トランプゲームを作ってみよう２ (JavaScript 編)

文責　：　高知大学名誉教授　中村　治

ここでは二人でプレイするカード・ゲーム「ボヘミアン・シュナイダー」を Javascript 作ってみます。

ボヘミアン・シュナイダーというチェコ生まれのゲームは、風変わりなトリックの取り方に特徴があります。松田道弘著「トランプゲーム辞典」東京堂出版に載っています。

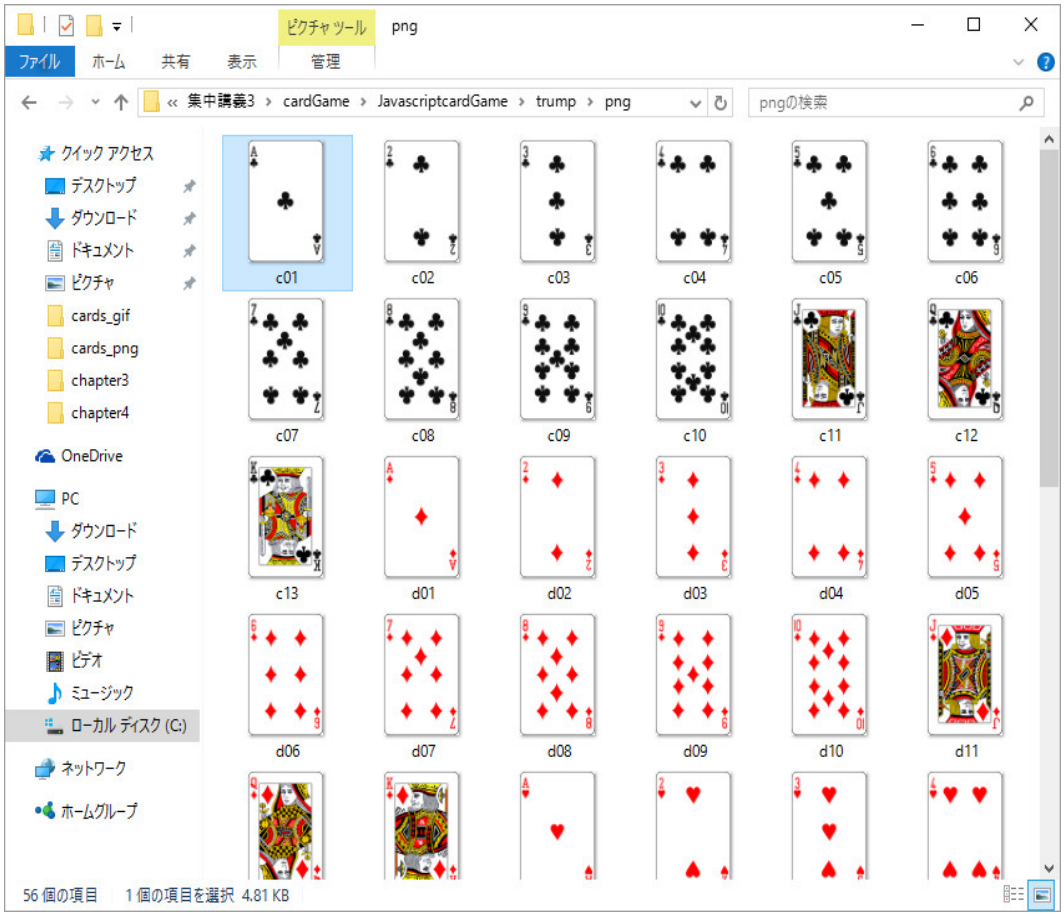
ルールは

1. 人数は 2 人
2. ２，３，４，５，６を除いた３２枚のカードを使用する
3. カードのランクは A, K, Q, J, 10, 9, 8, 7, です
4. 手札は各人 6 枚で、残りはストックとしてテーブルに伏せておきます。
5. ディーラーでない人がオープニング・リードをします。
6. リードされたスーツに関係なく、どのカードを出しても良いです。
7. リードされた方は、スーツに関係なく、リードされたカードのランクより一つ上のランクのカードを出せばそのトリックを取ることが出来ます。例えば、J がリードされたとすると、Q を出さない限り、このトリックを取れません。K や A ではこれません。相手がリードされたカードのランクより一つ上のランクのカードを出さなければ、リードした人がそのトリックを取ります。
8. トリックを取った人は、ストックのトップ・カードを取って手札に入れ、相手はその下のカードを取ります。
9. 各プレーヤーの手札は 6 枚に戻りました。
10. トリックを取った人がリードして、このトリックの勝者がストックのカードを取り、相手がその下のカードを取ります。
11. この手順をストックが無くなるまで続け、その後は手札が無くなるまでトリック争奪のプレイを続けます。
12. プレイが終わると取得トリックに含まれているカードの点数を数えてスコアを付けます。
13. カードの点数は A-１１点、K-４点、Q-３点、J-２点、10-１０点 、9、 8、 7 は０点です。

