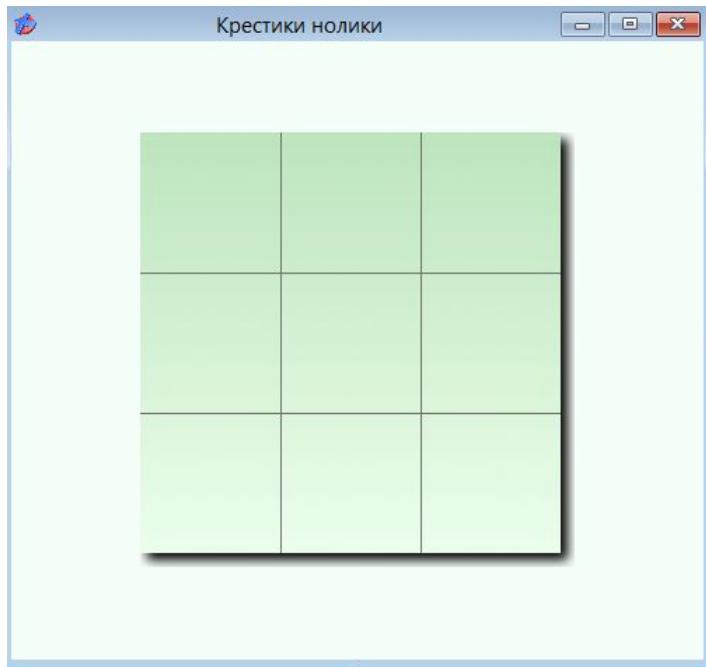


---

## СОЗДАНИЕ ИГРЫ КРЕСТИКИ НОЛИКИ

---

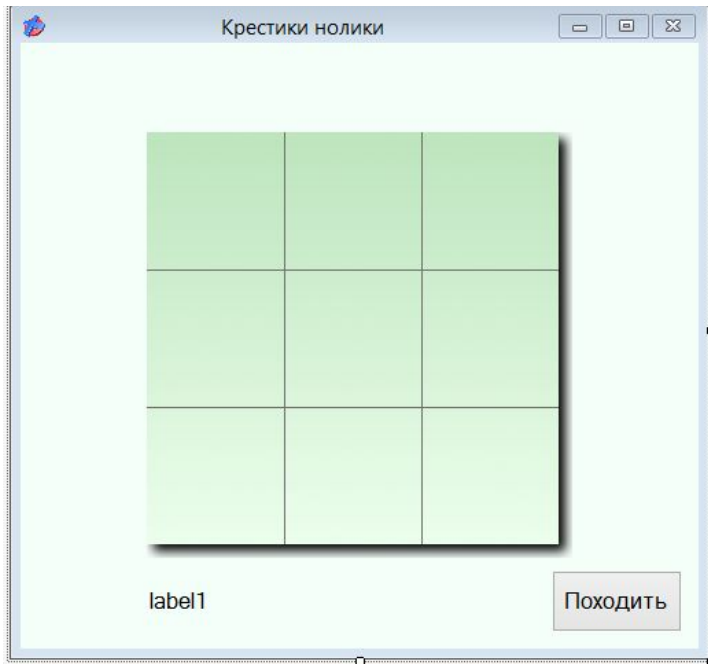
## ШАГ 1. НАСТРОЙКА ПАРАМЕТРОВ ФОРМЫ



1. Для настройки формы установите ее размер **510 точек** по горизонтали и **480 точек** по вертикале. Максимальный и минимальный размер укажите равный этим же значениям.
2. Назовите форму словом «**Крестики нолики**».
3. Для фона формы используйте файл из папки **Images** под имени **фон.png**, и разместите его по центру формы.
4. Для пиктограммы в строке заглавия используйте используйте файл из папки **Images** под имени **menu4.ico**.
5. Цвет фона формы необходимо установить **MintCream**.

