CO453 Application Programming

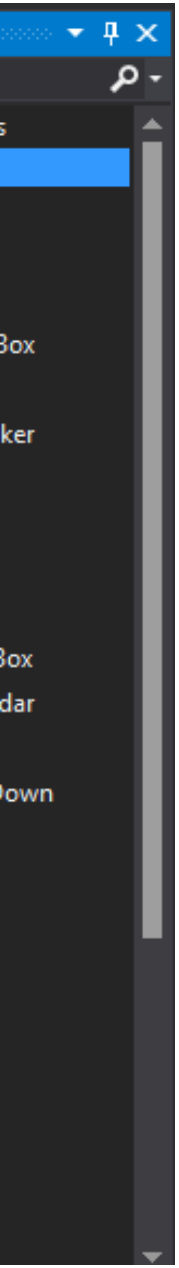
Week 7 – More controls

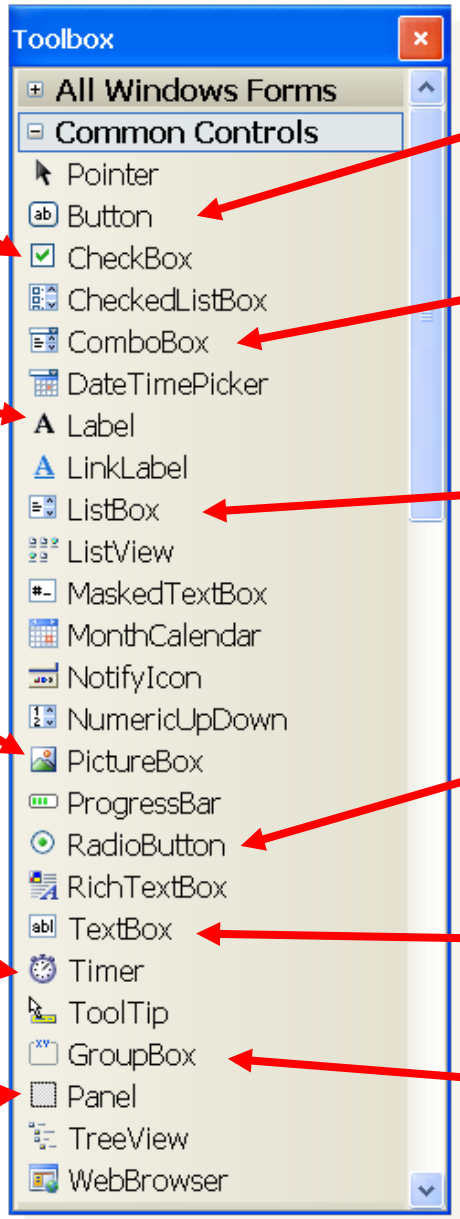
.NET part 3

More controls

- Radio Buttons
- Checkboxes
- ListBoxes
- ComboBoxes
- Scroll bars
- Menu bars
- MessageBoxes (or alerts)
- Timers

Other Objects on the ToolBox





Check Box

Label

PictureBox

Timer

Panel

Button

Combo Box

List Box

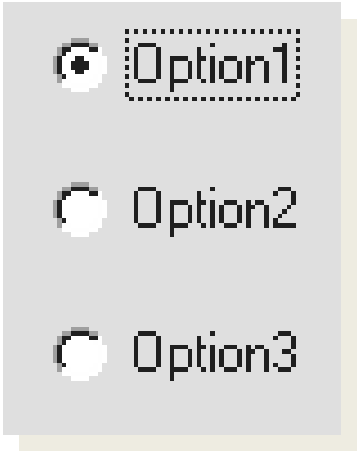
RadioButton

Text Box

GroupBox

Some More C# .NET Objects to use

1: RadioButtons



- Allows you to make choices
- Click one only .. others automatically deselect
- The **Checked** property is set **True** or **False**

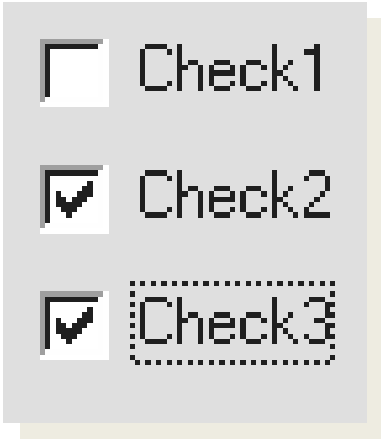
- Here a **GroupBox** has been added to the form
- Its **Text** has been changed to **Meal Choices**
- 3 **RadioButtons** added and **Text** changed as shown
- The **Checked** property of the Veg option is currently **true**

Example Code

```
if  (rbnVeg.Checked)
{
      vegCount ++ ;
}
```

Put this code in the **rbnVeg_CheckedChanged()** method

2: CheckBoxes



- Similar to RadioButtons
- But Multiple selections can be made
- The **Checked** property is True or False

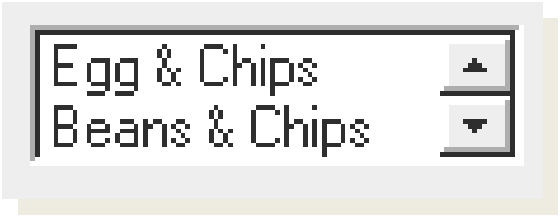
- Here a **GroupBox** has been added to the form
- Its **Text** changed to **Taste in Music**
- 3 **CheckBoxes** added and **Text** changed as shown
- The **Checked** property of the Rock CheckBox is currently **true**

Example Code

```
if  (chkJazz.Checked)
{
    tastePoints += 25 ;
}
```

Put this code in the **chkJazz_CheckedChanged()** method

3: ListBoxes



- Selection is made from a list of choices
- Scroll bars appear if the list is too long
- The **Sorted** property can be set **True** to sort the list items
- The **Items** property can be used to create a list

- Here a **ListBox** (named lstMenu) has been put in a **GroupBox** that has **Text** property **Lunch Menu**
- The **Items** property has been used to set up meals

Example Code

```
switch (lstMenu.Text)
{
    case  "Egg & Chips" :
                    lblBill.Text = "£3.50" ; break;
    case "Beans & Chips" :
                    lblBill.Text = "£3.75"; break;
    case "Chicken Curry" :
                    lblBill.Text = "£4.50"; break;
}
```

Put this code in the **lstMenu_SelectedIndexChanged()** method

4: ComboBoxes

- Like a ListBox and TextBox combined
- Users can be allowed to change or add new values
- The **DropDownStyle** property allows 3 styles of box:
  - DropDown combo (this is the default)
  - Simple (with no drop down option)
  - DropDownList (users **can't** add new items)

