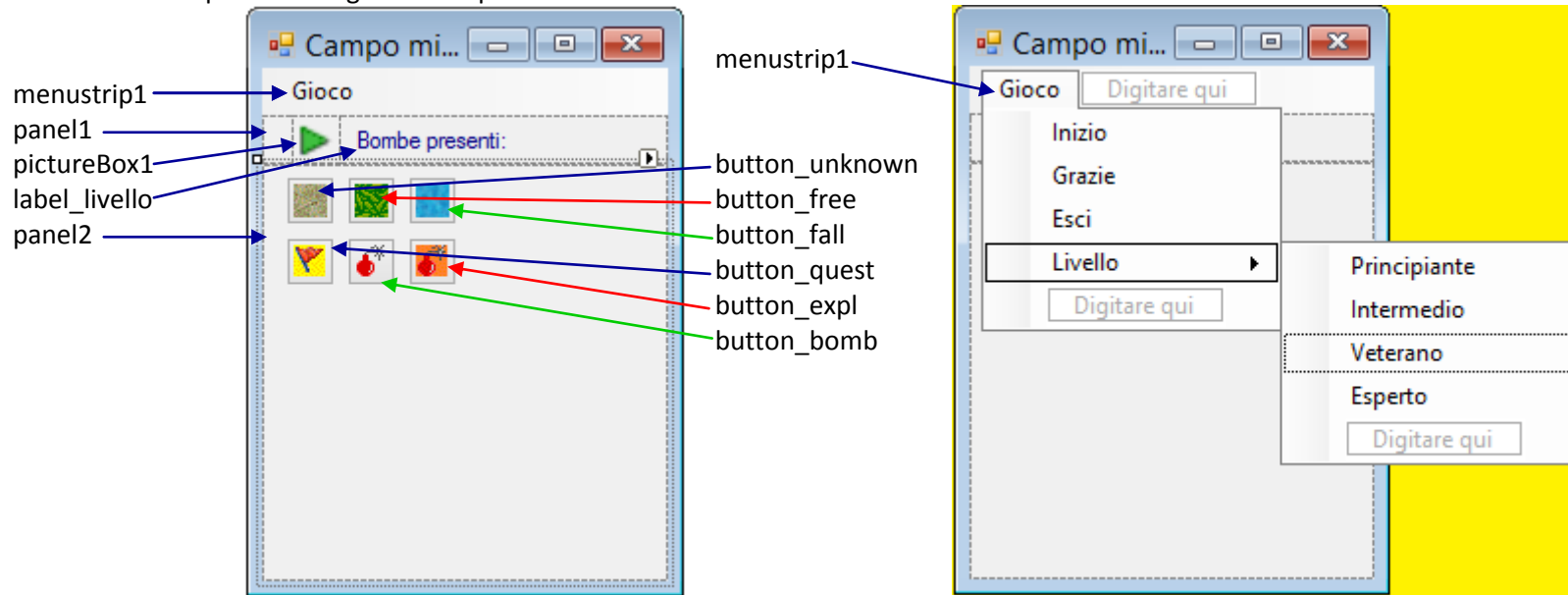


Campo Minato

Nel Form1 sono presenti i seguenti componenti:



Il codice è il seguente:

```

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;
/*****
 * PROGETTO: CAMPO MINATO *****/
 * SCUOLA: ITI G.M. ANGIOY SASSARI *****/
 * CLASSE: IV A INFORMATICA - A.S. 2013/14 *****/
 * DOCENTI: Prof. M. Cherchi, Prof. A. Zoccheddu *****/
 * STUDENTI: Gnocchi Domenica, Piano Guido, Carta Bianca *****/
 * *****/
namespace CampoMinato
{

