**Documentation for Web Technology Project**

By w66256

**Lab Title:**

Unleash Your Inner Champion: Building a C# Tic Tac Toe Mastermind.

**Introduction:**

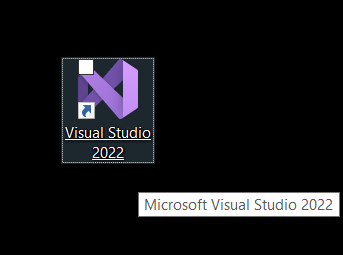
Tic Tac Toe, a timeless classic, offers a captivating blend of strategy and fun. In this tutorial, we'll delve into the heart of crafting a compelling implementation using **C#** in **Visual Studio.** Through this guided journey, you'll master essential programming concepts like **Buttons**, **Timers**, **Labels**, **Functions**, **If** Statements, **Loops**, **Lists**, and the **Random** class. By the end, you'll not only conquer the Tic Tac Toe grid but also develop an insightful AI opponent, culminating in a thrilling interactive experience.

**Prerequisites:**

* Basic understanding of **C#** programming syntax and concepts.
* Access to **Visual Studio.**

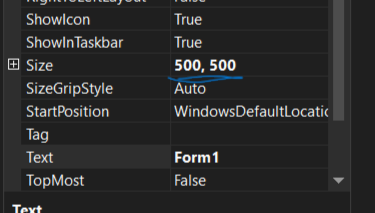
**Steps:**

1. Start **Visual Studio**.



Create a new project with **Visual Studio.** Under the **C#** programming language, choose Windows Form Application. Name the project tictactoe and click OK.

1. Make the following modifications to the new **form**'s properties:

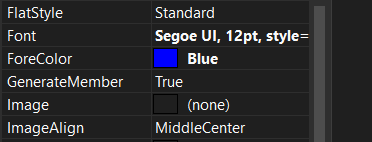


Set the size to - 500, 500.

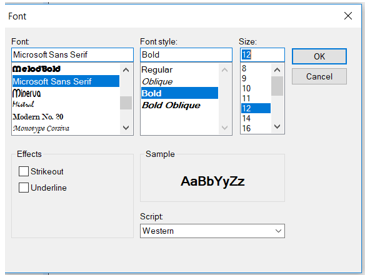
We will now need to add components to the **form**. Drag and drop two **labels** from the **tool box** onto the **form**.



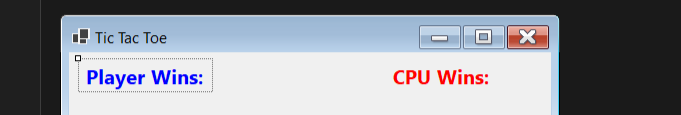
Place them both on the top of the **form**; they will display the player and computer/AI scores, respectively. Let us make some adjustments to the characteristics of both of these **labels**.



To access the **font** dialog box, click the three-dotted button under the **font** section in the properties.

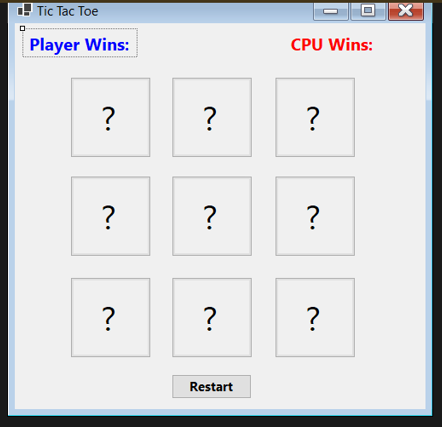


Make the typeface **BOLD** and **size 14**. Click **OK**.

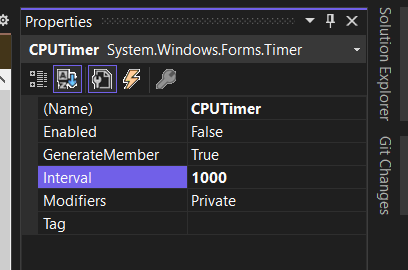


Change the content of **label 1** to Player Wins, and **label 2** to CPU Wins .   
  
Change the forecolours of **labels 1** and **2** to blue and red, respectively.

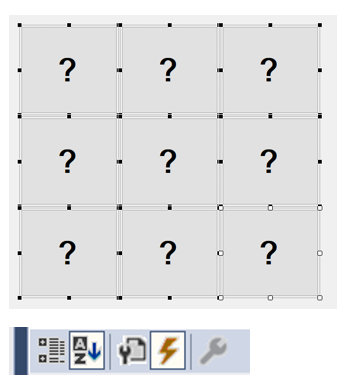
1. We now need to add a number more **buttons** to the screen. Let's start with merely one. We'll adjust its attributes, then copy and paste it as needed. Drag and drop a **button** onto the form.   
     
   Change the following properties.  
     
     
   Size - 95, 95   
     
   Text - Segoe UI, size 24px.



