

Gioco di carte

Risorse

Scaricare le immagini delle carte da

<http://www.jfitz.com/cards/classic-cards.zip>

Rinominare le carte nel modo seguente:

x_y dove **x** è il valore della carta, numerata da 1 (asso) a 13 (king) e **y** il seme della carta (0 cuori, 1 quadri, 2 fiori, 3 picche)

Di seguito il codice del progetto:

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

namespace Poker
{
    public partial class Form1 : Form
    {
        //-----GLOBALI
        public const int StandardTop = 50;
        public const int HighTop = 70;

        public class Carta
        {
            public string valore;
            public string seme;
            public PictureBox immagine;
            public Carta(int valore, int seme)
                : this(""+valore, ""+seme)
            {
            }
            public Carta(string valore, string seme)
            {
                this.valore = valore;
                this.seme = seme;
                int v = Convert.ToInt16(valore);
                int s = Convert.ToInt16(seme);
            }
        }
    }
}
```

```

        string nomecarta = valore + "_" + seme;
        string percorso = @"..\cards\"; //percorso relativo rispetto all'eseguibile
        string estensione = ".png";
        nomecarta = percorso + nomecarta + estensione;
        immagine = new PictureBox();
        immagine.Load(nomecarta);
    }
    public Carta() //crea solo un dorso
    {
        this.valore = "0";
        this.seme = "";
        string nomecarta = "dorso";
        string percorso = @"..\cards\"; //percorso relativo rispetto all'eseguibile
        string estensione = ".png";
        nomecarta = percorso + nomecarta + estensione;
        immagine = new PictureBox();
        immagine.Load(nomecarta);
    }
    public string Valore()
    {
        if (valore == "1")
            return "A";
        if (valore == "11")
            return "J";
        if (valore == "12")
            return "Q";
        if (valore == "13")
            return "K";
        return valore;
    }
}

//elementi del progetto
static Random casuale = new Random();
static List<Carta> mazzo;
static List<Carta> mano ;
static List<PictureBox> segnaPosto; //collezione di PictureBoxes
static Carta dorso;
static int fase;

//-----FUNZIONI
public void Inizio()

```

```

{
    CreaMazzo();
    CreaDorso();
    fase = 0; //fase iniziale
}

public void CreaMazzo()
{
    //creazione mazzo di carte
    mazzo = new List<Carta>();
    for (int valore = 1; valore <= 13; valore++)
        for (int seme = 1; seme <= 4; seme++)
        {
            Carta tmp = new Carta(valore, seme);
            mazzo.Add(tmp);
        }
    } // for for
} //metodo CreaMazzo()

public void CreaDorso()
{
    dorso = new Carta();
}

public void CreaImmagini()
{
    //creazione PictureBox sul form1
    segnaPosto = new List<PictureBox>();
    for (int i = 0; i < 6; i++)
    {
        PictureBox tmp = new PictureBox(); // immagine d'appoggio
        tmp.SizeMode = PictureBoxSizeMode.AutoSize;
        tmp.Top = StandardTop;
        tmp.Left = 20 + (i * 100);
        tmp.Tag = "" + i; // dato per individuare la carta
        segnaPosto.Add(tmp); // aggiungo in collezione di programma
        Controls.Add(tmp); // aggiungo nel Form1 (sua collezione di componenti)
        if ( i > 0 ) //tranne il mazzo
            segnaPosto[i].Click += carta_Click; //associo un metodo alla immagine della carta
    } // for
    segnaPosto[0].Image = dorso.immagine.Image;
    segnaPosto[0].Click += dorso_Click;
}

```

```
//metodo CreaImmagini()

public void NascondiImmagini()
{
    for (int i = 1; i < 6; i++)
    {
        segnaPosto[i].Visible = false;
    }
}

//metodo

public void AbilitaImmagini()
{
    for (int i = 1; i < 6; i++)
        segnaPosto[i].Enabled = true;
}

//metodo

public void DisabilitaImmagini()
{
    for (int i = 1; i < 6; i++)
        segnaPosto[i].Enabled = false;
}