```

```

public partial class Form1 : Form
{
    static Random estrai = new Random();
    static int max_ri = 8;
    static int max_co = 8;
    static int num_mine = 16;
    static Campo campo;
    static Button bt_unknown, bt_free, bt_fall, bt_quest, bt_bomb, bt_expl;

    public class MyCell
        : Button
    {
        const int dim = 25;
        public bool bomb;
        public int stato; //0==scoprire, 1==libero, 2==bloccato,
        public bool controllato;

        public MyCell(int top, int left)
            : base()
        {
            this.bomb = false;
            this.stato = 0; //libero
            this.controllato = false;
            this.Top = dim*top + 5;
            this.Left = dim * left + 5;
            this.Height = dim;
            this.Width = dim;
            this.Image = bt_unknown.Image;
            this.ForeColor = Color.White;
            this.Font = new Font("Consolas", 12, FontStyle.Bold);

            //il costruttore imposta il delegato gestore di evento clic sinistro
            this.Click += delegate(object sender, EventArgs e)
            {
                if (this.stato == 0) //da scoprire
                {
                    if (this.MyTag() == "1") //c'è una bomba
                    {
                        //sconfitta
                        campo.Sconfitta(top, left);
                    }
                }
                else

```

```

        {
            //gioca
            this.Image = bt_free.Image;
            this.stato = 1; //libero
            campo.mosse--;
            campo.VerificaVittoria();
            campo.Scopri(top, left);
        }
    };
    //il costruttore imposta il delegato gestore di evento clic destro
    this.MouseDown += delegate(object sender, MouseEventArgs e)
    {
        if (e.Button == System.Windows.Forms.MouseButtons.Right)
        {
            switch (this.stato)
            {
                case 0: //era libero
                    this.stato = 2; //bloccato
                    this.Image = bt_quest.Image;
                    return;
                case 2:
                    this.stato = 0; //libero
                    this.Image = bt_unknown.Image;
                    return;
            }
        }
    };
} //costruttore

public string MyTag()
{
    if (this.bomb)
        return "1";
    else
        return "0";
}
} //class MyCell *****

```

```

public class Campo
{
    public MyCell[,] griglia;
    public int mosse;

    public Campo()
    {
        griglia = new MyCell[max_ri, max_co];
        for (int ri = 0; ri < max_ri; ri++)
            for (int co = 0; co < max_co; co++)
                {
                    griglia[ri, co] = new MyCell(ri, co);
                }
        this.Minare();
    } //constructor

    private void Minare()
    {
        int minecount = num_mine; //numero mine da posizionare
        while (minecount > 0)
        {
            int ri = estrai.Next(0, max_ri);
            int co = estrai.Next(0, max_co);
            if (griglia[ri, co].bomb == false)
            {
                griglia[ri, co].bomb = true; //mette bomba
                minecount--;
            }
        }
        mosse = (max_co*max_ri) - num_mine; //mosse disponibili
    } //method Minare()
}

```

```

public void Sconfitta(int r, int c)
{
    //Close();
    for (int ri = 0; ri < max_ri; ri++)
        for (int co = 0; co < max_co; co++)
        {
            griglia[ri, co].stato = 2;
            if (griglia[ri, co].MyTag() == "1")
            {
                griglia[ri, co].BackColor = Color.Black;
                griglia[ri, co].Image = bt_bomb.Image;
            }
            else
            {
                griglia[ri, co].BackColor = Color.Green;
                griglia[ri, co].Image = bt_free.Image;
            }
        }
    griglia[r, c].BackColor = Color.Maroon;
    //griglia[r, c].Image = bt_expl.Image;
    MessageBox.Show("Hai perso!");
}

public void VerificaVittoria()
{
    if (mosse == 0)
        this.Vittoria();
}

public void Vittoria()
{
    MessageBox.Show("Hai vinto!");
}