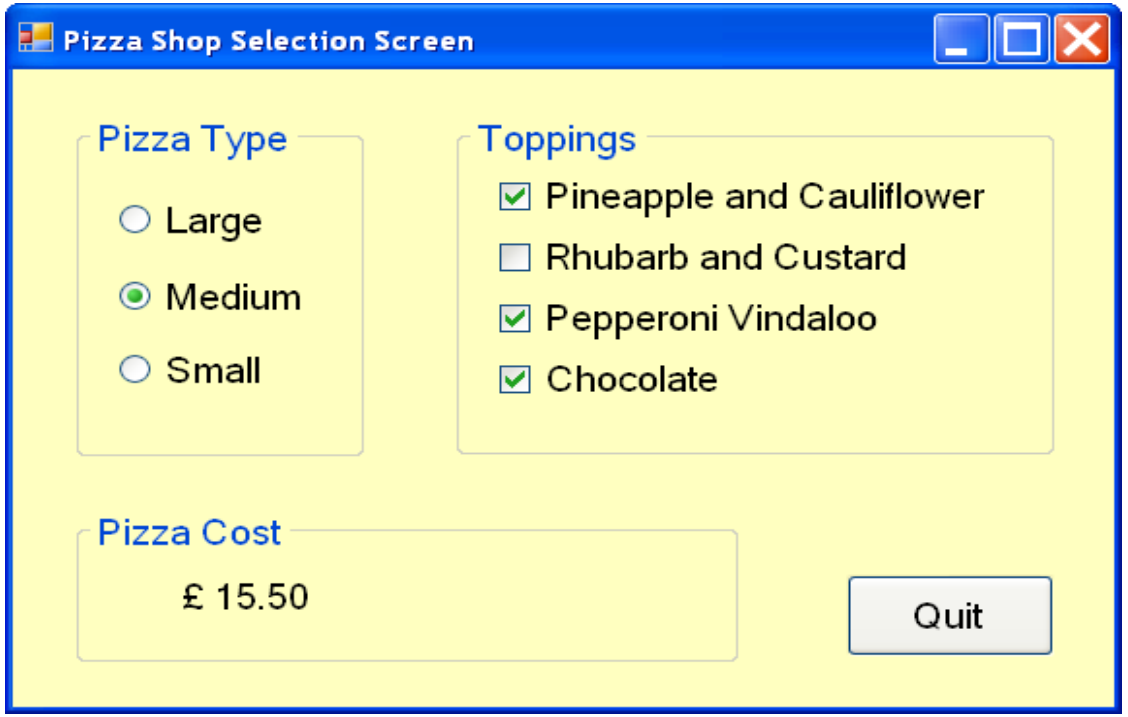
- Here a Dropdown **ComboBox** has been used
- The **Items** property was used to set up the meals
- The **Text** property has been set to **Grub Selection**
- Notice the dropdown combo takes up less space

Example Code

- Similar code can be used as for the list box
- the name of the box must obviously be changed to that of the combo box

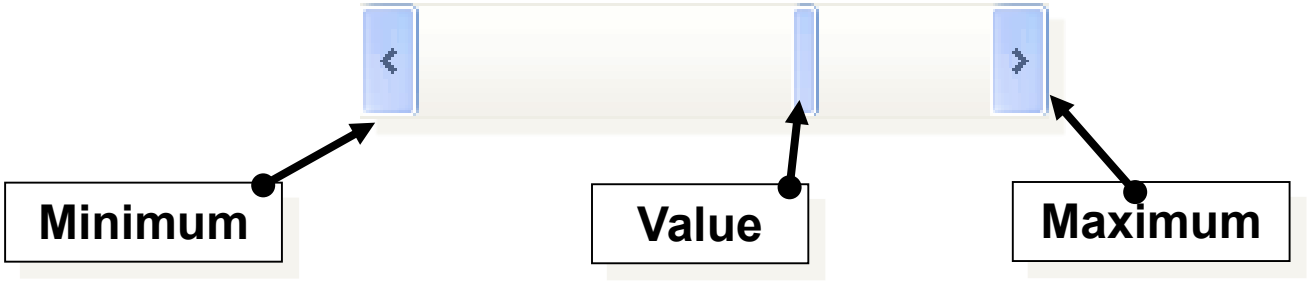
Activity

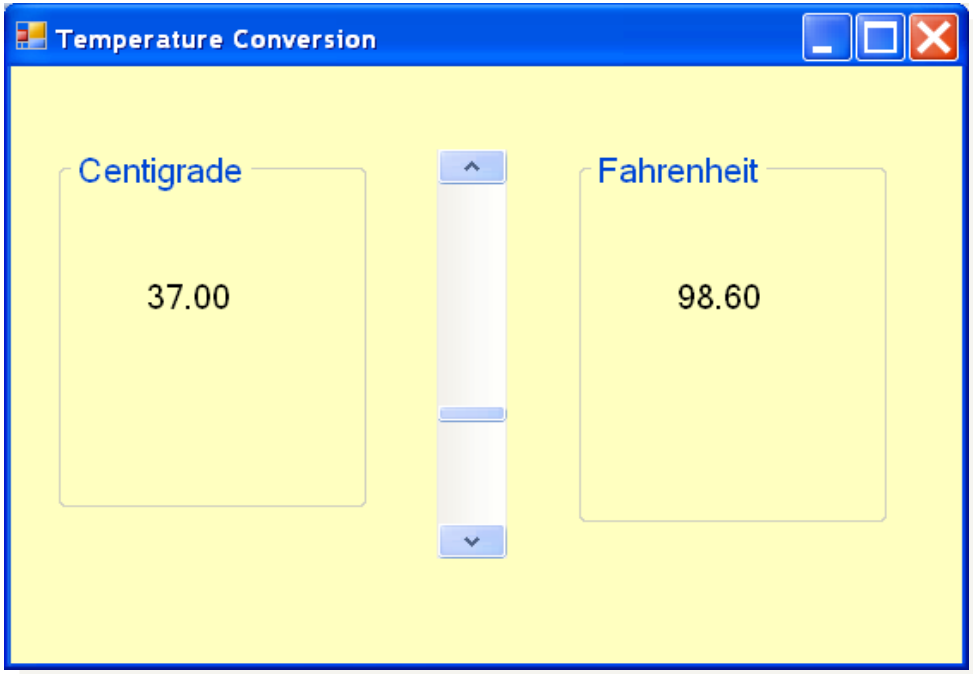
- Attempt to complete Task 3.2 (open PizzaShop)



(copy the ‘C# Progs’ folder from the L:/ drive if you haven’t already)

Scroll Bars (HScrollBar or VScrollBar)





```
degC = vsbTemp.Value;
lblCent.Text = degC.ToString("0.00");
```

Put code in the **Scroll()** method for the ScrollBar

The MessageBox
(for temporary outputs)

Simple MessageBox

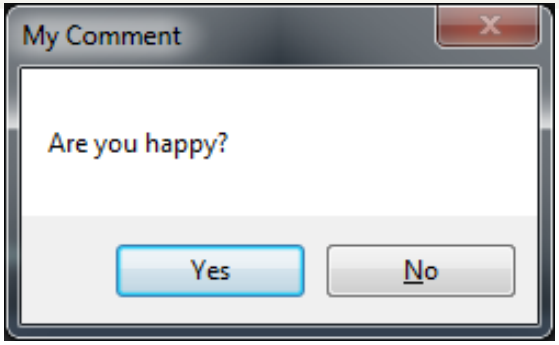
MessageBox.Show("Hello World!");

MessageBox.Show("Hello world!", "My Comment");

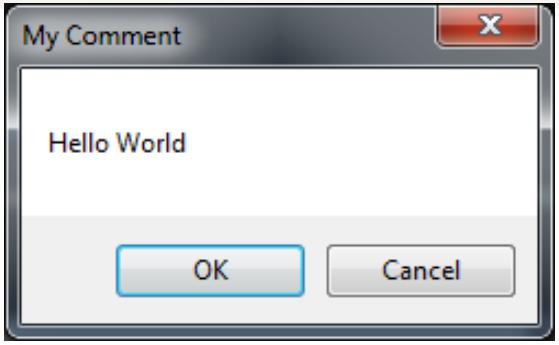
Stays on screen until you click OK .. Then disappears

