

Forum:[Hack Forums](#)

Thread: [\[C#\]\[SRC\]\[TuT\]Simple TicTacToe](#)

Ethereal - February 3, 2011, 9:43 am

Please note: All the information in this tutorial was written by me. I simplified everything, so as to avoid "Get/Set" clauses and Graphics, and instead use simplistic conditionals. Thank you Deities for helping me format it.

My example project is [here](#). The aim of this tutorial is to promote tidy code, and to utilise proper OOP potential. This means separating everything out into reusable code/functions, in order to increase efficiency.

NOTE* This tutorial is subject to update.

Before we get started! ([Click to View](#))

First and foremost, you must create your form. Design it however you want, but it must contain: 9 Square shaped buttons with no text – This is your game area

3 other buttons, other shapes is fine. These should contain the text "Reset", "X To Start" and "O to Start"

5 Labels. Three will be counters with the text "X won 0 times", "O won 0 times" and "You have drawn 0 times", and the other two will be "" and X's turn.

Next, rename your components to something more memorable, so as to not confuse yourself later on. I left the square buttons as they were, and renamed the other buttons to their respective text("Reset" to "Reset" etc.). The labels should be something along the lines of "xCounterLabel", "DrawCounterLabel" and "oCounterLabel", with the two turn-based labels being "xLabel" and "oLabel". It always irritates me when I find huge source codes with "button24" and "label9", with no clear indication of what each one is.

Now, double click the form and create several new variables:

PHP Code

```
<?php
bool X = true; //Used for testing who's turn it is.
bool O = false; //Ditto
int xWinCounter = 0; //The counters
int oWinCounter = 0; //Ditto
int DrawCounter = 0; //Ditto
This will be used to determine who's go it is, and to keep the counters working.
```

Next, create a new region of your choice, which will contain all of the "button_click" events. This is simply to keep our code tidy, without having to separate it into separate classes(which gets irritating).

PHP Code

```
<?php
#region Button Logic
//This region/section is for all of our "button_click" events. Lets keep our code tidy, eh?
```

```
#endregion //End the Button Logic region.
```

Next, click all of your buttons, and move their click events to this region(if needed).

PHP Code

```

<?php
#region Button Logic
//This region/section is for all of our "button_click" events. Lets keep our code tidy, eh?
private void button1_Click(object sender, EventArgs e)
/*The first button. The buttons on my form are numbered the same as a phone.*/
{

}
#endregion

```

Button Logic Region! [\(Click to View\)](#)

This may get a little confusing for you, but just look at the code. Don't forget that you can PM me anytime, if you need help.

The following code will make little sense to you at first, but you will understand it later on:

PHP Code

```

<?php
if (X == true) //If it is X's turn
{
    UpdateX(); //Update the turn
    button1.Text = "X"; //Write an 'X'
}
else if (O == true) //Or if it is O's turn
{
    UpdateO(); //Update the turn
    button1.Text = "O"; //Write an 'O' instead
}
button1.Enabled = false; //Disable the button, so the player cannot click twice.
IsWinTrue(); //Check if the person has won.
IsDrawTrue(); //Check if the person has drawn.
AreChoiceButtonsDisabled(); //Check if the choice buttons("X to Start"|"O to Start") are disabled.

```

First, the code checks to see if it is X's turn or not using the Boolean "X", and likewise O. This simple logic writes either an X or an O to the button, and then changes the turn.

Next, the code disables the button, so that it cannot be clicked again. This is to stop the previous value being overwritten.

Finally, the code checks if the person has Won, Drawn and if the switch turn buttons are disabled or not.

Phew! A lot to take in?

Comment out the non existent functions for now, and then add this code to all of the square buttons(changing the button the code refers to of course(1-9)).

The three other buttons should look like this:

PHP Code

```

<?php
private void Reset_Click(object sender, EventArgs e) //When the player tries to Reset
{
    ResetGame(); //Reset the game
}

private void XToStart_Click(object sender, EventArgs e) //When the player wants 'X' to start first

```

```

{
    UpdateO(); //Update 'O'. This function will act as if 'O' has just had a go, and switch the turn.
}

private void OToStart_Click(object sender, EventArgs e) //When the player wants 'O' to start first
{
    UpdateX(); //Update 'X'. This function will act as if 'X' has just had a go, and switch the turn.
}

```

In true OOP style, I have sorted the rest of the code into reusable functions. Read on for them!

