

トランプゲームを作ってみよう (C# 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 C# で作ってみます。

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。

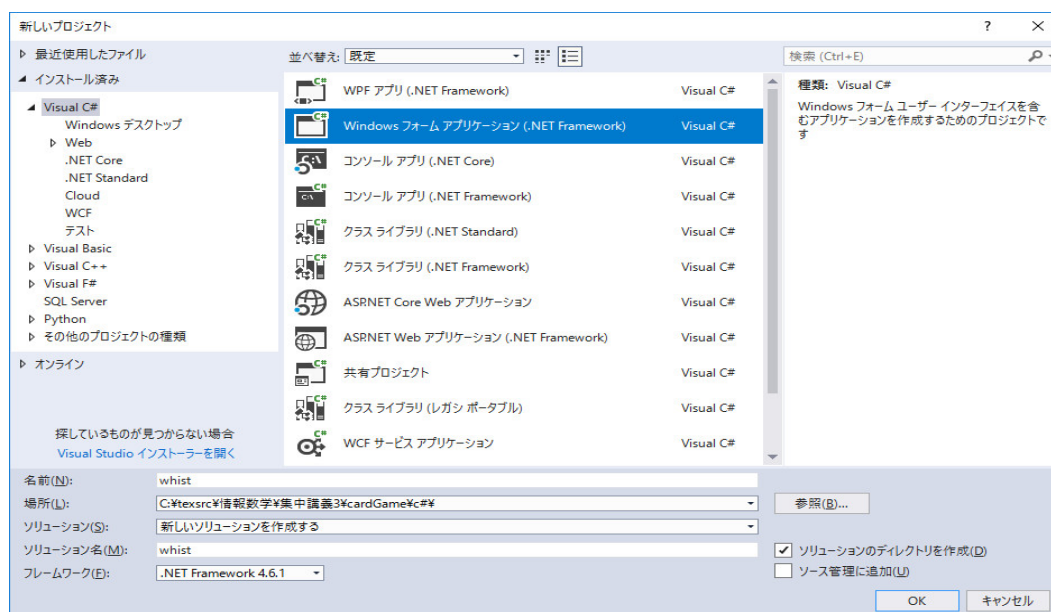
ルールは

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。

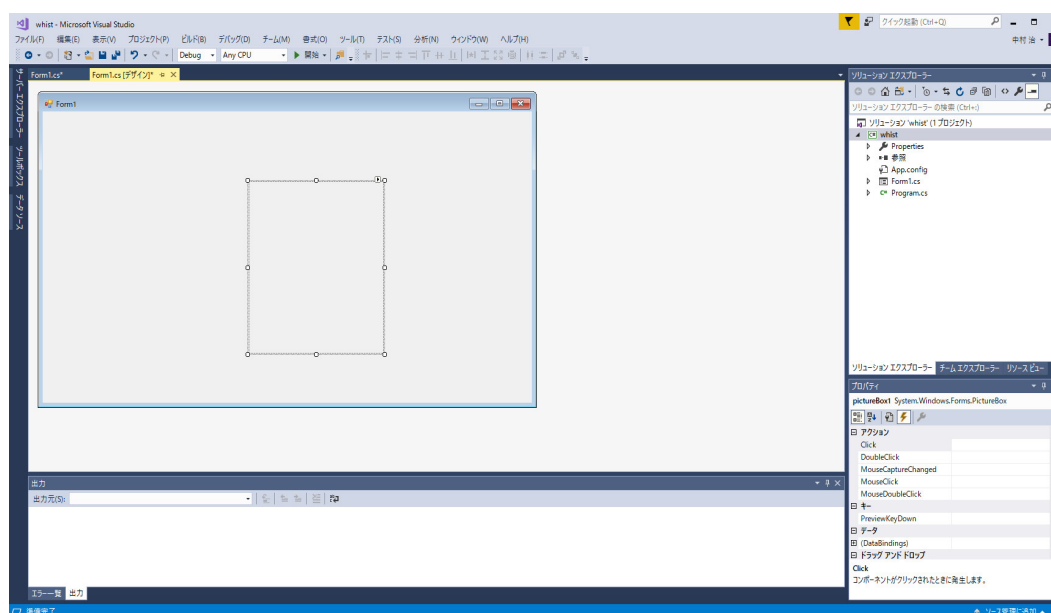
です。

このジャーマン・ホイストのプログラムを C# で作成し、コンピュータと対戦できるようにします。まだインストールしてなければ、フリーの Microsoft Visual Sutudio 2019 をインストールします。

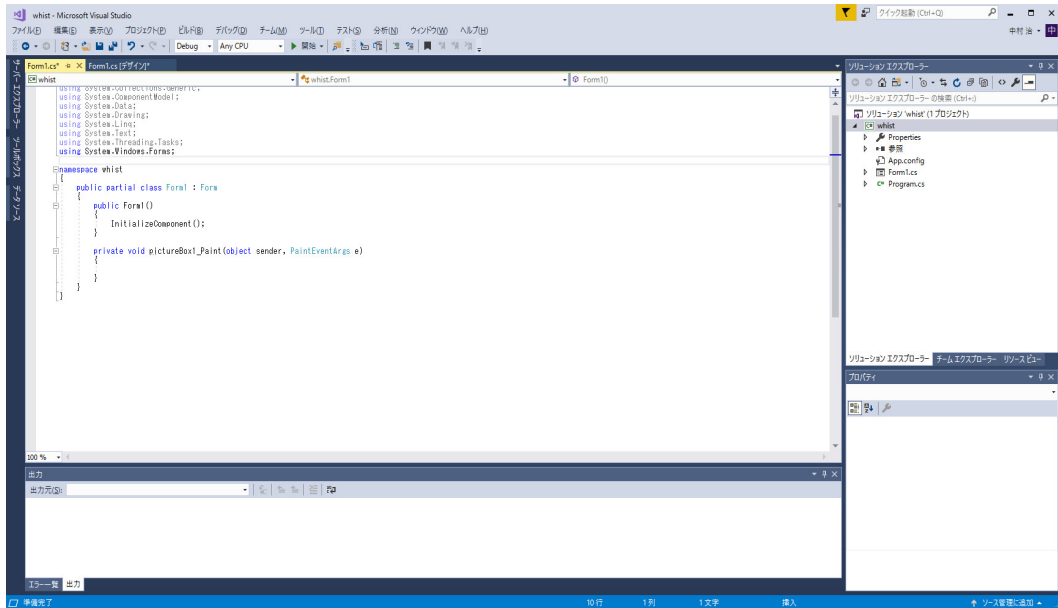
Microsoft Visual Sutudio を立ち上げ、C# の「ファイル」の「新規作成」の「プロジェクト」で



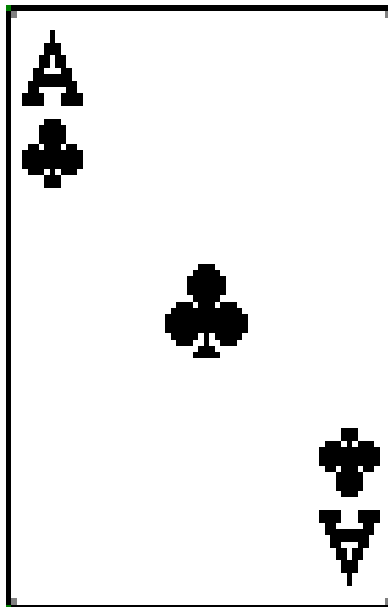
Windows フォームアプリケーションを選択します。ツールバーボックスから PictureBox を選び、ドラッグして適当な大きさにします。



PictureBox のイベントの一番下にある paint を選び、ダブルクリックします。



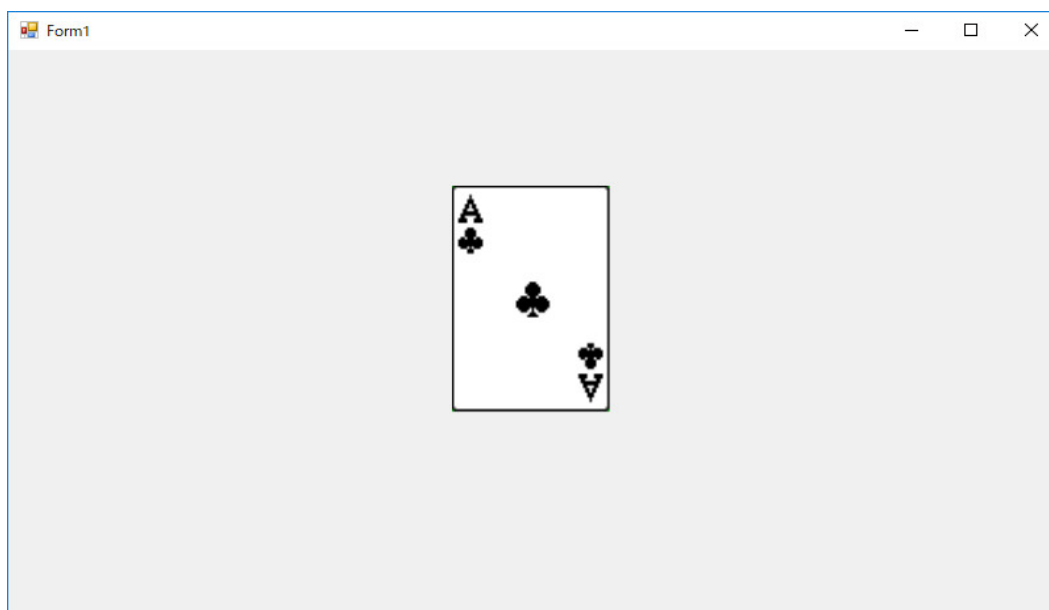
となります。この時、



のような画像が c1.png という名前であれば

```
private void pictureBox1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    Image image = Image.FromFile("c1.png");
    g.DrawImage(image, 0, 0, 120, 180);
}
```

とプログラミングすると、



を描いてくれます。これで良いなら VC++ のプログラミングの経験が生かされます。K とか Q とかの文字でなく、画像でカードを表示したいです。

今までのようにインターネットで無料素材のトランプの画像 (.gif でも .png でも .jpg でも良いです) を手に入れても良いですが、ここでは metapost を使って画像を自分で作ります。Windows に T_EX を導入しているとします。インターネットから必要なファイルを選んでインストールすることも出来るみたいですが、素人は奥村晴彦・黒木裕介著「改定第7版 LaTeX2 ϵ 美文書作成入門」の DVD-ROM を使ってインストールするのが楽です。以下で使っている metapost(mpost.exe) は L^AT_EX をインストールすると自動的にインストールされます。

```
def drawDia(expr x,y,s) =
pair d[];
d0 := (x-4*s, y);
d1 := (x, y+5*s);
d2 := (x+4*s, y);
d3 := (x, y-5*s);
fill d0--d1--d2--d3--cycle withcolor red;
enddef;
```

```
def drawInd(expr n) =
numeric r; picture p; string s;
r := 1;
s := "A2345678910JQK";
```

```

if n < 9:
    p:=substring(n,n+1)of s infont "ptmb8r" scaled 5r;
    draw p rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);
    draw p shifted (5mm, 120mm) withcolor (1,0,0);
else:
    if n = 9:
        p:=substring(n,n+2)of s infont "ptmb8r" scaled 5r;
        draw p xscaled 0.5 rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);
        draw p xscaled 0.5 shifted (5mm, 120mm) withcolor (1,0,0);
    else:
        p:=substring(n+1,n+2)of s infont "ptmb8r" scaled 5r;
        draw p rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);
        draw p shifted (5mm, 120mm) withcolor (1,0,0);
    fi
fi
drawDia(10mm, 110mm, 1mm);
drawDia(84mm, 35mm, 1mm);
enddef;

def drawFrame =
z0=(0mm, 0mm);
z1=(95mm, 0mm);
z2=(95mm, 146mm);
z3=(0mm, 146mm);
draw z0--z1--z2--z3--cycle;
enddef;

beginfig(1);
drawDia(47mm, 73mm, 2mm);
drawInd(0);
drawFrame;
endfig;
end

```

と適当なエディター（例えば、TeraPad）でプログラミングし、例えば、firstdia.mp
と名前を付け、コマンドブロックラインで

```
mpost firstdia.mp
```

とすると

```
コマンドプロンプト
c:\texsrc\metapost\meta>mpost firstdia.mp
This is MetaPost, version 1.803 (kpathsea version 6.1.1)
(mpost.mp (c:/texlive/2013/texmf-dist/metapost/base/plain.mp
Preloading the plain mem file, version 1.005) ) (./firstdia.mp [1] )
1 output file written: firstdia.1
Transcript written on firstdia.log.

c:\texsrc\metapost\meta>
```

のように、firstdia.1 という名前の画像を作ってくれます。

```
beginfig(1);
```

だから、firstdia.1 という名前で、

```
beginfig(2);
```

とすれば、firstdia.2 という名前になります。一つの mp ファイルで複数の絵を描くことが出来ます。この画像を見るためには、

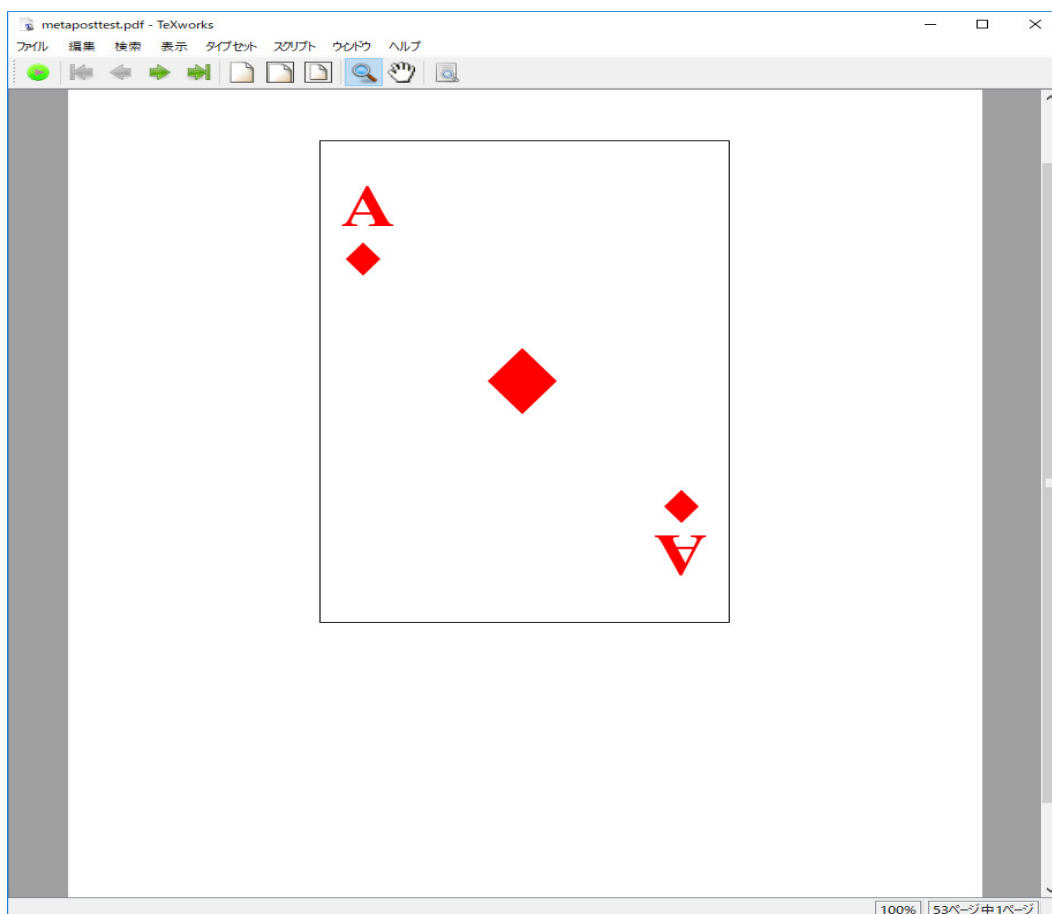
```
\documentclass[a4j,10pt]{jarticle}
\usepackage[dvipdfmx,hiresbb]{graphicx}
```

```
\pagestyle{plain}
```

```
%%%%%%%% TEXT START %%%%
\begin{document}
```

```
\begin{center}
\includegraphics[width=94mm,height=146mm]{firstdia.1}
\end{center}
\end{document}
```

