

제10장

TicTacToe 게임



교육 내용 소개

우리가 이번 시간에 배울 내용은 C#이라는 프로그래밍 언어를 이용해서 객체지향 프로그래밍을 하는 방법을 배웁니다. ‘C++ 프로그래밍만 배우면 되지 않습니까? 자바를 하면 되지 않습니까? C# 프로그래밍도 배워야 하나요?’

새 프로젝트 구성

Windows Forms 앱(.NET Framework) C# Windows 데스크톱

프로젝트 이름(I)

TicTacToe

위치(L)

C:\Users\USER\source\repos

솔루션 이름(M) ⓘ

TicTacToe

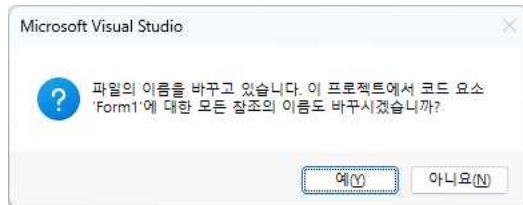
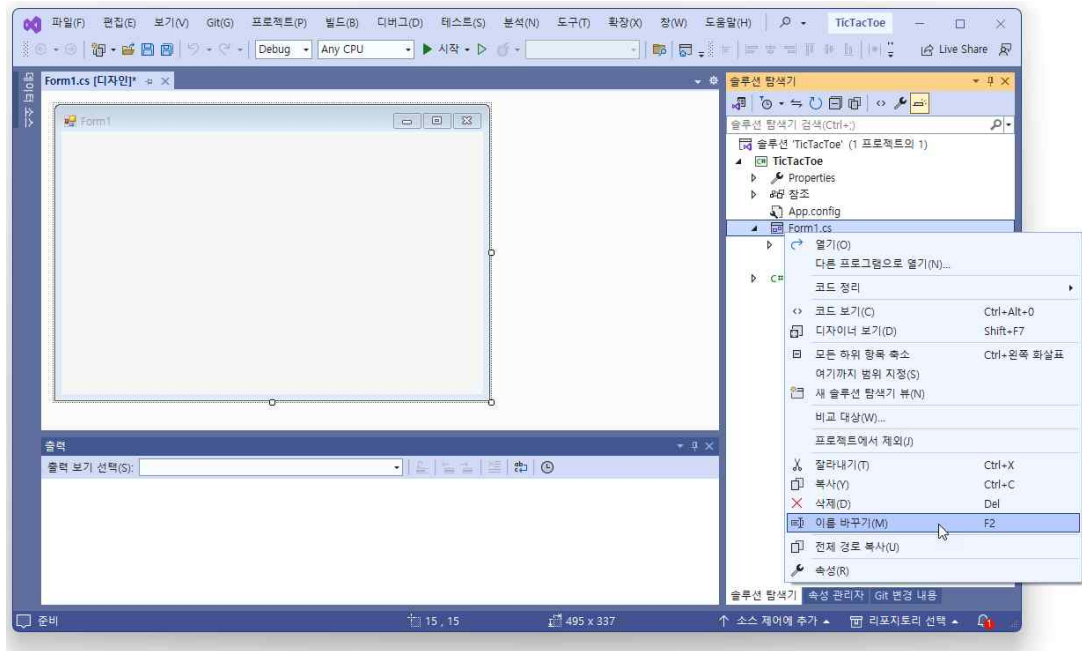
☐ 솔루션 및 프로젝트를 같은 디렉터리에 배치(D)

프레임워크(F)

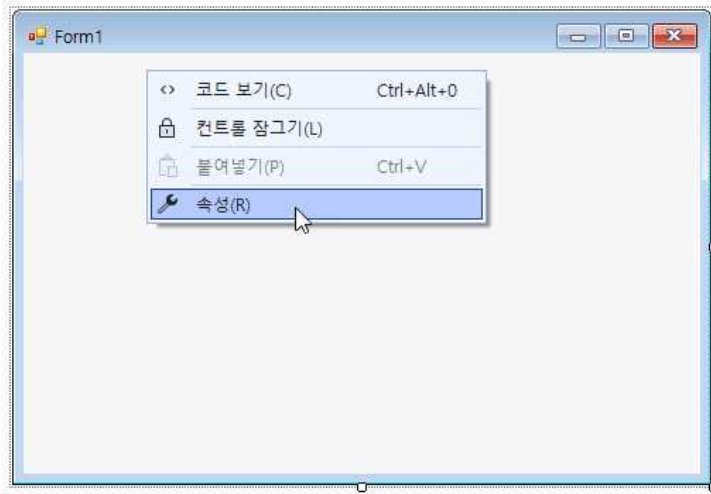
.NET Framework 4.7.2

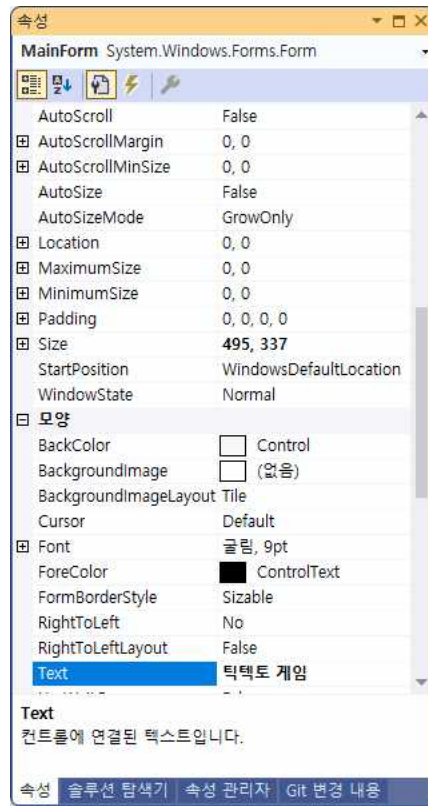
"C:\Users\USER\source\repos\TicTacToe\TicTacToe\TicTacToe"에 프로젝트이(가) 만들어집니다.

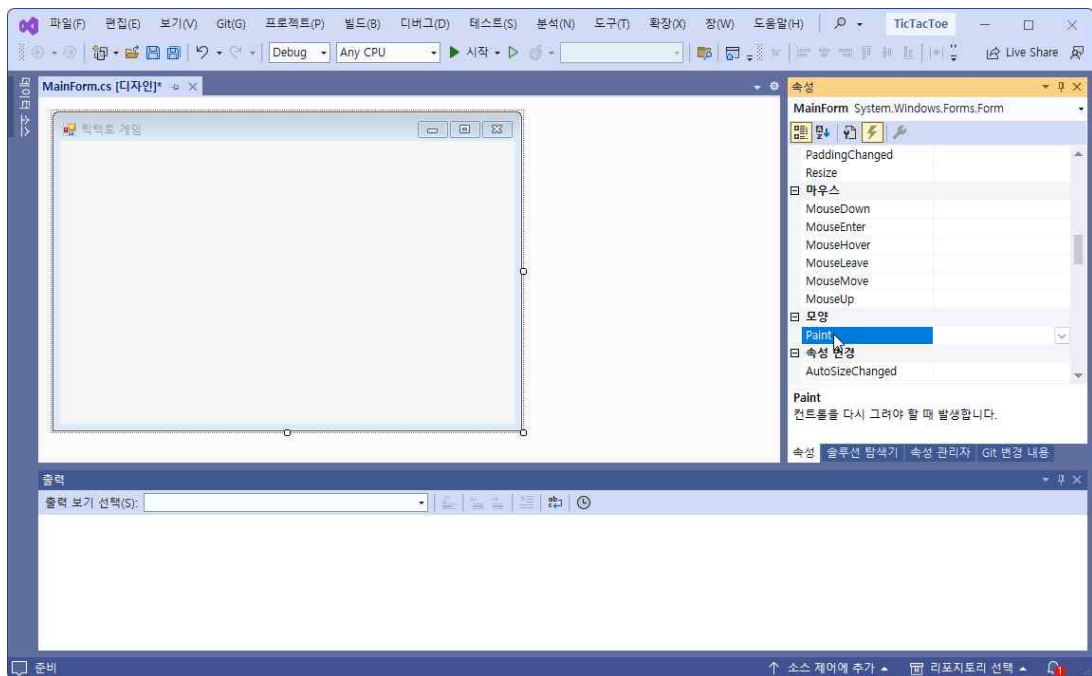
뒤로(B) 만들기(C)