```

```

public void Scopri(int ri, int co)
{
    griglia[ri, co].Image = bt_free.Image;
    griglia[ri, co].stato = 1; //libero
    griglia[ri, co].Text = "" + this.Calcola(ri, co);
    if (griglia[ri, co].Text == "0")
    {
        for (int r = ri - 1; r <= ri + 1; r++)
            for (int c = co - 1; c <= co + 1; c++)
                if ((r >= 0) && (r < max_ri)
                    && (c >= 0) && (c < max_co)
                    && ((r != ri) || (c != co)))
                    if ( (griglia[r, c].stato == 0)//libero
                        && (griglia[r, c].controllato == false))
                        {
                            Scopri(r, c); //chiamata ricorsiva !!!
                        }
    }
}

public int Calcola(int ri, int co)
{
    int cont = 0;
    for (int r = ri - 1; r <= ri + 1; r++)
        for (int c = co - 1; c <= co + 1; c++)
            if ((r >= 0) && (r < max_ri)
                && (c >= 0) && (c < max_co)
                && ((r != ri) || (c != co)))
                if (griglia[r, c].MyTag() == "1")
                    cont++;
    return cont;
}

```

```

//*****metodi
public void Inizio()
{
    Width = max_co * 25 + 17;
    Height = max_ri * 25 + 87;
    panel1.BackColor = Color.PaleGreen;
    panel2.BackColor = Color.Green;
    StartPosition = FormStartPosition.CenterScreen;
    panel2.Controls.Clear();
    labelLivello.Text = "Bombe nascoste: " + num_mine;
}

public void Crea_Griglia()
{
    campo = new Campo();
    panel2.Controls.Clear();
    for (int ri = 0; ri < max_ri; ri++)
        for (int co = 0; co < max_co; co++)
            panel2.Controls.Add(campo.griglia[ri,co]);
}

public void Livello(int liv)
{
    max_ri = liv;
    max_co = liv;
    switch (liv)
    {
        case 8:
            num_mine = liv * 2;
            break;
        case 12:
            num_mine = liv * 2;
            break;
        case 16:
            num_mine = liv * 3;
            break;
        case 20:
            num_mine = liv * 4;
            break;
    }
    Inizio();
} //Livello()

```

```
//*****gestori eventi
public Form1()
{
    InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
    //impostazioni base finestra
    FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedToolWindow;
    //creazione campo di gioco
    Inizio();
    // static Button bt_unknown, bt_free, bt_fall, bt_quest, bt_bomb;
    bt_unknown = button_unknown;
    bt_free    = button_free;
    bt_fall    = button_fall;
    bt_quest   = button_quest;
    bt_bomb    = button_bomb;
    bt_expl    = button_expl;
}
```



```

private void inizioToolStripMenuItem_Click(object sender, EventArgs e)
{
    Crea_Griglia();
}

private void grazieToolStripMenuItem_Click(object sender, EventArgs e)
{
    Grazie ringraziamenti = new Grazie();
    ringraziamenti.ShowDialog();
}

private void esciToolStripMenuItem_Click(object sender, EventArgs e)
{
    Close();
}

private void pictureBox1_Click(object sender, EventArgs e)
{
    Crea_Griglia();
}

private void intermedioToolStripMenuItem_Click(object sender, EventArgs e)
{
    Livello(12);
}

private void veteranoToolStripMenuItem_Click(object sender, EventArgs e)
{
    Livello(16);
}

private void espertoToolStripMenuItem_Click(object sender, EventArgs e)
{
    Livello(20);
}

} //Form1
} //namespace

```

Al progetto è stato aggiunto un secondo form informativo per i ringraziamenti:



In un progetto si dovrebbe sempre:

- 1) **Citare le fonti** da cui si è tratto parte del codice software, anche se parziale e che deve essere senza copyright restrittivi
- 2) **Citare le fonti** delle risorse come immagini, suoni, video, pagine web, ecc
- 3) **Citare la scuola e i docenti** che hanno assistito gli studenti, anche se solo parzialmente o moralmente
- 4) Indicare i nomi degli **autori e i link** per i contatti