という L^AT_EX のファイルを作り、コンパイルすれば

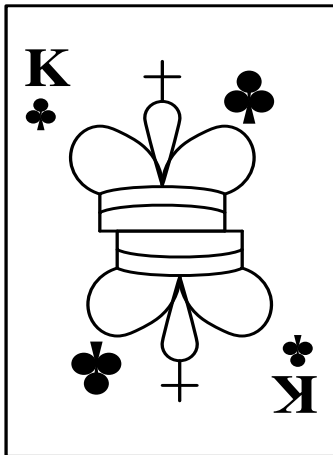
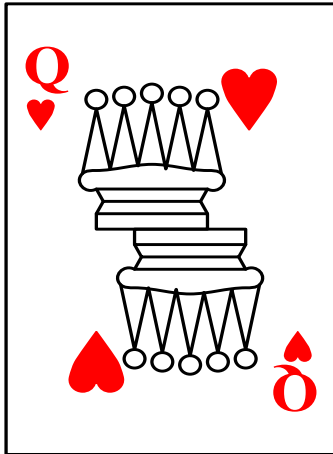
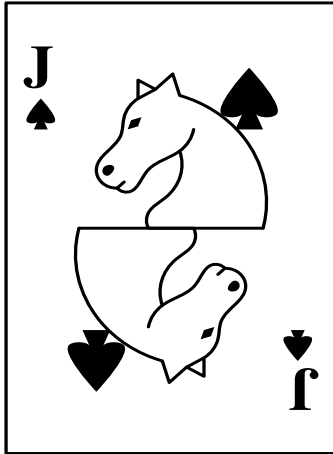


と表示してくれます。ダイヤのエースの札ができました。これを GIMP で切り取り処理して、「ファイル」「名前を付けてエクスポート」で.jpg ファイルなどを作れば良いです。

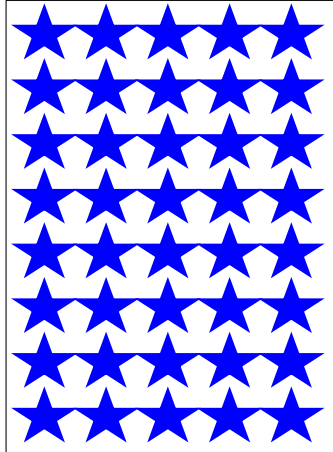
metapost のプログラミングは、metafont のプログラミングと殆ど同じで、文法は

John D. Hobby 著「METAPOST A USER'S MANUAL」

がインターネットから手に入りますが、文法は広範囲にわたり、正確な絵を簡単に描くために、細かな技法が使えるようになっていて複雑ですから、サンプル集をインターネットで探し、それを実際に打ち込み勉強するのがいいと思います。今の場合のように、何がしたいか目標がきちんと定まっているので、それに必要なことだけ勉強すればいいです。経験を積んでから、より良い方法がないか、文法書を精読する方が良いでしょう。最初から、完全な知識をと考えると失敗します。絵札でなければ、比較的感嘆ですが、市販のトランプのような絵札を描くのは大変です。ここでは



のように、チェスのナイト、クイーン、キングの絵文字を描くことにしました。自分で好みの絵にしてください。又、裏面は



としました。この metapost のプログラムは

```
def drawDia(expr x,y,s) =
pair d[];
d0 := (x-4*s, y);
d1 := (x, y+5*s);
d2 := (x+4*s, y);
d3 := (x, y-5*s);
fill d0--d1--d2--d3--cycle withcolor red;
enddef;

def drawSpade(expr x,y,s,f) =
path spade;
pair ss[];
ss0 := (0, 5*s);
ss1 := (4*s, -2.5*s);
ss2 := (s, -2.5*s);
ss3 := (2*s, -5*s);
spade := ss0{curl 0}..tension2..ss1..tension1.5..ss2--ss3;
spade := spade -- reverse spade xscaled -1 --cycle;
if f = 0:
fill spade shifted (x, y) withcolor black;
else:
fill spade rotated 180 shifted (x, y) withcolor black;
fi
enddef;

def drawHeart(expr x,y,s,f) =
```

```

path heart;
pair hh[];
hh0 := (0.1*s, -5*s);
hh1 := (3.2*s, 0);
hh2 := (4*s, 2.7*s);
hh3 := (2*s, 5*s);
hh4 := (0.1*s, 3.6*s);
heart := hh0{curl 0}..tension2..hh1..tension1.5..hh2..hh3..hh4;
heart := heart -- reverse heart xscaled -1 --cycle;
if f = 0:
  fill heart shifted (x, y) withcolor red;
else:
  fill heart rotated 180 shifted (x, y) withcolor red;
fi
enddef;

def drawClub(expr x,y,s,f) =
pair cc[];
cc0 := (0, 5*s);
cc1 := (1.8*s, 3.2*s);
cc2 := (0, 1.4*s);
cc3 := (-1.8*s, 3.2*s);

cc4 := (1.8*s, 1.8*s);
cc5 := (3.6*s, 0);
cc6 := (1.8*s, -1.8*s);
cc7 := (0, 0);

cc8 := (-1.8*s, 1.8*s);
cc9 := (-3.6*s, 0);
cc10 := (-1.8*s, -1.8*s);
cc11 := (0, 0);

cc12 := (0.9*s, -3.6*s);
cc13 := (0.1*s, 0);
cc14 := (0.3*s, 1.8*s);
cc15 := (-0.3*s, 1.8*s);
cc16 := (-0.1*s, 0);
cc17 := (-0.9*s, -3.6*s);

```

```

if f = 0:
  fill cc0..cc1..cc2..cc3..cycle shifted (x, y) withcolor black;
  fill cc4..cc5..cc6..cc7..cycle shifted (x, y) withcolor black;
  fill cc8..cc9..cc10..cc11..cycle shifted (x, y) withcolor black;
  fill cc12--cc13--cc14--cc15--cc16--cc17--cycle shifted (x, y)
    withcolor black;
else:
  fill cc0..cc1..cc2..cc3..cycle rotated 180 shifted (x, y) withcolor black;
  fill cc4..cc5..cc6..cc7..cycle rotated 180 shifted (x, y) withcolor black;
  fill cc8..cc9..cc10..cc11..cycle rotated 180 shifted (x, y) withcolor black;
  fill cc12--cc13--cc14--cc15--cc16--cc17--cycle rotated 180 shifted (x, y)
    withcolor black;
fi
enddef;

def drawStar(expr x,y) =
  pair A,B,C,D,E;
  A := (0,1cm);
  B := A rotated 72;
  C := B rotated 72;
  D := C rotated 72;
  E := D rotated 72;
  fill A--C--E--B--D--cycle shifted (x, y) withcolor blue;
enddef;

def drawInd(expr n) =
numeric r; picture p; string s;
r := 1;
s := "A2345678910JQK";
if n < 9:
  p:=substring(n,n+1)of s infont "ptmb8r" scaled 5r;
  draw p rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);
  draw p shifted (5mm, 120mm) withcolor (1,0,0);
else:
  if n = 9:
    p:=substring(n,n+2)of s infont "ptmb8r" scaled 5r;
    draw p xscaled 0.5 rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);
    draw p xscaled 0.5 shifted (5mm, 120mm) withcolor (1,0,0);

```

```

else:
  p:=substring(n+1,n+2)of s infont "ptmb8r" scaled 5r;
  draw p rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);
  draw p shifted (5mm, 120mm) withcolor (1,0,0);
fi
fi
drawDia(10mm, 110mm, 1mm);
drawDia(84mm, 35mm, 1mm);
enddef;

def drawIndSpade(expr n) =
numeric r; picture p; string s;
r := 1;
s := "A2345678910JQK";
if n < 9:
  p:=substring(n,n+1)of s infont "ptmb8r" scaled 5r;
  draw p rotated 180 shifted (90mm, 26mm) withcolor (0,0,0);
  draw p shifted (5mm, 120mm) withcolor (0,0,0);
else:
  if n = 9:
    p:=substring(n,n+2)of s infont "ptmb8r" scaled 5r;
    draw p xscaled 0.5 rotated 180 shifted (90mm, 26mm) withcolor (0,0,0);
    draw p xscaled 0.5 shifted (5mm, 120mm) withcolor (0,0,0);
  else:
    p:=substring(n+1,n+2)of s infont "ptmb8r" scaled 5r;
    draw p rotated 180 shifted (90mm, 26mm) withcolor (0,0,0);
    draw p shifted (5mm, 120mm) withcolor (0,0,0);
  fi
fi
drawSpade(10mm, 110mm, 1mm, 0);
drawSpade(84mm, 35mm, 1mm, 1);
enddef;

def drawIndHeart(expr n) =
numeric r; picture p; string s;
r := 1;
s := "A2345678910JQK";
if n < 9:
  p:=substring(n,n+1)of s infont "ptmb8r" scaled 5r;

```

```

draw p rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);
draw p shifted (5mm, 120mm) withcolor (1,0,0);
else:
  if n = 9:
    p:=substring(n,n+2)of s infont "ptmb8r" scaled 5r;
    draw p xscaled 0.5 rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);
    draw p xscaled 0.5 shifted (5mm, 120mm) withcolor (1,0,0);
  else:
    p:=substring(n+1,n+2)of s infont "ptmb8r" scaled 5r;
    draw p rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);
    draw p shifted (5mm, 120mm) withcolor (1,0,0);
  fi
fi
drawHeart(10mm, 110mm, 1mm, 0);
drawHeart(84mm, 35mm, 1mm, 1);
enddef;

def drawIndClub(expr n) =
numeric r; picture p; string s;
r := 1;
s := "A2345678910JQK";
if n < 9:
  p:=substring(n,n+1)of s infont "ptmb8r" scaled 5r;
  draw p rotated 180 shifted (90mm, 26mm) withcolor (0,0,0);
  draw p shifted (5mm, 120mm) withcolor (0,0,0);
else:
  if n = 9:
    p:=substring(n,n+2)of s infont "ptmb8r" scaled 5r;
    draw p xscaled 0.5 rotated 180 shifted (90mm, 26mm) withcolor (0,0,0);
    draw p xscaled 0.5 shifted (5mm, 120mm) withcolor (0,0,0);
  else:
    p:=substring(n+1,n+2)of s infont "ptmb8r" scaled 5r;
    draw p rotated 180 shifted (90mm, 26mm) withcolor (0,0,0);
    draw p shifted (5mm, 120mm) withcolor (0,0,0);
  fi
fi
drawClub(10mm, 110mm, 1.2mm, 0);
drawClub(84mm, 35mm, 1.2mm, 1);
enddef;

```

```

def drawFrame =
z0=(0mm, 0mm);
z1=(95mm, 0mm);
z2=(95mm, 146mm);
z3=(0mm, 146mm);
draw z0--z1--z2--z3--cycle;
enddef;

```

```

def drawKnight =
pair j[];
j0 := (54mm, 0mm);
j1 := (48mm, 30mm);
j2 := (30mm, 43mm);
j3 := (28mm, 50mm);
j4 := (23mm, 45mm);
j5 := (18mm, 42mm);
j6 := (14mm, 36mm);
j7 := (13mm, 30mm);
j8 := (6mm, 19mm);
j9 := (12mm, 13mm);
j10 := (14mm, 15mm);
j11 := (16mm, 12mm);
j12 := (18mm, 14mm);
j13 := (21mm, 19mm);
j14 := (30mm, 25mm);
j15 := (34mm, 32mm);
j16 := (26mm, 10mm);
j17 := (21mm, 5mm);
j18 := (21mm, 0mm);
j19 := (18mm, 48mm);
j20 := (15mm, 32mm);
j21 := (18mm, 33mm);
j22 := (19mm, 36mm);
j23 := (16mm, 35mm);
j24 := (9mm, 17mm);
j25 := (10.5mm, 18mm);
j26 := (11mm, 20mm);
j27 := (10mm, 20.5mm);

```

```

pickup pencircle scaled 0.01w;
path q[];
q1 = j0..j1..j2--j3--j4--j5..j6..j7..j8..j9--j10;
draw q1;
q2 = j9..j11..j12..j13..j14..j15;
draw q2;
q3 = j14..j16..j17..j18--j0;
draw q3;
q4 = j4--j19--j5;
draw q4;
q5 = j20--j21--j22--j23--cycle;
fill q5 withcolor black;
q6 = j24..j25..j26..j27..cycle;
fill q6 withcolor black;
picture knightpicture;
knightpicture := currentpicture;
currentpicture := nullpicture;
draw knightpicture shifted (20mm, 73mm);
draw knightpicture rotated 180 shifted (75mm, 73mm);
enddef;

