

トランプゲームを作ってみよう (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「ジャーマン・ホイスト」を Javascript 作ってみます。

ジャーマン・ホイストはホイストのバリエーションの一つで、2人でプレイするように工夫されたものです。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。

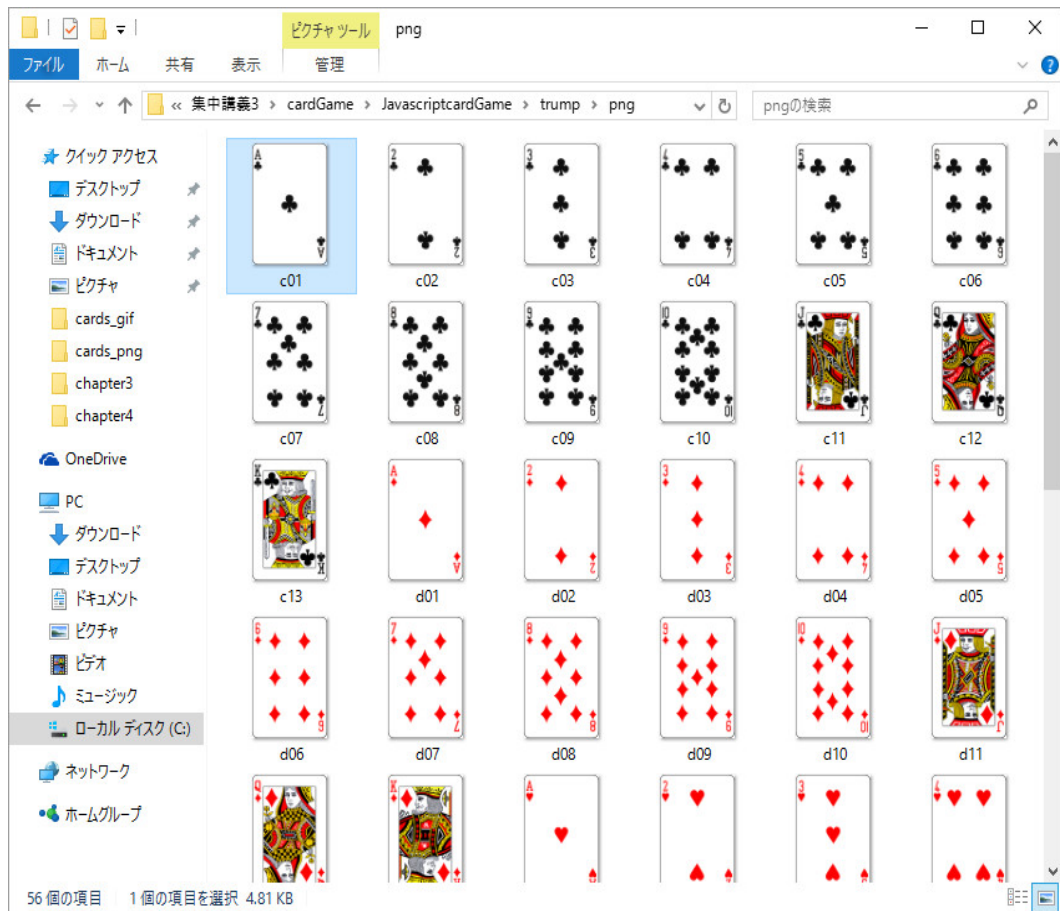
ルールは

1. 人数は2人
2. 52枚のカードを使用する
3. カードの順位は A, K, Q, J, 10, 9, 8, 7, 6, 5, 4, 3, 2 の順です
4. 手札は各人13枚で、残りはストックとしてテーブルに置き、ストックのトップカードだけを表向きにします。このカードのスーツが切り札になります。
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は必ず出さなければなりません。リードされたスーツを持っていないければ、どのカードを出しても良いです。
7. 最強の切り札を出した人か、または切り札が場に出ていなければリードされたスーツの一番高いランクのカードを出した人が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（切り札）を取って手札に入れ、相手はその下のカード（裏向き）を取ります。相手には見せません。
9. 各プレイヤーの手札は13枚に戻りました。この時、トリックを取った人は、次のリードをする前に、ストックの一番上のカードを表向きにします。
10. 第1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表向きのカードを取り、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の数を比べ、勝った人はその差を得点します。トリック数が同数の時は両者無得点です。

