

## Лабораторна робота 2

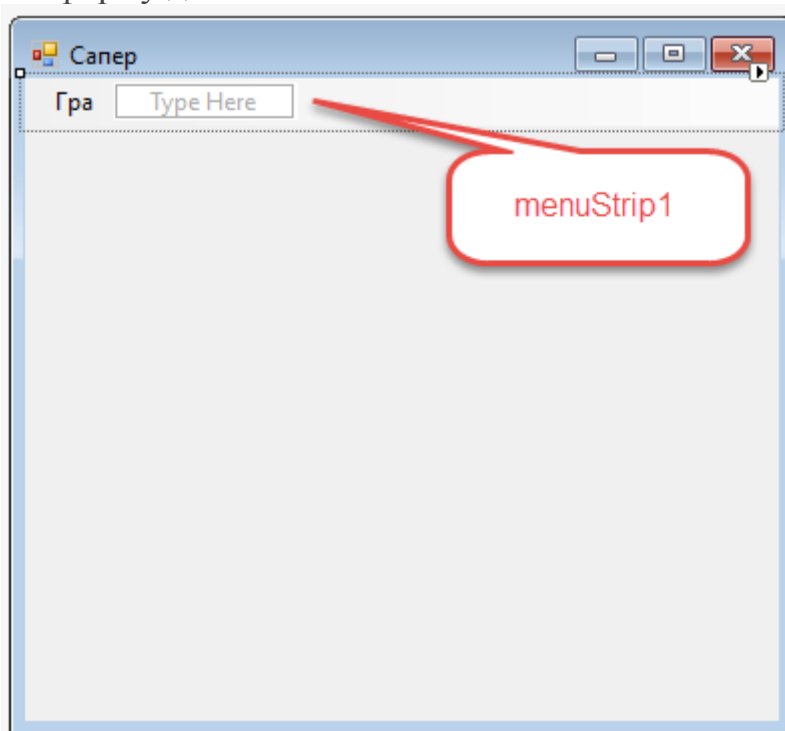
### Гра «Сапер»

Метою даної практичної роботи буде вивчення загального методу роботи з візуальними компонентами, а саме: програмне створення, програмний обробіток, та використання універсальних обробників подій. В даному випадку «програмне» мається на увазі – створення візуальної компоненти в кодї, а не переміщення мишею на форму. В лабораторній буде наведено обробка лише кнопок та текстових міток, але загальні методи одні і ті ж, і підходять до всіх візуальних компонент.

Спершу нам потрібно створити форму яка буде містити головне меню і відображати наші динамічно створені кнопки для гри.

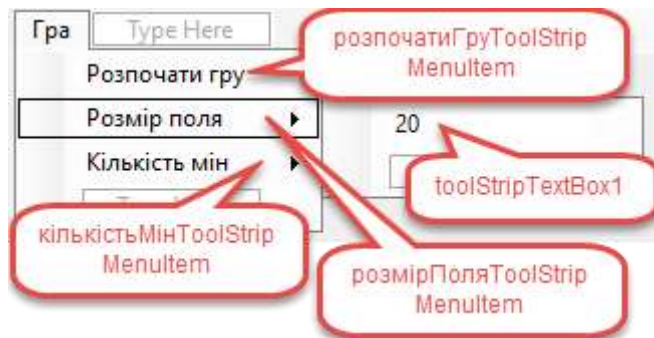
Якщо Ви ніколи не грали в сапера, то почитайте основні правила на вікіпедії. В нас буде дещо легша версія гри, перше, чого в нас не буде – це обробка натиску двох кнопок миші одночасно (це коли відкриваються послідовно прилеглі клітинки до порожньої). Також в нашій грі можна буде лише програти, процедури та функції обробки виграшу ми упустили, так як нам основне це процес гри, обробка натиску кнопок, обробка поля. Сама процедура виграшу – нецікава в плані візуального програмування. Ще в нашій грі не буде «прапорця», який помічає міну, і взагалі подія натиску правої кнопки миші буде відсутня.

