**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**



**АВТОМАТИЗОВАНЕ ПРОЕКТУВАННЯ КОМП’ЮТЕРНИХ СИСТЕМ**

Лабораторна робота № 5

Виконав: ст. гр. КІ-401

Мандат Л.Т.

Прийняв: Федак П.Р

Львів 2024

**Тема:** ознайомлення з автоматичним тестуванням.

**Завдання для виконання лабораторної роботи:**

1. Implement or use existing test framework;

2. Create a set of automated tests;

3. Test report should contain number of all tests, passed tests, failed tests, coverage;

4. Coverage must be more than 80;

**Хід виконання роботи:**

Для тестування програми написаної на мові c#, я обрала фреймворк xUnit.

Тестування проводиться лише клієнтської частини, апаратне забезпечення тестується окремо.

Код тестового файлу UnitTest1.cs :

using Xunit;

namespace TestingSystemTicTacToe

{

public class TicTacToeTests

{

private char[,] \_board;

private char \_currentPlayer;

public TicTacToeTests()

{

\_board = new char[3, 3]

{

{ ' ', ' ', ' ' },

{ ' ', ' ', ' ' },

{ ' ', ' ', ' ' }

};

\_currentPlayer = 'X';

}

private bool MakeMove(int row, int col)

{

if (\_board[row, col] != ' ') return false;

\_board[row, col] = \_currentPlayer;

\_currentPlayer = \_currentPlayer == 'X' ? 'O' : 'X';

return true;

}

private bool CheckWin()

{

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

{

if (\_board[i, 0] == \_board[i, 1] && \_board[i, 1] == \_board[i, 2] && \_board[i, 0] != ' ')

return true;

if (\_board[0, i] == \_board[1, i] && \_board[1, i] == \_board[2, i] && \_board[0, i] != ' ')

return true;

}

if (\_board[0, 0] == \_board[1, 1] && \_board[1, 1] == \_board[2, 2] && \_board[0, 0] != ' ')

return true;

if (\_board[0, 2] == \_board[1, 1] && \_board[1, 1] == \_board[2, 0] && \_board[0, 2] != ' ')

return true;

return false;

}

[Fact]

public void PlayerXStartsTheGame()

{

Assert.Equal('X', \_currentPlayer);

}

[Fact]

public void CheckWinReturnsFalseForEmptyBoard()

{

Assert.False(CheckWin());

}

[Fact]

public void PlayerCanMakeMove()

{

var result = MakeMove(0, 0);

Assert.True(result);

Assert.Equal('X', \_board[0, 0]);

}

[Fact]

public void PlayerCannotMoveToOccupiedCell()

{

MakeMove(0, 0);

var result = MakeMove(0, 0);

Assert.False(result);

}

[Fact]

public void ActivePlayerSwitchesAfterMove()

{

MakeMove(0, 0);

Assert.Equal('O', \_currentPlayer);

}

[Fact]

public void PlayerXWinsHorizontally()

{

MakeMove(0, 0); // X

MakeMove(1, 0); // O

MakeMove(0, 1); // X

MakeMove(1, 1); // O

MakeMove(0, 2); // X

Assert.True(CheckWin());

}

[Fact]

public void PlayerOWinsVertically()

{

MakeMove(0, 1); // X

MakeMove(0, 0); // O

MakeMove(1, 1); // X

MakeMove(1, 0); // O

MakeMove(2, 2); // X

MakeMove(2, 0); // O

Assert.True(CheckWin());

}

[Fact]

public void PlayerXWinsDiagonally()

{

MakeMove(0, 0); // X

MakeMove(1, 0); // O

MakeMove(1, 1); // X

MakeMove(2, 0); // O

MakeMove(2, 2); // X

Assert.True(CheckWin());

}

[Fact]

public void GameEndsInDraw()

