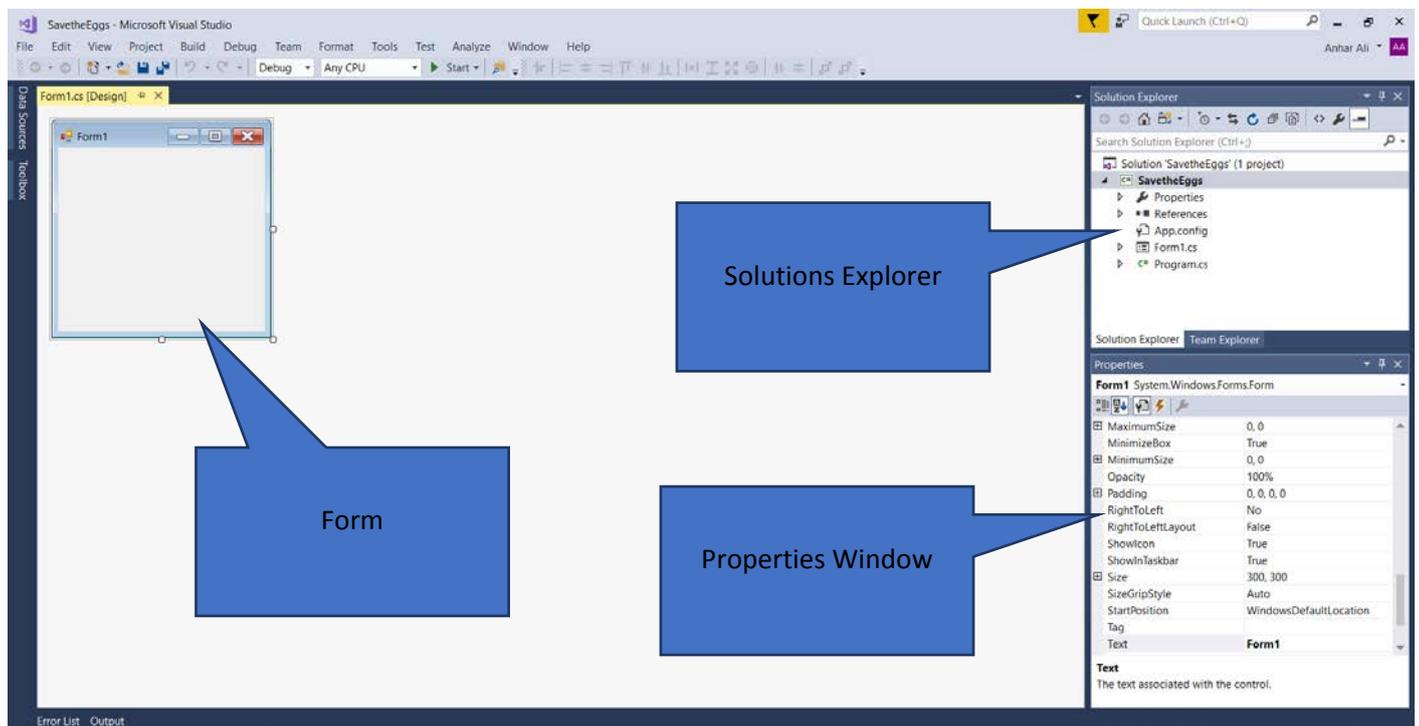
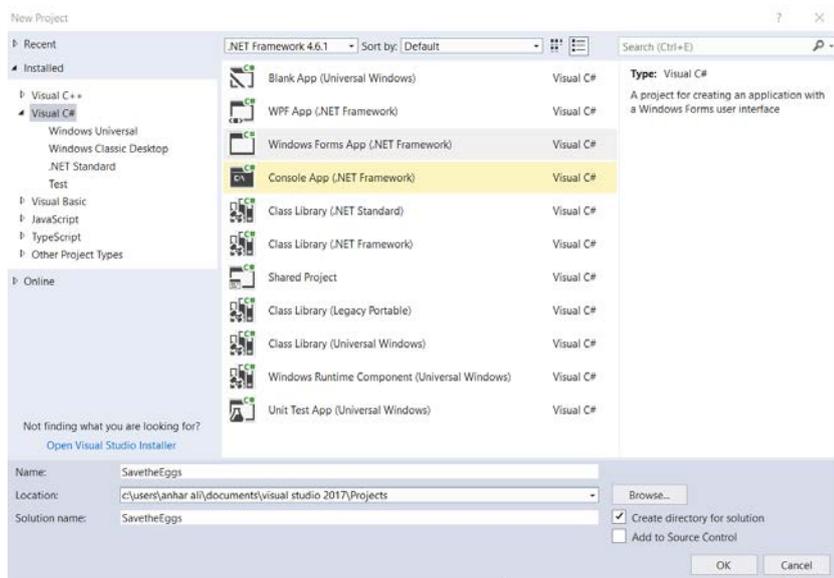