Program.cs는 TicTacToe.cs으로







```
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 TicTacToe
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();

            private void MainForm_Paint(object sender, PaintEventArgs e)
            {
                Graphics g = this.CreateGraphics();

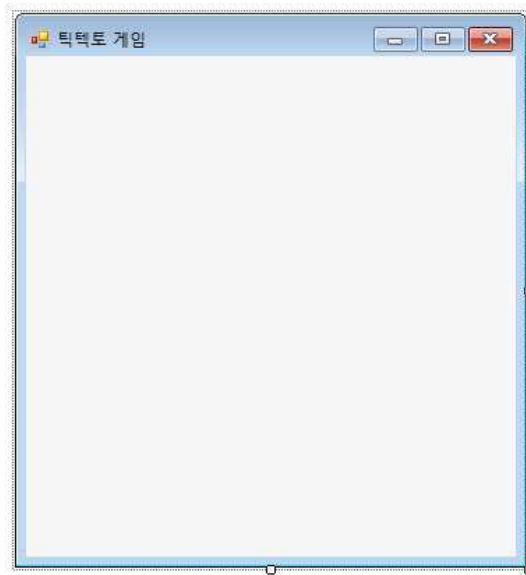
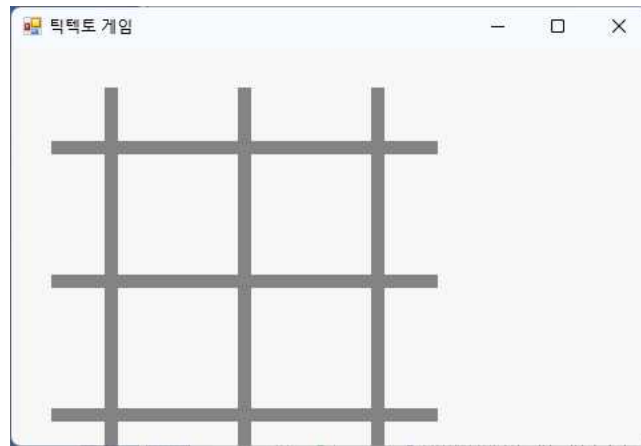
                Pen p = new Pen(new SolidBrush(Color.Gray), 10);

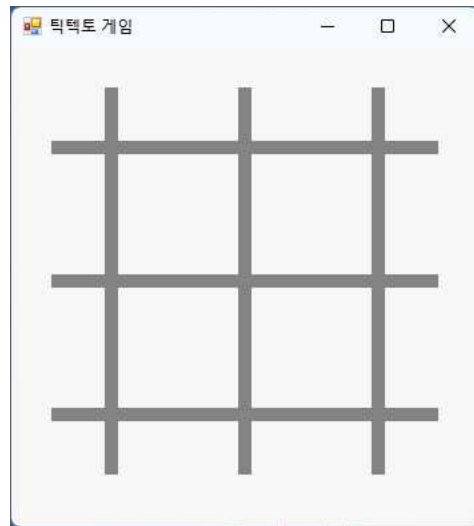
                g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
                g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
                g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);

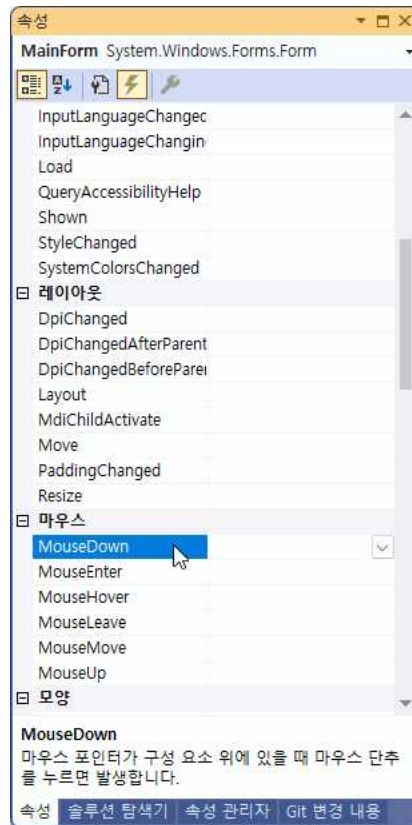
                g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
                g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
                g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);

            }
        }
    }
}

```







```

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 TicTacToe
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();

            private void MainForm_Paint(object sender, PaintEventArgs e)
            {
                Graphics g = this.CreateGraphics();

                Pen p = new Pen(new SolidBrush(Color.Gray), 10);

                g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
                g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
                g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);

                g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
                g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
                g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);
            }

            private void MainForm_MouseDown(object sender,
MouseEventArgs e)
            {

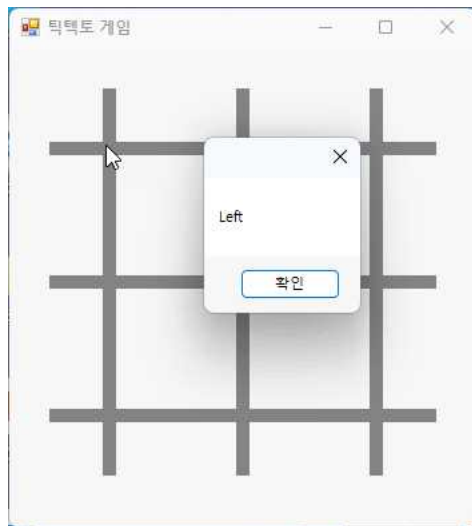
```

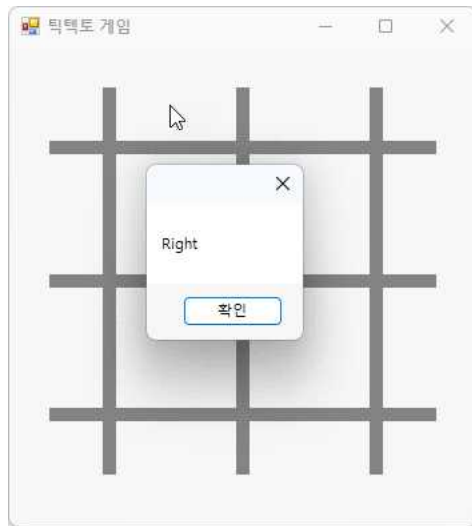
```

        if (e.Button == MouseButton.Left)
        {
            MessageBox.Show("Left");
        }
        else
        {
            MessageBox.Show("Right");
        }
    }
}
}

