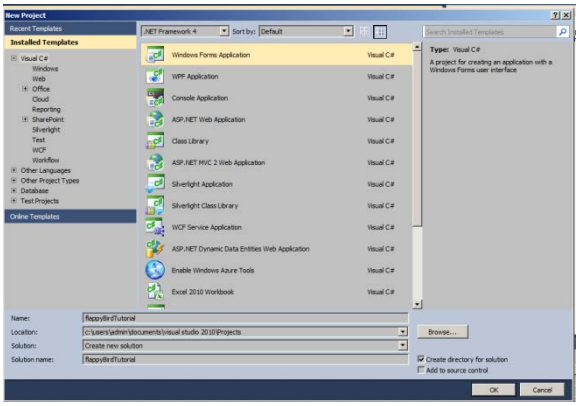
Creating a Flappy Bird game in Visual Studio Using C#.

First Create a new Windows Form Application in Visual Studio



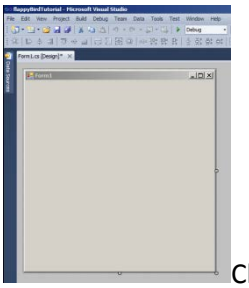
Name it FlappyBird and Click OK

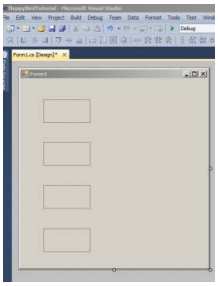
We need to add the necessary elements to the windows form

4 Picture boxes, 1 Timer Object and 4 Labels

Its important to name them correctly because we will call them in the code.

its easier to call bird.Left and pipeBottom.Left than it is to pictureBox1.Left and PictureBox2.Left.


Change the form size in the properties window to **513,640**



Add four picture boxes to the form.

Now name them accordingly.

1) **flappyBird**

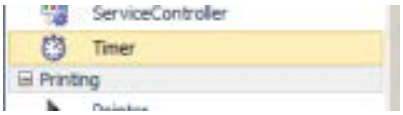
2) **pipeTop**

3) **pipeBottom**

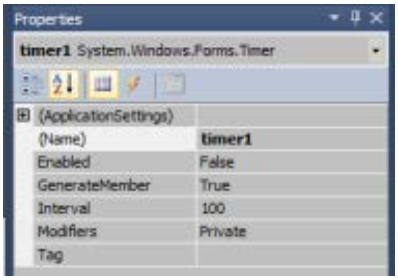
4) **ground**

Now add the timer to the form. Its under the components tab in the tool box

more tutorial **www.mooict.com**



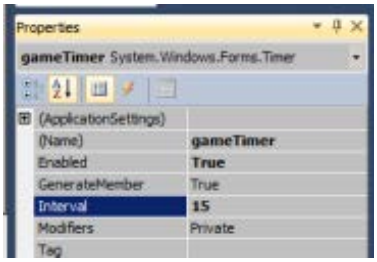
Drag and Drop on the Form



<- This is the default properties of the timer

Click on the timer1 and add the following properties to it -

name = **gameTimer** Enabled = **True**, Interval = **15**



We need a timer to animate the objects, check for collision and to know when to end the game.

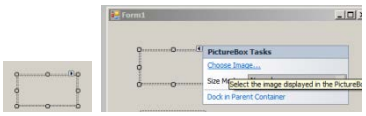
now to add the pictures in the right picture boxes

here are the pictures for flappy bird

resources available in the zip file on above this tutorial.

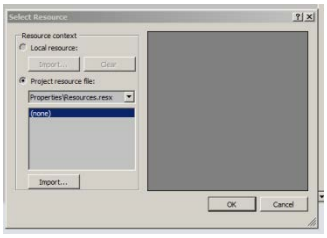
Each of these images are in PNG format and they contain transparency.