Some MessageBox Styles

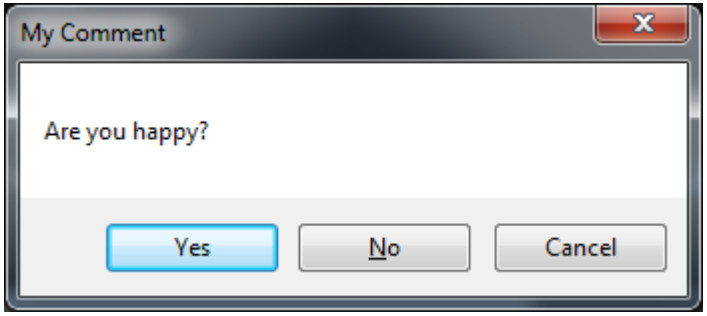
```
MessageBox.Show("Are You Happy?", "My Comment",
                MessageBoxButtons.YesNo);
```



MessageBoxButtons.YesNo



MessageBoxButtons.OKCancel

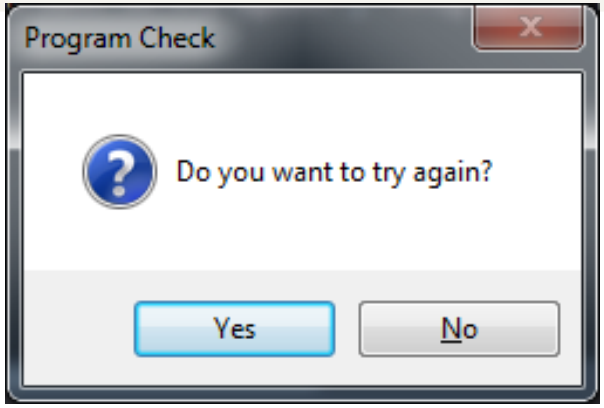


MessageBoxButtons.YesNoCancel

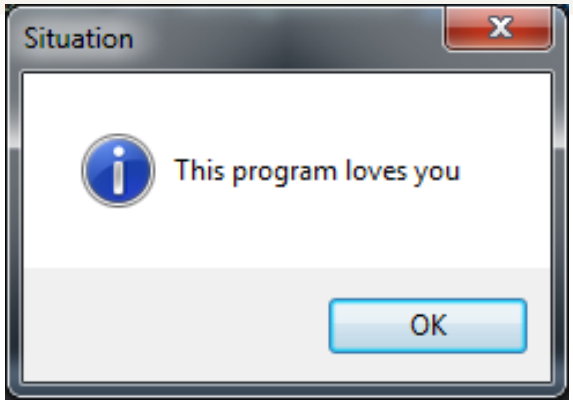
MessageBoxButtons.AbortRetryIgnore

More MessageBox Icons

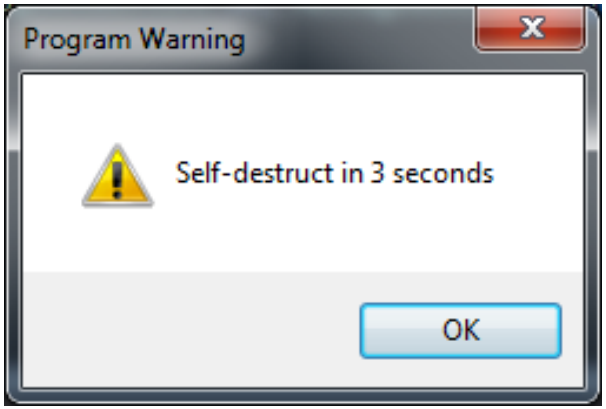
```
MessageBox.Show(" … ", " … ", MessageBoxButtons.YesNo,
                                MessageBoxIcon.Question);
```