```

다음 시간부터는 C# 라이브러리를 이용해서 폼 윈도우 화면을 갖는 프로





```
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 TicTacToe
{
    public partial class MainForm : Form
    {
        public MainForm()
```

```

    {
        InitializeComponent();
    }

    private void MainForm_Paint(object sender, PaintEventArgs e)
    {
        Graphics g = this.CreateGraphics();

        Pen p = new Pen(new SolidBrush(Color.Gray), 10);

        g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
        g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
        g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);

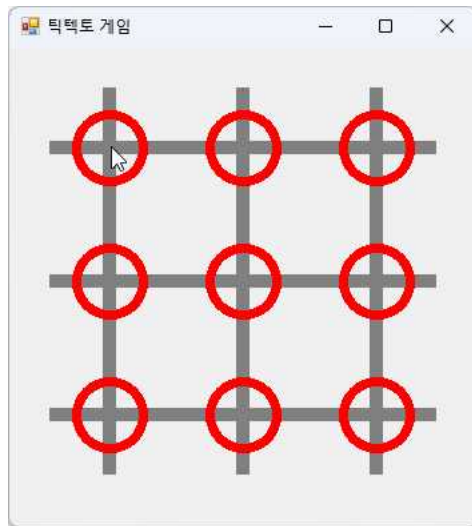
        g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
        g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
        g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);
    }

    private void MainForm_MouseDown(object sender,
    MouseEventArgs e)
    {
        Graphics g = this.CreateGraphics();

        if (e.Button == MouseButton.Left)
        {
            for (int i = 0; i < 3; i++)

```

```
        {  
            for (int j = 0; j < 3; j++)  
            {  
                g.DrawEllipse(  
                    new Pen(new SolidBrush(Color.Red), 7),  
                    i * 100 + 50,  
                    j * 100 + 50,  
                    50, 50);  
            }  
        }  
    }  
    else  
    {  
        MessageBox.Show("Right");  
    }  
}  
}
```



```
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 TicTacToe
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();

            private void MainForm_Paint(object sender, PaintEventArgs e)
            {
                Graphics g = this.CreateGraphics();

                Pen p = new Pen(new SolidBrush(Color.Gray), 10);

                g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
                g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
                g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);

                g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
                g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
                g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);
            }

            private void MainForm_MouseDown(object sender, MouseEventArgs e)
            {
                Graphics g = this.CreateGraphics();

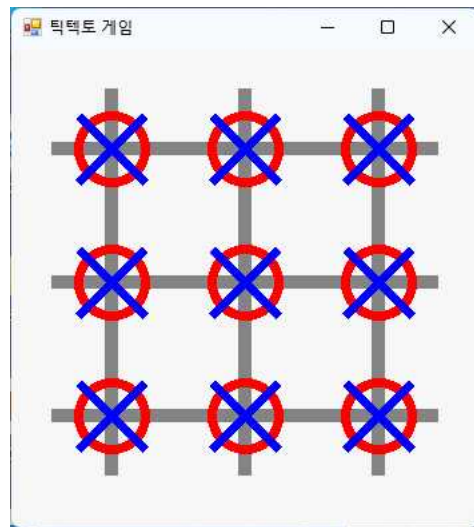
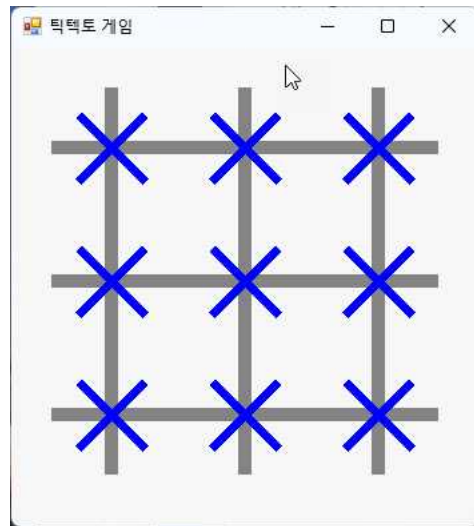
                if (e.Button == MouseButton.Left)
                {

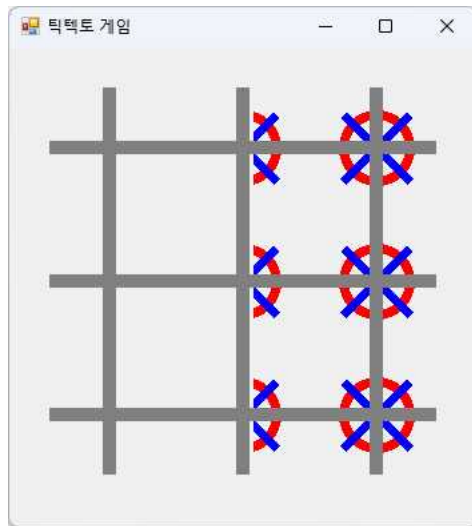
```

```

        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                g.DrawEllipse(
                    new Pen(new SolidBrush(Color.Red), 7),
                    i * 100 + 50,
                    j * 100 + 50,
                    50, 50);
            }
        }
    }
else {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7),
                i * 100 + 50, j * 100 + 50,
                i * 100 + 100, j * 100 + 100);
            g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7),
                i * 100 + 100, j * 100 + 50,
                i * 100 + 50, j * 100 + 100);
        }
    }
}
}
}
}

```





```
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 TicTacToe
{
    public partial class MainForm : Form
```

```

{
    public MainForm()
    {
        InitializeComponent();
    }

    private void MainForm_Paint(object sender, PaintEventArgs e)
    {
        Graphics g = this.CreateGraphics();

        Pen p = new Pen(new SolidBrush(Color.Gray), 10);

        g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
        g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
        g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);

        g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
        g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
        g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);
    }
}

```

```
int[, ] click_array = new int[3, 3];
```

```

private void MainForm_MouseDown(object sender, MouseEventArgs e)
{
    Graphics g = this.CreateGraphics();

    if (e.Button == MouseButtons.Left)
    {
        for (int i = 0; i < 3; i++)

```

```

        {
            for (int j = 0; j < 3; j++)
            {
                click_array[i, j] = 1; //O
            }
        }
    }
    else
    {
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                click_array[i, j] = 2; //O
            }
        }
    }

    Invalidate();
}
}
}

```

Invalidate 함수를 호출하면 Paint 이벤트가 발생합니다. 그러면 Paint 델리게이트에 연결되어 있는 MainForm_Paint 핸들러 함수가 실행됩니다. 여기에서 click_array 각각의 배열 값을 확인하여 1인 경우 0를 출력하도록 합니다.

```

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 TicTacToe
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();

            private void MainForm_Paint(object sender, PaintEventArgs e)
            {
                Graphics g = this.CreateGraphics();

                Pen p = new Pen(new SolidBrush(Color.Gray), 10);

                g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
                g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
                g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);

                g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
            }
        }
    }
}

```

```

g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);

```

```

        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                if (click_array[i, j] == 1)
                {
                    g.DrawEllipse(
                        new Pen(new SolidBrush(Color.Red), 7),
                        i * 100 + 50,
                        j * 100 + 50,
                        50, 50);
                }
            }
        }
    }
}

```

```

int[,] click_array = new int[3, 3];

```

```

private void MainForm_MouseDown(object sender, MouseEventArgs e)
{
    Graphics g = this.CreateGraphics();

    if (e.Button == MouseButton.Left)
    {
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)

```



```

        {
            click_array[i, j] = 1; //O
        }
    }
else
{
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            click_array[i, j] = 2; //O
        }
    }
}

Invalidate();
}
}
}

```

이제 값이 2일 경우에 X를 표시하도록 코드를 추가 합니다.

```

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 TicTacToe
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();

            private void MainForm_Paint(object sender, PaintEventArgs e)
            {
                Graphics g = this.CreateGraphics();

                Pen p = new Pen(new SolidBrush(Color.Gray), 10);

                g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
                g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
                g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);

                g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
                g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
                g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);

                for (int i = 0; i < 3; i++)
                {

```

```

        for (int j = 0; j < 3; j++)
        {
            if (click_array[i, j] == 1)
            {
                g.DrawEllipse(
                    new Pen(new SolidBrush(Color.Red), 7),
                    i * 100 + 50,
                    j * 100 + 50,
                    50, 50);
            }
            else if (click_array[i, j] == 2)
            {
                g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i *
100 + 50, j * 100 + 50, i * 100 + 100, j * 100 + 100);
                g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i *
100 + 100, j * 100 + 50, i * 100 + 50, j * 100 + 100);
            }
        }
    }
}