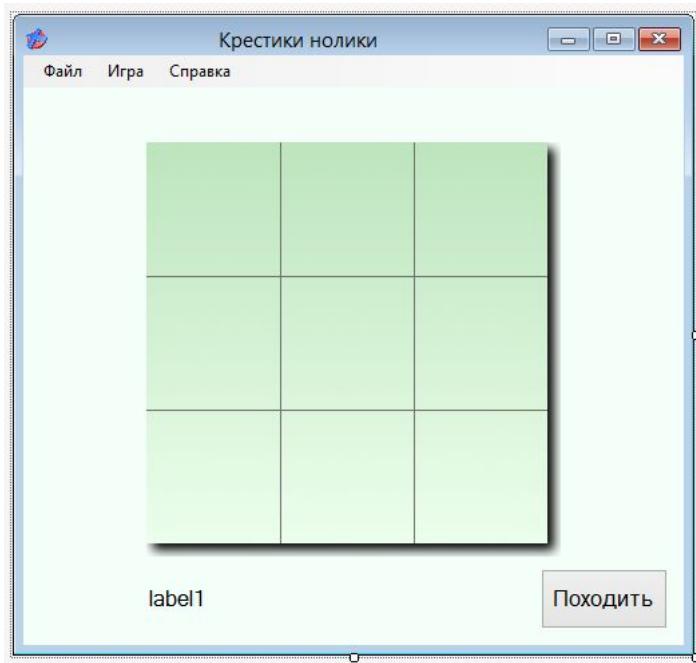


## ШАГ 2. ДОБАВЛЕНИЕ КНОПКИ И НАДПИСИ НА ФОРМУ



1. Для размещенных элементов изменяем размер шрифта на **12** и фон устанавливаем прозрачным.

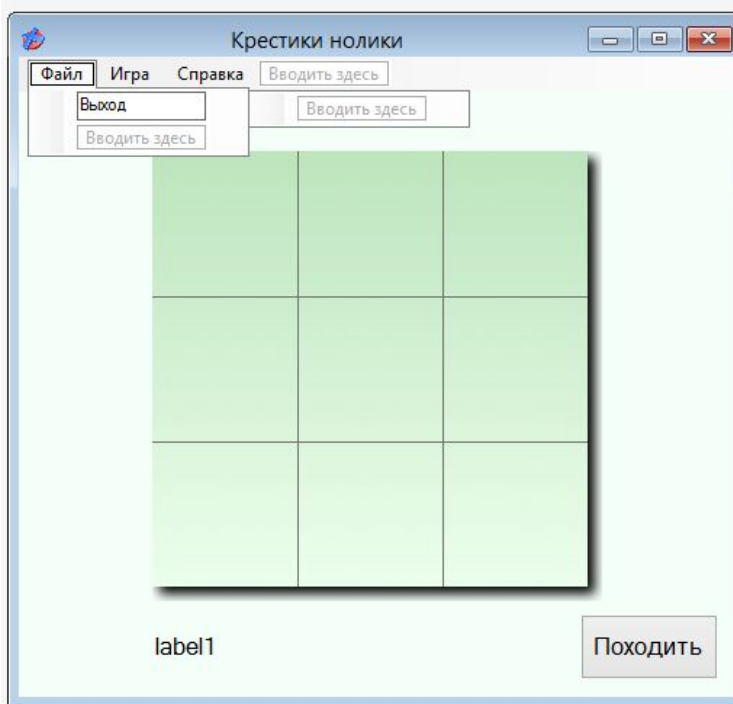
## ШАГ 3. ДОБАВЛЕНИЕ СТРОКИ МЕНЮ НА ФОРМУ



1. Создаем строку меню с пунктами в соответствии с указанными на изображении.

