

Final Project Help - Connect 4

Data to Track

The playing board is a canvas with a grid of buttons, 7 x 7.

```
// Create a two dimensional array of storage for the buttons
Button[ , ] board = new Button[7, 7];
```

```
// Create a Canvas control on your window. Make the height and width the same number. Use
a number that is evenly divisible by 7. 490 x 490 might be good. If you choose 490, then each
button is 70 x 70.
```

```
// Use code to create and position the buttons on the canvas
```

```
Loop - row 0 through 6
```

```
    Loop - column 0 through 6
```

```
        Create a button (see other advanced guide on creating controls with code.)
```

```
        Set the position of the button to be column * button width, row * button height
```

```
        Set the .Tag property of the button to be a Point object that will store the column
and row the button is on. For example: b.Tag = new Point (column, row);
```

```
        Set the Click event handler (all buttons use the same method)
```

```
        Set the button to be added to the list of children of the Canvas (you gave it a
name, right?)
```

Clicking a button

When the player clicks a button, they are really selecting the column they want to drop the token in.

Loop in reverse order, from the bottom of the column to the top looking for an open button, one that the Content is null.

In the click handler the button clicked is the sender

```
void ClickHandler (object sender, ... )
```

```
{
```

```
// Sender is the button that was clicked
```

```
Button bClicked = (Button) sender;
```

```
Point buttonPos = (Point) bClicked.Tag;
```

```
int column = (int)buttonPos.X;
```

Loop - row 6 through 0

```
    Button b = board[column, row];  
    If the button b.Content is null then it is open  
        Put an Ellipse in b.Content  
        Set the color to be the right player color  
        Break out of the loop with break;
```

Checking for four in a row

If you have a grid of 7 x 7 and you are looking for four in a row, up-down, left-right, or diagonal. The way to detect a win is to look through the board (buttons) for four buttons that have an Ellipse and the colors are all the same.

Loop through all rows and columns

Check if you can look down three buttons. Is row + 3 < 7?

Looking down from position [column, row]:

```
If (board[column, row].Content != null &&  
    board[column, row + 1] != null &&  
    board[column, row + 2] != null &&  
    board[column, row + 3] != null)  
{  
    // No check all four if they are the same color  
    Ellipse e1 = (Ellipse)board[column, row].Content;  
    Ellipse e2 = (Ellipse)board[column, row + 1].Content;  
    Ellipse e3 = (Ellipse)board[column, row + 2].Content;  
    Ellipse e4 = (Ellipse)board[column, row + 3].Content;  
  
    if (e1.Fill == e2.Fill && e2.Fill == e3.Fill && e3.Fill == e4.Fill)  
    {  
        // There are four in a row!!!  
        // Now what do you do? Similar to tic tac toe. :-)  
    }  
}
```

// Repeat for four across

// And four diagonally left

// And four diagonally right