```

```

int[,] click_array = new int[3, 3];

```

```

private void MainForm_MouseDown(object sender, MouseEventArgs e)
{
    Graphics g = this.CreateGraphics();

    if (e.Button == MouseButtons.Left)
    {
        for (int i = 0; i < 3; i++)

```

```

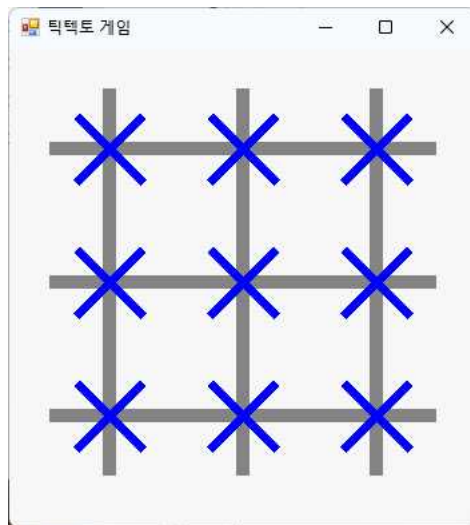
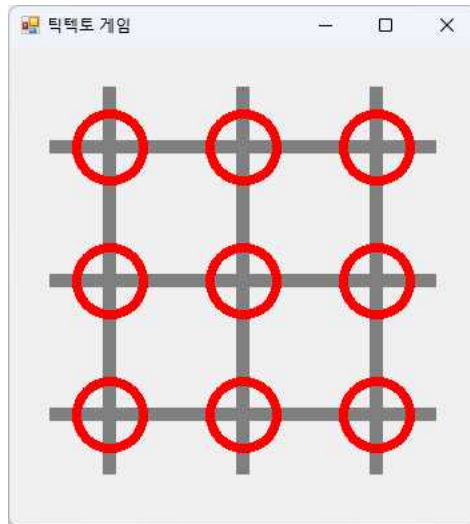
        {
            for (int j = 0; j < 3; j++)
            {
                click_array[i, j] = 1; //O
            }
        }
    }
    else
    {
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                click_array[i, j] = 2; //O
            }
        }
    }

    Invalidate();
}
}
}

```

프로그램을 실행하여 버튼을 누를 경우에는 click_array 배열에 1, 혹은 2 값을 할당하기만 합니다. 그런 다음 Invalidate를 호출하여 Paint 이벤트가 강제로 발생되도록 합니다. 이제 MainForm_Paint 핸들러 함수가 자동으로 실행이 되고, 여기에서 배열 값을 체크 한 후에 1이면 O를, 2이면 X를 표시합니다. 물론 배열에는 9개 값들이 모두가 1 혹은 2로 할당 되기 때문에 모두 O 혹은 X를 표시하게 됩니다. 그리고 폼 윈도우 일부가

가려졌다가 다시 표시되어도 O 혹은 X 표시가 잘되고 있습니다.





코드 추상화로 함수 만들기

이제까지 작성한 코드는 다음과 같습니다. 우리가 작성한 코드는 크게 보면 Paint 이벤트 발생할 때, 그리고 마우스를 클릭할 때 두 가지에 대해서 코드를 추가 작성한 것이 전부입니다. 그러면 그 안에 있는 코드들을 간단히 추상화 해 보겠습니다. 먼저 Paint 이벤트가 발생할 경우 실행되는 핸들러 함수인 MainForm_Paint 함수를 보겠습니다. 아래에서 강조한 코드는 무엇을 하는 코드일까요.?

```
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;

public partial class MainForm : Form
{
    public MainForm()
    {
        InitializeComponent();
    }

    private void MainForm_Paint (object sender, PaintEventArgs e)
    {
```

```

Graphics g = this.CreateGraphics();

Pen p = new Pen(new SolidBrush(Color.Gray), 10);

g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);

g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);

```

```

for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        if (click_array[i, j] == 1)
        {
            g.DrawEllipse(
                new Pen(new SolidBrush(Color.Red), 7),
                i * 100 + 50,
                j * 100 + 50,
                50, 50);
        }
        else if (click_array[i, j] == 2)
        {
            g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i * 100 +
50, j * 100 + 50,
i * 100 + 100, j * 100 + 100);
            g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i * 100 +
100, j * 100 + 50,

```

```

        i * 100 + 50, j * 100 + 100);
    }
}
}
}

```

```

int[,] click_array = new int[3, 3];

```

```

private void MainForm_MouseDown(object sender, MouseEventArgs e)
{
    Graphics g = this.CreateGraphics();

    if (e.Button == MouseButton.Left)
    {
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                click_array[i, j] = 1; //0
            }
        }
    }
    else
    {
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                click_array[i, j] = 2; //0
            }
        }
    }
}

```



```

        }

        Invalidate();
    }
}

```

앞서 강조한 코드는 게임판 격자(grid)를 그리는 코드입니다. 따라서 DrawGrid라는 함수로 코드 추상화하겠습니다.

```

public partial class MainForm : Form
{
    public MainForm()
    {
        InitializeComponent();
    }
}

```

```

public void DrawGrid()
{
    Graphics g = this.CreateGraphics();

    Pen p = new Pen(new SolidBrush(Color.Gray), 10);

    g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
    g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
    g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);

    g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
    g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
    g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);
}

```

```

private void MainForm_Paint(object sender, PaintEventArgs e)
{
    DrawGrid();

    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            if (click_array[i, j] == 1)
            {
                g.DrawEllipse(
                    new Pen(new SolidBrush(Color.Red), 7),
                    i * 100 + 50,
                    j * 100 + 50,
                    50, 50);
            }
            else if (click_array[i, j] == 2)
            {
                g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i * 100 +
50, j * 100 + 50,
i * 100 + 100, j * 100 + 100);
                g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i * 100 +
100, j * 100 + 50,
                    i * 100 + 50, j * 100 + 100);
            }
        }
    }
}

```

그러면 아래에 강조한 코드는 무엇을 하는 코드일까요?

```

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;

public partial class MainForm : Form
{
    public MainForm()
    {
        InitializeComponent();
    }

    public void DrawGrid()
    {
        Graphics g = this.CreateGraphics();

        Pen p = new Pen(new SolidBrush(Color.Gray), 10);

        g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
        g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
        g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);

        g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
        g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
        g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);
    }
}

```

```
private void MainForm_Paint(object sender, PaintEventArgs e)
{
```

```
    DrawGrid();
```

```
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                if (click_array[i, j] == 1)
                {
                    g.DrawEllipse(
                        new Pen(new SolidBrush(Color.Red), 7),
                        i * 100 + 50,
                        j * 100 + 50,
                        50, 50);
                }
                else if (click_array[i, j] == 2)
                {
                    g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i * 100 +
50, j * 100 + 50,
i * 100 + 100, j * 100 + 100);
                    g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i * 100 +
100, j * 100 + 50,
                        i * 100 + 50, j * 100 + 100);
                }
            }
        }
    }
```

```
int[,] click_array = new int[3, 3];
```

```

private void MainForm_MouseDown(object sender, MouseEventArgs e)
{
    Graphics g = this.CreateGraphics();

    if (e.Button == MouseButton.Left)
    {
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                click_array[i, j] = 1; //O
            }
        }
    }
    else
    {
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                click_array[i, j] = 2; //X
            }
        }
    }

    Invalidate();
}
}