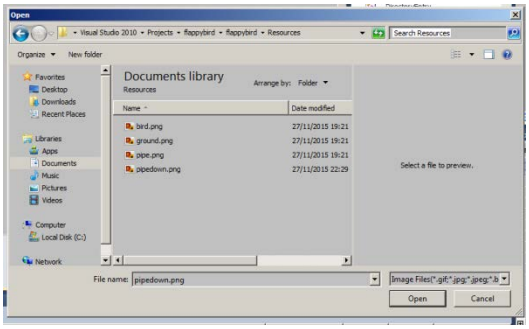
Click on the first picture box and Click on the white triangle on top > choose images



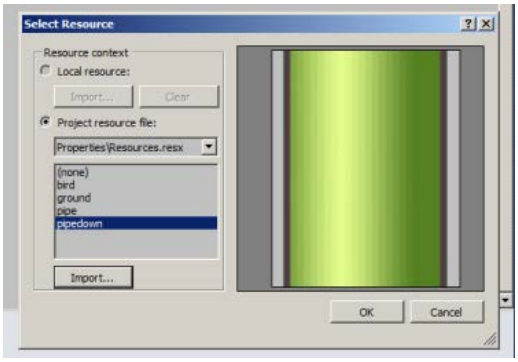
import the pictures to the resources

more tutorial **www.mooict.com**

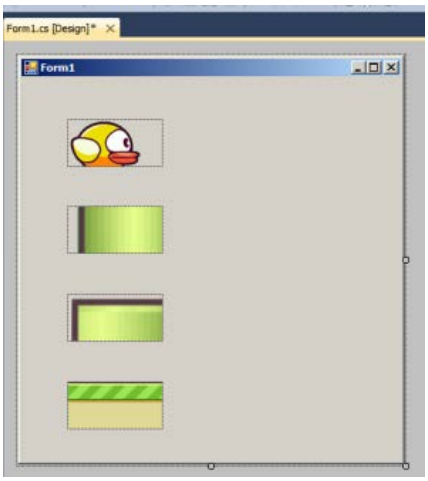




Select them all and click on open



Now go through each picture and set them to their appropriate spots.

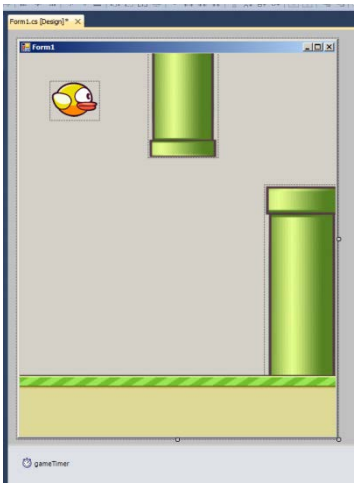


We need to make some adjustments to the form to suit the game.

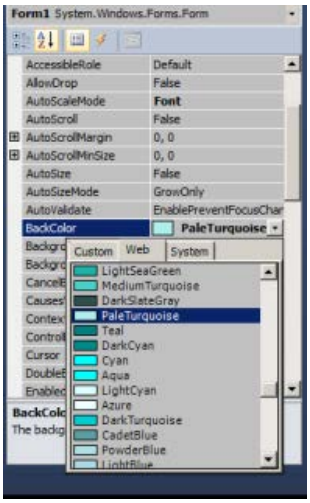
Make the form bigger to see more of the game level.

Adjust the images to and lay it out as followed

You can use stretch image setting in the size mode to fit the pictures in the picture box.