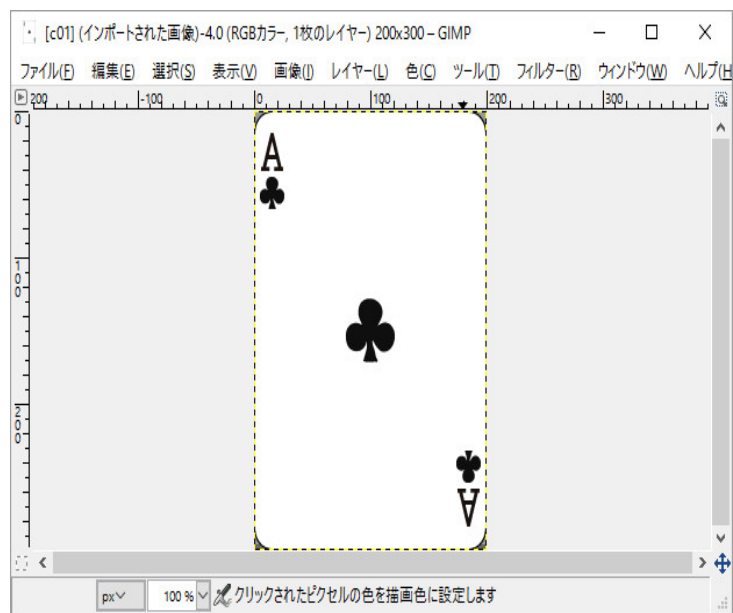
です。

このジャーマン・ホイストのプログラムを 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。

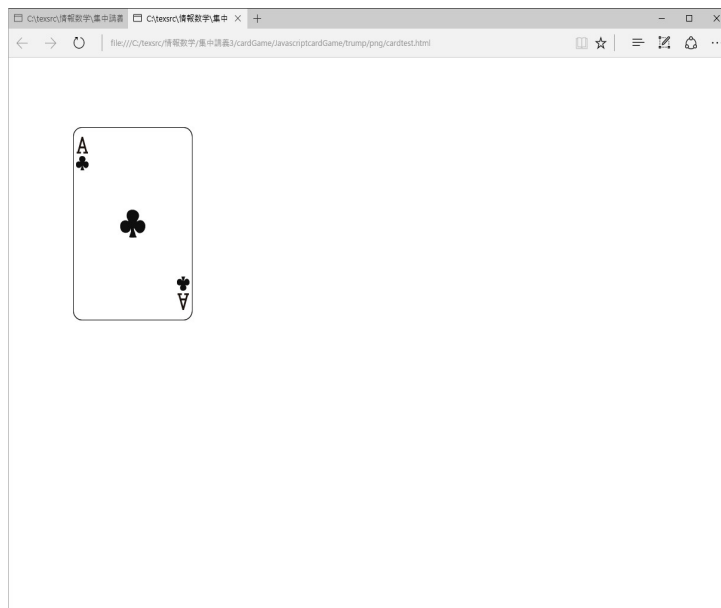
```

<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>

</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");

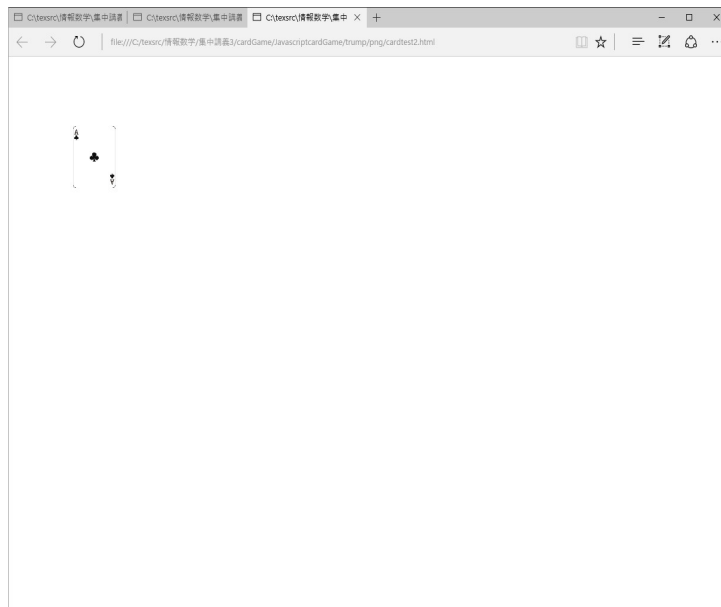
```

```

    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。

Python のプログラムを JavaScript のプログラムに翻訳します。

まず、カードのオブジェクトを作り、テストしてみます。Python のプログラムをそのまま使えるように、キャンバスと画像の大きさを同じにします。プログラムは

