| 일반 소스 공유 | >

(공유) 포커[텍사스 홀덤] v1.8 by seori : 완성!!

seori 챗봇 마스터 1:1 채팅
2020.02.07. 04:04 조회 55

첨부파일 1

텍사스 홀덤! 드디어 완성하였습니다!

(업데이트 내역)
v1.5 도움말을 추가하고 각종 오류들을 수정하였습니다.
v1.6 첫 배팅의 레이즈 최대 횟수를 줄였습니다.
v1.7 올인/사이드베팅 기능을 완성하였습니다.
v1.8 필요없는 스크립트들을 삭제하였습니다. (100줄 이상 줄어들었습니다. 나름 최적화..?)
주석을 달았습니다.
포커를 홀덤으로 다시 바꾸었고 일부 오타가 수정되었습니다.

```
const scriptName="홀덤.js"
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 1
//카드들입니다. s: 스페이드, d: 다이아몬드, h: 하트, c: 클로버. 숫자는 A가 등급이 가장 높으므로 14
var poker=false; var Gstart=false;
var player=[]; var coin=[]; var pot=[]; //선수의 게임 진행 정보
var player2=[]; var chips=[]; var pot2=[]; //폴드한 선수의 정보 저장
var card1; var card2; var cardlist=[]; var common=[]; //카드관련
var basicchips=100; var basicseed=1; //시드관련
var isBet=false; var beting=[]; var isCheck=0; var status=0; var money=0; var call=0; var m=0;
var pc="";
var x="";
var R;
var isFirstgame=true;
var allsee="\u200b".repeat(500);
var nagagi=[];
var preChat=[];
var turnoff=false; var onoff=[]; //종료관련
var allin=[]; var side=[]; //올인관련

function shuffle(a) { //배열을 섞는 함수. 카드 셔플에 이용.
   var j, x, i;
   for (i = a.length - 1; i > 0; i--) {
      j = Math.floor(Math.random() * (i + 1));
      x = a[i]; a[i] = a[j]; a[j] = x;
    }
    return a;
}

function seol(moyang){ //게임이 끝나고 '카드모양'에 등급을 매길 때 이용됩니다.
    return moyang.replace(/s/g,3).replace(/d/g,2).replace(/h/g,1).replace(/c/g,0);
}

function lee(sutja){ //족보 등급 계산을 위한 '숫자' 등급을 매기는 함수입니다.
    var isStraight=0; var sut=[]; var ja=[];

    for (var i=0;i<sutja.length;i++){
        sutja[i]=Number(sutja[i]);
```

```
        sut.push(Number(sutja[i]));
        ja.push(Number(sutja[i]));
    }
    for (var i=0;i<sutja.filter(e => e == 14).length;i++){
        sutja.push(1); sut.push(1); ja.push(1);
    } //추후 백스트레이트 계산을 위해 배열에 A(14)가 포함된 경우 1을 추가해줍니다.

    for (var i=14;i>1;i--) if (sutja.filter(e => e == i).length==4) return [9, i]; //포카드

    for (var i=14;i>1;i--){
        if (sutja.filter(e => e == i).length==3){
            for (k=0;k<3;k++) sut.splice(sut.indexOf(i),1);
            for (var j=14;j>1;j--) if (sut.filter(e => e == j).length>=2) return [8,i+0.01*j];
        }
    } //풀하우스

    for (var i=14;i>9;i--){
        if (sutja.filter(e => e == i).length>=1){
            isStraight+=1;
            if (isStraight==5) return [6,0];
        } else {
            isStraight=0;
            break;
        }
    } //마운틴

    for (var i=5;i>0;i--){
        if (sutja.filter(e => e == i).length>=1){
            isStraight+=1;
            if (isStraight==5) return [5,0];
        } else {
            isStraight=0;
            break;
        }
    } //백스트레이트

    for (var p=13; p>5; p--){
        for (var i=0;i<5;i++){
            if (sutja.filter(e => e == (p-i)).length >=1 ){
                isStraight+=1;
                if (isStraight==5) return [4,p];
            } else break;
        }isStraight=0;
    } //스트레이트

    for (var i=14;i>1;i--){ if (sutja.filter(e => e == i).length==3) return [3,i]; } //트리플

    for (var i=14;i>1;i--){
        if (sutja.filter(e => e == i).length==2){
            for (k=0;k<2;k++) ja.splice(ja.indexOf(i),1);
            for (var j=i-1;j>1;j--) if (ja.filter(e => e == j).length==2) return [2,i+0.01*j];
            return [1,i]; //원페어
        }
    }

    var ans=[0,Math.max.apply(null, sutja)];
    return ans; //탑(노페어)
}

function grade(CARD){ //모양 등급인 '플러시'의 경우를 생각하여 족보의 진짜 등급을 계산합니다.
    var Flushshape; var shape=[]; var isFlush=false;
    var num=[]; var card=[]; var Realgrade=[[]];

    for (var i=0; i<CARD.length; i++){
        shape.push(seol(CARD[i].split(" ")[0]));
        num.push(CARD[i].split(" ")[1]);
        card.push(Number(num[i])*10+Number(shape[i]));
    }
```

```
    for (var i=0; i<4; i++){
        if (shape.filter(e => e == i).length>=5){
            isFlush=true;
            Flushshape=i;
            break;
        }
    } //족보가 플러시인지 아닌지 판별합니다.

    if (isFlush==true){ //플러시일 경우