//metodo

public void Estrai5Carte()
{
    //estrae cinque carte a caso e le pone nella mano
    mano = new List<Carta>();
    for (int i = 1; i <= 5; i++)
    {
        try
        {
            int quale = casuale.Next(mazzo.Count);
            Carta miacarta = mazzo[quale];
            mano.Add(miacarta); // aggiungo la carta alla mano
            mazzo.RemoveAt(quale); // e la tolgo dal mazzo
        }
        catch
        {
            MessageBox.Show("Errore: carte finite!");
            return;
        }
    }
}

//metodo
```

```
public void SostituisciCarte()
{
    foreach (PictureBox p in segnaPosto)
        if (p.Top == HighTop)
            {
                int quale = casuale.Next(mazzo.Count);
                Carta miacarta = mazzo[quale];
                //sostituire carte nella mano
                int ndx = Convert.ToInt16(p.Tag)-1;
                mano[ndx] = miacarta;
                p.Image = mano[ndx].immagine.Image;
                p.Top = StandardTop;
            }
} //metodo

public void MostraCarte()
{
    int cont = 1;
    foreach (Carta c in mano)
        {
            segnaPosto[cont].Image = c.immagine.Image;
            segnaPosto[cont].Visible = true;
            cont++;
        }
} //metodo

private void VerificaVittoria()
{
    //verifica poker, tris e coppie
    string cosa, cosaltro;
    if (VerificaPoker(out cosa))
        {
            MessageBox.Show("WOW!!! Poker di " + cosa);
            Vincita(1000);
        } //
    else if (VerificaColore(out cosa))
        {
            MessageBox.Show("WOW!!! Colore di " + cosa);
            Vincita(500);
        }
    else if (VerificaScala(out cosa))
```

```

{
    MessageBox.Show("WOW!!! Scala al " + cosa);
    Vincita(300);
}
else if (VerificaTris(out cosa) && VerificaCoppia(out cosaltro))
{
    MessageBox.Show("WOW!!! Full di " + cosa + " e " + cosaltro);
    Vincita(250);
}
else if (VerificaTris(out cosa))
{
    MessageBox.Show("Buono!! Tris di " + cosa);
    Vincita(100);
}
else if (VerificaDoppiaCoppia(out cosa, out cosaltro))
{
    MessageBox.Show("Non male!! Doppia coppia di " + cosa + " e " + cosaltro);
    Vincita(10);
}
else if (VerificaCoppia(out cosa))
{
    MessageBox.Show("Coppia di " + cosa);
    Vincita(0);
}
else
{
    MessageBox.Show("Non hai vinto. Riprova...");
    Vincita(-100);
}
//finalizzazione
fase = 2;
button1.Visible = true;
DisabilitaImmagini();
} //metodo

private void Vincita(int q)
{
    if (q > 0)
        label1.Text = "Hai vinto " + q;
    else if (q == 0)
        label1.Text = "Non hai perso nulla ";
    else

```

```
        label1.Text = "Hai perso " + (-q);
        int capitale = Convert.ToInt16(labelCapitale.Text);
        capitale += q;
        labelCapitale.Text = "" + capitale;
    } //metodo

    private bool VerificaPoker(out string p)
    {
        for (int ndx = 0; ndx < 5; ndx++)
        {
            int cont = 1;
            string uffa = mano[ndx].Valore();
            for (int i = 0; i < 5; i++)
            {
                if ((ndx != i) && (uffa == mano[i].Valore()))
                    cont++;
            }
            if (cont == 4)
            {
                p = uffa;
                return true;
            }
        }
        p = "";
        return false;
    } //metodo

    private bool VerificaColore(out string semebase)
    {
        int cont = 1;
        semebase = mano[0].seme;
        for (int ndx = 1; ndx < 5; ndx++)
            if (semebase == mano[ndx].seme)
                cont++;
        if (cont == 5)
            return true;
        return false;
    } //metodo

    private bool VerificaTris(out string p)
    {
        for (int ndx = 0; ndx < 5; ndx++)
```