```

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

```

```

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

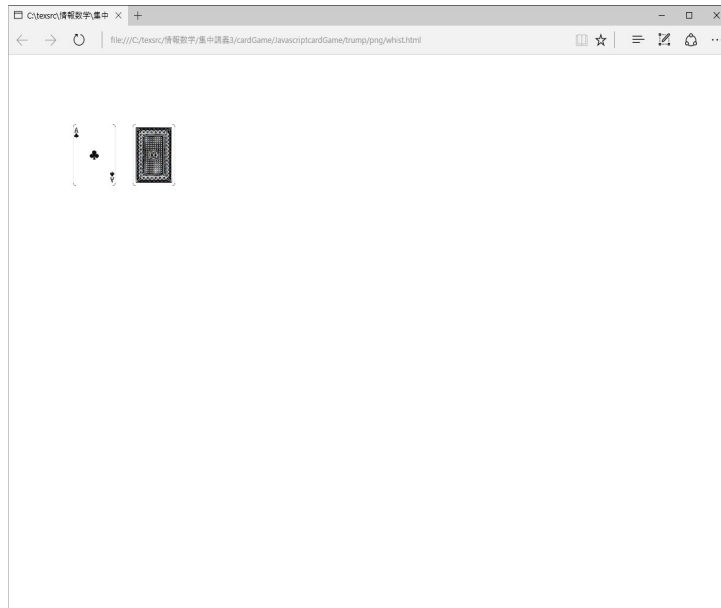
function drawCard(card, x, y, flag) {
    if (flag == true) {
        ctx.drawImage(card.image, 0, 0, 200,300, x, y, 71, 96);
    } else {
        ctx.drawImage(card.backimage, 0, 0, 200,300, x, y, 71, 96);
    }
}

function init() {
    var canvas = document.getElementById("canvas");
    ctx = canvas.getContext("2d");
    var image = document.getElementById("c1");
    var backimage = document.getElementById("z1");
    var card = new Card(0, 0, image, backimage);
    drawCard(card, 100, 100, true);
    drawCard(card, 200, 100, false);
}
</script>
</head>
<body onload="init()">
<canvas id="canvas" width="1000" height="600"></canvas>


</body>
</html>

```

です。実行すると



を描きます。

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

がカードのオブジェクトの定義で、

```
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);
    }
}
```

が、カードを座標 (x, y) に表示する関数です。flag==true なら表を表示し、flag==false なら裏を表示します。

次に、card_set() 関数を翻訳します。

```
<html>
<head>
<script>
var ctx;
var carddeck = [];
function Card(suit, rank, image, backimage) {
```

```

        this.suit = suit;
        this.rank = rank;
        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);
    }
}

function cardSet() {
    var backimage = document.getElementById("z1");
    carddeck = [];
    for (var s = 0; s < 4; s++) {
        for (var r = 0; r < 13; 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;
            }
            name = name + (r + 1);
            var image = document.getElementById(name);
            var card = new Card(s, r, image, backimage);
            carddeck.push(card);
        }
    }
}

function init() {
    var canvas = document.getElementById("canvas");
    ctx = canvas.getContext("2d");
    cardSet();
    drawCard(carddeck[0], 100, 100, true);
    drawCard(carddeck[50], 200, 100, true);
}
</script>
</head>

```