The Validation Region! [\(Click to View\)](#)

First, let's create this new region called "Validation".

PHP Code

```

<?php
#region Validation
    //This region will contain all of our data validation functions.
#endregion //The end of our Data validation section.

```

Inside this region, create three Void functions called "IsWinTrue", "IsDrawTrue" and "AreChoiceButtonsDisabled", as follows:

PHP Code

```

<?php
void IsWinTrue()
{
}

void IsDrawTrue()
{
}

void AreChoiceButtonsDisabled()
{
}

```

Inside the first(IsWinTrue), add this code:

PHP Code

```

<?php
void IsWinTrue()
{
    if (button1.Text == "X" && button2.Text == "X" && button3.Text == "X") /*If the game board looks
    like this: [X] [X] [X]
    [] [] []
    [] [] [] */
    {
        xWon(); //Then xWon.
    }
}

```

This code tests if Buttons 1, 2 and 3 contain "X", and if so, call the function "xWon". This is our

simple test to end the game if someone wins in this way.

Next, you must add the other possible Wins(3 Horizontal, 3 Vertical, 2 Diagonal), or if you are lazy, just add this:

PHP Code

```
<?php
else if (button1.Text == "O" && button2.Text == "O" && button3.Text == "O")
{
    oWon();
}
else if (button4.Text == "X" && button5.Text == "X" && button6.Text == "X")
{
    xWon();
}
else if (button4.Text == "O" && button5.Text == "O" && button6.Text == "O")
{
    oWon();
}
else if (button7.Text == "X" && button8.Text == "X" && button9.Text == "X")
{
    xWon();
}
else if (button7.Text == "O" && button8.Text == "O" && button9.Text == "O")
{
    oWon();
}
else if (button1.Text == "X" && button4.Text == "X" && button7.Text == "X")
{
    xWon();
}
else if (button1.Text == "O" && button4.Text == "O" && button7.Text == "O")
{
    oWon();
}
else if (button2.Text == "X" && button5.Text == "X" && button8.Text == "X")
{
    xWon();
}
else if (button2.Text == "O" && button5.Text == "O" && button8.Text == "O")
{
    oWon();
}
else if (button3.Text == "X" && button6.Text == "X" && button9.Text == "X")
{
    xWon();
}
else if (button3.Text == "O" && button6.Text == "O" && button9.Text == "O")
{
    oWon();
}
else if (button1.Text == "X" && button5.Text == "X" && button9.Text == "X")
{
    xWon();
}
```

```

}
else if (button1.Text == "O" && button5.Text == "O" && button9.Text == "O")
{
oWon();
}
else if (button3.Text == "X" && button5.Text == "X" && button7.Text == "X")
{
xWon();
}
else if (button3.Text == "O" && button5.Text == "O" && button7.Text == "O")
{
oWon();
}
}

```

Next is our "IsDrawTrue" function.

PHP Code

```
<?php
```

```
void IsDrawTrue()
```

```

{
    if (button1.Text != "" & button2.Text != "" & button3.Text != "" & //If every square is filled,
        button4.Text != "" & button5.Text != "" & button6.Text != "" & //and it hasn't been selected by
        button7.Text != "" & button8.Text != "" & button9.Text != "") //IsWinTrue
    {
        Draw(); //Call the Draw function.
    }
}

```

I don't really need to write anything else about this.

And the Final function, "AreChoiceButtonsDisabled":

PHP Code

```
<?php
```

```
void AreChoiceButtonsDisabled()
```

```

{
    if (button1.Text != null || button2.Text != null || button3.Text != null || //If any data value has
        button4.Text != null || button5.Text != null || button6.Text != null || //been entered into any
        button7.Text != null || button8.Text != null || button9.Text != null) //square
    {
        XToStart.Enabled = false; //Disable the X to start button
        OToStart.Enabled = false; //Disable to O to start button
    }
}

```

This simply tests whether or not the switch turn buttons are disabled or not. Potentially, you could combine them into a single button(Switch Turn), but that is up to you.