```

click_array 배열에 있는 값을 확인하여 1이면 O를 2이면 X를 표시하는 코드입니다. 따라서 DrawOX라는 함수로 코드 추상화하겠습니다.

```

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;

public partial class MainForm : Form
{
    public MainForm()
    {
        InitializeComponent();

    }

    public void DrawGrid()
    {
        Graphics g = this.CreateGraphics();

        Pen p = new Pen(new SolidBrush(Color.Gray), 10);

        g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
        g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
        g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);

        g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
        g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
        g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);
    }
}

```

```
}
```

```
public void DrawOX()
{
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            if (click_array[i, j] == 1)
            {
                g.DrawEllipse(
                    new Pen(new SolidBrush(Color.Red), 7),
                    i * 100 + 50,
                    j * 100 + 50,
                    50, 50);
            }
            else if (click_array[i, j] == 2)
            {
                g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i * 100 +
50, j * 100 + 50,
i * 100 + 100, j * 100 + 100);
                g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i * 100 +
100, j * 100 + 50,
                    i * 100 + 50, j * 100 + 100);
            }
        }
    }
}
```

```
private void MainForm_Paint(object sender, PaintEventArgs e)
{
```

```
        DrawGrid();  
        DrawOX();  
    }
```

```
int[,] click_array = new int[3, 3];
```

```
private void MainForm_MouseDown(object sender, MouseEventArgs e)  
{  
    Graphics g = this.CreateGraphics();  
  
    if (e.Button == MouseButtons.Left)  
    {  
        for (int i = 0; i < 3; i++)  
        {  
            for (int j = 0; j < 3; j++)  
            {  
                click_array[i, j] = 1; //O  
            }  
        }  
    }  
    else  
    {  
        for (int i = 0; i < 3; i++)  
        {  
            for (int j = 0; j < 3; j++)  
            {  
                click_array[i, j] = 2; //O  
            }  
        }  
    }  
}
```



```

        Invalidate();
    }
}

```

하지만 여기에는 한가지 문제가 있습니다. 그래픽스 개체가 없어서 일을 정리 하도록 코드를 추가 해야 합니다.

```

public void DrawOX()
{
    Graphics g = this.CreateGraphics();

    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            if (click_array[i, j] == 1)
            {
                g.DrawEllipse(
                    new Pen(new SolidBrush(Color.Red), 7),
                    i * 100 + 50,
                    j * 100 + 50,
                    50, 50);
            }
            else if (click_array[i, j] == 2)
            {
                g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i * 100 + 50, j
* 100 + 50, i * 100 + 100, j * 100 + 100);
                g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i * 100 + 100, j
* 100 + 50, i * 100 + 50, j * 100 + 100);
            }
        }
    }
}

```

```
}
```

그랬더니 Paint 이벤트 발생할 때 실행되는 MainForm_Paint 핸들러 함수가 다음과 같이 아주 깔끔하게 정리되었습니다.

```
private void MainForm_Paint(object sender, PaintEventArgs e)
{
    DrawGrid();
    DrawOX();
}
```

즉, 윈도우를 새로 그려야 할 때마다 판을 그린 후에 배열에 있는 값에 따라 O 혹은 X를 적절히 표시함을 쉽게 할 수 있습니다.

이제 마우스 버튼을 누를 경우 실행되는 코드를 살펴보겠습니다 먼저, 아래 강조된 코드는 무엇을 하는 코드인지 설명해보세요.

```
private void MainForm_MouseDown(object sender, MouseEventArgs e)
{
```

```
    Graphics g = this.CreateGraphics();
```

```
    if (e.Button == MouseButton.Left)
```

```
    {
```

```
        for (int i = 0; i < 3; i++)
```

```
        {
```

```
            for (int j = 0; j < 3; j++)
```

```
            {
```

```

        click_array[i, j] = 1; //O
    }
}
else
{
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            click_array[i, j] = 2; //O
        }
    }
}

```

```

        Invalidate();
    }
}

```

마우스 버튼을 누를 때 실행되는 핸들러 함수인데, 왼쪽 버튼을 누르면 배열에 1 값을, 오른쪽 버튼을 누르면 배열에 2 값을 할당합니다. 현재로는 단순히 모든 배열 위치에 동일하게 1 혹은 2를 할당하고 있습니다.

마우스로 특정 격자 위치를 클릭할 경우 이에 대응되는 배열 요소에 값 1 혹은 2를 할당하는 것이므로 SetArrayFromClick이라는 함수로 코드를 추상화해보겠습니다.

```

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;

public partial class MainForm : Form
{
    public MainForm()
    {
        InitializeComponent();

        public void DrawGrid()
        {
            Graphics g = this.CreateGraphics();

            Pen p = new Pen(new SolidBrush(Color.Gray), 10);

            g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
            g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
            g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);

```

```

        g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
        g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
        g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);
    }

    public void DrawOX()
    {
        Graphics g = this.CreateGraphics();

        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                if (click_array[i, j] == 1)
                {
                    g.DrawEllipse(
                        new Pen(new SolidBrush(Color.Red), 7),
                        i * 100 + 50,
                        j * 100 + 50,
                        50, 50);
                }
                else if (click_array[i, j] == 2)
                {
                    g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i
* 100 + 50, j * 100 + 50,
i * 100 + 100, j * 100 + 100);
                    g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i

```

```

* 100 + 100, j * 100 + 50,
                                i * 100 + 50, j * 100 + 100);
        }
    }
}
}

```

```

private void MainForm_Paint(object sender, PaintEventArgs e)
{
    DrawGrid();
    DrawOX();
}

```

```

int[,] click_array = new int[3, 3];

```

```

public void SetArrayFromClick()
{
    Graphics g = this.CreateGraphics();

    if (e.Button == MouseButtons.Left)
    {
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                click_array[i, j] = 1; //0
            }
        }
    }
}

```

```

        }
    }
    else
    {
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                click_array[i, j] = 2; //0
            }
        }
    }
}

```

```

private void MainForm_MouseDown(object sender, MouseEventArgs e)
{

```

```

    SetArrayFromClick();

```

```

    Invalidate();

```

```

}

```

```

}

```



데이터 추상화로 클래스 만들기

코드 추상화를 하여 함수들을 만들었습니다. 이제 관련 된 함수들을 밖으로 꺼내어 TicTacToe라는 클래스로 만들겠습니다. click_array 배열도 같이 옮겨서 데이터 추상화를 하도록 하겠습니다. 먼저 다음과 같이 비어 있는 클래스를 선언합니다.

```
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;
```

```
public class TicTacToeGame  
{  
  
}
```

```
public partial class MainForm : Form  
{  
    public MainForm()  
    {
```



```

        InitializeComponent();
    }

    public void DrawGrid()
    {
        Graphics g = this.CreateGraphics();

        Pen p = new Pen(new SolidBrush(Color.Gray), 10);

        g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
        g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
        g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);

        g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
        g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
        g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);
    }

```

이제 코드 추상화하여 만든 함수들, 그리고 배열을 TicTacToeGame 클래스로 옮깁니다.

```

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;
```

```
public class TicTacToeGame
{
```

```
    public void DrawGrid()
    {
        Graphics g = this.CreateGraphics();

        Pen p = new Pen(new SolidBrush(Color.Gray), 10);

        g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
        g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
        g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);

        g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
        g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
        g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);
    }
```

```
    public void DrawOX()
    {
        Graphics g = this.CreateGraphics();

        for (int i = 0; i < 3; i++)
        {
```

```

        for (int j = 0; j < 3; j++)
        {
            if (click_array[i, j] == 1)
            {
                g.DrawEllipse(
                    new Pen(new SolidBrush(Color.Red), 7),
                    i * 100 + 50,
                    j * 100 + 50,
                    50, 50);
            }
            else if (click_array[i, j] == 2)
            {
                g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i
* 100 + 50, j * 100 + 50,
i * 100 + 100, j * 100 + 100);
                g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i
* 100 + 100, j * 100 + 50,
                    i * 100 + 50, j * 100 + 100);
            }
        }
    }

    int[,] click_array = new int[3, 3];

    public void ClickAndSetArray()
    {