        for(var i=0;i<shape.length; i++){
            if (shape[i]!=Flushshape){
                card.splice(i,1);
                num.splice(i,1);
            }
        }
        if (lee(num)[0]==6) Realgrade[0].push(12); //로티플
        else if (lee(num)[0]==5) Realgrade[0].push(11); //백티플
        else if (lee(num)[0]==4) Realgrade[0].push(10); //스티플
        else Realgrade[0].push(7); //플러시
    } else Realgrade[0].push(lee(num)[0]); //플러시가 아닐 경우 숫자 등급을 그대로 반환합니다.

    num.sort((a,b)=>b-a);
    var nnum=0;
    card.sort((a,b)=>b-a);
    var sshape=0;
    for (var i=0;i<num.length;i++){
        nnum+=Math.pow(0.01,i)*num[i];
        sshape+=Math.pow(0.001,i)*card[i];
    }
    Realgrade[0].push(lee(num)[1]);
    Realgrade.push(nnum);
    Realgrade.push(sshape);
    return Realgrade;
    /* [[족보 등급, 등급에 해당하는 숫자], 하이 카드, 하이 카드에 따른 모양 등급]으로 전체 등급을 반
    등급이 무승부가 날 시 우승자를 판별하기 위함입니다.*/
}

function sh(shs){ //내용 출력을 위한 함수입니다.
    return shs.replace(/s /g,"♠").replace(/d /g,"♦").replace(/h /g,"❤").replace(/c /g,"♣").
}

function outgrade(put){ //족보 등급 출력을 위한 함수입니다.
    var first=Number(put[1]).toFixed(0);
    var second=Math.round((Number(put[1])-first)*100);
    var third=Number(put[0]).toFixed(0).replace("12", "로티플").replace("11", "백티플").replace

    var ans=first+", "+second+" "+third;

    if (first==0) return third;
    else if (second==0) return first+" "+third;
    else return ans;
}

function response(room, msg, sender, isGroupChat, replier, ImageDB, packageName, threadId){

    if (preChat[room]==msg) return 0;  //도배(따닥챗)방지
    preChat[room]=msg;

    var cmd=msg.split(" ")[0];
    var data=msg.replace(cmd+" ","");
    if (msg=="/홀덤") replier.reply("텍사스 홀덤 도움말\n(명령어, 룰 설명)"+allsee+"\n\n[명령어]\
    if (msg=="/홀덤 생성" && isGroupChat==true){
        if (poker==true) replier.reply("이미 생성된 방이 있습니다.");
        else {
            poker=true;
            R=room;
            replier.reply("방이 생성되었습니다.\n저에게 갠톡으로 '/홀덤 참가'라고 보내주시면 참가가
        }
    } //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                 } 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.cell(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("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==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 섞섞
242                     replier.reply("게임을 시작합니다.");
243                     for(var i=0; i<=player.length-1; i++){
244                         card1=cards[cards.length-1];
```

```
                        cards.pop();
                        card2=cards[cards.length-1];
                        cards.pop();
                        cardlist[i].push(card1);
                        cardlist[i].push(card2);
                        var ans="핸드 카드를 받았습니다.\n"+cardlist[i]+"\n현재 등급: "+outgrade
                        replier.reply(player[i],sh(ans));
                    } //처음에 받게 되는 핸드 카드를 받습니다.
                    java.lang.Thread.sleep(1000);
                    replier.reply("처음 기본 칩은 "+basicchips+"개입니다. "+basicseed+"/"+basics
                    coin[0]-=basicseed;
                    coin[1]-=basicseed*2;
                    chips[0]-=basicseed;
                    chips[1]-=basicseed*2;
                    money=basicseed*3;
                    pot[0]=basicseed;
                    pot2[0]=basicseed;
                    pot[1]=basicseed*2;
                    pot2[1]=basicseed*2;
                    //SB와 BB는 시드금액을 지불합니다.

                    for (var i=0;i<2*player.length;i++) beting.push(player[i%player.length]);
                    beting.push(player[0]);
                    beting.push(player[1]);
                    beting.splice(0,2);
                    //첫 베팅의 순서:SB의 좌측부터 시작
                    pc="";
                    for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+": "+pot[k]+"개 베팅/
                    replier.reply(beting[m]+"님부터 베팅을 시작합니다.\n/콜, /폴드, /레이즈 (숫자