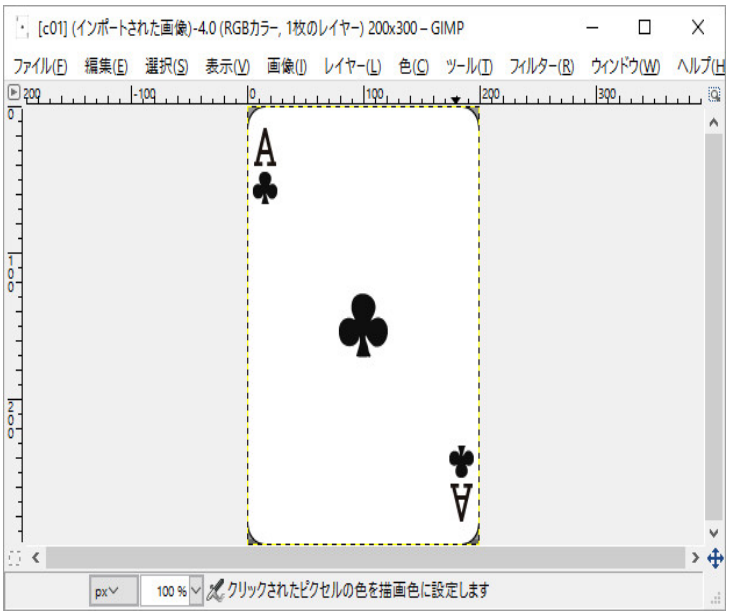
です。

この「ボヘミアン・シュナイダー」のプログラムを JavaScript で作成し、コンピュータと対戦できるようにします。

まずインターネットで無料素材のトランプの画像（.png）を手に入れます。



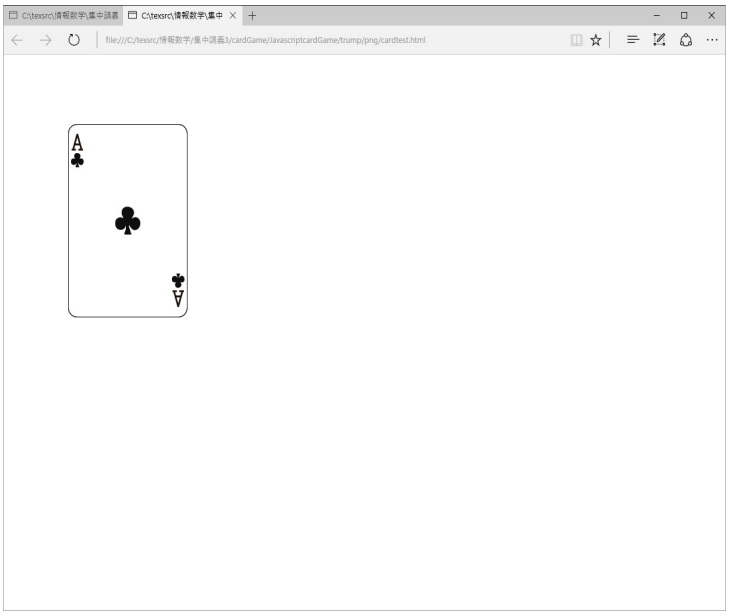
今回手に入れた画像は GIMP で調べると



200 × 300 ピクセルです。

```
<html>
<head>
<script>
var ctx;
function init() {
    var canvas = document.getElementById("canvas");
    ctx = canvas.getContext("2d");
    var card = document.getElementById("c01");
    ctx.drawImage(card, 100, 100);
}
</script>
</head>
<body onload="init()">
<canvas id="canvas" width="400" height="400"></canvas>
<img id="c1" src="c01.png" style="display:none" />
</body>
</html>
```