```

```

        Graphics g = this.CreateGraphics();

        if (e.Button == MouseButton.Left)
        {
            for (int i = 0; i < 3; i++)
            {
                for (int j = 0; j < 3; j++)
                {
                    click_array[i, j] = 1; //O
                }
            }
        }
        else
        {
            for (int i = 0; i < 3; i++)
            {
                for (int j = 0; j < 3; j++)
                {
                    click_array[i, j] = 2; //O
                }
            }
        }
    }
}

```

```

public partial class MainForm : Form
{

```

```

public MainForm()
{
    InitializeComponent();
}

private void MainForm_Paint(object sender, PaintEventArgs e)
{
    DrawGrid();
    DrawOX();
}

private void MainForm_MouseDown(object sender, MouseEventArgs e)
{
    ClickAndSetArray();
    Invalidate();
}
}

```

폼 윈도를 새로 그려야 할 때 발생하는 이벤트인 Paint 이벤트 혹은 델리게이트에 연결되어 실행되는 핸들 함수 MainForm_Paint와 마우스 버튼을 눌렀을 때 실행되는 핸들러 함수 MainForm_MouseDown 핸들러 함수 두 개를 제외한 모든 함수와 배열 코드를 TicTacToeGame 클래스로 옮겼습니다. 옮기고 보니 MainForm 클래스가 매우 보기 편하게 정리되었네요. 하지만 오류가 발생합니다. 클래스는 객체만들라고 있는 것이므로 객체를 만든 후 멤버 함수를 호출하도록 코드를 변경해야 합니다. 다음과 같이 말이죠.

```

public partial class MainForm : Form
{
    TicTacToeGame game = new TicTacToeGame();

    public MainForm()
    {
        InitializeComponent();

        private void MainForm_Paint(object sender, PaintEventArgs e)
        {
            game.DrawGrid();
            game.DrawOX();
        }

        private void MainForm_MouseDown(object sender, MouseEventArgs e)
        {
            game.ClickAndSetArray();

            Invalidate();
        }
    }
}

```

하지만 불행이도 아래 강조한 부분에서 오류가 발생합니다. CreateGraphics는 MainForm 클래스의 멤버 함수이므로 MainForm 객체를 파라미터로 전달받아 CreateGraphics 멤버 함수를 호출해야 합니다. 먼저 다음과 같이 game 객체의 멤버 함수 호출 시 MainForm 객체를 파

라미터로 전달합니다.

```
public partial class MainForm : Form
{
    TicTacToeGame game = new TicTacToeGame();

    public MainForm()
    {
        InitializeComponent();
    }

    private void MainForm_Paint(object sender, PaintEventArgs e)
    {
        game.DrawGrid(this);
        game.DrawOX(this);
    }

    private void MainForm_MouseDown(object sender, MouseEventArgs e)
    {
        game.ClickAndSetArray(this);

        Invalidate();
    }
}
```

this를 파라미터로 전달받기 위하여 TicTacToeGame 클래스를 다음과 같이 수정합니다.

```
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;
```

```
public class TicTacToeGame
{
```

```
    public void DrawGrid(Form f)
```

```
    {
```

```
        Graphics g = f.CreateGraphics();
```

```
        Pen p = new Pen(new SolidBrush(Color.Gray), 10);
```

```
        g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
```

```
        g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
```

```
        g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);
```

```
        g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
```

```
        g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
```

```
        g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);
```

```
    }
```



```

public void DrawOX(Form f)
{
    Graphics g = f.CreateGraphics();

    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            if (click_array[i, j] == 1)
            {
                g.DrawEllipse(
                    new Pen(new SolidBrush(Color.Red), 7),
                    i * 100 + 50,
                    j * 100 + 50,
                    50, 50);
            }
            else if (click_array[i, j] == 2)
            {
                g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i
* 100 + 50, j * 100 + 50,
i * 100 + 100, j * 100 + 100);
                g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i
* 100 + 100, j * 100 + 50,
                    i * 100 + 50, j * 100 + 100);
            }
        }
    }
}

```

```
}
```

```
int[,] click_array = new int[3, 3];
```

```
public void ClickAndSetArray(Form f)
```

```
{
```

```
    Graphics g = f.CreateGraphics();
```

```
    if (e.Button == MouseButtons.Left)
```

```
    {
```

```
        for (int i = 0; i < 3; i++)
```

```
        {
```

```
            for (int j = 0; j < 3; j++)
```

```
            {
```

```
                click_array[i, j] = 1; //0
```

```
            }
```

```
        }
```

```
    }
```

```
    else
```

```
    {
```

```
        for (int i = 0; i < 3; i++)
```

```
        {
```

```
            for (int j = 0; j < 3; j++)
```

```
            {
```

```
                click_array[i, j] = 2; //0
```

```
            }
```

```
        }
```

```

        }
    }
}

public partial class MainForm : Form
{
    TicTacToeGame game = new TicTacToeGame();

    public MainForm()
    {
        InitializeComponent();

        private void MainForm_Paint(object sender, PaintEventArgs e)
        {
            game.DrawGrid(this);
            game.DrawOX(this);
        }

        private void MainForm_MouseDown(object sender, MouseEventArgs e)
        {
            game.ClickAndSetArray(this);

            Invalidate();
        }
    }
}

```

마지막으로, ClickAndSetArray 멤버 함수를 보면 마우스 오른쪽 버튼을 눌렀는지 왼쪽 버튼을 눌렀는지 체크하기 위해서 파라미터 e를 받아야 하네요. 이 파라미터 또한 함수 호출할 때 넘겨주고 멤버 함수에서 이를 사용하도록 아래와 같이 코드를 수정합니다.

```
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;

public class TicTacToeGame
{
    public void DrawGrid(Form f)
    {
        Graphics g = f.CreateGraphics();

        Pen p = new Pen(new SolidBrush(Color.Gray), 10);

        g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
        g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
        g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);
```

```

        g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
        g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
        g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);
    }

    public void DrawOX(Form f)
    {
        Graphics g = f.CreateGraphics();

        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                if (click_array[i, j] == 1)
                {
                    g.DrawEllipse(
                        new Pen(new SolidBrush(Color.Red), 7),
                        i * 100 + 50,
                        j * 100 + 50,
                        50, 50);
                }
                else if (click_array[i, j] == 2)
                {
                    g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i
* 100 + 50, j * 100 + 50, i * 100 + 100, j * 100 + 100);
                    g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i
* 100 + 100, j * 100 + 50, i * 100 + 50, j * 100 + 100);
                }
            }
        }
    }
}

```

```

        }
    }
}

```

```
int[,] click_array = new int[3, 3];
```

```
public void ClickAndSetArray(Form f, MouseEventArgs e)
{

```

```
    Graphics g = f.CreateGraphics();

```

```
    if (e.Button == MouseButton.Left)
    {

```

```
        for (int i = 0; i < 3; i++)
        {

```

```
            for (int j = 0; j < 3; j++)
            {

```

```
                click_array[i, j] = 1; //0
            }
        }
    }

