

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УНІВЕРСИТЕТ МИТНОЇ СПРАВИ ТА ФІНАНСІВ
ФАКУЛЬТЕТ ІННОВАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра комп'ютерних наук та інженерії програмного забезпечення

Task3

З дисципліни «ASP.NET»

Виконав:

Студент групи К21-2

Кугай Владислав

Дніпро

2025

TASK №3

Завдання: Переробіть гру з другої роботи з використанням ASP.NET Forms, та розташуйте її на безкоштовному хостінгу.

Виконання: У цьому проєкті використовується мова програмування C# разом із технологією Windows Forms на платформі .NET Framework. Це класичний підхід до створення настільних застосунків із графічним інтерфейсом. Основна форма програми — Form1 — автоматично генерується під час запуску і має розміри 320×340 пікселів. На цій формі динамічно створюються 9 кнопок, які формують ігрове поле розміром 3×3. Користувач робить хід, натискаючи на будь-яку вільну кнопку, після чого система автоматично передає хід комп'ютеру. Комп'ютер обирає випадкову клітинку для розміщення символу "О", використовуючи вбудований механізм генерації випадкових чисел. Уся логіка гри реалізована через окремі методи: створення кнопок, обробка події натискання, перевірка на перемогу або нічию, а також виконання ходу комп'ютера. Після кожного ходу перевіряється, чи є переможець або чи заповнене поле повністю. Якщо гра завершується, виводиться відповідне повідомлення через вікно MessageBox, а всі кнопки блокуються, щоб запобігти подальшим діям. Інтерфейс користувача залишається простим і зрозумілим, з великими кнопками, розташованими рівномірно в сітці, що робить гру зручною для користувача.

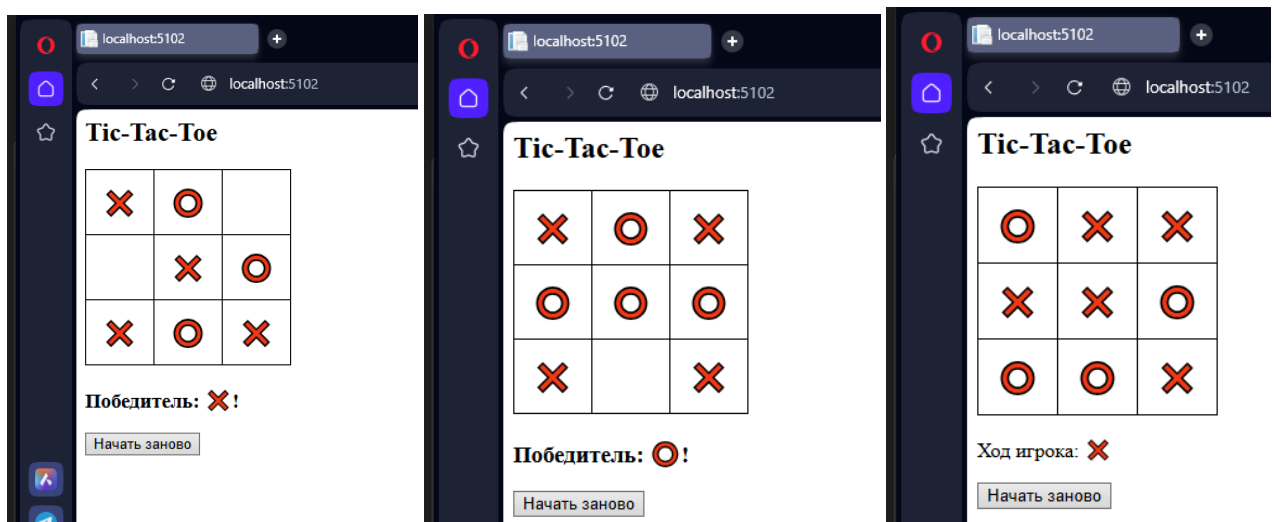


Рисунок 1. Робота додатку «Хрестики-нулики»


```

var builder = WebApplication.CreateBuilder(args);

// Додаємо підтримку MVC (контролери + представлення)
builder.Services.AddControllersWithViews();

var app = builder.Build();

// Увімкнення обробки статичних файлів (якщо є CSS/JS)
app.UseStaticFiles();

// Налаштування маршрутизації
app.UseRouting();

// (Не обов'язково, але якщо буде автентифікація/авторизація)
app.UseAuthorization();

// Додаємо маршрут за замовчуванням:
// http://localhost:5102 -> HomeController -> Index()
app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();