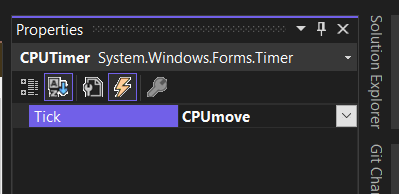
Add one additional **button** to the scene and update the wording to "Restart." This **button** will reset all of the other **buttons**. Do not add any tags to the restart **button**.   
  
We need to include a **Timer** component on this form. Drag and drop a **timer** from the **toolbox** into the **form**.



Now we need to start adding events to our game. Select all of the question mark **buttons** on the **form**. And I arrived to the events window.



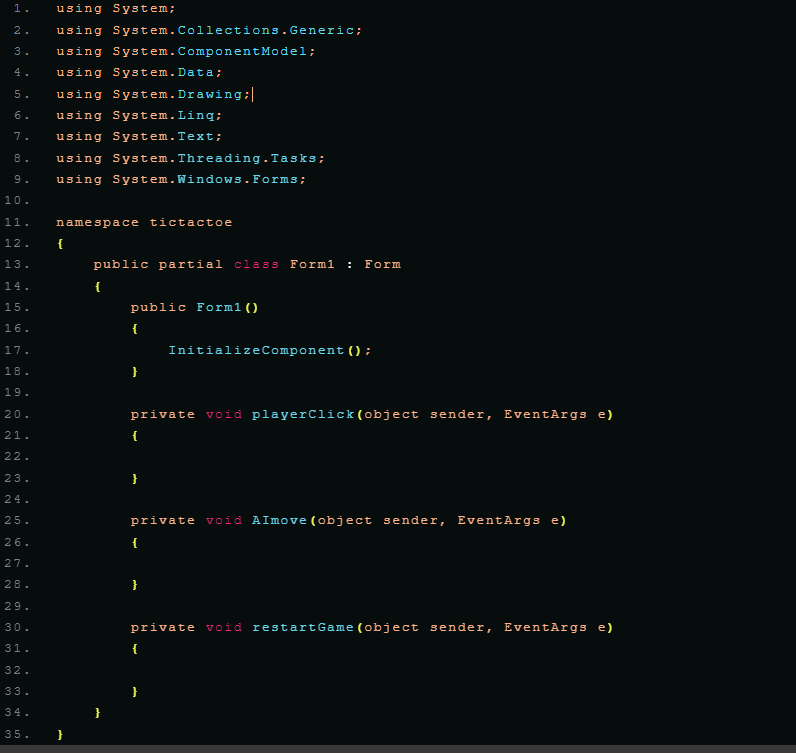
While all of the **buttons** are highlighted, click the lightning bolt icon in the properties panel to get to the events window.   
  
In the events box, locate the click event and fill in **CPUmove** and hit enter.



Now click on the **CPUTimer** object.

As previously, in the events box, type **CPUmove** and hit Enter.

Now we have all of our events in place time to add the code.



4. Defining Variables and Functions:

- Create an **enum** called "Player" with values "X" and "O".

- Declare variables:

- player (of type "Player")

- random (new Random object)

- **playerWinCount** (integer, initialized to 0)

- **cpuWinCount** (integer, initialized to 0)

- **buttons** (list of Button objects)

- Create functions:

- **restartGame**:

- Clears the board by disabling all buttons, setting their text to "?", and changing **color** to default.

**- checkGame:**

- Checks all winning conditions (horizontal, vertical, diagonal) for both players and CPU.

- Shows a message box and updates win counts if a win is detected.

- Stops the **timer** and calls **restartGame.**

5. Implementing Event Handlers:

- **cpuMove**:

- Checks if any buttons are still enabled.

- If yes, randomly select an enabled button, disable it, set its text and **color**, and remove it from the buttons list.

- Call **checkGame**.

- Stop the timer.

- **playerClick**:

- Identify the clicked button.

- Set the button's text to the current player ("X" or "O").

- Disable the button.

- Remove the button from the buttons list.

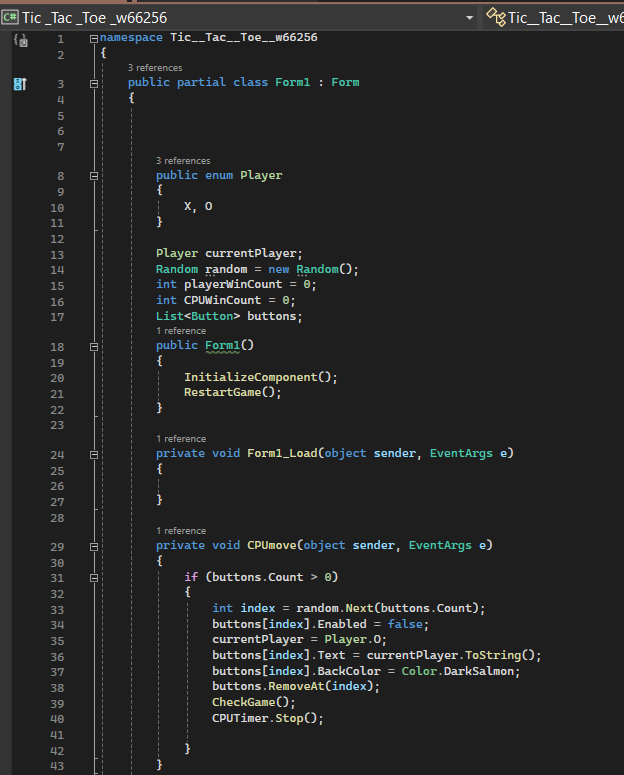
- Call **checkGame**.

