Marwadi University
Marwadi Chandarana Group

NAAC A+

FACULTY OF COMPUTER APPLICATIONS

A c# Project for Academic Year 2025 - 2026

Tic Tac Toe Project

Submitted in partial fulfillment of the requirement for the award of the degree

Bachelor of Application (BCA)

SubmittedBy : Vishal Parekh (92300527233)
: Neel Gardhariya (92300527211)

SubmittedTo:
Dr. Jignesh Hirapara

DECLARATION

I/We hereby declare that this project work entitled **Tic Tac Toe Project** is a record done by me.

I also declare that the matter embodied in this project is genuine work done by me and has not been submitted whether to this University or to any other University / Institute for the fulfillment of the requirement of any course of study.

Place :University

Date : 15-04-2025

Student Name-1: (Enrollment No) **Signature:** _____
Student Name-2: (Enrollment no) **Signature**: _____

FLOW CHART OF CODE:

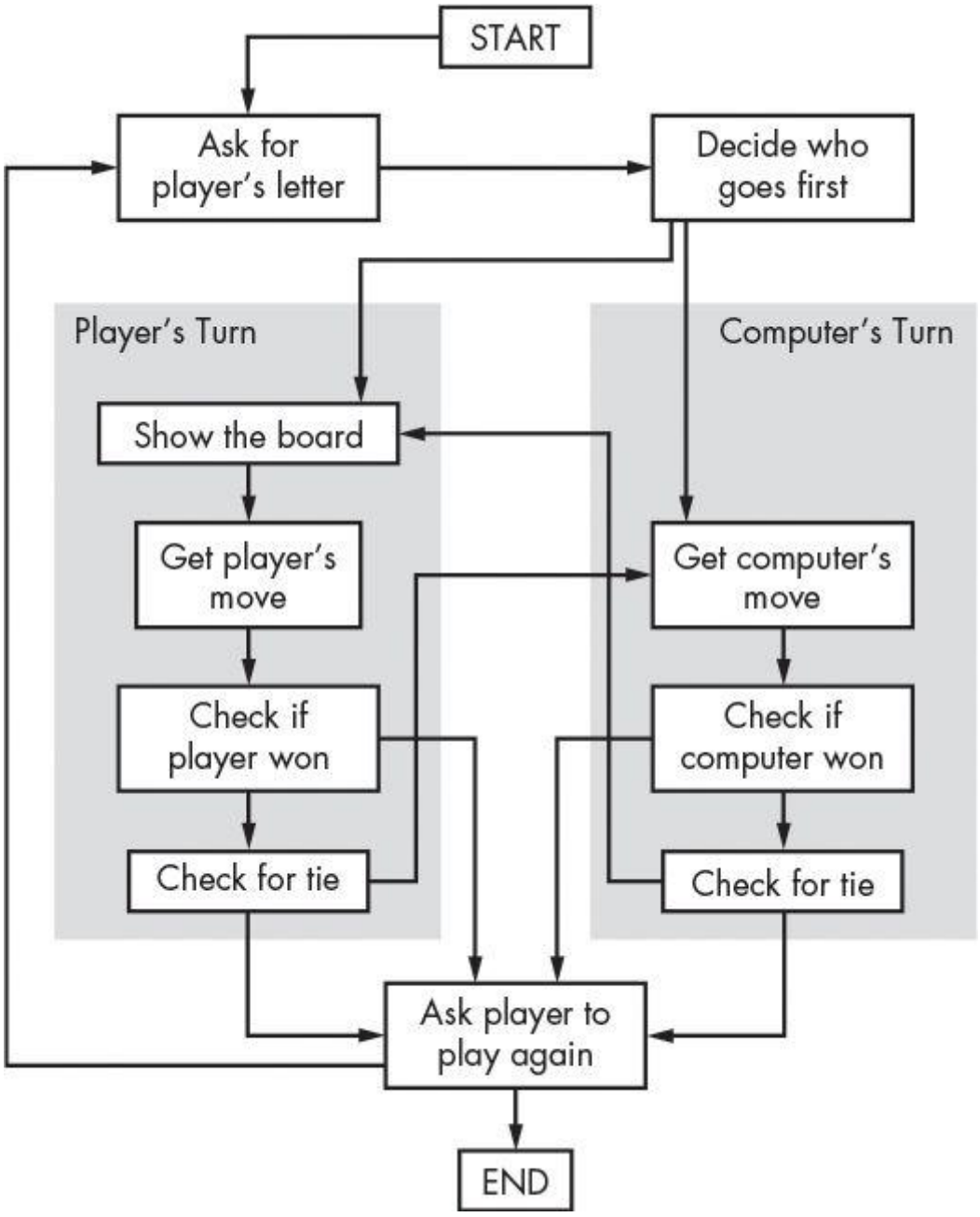


Table Of content:

Introduction to Project Defination :

The Tic Tac Toe Game is a simple console based application designed to allow two players to play the classic game of Tic Tac Toe. The program is developed in C# using a modular structure and stores minimal data during execution. The system provides an interactive text-based interface for users to play the game, manage turns, and determine the winner.