## C# Tutorial – Create a Save The Eggs Item Drop Game in Visual Studio

Start Visual Studio, Start a new project. Under the C# language, choose Windows Form Application. Name the project **savetheeggs** and click ok.



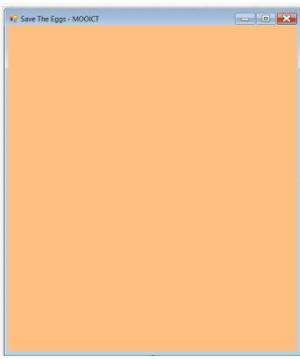
This is the new empty form

Now while the form is selected, go to the **properties window** and change the following settings for the form.

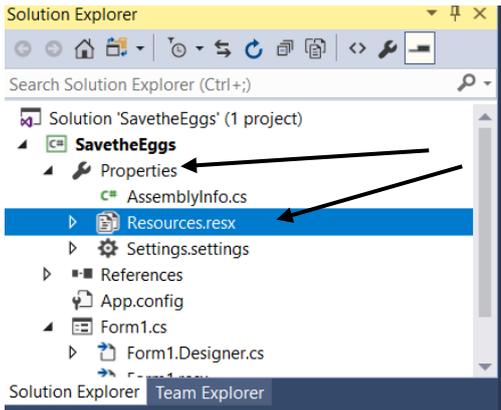
Back Color - **255, 192, 128** -- this is a dark orange colour for the game.

Size - **628, 741**

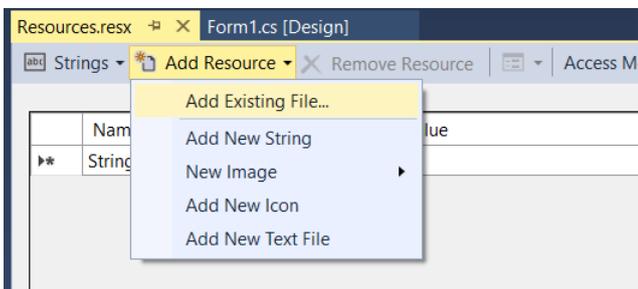
Text – **Save the Eggs MOOICT**



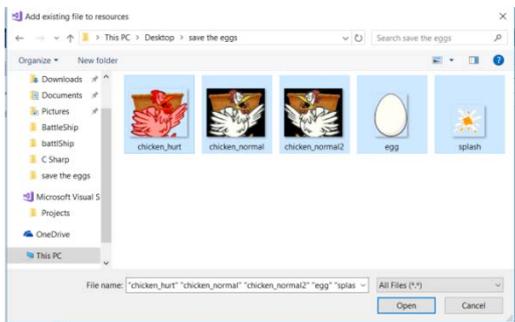
This is the result. This will be our game canvas. We will need to add the following components to this form. Before we begin adding the components we need to import our resources to this project.



Under the solutions explorer, expand the properties option and you see resources. Now double click on resources window.



From the Add Resource Drop down click on Add Existing File



Select all the images and click open



Now all the game assets been imported to the project.

Make sure you save the project at this point.

Now lets go back to adding components to the game.

Open the Tool Box and start adding the following components, if the tool box isn't present then go to View – Tool Box

Add 2  Labels to the screen

Place them top of the screen as followed

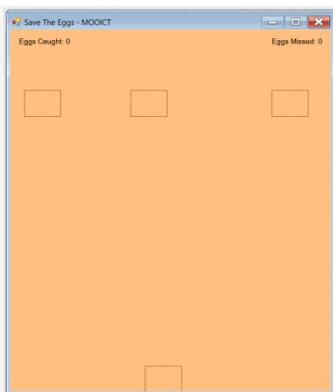


Now change the text for the following labels



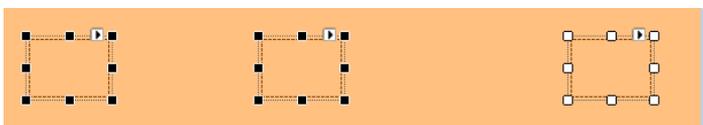
These two will be used show how many eggs we caught and how many we missed.