                    isBet=true; //베팅시작
                } else replier.reply("이미 게임이 시작되었습니다.");
            } else replier.reply("아직 방이 만들어지지 않았습니다.");
        } else replier.reply("최소 참가자는 2명입니다.");
    }

    if (isFirstgame==false && cmd=="/업" && isNaN(data)==false){ //시드 올릴까
       data=Number(data);
        basicseed+=Math.floor(data);
        if (basicseed/basicchips>0.05){
            basicseed=Math.floor(basicchips*0.05);
            replier.reply("최고시드("+basicseed+"개)로 자동 설정되었습니다.");
            isFirstgame=true;
        } else {
            replier.reply("시드가 "+data+"개 올라 "+basicseed+"개가 되었습니다.");
            isFirstgame=true;
        }
    } else if (isFirstgame==false && msg=="/킵") isFirstgame=true; //올리지 말까

    if (isBet==true){ //베팅 스크립트
        if (msg == "/체크"){ //체크
            if (isCheck==0 && sender==beting[m] && room==R && status>=1){
                replier.reply("체크! 다음 "+beting[m+1]+"님 베팅해주세요.\n/콜, /폴드, /레이즈 (숫
                m+=1;
              isCheck+=1;
            }
        } else if (msg=="/콜" && sender==beting[m] && room==R){ //콜
            if (status==0 || isCheck!=0){

                if (allin.indexOf(sender)==-1 && (Math.max.apply(null, pot)-pot[player.indexOf(
                    var whole=coin[player.indexOf(sender)];
                    coin[player.indexOf(sender)]==0;
                    chips[player2.indexOf(sender)]==0;
                    money += whole;
                    pot[player.indexOf(sender)]+=whole;
                    pot2[player2.indexOf(sender)]+=whole;
                    side.push(money);
                    allin.push(sender);
                    replier.reply(sender+"님 올인! 사이드베팅이 적용됩니다."); //콜 금액이 보유금
                } else if (allin.indexOf(sender)!=-1) replier.reply(sender+"님은 올인하셨으므로
```

```
                else {
                    coin[player.indexOf(sender)]=coin[player.indexOf(sender)]-Math.max.apply(nu
                    chips[player2.indexOf(sender)]=chips[player2.indexOf(sender)]-Math.max.appl
                    money = money+ Math.max.apply(null, pot)-pot[player.indexOf(sender)];
                    pot[player.indexOf(sender)]=Math.max.apply(null, pot);
                    pot2[player2.indexOf(sender)]=Math.max.apply(null, pot2);
                }
                pc="";
                for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+": "+pot[k]+"개 베팅/"+co
                pc=pc.trim();
                call +=1; //콜 횟수

                if (call==player.length-1){ //콜을 마지막으로 베팅 완료
                    replier.reply("베팅이 완료되었습니다.\n\nPOT: "+money+"개\n\n"+pc.trim());
                    isBet=false;
                    status+=1;
                    call=0;
                 isCheck=0;
                } else { //베팅이 완료되지 않았을 시 다음 주자로 넘어갑니다.
                    replier.reply("다음 "+beting[m+1]+"님 베팅해주세요.\n/콜, /폴드, /레이즈 (숫
                    m+=1;
                 isCheck+=1;
                }
            }
        } else if (msg=="/폴드" && sender==beting[m] && room==R){ //폴드
            if (player.length != 2) { //플레이어가 2인 이상일때
                coin.splice(player.indexOf(sender),1);
                pot.splice(player.indexOf(sender),1);
                cardlist.splice(player.indexOf(sender),1);
                player.splice(player.indexOf(sender),1);
                if (allin.indexOf(sender)!=-1){
                    side.splice(allin.indexOf(sender),1);
                    allin.splice(allin.indexOf(sender),1);
                }
                for (var i=0;i<=2;i++) beting.splice(beting.indexOf(sender),1);
                pc="";
                for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+": "+pot[k]+"개 베팅/"+co
                replier.reply(sender+"님 폴드(다이)하셨습니다.");

                if (nagagi.indexOf(sender)!=-1){
                    chips.splice(player2.indexOf(sender),1);
                   pot2.splice(player2.indexOf(sender),1);