PREAMBLE

Module Description :

The system is divided into the following modules:

- Game Board Module: Manages the 3x3 grid.
- Player Module: Handles player inputs and symbols (X or O).
- Game Logic Module: Checks win conditions and draw scenarios.
- Display Module: Renders the board and messages in the console.
- Replay Module: Allows players to restart the game after a match.

Study Of Existing System :

Manual versions of the Tic Tac Toe game (on paper) are easy but lack features such as error checks and replay options. Existing digital versions may include ads or be too complex for beginners trying to learn programming.

Inefficiency:

- Manual play does not prevent illegal moves or check for win conditions automatically.

- Human errors can affect the fairness or flow of the game.

Higher Probability of Human Error:

- Mistakes in tracking turns or marking the board can lead to confusion.

No Replay or Undo Options:

- Manual gameplay lacks features to easily

Restart or undo moves :

- Limited Engagement:

  Without a visual or interactive experience, manual play may feel static.

TECHNICAL DESCRIPTION

Hardware Requirements:

- Processor: Intel Core i3 or higher
- RAM: 2 GB or more
- Storage: Minimal (for code files)
- Input: Keyboard
- Display: Console output

Software Requirements:

- Windows OS
- .NET Framework
- Visual Studio / Visual Studio Code

System Design And Development :

Diagrams as Applicable

Algorithm:

1. Initialize game board

2. Display empty board

3. Repeat until game ends:
    - Accept player input
    - Update board
    - Check for win/draw
    - Switch player

4. Display result and ask for replay

Database Design / File Structure:

No external files or database required for this project.
All data is managed in runtime memory using arrays and variables.

Menu Design

A basic numbered menu allows the user to:

1. Start Game

2. View Instructions

3. Exit

Screen Design :

Console output showing board and options:

```
=== TIC TAC TOE GAME ===
 1 | 2 | 3
 ---------
 4 | 5 | 6
 ---------
 7 | 8 | 9

PLAYER X, CHOOSE YOUR MOVE:
```

Code of the Module

Main class: Program

- Sub-methods: StartGame(), DisplayBoard(), PlayerMove(), CheckWinner(), ResetGame()
- Class: TicTacToeGame (contains board and logic)