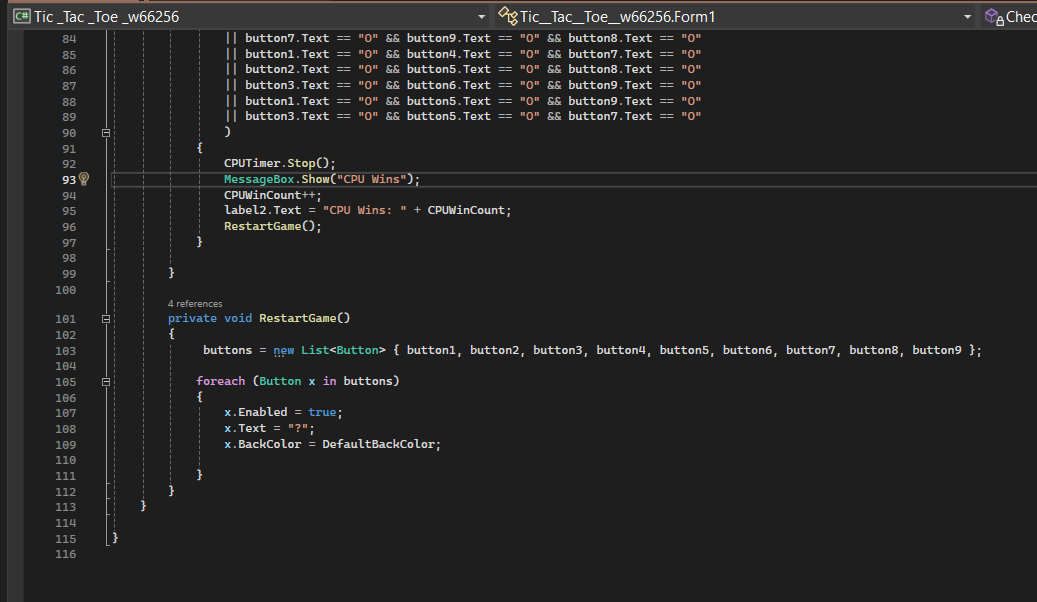
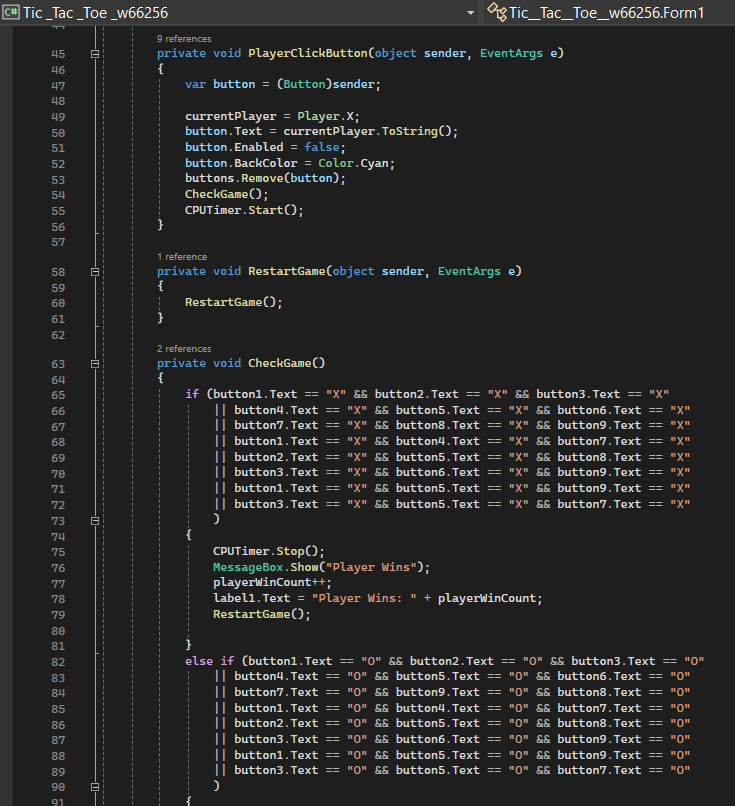
- Start the **timer**.

- **restartGame**:

- Call **restartGame** function.

**These are codes:­­­**



6. Testing and Running:

- Run the application and test the gameplay.

- Make sure both players and CPU can win, and wins are tracked correctly.

**Additional Notes:**

- The provided steps are a simplified guide.

- Remember to adjust the provided code if you have made changes to the button names or labels.

I hope this step-by-step solution helps you create your own Tic Tac Toe game!