                    player2.splice(player2.indexOf(sender),1);

                    replier.reply(sender+"님이 나가셨습니다.\n남은 참가자: "+player2);
                   nagagi.splice(nagagi.indexOf(sender),1);
                } //폴드 시 게임에서 완전 나가집니다.

                if (call==player.length-1){ //폴드를 마지막으로 베팅이 완료될 경우
                    replier.reply("베팅이 완료되었습니다.\n\nPOT: "+money+"개\n\n"+pc.trim());
                    isBet=false;
                    status+=1;
                    call=0;
                 isCheck=0;
                } else { //베팅이 완료되지 않았을 경우
                    m=m-Math.floor(m/(player.length+1));
                    if (isCheck==0) replier.reply("다음 "+beting[m]+"님 베팅해주세요.\n/체크, /폴
                    //첫 주자가 폴드할 경우
                    else replier.reply("다음 "+beting[m]+"님 베팅해주세요.\n/콜, /폴드, /레이즈 (
                    //아닐 경우
                }
            } else { //플레이어가 2명이었을 경우. 폴드 시 남은 1명이 자동 승리하게 됩니다.
                player.splice(player.indexOf(sender),1);
                chips[player2.indexOf(player[0])]+=money;
                pc="현재 칩 보유량";
                for (var k=0; k<=player2.length-1;k++) pc=pc+"\n"+player2[k]+": "+chips[k]+"개"
                replier.reply("나머지 선수가 모두 폴드하여 "+player+"님 승리입니다!\n\n"+pc);
                if (nagagi.length!=0){
                    for (i=0;i<nagagi.length;i++){
```

```
                for (i=0;i<nagagi.length;i++){
                    player2.splice(player2.indexOf(nagagi[i]),1);
                    chips.splice(player2.indexOf(nagagi[i]),1);
                }
                replier.reply(nagagi+"님이 나가셨습니다.\n남은 참가자: "+player2);
            }
            Gstart=false;
            player=[]; coin=[]; pot=[]; pot2=[];
            cardlist=[]; common=[];
            allin=[]; side=[];
            player2.unshift(player2[player2.length-1]);
            chips.unshift(chips[chips.length-1]);
            player2.pop();
            chips.pop();
            //게임이 끝날 때마다 자리가 한칸씩 밀려나 SB와 BB가 바뀝니다.
            for ( var i=0; i<=player2.length-1;i++){
                coin.push(chips[i]);
                player.push(player2[i]);
            }
            for(var j=0; j<=player.length-1; j++){
                pot.push(0);
                pot2.push(0);
                cardlist.push([]);
            }
            beting=[]; money=0; call=0; m=0; isBet=false; status=0;
            x="";
            isFirstgame=false;
            nagagi=[];
          isCheck=0;
            replier.reply("새로운 참가자가 저에게 갠톡으로 '/홀덤 참가'를 보내주시면 참가 처리
        }
    } else if (cmd=="/레이즈" && sender==beting[m] && room==R){ //레이즈
        if (isNaN(data) == false){
            if ((data-Math.floor(data))==0){
                if ((Math.max.apply(null, pot)-pot[player.indexOf(sender)])<data && data<=(
                    if (m<beting.length-player.length){
                        if(allin.indexOf(sender)==-1){
                            data=Number(data);
                            if (data>=coin[player.indexOf(sender)]) { //레이즈 금액이 보유금
                                var whole=coin[player.indexOf(sender)];
                                coin[player.indexOf(sender)]==0;
                                chips[player2.indexOf(sender)]==0;
                                money += whole;
                                pot[player.indexOf(sender)]+=whole;
                                pot2[player2.indexOf(sender)]+=whole;
                                side.push(money);
                                allin.push(sender);
                                replier.reply(sender+"님 올인! 사이드베팅이 적용됩니다.");
                            } else { //아닐 경우
                                pot[player.indexOf(sender)]+=data;
                                pot2[player2.indexOf(sender)]+=data;
                                coin[player.indexOf(sender)]-=data;
                                chips[player2.indexOf(sender)]-=data;
                                money +=data;