```

```

def drawQueen =
pair q[];
q0 := (12mm, 0mm);
q1 := (44mm, 0mm);
q2 := (44mm, 5mm);
q3 := (12mm, 5mm);
q4 := (14mm, 9mm);
q5 := (42mm, 9mm);
q6 := (12mm, 13mm);
q7 := (44mm, 13mm);

q8 := (8mm, 18mm);
q9 := (12mm, 19mm);
q10 := (20mm, 20mm);
q11 := (28mm, 21mm);
q12 := (36mm, 20mm);
q13 := (44mm, 19mm);
q14 := (48mm, 18mm);

```

```

q15 := (12mm, 39.0mm);
q16 := (16mm, 19.5mm);
q17 := (20mm, 40mm);
q18 := (24mm, 20.5mm);
q19 := (28mm, 41mm);
q20 := (32mm, 19.5mm);
q21 := (36mm, 40mm);
q22 := (40mm, 19mm);
q23 := (44mm, 39.0mm);

pickup pencircle scaled 0.01w;

draw q0--q1--q2--q3--cycle;
draw q3--q4..q5--q2;
draw q4--q6..q7--q5;
draw q6..q8..q9..q10..q11..q12..q13..q14..q7;
draw q8--q15--q16--q17--q18--q19--q20--q21--q22--q23--q14;
draw fullcircle scaled 6mm shifted q15 shifted (0, 3mm);
draw fullcircle scaled 6mm shifted q17 shifted (0, 3mm);
draw fullcircle scaled 6mm shifted q19 shifted (0, 3mm);
draw fullcircle scaled 6mm shifted q21 shifted (0, 3mm);
draw fullcircle scaled 6mm shifted q23 shifted (0, 3mm);
picture queenpicture;
queenpicture := currentpicture;
currentpicture := nullpicture;
draw queenpicture shifted (14mm, 73mm);
draw queenpicture rotated 180 shifted (81mm, 73mm);
enddef;

def drawKing =
pair k[];
k0 := (15mm, 0mm);
k1 := (51mm, 0mm);
k2 := (15mm, 6mm);
k3 := (51mm, 6mm);
k4 := (15mm, 12mm);
k5 := (51mm, 12mm);

```



```

k6 := (7mm, 27mm);
k7 := (15mm, 34mm);
k8 := (21mm, 32mm);
k9 := (27mm, 28mm);
k10 := (33mm, 15mm);
k11 := (39mm, 28mm);
k12 := (45mm, 32mm);
k13 := (51mm, 34mm);
k14 := (59mm, 27mm);

k15 := (28mm, 36mm);
k16 := (33mm, 42mm);
k17 := (38mm, 36mm);

k18 := (33mm, 55mm);
k19 := (28mm, 50mm);
k20 := (38mm, 50mm);

pickup pencircle scaled 0.01w;
draw k0..tension 4..k1;
draw k0--k2{dir 30}..tension 2..k3--k1;
draw k2--k4{dir 30}..tension 2..k5--k3;
draw k4..k6..k7..k8..k9..k10;
draw k5..k14..k13..k12..k11..k10;
draw k10..tension 3..k15..k16..k17..tension 3..k10;
draw k16--k18;
draw k19--k20;
picture kingpicture;
kingpicture := currentpicture;
currentpicture := nullpicture;
draw kingpicture shifted (12mm, 73mm);
draw kingpicture rotated 180 shifted (83mm, 73mm);
enddef;

w = 95mm;

beginfig(1);
drawDia(47mm, 73mm, 2mm);

```

```

drawInd(0);
drawFrame;
endfig;
beginfig(2);
drawDia(47mm, 30mm, 2mm);
drawDia(47mm, 115mm, 2mm);
drawInd(1);
drawFrame;
endfig;
beginfig(3);
drawDia(47mm, 30mm, 2mm);
drawDia(47mm, 73mm, 2mm);
drawDia(47mm, 115mm, 2mm);
drawInd(2);
drawFrame;
endfig;
beginfig(4);
drawDia(26mm, 30mm, 2mm);
drawDia(26mm, 115mm, 2mm);
drawDia(70mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawInd(3);
drawFrame;
endfig;
beginfig(5);
drawDia(26mm, 30mm, 2mm);
drawDia(26mm, 115mm, 2mm);
drawDia(70mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawDia(47mm, 73mm, 2mm);
drawInd(4);
drawFrame;
endfig;
beginfig(6);
drawDia(26mm, 30mm, 2mm);
drawDia(26mm, 115mm, 2mm);
drawDia(70mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawDia(26mm, 73mm, 2mm);

```

```

drawDia(70mm, 73mm, 2mm);
drawInd(5);
drawFrame;
endfig;
beginfig(7);
drawDia(26mm, 30mm, 2mm);
drawDia(26mm, 115mm, 2mm);
drawDia(70mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawDia(26mm, 73mm, 2mm);
drawDia(70mm, 73mm, 2mm);
drawDia(47mm, 50mm, 2mm);
drawInd(6);
drawFrame;
endfig;
beginfig(8);
drawDia(26mm, 30mm, 2mm);
drawDia(26mm, 115mm, 2mm);
drawDia(70mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawDia(26mm, 73mm, 2mm);
drawDia(70mm, 73mm, 2mm);
drawDia(47mm, 50mm, 2mm);
drawDia(47mm, 95mm, 2mm);
drawInd(7);
drawFrame;
endfig;
beginfig(9);
drawDia(26mm, 30mm, 2mm);
drawDia(26mm, 115mm, 2mm);
drawDia(70mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawDia(26mm, 58mm, 2mm);
drawDia(70mm, 58mm, 2mm);
drawDia(26mm, 86mm, 2mm);
drawDia(70mm, 86mm, 2mm);
drawDia(47mm, 73mm, 2mm);
drawInd(8);
drawFrame;

```

```

endfig;
beginfig(10);
drawDia(26mm, 30mm, 2mm);
drawDia(26mm, 115mm, 2mm);
drawDia(70mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawDia(26mm, 58mm, 2mm);
drawDia(70mm, 58mm, 2mm);
drawDia(26mm, 86mm, 2mm);
drawDia(70mm, 86mm, 2mm);
drawDia(47mm, 44mm, 2mm);
drawDia(47mm, 101mm, 2mm);
drawInd(9);
drawFrame;
endfig;
beginfig(11);
drawKnight;
drawDia(26mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawInd(10);
drawFrame;
endfig;
beginfig(12);
drawQueen;
drawDia(26mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawInd(11);
drawFrame;
endfig;
beginfig(13);
drawKing;
drawDia(26mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawInd(12);
drawFrame;
endfig;

beginfig(14);
drawSpade(47mm, 73mm, 4mm, 0);

```

```

drawIndSpade(0);
drawFrame;
endfig;
beginfig(15);
drawSpade(47mm, 30mm, 2mm, 1);
drawSpade(47mm, 115mm, 2mm, 0);
drawIndSpade(1);
drawFrame;
endfig;
beginfig(16);
drawSpade(47mm, 30mm, 2mm, 1);
drawSpade(47mm, 73mm, 2mm, 0);
drawSpade(47mm, 115mm, 2mm, 0);
drawIndSpade(2);
drawFrame;
endfig;
beginfig(17);
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(26mm, 115mm, 2mm, 0);
drawSpade(70mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);
drawIndSpade(3);
drawFrame;
endfig;
beginfig(18);
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(26mm, 115mm, 2mm, 0);
drawSpade(70mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);
drawSpade(47mm, 73mm, 2mm, 0);
drawIndSpade(4);
drawFrame;
endfig;
beginfig(19);
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(26mm, 115mm, 2mm, 0);
drawSpade(70mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);
drawSpade(26mm, 73mm, 2mm, 0);

```

```

drawSpade(70mm, 73mm, 2mm, 0);
drawIndSpade(5);
drawFrame;
endfig;
beginfig(20);
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(26mm, 115mm, 2mm, 0);
drawSpade(70mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);
drawSpade(26mm, 73mm, 2mm, 0);
drawSpade(70mm, 73mm, 2mm, 0);
drawSpade(47mm, 50mm, 2mm, 1);
drawIndSpade(6);
drawFrame;
endfig;
beginfig(21);
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(26mm, 115mm, 2mm, 0);
drawSpade(70mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);
drawSpade(26mm, 73mm, 2mm, 0);
drawSpade(70mm, 73mm, 2mm, 0);
drawSpade(47mm, 50mm, 2mm, 1);
drawSpade(47mm, 95mm, 2mm, 0);
drawIndSpade(7);
drawFrame;
endfig;
beginfig(22);
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(26mm, 115mm, 2mm, 0);
drawSpade(70mm, 30mm, 2mm, 01);
drawSpade(70mm, 115mm, 2mm, 0);
drawSpade(26mm, 58mm, 2mm, 1);
drawSpade(70mm, 58mm, 2mm, 1);
drawSpade(26mm, 86mm, 2mm, 0);
drawSpade(70mm, 86mm, 2mm, 0);
drawSpade(47mm, 73mm, 2mm, 0);
drawIndSpade(8);
drawFrame;

```

```

endfig;
beginfig(23);
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(26mm, 115mm, 2mm, 0);
drawSpade(70mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);
drawSpade(26mm, 58mm, 2mm, 1);
drawSpade(70mm, 58mm, 2mm, 1);
drawSpade(26mm, 86mm, 2mm, 0);
drawSpade(70mm, 86mm, 2mm, 0);
drawSpade(47mm, 44mm, 2mm, 1);
drawSpade(47mm, 101mm, 2mm, 0);
drawIndSpade(9);
drawFrame;
endfig;
beginfig(24);
drawKnight;
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);
drawIndSpade(10);
drawFrame;
endfig;
beginfig(25);
drawQueen;
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);
drawIndSpade(11);
drawFrame;
endfig;
beginfig(26);
drawKing;
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);
drawIndSpade(12);
drawFrame;
endfig;

beginfig(27);
drawHeart(47mm, 73mm, 2mm, 0);

```

```

drawIndHeart(0);
drawFrame;
endfig;
beginfig(28);
drawHeart(47mm, 30mm, 2mm, 1);
drawHeart(47mm, 115mm, 2mm, 0);
drawIndHeart(1);
drawFrame;
endfig;
beginfig(29);
drawHeart(47mm, 30mm, 2mm, 1);
drawHeart(47mm, 73mm, 2mm, 0);
drawHeart(47mm, 115mm, 2mm, 0);
drawIndHeart(2);
drawFrame;
endfig;
beginfig(30);
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(26mm, 115mm, 2mm, 0);
drawHeart(70mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);
drawIndHeart(3);
drawFrame;
endfig;
beginfig(31);
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(26mm, 115mm, 2mm, 0);
drawHeart(70mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);
drawHeart(47mm, 73mm, 2mm, 0);
drawIndHeart(4);
drawFrame;
endfig;
beginfig(32);
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(26mm, 115mm, 2mm, 0);
drawHeart(70mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);
drawHeart(26mm, 73mm, 2mm, 0);

```



```

drawHeart(70mm, 73mm, 2mm, 0);
drawIndHeart(5);
drawFrame;
endfig;
beginfig(33);
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(26mm, 115mm, 2mm, 0);
drawHeart(70mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);
drawHeart(26mm, 73mm, 2mm, 0);
drawHeart(70mm, 73mm, 2mm, 0);
drawHeart(47mm, 50mm, 2mm, 1);
drawIndHeart(6);
drawFrame;
endfig;
beginfig(34);
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(26mm, 115mm, 2mm, 0);
drawHeart(70mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);
drawHeart(26mm, 73mm, 2mm, 0);
drawHeart(70mm, 73mm, 2mm, 0);
drawHeart(47mm, 50mm, 2mm, 1);
drawHeart(47mm, 95mm, 2mm, 0);
drawIndHeart(7);
drawFrame;
endfig;
beginfig(35);
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(26mm, 115mm, 2mm, 0);
drawHeart(70mm, 30mm, 2mm, 01);
drawHeart(70mm, 115mm, 2mm, 0);
drawHeart(26mm, 58mm, 2mm, 1);
drawHeart(70mm, 58mm, 2mm, 1);
drawHeart(26mm, 86mm, 2mm, 0);
drawHeart(70mm, 86mm, 2mm, 0);
drawHeart(47mm, 73mm, 2mm, 0);
drawIndHeart(8);
drawFrame;

```

```

endfig;
beginfig(36);
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(26mm, 115mm, 2mm, 0);
drawHeart(70mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);
drawHeart(26mm, 58mm, 2mm, 1);
drawHeart(70mm, 58mm, 2mm, 1);
drawHeart(26mm, 86mm, 2mm, 0);
drawHeart(70mm, 86mm, 2mm, 0);
drawHeart(47mm, 44mm, 2mm, 1);
drawHeart(47mm, 101mm, 2mm, 0);
drawIndHeart(9);
drawFrame;
endfig;
beginfig(37);
drawKnight;
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);
drawIndHeart(10);
drawFrame;
endfig;
beginfig(38);
drawQueen;
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);
drawIndHeart(11);
drawFrame;
endfig;
beginfig(39);
drawKing;
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);
drawIndHeart(12);
drawFrame;
endfig;

beginfig(40);
drawIndClub(0);

```

```

drawClub(47mm, 73mm, 2mm, 0);
drawFrame;
endfig;
beginfig(41);
drawClub(47mm, 30mm, 2mm, 1);
drawClub(47mm, 115mm, 2mm, 0);
drawIndClub(1);
drawFrame;
endfig;
beginfig(42);
drawClub(47mm, 30mm, 2mm, 1);
drawClub(47mm, 73mm, 2mm, 0);
drawClub(47mm, 115mm, 2mm, 0);
drawIndClub(2);
drawFrame;
endfig;
beginfig(43);
drawClub(26mm, 30mm, 2mm, 1);
drawClub(26mm, 115mm, 2mm, 0);
drawClub(70mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);
drawIndClub(3);
drawFrame;
endfig;
beginfig(44);
drawClub(26mm, 30mm, 2mm, 1);
drawClub(26mm, 115mm, 2mm, 0);
drawClub(70mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);
drawClub(47mm, 73mm, 2mm, 0);
drawIndClub(4);
drawFrame;
endfig;
beginfig(45);
drawClub(26mm, 30mm, 2mm, 1);
drawClub(26mm, 115mm, 2mm, 0);
drawClub(70mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);
drawClub(26mm, 73mm, 2mm, 0);