で、、カードを表示できます。

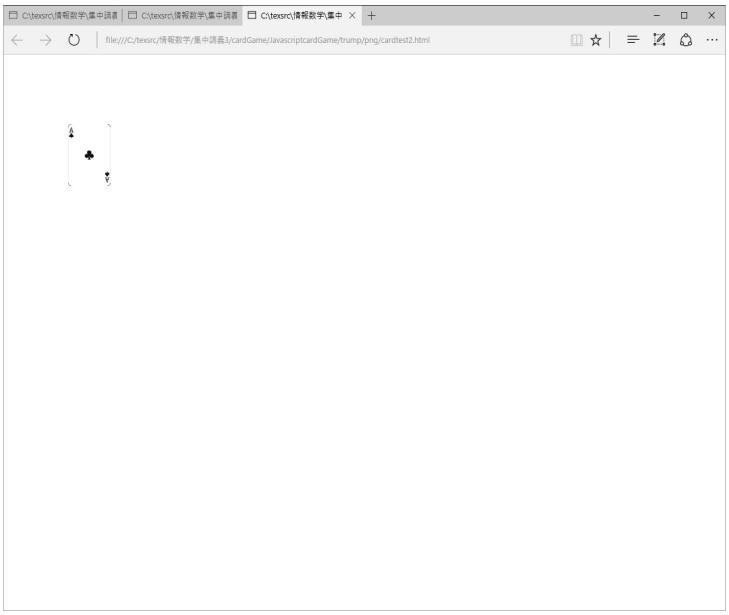


となります。

```
<html>
<head>
<script>
var ctx;
function init() {
    var canvas = document.getElementById("canvas");
```

```
    ctx = canvas.getContext("2d");
    var card = new Image();
    card.src = "c01.png";
    ctx.drawImage(card, 0, 0, 200, 300, 100, 100, 71, 96);
}
</script>
</head>
<body onload="init()">
<canvas id="canvas" width="400" height="300"></canvas>
</body>
</html>
```

とすれば、



を描きます。但し、Google Chrome では表示できません。色々なブラウザで表示できるように、プログラムが長くなりますが、前者のプログラムを使うことにします。JavaScript のデバッグの為に Google Chrome では表示できなければ困ります。