Now add 4 picture boxes  to the screen place them as followed



4 picture boxes been placed on the screen. The top 3 picture boxes will be the eggs and bottom picture will be the chicken.

Select the all 3 picture boxes from the top. You can drag and select all 3

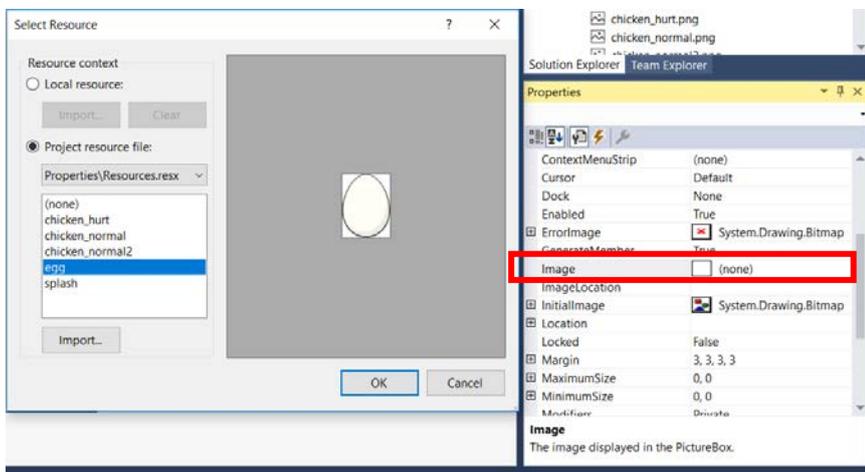


Look in the properties window and change the following

**Size Mode – Stretch Image**

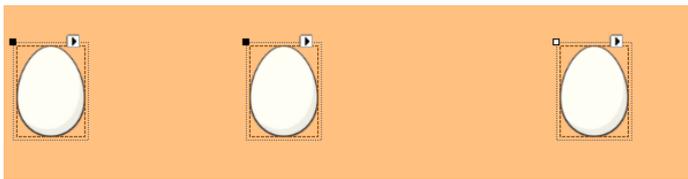
**Tag – Eggs**

Under the **Image** option click on the 3 dots and this window will come up



**Size - 54, 66**

Pick the egg image and click OK.



Yep them are some tasty looking eggs, let's move on.

Now for the chicken. Click on the bottom picture and change the following to the properties window.

**(Name)** – chicken

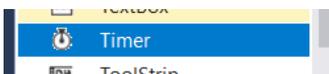
**Image** – chicken\_normal2

**Size** – 93, 83

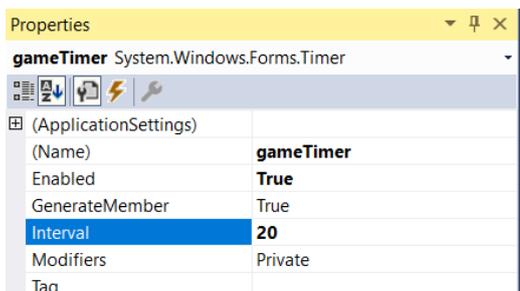
**Size Mode** – Stretch Image



**Now lets add a timer object to the game**



Drag and drop the timer to the form and change the following to its properties



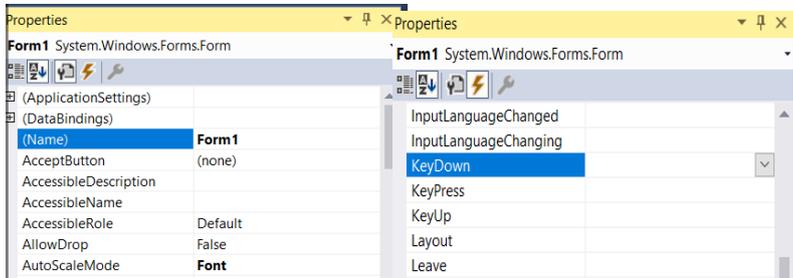
Change the name to gameTimer, set enabled to True and change the interval to 20.

## Adding Events

### Adding the Key down and Key Up event

Click on the form make sure nothing else is selected other than the form itself

In the properties window there is a lightning bolt icon, that icon means the events manager. All the events we will need to add are there, click on that image and it will bring up the



We are looking for two events, key down and key up

Inside the key down event type – **keydown** and press enter

Inside the key up event type – **keyup** and press enter

Both of these actions will take you to the code view but you can always use the top tabs to come back to the design view.