                            }
                            call=0; //레이즈 시 콜 횟수가 초기화됩니다.
                            pc="";
                            for (var k=0; k<=player.length-1;k++) pc=pc+player[k]+": "+pot[
                            replier.reply("다음 "+beting[m+1]+"님 베팅해주세요.\n/콜, /폴드,
                            m+=1;
                            isCheck+=1;
                        } else replier.reply("올인한 플레이어는 /콜, /폴드 만 가능합니다.");
                    } else replier.reply("레이즈 횟수 최대치 달성으로 /콜, /폴드 만 가능합니다
                      //레이즈는 최대 2번(콜과 폴드를 하지 않아야) 가능합니다.
                    } else replier.reply("베팅금액이 콜 이하이거나 레이즈 최대치("+(2*Math.max.ap
                } else replier.reply("자연수만 입력해주세요.");
            } else replier.reply("숫자만 입력해주세요.");
        }
    }
```

```
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 이상의 베팅에서는 SB부터 베팅을 하도록 순서가 변경됩니다.
462         var ans="공통 카드 3장을 공개합니다.\n"+common; //공통 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]+"\n현재 등급: "+outgrade(grade(cardlist[i])[0
466             replier.reply(player[i],sh(ans));
467         } //공통 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]+"\n현재 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
```

```
        for (var i=0;i<winners.length;i++){
            chips[player2.indexOf(winners[i])]+=prize[i];
            for (var j=i;j<prize.length-1;j++) prize[j+1]-=prize[i];
        } //상금 분배
        pc="현재 칩 보유량";
        for (var k=0; k<=player2.length-1;k++) pc=pc+"\n"+player2[k]+": "+chips[k]+"개";
        replier.reply("승리자는 "+winner+"님입니다!\n\n"+pc);
        //결과 출력
        if (nagagi.length!=0){
            for (i=0;i<nagagi.length;i++){
                player2.splice(player2.indexOf(nagagi[i]),1);
                chips.splice(player2.indexOf(nagagi[i]),1);
            }
            replier.reply(nagagi+"님이 나가셨습니다.\n남은 참가자: "+player2);
        } //나가기
        Gstart=false;
        coin=[]; player=[]; pot=[]; pot2=[];
        cardlist=[]; common=[];
        player2.unshift(player2[player2.length-1]);
        chips.unshift(chips[chips.length-1]);
        player2.pop();
        chips.pop(); //자리가 바뀝니다
        for ( var i=0; i<=player2.length-1;i++){
            coin.push(chips[i]);
            player.push(player2[i]);
        }
        for(var j=0; j<=player.length-1; j++){
            pot.push(0);
            pot2.push(0);
            cardlist.push([]);
        }
        beting=[]; money=0; call=0; m=0; isBet=false; status=0;
        x="";
        isFirstgame=false;
        nagagi=[];
       isCheck=0;
        allin=[]; side=[];
        replier.reply("새로운 참가자가 저에게 갠톡으로 '/홀덤 참가'를 보내주시면 참가 처리됩니다.\n
    }

    if (msg=="/나가기" && player2.indexOf(sender)!=-1 && nagagi.indexOf(sender)==-1){
        if (Gstart==true){ //게임이 진행중인 경우 나가기가 예약됨
            nagagi.push(sender);
            replier.reply(R, sender+"님의 나가기가 예약되었습니다. 폴드 시 혹은 한 게임이 끝날 시
        } else { //게임이 진행 중이 아닌 경우 바로 나가짐
            player.splice(player.indexOf(sender),1);
            player2.splice(player2.indexOf(sender),1);
            replier.reply(R, sender+"님이 나가셨습니다.\n남은 참가자: "+player2);
        }
    }

    if (msg=="/나가기취소" && player2.indexOf(sender)!=-1 && nagagi.indexOf(sender)!=-1){
        nagagi.splice(nagagi.indexOf(sender),1);
        replier.reply(R, sender+"님 나가기 예약 취소");
    } //나가기 예약이 되어 있었을 경우 취소

    if (Gstart==true && player2.length==1){
        replier.reply("참가자가 1명밖에 남지 않아 게임이 자동 종료됩니다.");
        Api.compile("홀덤.js");
    } //모두 나가고 1명밖에 남지 않았을때 자동 게임 종료

    if (msg=="/홀덤 종료" && player2.indexOf(sender)!=-1){
        turnoff=true;
        onoff.push(sender);
        replier.reply("/찬성 으로 종료 찬성이 가능합니다. 5초 이내 과반수 이상 찬성 시 게임이 종료
        java.lang.Thread.sleep(5000);
        if (onoff.length>=player2.length/2){
            replier.reply(onoff.length+"명의 찬성으로 게임이 종료됩니다.");
            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
600 }
```