VC++ の時のように、拡大・縮小など細かな細工も出来ます。カードの画像を Canvas の任意の位置に表示する方法が分かったので、プログラムが作れます。

トランプゲームを作ってみよう (JavaScript 編) の「ジャーマン・ホイスト」のプログラミングを学習していると仮定して、その変更点だけ説明します。

プログラムの最終形は次のようになります。

```
<html>
<head>
<script>
var ctx;
var carddeck = [];
```

```
var comHand = [];
var userHand = [];
var Yama = [];
var getCom = [];
var getUser = [];
var dispCom = [];
var gameOverP = false;
var comPlayP = false;

function Card(suit, rank, value, image, backimage) {
    this.suit = suit;
    this.rank = rank;
    this.value = value;
    this.image = image;
    this.backimage = backimage;
}

function drawCard(card, x, y, flag) {
    if (flag == true) {
        ctx.drawImage(card.image, 0, 0, 200, 300, x, y, 71, 96);
    } else {
        ctx.drawImage(card.backimage, 0, 0, 200, 300, x, y, 71, 96);
    }
    ctx.strokeStyle = "blue";
    ctx.lineWidth = 1;
    ctx.strokeRect(x, y, 71, 96);
}

function cardSet() {
    var backimage = document.getElementById("z1");
    carddeck = [];
    for (var s = 0; s < 4; s++) {
        for (var r = 0; r < 8; r++) {
            var name = "";
            switch (s) {
                case 0: name = "c"; break;
                case 1: name = "d"; break;
                case 2: name = "h"; break;
                case 3: name = "s"; break;
            }
            var rank, value;
            switch (r) {
                case 0: rank = 14; value = 11; name +=  1; break;
```

```
                case 1: rank = 13; value =  4; name += 13; break;
                case 2: rank = 12; value =  3; name += 12; break;
                case 3: rank = 11; value =  2; name += 11; break;
                case 4: rank = 10; value = 10; name += 10; break;
                case 5: rank =  9; value =  0; name +=  9; break;
                case 6: rank =  8; value =  0; name +=  8; break;
                case 7: rank =  7; value =  0; name +=  7; break;
            }
            var image = document.getElementById(name);
            var card = new Card(s, rank, value, image, backimage);
            carddeck.push(card);
        }
    }
}

function cardInitialize() {
    i = carddeck.length
    while (i > 0) {
        var s = Math.floor(Math.random() * i);
        var temp = carddeck[--i]
        carddeck[i] = carddeck[s]
        carddeck[s] = temp
    }
}

function gameInitialize() {
    comHand = []
    userHand = []
    Yama = []

    cardInitialize()
    var index = 0
    for (var i = 0; i < 6; i++) {
        comHand.push(carddeck[index]);
        index++;
    }
    for (var i = 0; i < 6; i++) {
        userHand.push(carddeck[index]);
        index++;
    }
    for (var i = 0; i < 20; i++) {
        Yama.push(carddeck[index]);
        index++;
```

```
    }
    getCom = [];
    getUser = [];
    dispCom = [];
    gameOverP = false;
    comPlayP = false;
}

function ShowBan() {
    ctx.clearRect(0, 0, 1000, 600);
    var pos = 10;
    for (var i = 0; i < comHand.length; i++) {
        drawCard(comHand[i], pos, 10, false);
        pos += 72;
    }
    pos = 10;
    for (var i = 0; i < getCom.length; i++) {
        drawCard(getCom[i], pos, 110, true);
        pos += 20;
    }
    pos = 10;
    if (Yama.length > 0) {
        drawCard(Yama[Yama.length-1], pos, 210, false);
    }
    // pos = 145;
    pos = 245;
    if (dispCom.length > 0) {
        drawCard(dispCom[0], pos, 210, true);
    }
    pos = 10;
    for (var i = 0; i < getUser.length; i++) {
        drawCard(getUser[i], pos, 310, true);
         pos += 20;
    }
    pos = 10;
    for (var i = 0; i < userHand.length; i++) {
        drawCard(userHand[i], pos, 410, true);
        pos += 72;
    }
    ctx.font = "24px 'Times New Roman'";
    ctx.strokeStyle = "blue";
    ctx.fillStyle = "blue";
    if (gameOverP == false) {
```

```
        ctx.fillText("カードを選んで下さい", 400, 265);
    } else {
        var comValue = 0;
        for (var i = 0; i < getCom.length; i++) {
            comValue += getCom[i].value;
        }
        var userValue = 0;
        for (var i = 0; i < getUser.length; i++) {
            userValue += getUser[i].value;
        }
        var drawString = "Game Over ";
        drawString += userValue;
        drawString += " : ";
        drawString += comValue;
        if (comValue > userValue) {
            drawString += " コンピュータの勝ちです。頑張ってね！";
            ctx.strokeText(drawString, 200, 265);
        } else if (comValue < userValue) {
            drawString += " あなたの勝ちです。強いですね！";
            ctx.strokeText(drawString,200, 265);
        } else {
            drawString += "  引き分けです。もう一度しませんか？";
            ctx.strokeText(drawString, 200, 265);
        }
        drawString = "もう一度ゲームをするためには「ゲーム再スタート」のボタンをクリッ
クしてください";
        ctx.strokeText(drawString, 200, 365);
    }
}

function mousePress(event) {
    var mX = event.offsetX;
    var mY = event.offsetY;
    var userIndex = -1;
    var pos = 10;
    for (var i = 0; i< userHand.length; i++) {
        if (mX>pos && mX<pos+72 && mY>410 && mY<510) {
            userIndex = i;
            break;
        }
        pos += 72;
    }
    if (userIndex < 0) {
```

```
    return;
}
if (comPlayP == false) {
    var suit = userHand[userIndex].suit;
    var rank = userHand[userIndex].rank;
    var comIndex = -1;
    for (var i = 0; i < comHand.length; i++) {
        if (comHand[i].suit == suit) {
            comIndex = i;
            break;
        }
    }
    if (comIndex < 0) {
        for (var i = 0; i < comHand.length; i++) {
            if (comHand[i].rank == rank + 1) {
                comIndex = i;
                break;
            }
        }
        if (comIndex < 0) {
            for (var i = 0; i < comHand.length; i++) {
                if (comHand[i].rank < 10) {
                    comIndex = i;
                    break;
                }
            }
            if (comIndex < 0) {
                var temp = [];
                for (var i = 0; i < comHand.length; i++) {
                    if ((comHand[i].rank != 14) && (comHand[i].rank != 10)) {
                        temp.push(i);
                    }
                }
                if (temp.length > 0) {
                    comIndex = temp[Math.floor(Math.random() * temp.length)];
                } else {
                    comIndex = Math.floor(Math.random() * comHand.length);
                }
            }
        }
    }
    if (comHand[comIndex].rank == rank + 1) {
        getCom.push(userHand[userIndex]);
```

```
            getCom.push(comHand[comIndex]);
            comPlayP = true;
        } else {
                getUser.push(userHand[userIndex]);
                getUser.push(comHand[comIndex]);
                comPlayP = false;
        }
        userHand.splice(userIndex, 1);
        comHand.splice(comIndex, 1);
    } else { // コンピュータがリードしている
        var suit = userHand[userIndex].suit;
        var rank = userHand[userIndex].rank;
        if (dispCom[0].rank + 1 == rank) {
            getUser.push(userHand[userIndex]);
            getUser.push(dispCom[0]);
            comPlayP = false;
        } else {
            getCom.push(userHand[userIndex]);
            getCom.push(dispCom[0]);
            comPlayP = true;
        }
        userHand.splice(userIndex, 1);
    }
    if (Yama.length > 0) {
        if (comPlayP == true) { // 次はコンピュータのリード
            comHand.push(Yama[Yama.length-1]);
            Yama.splice(Yama.length-1, 1);
            userHand.push(Yama[Yama.length-1]);
            Yama.splice(Yama.length-1, 1);
        } else { // 次はユーザーのリード
            userHand.push(Yama[Yama.length-1]);
            Yama.splice(Yama.length-1, 1);
            comHand.push(Yama[Yama.length-1]);
            Yama.splice(Yama.length-1, 1);
        }
    }
    if (comHand.length == 0) {
        gameOverP = true;
        dispCom = [];
    } else if (comPlayP == true) { // 次はコンピュータのリード
        var comIndex = -1;
        for (var i = 0; i < comHand.length; i++) {
            if (comHand[i].rank == 14) {
```

```
                comIndex = i;
                break;
            }
        }
        if (comIndex < 0) {
            comIndex = Math.floor(Math.random() * comHand.length);
        }
        dispCom = [comHand[comIndex]];
        comHand.splice(comIndex, 1);
    } else { // 次はユーザーのリード
        dispCom = [];
    }
    ShowBan();
}

function init() {
    var canvas = document.getElementById("canvas");
    ctx = canvas.getContext("2d");
    canvas.onmousedown = mousePress;
    cardSet();
    gameInitialize();
    ShowBan();
}

function gameStart() {
    cardSet();
    gameInitialize();
    ShowBan();
}

</script>
</head>
<body onload="init()">
<p>
これはボヘミアン・シュナイダーと
いう名前の二人でするトランプゲームです。
場に出されたカード（リードされ
たカード）よりランクが一つ上の
カード（スーツは何でもよい）を
手札から出せば、そのトリックを
取れます。そうでなければ、リード
した人がそのトリックを取ります。
カードのランクは A,K,Q,J,10
```