```

<body onload="init()">
<canvas id="canvas" width="1000" height="600"></canvas>










































```



```









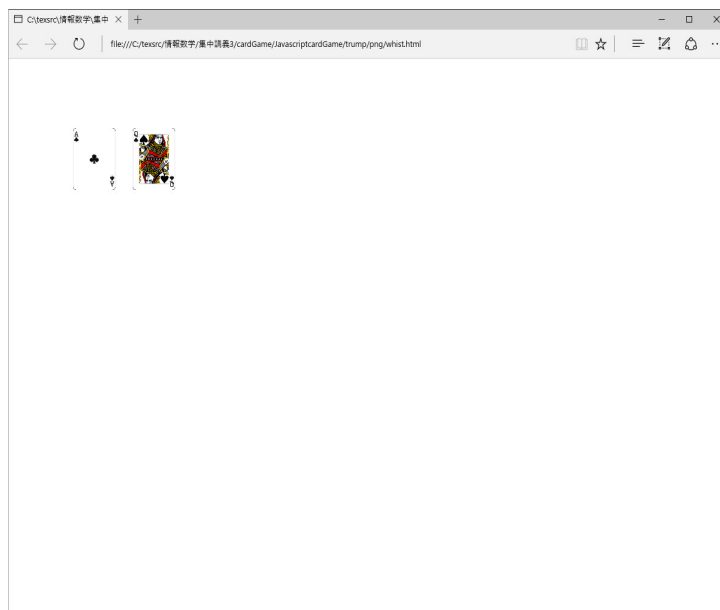




</body>
</html>

```

実行すると



を描きます。

次に `card_initialize()` を翻訳します。

```

<html>
<head>
<script>
var ctx;
var carddeck = [];
function Card(suit, rank, image, backimage) {
    this.suit = suit;
    this.rank = rank;
    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);
        }
    }

    function cardSet() {
        var backimage = document.getElementById("z1");
        carddeck = [];
        for (var s = 0; s < 4; s++) {
            for (var r = 0; r < 13; 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;
                }
                name = name + (r + 1);
                var image = document.getElementById(name);
                var card = new Card(s, r, 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 init() {
        var canvas = document.getElementById("canvas");
    }

```

```

        ctx = canvas.getContext("2d");
        cardSet();
        cardInitialize();
        drawCard(carddeck[0], 100, 100, true);
        drawCard(carddeck[50], 200, 100, true);
    }
</script>
</head>
<body onload="init()">
<canvas id="canvas" width="1000" height="600"></canvas>


































```

```

















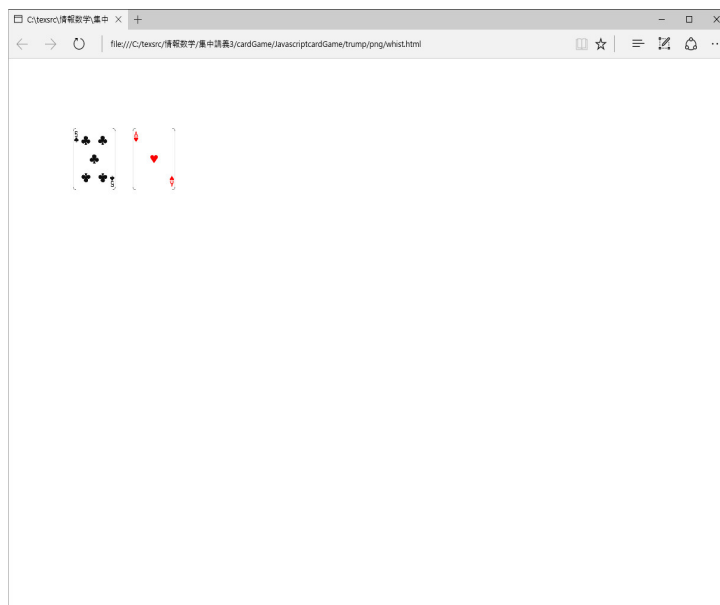




</body>
</html>

```

実行すると



のような図を描きます。

次に `gameInitialize()` を翻訳します。

```
<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;
var trump = 0;