Розпишемо форму детальніше:

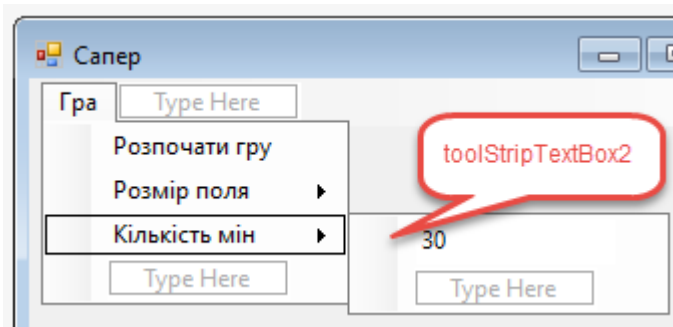


На форму потрібно буде додати лише один компонент – головне меню.

В свою чергу, головне меню буде містити такі елементи:

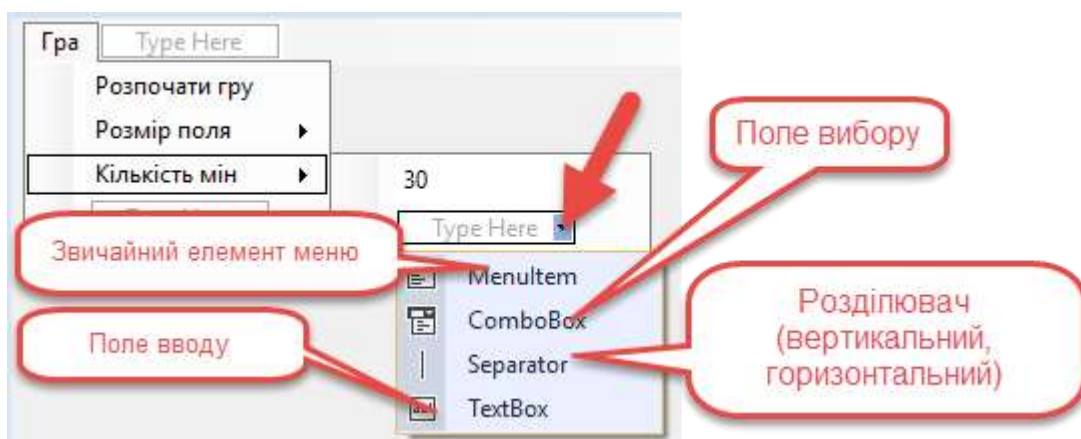


І елемент «Кількість мін»:



Назви компонентів подані для того, щоб Ви знали, який із наведених нижче подій в коді, відноситься до якої компоненти. І, у Вас може виникнути ситуація, коли на формі назви компонент будуть відрізнятися від наведених в прикладі. Наприклад компоненти, які були створені в іншій послідовності, будуть мати номери в назвах компоненти інші, бо вони нумеруються по порядку, а порядок створення вже інший.

По замовчуванню, при доданні створюється стандартний елемент меню. Створити ж, текстовий елемент, роздільник чи список вибору можна так:



Натискаємо правою кнопкою миші там, де на малюнку показує стрілка, та обираємо потрібний вид елемента меню.