,9,8,7 の順です。
トリックを取った人が次のリードを
します。最初のリードはあなたです。
山にカードがある間はそれぞれ一枚
づつ手札に補給されます。
手札が無くなれば、それぞれ取った
トリックの点数を数えます。
A は 11 点、K は 4 点、Q は 3 点、J は
2 点、10 は 10 点、その他は 0 点です。
あなたの手札は一番下の段です。
マウスでクリックして選びます。
コンピュータがリードしたときは
山の右側にリード
したカードが表示されます。
参考文献：松田道弘著「トランプ
ゲーム辞典」東京堂出版
</p>
```
<canvas id="canvas" width="1000" height="600"></canvas>
<img id="c1" src="c01.png" style="display:none" />
<img id="c2" src="c02.png" style="display:none" />
<img id="c3" src="c03.png" style="display:none" />
<img id="c4" src="c04.png" style="display:none" />
<img id="c5" src="c05.png" style="display:none" />
<img id="c6" src="c06.png" style="display:none" />
<img id="c7" src="c07.png" style="display:none" />
<img id="c8" src="c08.png" style="display:none" />
<img id="c9" src="c09.png" style="display:none" />
<img id="c10" src="c10.png" style="display:none" />
<img id="c11" src="c11.png" style="display:none" />
<img id="c12" src="c12.png" style="display:none" />
<img id="c13" src="c13.png" style="display:none" />
<img id="d1" src="d01.png" style="display:none" />
<img id="d2" src="d02.png" style="display:none" />
<img id="d3" src="d03.png" style="display:none" />
<img id="d4" src="d04.png" style="display:none" />
<img id="d5" src="d05.png" style="display:none" />
<img id="d6" src="d06.png" style="display:none" />
<img id="d7" src="d07.png" style="display:none" />
<img id="d8" src="d08.png" style="display:none" />
<img id="d9" src="d09.png" style="display:none" />
<img id="d10" src="d10.png" style="display:none" />
<img id="d11" src="d11.png" style="display:none" />
<img id="d12" src="d12.png" style="display:none" />
```

```
<img id="d13" src="d13.png" style="display:none" />
<img id="h1" src="h01.png" style="display:none" />
<img id="h2" src="h02.png" style="display:none" />
<img id="h3" src="h03.png" style="display:none" />
<img id="h4" src="h04.png" style="display:none" />
<img id="h5" src="h05.png" style="display:none" />
<img id="h6" src="h06.png" style="display:none" />
<img id="h7" src="h07.png" style="display:none" />
<img id="h8" src="h08.png" style="display:none" />
<img id="h9" src="h09.png" style="display:none" />
<img id="h10" src="h10.png" style="display:none" />
<img id="h11" src="h11.png" style="display:none" />
<img id="h12" src="h12.png" style="display:none" />
<img id="h13" src="h13.png" style="display:none" />
<img id="s1" src="s01.png" style="display:none" />
<img id="s2" src="s02.png" style="display:none" />
<img id="s3" src="s03.png" style="display:none" />
<img id="s4" src="s04.png" style="display:none" />
<img id="s5" src="s05.png" style="display:none" />
<img id="s6" src="s06.png" style="display:none" />
<img id="s7" src="s07.png" style="display:none" />
<img id="s8" src="s08.png" style="display:none" />
<img id="s9" src="s09.png" style="display:none" />
<img id="s10" src="s10.png" style="display:none" />
<img id="s11" src="s11.png" style="display:none" />
<img id="s12" src="s12.png" style="display:none" />
<img id="s13" src="s13.png" style="display:none" />
<img id="z1" src="z01.png" style="display:none" />
<button onclick="gameStart()">ゲーム再スタート</buton>
</body>
</html>
```