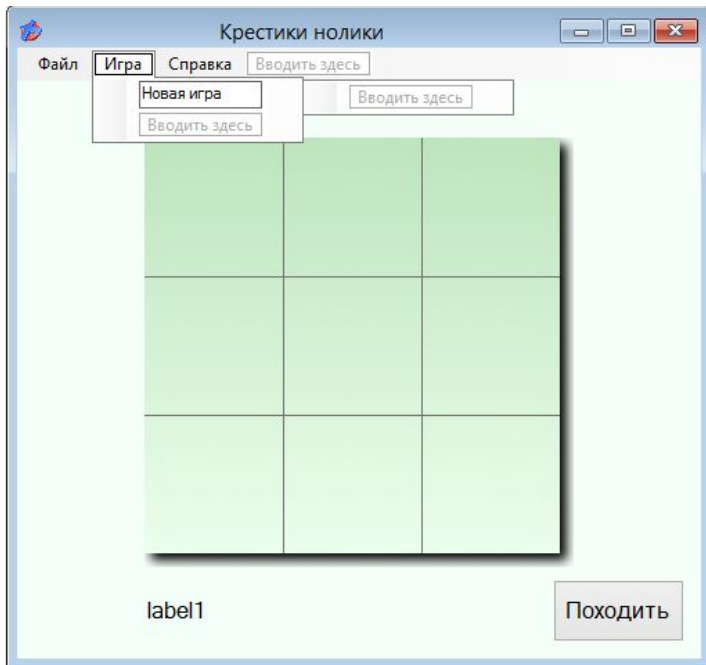


## ШАГ 3. ДОБАВЛЕНИЕ СТРОКИ МЕНЮ НА ФОРМУ



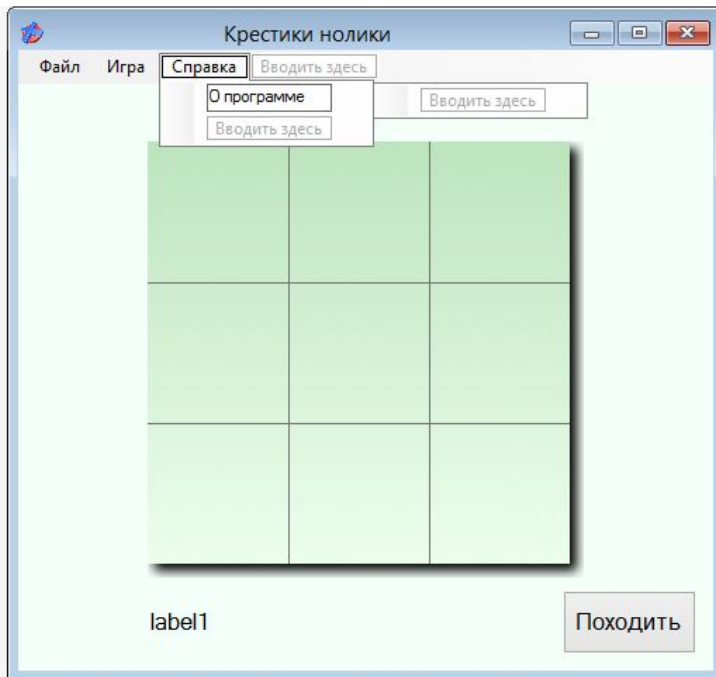
1. В пункте меню **Файл** создаем команду **Выход**.
2. Для команды **Выход** прописуем программный код аналогичный как и в предыдущем приложении.

## ШАГ 3. ДОБАВЛЕНИЕ СТРОКИ МЕНЮ НА ФОРМУ



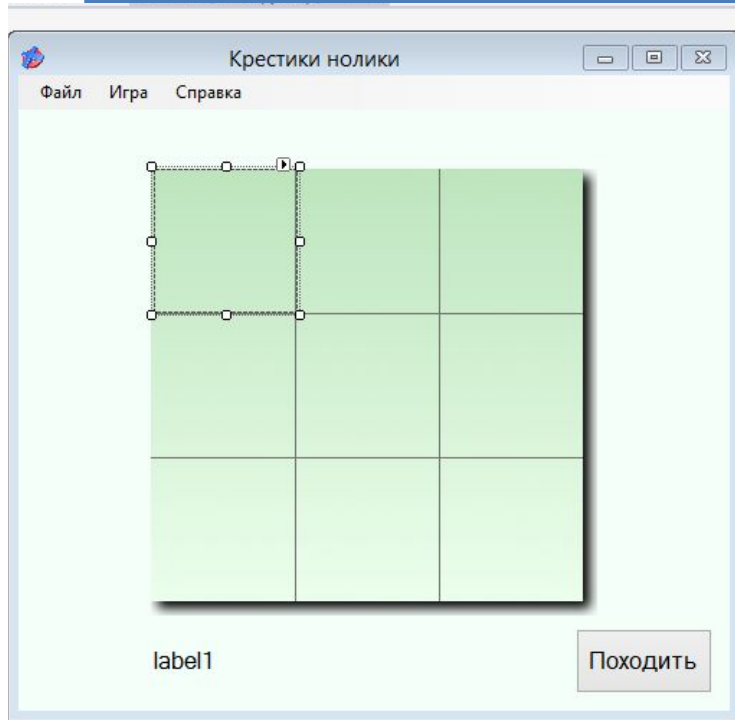
1. В пункте меню **Игра** создаем команду **Новая игра**.
2. Для команды **Новая игра** пропишем программный код в дальнейшем через несколько шагов.

## ШАГ 3. ДОБАВЛЕНИЕ СТРОКИ МЕНЮ НА ФОРМУ



- 1. В пункте меню **Справка** создаем команду **О программе**.
- 2. Для команды **О программе** создаем новую форму и прописуем программный код аналогичный как и в предыдущем приложении.