이제 7포커를 만들 준비가 되었습니다.

각종 오류 지적해주시면 정말 감사하겠습니다.
질문은 댓글로

seori님의 게시글 더보기 >

좋아요 1 댓글 4 공유 | 신고

댓글 등록순 최신순  댓글알림

카톡우앙
고생추~~
2020.02.07. 05:57 답글쓰기

seori 작성자

2020.02.07. 06:07 답글쓰기

새름
멋지셔용!! 수고하셨어요
2020.02.07. 07:17 답글쓰기

seori 작성자
끄악 몇시간도 안돼서 오류발견
2020.02.07. 09:49 답글쓰기

Hibot
댓글을 남겨보세요

등록

글쓰기 답글 목록 ▲ TOP

'| 일반 소스 공유 |' 게시판 글

이 게시판 새글 구독하기

운세 날짜 추가 소스요 ㅜ [3]	bidforpower23	2020.02.07.
더이상 (은)는, (이)가, (을)를을 할 필요가 없습니다. [15]	Violet	2020.02.07.
(공유) 포커[텍사스 홀덤] v1.8 by seori : 완성!! [4]	seori	2020.02.07.
제 첫 소스에요!! [19]	FFaker	2020.02.06.
jsoup를 이용한 두번째 소스 [2]	deadmau5	2020.02.06.

1 2 3

전체보기

이 카페 인기글

안녕하세요

ㅇㄷ

♡0 💬5

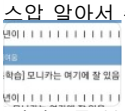
네이버에서 지역별 날씨 정보 크롤링하기 / 스압 알아서 주의

틱택토 (Player vs Player)

안녕하세요

tomohong

♡1 💬3



[카카오톡 봇] 양산형 자동학습 봇 소스 공유

틱택토 (카카오링크 적용)

혹시 여기서 변수를 저장하는 부분이 어딘지

도마 dosmi2005

타 메신저 사용시 무한반복

♡0 💬5

NaN

♡0 💬1

자동학습 봇 특정 말만 무시하기?

헤히

♡0 💬4

1 2 3 4 5