```

```

drawClub(70mm, 73mm, 2mm, 0);
drawIndClub(5);
drawFrame;
endfig;
beginfig(46);
drawClub(26mm, 30mm, 2mm, 1);
drawClub(26mm, 115mm, 2mm, 0);
drawClub(70mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);
drawClub(26mm, 73mm, 2mm, 0);
drawClub(70mm, 73mm, 2mm, 0);
drawClub(47mm, 50mm, 2mm, 1);
drawIndClub(6);
drawFrame;
endfig;
beginfig(47);
drawClub(26mm, 30mm, 2mm, 1);
drawClub(26mm, 115mm, 2mm, 0);
drawClub(70mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);
drawClub(26mm, 73mm, 2mm, 0);
drawClub(70mm, 73mm, 2mm, 0);
drawClub(47mm, 50mm, 2mm, 1);
drawClub(47mm, 95mm, 2mm, 0);
drawIndClub(7);
drawFrame;
endfig;
beginfig(48);
drawClub(26mm, 30mm, 2mm, 1);
drawClub(26mm, 115mm, 2mm, 0);
drawClub(70mm, 30mm, 2mm, 01);
drawClub(70mm, 115mm, 2mm, 0);
drawClub(26mm, 58mm, 2mm, 1);
drawClub(70mm, 58mm, 2mm, 1);
drawClub(26mm, 86mm, 2mm, 0);
drawClub(70mm, 86mm, 2mm, 0);
drawClub(47mm, 73mm, 2mm, 0);
drawIndClub(8);
drawFrame;

```

```

endfig;
beginfig(49);
drawClub(26mm, 30mm, 2mm, 1);
drawClub(26mm, 115mm, 2mm, 0);
drawClub(70mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);
drawClub(26mm, 58mm, 2mm, 1);
drawClub(70mm, 58mm, 2mm, 1);
drawClub(26mm, 86mm, 2mm, 0);
drawClub(70mm, 86mm, 2mm, 0);
drawClub(47mm, 44mm, 2mm, 1);
drawClub(47mm, 101mm, 2mm, 0);
drawIndClub(9);
drawFrame;
endfig;
beginfig(50);
drawKnight;
drawClub(26mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);
drawIndClub(10);
drawFrame;
endfig;
beginfig(51);
drawQueen;
drawClub(26mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);
drawIndClub(11);
drawFrame;
endfig;
beginfig(52);
drawKing;
drawClub(26mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);
drawIndClub(12);
drawFrame;
endfig;

beginfig(53);
for x=1 upto 5:

```

```

    for y=1 upto 8:
      drawStar(50*x-19, 50*y-17);
    endfor
  endfor
drawFrame;
endfig;

end

```

です。素人が作った長々と同じようなことが書いてあるプログラムで、本当はプログラムを短くする技法があると思いますが、一度しか使わないプログラムなので、これで良いことにします。コピー アンド ペーストで、同じことを繰り返しているだけなので、丸二日で作ることが出来ます。

```

\documentclass[a4j,12pt]{jarticle}

\usepackage[dvipdfmx, hiresbb]{graphicx}

\pagestyle{plain}

%%%%%   TEXT START   %%%%%

\begin{document}

\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.1}
\end{center}

\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.2}
\end{center}

\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.3}
\end{center}

\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.4}
\end{center}

```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.5}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.6}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.7}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.8}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.9}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.10}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.11}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.12}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.13}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.14}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.15}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.16}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.17}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.18}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.19}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.20}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.21}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.22}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.23}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.24}
\end{center}
```



```
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.25}  
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.26}  
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.27}  
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.28}  
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.29}  
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.30}  
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.31}  
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.32}  
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.33}  
\end{center}
```

```
\begin{center}
```

```
\includegraphics[width=0.3\linewidth,height=6cm]{dia.34}  
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.35}  
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.36}  
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.37}  
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.38}  
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.39}  
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.40}  
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.41}  
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.42}  
\end{center}
```

```
\begin{center}  
\includegraphics[width=0.3\linewidth,height=6cm]{dia.43}  
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.44}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.45}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.46}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.47}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.48}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.49}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.50}
\end{center}
```

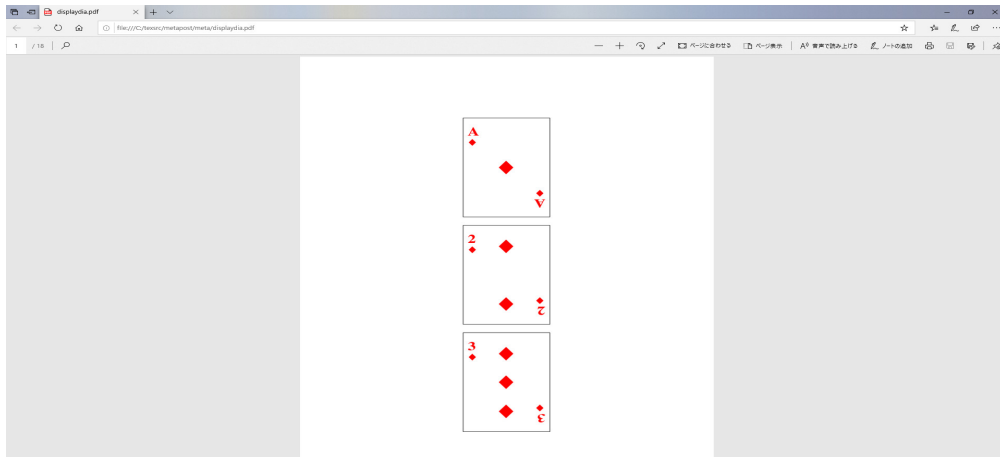
```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.51}
\end{center}
```

```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.52}
\end{center}
```

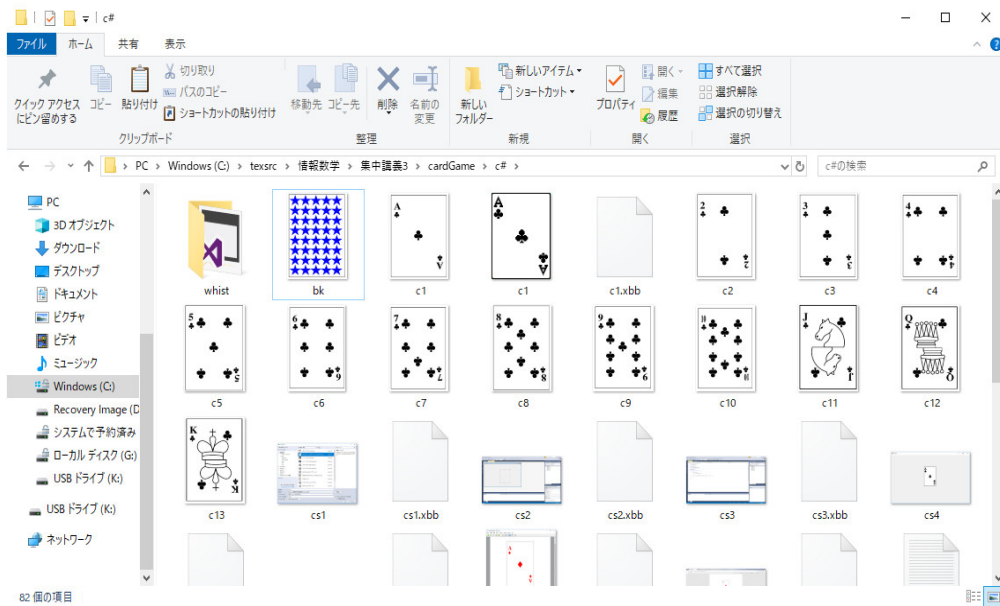
```
\begin{center}
\includegraphics[width=0.3\linewidth,height=6cm]{dia.53}
\end{center}
```

`\end{document}`

と L^AT_EX で原稿を作れば、

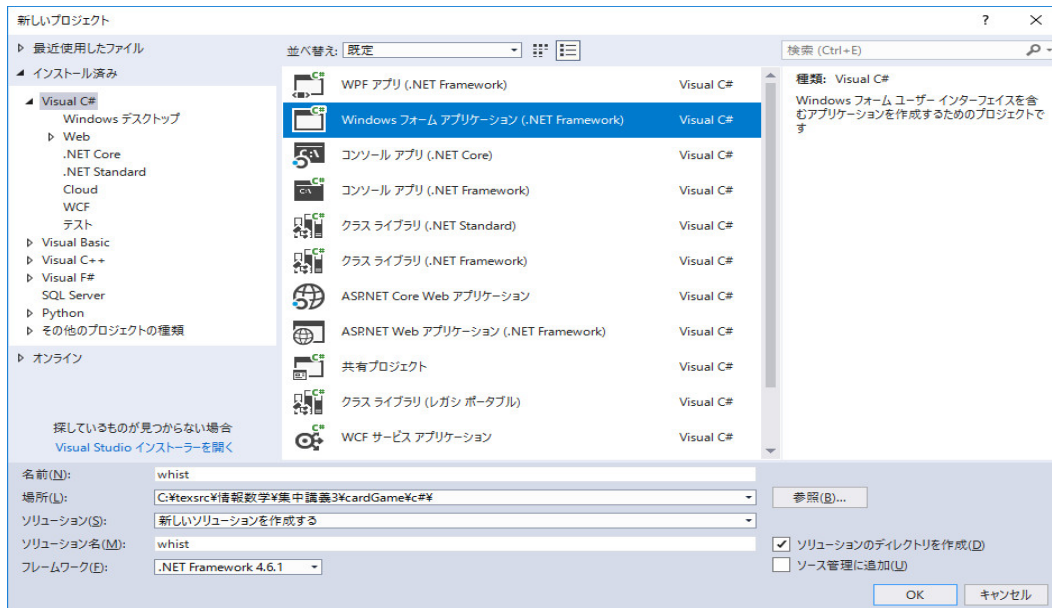


のように表示されるので、これを GIMP で切り取り、一枚ずつの.jpg の画像を作ります。



これで、オリジナルなカードの画像が作れたので、二人でプレイするカード・ゲーム「ジャーマン・ホイスト」を C# で作っていきます。本質は、他の言語で作ったものと同じです。

Microsoft Visual Sutudio を立ち上げ、C# の「ファイル」の「新規作成」の「プロジェクト」で



Windows フォームアプリケーションを選択します。ツールバーボックスから PictureBox を選び、ドラッグして適当な大きさにします。

まず、カードの情報を class で定義し、カードの集合を List で定義します。カードにはスーツ (♣ = 0, ◇ = 1, ♥ = 2, ♠ = 3) とランク (0, 1, 2, ..., 12) と裏表の画像をセットします。whist ではジョーカーを使わないので、ジョーカーの画像は作っていません。欲しければ、上の記述を参考に自分で画像を作ってください。

```
public class Card {
    public int suit, rank;
    public Image image, image2;
    public Card(int s, int r, Image img, Image img2) {
        this.suit = s;
        this.rank = r;
        this.image = img;
        this.image2 = img2;
    }
}

public Card card = null;
public List<Card> carddeck = new List<Card>();
```

C# で画像を読み込む方法がインターネットで調べてもよく分かりません。私ができる方法はとにかく一度コンパイルして、whist\bin\Debug 及び whist\bin\Release に画像のファイルをすべてコピーすると次のようなプログラムで、画像を表示できます。ここで、プログラム名を whist としています。

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace whist
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            for (int s=0; s<4; s++)
            {
                for (int r=0; r<13; r++)
                {
                    string name = "";
                    if (s == 0)
                    {
                        name = "c";
                    } else if (s == 1)
                    {
                        name = "d";
                    } else if (s == 2)
                    {
                        name = "h";
                    } else
                    {
                        name = "s";
                    }
                    name += r+1;
                    name += ".jpg";

                    Image img = Image.FromFile(name);
                }
            }
        }
    }
}

```

```

        Image img2 = Image.FromFile("bk.jpg");
        card = new Card(s, r, img, img2);
        cards.Add(card);
    }
}

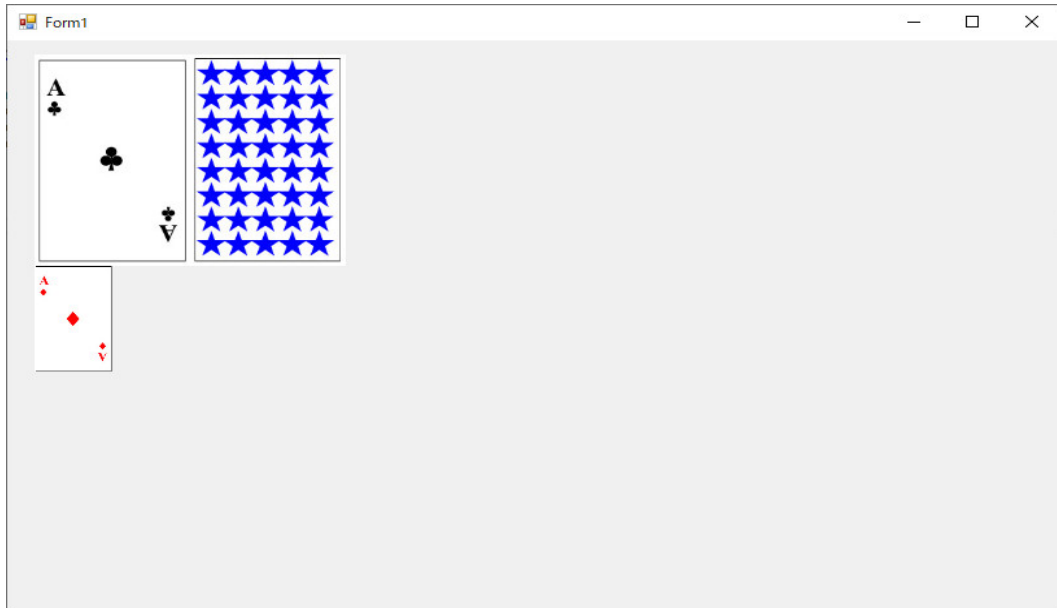
public class Card {
    public int suit, rank;
    public Image image, image2;
    public Card(int s, int r, Image img, Image img2) {
        this.suit = s;
        this.rank = r;
        this.image = img;
        this.image2 = img2;
    }
}

public Card card = null;
public List<Card> carddeck = new List<Card>();

private void pictureBox1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    g.DrawImage(carddeck[0].image, 0, 0, 120, 180);
    g.DrawImage(carddeck[0].image2, 120, 0, 120, 180);
    g.DrawImage(carddeck[13].image, 0, 180, 60, 90);
}
}
}