{

MakeMove(0, 0); // X

MakeMove(0, 1); // O

MakeMove(0, 2); // X

MakeMove(1, 1); // O

MakeMove(1, 0); // X

MakeMove(1, 2); // O

MakeMove(2, 1); // X

MakeMove(2, 0); // O

MakeMove(2, 2); // X

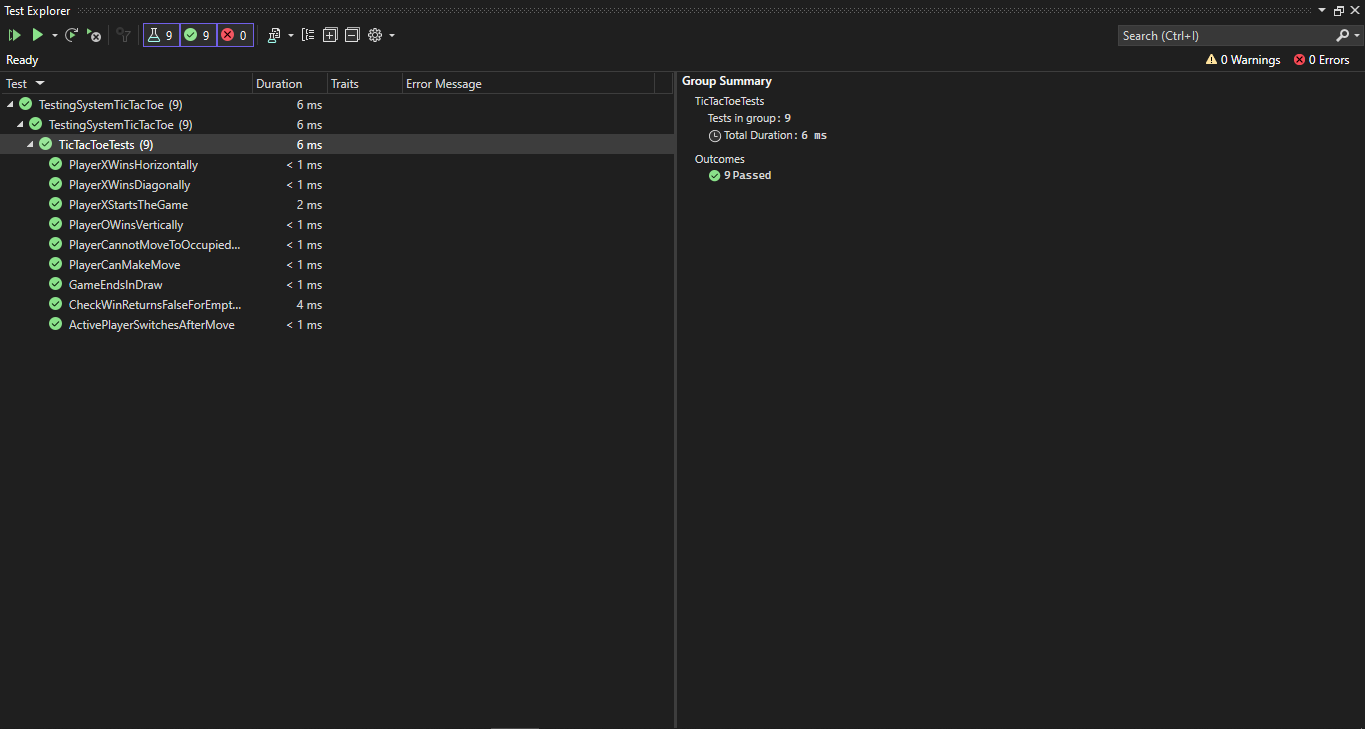
Assert.False(CheckWin());

}

}

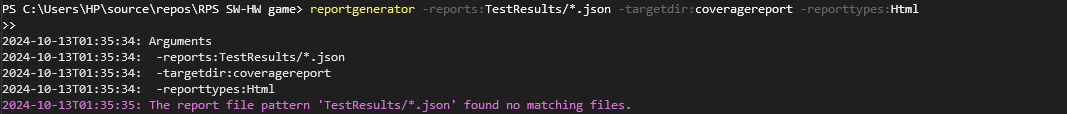
}

Результати виконання тестів:



Як бачимо всі тести пройшли успішно!

Далі встановила і виконала утиліту Coverlet для формування звіту про тестування :



**Висновок:**

У цій лабораторній роботі було досліджено процес розробки тестування для методів проєкту, що включає гру "Камінь, Ножиці, Папір". Основна увага була зосереджена на створенні автоматизованих тестів з використанням фреймворку xUnit, що забезпечило ефективну перевірку функціональності програми.