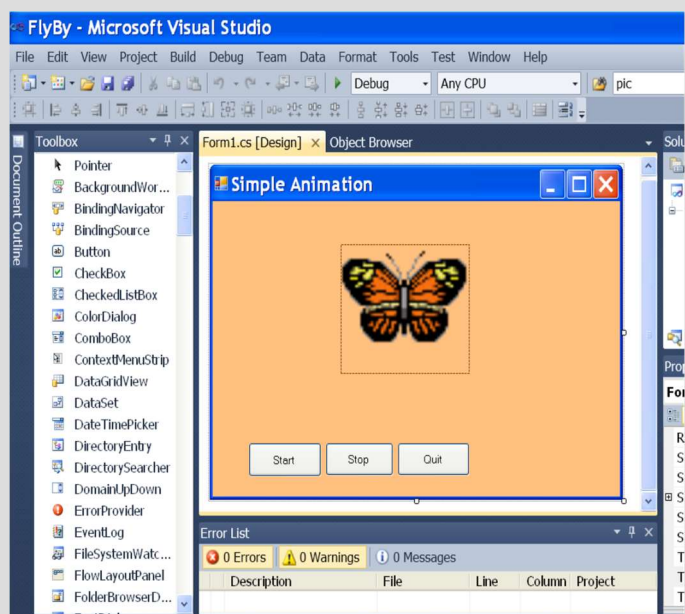


Windows Programming



.NET Windows Programming Using C#



Application Programming (CO453)

Part C – Weeks 9-13

Using MS Visual Studio C#

A beginner's guide

Starting C#

- First get into Windows
- Double-click the **Visual Studio** icon on the desktop (or use Start/programs)
- The first time it is run, you will need to choose the **Visual C# Development Settings**

Starting a NEW Project

- Select **File** from the menubar, followed by **New** then **Project**
- Select a template: **Console Application**
- Click **Browse** to find suitable **Location** for storing your project
- Enter a **Name** for your new project, then click **OK**
- Now you can type your program code into the new window

Opening an EXISTING Project

- Select **File** from the menubar, followed by **Open** followed by **Project**
- In the **Open Project** dialog window, select the correct folder and **.csproj** file
- Click **Open** and the project should be loaded together with any source code **.cs** files
- To see the code, you may have to click the **.cs** file in the Solution Explorer window

Compiling a Program

- Make sure the project and program are open
- Select **Build** from the menubar, followed by **Build Solution** (or use **F6** key)
 - Edit any syntax errors that show up in the bottom of the editor window

Running (executing) a Program

- Make sure the project and program are open
- Select **Debug** from the menubar, followed by **Start Without Debugging** (or use **Ctrl + F5**)
- If the program doesn't work as expected, you have logical errors and have to correct these.

Saving a Program

- Choose **File** from the menubar, followed by **Save All**
 - This saves to the location previously chosen

Copying a Project

- If you want to take a copy of a project so that you can make changes while keeping the original, go to where the project has been saved and copy the whole project folder
 - This folder holds the many files required by the project .. paste it to a new location

Closing down Visual C#

- Select **File** from the menubar, followed by **Exit**

Running a Program without Visual Studio

- Open the project folder
- Open its **bin** folder
- Open its **debug** folder
- Find the **exe** file for the project .. you can copy and paste this to a different area and double-click this to run your project without using Visual Studio.

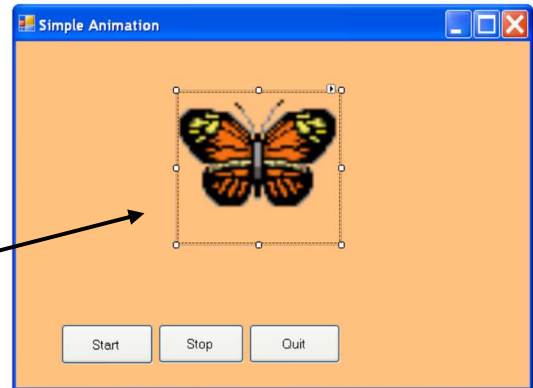
**C# Windows
Programming
Unit 5

Animation**

5.1 Timer Control Animation

A **Timer** control can be set to **tick** at a regular rate from a thousandth of a second upwards. Each timer **Tick** event can be used to trigger some action, such as a moving a graphic, picking up the system time, etc.