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

function cardSet() {
    var backimage = document.getElementById("z1");
    carddeck = [];
    for (var s = 0; s < 4; s++) {
        for (var r = 0; r < 13; 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;
            }
            name = name + (r + 1);
            var image = document.getElementById(name);

```

```

        var card = new Card(s, r, 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 < 13; i++) {
        comHand.push(carddeck[index]);
        index++;
    }
    for (var i = 0; i < 13; i++) {
        userHand.push(carddeck[index]);
        index++;
    }
    for (var i = 0; i < 26; i++) {
        Yama.push(carddeck[index]);
        index++;
    }
    getCom = [];
    getUser = [];
    dispCom = [];
    gameOverP = false;
    comPlayP = false;
    trump = Yama[Yama.length-1].suit;
}

```

```

function init() {
    var canvas = document.getElementById("canvas");
    ctx = canvas.getContext("2d");
    cardSet();
    gameInitialize();
    drawCard(comHand[0], 100, 100, true);
    drawCard(userHand[5], 200, 100, true);
    drawCard(Yama[25], 300, 100, false);
}
</script>
</head>
<body onload="init()">
<canvas id="canvas" width="1000" height="600"></canvas>































```

```



















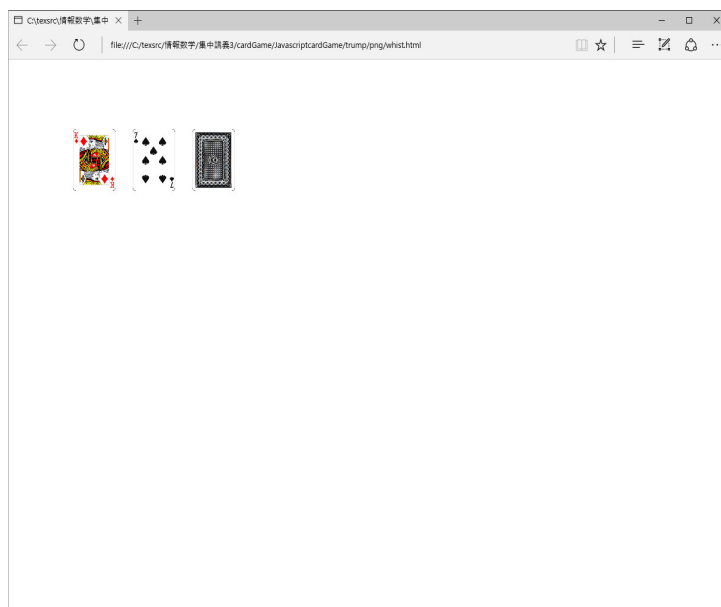





</body>
</html>

```

実行すると



のような図を描きます。

次に ShowBan() を翻訳します。

```
<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;
var trump = 0;

function Card(suit, rank, image, backimage) {
    this.suit = suit;
    this.rank = rank;
    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 < 13; 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;
    }
    name = name + (r + 1);
    var image = document.getElementById(name);
    var card = new Card(s, r, 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 < 13; i++) {
        comHand.push(carddeck[index]);
        index++;
    }
    for (var i = 0; i < 13; i++) {
        userHand.push(carddeck[index]);
        index++;
    }
    for (var i = 0; i < 26; i++) {
        Yama.push(carddeck[index]);
        index++;
    }
    getCom = [];
    getUser = [];

```

```

    dispCom = [];
    gameOverP = false;
    comPlayP = false;
    trump = Yama[Yama.length-1].suit;
}

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, true);
    }
    // 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 = getCom.length / 2;
        var userValue = getUser.length / 2;
        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);
        }
    }
}

```