```

実行すると



となります。これで、ともかくカードゲームのプログラムを作る最初の難関をクリアしました。

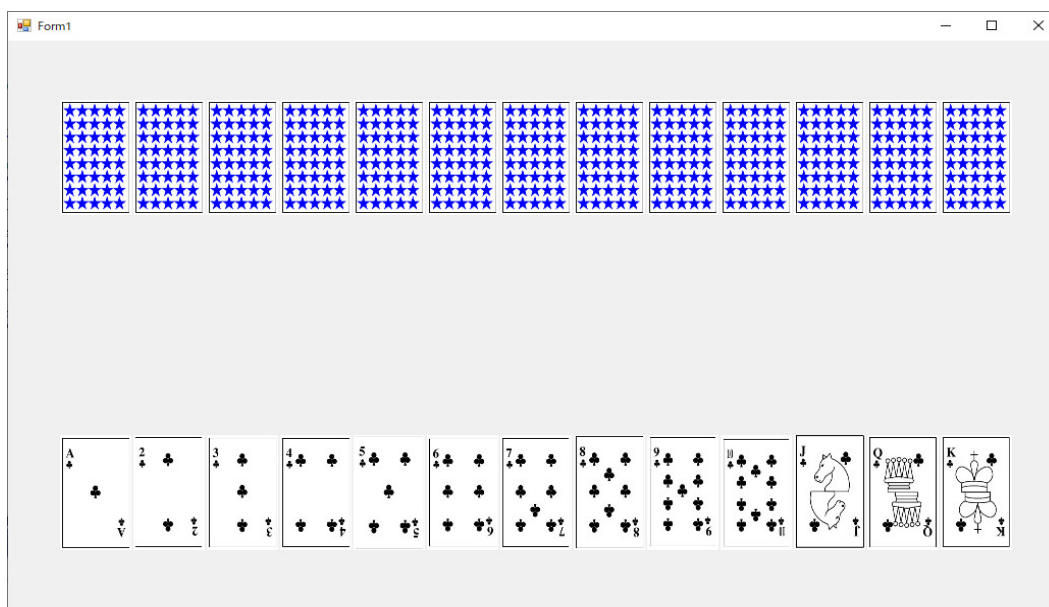
次に考えることは、pictureBox のサイズと表示するカードの画像のサイズの決定です。

Form のサイズを 1050, 640 に、pictureBox のサイズを 1000, 600 にセットします。

```
private void pictureBox1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    int pos = 30;

    for (int i = 0; i < 13; i++)
    {
        g.DrawImage(cards[i].image2, pos, 50, 70, 120);
        pos += 72;
    }
    pos = 30;
    for (int i = 0; i < 13; i++)
    {
        g.DrawImage(cards[i].image, pos, 400, 70, 120);
        pos += 72;
    }
}
```


で、



となります。このサイズでプログラミングします。

プログラム全体をオブジェクト指向プログラミングで作ることもできますが、それは Java のプログラムを参照してもらうことにし、ここでは普通のプログラミングをします。

必要な情報は既に作っているカード全体のリスト `carddeck` の他に、コンピュータの持ち札のリスト `comHand`、ユーザーの持ち札のリスト `userHand`、山のカードのリスト `yama`、コンピュータが手に入れたカード `getCom`、ユーザーが手に入れたカード `getUser`、リードされたカード `dispCom`、切り札のカード `trump`、ゲームが終わったかを示す `gameOverP`、コンピュータの手番かを示す `comPlayP` が必要です。

これらを追加します。

```
public List<Card> carddeck = new List<Card>();
public List<Card> comHand = new List<Card>();
public List<Card> userHand = new List<Card>();
public List<Card> yama = new List<Card>();
public Card getCom = null;
public Card getUser = null;
public Card dispCom = null;
public Card trump = null;
public bool gameOverP = false;
public bool comPlayP = true;
```

次に

```

private void card_initialize()
{
    Random r1 = new System.Random();
    int i = carddeck.Count();
    while (i > 0)
    {
        int s = r1.Next(0, i);
        Card temp = carddeck[i - 1];
        carddeck[i - 1] = carddeck[s];
        carddeck[s] = temp;
        i = i - 1;
    }
}

```

を定義し、Form1() に card_initialize(); を追加すると

```

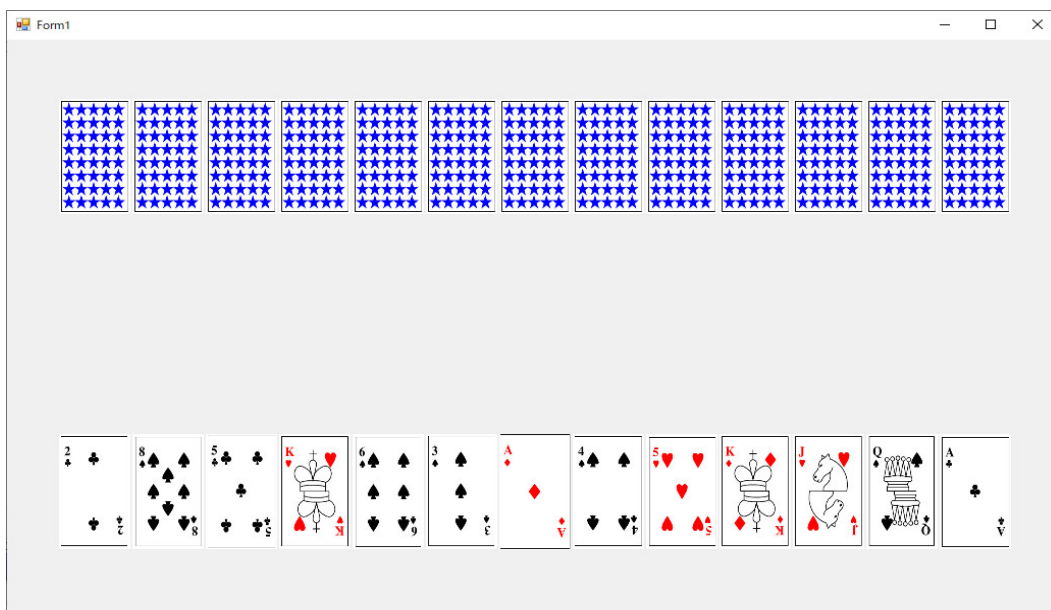
public Form1()
{
    InitializeComponent();
    for (int s=0; s<4; s++)
    {
        for (int r=0; r<13; r++)
        {
            string name = "";
            if (s == 0)
            {
                name = "c";
            } else if (s == 1)
            {
                name = "d";
            } else if (s == 2)
            {
                name = "h";
            } else
            {
                name = "s";
            }
            name += r+1;
            name += ".jpg";
        }
    }
}

```

```

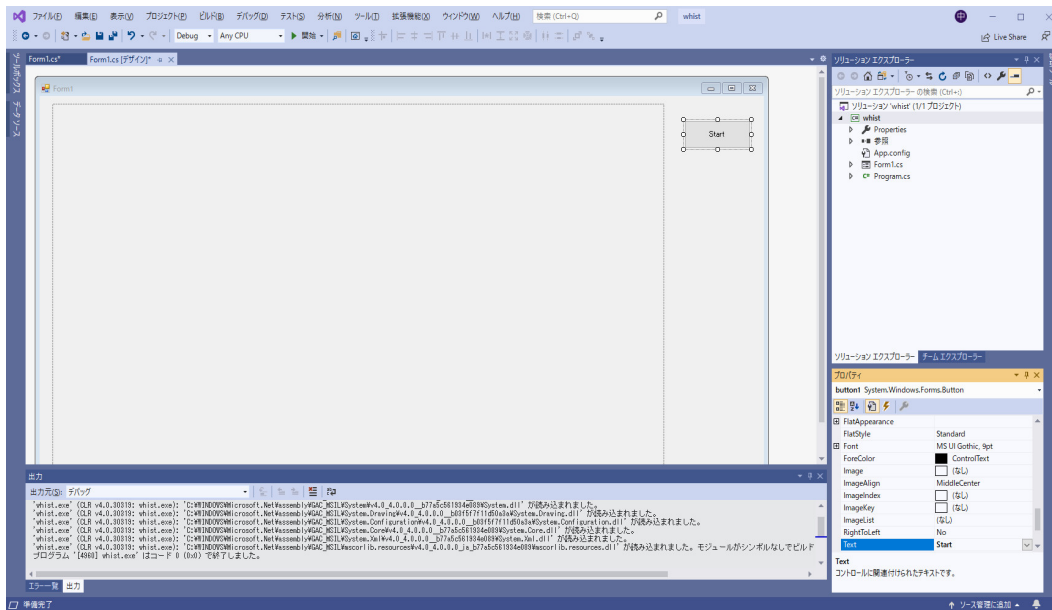
        Image img = Image.FromFile(name);
        Image img2 = Image.FromFile("bk.jpg");
        card = new Card(s, r, img, img2);
        carddeck.Add(card);
    }
}
card_initialize();
}

```



となります。carddeck がシャッフルされました。

ここで、C++ のプログラムに合わせて、次のように修正します。



と Form のサイズを大きくし、ツールボックスから Button を選び、ドラッグして Form に貼り付けます。button1 の Text を Start にします。button1 をダブルクリックします。

```
private void button1_Click(object sender, EventArgs e)
{
}
}
```

が自動的に書き込まれます。ここに

```
private void button1_Click(object sender, EventArgs e)
{
    gameInitialize();
    ShowBan();
}
}
```

と書き込み、手続き ShowBan() を

```
private void ShowBan()
{
    Graphics g = pictureBox1.CreateGraphics();

    // Create solid brush.
```

```

SolidBrush whiteBrush = new SolidBrush(Color.White);

// Create rectangle.
Rectangle rect = new Rectangle(0, 0, 1000, 600);

// Fill rectangle to screen.
g.FillRectangle(whiteBrush, rect);

int pos = 30;

for (int i = 0; i < 13; i++)
{
    g.DrawImage(comHand[i].image2, pos, 50, 70, 120);
    pos += 72;
}
pos = 30;
for (int i = 0; i < 13; i++)
{
    g.DrawImage(userHand[i].image, pos, 400, 70, 120);
    pos += 72;
}

if (yama?.Count > 0)
{
    g.DrawImage(yama[0].image, 30, 225, 70, 120);
}
}

```

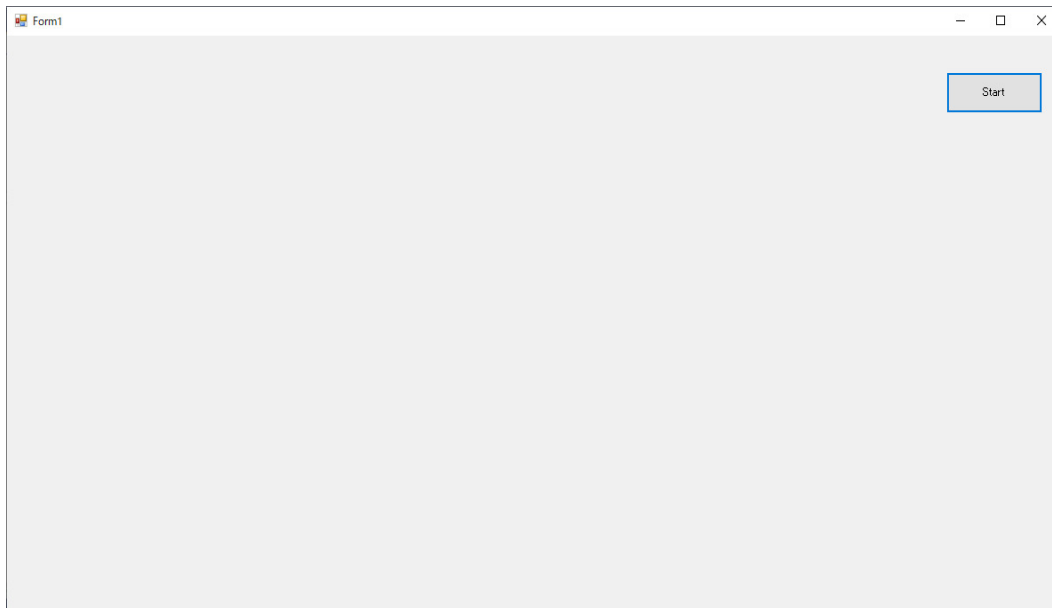
と定義します。一行目が修正しただけで、他は `pictureBox1_Paint(object sender, PaintEventArgs e)` に書いていたままです。 `pictureBox1_Paint(object sender, PaintEventArgs e)` は

```

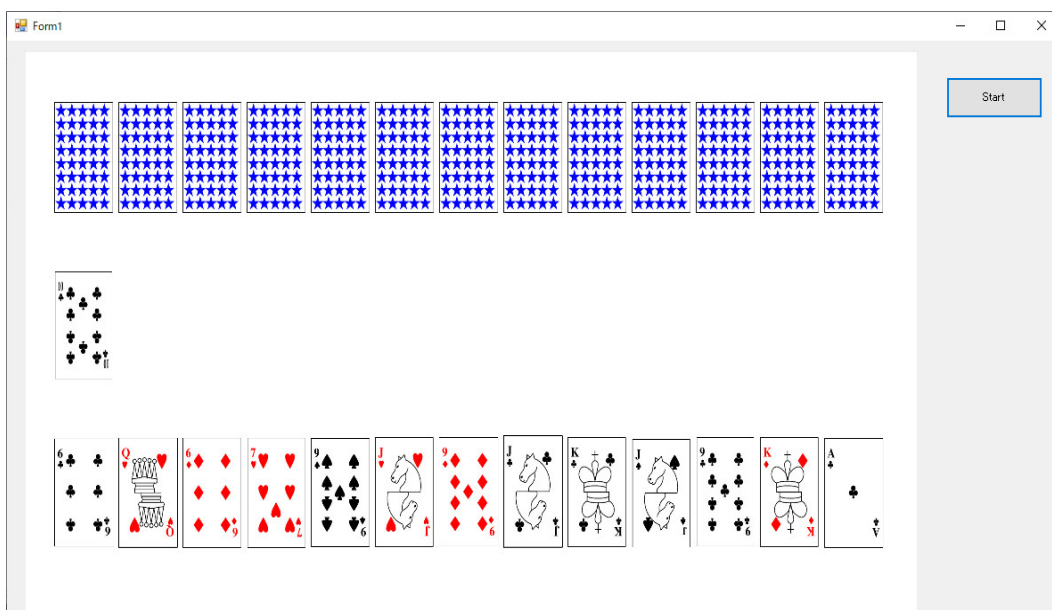
private void pictureBox1_Paint(object sender, PaintEventArgs e)
{
}