The Extra Functions Region! ([Click to View](#))

This is our final region. Almost there, eh? This section will contain the rest of our functions, the ones that haven't been declared yet.

The code is commented to aid with your understanding, but I don't feel that I need to explain each part. If I am mistaken, please correct me.

PHP Code

```
<?php
```

```
#region Extra Functions
```

```

void UpdateX() //Update X - X has taken a turn.
{
    X = false; //Change the bool to true/false, for validation on each 1-9 button event.
    O = true; //Ditto
    xLabel.Text = null; //Alter the labels - this is just for the GUI
    oLabel.Text = "O's Turn"; //Ditto
}

void UpdateO() //Update O - O has taken a turn.
{
    X = true; //Change the bool to true/false, for validation on each 1-9 button event.
    O = false; //Ditto
    xLabel.Text = "X's Turn"; //Alter the labels - this is just for the GUI
    oLabel.Text = null; //Ditto
}

void EnableButtons() //Enable the buttons. They are disabled after clicked
{
    button1.Enabled = true;
    button2.Enabled = true;
    button3.Enabled = true;
    button4.Enabled = true;
    button5.Enabled = true;
    button6.Enabled = true;
    button7.Enabled = true;
    button8.Enabled = true;
    button9.Enabled = true;
}

void EnableChoiceButtons()
//Enable the choice buttons that are disabled when you start the game.
{
    XToStart.Enabled = true;
    OToStart.Enabled = true;
}

void xWon() //If 'X' wins
{
    MessageBox.Show("X Wins!"); //Show a Message
    ResetGame(); //Reset the game
    xWinCounter++; //Increase the count by one.
    xCounterLabel.Text = "X won " + Convert.ToString(xWinCounter) + " time(s)";
//Update the counter label
}

void oWon() //If 'O' wins
{
    MessageBox.Show("O Wins!"); //Show a Message
    ResetGame(); //Reset the game
    oWinCounter++; //Increase the count by one.
    oCounterLabel.Text = "O won " + Convert.ToString(oWinCounter) + " time(s)";
//Update the counter label
}

```

```

void Draw() //If it's a draw
{
    MessageBox.Show("Draw!"); //Show a Message
    ResetGame(); //Reset the game
    DrawCounter++; //Increase the count by one.
    DrawCounterLabel.Text = "You have drawn " + Convert.ToString(DrawCounter) + " time(s)";
//Update the counter label
}

void ResetButtonText() //Reset the button text
{
    button1.Text = null;
    button2.Text = null;
    button3.Text = null;
    button4.Text = null;
    button5.Text = null;
    button6.Text = null;
    button7.Text = null;
    button8.Text = null;
    button9.Text = null;
}

void ResetGame() //Reset function
{
    ResetButtonText(); //Call the reset text function
    EnableButtons(); //Enable the buttons
    EnableChoiceButtons(); //Enable the choice buttons
}

#endregion

```

Cephei - February 3, 2011, 10:03 am

I like it. Thanks for the release.

RiperFish - February 3, 2011, 10:03 am

Thank's for the Tut budy (original = 5/5).

GamingManiac - February 3, 2011, 12:16 pm

Thanks for this - I did a C++ console program that does the same thing, just with no GUI. so i understand the logic

Very nice dude

Fake. - February 3, 2011, 12:27 pm

Nooo, don't use simple conditionals to check the wins):
Think of arrays and such.

Ethereal - February 3, 2011, 12:31 pm

(Feb 3, 2011 01:27 PM)jacktheripper Wrote: [\[CITE\]](#) Nooo, don't use simple conditionals to check the wins):

Think of arrays and such.
Incase you didn't read:

I said at the top that I had simplified the code for people to learn from.

Fake. - *February 3, 2011, 9:41 pm*

(Feb 3, 2011 01:31 PM)Ethereal Wrote: [\[CITE\]](#) Incase you didn't read:

I said at the top that I had simplified the code for people to learn from.
Ah shit, my bad. I read everything else, sorry about that.

Ethereal - *February 4, 2011, 10:57 am*

(Feb 3, 2011 10:41 PM)jacktheripper Wrote: [\[CITE\]](#) Ah shit, my bad. I read everything else, sorry about that.

It's no problem.

People, feel free to post is something confuses you :)