Form1.CS is the codes for the game, resources we had open before and Form1.CS [Design] is the game canvas screen

Adding event to the game timer

Now click on the game timer and in the properties window, check the events manager



You will see a Tick event, the timer only has one event. Type **gameTick** and press enter

Now we have all the events for this game.

This is what the code view looks like thus far.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SavetheEggs
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

```

private void keyisdown(object sender, KeyEventArgs e)
{
}

private void keyisup(object sender, KeyEventArgs e)
{
}

private void gameTick(object sender, EventArgs e)
{
}
}

```

We need to add a game reset function here. This function will run when the game starts and if the player wants to play the game again when the level is over.

This will be a custom function, we will have to create it ourselves. Check below

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SavetheEggs
{
public partial class Form1 :Form
{
public Form1()
{
InitializeComponent();
}

private void keyisdown(object sender, KeyEventArgs e)
{
}

private void keyisup(object sender, KeyEventArgs e)
{
}

private void gameTick(object sender, EventArgs e)
{
}

private void reset()
{
}
}
}

```

So far we have the empty functions and events. Now we need to start adding the variables and instruction to the game.

These will go before `public Form1()` line.

// this green text in the code are comments, we are using them to help you understand the code better and they are good practice to keep doing while programming so you know what you are writing in the code.

More tutorials on [www.mooict.com](http://www.mooict.com)

// each line of comment will explain what we are doing in the code, so the variables are self explained in the code.

```
bool goleft; // this boolean will be used to check if the player can move left
bool goright; // this boolean will be used to check if the player can move right
int speed = 8; // this is the default speed for the Eggs dropping
int score = 0; // this is the default score value
int missed = 0; // this is the default miss value
Random rndY = new Random(); //this will be used to generate a random Y location
Random rndX = new Random(); // this will be used to generate a random X location
PictureBox splash = new PictureBox(); // create a new splash picture box, this will added dynamically
```

Inside the public form1 function add the following line highlighted below. The initialize component is there from the beginning because its where visual studio can load up all the buttons, text boxes and form for the application. DO NOT DELETE THIS LINE.

```
public Form1()
{
InitializeComponent();
reset(); // invoke the reset game function
}
```

This is the reset function which we created earlier, with this function we can load up all of the default values and positions in the game. At this very moment its empty but we will be filling it in soon.

Lets look into the key down function

```
private void keyisdown(object sender, KeyEventArgs e)
{
if (e.KeyCode == Keys.Left)
{
// if the left key is pressed change the go left to true
goleft = true;
}
if (e.KeyCode == Keys.Right)
{
// if the right key is pressed change the go right to true
goright = true;
}
}
```

This is the key down function as you can see we are checking if the player has pressed the LEFT or RIGHT key and if they did we change the go left or go right Boolean to true.

In the key up function it's the reversal of these actions, lets check it out

```
private void keyisup(object sender, KeyEventArgs e)
{
if (e.KeyCode == Keys.Left)
{
// if the left key is up then change the go left to false
goleft = false;
}
if (e.KeyCode == Keys.Right)
{
// if the right key is up then change the go right to false
goright = false;
}
}
```

So, if you look closely, we are once again checking for the left and right key only this time this event will fire up when the left or right key is up and when they are we change the go left and go right Boolean to false.

This way the chicken object won't continuously move left or right it will only move when the player is holding the button down.

Now lets take a deep look at the game tick function. Everything that happens in this game will be instructed in this function.

```
private void gameTick(object sender, EventArgs e)
{
    label1.Text = "Eggs Caught: " + score; // show the score on Eggs Caught label
    label2.Text = "Eggs Missed: " + missed; // Show the misses on Eggs Missed label

    // if the go left boolean is true AND chickens left is greater than 0
    if (goleft == true && chicken.Left > 0)
    {
        // then we move the chicken 12 pixels to the left
        chicken.Left -= 12;
        //checken image will be change to the one moving left
        chicken.Image = Properties.Resources.chicken_normal2;
    }
    //if the go right is true AND chickens width and left is less than form width
    // meaning the chicken is still within the frame of the game
    if (goright == true && chicken.Left + chicken.Width < this.ClientSize.Width)
    {
        // move the chicken 12 pixels to the right
        chicken.Left += 12;
        // change the chicken image to the one moving right
        chicken.Image = Properties.Resources.chicken_normal;
    }