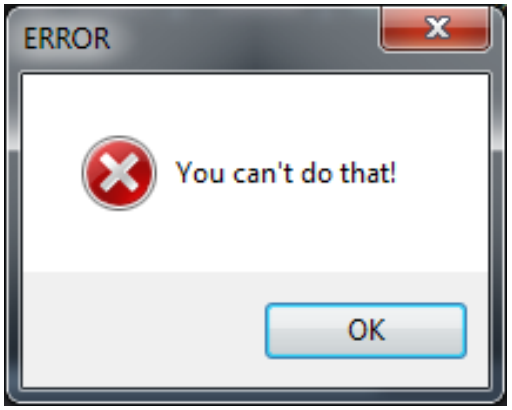
MessageBoxIcon.Question



MessageBoxIcon.Information



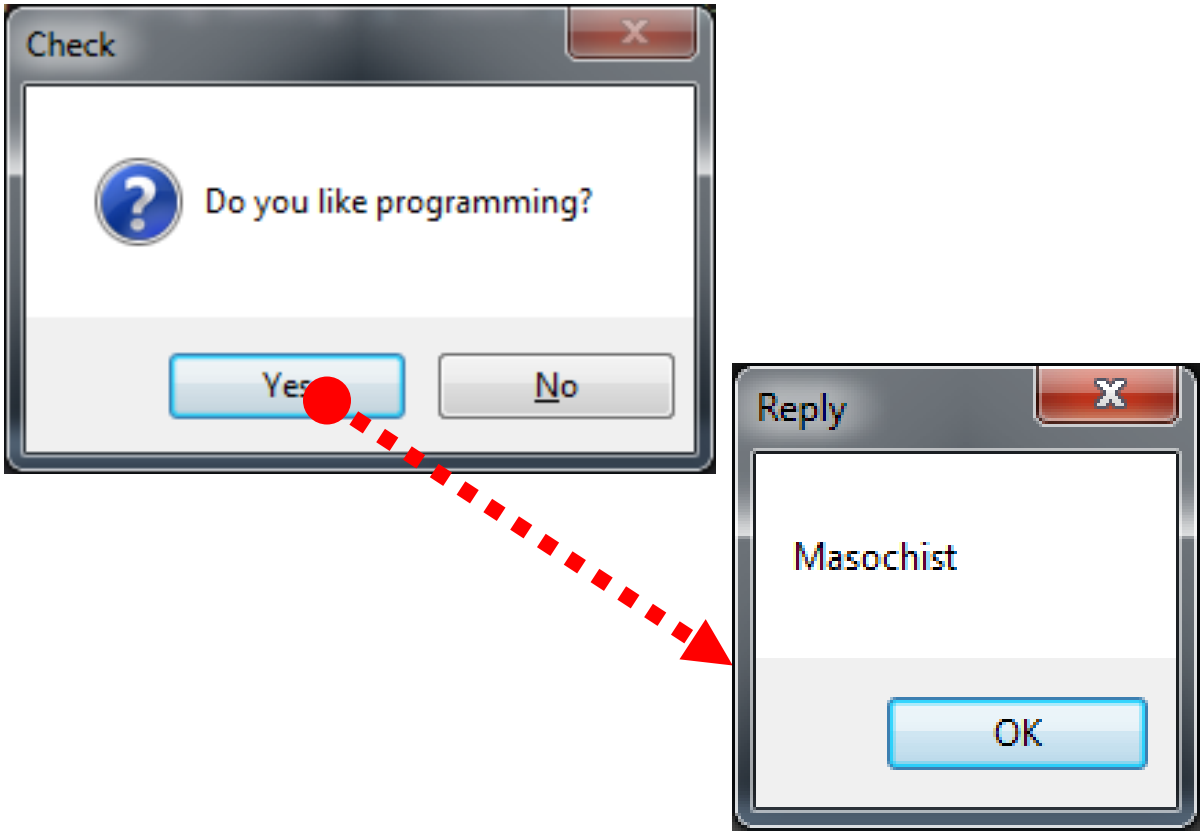
MessageBoxIcon.Warning



MessageBoxIcon.Error

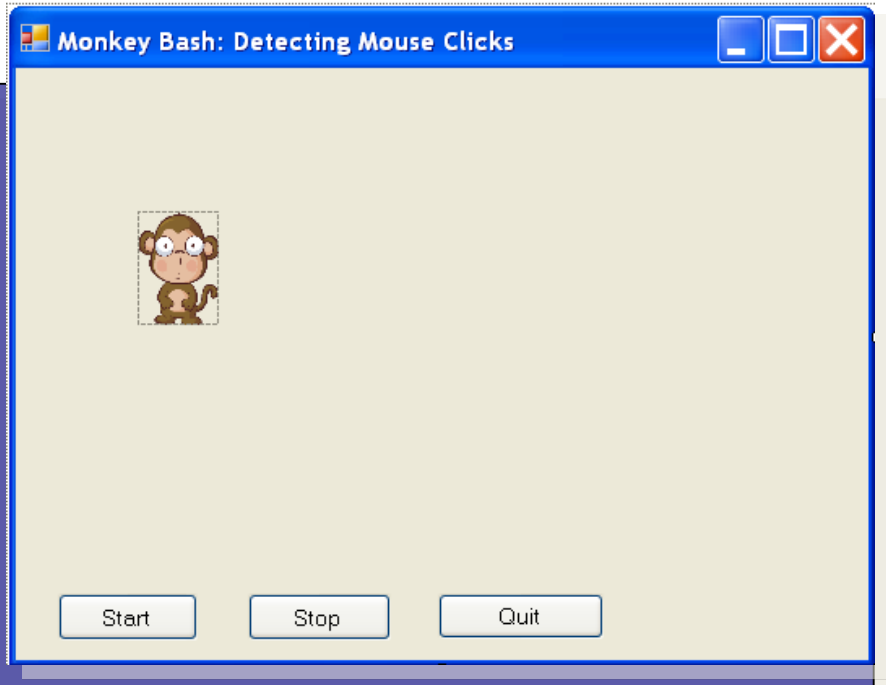
Responding to MessageBoxes

How can we detect which button was pressed and respond to it?



Code to Respond to MessageBox

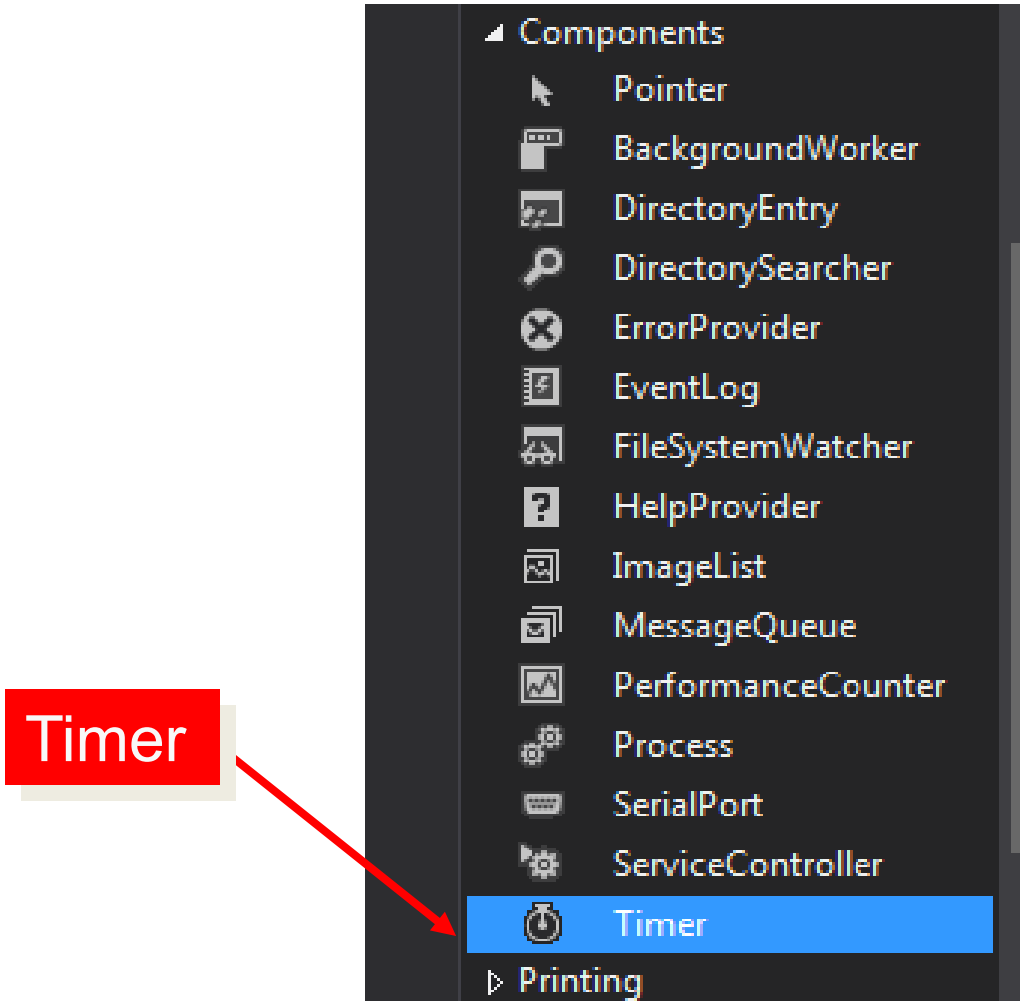
```
DialogResult  response;     // define a variable for result

response = MessageBox.Show("Do you like programming?",
                           "Check", MessageBoxButtons.YesNo);

if (response == DialogResult.Yes)
{
      MessageBox.Show("Masochist!", "Reply");
}
```