まず、

```
var ctx;
var carddeck = [];
var comHand = [];
var userHand = [];
var Yama = [];
var getCom = [];
var getUser = [];
var dispCom = [];
var gameOverP = false;
var comPlayP = false;
```

は大域変数の宣言です。切り札が無いので trump は不要です。
　次の

```
function Card(suit, rank, value, image, backimage) {
    this.suit = suit;
    this.rank = rank;
    this.value = value;
    this.image = image;
    this.backimage = backimage;
}
```

は、カードのオブジェクトです。suit と rank と value と image と backimage を持っています。value が追加されています。
　次の

```
function drawCard(card, x, y, flag) {
    if (flag == true) {
        ctx.drawImage(card.image, 0, 0, 200, 300, x, y, 71, 96);
    } else {
        ctx.drawImage(card.backimage, 0, 0, 200, 300, x, y, 71, 96);
    }
    ctx.strokeStyle = "blue";
    ctx.lineWidth = 1;
    ctx.strokeRect(x, y, 71, 96);
}
```

は、同じでいいです。カードを指定された位置（x, y）に描く関数です。flag で表を描くか、裏を描くかを指定しています。
　次の

```
function cardSet() {
    var backimage = document.getElementById("z1");
    carddeck = [];
    for (var s = 0; s < 4; s++) {
        for (var r = 0; r < 8; r++) {
            var name = "";
            switch (s) {
                case 0: name = "c"; break;
                case 1: name = "d"; break;
                case 2: name = "h"; break;
                case 3: name = "s"; break;
            }
            var rank, value;
            switch (r) {
                case 0: rank = 14; value = 11; name +=  1; break;
                case 1: rank = 13; value =  4; name += 13; break;
```

```
                case 2: rank = 12; value =  3; name += 12; break;
                case 3: rank = 11; value =  2; name += 11; break;
                case 4: rank = 10; value = 10; name += 10; break;
                case 5: rank =  9; value =  0; name +=  9; break;
                case 6: rank =  8; value =  0; name +=  8; break;
                case 7: rank =  7; value =  0; name +=  7; break;
            }
            var image = document.getElementById(name);
            var card = new Card(s, rank, value, image, backimage);
            carddeck.push(card);
        }
    }
}
```

は、３２枚のカードを carddeck にセットしています。ランクを判定しやすくするために、エースのランクを１４にしています。value に最後に得点を計算するために、各カードの点数をセットしています。

次の

```
function cardInitialize() {
    i = carddeck.length
    while (i > 0) {
        var s = Math.floor(Math.random() * i);
        var temp = carddeck[--i]
        carddeck[i] = carddeck[s]
        carddeck[s] = temp
    }
}
```

は、同じです。カードをシャッフルしています。

次の

```
function gameInitialize() {
    comHand = []
    userHand = []
    Yama = []

    cardInitialize()
    var index = 0
    for (var i = 0; i < 6; i++) {
        comHand.push(carddeck[index]);
        index++;
    }
    for (var i = 0; i < 6; i++) {
        userHand.push(carddeck[index]);
```

```
        index++;
    }
    for (var i = 0; i < 20; i++) {
        Yama.push(carddeck[index]);
        index++;
    }
    getCom = [];
    getUser = [];
    dispCom = [];
    gameOverP = false;
    comPlayP = false;
}
```

は、カードを配り、大域変数をセットしています。
次の

```
function ShowBan() {
    ctx.clearRect(0, 0, 1000, 600);
    var pos = 10;
    for (var i = 0; i < comHand.length; i++) {
        drawCard(comHand[i], pos, 10, false);
        pos += 72;
    }
    pos = 10;
    for (var i = 0; i < getCom.length; i++) {
        drawCard(getCom[i], pos, 110, true);
        pos += 20;
    }
    pos = 10;
    if (Yama.length > 0) {
        drawCard(Yama[Yama.length-1], pos, 210, false);
    }
    // pos = 145;
    pos = 245;
    if (dispCom.length > 0) {
        drawCard(dispCom[0], pos, 210, true);
    }
    pos = 10;
    for (var i = 0; i < getUser.length; i++) {
        drawCard(getUser[i], pos, 310, true);
         pos += 20;
    }
    pos = 10;
    for (var i = 0; i < userHand.length; i++) {
```

```
        drawCard(userHand[i], pos, 410, true);
        pos += 72;
    }
    ctx.font = "24px 'Times New Roman'";
    ctx.strokeStyle = "blue";
    ctx.fillStyle = "blue";
    if (gameOverP == false) {
        ctx.fillText("カードを選んで下さい", 400, 265);
    } else {
        var comValue = 0;
        for (var i = 0; i < getCom.length; i++) {
            comValue += getCom[i].value;
        }
        var userValue = 0;
        for (var i = 0; i < getUser.length; i++) {
            userValue += getUser[i].value;
        }
        var drawString = "Game Over ";
        drawString += userValue;
        drawString += " : ";
        drawString += comValue;
        if (comValue > userValue) {
            drawString += " コンピュータの勝ちです。頑張ってね！";
            ctx.strokeText(drawString, 200, 265);
        } else if (comValue < userValue) {
            drawString += " あなたの勝ちです。強いですね！";
            ctx.strokeText(drawString,200, 265);
        } else {
            drawString += "  引き分けです。もう一度しませんか？";
            ctx.strokeText(drawString, 200, 265);
        }
        drawString = "もう一度ゲームをするためには「ゲーム再スタート」のボタンをクリッ
クしてください";
        ctx.strokeText(drawString, 100, 365);
    }
}
```