- Open the existing Windows application called **FlyBy** which has a PictureBox (**pbxBfly**) and 3 buttons
- This project has 8 butterfly images **bfly0.gif** to **bfly7.gif** to be animated : double-click on the form and look at the code that has been added to run when the Form loads:



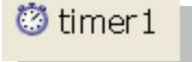
```
private void Form1_Load(object sender, EventArgs e)
{
    // store all 8 images in an array called images[]
    for(int i = 0; i < MAX; i++)
    {
        images[i] = Image.FromFile("bfly" + i + ".gif");
    }
}
```

- Note that earlier in the program **MAX** and the **images[]** array have been defined:


```
const int MAX = 8; // there are 8 images
Image[] images = new Image[MAX]; // set up an array to hold them
```

Animating the Butterflies

You have seen that when the form loads, all 8 butterfly image files are stored in the **images[]** array. What we want to do now is add a **timer** and get this to display each picture in turn in the **pbxBfly** PictureBox

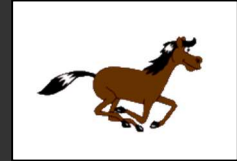
- Add a **timer** control to the form and set its **interval** property to 100 (so it ticks every 100 ms) 
- Add code to **btnStart** and **btnStop** to start and stop the timer .. this can be done using the code: **timer1.Enabled = true;** (or false)
- Now double-click the timer to enter the **timer1_Tick()** method.
- Add code to put a different image into **pbxBfly** each time the clock ticks:

```
private void timer1_Tick(object sender, EventArgs e)
{
    pbxBfly.Image = images[count]; // get an image from the images array
    count ++; // move on to the next image
}
```

- You will also need to initialise the **count** variable to zero (do this at the top, near the MAX definition: **int count = 0;**)

Task 5.1: Animated Horses

1. First complete the **FlyBy** project above so that all 8 butterflies are displayed in turn when the Start button is clicked .. note that you get an **error** when the count reaches 8 (why?)
2. Add some code to the **timer1_Tick()** method to avoid this error by resetting the count to zero when it reaches 8
3. Get the animation to run smoothly until the Stop button is clicked
4. Now get the animation to work with 8 horse Images:
 - These are in the **Images** folder (horsey0.gif etc.)
 - copy these images to the FlyBy **Debug** folder (in the **bin** folder) .. the butterfly images are also there
 - Modify the code to work with these horsey images

**5.2****Using a Timer to update a Time Display**

- Add a **Panel** to the **FlyBy** form and put 2 labels into the panel, naming them **lblDate** and **lblTime**. These will show the Date and Time respectively
- Move the Panel to the top right of the form and change its **BackColor**
- Add **another** timer and set its **interval** property to 100, **Enabled** property to True
- Now double-click this timer and add some code to its **timer2_Tick()** method:

lblDate
lblTime

```
lblDate.Text = DateTime.Now.ToLongDateString(); // pick up date and convert
lblTime.Text = DateTime.Now.ToLongTimeString(); // pick up time and convert
```

Note: **DateTime.Now** gets system Date and Time and converts them to appropriate strings before displaying.

- Run the program and see the current Date and Time, updated every 0.1 seconds

11 September 2007
13:42:47

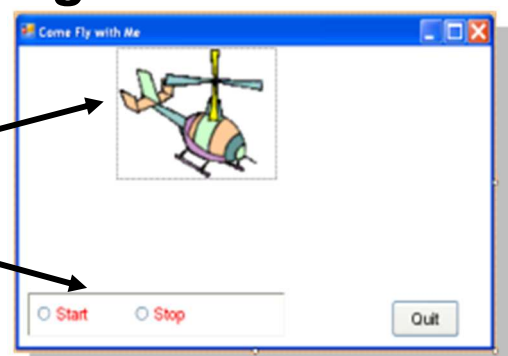
Task 5.2: Date and Time

1. Complete the above task with 2 timers and a working Date and Time Display

5.3**Using a Timer to Move an Image**

We shall now use a timer to move an image in a PictureBox across the form.