```

と空白にします。ビルドし、実行すると



となります。Start のボタンをクリックすると



となります。

画面の設計を間違えていました。獲得したカードを表示するスペースを忘れていました。縦にカードを5列表示しなければならないことを忘れていました。

```
private void ShowBan()  
{  
    Graphics g = pictureBox1.CreateGraphics();
```

```

// Create solid brush.
SolidBrush whiteBrush = new SolidBrush(Color.White);

// Create rectangle.
Rectangle rect = new Rectangle(0, 0, 1000, 600);

// Fill rectangle to screen.
g.FillRectangle(whiteBrush, rect);

int pos = 30;

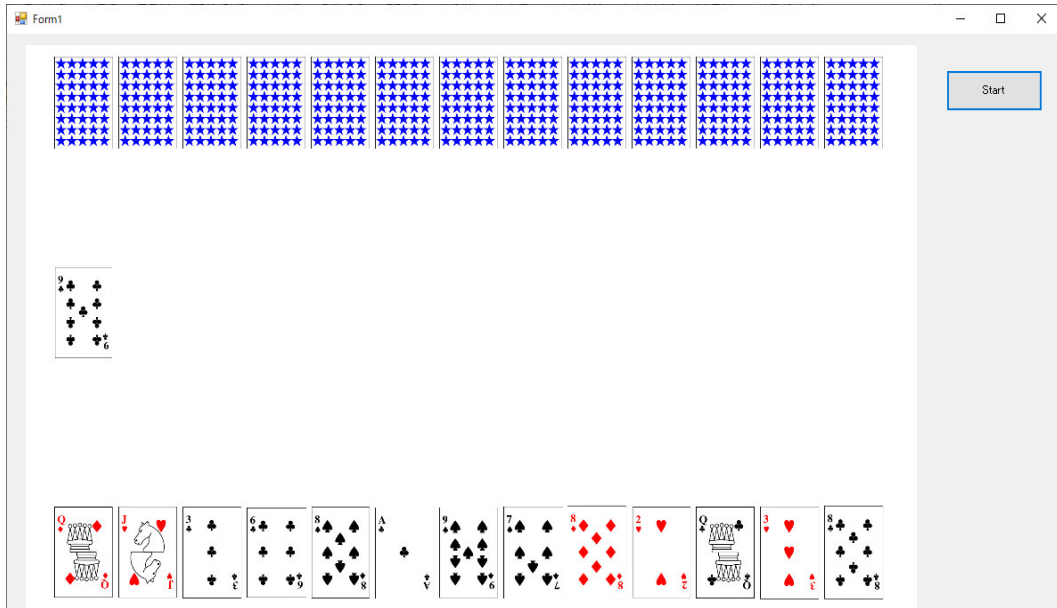
for (int i = 0; i < 13; i++)
{
    g.DrawImage(comHand[i].image2, pos, 10, 70, 100);
    pos += 72;
}
pos = 30;
for (int i = 0; i < 13; i++)
{
    g.DrawImage(userHand[i].image, pos, 480, 70, 100);
    pos += 72;
}

if (yama?.Count > 0)
{
    g.DrawImage(yama[0].image, 30, 250, 70, 100);
}
}

```

と修正します。

Start のボタンをクリックすると



となります。

更に、getCom と getUser のデータ構造も間違っています。C++ のプログラムを作ったのはずいぶん前のことで、Python のプログラムを見ながらプログラミングを始めたので失敗しました。getCom と getUser はコンピュータとユーザーが獲得したカードを保持しなければならないので、これらも Card のリストでなければなりません。

データの宣言部分を

```
public Card card = null;
public List<Card> carddeck = new List<Card>();
public List<Card> comHand = new List<Card>();
public List<Card> userHand = new List<Card>();
public List<Card> yama = new List<Card>();
public List<Card> getCom = new List<Card>();
public List<Card> getUser = new List<Card>();
public Card dispCom = null;
public Card trump = null;
public bool gameOverP = false;
public bool comPlayP = true;
```

と修正します。6行目と7行目を修正しています。

手続き gameInitialize() を

```
private void gameInitialize()
{
```



```

comHand.Clear();
userHand.Clear();
yama.Clear();

for (int i=0; i<13; i++)
{
    comHand.Add(carddeck[i]);
}
for (int i = 13; i < 26; i++)
{
    userHand.Add(carddeck[i]);
}
for (int i = 26; i < 52; i++)
{
    yama.Add(carddeck[i]);
}

getCom.Clear();
getUser.Clear();
dispCom = null;
gameOverP = false;
comPlayP = false;
trump = yama[0];
}

```

と修正します。最後の行を削除し、`getCom.Clear()`、`getUser.Clear()`と修正しています。

ユーザーが先手ですから、カードを選んでリードするよう促すメッセージを表示するようにします。

```

if (gameOverP == false)
{
    if (comPlayP == false)
    {
        //フォントオブジェクトの作成
        Font fnt = new Font("MS UI Gothic", 20);
        //文字列を位置 (0,0)、青色で表示
        g.DrawString("カードを選んでください",
            fnt, Brushes.Blue, 200, 400);
    }
}

```

```

        //リソースを解放する
        fnt.Dispose();
    }
} else
{
    //フォントオブジェクトの作成
    Font fnt = new Font("MS UI Gothic", 20);
    //文字列を位置 (0,0)、青色で表示
    g.DrawString("ゲームオーバー",
        fnt, Brushes.Blue, 200, 400);

    //リソースを解放する
    fnt.Dispose();
}
g.Dispose();

```

を ShowBan() の最後に追加します。更に、ShowBan() がゲーム途中の情報（それぞれが獲得したカード：すなわち getCom と getUser の中身）も表示できるように

```

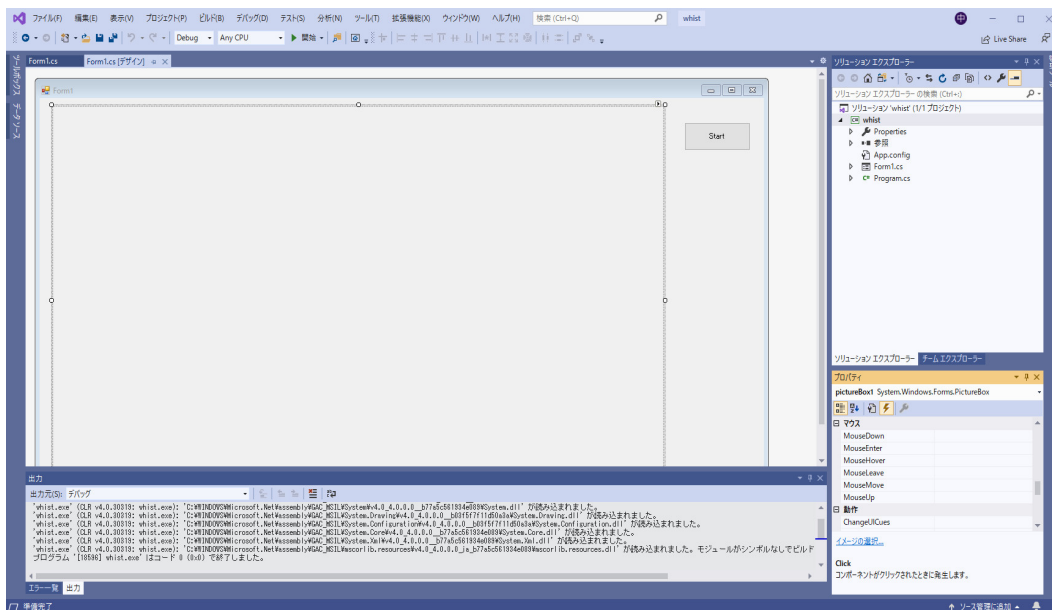
if (getCom?.Count > 0)
{
    for (int i = 0; i < getCom.Count(); i++)
    {
        g.DrawImage(getCom[i].image, 30+36*i, 130, 70, 100);
    }
}
if (getUser?.Count > 0)
{
    for (int i = 0; i < getUser.Count(); i++)
    {
        g.DrawImage(getUser[i].image, 30 + 36 * i, 370, 70, 100);
    }
}

```

を上プログラムの直前に追加します。

次に、ユーザーがカードを出す時、マウスで出すカードを選択できるようにします。

「デザイン」のページを選択し、pictureBox をクリックし、「プロパティ」のボックスの上部の左から4番目のアイコンをクリックします。「マウス」の項目を表示します。



となります。MouseDown をダブルクリックします。

```
private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
}
}
```

が自動的に作られます。

デバッグのために、データの宣言部分の最後に

```
public Card selectCard = null;
public Card selectCard2 = null;
int userIndex = -1;
int comIndex = -1;
```

を追加し、ShowBan() の最後の行の上に

```
if (selectCard != null)
{
    g.DrawImage(selectCard.image, 500, 250, 70, 100);
}
if (selectCard2 != null)
{
    g.DrawImage(selectCard2.image, 700, 250, 70, 100);
}
```

を追加し、

```

private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    selectCard = null;
    selectCard2 = null;

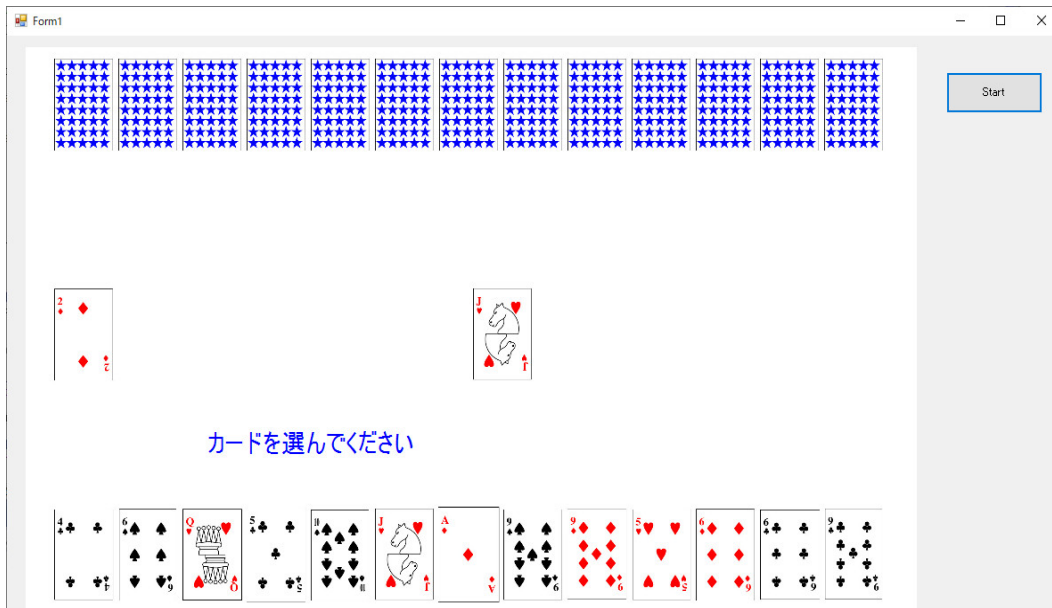
    int mX = e.X;
    int mY = e.Y;
    userIndex = -1;
    int pos = 29;
    for (int i = 0; i < userHand.Count(); i++)
    {
        if ((mX > pos && mX < pos + 72) && (mY > 480 && mY < 580))
        {
            userIndex = i;
            break;
        }
        pos += 72;
    }
    if (userIndex < 0)
        return;

    selectCard = userHand[userIndex];

    ShowBan();
}
}

```

と書き込みます。実行し、Start のボタンをクリックしとマウスで下段のカードをクリックすると



のようになります。プログラムが正しいみたいです。pictureBox1_MouseDown()の最後の行の ShowBan(); を削除します。

マウスで出すカードを選んだ後の処理のプログラミングを続けます。

ユーザーがカードを出すのは、ユーザーの手番で、ユーザーがリードした場合とコンピュータの手番で、コンピュータのリードに対してカードを出す場合があります。ゲームの最初はユーザーの手番ですから、まず、ユーザーの手番で、ユーザーがリードした場合の処理をプログラミングします。どちらの手番かは comPlayP == true なら、コンピュータの手番、comPlayP == false なら、ユーザーの手番と約束します。ユーザーがリードするとコンピュータが出すカードを決めなくてはなりません。強くするためには、シミュレーションしてみても、手を決めるなど、AI の深層学習などの勉強が必要みたいですが、抽象的な議論の本ばかりで、私が読んでも理解できる具体的な例を書いてくれている本をまだ見つけてないので、単純なアルゴリズムで、ルールに反しないコンピュータの手を決めることにします。

pictureBox1_MouseDown() を

```
private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    selectCard = null;
    selectCard2 = null;

    int mX = e.X;
    int mY = e.Y;
    userIndex = -1;
    int pos = 29;
    for (int i = 0; i < userHand.Count(); i++)
```

```

{
    if ((mX > pos && mX < pos + 72) && (mY > 480 && mY < 580))
    {
        userIndex = i;
        break;
    }
    pos += 72;
}
if (userIndex < 0)
    return;

selectCard = userHand[userIndex];

comPlayP = false;

if (!comPlayP) // ユーザーの手番
{
    int suit = userHand[userIndex].suit;
    int rank = userHand[userIndex].rank;
    comIndex = -1;
    // リードされたスーツの手札があれば、それを出す
    for (int i = 0; i < comHand.Count(); i++)
    {
        if (comHand[i].suit == suit)
        {
            comIndex = i;
            break;
        }
    }
    if (comIndex < 0) // リードされたスーツの手札が無かった
    {
        // 切り札のスーツの手札があれば、それを出す
        for (int i = 0; i < comHand.Count(); i++)
        {
            if (comHand[i].suit == trump.suit)
            {
                comIndex = i;
                break;
            }
        }
    }
}

```

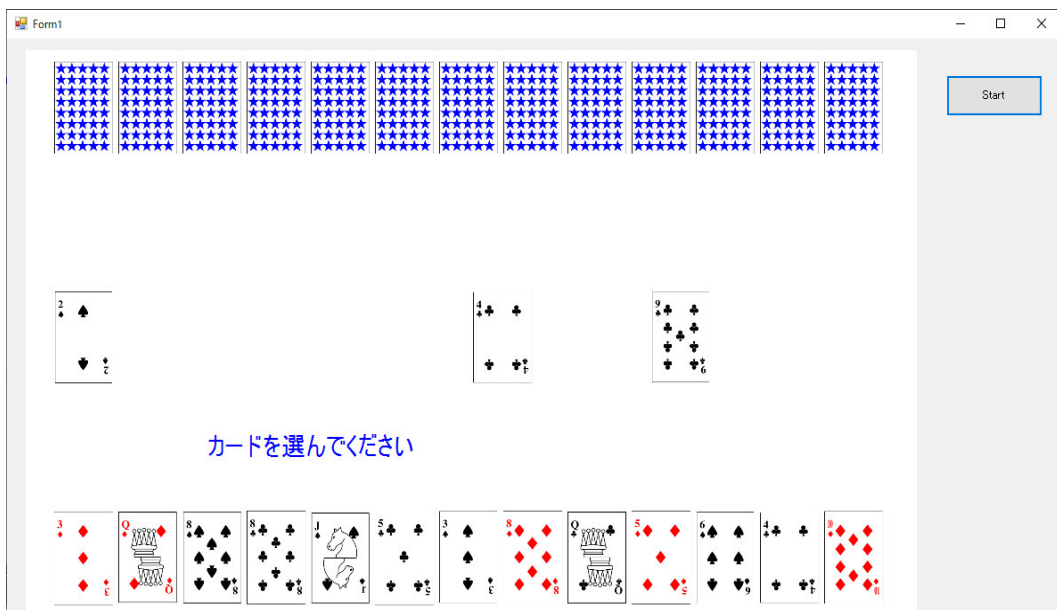
```

    }
}
if (comIndex < 0) // 切り札のスーツの手札が無かった
{
    // 乱数で手を選ぶ
    Random r1 = new System.Random();
    int i = comHand.Count();
    comIndex = r1.Next(0, i);
}
selectCard2 = comHand[comIndex];
}
ShowBan();
}
}
}