は、微調整が必要です。現在の局面を表示します。

次の

```
function mousePress(event) {
    var mX = event.offsetX;
    var mY = event.offsetY;
    var userIndex = -1;
```

```
var pos = 10;
for (var i = 0; i< userHand.length; i++) {
    if (mX>pos && mX<pos+72 && mY>410 && mY<510) {
        userIndex = i;
        break;
    }
    pos += 72;
}
if (userIndex < 0) {
    return;
}
if (comPlayP == false) {
    var suit = userHand[userIndex].suit;
    var rank = userHand[userIndex].rank;
    var comIndex = -1;
    for (var i = 0; i < comHand.length; i++) {
        if (comHand[i].suit == suit) {
            comIndex = i;
            break;
        }
    }
    if (comIndex < 0) {
        for (var i = 0; i < comHand.length; i++) {
            if (comHand[i].rank == rank + 1) {
                comIndex = i;
                break;
            }
        }
        if (comIndex < 0) {
            for (var i = 0; i < comHand.length; i++) {
                if (comHand[i].rank < 10) {
                    comIndex = i;
                    break;
                }
            }
            if (comIndex < 0) {
                var temp = [];
                for (var i = 0; i < comHand.length; i++) {
                    if ((comHand[i].rank != 14) && (comHand[i].rank != 10)) {
                        temp.push(i);
                    }
                }
                if (temp.length > 0) {
```

```
                    comIndex = temp[Math.floor(Math.random() * temp.length)];
                } else {
                    comIndex = Math.floor(Math.random() * comHand.length);
                }
            }
        }
    }
    if (comHand[comIndex].rank == rank + 1) {
        getCom.push(userHand[userIndex]);
        getCom.push(comHand[comIndex]);
        comPlayP = true;
    } else {
            getUser.push(userHand[userIndex]);
            getUser.push(comHand[comIndex]);
            comPlayP = false;
    }
    userHand.splice(userIndex, 1);
    comHand.splice(comIndex, 1);
} else { // コンピュータがリードしている
    var suit = userHand[userIndex].suit;
    var rank = userHand[userIndex].rank;
    if (dispCom[0].rank + 1 == rank) {
        getUser.push(userHand[userIndex]);
        getUser.push(dispCom[0]);
        comPlayP = false;
    } else {
        getCom.push(userHand[userIndex]);
        getCom.push(dispCom[0]);
        comPlayP = true;
    }
    userHand.splice(userIndex, 1);
}
if (Yama.length > 0) {
    if (comPlayP == true) { // 次はコンピュータのリード
        comHand.push(Yama[Yama.length-1]);
        Yama.splice(Yama.length-1, 1);
        userHand.push(Yama[Yama.length-1]);
        Yama.splice(Yama.length-1, 1);
    } else { // 次はユーザーのリード
        userHand.push(Yama[Yama.length-1]);
        Yama.splice(Yama.length-1, 1);
        comHand.push(Yama[Yama.length-1]);
        Yama.splice(Yama.length-1, 1);
```

```
        }
    }
    if (comHand.length == 0) {
        gameOverP = true;
        dispCom = [];
    } else if (comPlayP == true) { // 次はコンピュータのリード
        var comIndex = -1;
        for (var i = 0; i < comHand.length; i++) {
            if (comHand[i].rank == 14) {
                comIndex = i;
                break;
            }
        }
        if (comIndex < 0) {
            comIndex = Math.floor(Math.random() * comHand.length);
        }
        dispCom = [comHand[comIndex]];
        comHand.splice(comIndex, 1);
    } else { // 次はユーザーのリード
        dispCom = [];
    }
    ShowBan();
}
```

は、微調整が必要です。ゲームのルールに応じて修正します。コンピュータの手の選び方を書いていますが、適切なものではありません。強くするにはどのようにすればいいかは自分で考えて下さい。

次の

```
function init() {
    var canvas = document.getElementById("canvas");
    ctx = canvas.getContext("2d");
    canvas.onmousedown = mousePress;
    cardSet();
    gameInitialize();
    ShowBan();
}
```