```

```
    else
    {

```

```
        for (int i = 0; i < 3; i++)
        {

```

```
            for (int j = 0; j < 3; j++)
            {

```

```

        click_array[i, j] = 2; //O
    }
}
}
}
}
}

```

```

public partial class MainForm : Form
{
    TicTacToeGame game = new TicTacToeGame();

    public MainForm()
    {
        InitializeComponent();
    }

    private void MainForm_Paint(object sender, PaintEventArgs e)
    {
        game.DrawGrid(this);
        game.DrawOX(this);
    }

    private void MainForm_MouseDown(object sender, MouseEventArgs e)
    {
        game.ClickAndSetArray(this, e);

        Invalidate();
    }
}

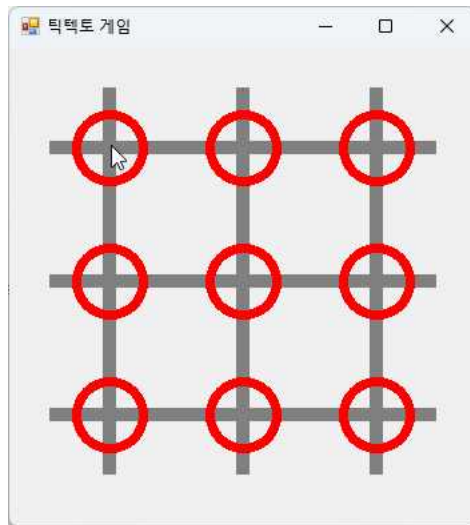
```

```

    }
}

```

이제 드디어 오류가 하나도 발생하지 않습니다. 이 코드를 컴파일하고 실행하면 이전처럼 잘 실행됨을 알 수 있습니다. 왼쪽 버튼을 누르면 O 표시가, 오른쪽 버튼을 누르면 X 표시가 폼 윈도우에 잘 그려지고, 또한 윈도우 일부를 가렸다가 다시 나타나게 하여도 폼 윈도우에 있는 내용이 지워지지 않고 표시됨을 알 수 있습니다



드디어 TicTacToeGame 클래스가 완성되었습니다. 이제 이 클래스만 보고 관하고 있으면 여러분은 아래와 같이 간단히 코드를 작성하면 금방 TicTacToe 게임을 구현할 수 있게 됩니다.

```
public partial class MainForm : Form
```



```

{
    TicTacToeGame game = new TicTacToeGame();

    public MainForm()
    {
        InitializeComponent();
    }

    private void MainForm_Paint(object sender, PaintEventArgs e)
    {
        game.DrawGrid(this);
        game.DrawOX(this);
    }

    private void MainForm_MouseDown(object sender, MouseEventArgs e)
    {
        game.ClickAndSetArray(this, e);

        Invalidate();
    }
}

```

이제까지 틱택토 게임 코드를 작성해 보았습니다. 이렇게 코드 추상화하여 함수를 만들고, 함수들과 관련된 배열 혹은 변수를 묶어 데이터 추상화하여 클래스를 만들고, 결과적으로 이러한 코드는 클래스 라이브러리가 됩니다. 이를 보관하고 있다가 나중에 6개월 혹은 1년이 지나서도 간단하게, 손쉽게 재사용하여 코딩할 수 있으면 여러분이 새로운 프로젝트 개발 시 매우 큰 도움이 됩니다. 이렇듯 프로그래밍도 사전에 어떤 내용, 클래스를 미리 작성하여 준비해 두었다가 나중에 필요할 때 사용하도록 함으로써 코딩 효율을 매우 높일 수 있습니다. 프로그래밍을 잘한다는 것은 바로 이런 것을 의미합니다.

물론 아직 게임 프로그램이 다 완성된 것은 아닙니다. 마우스로 특정 격자 위치를 클릭하면 그곳과 매칭된 `click_array` 배열 인덱스가 나와야 합니다. 가령 (0, 0), (1, 0), ..., (2, 2) 등과 같은 배열 인덱스가 반환되어서 해당 배열 인덱스 위치만 1 혹은 2 값을 갖도록 값을 할당해야 합니다.

마우스로 격자 위치를 클릭할 때 X Y 좌표가 어떻게 나오는지 확인하기 위해서 아래와 같이 코드를 작성합니다.

```
public partial class MainForm : Form
{
    TicTacToeGame game = new TicTacToeGame();

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

```

private void MainForm_Paint(object sender, PaintEventArgs e)
{
    game.DrawGrid(this);
    game.DrawOX(this);
}

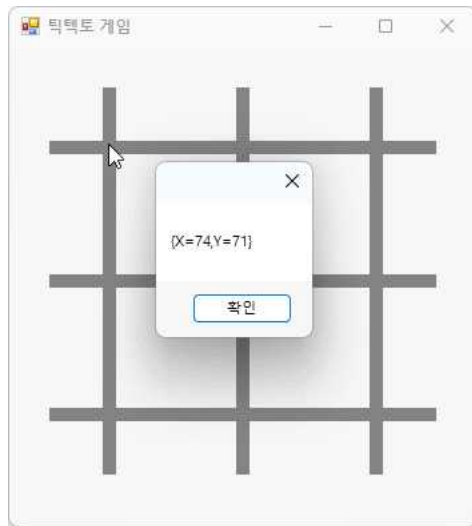
private void MainForm_MouseDown(object sender, MouseEventArgs e)
{
    String tmp = e.Location.ToString();
    MessageBox.Show(tmp);

    game.ClickAndSetArray(this, e);

    Invalidate();
}
}

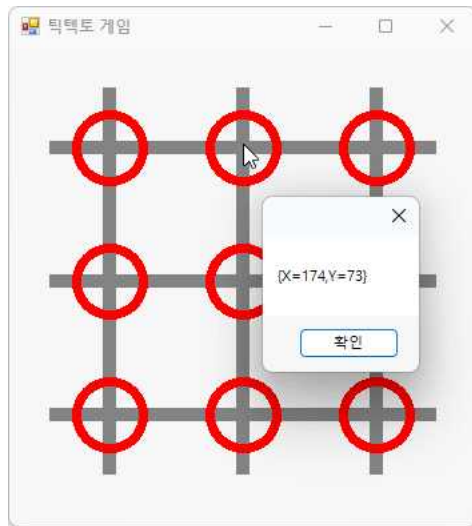
```

아래와 같이 첫 번째 격차 위치를 대략 클릭하면 아래와 같이 X, Y 좌표가 표시됩니다.



이를 통해서 첫 번째 격차는, 픽셀 좌표로 볼 때 $X=70$, $Y=70$ 으로 예상할 수 있습니다.

두 번째 격자 위치에서 마우스를 클릭해 보면 아래와 같이 대략 표시 됩니다.



이러한 사실을 통해서 X 좌표와 Y 좌표 첫 위치는 70 픽셀 위치이고 100 픽셀 씩 떨어진 위치에 격자 존재함을 예상할 수 있겠네요. 격자 위치가 대략 다음과 같다고 할 수 있겠습니다.

(70, 70), (170, 70), (270, 70)
 (70, 170), (170, 170), (270, 170)
 (70, 270), (270, 170), (270, 270)

위의 위치에 따라서 아래와 같은 좌표를 반환하도록 하면 됩니다.

(0, 0), (1, 0), (2, 0)
 (0, 1), (1, 1), (2, 1)
 (0, 2), (1, 2), (2, 2)

이제 격자 위치 정보를 배열 인덱스 정보로 바꾸는 방법을 찾아 보도록 하겠습니다. 앞서 설명한 것처럼 9개 각 격자의 위치는 다음과 같습니다.

(70, 70), (170, 70), (270, 70)
(70, 170), (170, 170), (170, 270)
(70, 270), (270, 170), (270, 270)

X Y 좌표를 70 씩 빼면 어떻게 될까요 그 결과는 아래와 같습니다

(0, 0), (100, 0), (200, 0)
(0, 100), (100, 100), (100, 200)
(0, 200), (200, 100), (200, 200)

이제 여기에서 X, Y 좌표 각각을 100으로 나누었을 때 몫은 얼마가 될까요? 다음과 같습니다.

(0, 0), (1, 0), (2, 0)
(0, 1), (1, 1), (1, 2)
(0, 2), (2, 1), (2, 2)

여러분들이 마우스 커서로 (0, 100), (100, 200) 등 정확한 격자 위치를 클릭했을 경우에는 위와 같이 좌표를 얻을 수 있습니다. 하지만 정확한 위치를 클릭하지 못하고 근처를 클릭할 경우에는 어떻게 될까요?

가령, (67, 75), (163, 277) 등과 같은 위치를 클릭할 수도 있습니다. 이 좌표에 위와 같은 규칙을 적용해보면 아래와 같습니다.

$((67-70) / 100, (75-70)/100), ((163-70)/100), ((277-70)/100)$

따라서 결과는 다음과 같습니다.

$(-3/100, 5/100), (93/100, 207/100)$

따라서, 아래와 같이 최종 값을 얻을 수 있습니다.

$(-0.03, 0.05), (0.93, 2.07)$

이때 위 각 좌표값에 대해 반올림하면 어떻게 될까요? 결과는 다음과 같습니다.

$(0, 0), (1, 2)$

클릭한 마우스 좌표 에 대하여 다음과 같이 세가지를 적용 하면 그의 해당하는 배열 인덱스 값을 얻을 수 있습니다.

1. 클릭한 좌표 (X, Y)에 대해 70을 뺀다.
2. 100으로 나눈다.
3. 그 결과를 반올림한다.
4. 그렇게 하여 구한 배열 인덱스 위치에 1 혹은 2 값을 할당한다.

이를 구현한 코드는 다음과 같습니다.