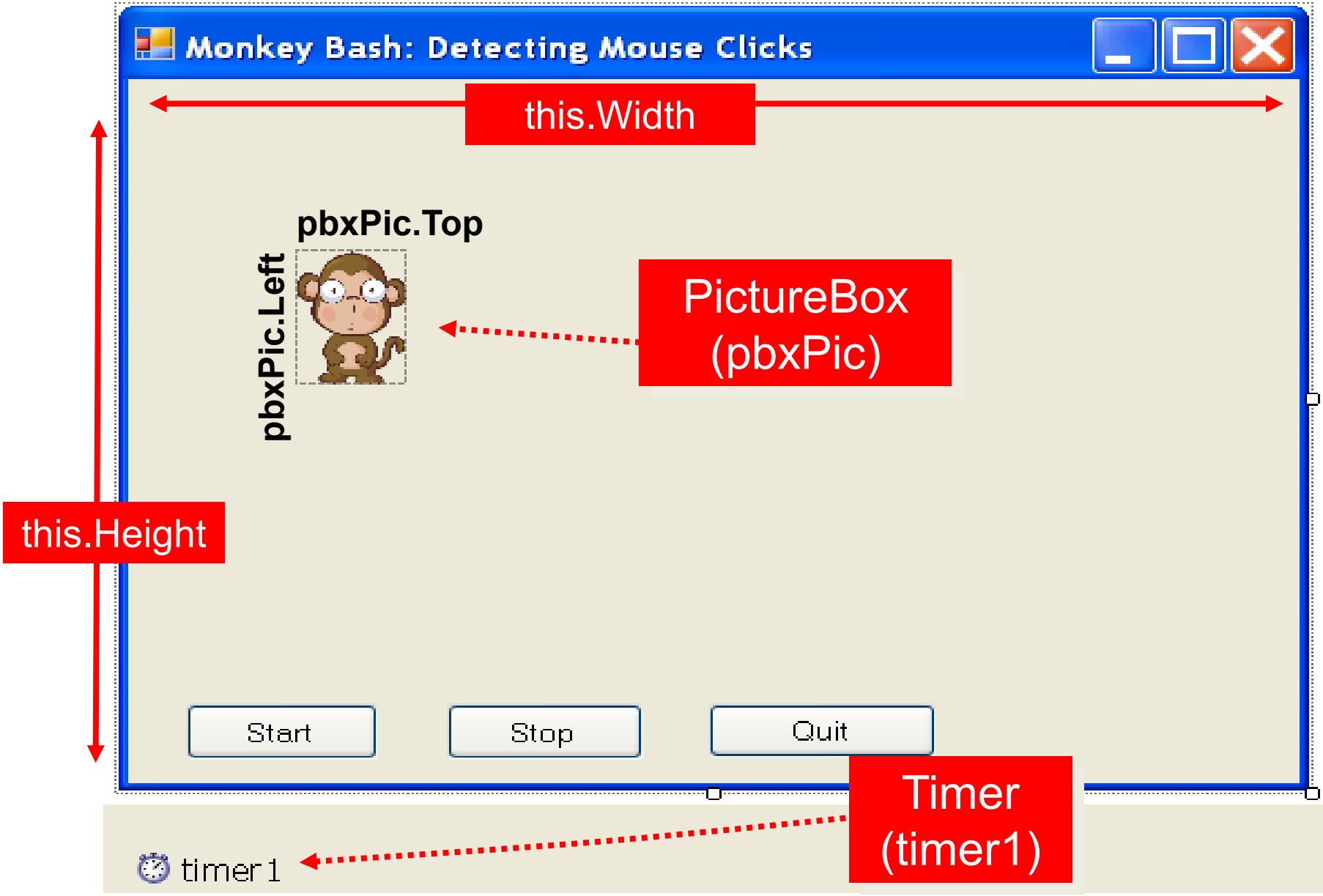
Monkey Bash

(Using a timer control and detecting mouse clicks)

The Timer Control



PictureBox moves at random



Using Timer to Move Monkey

Form1.cs [Design]* | Form1.cs* | Start Page

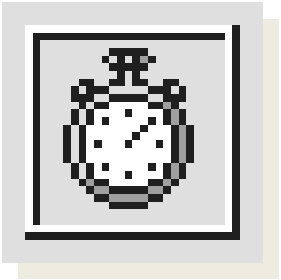
Mouses.Form1 | timer1_Tick(object sender, EventArgs e

```
private void btnStart_Click(object sender, EventArgs e)
{
    timer1.Enabled = true;          // start the timer
}

private void btnStop_Click(object sender, EventArgs e)
{
    timer1.Enabled = false;         // stop the timer
}

private void timer1_Tick(object sender, EventArgs e)
{
    x = rand.Next(this.Width - 100);
    y = rand.Next(this.Height - 100);
    pbxPic.Left = x;              // set new Left positio
    pbxPic.Top = y;               // set new Top position
    Refresh();                    // redraw in new positi
}
```

A Note about Random

```
// create a new random object called rand
Random rand = new Random();

// pick x randomly between 0 and the form width
x = rand.Next(this.width);

// using   this.width – 100  ensures the monkey
// doesn't overlap the right edge
```

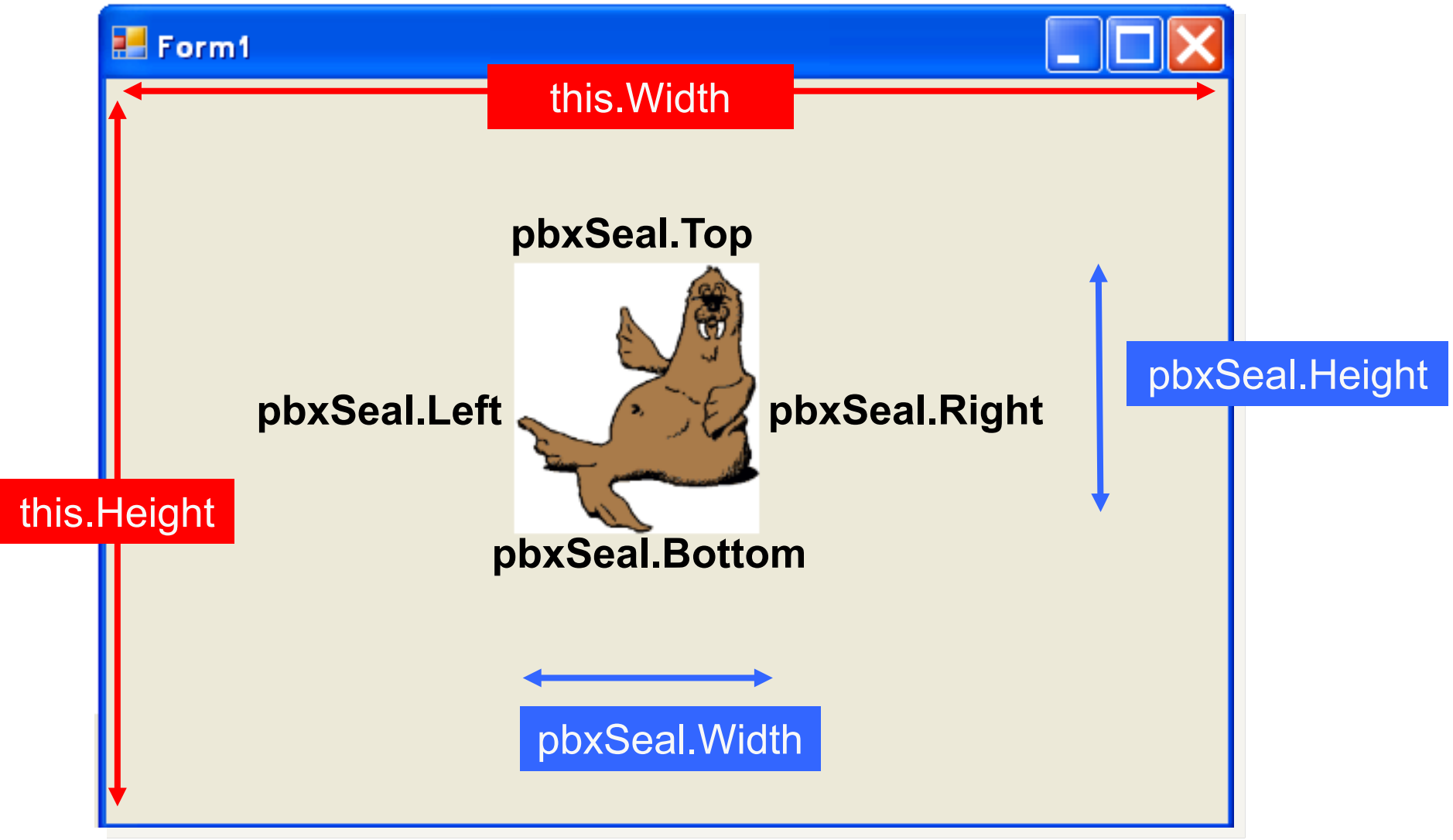
The Timer Control



- If you use a timer, it stays **hidden** from view when you run the program
- It just '**ticks**' quietly as the program runs
- The **Interval** property can used to set the 'ticking' interval to a number of milliseconds

  n.b. 1000 milliseconds = 1 second
- The **Enabled** property can be set **True** or **False** to stop or start the timer
- The Timer's **Tick()** Method can have code attached to do something each 'tick'
- e.g. move an image every 100 ms
  this allows you to do … animation!