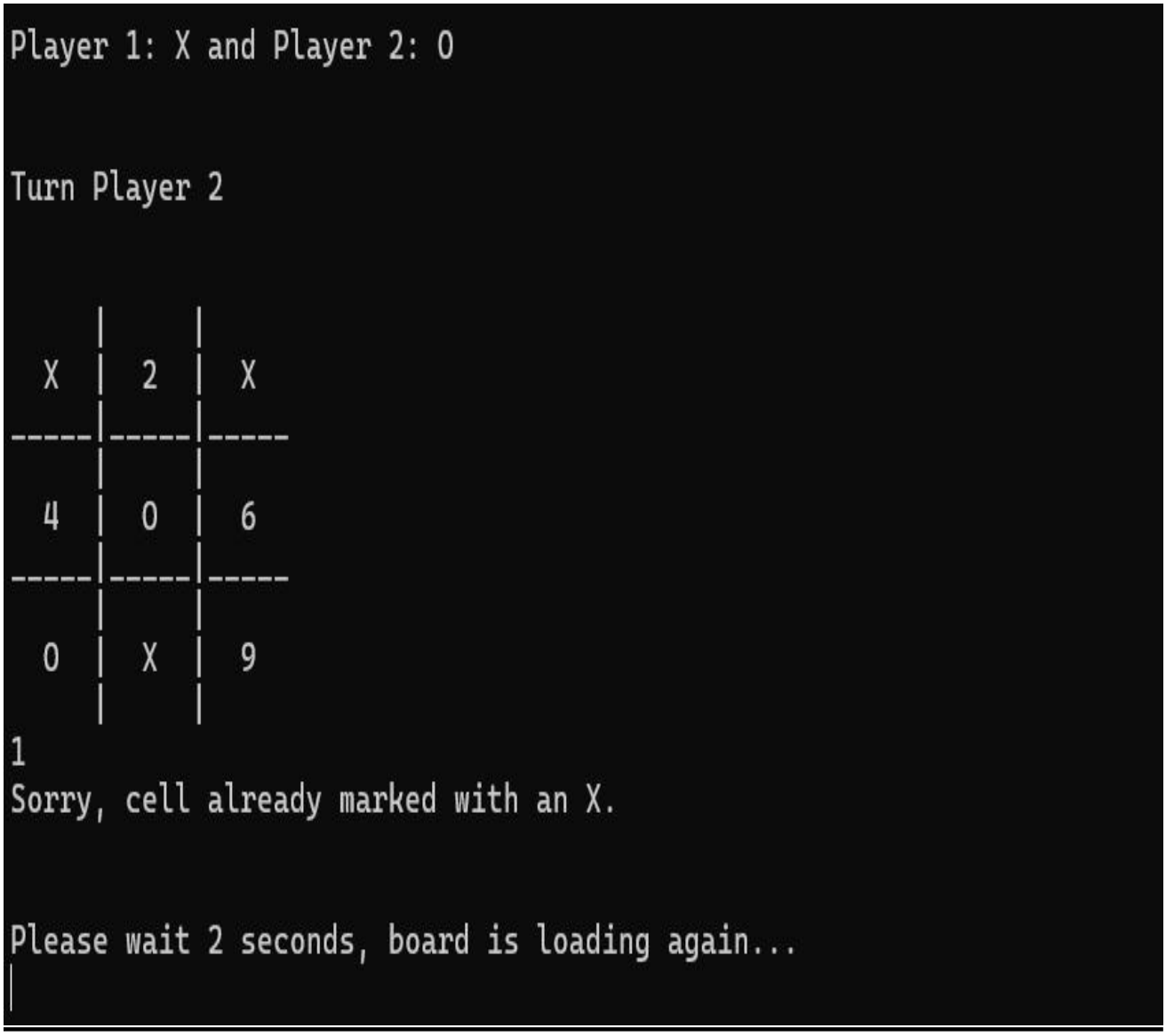
```
using System;

0 references
class Program
{
    static char[] arr = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' };
    static int player = 1;
    static int choice;
    static int flag = 0;

    0 references
    static void Main(string[] args)
    {
        Console.WriteLine("Tic Tac Toe Game!");
        Console.WriteLine("----------------");

        do
        {
            Console.Clear();
            Console.WriteLine("Player 1: X and Player 2: 0");
            Console.WriteLine("\n");
            if (player % 2 == 0)
            {
                Console.WriteLine("Turn Player 2");
            }
            else
            {
                Console.WriteLine("Turn Player 1");
            }
```