## ШАГ 4. ДОБАВЛЕНИЕ ОБЪЕКТОВ PICTUREBOX НА ФОРМУ

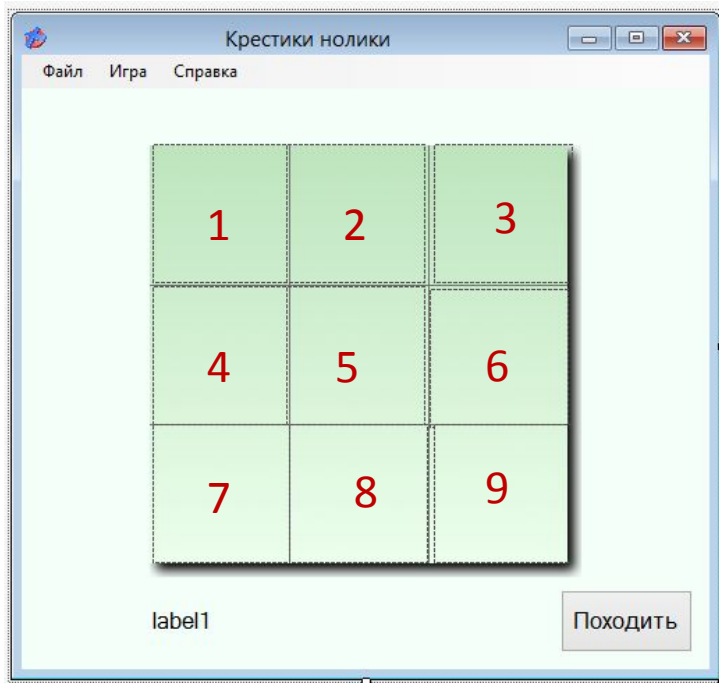


1. Перетянув на форму объект PictureBox изменяем ее размер на 100x100.
2. Задаем прозрачный фон.
3. Располагаем PictureBox как показано на рисунке над первой клеткой поля игры.



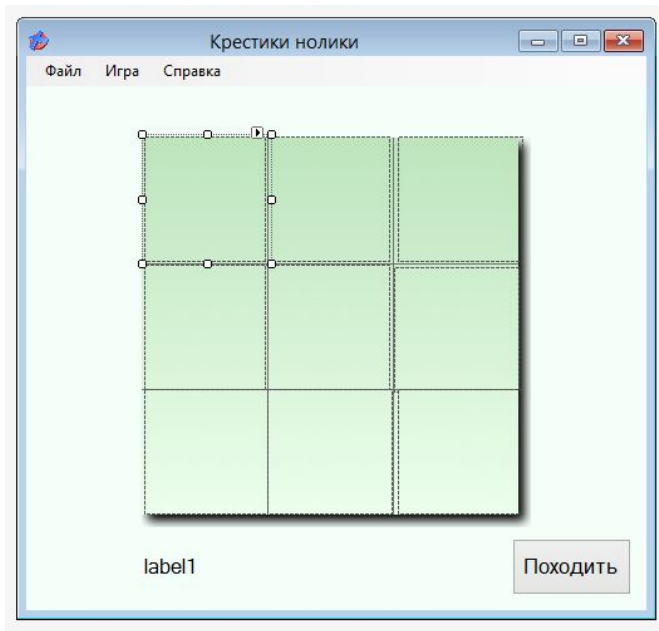


## ШАГ 4. ДОБАВЛЕНИЕ ОБЪЕКТОВ PictureBox НА ФОРМУ



1. Над остальными ячейками располагаем объекты `PictureBox`, копии первого объекта, согласно нумерации указанной на изображения.

## ШАГ 5. ДОБАВЛЕНИЕ ПРОГРАМНОГО КОДА ДЛЯ ОБЪЕКТОВ PICTUREBOX



1. Перед тем как написать программный код необходимо в папку `\Visual Studio 2010\Projects\Крестики нолики \ крестики нолики\bin\Debug\` необходимо перебросить файлы `x.png`, `0.png`, `none.png` из папки `Images`.
2. Кликаем два раза левой мыши по первому `PictureBox`.