    //below for loop will check the eggs dropping from the top
    // for each Control we are calling X in this form
    foreach (Control X in this.Controls)
    {
        // if X is a type of picture box AND it has the tag of Eggs
        if (X is PictureBox && X.Tag == "Eggs")
        {
            // then move X towards the bottom
            X.Top += speed;

            // if the EGGS [X] reaches bottom of the screen
            if (X.Top + X.Height > this.ClientSize.Height)
            {
                // if the egg hit the floor then we show the splash image
                splash.Image = Properties.Resources.splash; // set the splash image
                splash.Location = X.Location; // splash shows up on the same location as the egg
                splash.Height = 59; // set the height
                splash.Width = 60; // set the width
                splash.BackColor = System.Drawing.Color.Transparent; // apply transparent background to the picture
                box

                this.Controls.Add(splash); // add the splash picture to the form

                X.Top = rndY.Next(80, 300) * -1; // position the eggs to a random Y location
                X.Left = rndX.Next(5, this.ClientSize.Width - X.Width); // position the eggs to a random X location
                missed++; // add 1 to the missed integer
                chicken.Image = Properties.Resources.chicken_hurt; // change the chicken image to hurt image
            }

            // if the eggs bound with the chicken image
            // if both picture boxes collide
            if (X.Bounds.Intersects(chicken.Bounds))
            {
                X.Top = rndY.Next(100, 300) * -1; // position the eggs to a random Y location
                X.Left = rndX.Next(5, this.ClientSize.Width - X.Width); // position the eggs to a random X location
                score++; // add 1 to the score
            }

            // if the score is equals to or greater than 20
            if (score >= 20)
            {
                speed = 16; // increase the eggs speed to 20
            }

            // if the missed number is greater than 5
            // we need to stop the game
            if (missed > 5)
            {
                gameTimer.Stop(); // stop the game timer
                // show the message box to say game is over.
                MessageBox.Show("Game Over!! We lost good Eggs" + "\r\n" + "Click OK to restart");
                // once the players clicks OK we restart the game again
                reset();
            }
        }
    }
}
```



```
{  
}
```

Inside the loop we have a if statement that's checking if X is a picture box and x has a tag of EGGs. Remember we added the Eggs tags to the egg picture box earlier, this is why.

Everything we want to do with these eggs can go inside of this if statement. Remember opening and closing the curly brackets.

```
// then move X towards the bottom
```

```
X.Top += speed;
```

Firstly we want to drop the eggs from the top. Soon as the game starts, the loop will check the picture boxes, when it finds the ones has the Eggs tag it will start to add their speed downward.

```
// if the EGGs [X] reaches bottom of the screen
```

```
if (X.Top + X.Height > this.ClientSize.Height)
```

```
{
```

```
// if the egg hit the floor then we show the splash image
```

```
splash.Image = Properties.Resources.splash; // set the splash image
```

```
splash.Location = X.Location; // splash shows up on the same location as the egg
```

```
splash.Height = 59; // set the height
```

```
splash.Width = 60; // set the width
```

```
splash.BackColor = System.Drawing.Color.Transparent; // apply transparent background to the picture box
```

```
this.Controls.Add(splash); // add the splash picture to the form
```

```
X.Top = rndY.Next(80, 300) * -1; // position the eggs to a random Y location
```

```
X.Left = rndX.Next(5, this.ClientSize.Width - X.Width); // position the eggs to a random X location
```

```
missed++; // add 1 to the missed integer
```

```
chicken.Image = Properties.Resources.chicken_hurt; // change the chicken image to hurt image
```

```
}
```

The if statement above is checking if the eggs are hitting the floor, so inside the if statement where the conditions are give we have a condition that's checking if X.TOP + X.Height meaning the eggs top value and height of the object is greater than the forms height then we do the following

We put the splash image on the splash picture box

We assign the splash location to the X's location

Splash has the height of 59 and Width of 60

Setting the splash picture boxes background to transparent

```
this.Controls.Add(splash); // add the splash picture to the form
```

This line above will add the splash image to the form.

So when the eggs hit the ground and the splash image shows up we want to reset the egg back to the top so it can fall back, however we don't want the egg to fall to the same place we want to randomise the location meaning the X and Y positions.