```

function init() {
    var canvas = document.getElementById("canvas");
    ctx = canvas.getContext("2d");
    cardSet();
    gameInitialize();
    ShowBan();
}
</script>
</head>
<body onload="init()">
<canvas id="canvas" width="1000" height="600"></canvas>














```

```









































</body>
</html>

```

実行すると



のような図を描きます。但し、drawCard() を

```
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);
}
```

と修正しています。

次に buttonPress(event) を翻訳します。これはマウスで手札をクリックしたときの処理をプログラミングしたもので、mousePress() とします。

プログラムは

```
<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;
var trump = 0;

function Card(suit, rank, image, backimage) {
    this.suit = suit;
    this.rank = rank;
    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 < 13; 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;
            }
            name = name + (r + 1);
            var image = document.getElementById(name);
            var card = new Card(s, r, 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 < 13; i++) {
        comHand.push(carddeck[index]);
        index++;
    }
    for (var i = 0; i < 13; i++) {
        userHand.push(carddeck[index]);
        index++;
    }
    for (var i = 0; i < 26; i++) {
        Yama.push(carddeck[index]);
        index++;
    }
    getCom = [];
    getUser = [];
    dispCom = [];
    gameOverP = false;
    comPlayP = false;
    trump = Yama[Yama.length-1].suit;
}

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, true);
}
// 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 = getCom.length / 2;
    var userValue = getUser.length / 2;
    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);
    }
}

}

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].suit == trump) {
                    comIndex = i;
                    break;
                }
            }
        }
    }
}

```

```

    }
    if (comIndex < 0) {
        comIndex = Math.floor(Math.random() * comHand.length);
    }
}
if (comHand[comIndex].suit == userHand[userIndex].suit) {
    if (comHand[comIndex].rank == 0) {
        getCom.push(userHand[userIndex]);
        getCom.push(comHand[comIndex]);
        comPlayP = true;
    } else if (userHand[userIndex].rank == 0) {
        getUser.push(userHand[userIndex]);
        getUser.push(comHand[comIndex]);
        comPlayP = false;
    } else if (comHand[comIndex].rank > userHand[userIndex].rank) {
        getCom.push(userHand[userIndex]);
        getCom.push(comHand[comIndex]);
        comPlayP = true;
    } else {
        getUser.push(userHand[userIndex]);
        getUser.push(comHand[comIndex]);
        comPlayP = false;
    }
} else if (comHand[comIndex].suit == trump) {
    getCom.push(userHand[userIndex]);
    getCom.push(comHand[comIndex]);
    comPlayP = true;
} else if (userHand[userIndex].suit == trump) {
    getUser.push(userHand[userIndex]);
    getUser.push(comHand[comIndex]);
    comPlayP = false;
} 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 (suit != dispCom[0].suit) {
        flag = false;
    }
}

```

```

for (var i = 0; i < userHand.length; i++) {
    if (userHand[i].suit == dispCom[0].suit) {
        flag = true;
        break;
    }
}
if (flag == true) {
    var s = "リードは";
    if (dispCom[0].suit == 0) {
        s += "クラブ";
    } else if (dispCom[0].suit == 1) {
        s += "ダイヤ";
    } else if (dispCom[0].suit == 2) {
        s += "ハート";
    } else {
        s += "スペード";
    }
    s += "です";
    ctx.font = "24px 'Times New Roman'";
    ctx.fillStyle = "red";
    ctx.fillText(s, 400, 365);
    return;
}
}
if (dispCom[0].suit == userHand[userIndex].suit) {
    if (dispCom[0].rank == 0) {
        getCom.push(userHand[userIndex]);
        getCom.push(dispCom[0]);
        comPlayP = true;
    } else if (userHand[userIndex].rank == 0) {
        getUser.push(userHand[userIndex]);
        getUser.push(dispCom[0]);
        comPlayP = false;
    } else if (dispCom[0].rank > userHand[userIndex].rank) {
        getCom.push(userHand[userIndex]);
        getCom.push(dispCom[0]);
        comPlayP = true;
    } else {
        getUser.push(userHand[userIndex]);
        getUser.push(dispCom[0]);
        comPlayP = false;
    }
} else if (dispCom[0].suit == trump) {

```