```

と修正します。

実行し、Start のボタンをクリックしとマウスで下段のカードをクリックすると



のようになります。今のところ、プログラムが正しいみたいです。

次に、ユーザーの出した手とコンピュータの出した手を比較し、どちらが勝ったかを判断する関数

```

private bool comWinP(Card ccard, Card ucard)
{
    if (ccard.suit == ucard.suit)
    {

```

```

        if (ccard.rank == 0)
        {
            return true;
        }
        else if (ucard.rank == 0)
        {
            return false;
        }
        else if (ccard.rank > ucard.rank)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    else if (ccard.suit == trump.suit)
    {
        return true;
    }
    else if (ucard.suit == trump.suit)
    {
        return false;
    }
    else if (comPlayP)
    {
        return true;
    }
    else
    {
        return false;
    }
}

```

を作り、pictureBox1_MouseDown() を

```

private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    selectCard = null;

```



```

selectCard2 = null;

int mX = e.X;
int mY = e.Y;
userIndex = -1;
int pos = 29;
for (int i = 0; i < userHand.Count(); i++)
{
    if ((mX > pos && mX < pos + 72) && (mY > 480 && mY < 580))
    {
        userIndex = i;
        break;
    }
    pos += 72;
}
if (userIndex < 0)
    return;

selectCard = userHand[userIndex];

comPlayP = false;

if (!comPlayP) // ユーザーの手番
{
    int suit = userHand[userIndex].suit;
    int rank = userHand[userIndex].rank;
    comIndex = -1;
    // リードされたスーツの手札があれば、それを出す
    for (int i = 0; i < comHand.Count(); i++)
    {
        if (comHand[i].suit == suit)
        {
            comIndex = i;
            break;
        }
    }
    if (comIndex < 0) // リードされたスーツの手札が無かった
    {
        // 切り札のスーツの手札があれば、それを出す
    }
}

```

```

    for (int i = 0; i < comHand.Count(); i++)
    {
        if (comHand[i].suit == trump.suit)
        {
            comIndex = i;
            break;
        }
    }
}
if (comIndex < 0) // 切り札のスーツの手札が無かった
{
    // 乱数で手を選ぶ
    Random r1 = new System.Random();
    int i = comHand.Count();
    comIndex = r1.Next(0, i);
}
selectCard2 = comHand[comIndex];
comHand.RemoveAt(comIndex);
userHand.RemoveAt(userIndex);
if (comWinP(selectCard2, selectCard))
{
    getCom.Add(selectCard2);
    getCom.Add(selectCard);
    if (yama?.Count > 0)
    {
        comHand.Add(yama[0]);
        userHand.Add(yama[1]);
        yama.RemoveAt(0);
        yama.RemoveAt(0);
    }
    comPlayP = true;
}
else
{
    getUser.Add(selectCard2);
    getUser.Add(selectCard);
    if (yama?.Count > 0)
    {
        userHand.Add(yama[0]);
    }
}

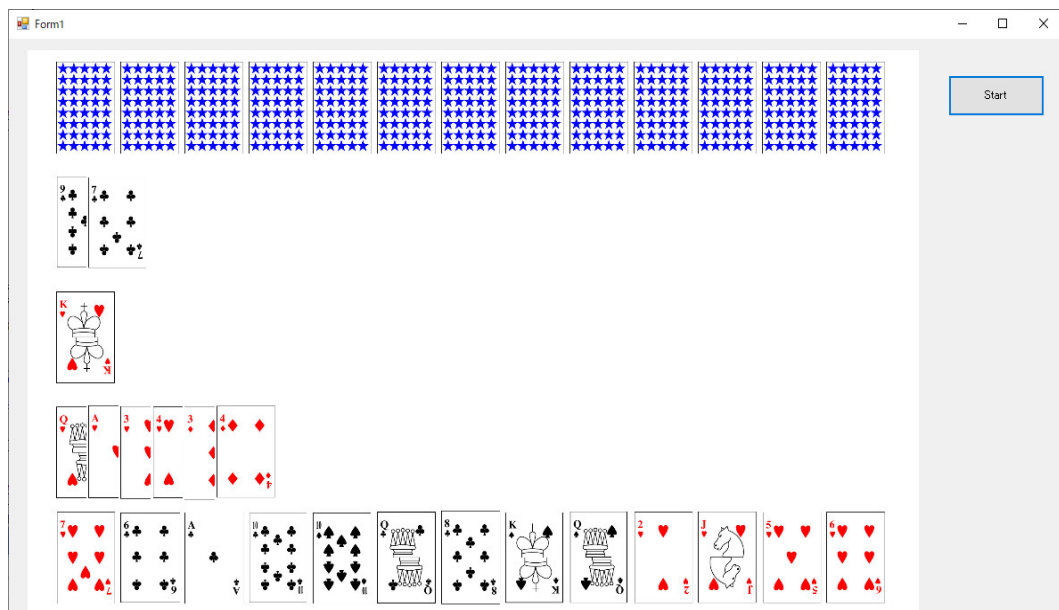
```

```

        comHand.Add(yama[1]);
        yama.RemoveAt(0);
        yama.RemoveAt(0);
    }
    comPlayP = true;
}
selectCard = null;
selectCard2 = null;
}
ShowBan();
}

```

と修正します。実行し、Start のボタンをクリックしとマウスで下段のカードをユーザーが勝つ限りクリックすると



のようになります。今のところ、プログラムが正しいみたいです。

次に、コンピュータの手番で、コンピュータのリードに対して、ユーザーがカードを出した時の処理をプログラミングします。残りを一気にプログラミングします。dispCom にはコンピュータのリードしたカードをセットします。

まず、pictureBox1_MouseDown() の上の方にある

```
comPlayP = false;
```

を削除します。次に、ShowBan() の

```

if (selectCard != null)
{
    g.DrawImage(selectCard.image, 500, 250, 70, 100);
}
if (selectCard2 != null)
{
    g.DrawImage(selectCard2.image, 700, 250, 70, 100);
}

```

を削除します。更に、手札が少なくなった場合のことを考えていませんでした。
 ShowBan() の

```

for (int i = 0; i < 13; i++)
{
    g.DrawImage(comHand[i].image2, pos, 10, 70, 100);
    pos += 72;
}
pos = 30;
for (int i = 0; i < 13; i++)
{
    g.DrawImage(userHand[i].image, pos, 480, 70, 100);
    pos += 72;
}

```

を

```

for (int i = 0; i < comHand.Count(); i++)
{
    g.DrawImage(comHand[i].image2, pos, 10, 70, 100);
    pos += 72;
}
pos = 30;
for (int i = 0; i < userHand.Count(); i++)
{
    g.DrawImage(userHand[i].image, pos, 480, 70, 100);
    pos += 72;
}

```

と修正します。次に、pictureBox1_MouseDown() を

```

private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    selectCard = null;

    int mX = e.X;
    int mY = e.Y;
    userIndex = -1;
    int pos = 29;
    for (int i = 0; i < userHand.Count(); i++)
    {
        if ((mX > pos && mX < pos + 72) && (mY > 480 && mY < 580))
        {
            userIndex = i;
            break;
        }
        pos += 72;
    }
    if (userIndex < 0)
        return;

    selectCard = userHand[userIndex];

    if (!comPlayP) // ユーザーの手番
    {

        int suit = userHand[userIndex].suit;
        int rank = userHand[userIndex].rank;

        comIndex = -1;
        // リードされたスーツの手札があれば、それを出す
        for (int i = 0; i < comHand.Count(); i++)
        {
            if (comHand[i].suit == suit)
            {
                comIndex = i;
                break;
            }
        }
        if (comIndex < 0) // リードされたスーツの手札が無かった
    }
}

```

```

{
    // 切り札のスーツの手札があれば、それを出す
    for (int i = 0; i < comHand.Count(); i++)
    {
        if (comHand[i].suit == trump.suit)
        {
            comIndex = i;
            break;
        }
    }
}
if (comIndex < 0) // 切り札のスーツの手札が無かった
{
    // 乱数で手を選ぶ
    Random r1 = new System.Random();
    int i = comHand.Count();
    comIndex = r1.Next(0, i);
}
selectCard2 = comHand[comIndex];
comHand.RemoveAt(comIndex);
userHand.RemoveAt(userIndex);
if (comWinP(selectCard2, selectCard))
{
    getCom.Add(selectCard2);
    getCom.Add(selectCard);
    if (yama?.Count > 0)
    {
        comHand.Add(yama[0]);
        userHand.Add(yama[1]);
        yama.RemoveAt(0);
        yama.RemoveAt(0);
    }
    comPlayP = true;
}
else
{
    getUser.Add(selectCard2);
    getUser.Add(selectCard);
    if (yama?.Count > 0)

```

```

        {
            userHand.Add(yama[0]);
            comHand.Add(yama[1]);
            yama.RemoveAt(0);
            yama.RemoveAt(0);
        }
        comPlayP = false;
    }
    selectCard = null;
    selectCard2 = null;
    dispCom = null;
} else
{
    // ユーザーの手がルールを守っているか?
    int suit = userHand[userIndex].suit;
    int rank = userHand[userIndex].rank;
    if (suit != dispCom.suit)
    {
        bool flag = false;
        for (int i=0; i<userHand.Count(); i++)
        {
            if (userHand[i].suit == dispCom.suit)
            {
                flag = true; break;
            }
        }
        if (flag)
        {
            string str = "リードは";
            switch (dispCom.suit)
            {
                case 0: str += "クラブ"; break;
                case 1: str += "ダイヤ"; break;
                case 2: str += "ハート"; break;
                case 3: str += "スペード"; break;
            }
            str += "です";
            MessageBox.Show(str);
            selectCard = null;
        }
    }
}

```

```

        return;
    }
}
if (comWinP(dispCom, selectCard))
{
    getCom.Add(selectCard2);
    getCom.Add(selectCard);
    if (yama?.Count > 0)
    {
        comHand.Add(yama[0]);
        userHand.Add(yama[1]);
        yama.RemoveAt(0);
        yama.RemoveAt(0);
    }
    comPlayP = true;
}
else
{
    getUser.Add(selectCard);
    getUser.Add(selectCard2);
    if (yama?.Count > 0)
    {
        userHand.Add(yama[0]);
        comHand.Add(yama[1]);
        yama.RemoveAt(0);
        yama.RemoveAt(0);
    }
    comPlayP = false;
}
comHand.RemoveAt(comIndex);
userHand.RemoveAt(userIndex);
selectCard = null;
selectCard2 = null;
dispCom = null;
}
if (comHand.Count() == 0)
{
    gameOverP = true;
} else if (comPlayP)

```



```

    {
        // 乱数で手を選ぶ
        Random r1 = new System.Random();
        int i = comHand.Count();
        comIndex = r1.Next(0, i);
        dispCom = comHand[comIndex];
        selectCard2 = comHand[comIndex];
    }
    ShowBan();
}

```

と修正します。最後に、ShowBan() を

```

private void ShowBan()
{
    Graphics g = pictureBox1.CreateGraphics();

    // Create solid brush.
    SolidBrush whiteBrush = new SolidBrush(Color.White);

    // Create rectangle.
    Rectangle rect = new Rectangle(0, 0, 1000, 600);

    // Fill rectangle to screen.
    g.FillRectangle(whiteBrush, rect);

    int pos = 30;
    for (int i = 0; i < comHand.Count(); i++)
    {
        g.DrawImage(comHand[i].image2, pos, 10, 70, 100);
        pos += 72;
    }
    pos = 30;
    for (int i = 0; i < userHand.Count(); i++)
    {
        g.DrawImage(userHand[i].image, pos, 480, 70, 100);
        pos += 72;
    }
    if (yama?.Count > 0)
    {

```

```

        g.DrawImage(yama[0].image, 30, 250, 70, 100);
    }
    if (getCom?.Count > 0)
    {
        for (int i = 0; i < getCom.Count(); i++)
        {
            g.DrawImage(getCom[i].image, 30 + 36 * i, 130, 70, 100);
        }
    }
    if (getUser?.Count > 0)
    {
        for (int i = 0; i < getUser.Count(); i++)
        {
            g.DrawImage(getUser[i].image, 30 + 36 * i, 370, 70, 100);
        }
    }
    if (gameOverP == false)
    {
        if (comPlayP == false)
        {
            //フォントオブジェクトの作成
            Font fnt = new Font("MS UI Gothic", 20);
            //文字列を位置(0,0)、青色で表示
            g.DrawString("カードを選んでください",
                fnt, Brushes.Blue, 200, 400);
            //リソースを解放する
            fnt.Dispose();
        }
    }
    else
    {
        //フォントオブジェクトの作成
        Font fnt = new Font("MS UI Gothic", 20);
        //文字列を位置(0,0)、青色で表示
        g.DrawString("ゲームオーバー", fnt, Brushes.Blue, 200, 400);
        string str = "";
        if (getCom.Count() > getUser.Count())
        {
            str = "コンピュータの勝ちです!";
        }
    }
}

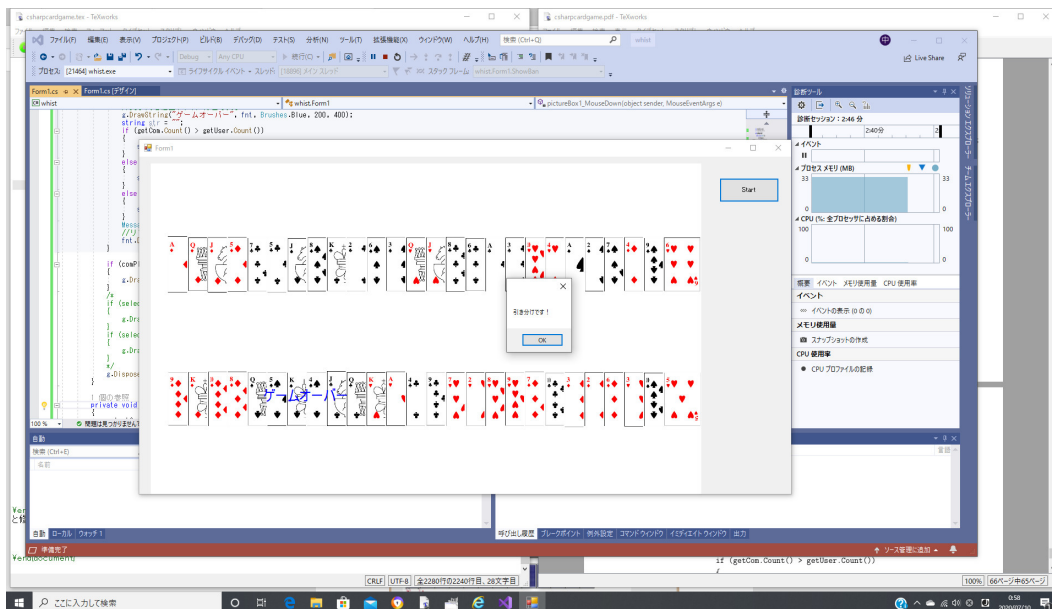
```

```

    }
    else if (getCom.Count() < getUser.Count())
    {
        str = "あなたの勝ちです!";
    }
    else
    {
        str = "引き分けです!";
    }
    MessageBox.Show(str);
    //リソースを解放する
    fnt.Dispose();
}
if (comPlayP)
{
    g.DrawImage(disCom.image, 400, 250, 70, 100);
}
g.Dispose();
}