Тепер потрібно описати події і методи нашої форми, нижче приведено код з коментарями. Блок підключення простору імен, ми упустили.

```

namespace WindowsFormsApplication1 //простір імен нашого додатку
{
    public partial class Form1 : Form
    {
        int[,] f = new int [51,51]; //масив значень поля (зберігає числа під кнопками)
255 - міна
        Button[,] b = new Button [51,51]; //масив кнопок (кнопки, які відображаються над
числами)
        Label[,] l = new Label[51, 51]; //масив міток під кнопками (візуальні мітки, які
відображають числа, які знаходяться під кнопками))
        int fsize = 20; //розмір поля по-замовчуванню
        int status = 0; //(0 - гра не розпочата, 1 - гра триває, 2 - програна)

        public void f_to_zero() { // прописує 0-лі в масив f
            int x, y; //змінні ля почергового перебору елементів
            for (x = 0; x <= 50; x++) //в нас максимальна ширина поля 50
                for (y = 0; y <= 50;y++) // в нас максимальна висота поля 50
                {
                    f[x,y]=0; //присвоюємо 0 елементу
                }
        }

        private void but_clck(object sender, EventArgs e) //процедура обробки натискання
кнопки мінного поля
        {
            Button o = ((Button)sender); //зводимо викликаючий об'єкт до кнопки
            Label ol = ((Label)o.Tag); //перетворюємо тег кнопки до мітки
            o.Visible = false; //приховуємо кнопку
            ol.Visible = true; //показуємо мітку з числом
            if (ol.Text == "B") //перевіряємо, чи ми не знаходимося на міні
            { //якщо все-таки знаходимося
                MessageBox.Show("Loooser!"); //виводимо повідомлення по те, що користувач
програв
                b_hide(); //приховуємо всі кнопки
                l_show(fsize); //показуємо всі мітки
                ol.Font = new Font(ol.Font, FontStyle.Bold); //виділяємо ту клітинку, на
якій "підірвався" гравець
                status = 2; //мінємо статус на гра програна
            }
        }

        public void b_l_create() { //створює мітки та кнопки
            int x, y, h; //допоміжні змінні
            //x, y - для обходу двовимірного масиву, h - для вирахування позиції
слідуючої кнопки
            h=menuStrip1.Height; //дізнаємося висоту рядка меню
            for (x = 0; x <= 50; x++) //перебираємо наші масиви з ліва направо
                for (y = 0; y <= 50;y++ ) //зверху в низ, через змінні x, y
                {
                    b[x,y]=new Button(); //створюємо наступну кнопку
                    l[x, y] = new Label(); //створюємо наступну мітку
                    l[x, y].Parent = this; //встановлюємо батька мітки нашу форму
                    b[x, y].Parent = this; //встановлюємо батька кнопки нашу форму
                    b[x, y].Width = 20; //ширина кнопки
                    b[x, y].Height = 20; //висота кнопки
                    l[x, y].Width = 20; //ширина мітки
                    l[x, y].Height = 20; //висота мітки
                    b[x, y].Top = y * 20 + h; //вираховуємо позицію кнопки від верхнього
краю форми з поправкою на ширину меню
                    l[x, y].Top = y * 20 + h; //вираховуємо позицію мітки від верхнього
краю форми з поправкою на ширину меню
                    b[x, y].Left = x * 20; //вираховуємо позицію кнопки від лівого краю
форми

```

```

        l[x, y].Left = x * 20;//вираховуємо позицію мітки від лівого краю
форми
        b[x, y].Visible = false;//приховуємо кнопку
        l[x, y].Visible = false;//приховуємо мітку
        b[x, y].Tag = l[x, y];//закріплюємо за кнопкою мітку, яка
міститься під нею
        b[x, y].Click += new System.EventHandler(this.but_clck);//реєструємо
універсальну функцію обробітку події
        //натиску для кожної кнопки
    }
}

public void b_show(int s) { //показує s кнопок поля (s - ширина поля)
    int x, y;//змінні для обходу двовимірного масиву
    for (x=0;x<s;x++)//проходимо масив зліва направо, зверху вниз
        for (y = 0; y <s; y++) { //через змінні x, y
            b[x, y].Visible=true;//показуємо кнопку
        }
    //встановлюємо розміри вікна, щоб усі міни помістилися
    this.Height = menuStrip1.Height + s * 20;//змінюємо висоту вікна з
врахуванням висоти меню
    this.Width = s * 20+s;//змінюємо ширину вікна
}

public void b_hide() { // ховає всі кнопки
    int x, y;//змінні для обходу масиву
    for (x=0;x<fsize;x++)//перебираємо масив зліва направо, зверху вниз
        for (y = 0; y <= fsize; y++) { //через x, y
            b[x, y].Visible = false;//ховаємо кнопку
        }
}

public void l_show(int s) //показує текстові мітки поля (s - розмір поля)
{
    int x, y;//змінні для обходу масиву
    for (x=0;x<s;x++)//перебираємо масив зліва направо, зверху вниз
        for (y = 0; y < s; y++)//через x, y
        {
            l[x, y].Visible = true;//показуємо мітку
        }
}

public void l_hide() //ховає всі текстові мітки
{
    int x, y;//змінні для обходу масиву
    for (x=0;x<fsize;x++)//перебираємо масив зліва направо, зверху вниз
        for (y = 0; y <=fsize; y++)//через x, y
        {
            l[x, y].Visible = false;//приховуємо мітку
            l[x, y].Font = new Font(l[x,y].Font, FontStyle.Regular);//змінюємо
параметри шрифту на звичайний
            //так, як в нас, наприклад клітинка де гравець програє відображається
жирним
        }
}

public void generate_bombs(int bm,int fs) { //розкидає міни по полю та переносить
надписи на поле
    //де:
    //int bm - кількість мін на полі,
    //int fs - розмір поля
    int i,x,y;//допоміжні змінні
    //i - для розміщення мінок на полі в потрібній кількості,
    //x,y - для обходу масиву поля
    Random rnd = new Random();//ініціалізуємо генератор випадкових чисел