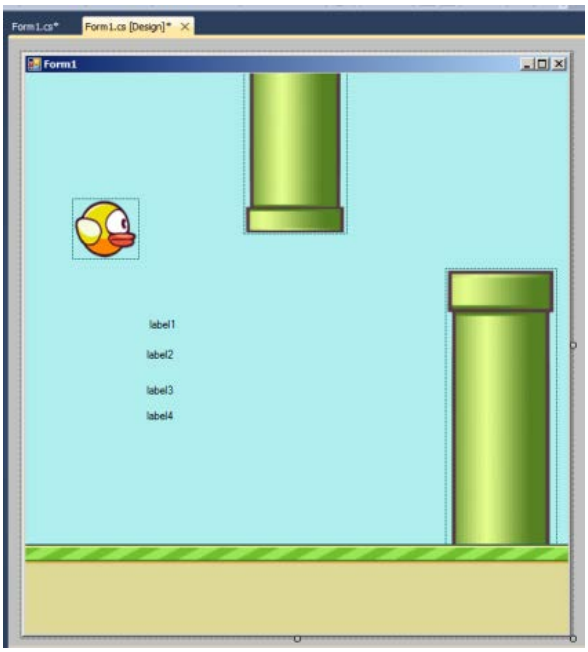
Lastly change the background colour on the form to blue



Now add four labels to the game

1 label for showing score and 3 other labels for end game credits.

We will hide the end game credits till the game ends.



name the following labels

more tutorial **www.mooict.com**

label1 change to **scoreText**

label2 change to **endText1**

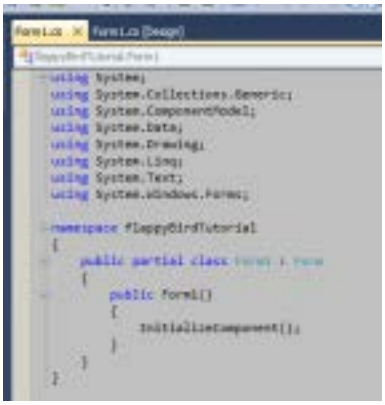
label3 change to **endText2**

label4 change to **gameDesigner**

you can format these labels to suit any colour, font or size you choose.

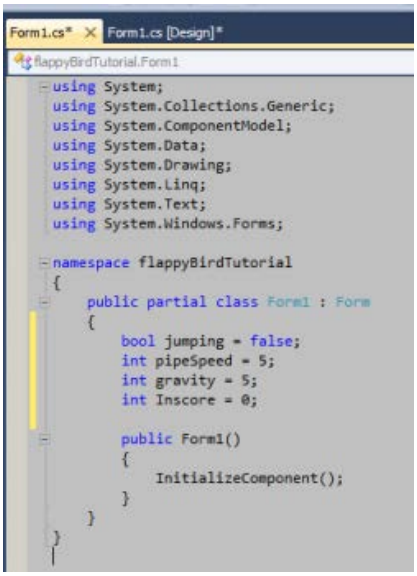
Go to the code view of the form

Right click on the form click on view code

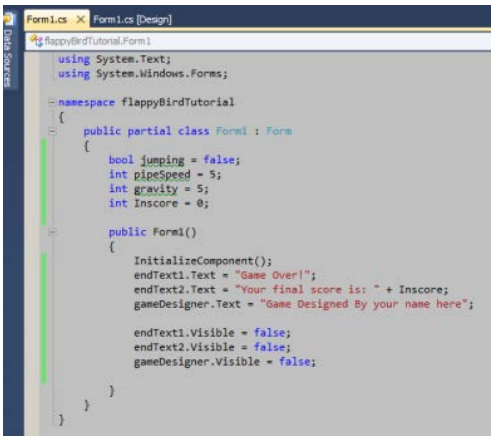


Set the core variables for the game

```
bool jumping = false;

intpipeSpeed = 5;

int gravity = 5;

intInscore = 0;
```



set the end screen labels and set them to invisible for now.



