 구현 완료

1 2 3 4 5