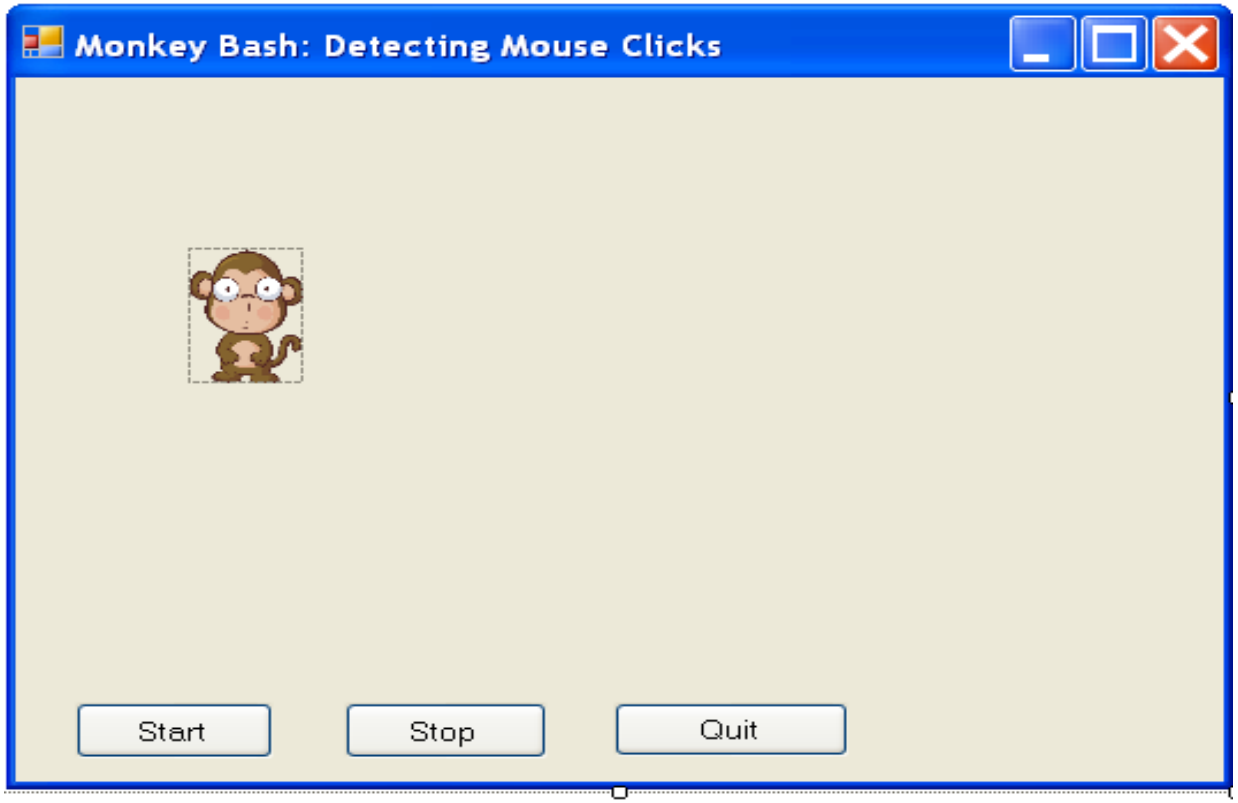
A note on Graphics Coordinates



Note: Objects have a Top, Bottom, Left, Right, Width, Height.
Coordinates start from 0,0 at the top left corner