## ШАГ 5. ДОБАВЛЕНИЕ ПРОГРАМНОГО КОДА ДЛЯ ОБЪЕКТОВ PICTUREBOX

```
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApplication4
{
    public partial class Form1 : Form
    {
        int[,] znacheniyеYacheyki = new int[3, 3] { { 0, 0, 0 }, { 0, 0, 0 }, { 0, 0, 0 } };

        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

1. Создаем двухмерный массив доступный для всех элементов в созданных в форме 3x3 элемента состоящий из целых чисел. Сразу заполняем его при объявлении нулями. Для пустых ячеек мы будем использовать 0, для «крестиков» - 1, а для «НОЛИКОВ» - -1.



## ШАГ 5. ДОБАВЛЕНИЕ ПРОГРАМНОГО КОДА ДЛЯ ОБЪЕКТОВ PICTUREBOX

```
private void pictureBox1_Click(object sender, EventArgs e)
{
    if (znachenijeYacheyki[0, 0] == 0)
    {
        pictureBox1.Image = Image.FromFile(@"x.png");
        znachenijeYacheyki[0, 0] = 1;
    }
    else if (znachenijeYacheyki[0, 0] == 1)
    {
        pictureBox1.Image = Image.FromFile(@"none.png");
        znachenijeYacheyki[0, 0] = 0;
    }
}
```

В процедуру при клике по первому **PictureBox**, добавляем оператор выбора который будет осуществлять проверку состояния ячейки массива. Если значение ячейки массива будет равно **0**, что значит в ней нет ни «нуля» ни «крестика», тогда в эту ячейку массива записуется **1** и в **PictureBox1** отображается изображение «крестика», а если значение ячейки массива будет равно **1** тогда в ней находится «крестик» и в нее записывается **0**, а отображается пустая ячейка.

## ШАГ 5. ДОБАВЛЕНИЕ ПРОГРАМНОГО КОДА ДЛЯ ОБЪЕКТОВ PICTUREBOX

1	2	3
4	5	6
7	8	9

[0,0]	[0,1]	[0,2]
[1,0]	[1,1]	[1,2]
[2,0]	[2,1]	[2,2]

Для остальных клеток поля добавляем код аналогично как и в первой лишь изменяя номер объекта `PictureBox` и адрес ячейки массива.

**ПРИМЕР** для второй ячейки:

```
if (znachenijeYacheyki[0, 1] == 0)
{
    pictureBox2.Image = Image.FromFile(@"x.png");
    znachenijeYacheyki[0, 1] = 1;
}
else
if (znachenijeYacheyki[0, 1] == 1)
{
    pictureBox2.Image = Image.FromFile(@"none.png");
    znachenijeYacheyki[0, 1] = 0;
}
```



## ШАГ 6. ДОБАВЛЕНИЕ ПРОГРАМНОГО КОДА ДЛЯ КНОПКИ ХОДИТЬ

```
namespace WindowsFormsApplication4
{
    public partial class Form1 : Form
    {
        int[,] znacheniyеYacheyki = new int[3, 3] { { 0, 0, 0 }, { 0,
        int i = 0, j = 0;

    }

    private void button1_Click(object sender, EventArgs e)
    {
        for (i = 0; i <= 2; i++)
            for (j = 0; j <= 2; j++)
                if (znacheniyеYacheyki[i, j] == 0)
                {
                    znacheniyеYacheyki[i, j] = -1;
                }
    }
}
```

В процедуру при клике по кнопке добавляем оператор цикла который осуществляет проверку всех ячеек начиная от первой на наличие пустых ячеек. И если ячейка пустая, то в нее записывается «НОЛИК», а именно в ячейку массива записывается **-1**.

Для удобства в дальнейшей работе переменные **i, j** которые используются для итерации циклов объявим для всей формы.





## ШАГ 6. ДОБАВЛЕНИЕ ПРОГРАМНОГО КОДА ДЛЯ КНОПКИ ХОДИТЬ

```
private void button1_Click(object sender, EventArgs e)
{
    for (i = 0; i <= 2; i++)
        for (j = 0; j <= 2; j++)
            if (znacheniyеYacheyki[i, j] == 0)
            {
                znacheniyеYacheyki[i, j] = -1;
                if (i == 0)
                {
                    if (j == 0) pictureBox1.Image = Image.FromFile(@"0.png");
                    else if (j == 1) pictureBox2.Image = Image.FromFile(@"0.png");
                    else pictureBox3.Image = Image.FromFile(@"0.png");
                }
                else if (i == 1)
                {
                    if (j == 0) pictureBox4.Image = Image.FromFile(@"0.png");
                    else if (j == 1) pictureBox5.Image = Image.FromFile(@"0.png");
                    else pictureBox6.Image = Image.FromFile(@"0.png");
                }
                else
                {
                    if (j == 0) pictureBox7.Image = Image.FromFile(@"0.png");
                    else if (j == 1) pictureBox8.Image = Image.FromFile(@"0.png");
                    else pictureBox9.Image = Image.FromFile(@"0.png");
                }
            }
        i = 3;
        break;
    }
}
```

Для отображения ноликов на игровом поле необходимо добавить программный код в тело цикла проверки ячеек на пустоту.

С помощью вложенного оператора ветвления будет происходить анализ адреса ячейки массива для вывода нолика в правильном **PictureBox**.

Также добавляем оператор **break** для преждевременного окончания цикла при нахождении пустой ячейки.



## ШАГ 6. ДОБАВЛЕНИЕ ПРОГРАМНОГО КОДА ДЛЯ КНОПКИ ХОДИТЬ