は、同じでいいです。最初の処理を書いています。

次の

```
function gameStart() {
    cardSet();
    gameInitialize();
    ShowBan();
```

```
}
```

は、同じでいいです。「ゲーム再スタート」のボタンをクリックしたときの処理を書いています。
次の

```
<body onload="init()">
<p>
これはボヘミアン・シュナイダーと
いう名前の二人でするトランプゲームです。
場に出されたカード（リードされ
たカード）よりランクが一つ上の
カード（スーツは何でもよい）を
手札から出せば、そのトリックを
取れます。そうでなければ、リード
した人がそのトリックを取ります。
カードのランクは A,K,Q,J,10
,9,8,7 の順です。
トリックを取った人が次のリードを
します。最初のリードはあなたです。
山にカードがある間はそれぞれ一枚
づつ手札に補給されます。
手札が無くなれば、それぞれ取った
トリックの点数を数えます。
A は 11 点、K は 4 点、Q は 3 点、J は
2 点、10 は 10 点、その他は 0 点です。
あなたの手札は一番下の段です。
マウスでクリックして選びます。
コンピュータがリードしたときは
山の右側にリード
したカードが表示されます。
参考文献：松田道弘著「トランプ
ゲーム辞典」東京堂出版
</p>
<canvas id="canvas" width="1000" height="600"></canvas>
<img id="c1" src="c01.png" style="display:none" />
<img id="c2" src="c02.png" style="display:none" />
<img id="c3" src="c03.png" style="display:none" />
<img id="c4" src="c04.png" style="display:none" />
<img id="c5" src="c05.png" style="display:none" />
<img id="c6" src="c06.png" style="display:none" />
<img id="c7" src="c07.png" style="display:none" />
<img id="c8" src="c08.png" style="display:none" />
<img id="c9" src="c09.png" style="display:none" />
<img id="c10" src="c10.png" style="display:none" />
```

```
<img id="c11" src="c11.png" style="display:none" />
<img id="c12" src="c12.png" style="display:none" />
<img id="c13" src="c13.png" style="display:none" />
<img id="d1" src="d01.png" style="display:none" />
<img id="d2" src="d02.png" style="display:none" />
<img id="d3" src="d03.png" style="display:none" />
<img id="d4" src="d04.png" style="display:none" />
<img id="d5" src="d05.png" style="display:none" />
<img id="d6" src="d06.png" style="display:none" />
<img id="d7" src="d07.png" style="display:none" />
<img id="d8" src="d08.png" style="display:none" />
<img id="d9" src="d09.png" style="display:none" />
<img id="d10" src="d10.png" style="display:none" />
<img id="d11" src="d11.png" style="display:none" />
<img id="d12" src="d12.png" style="display:none" />
<img id="d13" src="d13.png" style="display:none" />
<img id="h1" src="h01.png" style="display:none" />
<img id="h2" src="h02.png" style="display:none" />
<img id="h3" src="h03.png" style="display:none" />
<img id="h4" src="h04.png" style="display:none" />
<img id="h5" src="h05.png" style="display:none" />
<img id="h6" src="h06.png" style="display:none" />
<img id="h7" src="h07.png" style="display:none" />
<img id="h8" src="h08.png" style="display:none" />
<img id="h9" src="h09.png" style="display:none" />
<img id="h10" src="h10.png" style="display:none" />
<img id="h11" src="h11.png" style="display:none" />
<img id="h12" src="h12.png" style="display:none" />
<img id="h13" src="h13.png" style="display:none" />
<img id="s1" src="s01.png" style="display:none" />
<img id="s2" src="s02.png" style="display:none" />
<img id="s3" src="s03.png" style="display:none" />
<img id="s4" src="s04.png" style="display:none" />
<img id="s5" src="s05.png" style="display:none" />
<img id="s6" src="s06.png" style="display:none" />
<img id="s7" src="s07.png" style="display:none" />
<img id="s8" src="s08.png" style="display:none" />
<img id="s9" src="s09.png" style="display:none" />
<img id="s10" src="s10.png" style="display:none" />
<img id="s11" src="s11.png" style="display:none" />
<img id="s12" src="s12.png" style="display:none" />
<img id="s13" src="s13.png" style="display:none" />
<img id="z1" src="z01.png" style="display:none" />
```

```
<button onclick="gameStart()">ゲーム再スタート</buton>
</body>
```

は、ゲームのルールの説明文と canvas の設定とカードの画像の設定とボタンの設定です。

実行すると

のような途中経過になります。「ゲーム再スタート」のボタンを押すと再度ゲームをすることが出来ます。

コンピュータはルール通りプレイするだけです。カードをソートし表示を見やすくしたり、種々の情報を集め強くするのは各自で考えてください。トランプゲームのプログラムを作るときの参考にしてください。