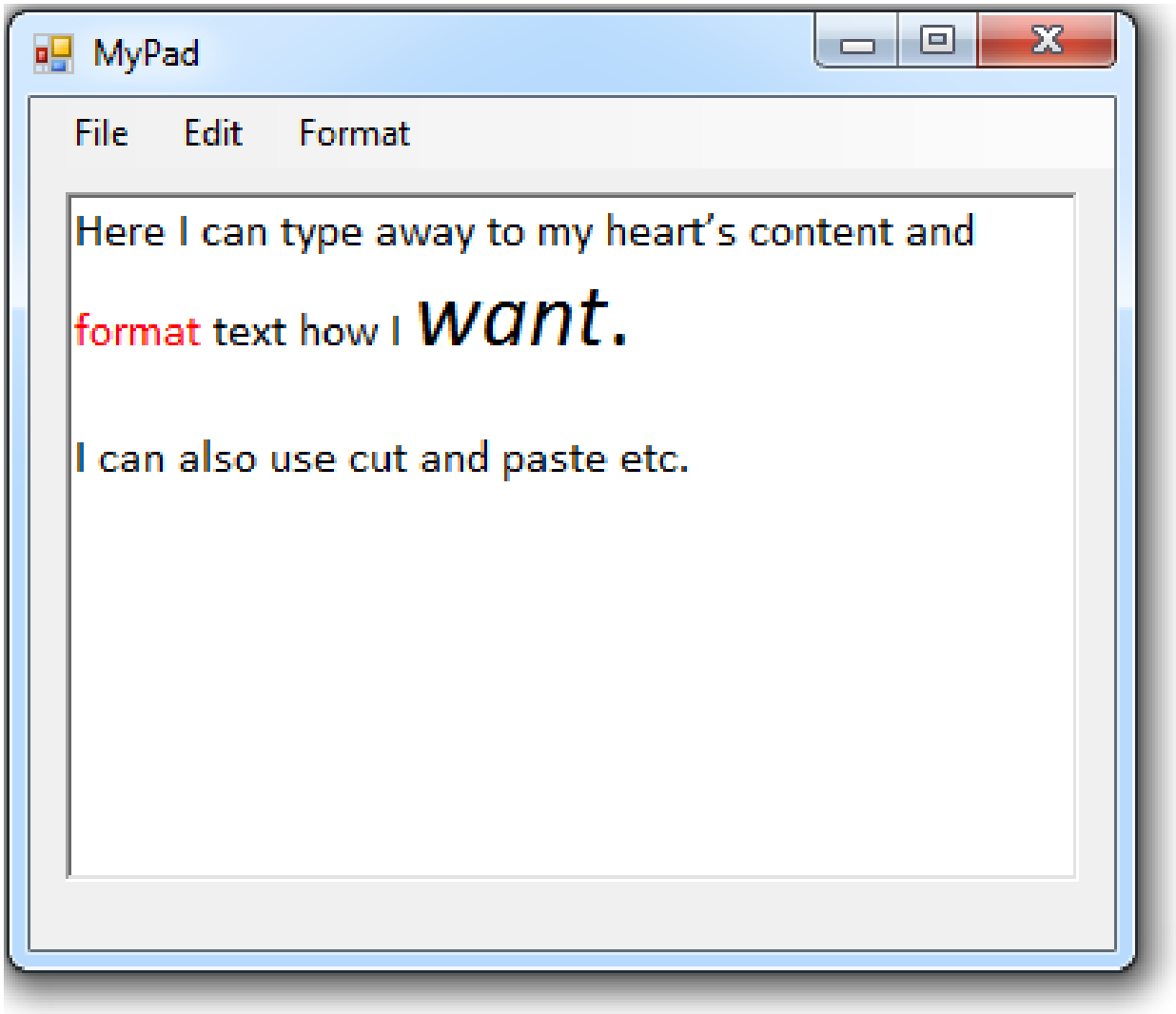
Activity

- Finish Task 3.5 (open MonkeyBash) and see how the timer works



RichTextBox

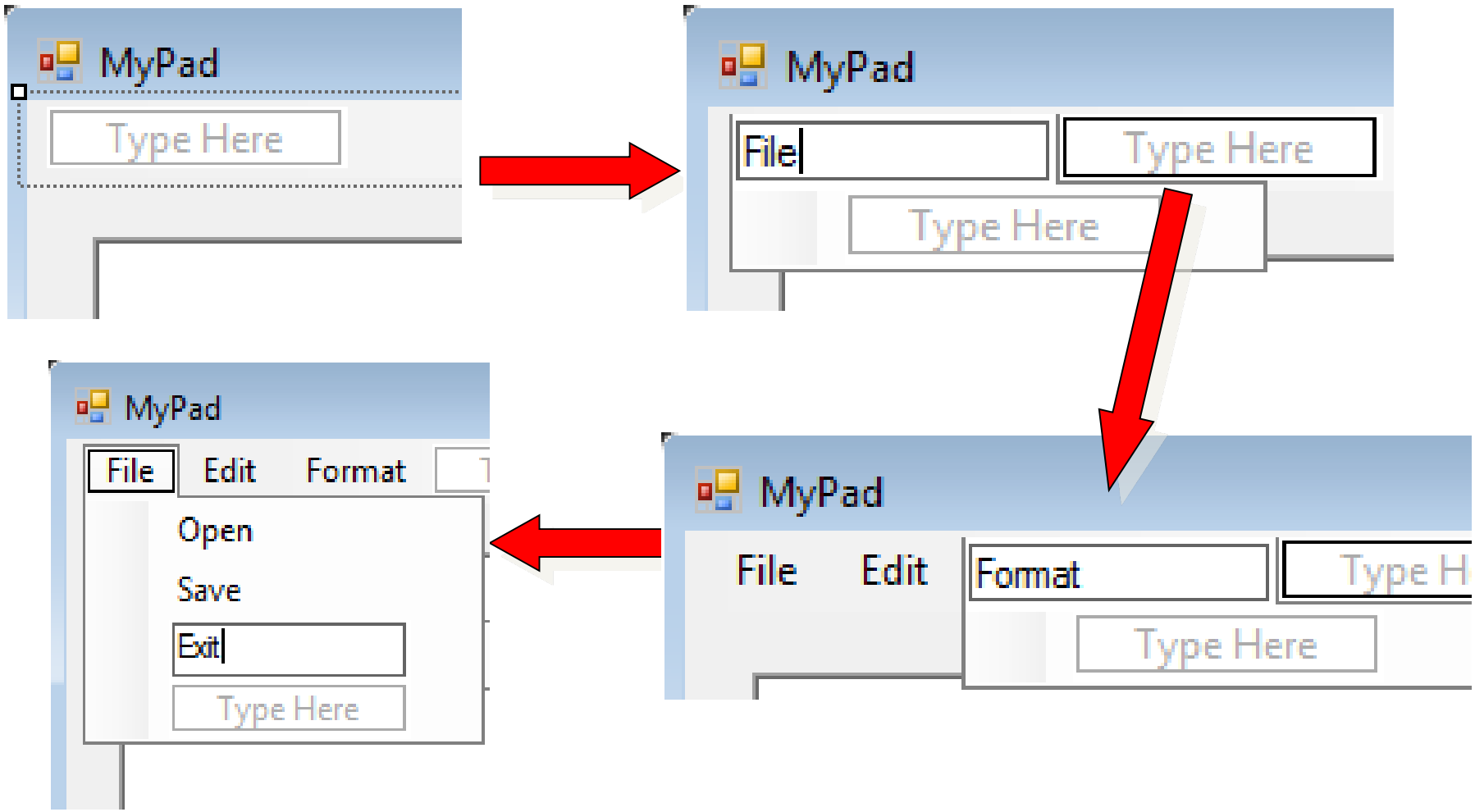
Adding a RichTextBox to a Form



For large amounts of formatted text

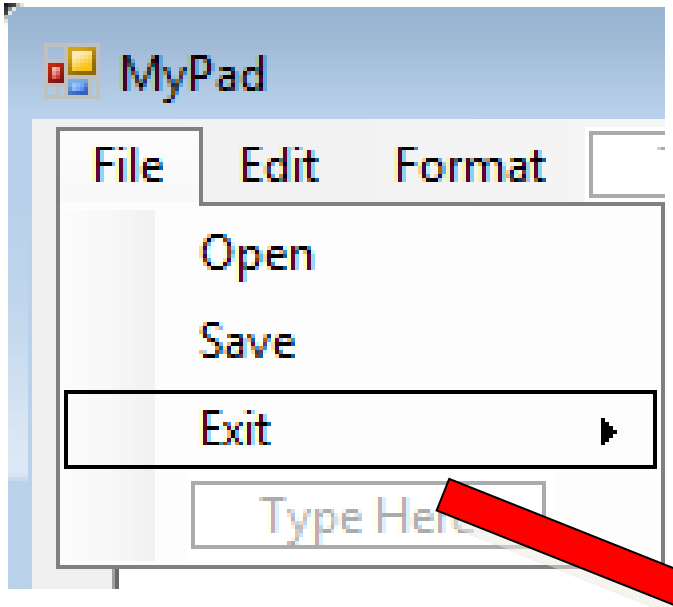
MenuStrip

Adding a MenuStrip to a Form



Add the menu items that you need, step by step

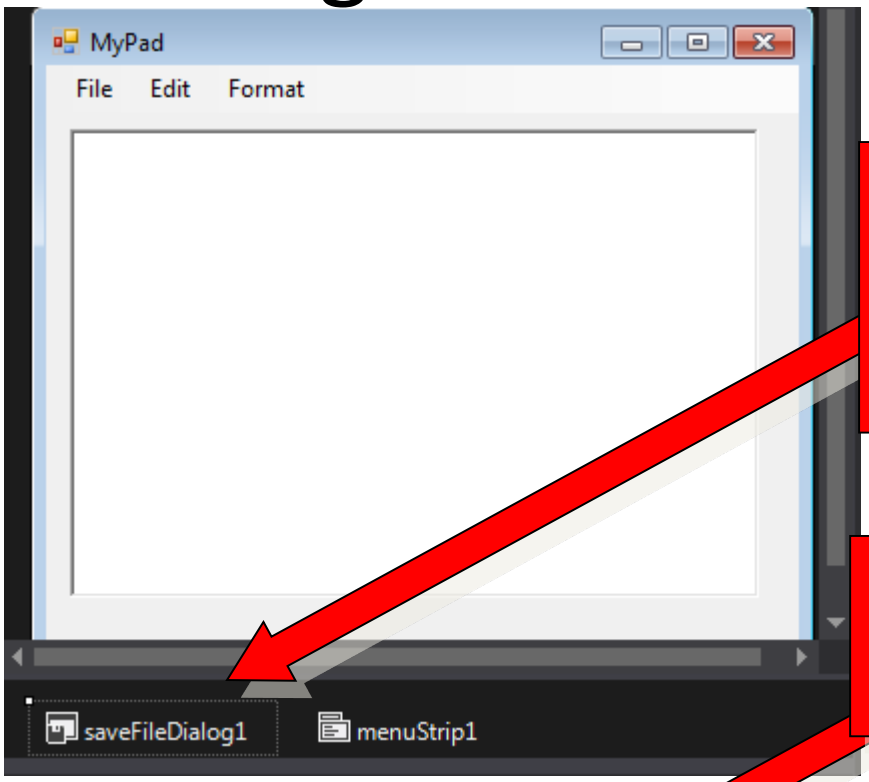
Coding a Menu Item



Double-click the item
to get to the code window

```
private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
        Application.Exit();
}
```