Note - do the following instructions under the initializeComponent();

```
endText1.Text = "Game Over!";

endText2.Text = "Your final score is: " + Inscore;

gameDesigner.Text = "Game Designed By your name here";

endText1.Visible = false;

endText2.Visible = false;

gameDesigner.Visible = false;
```

We need 3 functions that will be triggered by various events

1) Timer function

2) keydown function

3) key up function

4) game end function

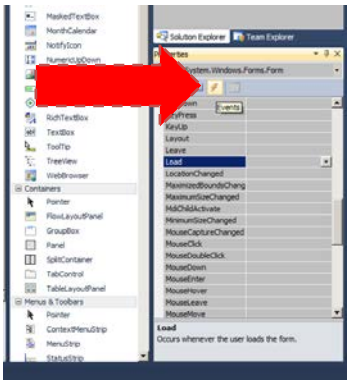
1) Add the timer function

It's simple just double click on the **gametimer**(timer which we added earlier) and it will automatically add the function code

<-- double click this. Visual studio will automatically add the code for this event, we will need add our game logic inside this event.

2) Add the key down function

click on the form and click on the event button in the properties window



Find the **key down** option and type **inGameKeyDown**, press enter

Do the same for 3) key up function

Find the **key up** option and type in **GameKeyUp** press enter

Last create a function to be used when we want to the end the game.

```
private void endGame()
{
}
```

Check the screen shot below

```
namespace flappyBirdTutorial
{
    public partial class Form1 : Form
    {
        bool jumping = false;
        int pipeSpeed = 5;
        int gravity = 5;
        int Inscore = 0;

        public Form1()
        {
            InitializeComponent();
            endText1.Text = "Game Over!";
            endText2.Text = "Your final score is: " + Inscore;
            gameDesigner.Text = "Game Designed By your name here";

            endText1.Visible = false;
            endText2.Visible = false;
            gameDesigner.Visible = false;

        }

        private void gameTimer_Tick(object sender, EventArgs e)
        {

        }

        private void GameKeyDown(object sender, KeyEventArgs e)
        {

        }

        private void GameKeyUp(object sender, KeyEventArgs e)
        {

        }

        private void endGame()
        {

        }
    }
}
```

All the animations are done through the gametimer function.

Character movements will be done through the key down and key up.

Starting with the game timer:

Inside the **gameTimer_Tick** function enter the following code

```
pipeBottom.Left -= pipeSpeed;
```

more tutorial **www.mooict.com**

```
pipeTop.Left -= pipeSpeed;

flappyBird.Top += gravity;
```

This will scroll the pipes to the left and drop the bird using the gravity variable.

Each tick moves the picture boxes to left according to the speed we set in the pipe speed variable.

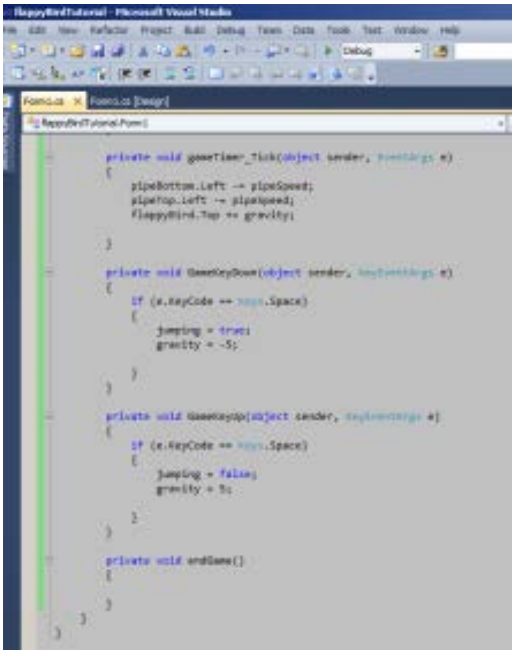
Enter the following code in the keydown function, this will reverse the gravity and make the character jump.

```
if (e.KeyCode == Keys.Space)

            {

jumping = true;

gravity = -5;

            }
```

enter the following code in the key up function, this will enable gravity again to the character

```
if (e.KeyCode == Keys.Space)

{

jumping = false;

gravity = 5;

}
```

Now if you test it you will see the animations and key board controls happening.



Problems:

There is no way the game ends.