```

GameState.cs

```

using System;
using System.Text;

namespace Task3.Models
{
    public class GameState
    {
        private const string PlayerSymbol = "X";
        private const string ComputerSymbol = "O";

        public string[,] Board { get; set; } = new string[3, 3];
        public string CurrentPlayer { get; set; } = PlayerSymbol;
        public string Winner { get; set; }

        public GameState()
        {
            for (int i = 0; i < 3; i++)
                for (int j = 0; j < 3; j++)
                    Board[i, j] = null;
        }

        public void MakeMove(int row, int col)
        {
            if (Board[row, col] == null && Winner == null)
            {
                Board[row, col] = CurrentPlayer;

                if (CheckWinner())
                {
                    Winner = CurrentPlayer;
                    return;
                }

                if (IsBoardFull())
                    return;
            }
        }
    }
}

```

```

erSymbol;

        CurrentPlayer = CurrentPlayer == PlayerSymbol ? ComputerSymbol : Play-
erSymbol;

        if (CurrentPlayer == ComputerSymbol)
            ComputerMove();
    }
}

private void ComputerMove()
{
    Random rand = new Random();
    var emptyCells = new List<(int, int)>();

    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            if (Board[i, j] == null)
                emptyCells.Add((i, j));

    if (emptyCells.Count > 0)
    {
        var (row, col) = emptyCells[rand.Next(emptyCells.Count)];
        Board[row, col] = ComputerSymbol;

        if (CheckWinner())
        {
            Winner = ComputerSymbol;
            return;
        }

        if (!IsBoardFull())
            CurrentPlayer = PlayerSymbol;
    }
}

private bool CheckWinner()
{
    return CheckLines(Board[0, 0], Board[0, 1], Board[0, 2]) ||
           CheckLines(Board[1, 0], Board[1, 1], Board[1, 2]) ||
           CheckLines(Board[2, 0], Board[2, 1], Board[2, 2]) ||
           CheckLines(Board[0, 0], Board[1, 0], Board[2, 0]) ||
           CheckLines(Board[0, 1], Board[1, 1], Board[2, 1]) ||
           CheckLines(Board[0, 2], Board[1, 2], Board[2, 2]) ||
           CheckLines(Board[0, 0], Board[1, 1], Board[2, 2]) ||
           CheckLines(Board[0, 2], Board[1, 1], Board[2, 0]);
}

private bool CheckLines(string a, string b, string c)
{
    return a != null && a == b && b == c;
}

private bool IsBoardFull()
{
    foreach (var cell in Board)
        if (cell == null) return false;
    return true;
}

public string GetBoardAsString()
{
    var sb = new StringBuilder();
    for (int i = 0; i < 3; i++)
    {
        sb.AppendLine(string.Join(" | ",
            Board[i, 0] ?? "□",

```

```

        Board[i, 1] ?? "□",
        Board[i, 2] ?? "□"));
        if (i < 2) sb.AppendLine("-----");
    }
    return sb.ToString();
}
}
}

```

HomeController.cs

```

using Microsoft.AspNetCore.Mvc;
using Task3.Models;

namespace Task3.Controllers
{
    public class HomeController : Controller
    {
        private static GameState game = new GameState();

        public IActionResult Index()
        {
            return View(game);
        }

        public IActionResult Move(int row, int col)
        {
            game.MakeMove(row, col);
            return RedirectToAction("Index");
        }

        public IActionResult Reset()
        {
            game = new GameState();
            return RedirectToAction("Index");
        }
    }
}

```

Index.cshtml

```

@model Task3.Models.GameState

<h2>Tic-Tac-Toe</h2>

<table style="border-collapse: collapse;">
    @for (int row = 0; row < 3; row++)
    {
        <tr>
            @for (int col = 0; col < 3; col++)
            {
                <td style="width: 60px; height: 60px; text-align: center; border: 1px
solid black;">
                    @if (Model.Board[row, col] == null && Model.Winner == null)
                    {
                        <a href="/Home/Move?row=@row&col=@col" style="display: block;
width: 100%; height: 100%; text-decoration: none;"></a>
                    }
                    else
                    {
                        <span style="font-size: 24px;">@Model.Board[row, col]</span>
                    }
                </td>
            }
        }
    }

```

```

        </tr>
    }
</table>

@if (Model.Winner != null)
{
    <h3>Победитель: @Model.Winner!</h3>
}
else
{
    <p>Ход игрока: <strong>@Model.CurrentPlayer</strong></p>
}

<form asp-controller="Home" asp-action="Reset" method="post">
    <button type="submit">Начать заново</button>
</form>

```

Висновок: Створено гру "Хрестики-Нулики" з графічним інтерфейсом на Windows Forms. Реалізовано динамічне створення елементів керування, обробку подій і базову ігрову логіку. Отримано практичні навички роботи з формами, кнопками та обробниками подій у C#.

Посилання: https://github.com/NoGeX7/Task_3