Using a Common Dialog for Save



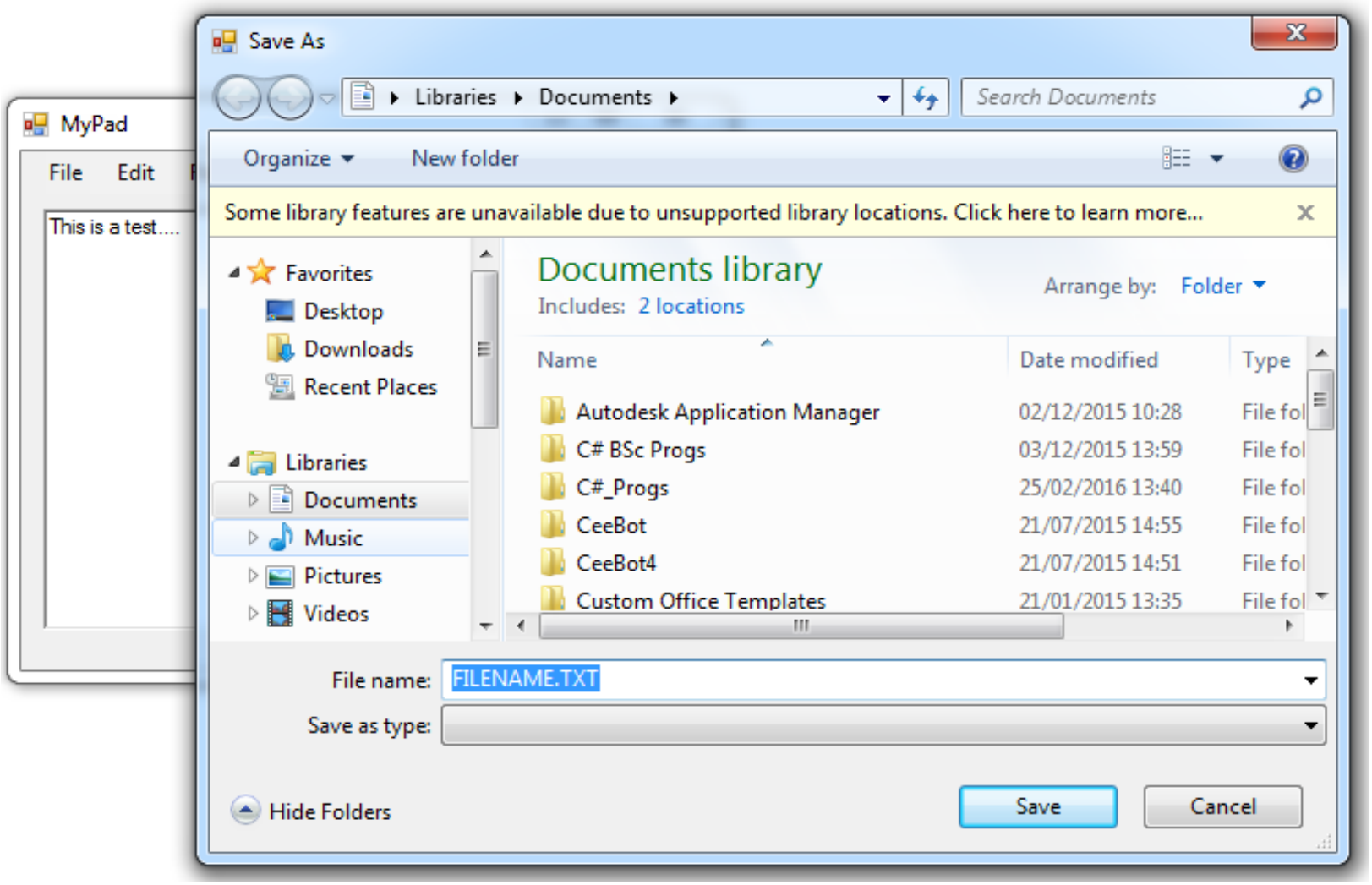
add a **SaveFileDialog** from the Toolbox

- then rename it **sfd**

then double-click on **Save** in the File menu

```
private void saveToolStripMenuItem_Click(object sender, EventArgs e)
{
        sfd.ShowDialog();
        txtMain.SaveFile(sfd.FileName)
}
```

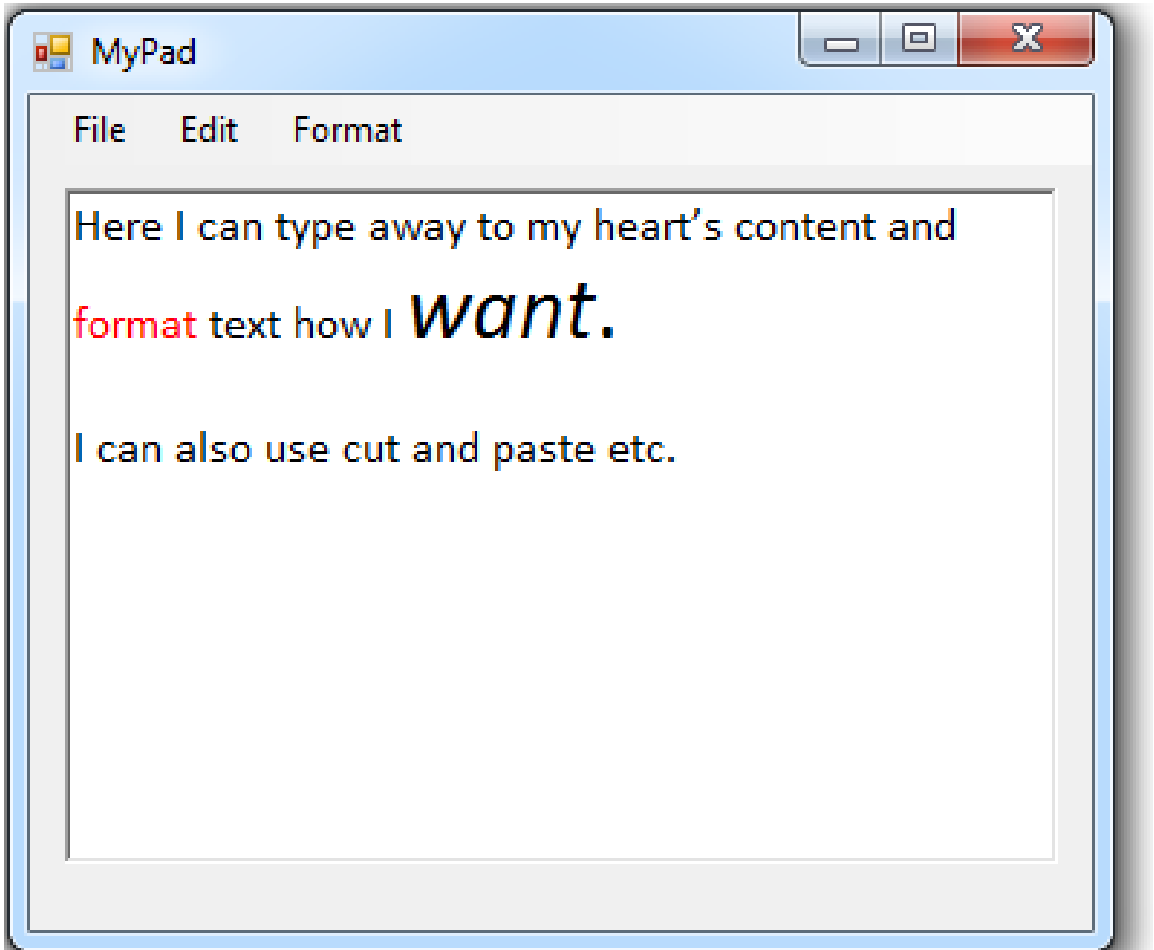
The Result



A Powerful Common Dialog for Save

Activity

- Make a start on Task 3.7 - Task 3.9



The Last Slide