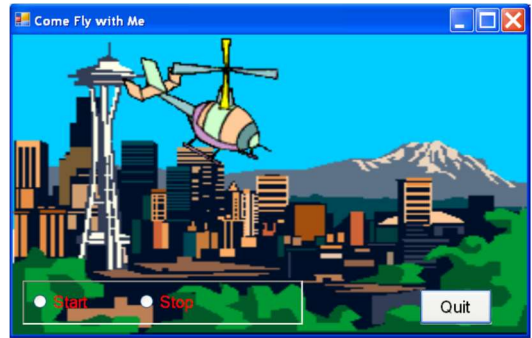
- Open and run the existing Windows application called **Animation**
- Notice it has a PictureBox (**pbxMoving**) and 2 radiobuttons in a Panel
- Look at the code and see how 4 images **copter0.gif** to **copter3.gif** are being animated using a timer control (as you have done before)



Setting a Background Image

Lets improve the scene by adding a picture:

- Select the Form and its **BackgroundImage** property.
- Import an image from the **Town and Sky** folder (in the images folder) .. try **City1.wmf**
- Make sure the **BackgroundImageLayout** property is set to **Stretch**
- Run the program again and see the difference
- Now set the **WindowState** property to **Maximized** so your form fills the screen!
- You may now want to change the **Anchor** properties of the Panel and the Quit button to anchor these to **Bottom, Left** and **Bottom, Right** respectively.



Moving the Helicopter PictureBox

Now we shall try to move the **pbxMoving** PictureBox across the screen .. a little bit for each tick of the timer.

- Double-click the **Timer** and enter some more code to move the Left side of the PictureBox by 10 pixels:

```
pbxMoving.Left = pbxMoving.Left + 10 ; // add 10 to the Left coordinate
```

- Now run the program and you should see the image move across the screen .. every tick of the timer adds 10 to the Left coordinate (coordinates start at 0, 0 at top left of the form)
- Note: the helicopter disappears off the right side of the form, never to return!

Getting the PictureBox to reappear at the Left

We can get the helicopter to reappear at the left of the form when it reaches the right.

- Add the following code to the **timer1_Tick()** method:

```
if (pbxMoving.Left >= this.Width) // has picture left side reached edge of this  
form?  
{  
    pbxMoving.Left = 0 ; // set Left coordinate of picture to zero  
}
```

- Now try again and see if it works

Task 5.3: Moving Helicopter

1. Complete the task above so the helicopter moves across a full screen background and reappears at the left side
2. Add some more code to the **timer1_Tick()** method to get the picturebox to steadily move down the screen each time it reappears on the left.
Hint: use a similar method to the one used to move from left to right ... you will need to check whether the **Top** of the picturebox has reached **this.Height** (i.e. the bottom of the form)
3. Try altering the timer1 interval property to get the animation to move faster
4. Extra : How can you set the rotation of the helicopter to a different speed from the movement across the screen?

5.4 Changing the Background Image

We shall now get the background picture to change to a new image (**City2.wmf**) when the helicopter reaches the edge of the form.

- We need to add some code to the **timer1_Tick()** method to use a new image as the Background. The way to do this successfully is to first create a new **Bitmap** from the required image file:

```
Bitmap bm = new Bitmap("City2.wmf");    // create a new bitmap called bm
this.BackgroundImage = bm;             // use bm as the background image
```

- You need to add this code in the **correct** place in the existing code .. can you see where it should go?

Task 5.4: Changing Backgrounds

- Complete the task above so the background image changes as the helicopter reaches the right side
- Now see if you can add more code to get the background to **alternate** between **City1.wmf** and **City2.wmf** so that the background changes every time the helicopter reaches the edge

5.5 Controlling Speed with a ScrollBar

We shall now add a horizontal **ScrollBar** to alter the speed of our helicopter.

- Find the **HScrollBar** in the ToolBox and add one to the Panel, underneath the two radiobuttons. Name it **hsbSpeed**
 - The ScrollBar **Value** property ranges from 0 to 100 depending on the position of the slider.
 - We can use this Value to alter the **Interval** of the timer to speed up or slow down the animation (note: smaller Interval = faster animation!)
- Double-click the ScrollBar and add code to the **hsbSpeed_Scroll()** method to change the Interval of the Timer:

```
int amount = hsbSpeed.Value + 1;    // get the Value and add 1 to avoid
zero
```

Task 5.5: Speed Control

- Get the scrollbar to work properly

Your Log Book

- In your logbook you should include the following for the **FlyBy** and **Animation** Projects:
 - Headings and task summaries
 - Screen shots of the running programs
 - Code taken from the **Form1.cs** commented with your **name**, **date** and **project details**

Unit 5: Independent Study

Task 5.6: Flying Pigs

1. Add another PictureBox and use the 4 **Pigs** images Pig0.gif to Pig3.gif (in the images folder) to animate a flying Pig alongside your helicopter.
2. Get the Pig to move across the screen like the helicopter.



Task 5.7: More Backgrounds

1. Find a third background picture to use in your project
2. Get the project to cycle between 3 backgrounds instead of 2