```

```

for (i = 0; i < bm; i++)//генеруємо bm ітерацій (bm - кількість мінок)
{
    f[rnd.Next(1, fs + 1), rnd.Next(1, fs + 1)] = 255;//присвоюємо випадковим
координатам масиву мінок
    // число 255 - яке в нас позначає міну
}
//тепер нам наш числовий масив потрібно обробити, а саме:
//вирахувати числа суміжних з мінами клітинок
for (x=1;x<=fs;x++)//перебираємо масив зліва направо, зверху вниз
for (y = 1; y <= fs; y++)//через x, y
    if (f[x, y] == 255)//якщо клітинка містить мінку
    {
        //додаємо +1 до всіх 8-ми суміжних клітинок
        f[x - 1, y - 1]++;//
        f[x - 1, y]++;    //
        f[x - 1, y + 1]++;// [+1][+1][+1]
        f[x, y - 1]++;    // [+1][ B][+1]
        f[x, y + 1]++;    // [+1][+1][+1]
        f[x + 1, y - 1]++;//
        f[x + 1, y]++;    //
        f[x + 1, y + 1]++;//
    }
//тепер потрібно всі числа і міни перенести в масив міток, які знаходяться
під кнопками
for (x=0;x<fs;x++)//перебираємо масив зліва направо, зверху вниз
for (y = 0; y < fs; y++)//через x, y
{
    if (f[x+1,y+1]<255)//перевіряємо чи не міститься в клітинці міна
    {l[x, y].Text = f[x + 1, y + 1].ToString();} //якщо не міститься
зводимо число до рядка і присвоюємо мітці
    else {l[x,y].Text="B";}//якщо міна, то присвоюємо мітці «B»
}
}

/*
* Завантажуватись додаток буде відносно довго,
* так як, на створення 2500 (максимальний розмір поля 50x50)
* кнопок та міток і прив'язка їх між собою, займає трохи часу
*/

public Form1()//конструктор форми
{
    InitializeComponent();
    f_to_zero();//обнуляємо числовий масив поля
    b_l_create();//створюємо мітки та кнопки поля
}

private void розпочатиГруToolStripMenuItem_Click(object sender, EventArgs e)
//натискання елемента меню початку гри
{
    int bmv = Int32.Parse(toolStripTextBox2.Text);//визначаємо з меню кількість
мін
    int fsv = Int32.Parse(toolStripTextBox1.Text);//визначаємо з меню розмір поля
    fsize = fsv;//зберігаємо розмір поля в глобальних змінних
    if (this.status == 0) {//статус 0 - гра ще не починалась
        generate_bombs(bmv,fsv);//генеруємо мінки, ініціалізуємо поле
        b_show(fsv);//показуємо кнопки fsv - розмір поля
        status = 1;//змінюємо статус на 1 - гра розпочалась
    } else // якщо статус не 0 додатково перевіряємо:
        //статус 1 - гра триває
    if (status == 1) { MessageBox.Show("You need to loose first!"); }//виводимо
повідомлення, що потрібно спочатку програти
    if (status == 2) //статус 2 - гравець програв
    {
        f_to_zero();//обнуляємо (очищуємо) числове представлення поля
    }
}

```