```

        getCom.push(userHand[userIndex]);
        getCom.push(disCom[0]);
        comPlayP = true;
    } else if (userHand[userIndex].suit == trump) {
        getUser.push(userHand[userIndex]);
        getUser.push(disCom[0]);
        comPlayP = false;
    } else {
        getCom.push(userHand[userIndex]);
        getCom.push(disCom[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;
    disCom = [];
} else if (comPlayP == true) { // 次はコンピュータのリード
    var comIndex = Math.floor(Math.random() * comHand.length);
    disCom = [comHand[comIndex]];
    comHand.splice(comIndex, 1);
} else { // 次はユーザーのリード
    disCom = [];
}
ShowBan();
}

function init() {
    var canvas = document.getElementById("canvas");

```

```

    ctx = canvas.getContext("2d");
    canvas.onmousedown = mousePress;
    cardSet();
    gameInitialize();
    ShowBan();
}
</script>
</head>
<body onload="init()">
<canvas id="canvas" width="1000" height="600"></canvas>


































```

```



















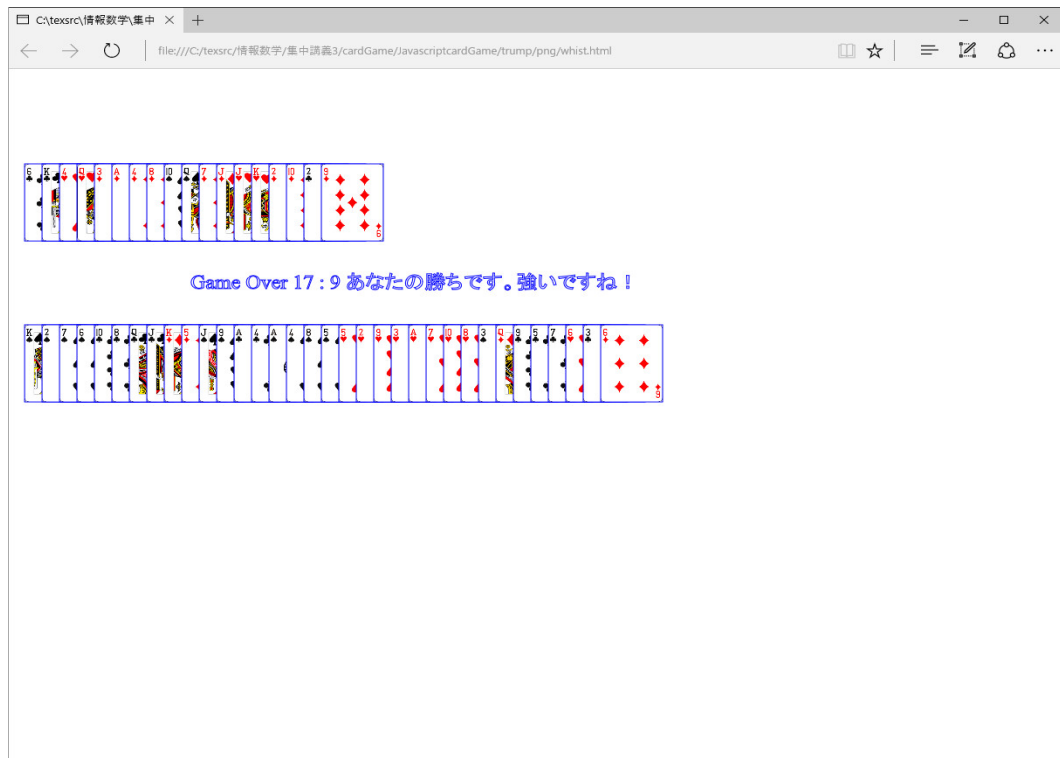


</body>
</html>

```

実行すると





のような図を描きます。

最後に、再スタートボタンを追加し、ボタンを押すと `gameStart()` を実行するようにします。