```
{
    int cont = 1;
    string uffa = mano[ndx].Valore();
    for (int i = 0; i < 5; i++)
    {
        if ((ndx != i) && (uffa == mano[i].Valore()))
            cont++;
    }
    if (cont == 3)
    {
        p = uffa;
        return true;
    }
}
p = "";
return false;
} //metodo

private bool VerificaCoppia(out string p)
{
    for (int ndx = 0; ndx < 5; ndx++)
    {
        int cont = 1;
        string uffa = mano[ndx].Valore();
        for (int i = 0; i < 5; i++)
        {
            if ((ndx != i) && (uffa == mano[i].Valore()))
                cont++;
        }
        if (cont == 2)
        {
            p = uffa;
            return true;
        }
    }
    p = "";
    return false;
} //metodo

private bool VerificaDoppiaCoppia(out string prima, out string seconda)
{
    if (VerificaCoppia(out prima))
```



```

{
    for (int ndx = 0; ndx < 5; ndx++)
    {
        int cont = 1;
        seconda = mano[ndx].Valore();
        for (int i = 0; i < 5; i++)
        {
            if ((prima != seconda) && (ndx != i) && (seconda == mano[i].Valore()))
                cont++;
        }
        if (cont == 2)
        {
            return true;
        }
    }
}
seconda = "";
return false;
} //metodo

private bool VerificaScala(out string p)
{
    //prendo tutti i valori delle carte
    int[] valori = new int[5];
    for (int i = 0; i < 5; i++)
        valori[i] = Convert.ToInt16(mano[i].valore);
    //li ordino
    for (int i = 0; i < 5; i++)
        for (int c = 0; c < 5; c++)
            if (valori[i] > valori[c])
            {
                int tmp = valori[i];
                valori[i] = valori[c];
                valori[c] = tmp;
            }
    //verifico se è una scala ordinata crescente
    bool ris = true;
    for (int i = 0; i < 4; i++)
        ris = ris && (valori[i]+1 == valori[i+1]);
    //rendo i risultati
    p = "" + valori[4];
    return ris;
}

```

```

}

//-----GESTORI EVENTO
public Form1()
{
    InitializeComponent();
} //gestore evento

private void Form1_Load(object sender, EventArgs e)
{
    BackColor = Color.Green;
    panel1.BackColor = Color.GreenYellow;
    panel2.BackColor = Color.ForestGreen;
    StartPosition = FormStartPosition.CenterScreen; //posiziona la finestra al centro schermo
    Inizio();
    CreaImmagini();
    button1.Visible = false;
} //gestore evento

private void dorso_Click(object sender, EventArgs e)
{
    switch (fase)
    {
        case 0: //fase iniziale
            //estrae cinque carte a caso e le visualizza
            Estrai5Carte();
            MostraCarte();
            fase++;
            label1.Text = "Selezionare le carte da sostituire, prego";
            AbilitaImmagini();
            return;
        case 1: //fase finale
            //sostituisce le carte selezionate e le visualizza
            SostituisciCarte();
            MostraCarte();
            segnaPosto[0].Enabled = false;
            label1.Text = "Carte sostituite";
            VerificaVittoria();
            return;
        default: //attesa ripetizione
            MessageBox.Show("errore programma");
    }
}

```

```
        return;
    } //switch
} //gestore evento

private void carta_Click(object sender, EventArgs e)
{
    //rileva il clic sulla carta estratta
    PictureBox p = (sender as PictureBox);
    //determina la carta del mazzo cliccata
    int codice = Convert.ToInt16(p.Tag)-1;

    if (p.Top == StandardTop)
        p.Top = HighTop;
    else
        p.Top = StandardTop;
    label1.Text = "Dopo la selezione, clic sul mazzo, prego";

} //gestore evento

private void button1_Click(object sender, EventArgs e)
{
    Inizio();
    NascondiImmagini();
    button1.Visible = false;
    segnaPosto[0].Enabled = true;
    label1.Text = "Per distribuire le carte, fare un clic sul mazzo, prego";

} //gestore evento

private void button2_Click(object sender, EventArgs e)
{
    Close();
} //gestore evento
}
}
```