```
public partial class Form1 : Form
{
    int[,] znacheniyеYacheyki = new int[3, 3] { { 0, 0, 0 }, { 0,
    int i = 0, j = 0;
    string otvet = "Ваш Ход";

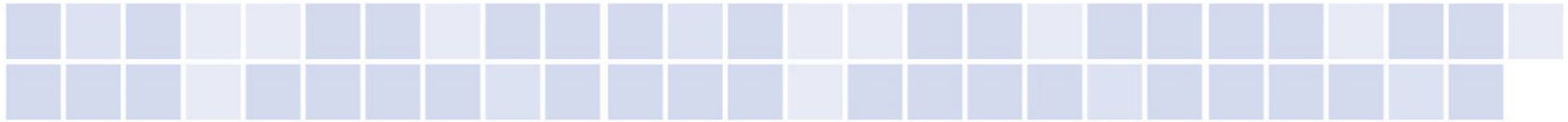
    public Form1()
    {
        InitializeComponent();
    }

}

private void Form1_Load(object sender, EventArgs e)
{
    label1.Text = "Ваш ход";
}
}
```

Для индикации состояния игры используется элемент интерфейса **Label1**. Так как игрок всегда ходит первым то при загрузке приложения необходимо в элементе **Label1** необходимо отражать словосочетания «Ваш ход». Для этого создадим переменную **otvet** которой присвоим это словосочетание. А при загрузке формы переменную необходимо присваивать элементу **Label1**, для создания необходимой процедуры необходимо предварительно дважды кликнуть по форме





## ШАГ 7. ДОБАВЛЕНИЕ ПУСТОЙ РАМКИ С КОДА ДЛЯ ПУНКТА МЕНЮ НОВАЯ ИГРА

```
        }  
    }  
  
private void новаяИграToolStripMenuItem_Click(object sender, EventArgs e)  
{  
    label1.Text = "Ваш ход";  
    for (i = 0; i <= 2; i++)  
        for (j = 0; j <= 2; j++)  
            znacheniyеYacheyki[i, j] = 0;  
  
    pictureBox1.Image = Image.FromFile(@"none.png");  
    pictureBox2.Image = Image.FromFile(@"none.png");  
    pictureBox3.Image = Image.FromFile(@"none.png");  
    pictureBox4.Image = Image.FromFile(@"none.png");  
    pictureBox5.Image = Image.FromFile(@"none.png");  
    pictureBox6.Image = Image.FromFile(@"none.png");  
    pictureBox7.Image = Image.FromFile(@"none.png");  
    pictureBox8.Image = Image.FromFile(@"none.png");  
  
    }  
}
```

При нажатии по команде новая игра будет происходить обнуление всех ячеек массива, замена всех «крестиков» и «ноликов» на пустые ячейки. Также вывод надписи «Ваш ход»



## ШАГ 8. ВЫВОД РЕЗУЛЬТАТА ИГРЫ