```
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;

public class TicTacToeGame
{
    public void DrawGrid(Form f)
    {
        Graphics g = f.CreateGraphics();

        Pen p = new Pen(new SolidBrush(Color.Gray), 10);

        g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
        g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
        g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);

        g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
        g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
        g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);
    }

    public void DrawOX(Form f)
    {

```



```

Graphics g = f.CreateGraphics();

for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        if (click_array[i, j] == 1)
        {
            g.DrawEllipse(
                new Pen(new SolidBrush(Color.Red), 7),
                i * 100 + 50,
                j * 100 + 50,
                50, 50);
        }
        else if (click_array[i, j] == 2)
        {
            g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i
* 100 + 50, j * 100 + 50,
i * 100 + 100, j * 100 + 100);
            g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i
* 100 + 100, j * 100 + 50,
                i * 100 + 50, j * 100 + 100);
        }
    }
}
}

```

```
int[,] click_array = new int[3, 3];
```

```
public void ClickAndSetArray(Form f, MouseEventArgs e)
```

```
{
```

```
    Graphics g = f.CreateGraphics();
```

```
    if (e.Button == MouseButtons.Left)
```

```
    {
```

```
        int i = (int)Math.Round((e.X - 70) / 100.0);
```

```
        int j = (int)Math.Round((e.Y - 70) / 100.0);
```

```
        click_array[i, j] = 1; //O
```

```
    }
```

```
    else
```

```
    {
```

```
        int i = (int)Math.Round((e.X - 70) / 100.0);
```

```
        int j = (int)Math.Round((e.Y - 70) / 100.0);
```

```
        click_array[i, j] = 2; //X
```

```
    }
```

```
}
```

```
}
```

```
public partial class MainForm : Form
```

```
{
```

```
    TicTacToeGame game = new TicTacToeGame();
```

```

public MainForm()
{
    InitializeComponent();
}

private void MainForm_Paint(object sender, PaintEventArgs e)
{
    game.DrawGrid(this);
    game.DrawOX(this);
}

private void MainForm_MouseDown(object sender, MouseEventArgs e)
{
    game.ClickAndSetArray(this, e);

    Invalidate();
}
}

```

이로써 마우스로 클릭한 위치에 대한 배열 index 값을 알아내고, 그 index 값을 이용하여 왼쪽 버튼을 누르면 1을, 오른쪽 버튼을 누르면 2 값을 할당합니다. 마우스를 클릭한 경우 이렇게 값만 할당하고, 나중에 Paint 이벤트 핸들러 함수에서 배열 전체 값을 확인하여 O 혹은 X 표시를 하게 됩니다.

사실 아직 모든 코드를 구현한 것은 아닙니다. 예를 들어서 이미 돌을 둔 곳은 더 이상 두지 못하도록 하는 코드도 작성해야 합니다.

이는 아래와 같이 배열 인덱스를 구한 후 그 위치의 값이 1 혹은 2가 설정되지 않았을 경우에만 합을 할당하도록 하면 됩니다.

```
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;

public class TicTacToeGame
{
    public void DrawGrid(Form f)
    {
        Graphics g = f.CreateGraphics();

        Pen p = new Pen(new SolidBrush(Color.Gray), 10);

        g.DrawLine(p, 50 + 25, 30, 50 + 25, 320);
        g.DrawLine(p, 150 + 25, 30, 150 + 25, 320);
        g.DrawLine(p, 250 + 25, 30, 250 + 25, 320);

        g.DrawLine(p, 30, 50 + 25, 320, 50 + 25);
        g.DrawLine(p, 30, 150 + 25, 320, 150 + 25);
```

```

        g.DrawLine(p, 30, 250 + 25, 320, 250 + 25);
    }

    public void DrawOX(Form f)
    {
        Graphics g = f.CreateGraphics();

        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                if (click_array[i, j] == 1)
                {
                    g.DrawEllipse(
                        new Pen(new SolidBrush(Color.Red), 7),
                        i * 100 + 50,
                        j * 100 + 50,
                        50, 50);
                }
                else if (click_array[i, j] == 2)
                {
                    g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i
* 100 + 50, j * 100 + 50,
i * 100 + 100, j * 100 + 100);
                    g.DrawLine(new Pen(new SolidBrush(Color.Blue), 7), i
* 100 + 100, j * 100 + 50,
i * 100 + 50, j * 100 + 100);
                }
            }
        }
    }
}

```

```

        }
    }
}

```

```

int[,] click_array = new int[3, 3];

```

```

public void ClickAndSetArray(Form f, MouseEventArgs e)
{

```

```

    Graphics g = f.CreateGraphics();

```

```

    if (e.Button == MouseButtons.Left)
    {

```

```

        int i = (int)Math.Round((e.X - 70) / 100.0);

```

```

        int j = (int)Math.Round((e.Y - 70) / 100.0);

```

```

        if (!(click_array[i, j] == 1 | click_array[i, j] == 2))

```

```

            click_array[i, j] = 1; //O

```

```

        }

```

```

    else

```

```

    {

```

```

        int i = (int)Math.Round((e.X - 70) / 100.0);

```

```

        int j = (int)Math.Round((e.Y - 70) / 100.0);

```

```

        if (!(click_array[i, j] == 1 | click_array[i, j] == 2))

```

```

            click_array[i, j] = 2; //X

```

```

    }

```

```

    }
}

public partial class MainForm : Form
{
    TicTacToeGame game = new TicTacToeGame();

    public MainForm()
    {
        InitializeComponent();
    }

    private void MainForm_Paint(object sender, PaintEventArgs e)
    {
        game.DrawGrid(this);
        game.DrawOX(this);
    }

    private void MainForm_MouseDown(object sender, MouseEventArgs e)
    {
        game.ClickAndSetArray(this, e);

        Invalidate();
    }
}

```

돌을 두었을 때 매번 게임이 끝났는지도 확인해야 합니다. 틱택토 혹은 오목 게임에서 같은 색의 돌이 연이어 있을 경우 게임을 이긴 것으로 판정합니다. `click_array` 배열을 이용하여 이를 확인하는 코드를 작성하면 됩니다. 이 코드는 여러분에게 숙제로 남기도록 하겠습니다.