Tic Tac Toe

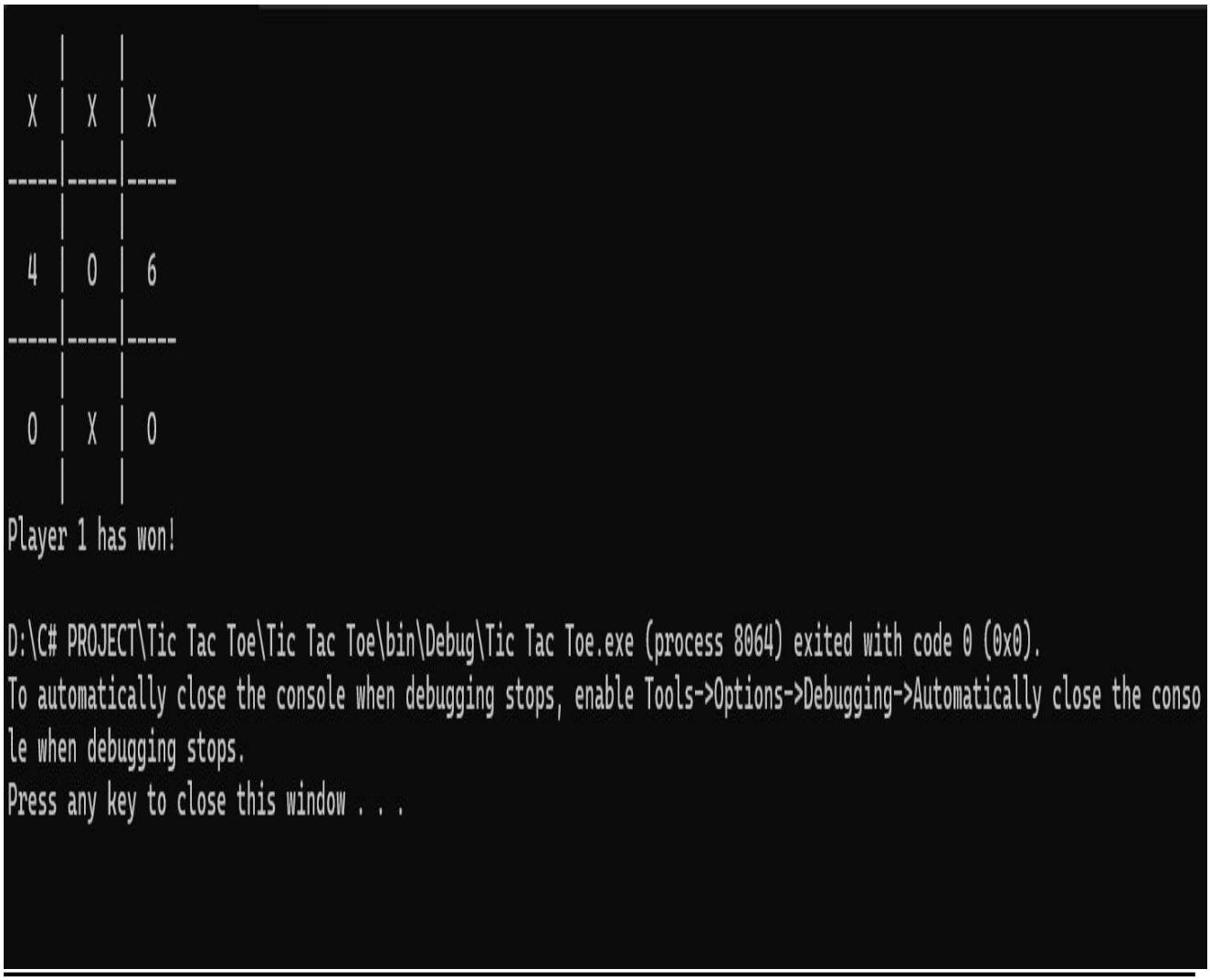
```
2 references
private static void Board()
{
    Console.WriteLine("     |     |     ");
    Console.WriteLine("  {0}  |  {1}  |  {2}  ", arr[1], arr[2], arr[3]);
    Console.WriteLine("_____|_____|_____");
    Console.WriteLine("     |     |     ");
    Console.WriteLine("  {0}  |  {1}  |  {2}  ", arr[4], arr[5], arr[6]);
    Console.WriteLine("_____|_____|_____");
    Console.WriteLine("     |     |     ");
    Console.WriteLine("  {0}  |  {1}  |  {2}  ", arr[7], arr[8], arr[9]);
    Console.WriteLine("     |     |     ");
}

1 reference
private static int CheckWin()
{
    #region Horzontal Winning Condtion

    if (arr[1] == arr[2] && arr[2] == arr[3])
        return 1;

    else if (arr[4] == arr[5] && arr[5] == arr[6])
        return 1;

    else if (arr[7] == arr[8] && arr[8] == arr[9])
        return 1;
    #endregion

    #region Vertical Winning Condition

    else if (arr[1] == arr[4] && arr[4] == arr[7])
        return 1;

    else if (arr[2] == arr[5] && arr[5] == arr[8])
        return 1;

    else if (arr[3] == arr[6] && arr[6] == arr[9])
        return 1;
    #endregion
```

```
#region Diagonal Winning Condition
if (arr[1] == arr[5] && arr[5] == arr[9])
    return 1;
if (arr[3] == arr[5] && arr[5] == arr[7])
    return 1;
#endregion

#region Draw Condition

else if (arr[1] != '1' && arr[2] != '2' && arr[3] != '3' &&
        arr[4] != '4' && arr[5] != '5' && arr[6] != '6' &&
        arr[7] != '7' && arr[8] != '8' && arr[9] != '9')
    return -1;
#endregion

else
    return 0;
```

"Format of the output"



"Player Can not choose repeated shell"



"Player 1 Has Won"

"Player 2 Has Won"

Conclusion :

The Tic Tac Toe Game project provides a simple yet effective platform to learn programming concepts like arrays, loops, conditions, and modular structure. It enhances understanding of game logic implementation using C#.

Learning During Project Work :

- Fundamentals of game logic in C#
- Working with arrays and conditions
- User input validation
- Creating a menu-based console app
- Developing replay and reset functionalities


THANK YOU

Tic Tac Toe