```

と修正します。ビルドし、実行すると



のようになります。取ったカードを表示するスペースがないみたいです。

ShowBan() の

```
if (getCom?.Count > 0)
```

```

{
    for (int i = 0; i < getCom.Count(); i++)
    {
        g.DrawImage(getCom[i].image, 30 + 36 * i, 130, 70, 100);
    }
}
if (getUser?.Count > 0)
{
    for (int i = 0; i < getUser.Count(); i++)
    {
        g.DrawImage(getUser[i].image, 30 + 36 * i, 370, 70, 100);
    }
}

```

を

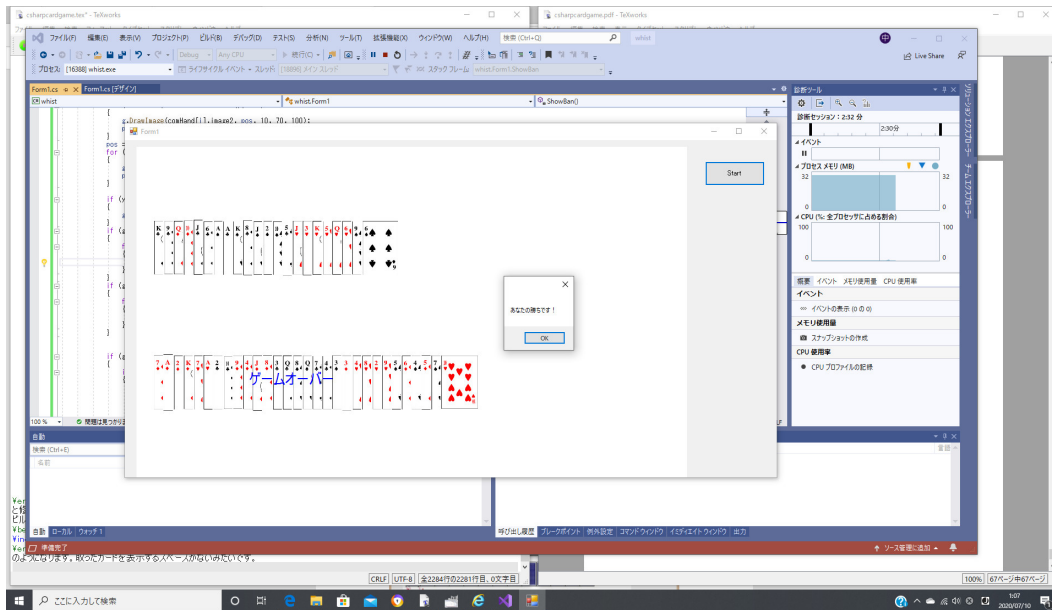
```

if (getCom?.Count > 0)
{
    for (int i = 0; i < getCom.Count(); i++)
    {
        g.DrawImage(getCom[i].image, 30 + 18 * i, 130, 70, 100);
    }
}
if (getUser?.Count > 0)
{
    for (int i = 0; i < getUser.Count(); i++)
    {
        g.DrawImage(getUser[i].image, 30 + 18 * i, 370, 70, 100);
    }
}

```

と修正します。

ビルドし、実行すると



のようになります。今度は表示に余裕があります。
最終的なプログラムの全体は

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Deployment.Application;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace whist
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            for (int s = 0; s < 4; s++)
            {
                for (int r = 0; r < 13; r++)
                {
```

```

        string name = "";
        if (s == 0)
        {
            name = "c";
        }
        else if (s == 1)
        {
            name = "d";
        }
        else if (s == 2)
        {
            name = "h";
        }
        else
        {
            name = "s";
        }
        name += r + 1;
        name += ".jpg";

        Image img = Image.FromFile(name);
        Image img2 = Image.FromFile("bk.jpg");
        card = new Card(s, r, img, img2);
        carddeck.Add(card);
    }
}
card_initialize();
gameInitialize();
}

public class Card
{
    public int suit, rank;
    public Image image, image2;
    public Card(int s, int r, Image img, Image img2)
    {
        this.suit = s;
        this.rank = r;
        this.image = img;
    }
}

```

```

        this.image2 = img2;
    }
}

public Card card = null;
public List<Card> carddeck = new List<Card>();
public List<Card> comHand = new List<Card>();
public List<Card> userHand = new List<Card>();
public List<Card> yama = new List<Card>();
public List<Card> getCom = new List<Card>();
public List<Card> getUser = new List<Card>();
public Card dispCom = null;
public Card trump = null;
public bool gameOverP = false;
public bool comPlayP = true;
public Card selectCard = null;
public Card selectCard2 = null;
int userIndex = -1;
int comIndex = -1;

private void pictureBox1_Paint(object sender, PaintEventArgs e)
{
}

private void card_initialize()
{
    Random r1 = new System.Random();
    int i = carddeck.Count();
    while (i > 0)
    {
        int s = r1.Next(0, i);
        Card temp = carddeck[i - 1];
        carddeck[i - 1] = carddeck[s];
        carddeck[s] = temp;
        i = i - 1;
    }
}

private void gameInitialize()
{
    comHand.Clear();
    userHand.Clear();
}

```

```

yama.Clear();

for (int i = 0; i < 13; i++)
{
    comHand.Add(carddeck[i]);
}
for (int i = 13; i < 26; i++)
{
    userHand.Add(carddeck[i]);
}
for (int i = 26; i < 52; i++)
{
    yama.Add(carddeck[i]);
}
getCom.Clear();
getUser.Clear();
dispCom = null;
gameOverP = false;
comPlayP = false;
trump = yama[0];
}

private void button1_Click(object sender, EventArgs e)
{
    gameInitialize();
    ShowBan();
}

private void ShowBan()
{
    Graphics g = pictureBox1.CreateGraphics();

    // Create solid brush.
    SolidBrush whiteBrush = new SolidBrush(Color.White);

    // Create rectangle.
    Rectangle rect = new Rectangle(0, 0, 1000, 600);

    // Fill rectangle to screen.

```



```

g.FillRectangle(whiteBrush, rect);

int pos = 30;

for (int i = 0; i < comHand.Count(); i++)
{
    g.DrawImage(comHand[i].image2, pos, 10, 70, 100);
    pos += 72;
}
pos = 30;
for (int i = 0; i < userHand.Count(); i++)
{
    g.DrawImage(userHand[i].image, pos, 480, 70, 100);
    pos += 72;
}

if (yama?.Count > 0)
{
    g.DrawImage(yama[0].image, 30, 250, 70, 100);
}
if (getCom?.Count > 0)
{
    for (int i = 0; i < getCom.Count(); i++)
    {
        g.DrawImage(getCom[i].image, 30 + 18 * i, 130, 70, 100);
    }
}
if (getUser?.Count > 0)
{
    for (int i = 0; i < getUser.Count(); i++)
    {
        g.DrawImage(getUser[i].image, 30 + 18 * i, 370, 70, 100);
    }
}

if (gameOverP == false)
{
    if (comPlayP == false)
    {

```

```

        //フォントオブジェクトの作成
        Font fnt = new Font("MS UI Gothic", 20);
        //文字列を位置 (0,0)、青色で表示
        g.DrawString("カードを選んでください",
            fnt, Brushes.Blue, 200, 400);
        //リソースを解放する
        fnt.Dispose();
    }
}
else
{
    //フォントオブジェクトの作成
    Font fnt = new Font("MS UI Gothic", 20);
    //文字列を位置 (0,0)、青色で表示
    g.DrawString("ゲームオーバー", fnt, Brushes.Blue, 200, 400);
    string str = "";
    if (getCom.Count() > getUser.Count())
    {
        str = "コンピュータの勝ちです!";
    }
    else if (getCom.Count() < getUser.Count())
    {
        str = "あなたの勝ちです!";
    }
    else
    {
        str = "引き分けです!";
    }
    MessageBox.Show(str);
    //リソースを解放する
    fnt.Dispose();
}
if (comPlayP)
{
    g.DrawImage(disCom.image, 400, 250, 70, 100);
}
g.Dispose();
}

```

```

private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    selectCard = null;

    int mX = e.X;
    int mY = e.Y;
    userIndex = -1;
    int pos = 29;
    for (int i = 0; i < userHand.Count(); i++)
    {
        if ((mX > pos && mX < pos + 72) && (mY > 480 && mY < 580))
        {
            userIndex = i;
            break;
        }
        pos += 72;
    }
    if (userIndex < 0)
        return;

    selectCard = userHand[userIndex];

    if (!comPlayP) // ユーザーの手番
    {
        int suit = userHand[userIndex].suit;
        int rank = userHand[userIndex].rank;

        comIndex = -1;
        // リードされたスーツの手札があれば、それを出す
        for (int i = 0; i < comHand.Count(); i++)
        {
            if (comHand[i].suit == suit)
            {
                comIndex = i;
                break;
            }
        }
        if (comIndex < 0) // リードされたスーツの手札が無かった
        {

```

```

// 切り札のスーツの手札があれば、それを出す
for (int i = 0; i < comHand.Count(); i++)
{
    if (comHand[i].suit == trump.suit)
    {
        comIndex = i;
        break;
    }
}
}
if (comIndex < 0) // 切り札のスーツの手札が無かった
{
    // 乱数で手を選ぶ
    Random r1 = new System.Random();
    int i = comHand.Count();
    comIndex = r1.Next(0, i);
}
selectCard2 = comHand[comIndex];
comHand.RemoveAt(comIndex);
userHand.RemoveAt(userIndex);
if (comWinP(selectCard2, selectCard))
{
    getCom.Add(selectCard2);
    getCom.Add(selectCard);
    if (yama?.Count > 0)
    {
        comHand.Add(yama[0]);
        userHand.Add(yama[1]);
        yama.RemoveAt(0);
        yama.RemoveAt(0);
    }
    comPlayP = true;
}
else
{
    getUser.Add(selectCard2);
    getUser.Add(selectCard);
    if (yama?.Count > 0)
    {

```

```

        userHand.Add(yama[0]);
        comHand.Add(yama[1]);
        yama.RemoveAt(0);
        yama.RemoveAt(0);
    }
    comPlayP = false;
}
selectCard = null;
selectCard2 = null;
dispCom = null;
} else
{
    // ユーザーの手がルールを守っているか?
    int suit = userHand[userIndex].suit;
    int rank = userHand[userIndex].rank;
    if (suit != dispCom.suit)
    {
        bool flag = false;
        for (int i=0; i<userHand.Count(); i++)
        {
            if (userHand[i].suit == dispCom.suit)
            {
                flag = true; break;
            }
        }
        if (flag)
        {
            string str = "リードは";
            switch (dispCom.suit)
            {
                case 0: str += "クラブ"; break;
                case 1: str += "ダイヤ"; break;
                case 2: str += "ハート"; break;
                case 3: str += "スペード"; break;
            }
            str += "です";
            MessageBox.Show(str);
            selectCard = null;
            return;
        }
    }
}

```

```

    }
}
if (comWinP(dispatchCom, selectCard))
{
    getCom.Add(selectCard2);
    getCom.Add(selectCard);
    if (yama?.Count > 0)
    {
        comHand.Add(yama[0]);
        userHand.Add(yama[1]);
        yama.RemoveAt(0);
        yama.RemoveAt(0);
    }
    comPlayP = true;
}
else
{
    getUser.Add(selectCard);
    getUser.Add(selectCard2);
    if (yama?.Count > 0)
    {
        userHand.Add(yama[0]);
        comHand.Add(yama[1]);
        yama.RemoveAt(0);
        yama.RemoveAt(0);
    }
    comPlayP = false;
}
comHand.RemoveAt(comIndex);
userHand.RemoveAt(userIndex);
selectCard = null;
selectCard2 = null;
dispatchCom = null;
}
if (comHand.Count() == 0)
{
    gameOverP = true;
} else if (comPlayP)
{

```

```

        // 乱数で手を選ぶ
        Random r1 = new System.Random();
        int i = comHand.Count();
        comIndex = r1.Next(0, i);
        dispCom = comHand[comIndex];
        selectCard2 = comHand[comIndex];
    }
    ShowBan();
}
private bool comWinP(Card ccard, Card ucard)
{
    if (ccard.suit == ucard.suit)
    {
        if (ccard.rank == 0)
        {
            return true;
        }
        else if (ucard.rank == 0)
        {
            return false;
        }
        else if (ccard.rank > ucard.rank)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    else if (ccard.suit == trump.suit)
    {
        return true;
    }
    else if (ucard.suit == trump.suit)
    {
        return false;
    }
    else if (comPlayP)

```

```
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}
```

です。強くすることや手札をソートして見やすくすることなどは各自で改良してください。

Visual studio の C++ では、簡単に Windows のプログラムは作れなくなりましたが、C# なら、C++ のプログラミング経験を生かして、比較容易に、C++ のプログラムをまねして作ることがこのように出来ます。