```
<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;
var trump = 0;

function Card(suit, rank, image, backimage) {
    this.suit = suit;
    this.rank = rank;
    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 < 13; 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;
            }
            name = name + (r + 1);
            var image = document.getElementById(name);
            var card = new Card(s, r, 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 < 13; i++) {
        comHand.push(carddeck[index]);
        index++;
    }
    for (var i = 0; i < 13; i++) {
        userHand.push(carddeck[index]);
        index++;
    }
    for (var i = 0; i < 26; i++) {
        Yama.push(carddeck[index]);
        index++;
    }
    getCom = [];
    getUser = [];
    dispCom = [];
    gameOverP = false;
    comPlayP = false;
    trump = Yama[Yama.length-1].suit;
}

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, true);
    }
}

```

```

    // 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 = getCom.length / 2;
        var userValue = getUser.length / 2;
        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);
        }
    }
}

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].suit == trump) {
                comIndex = i;
                break;
            }
        }
        if (comIndex < 0) {
            comIndex = Math.floor(Math.random() * comHand.length);
        }
    }
    if (comHand[comIndex].suit == userHand[userIndex].suit) {
        if (comHand[comIndex].rank == 0) {
            getCom.push(userHand[userIndex]);
            getCom.push(comHand[comIndex]);
            comPlayP = true;
        } else if (userHand[userIndex].rank == 0) {
            getUser.push(userHand[userIndex]);
            getUser.push(comHand[comIndex]);
            comPlayP = false;
        } else if (comHand[comIndex].rank > userHand[userIndex].rank) {

```

```

        getCom.push(userHand[userIndex]);
        getCom.push(comHand[comIndex]);
        comPlayP = true;
    } else {
        getUser.push(userHand[userIndex]);
        getUser.push(comHand[comIndex]);
        comPlayP = false;
    }
} else if (comHand[comIndex].suit == trump) {
    getCom.push(userHand[userIndex]);
    getCom.push(comHand[comIndex]);
    comPlayP = true;
} else if (userHand[userIndex].suit == trump) {
    getUser.push(userHand[userIndex]);
    getUser.push(comHand[comIndex]);
    comPlayP = false;
} 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 (suit != dispCom[0].suit) {
        flag = false;
        for (var i = 0; i < userHand.length; i++) {
            if (userHand[i].suit == dispCom[0].suit) {
                flag = true;
                break;
            }
        }
    }
    if (flag == true) {
        var s = "リードは";
        if (dispCom[0].suit == 0) {
            s += "クラブ";
        } else if (dispCom[0].suit == 1) {
            s += "ダイヤ";
        } else if (dispCom[0].suit == 2) {
            s += "ハート";
        } else {

```

```

        s += "スペード";
    }
    s += "です";
    ctx.font = "24px 'Times New Roman'";
    ctx.fillStyle = "red";
    ctx.fillText(s, 400, 365);
    return;
}
}
if (dispCom[0].suit == userHand[userIndex].suit) {
    if (dispCom[0].rank == 0) {
        getCom.push(userHand[userIndex]);
        getCom.push(dispCom[0]);
        comPlayP = true;
    } else if (userHand[userIndex].rank == 0) {
        getUser.push(userHand[userIndex]);
        getUser.push(dispCom[0]);
        comPlayP = false;
    } else if (dispCom[0].rank > userHand[userIndex].rank) {
        getCom.push(userHand[userIndex]);
        getCom.push(dispCom[0]);
        comPlayP = true;
    } else {
        getUser.push(userHand[userIndex]);
        getUser.push(dispCom[0]);
        comPlayP = false;
    }
} else if (dispCom[0].suit == trump) {
    getCom.push(userHand[userIndex]);
    getCom.push(dispCom[0]);
    comPlayP = true;
} else if (userHand[userIndex].suit == trump) {
    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 = 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()">

```

```

<canvas id="canvas" width="1000" height="600"></canvas>











































```



```












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

```

プログラムが完成しました。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。

単に、Python のプログラムを JavaScript に翻訳しただけですから、細かな説明はしません。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。