Pipes scroll to the side and disappear

more tutorial **www.mooict.com**

Scoring doesn't work.

We solve these problems using collision detection in C#.

Every picture box has a property called bounds.

Enter the following code in the timer function

```
if (flappyBird.Bounds.IntersectsWith(ground.Bounds))
            {
endGame();
            }
elseif (flappyBird.Bounds.IntersectsWith(pipeBottom.Bounds))
            {
endGame();
            }
elseif (flappyBird.Bounds.IntersectsWith(pipeTop.Bounds))
            {
endGame();
            }
```

Bounds check for height and width of each of the picture box. Intersects with will check the height and width of another picture against the first one and check to see if they are colliding.

Once the program determines that picture boxes are colliding with each other, we will end the game.

Add the following inside the end game function

```
gameTimer.Stop();
```

This code will manually stop the timer from running.

The timer controls everything in the game. If we stop that we can stop the whole game.

now test the game.

Second problem are the pipes animating only once.

Enter the following code in the game timer function

```
if (pipeBottom.Left< -80)
            {
pipeBottom.Left = 1000;
Inscore += 1;
            }
elseif (pipeTop.Left< -95)
            {
pipeTop.Left = 1100;
Inscore += 1;
            }
```

The code above checks whether the pipes have left the screen and gone beyond -80px to the left. If it has then we change the pipes left position to the far right on the screen which creates an illusion of animation and keeps the game going. Instead of creating and recreating the objects we recycle the same one over and over.

After the pipes left the screen we increase the points by one.

Problem 3.

To show the score on screen while playing enter the following under the **flappyBird.Top += gravity;**
more tutorial **www.mooict.com**

```
scoreText.Text = "" + Inscore;
```

Test the game out.

Now to end the game.

Enter the following code inside the end game function

```
endText1.Visible = true;
endText2.Visible = true;
gameDesigner.Visible = true;
```

Play the game.

You can do the following:

Add your own obstacles

Increase the speed

Add a restart button

etc

Full Code:

```
using System;
using System.Collections.Generic;

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace flappyBirdTutorial
{
publicpartialclassForm1 : Form
    {
bool jumping = false;
int pipeSpeed = 5;
int gravity = 5;
int Inscore = 0;

public Form1()
        {
InitializeComponent();
            endText1.Text = "Game Over!";
            endText2.Text = "Your final score is: " + Inscore;
            gameDesigner.Text = "Game Designed By your name here";
            endText1.Visible = false;
            endText2.Visible = false;
            gameDesigner.Visible = false;

        }

private void gameTimer_Tick(object sender, EventArgs e)
        {
      pipeBottom.Left -= pipeSpeed;
      pipeTop.Left -= pipeSpeed;
      flappyBird.Top += gravity;
      scoreText.Text = "" + Inscore;


if (pipeBottom.Left < -80)
            {
pipeBottom.Left = 1000;
Inscore += 1;
            }
elseif (pipeTop.Left< -95)
            {
pipeTop.Left = 1100;
Inscore += 1;
            }

if (flappyBird.Bounds.IntersectsWith(ground.Bounds))
            {
endGame();
            }
elseif (flappyBird.Bounds.IntersectsWith(pipeBottom.Bounds))
            {
endGame();
            }
elseif (flappyBird.Bounds.IntersectsWith(pipeTop.Bounds))
            {
endGame();
            }
        }

private void GameKeyDown(object sender, KeyEventArgs e)
        {
if (e.KeyCode == Keys.Space)
            {
jumping = true;
gravity = -5;
            }
        }

private void GameKeyUp(object sender, KeyEventArgs e)
        {
if (e.KeyCode == Keys.Space)
            {
jumping = false;
gravity = 5;
            }
        }

private void endGame()
        {
            gameTimer.Stop();
            endText1.Visible = true;
            endText2.Visible = true;
            gameDesigner.Visible = true;
        }
    }
}
```