```
}  
  
private void button1_Click(object sender, EventArgs e)  
{  
    if (znacheniyеYacheyki[0, 0] == 1 && znacheniyеYacheyki[0, 1] == 1 && znacheniyеYacheyki[0, 2] ==  
    if (znacheniyеYacheyki[1, 0] == 1 && znacheniyеYacheyki[1, 1] == 1 && znacheniyеYacheyki[1, 2] ==  
    if (znacheniyеYacheyki[2, 0] == 1 && znacheniyеYacheyki[2, 1] == 1 && znacheniyеYacheyki[2, 2] ==  
    if (znacheniyеYacheyki[0, 0] == 1 && znacheniyеYacheyki[1, 0] == 1 && znacheniyеYacheyki[2, 0] ==  
    if (znacheniyеYacheyki[0, 1] == 1 && znacheniyеYacheyki[1, 1] == 1 && znacheniyеYacheyki[2, 1] ==  
    if (znacheniyеYacheyki[0, 2] == 1 && znacheniyеYacheyki[1, 2] == 1 && znacheniyеYacheyki[2, 2] ==  
    if (znacheniyеYacheyki[0, 0] == 1 && znacheniyеYacheyki[1, 1] == 1 && znacheniyеYacheyki[2, 2] ==  
    if (znacheniyеYacheyki[0, 2] == 1 && znacheniyеYacheyki[1, 1] == 1 && znacheniyеYacheyki[2, 0] ==  
    for (i = 0; i <= 2; i++)  
        for (j = 0; j <= 2; j++)  
            if (znacheniyеYacheyki[i, j] == 0)  
            {  
                znacheniyеYacheyki[i, j] = -1;  
                if (i == 0)  
                {  
                    if (j == 0) pictureBox1.Image = Image.FromFile(@"0.png");  
                    else if (j == 1) pictureBox2.Image = Image.FromFile(@"0.png");  
                    else pictureBox3.Image = Image.FromFile(@"0.png");  
  
                    if (j == 0) pictureBox4.Image = Image.FromFile(@"0.png");  
                    else if (j == 1) pictureBox5.Image = Image.FromFile(@"0.png");  
                    else pictureBox6.Image = Image.FromFile(@"0.png");  
                }  
                else  
                {  
                    if (j == 0) pictureBox7.Image = Image.FromFile(@"0.png");  
                    else if (j == 1) pictureBox8.Image = Image.FromFile(@"0.png");  
                    else pictureBox9.Image = Image.FromFile(@"0.png");  
                }  
                i = 3;  
                break;  
            }  
    if (znacheniyеYacheyki[0, 0] == -1 && znacheniyеYacheyki[0, 1] == -1 && znacheniyеYacheyki[0, 2] ==  
    if (znacheniyеYacheyki[1, 0] == -1 && znacheniyеYacheyki[1, 1] == -1 && znacheniyеYacheyki[1, 2] ==  
    if (znacheniyеYacheyki[2, 0] == -1 && znacheniyеYacheyki[2, 1] == -1 && znacheniyеYacheyki[2, 2] ==  
    if (znacheniyеYacheyki[0, 1] == -1 && znacheniyеYacheyki[1, 2] == -1 && znacheniyеYacheyki[2, 2] ==  
    if (znacheniyеYacheyki[0, 0] == -1 && znacheniyеYacheyki[1, 1] == -1 && znacheniyеYacheyki[2, 2] ==  
    if (znacheniyеYacheyki[0, 2] == -1 && znacheniyеYacheyki[1, 1] == -1 && znacheniyеYacheyki[2, 0] ==  
  
    label11.Text = otvet;  
}
```

Для проверки результатов ходов необходимо анализировать содержимое строк, столбцов и диагоналей. Если все три элемента равны **1** то это победа игрока, а если три **-1** то это поражение игрока.

Проверку победы необходимо проводить перед ходом компьютера, а проверку поражения после.

Последней командой процедуры станет вывод на экран результатов хода.



## ШАГ 8. ВЫВОД РЕЗУЛЬТАТА ИГРЫ

Программный код для проверки победы пользователя:

```
if (znacheniyеYacheyki[0, 0] == 1 && znacheniyеYacheyki[0, 1] == 1 && znacheniyеYacheyki[0, 2] == 1) otvet = "Вы победили";  
if (znacheniyеYacheyki[1, 0] == 1 && znacheniyеYacheyki[1, 1] == 1 && znacheniyеYacheyki[1, 2] == 1) otvet = "Вы победили";  
if (znacheniyеYacheyki[2, 0] == 1 && znacheniyеYacheyki[2, 1] == 1 && znacheniyеYacheyki[2, 2] == 1) otvet = "Вы победили";  
if (znacheniyеYacheyki[0, 0] == 1 && znacheniyеYacheyki[1, 0] == 1 && znacheniyеYacheyki[2, 0] == 1) otvet = "Вы победили";  
if (znacheniyеYacheyki[0, 1] == 1 && znacheniyеYacheyki[1, 1] == 1 && znacheniyеYacheyki[2, 1] == 1) otvet = "Вы победили";  
if (znacheniyеYacheyki[0, 2] == 1 && znacheniyеYacheyki[1, 2] == 1 && znacheniyеYacheyki[2, 2] == 1) otvet = "Вы победили";  
if (znacheniyеYacheyki[0, 0] == 1 && znacheniyеYacheyki[1, 1] == 1 && znacheniyеYacheyki[2, 2] == 1) otvet = "Вы победили";  
if (znacheniyеYacheyki[0, 2] == 1 && znacheniyеYacheyki[1, 1] == 1 && znacheniyеYacheyki[2, 0] == 1) otvet = "Вы победили";
```

Программный код для проверки победы пользователя:

```
if (znacheniyеYacheyki[0, 0] == -1 & znacheniyеYacheyki[0, 1] == -1 & znacheniyеYacheyki[0, 2] == -1) otvet = "Вы проиграли";  
if (znacheniyеYacheyki[1, 0] == -1 & znacheniyеYacheyki[1, 1] == -1 & znacheniyеYacheyki[1, 2] == -1) otvet = "Вы проиграли";  
if (znacheniyеYacheyki[2, 0] == -1 & znacheniyеYacheyki[2, 1] == -1 & znacheniyеYacheyki[2, 2] == -1) otvet = "Вы проиграли";  
if (znacheniyеYacheyki[0, 1] == -1 & znacheniyеYacheyki[1, 2] == -1 & znacheniyеYacheyki[2, 2] == -1) otvet = "Вы проиграли";  
if (znacheniyеYacheyki[0, 0] == -1 & znacheniyеYacheyki[1, 1] == -1 & znacheniyеYacheyki[2, 2] == -1) otvet = "Вы проиграли";  
if (znacheniyеYacheyki[0, 2] == -1 & znacheniyеYacheyki[1, 1] == -1 & znacheniyеYacheyki[2, 0] == -1) otvet = "Вы проиграли";  
label1.Text = otvet;
```



## ШАГ 9 УЛУЧШЕНИЕ «ИГРАБЕЛЬНОСТИ»

```
public partial class Form1 : Form
{
    int[,] znacheniyеYacheyki = new int[3, 3] { { 0, 0, 0 }, { 0, 0, 0 }, { 0, 0, 0 } };
    int i = 0, j = 0;
    string otvet = "Ваа Ход";
    bool uslovie = true;

    public Form1()
    {
        // ...
    }

    private void button1_Click(object sender, EventArgs e)
    {
        uslovie = true;
        if (znacheniyеYacheyki[0, 0] == 1 && znacheniyеYacheyki[0, 1] == 1 && znacheniyеYacheyki[0, 2] == 1 && znacheniyеYacheyki[1, 0] == 1 && znacheniyеYacheyki[1, 1] == 1 && znacheniyеYacheyki[1, 2] == 1 && znacheniyеYacheyki[2, 0] == 1 && znacheniyеYacheyki[2, 1] == 1 && znacheniyеYacheyki[2, 2] == 1)
        {
            otvet = "Выиграл";
        }
        while (uslovie)
        {
            Random rand = new Random();
            i = rand.Next(0, 3);
            j = rand.Next(0, 3);
            if (znacheniyеYacheyki[i, j] == 0)
            {
                uslovie = false;
                znacheniyеYacheyki[i, j] = -1;

                if (i == 0)
                {
                    if (j == 0) pictureBox1.Image = Image.FromFile(@"0.png");
                    else if (j == 1) pictureBox2.Image = Image.FromFile(@"0.png");
                    else pictureBox3.Image = Image.FromFile(@"0.png");
                }
                else if (i == 1)
                {
                    if (j == 0) pictureBox4.Image = Image.FromFile(@"0.png");
                    else if (j == 1) pictureBox5.Image = Image.FromFile(@"0.png");
                    else pictureBox6.Image = Image.FromFile(@"0.png");
                }
                else if (i == 2)
                {
                    if (j == 0) pictureBox7.Image = Image.FromFile(@"0.png");
                    else if (j == 1) pictureBox8.Image = Image.FromFile(@"0.png");
                    else pictureBox9.Image = Image.FromFile(@"0.png");
                }
            }
        }
    }
}
```

Для улучшения «играбельности» вместо последовательного вывода в первые попавшиеся пустые ячейки **«НОЛИКОВ»**, реализуем вывод через генератор случайных чисел. Для этого необходимо добавить одну логическую переменную **uslovie**, и изменить тип цикла с **For** на **While**, так как нам неизвестно точное количество повторений генератора случайных чисел пока он не попадет в пустую ячейку.