```
X.Top = rndY.Next(80, 300) * -1; // position the eggs to a random Y location
```

```
X.Left = rndX.Next(5, this.ClientSize.Width - X.Width); // position the eggs to a random X location
```

So when it hits the ground we set the X.TOP = random Y number between 80, 300 now we need to have negative value to them for they go back to top of the screen so we are multiplying the number by -1.

For the x left axis we are randomising the rand X with minimum of 5 and size of the form width. This way it can spawn any where on the screen.

```
missed++; // add 1 to the missed integer
chicken.Image = Properties.Resources.chicken_hurt; // change the chicken image to hurt image
```

once all that is done since this was missed egg we add 1 to the missed integer and change the chickens image to the hurt image from the resources.

```
// if the eggs bound with the chicken image
// if both picture boxes collide
if (X.Bounds.IntersectsWith(chicken.Bounds))
{
    X.Top = rndY.Next(100, 300) * -1; // position the eggs to a random Y location
    X.Left = rndX.Next(5, this.ClientSize.Width - X.Width); // position the eggs to a random X
location
    score++; // add 1 to the score
}
```

In the above if statement we are checking if the egg is colliding with the chicken, if so then we once again randomise the eggs location and this time we add 1 to the score for the player

```
// if the score is equals to or greater than 20
if (score >= 20)
{
    speed = 16; // increase the eggs speed to 20
}
```

In the game we want to make it more challenging for the player, so in the if statement above we are checking if score is equals to or greater than 20 then we change the speed to 16. This means the eggs will start to fall faster.

```
// if the missed number is greater than 5
// we need to stop the game
if (missed > 5)
{
    gameTimer.Stop(); // stop the game timer
    // show the message box to say game is over.
    MessageBox.Show("Game Over!! We lost good Eggs" + "\r\n" + "Click OK to restart");
    // once the players clicks OK we restart the game again
    reset();
}
```

In the if statement above its checking if the player has missed more than 5 eggs then we stop the timer, show a message box that says the game is over and they can click OK to restart. Then we are invoking the reset function so the game starts again.

If you are unsure about anything in the function please refer back to the page and check the code from there. Make sure all the brackets and curly brackets are present.

We have one last function to write now, this is the reset function

Lets take a look below

```
private void reset()
{
    // check all of the controls with this loop
    // create a control called X and check it in the form components
    foreach (Control X in this.Controls)
    {
        /// if X is a picture box and it has a tag of Eggs
        if (X is PictureBox && X.Tag == "Eggs")
        {
            // we move it to top of the screen
            X.Top = rndY.Next(80, 300) * -1; // give it a random y location
            X.Left = rndX.Next(5, this.ClientSize.Width - X.Width); // give it a random x location
        }
    }

    chicken.Left = this.ClientSize.Width / 2; // reset the chicken to middle of the form
    chicken.Image = Properties.Resources.chicken_normal2; // change the chicken picture to face left
```

```

    score = 0; // set score to 0
    missed = 0; // set missed to 0
    speed = 8; // set speed to 8

    goleft = false; // set go left to false
    goright = false; // set go right to false
    gameTimer.Start(); // start the game timer
}

```

By now you should be almost familiar with all of the codes but lets explain it again

```

// check all of the controls with this loop
// create a control called X and check it in the form components
foreach (Control X in this.Controls)
{
    /// if X is a picture box and it has a tag of Eggs
    if (X is PictureBox && X.Tag == "Eggs")
    {
        // we move it to top of the screen
        X.Top = rndY.Next(80, 300) * -1; // give it a random y location
        X.Left = rndX.Next(5, this.ClientSize.Width - X.Width); // give it a random x location
    }
}

```

This is the for loop we are using to randomize the location of the eggs on the form, its simply looking for picture boxes with the tag eggs and it will move them above the form and randomize the X location so it looks like they are falling from random places on the form

```

chicken.Left = this.ClientSize.Width / 2; // reset the chicken to middle of the form
chicken.Image = Properties.Resources.chicken_normal2; // change the chicken picture to face left

```

We are moving the chicken to middle of the form and then we are setting image to the left facing one

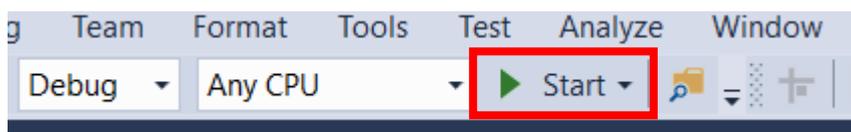
```

score = 0; // set score to 0
missed = 0; // set missed to 0
speed = 8; // set speed to 8

goleft = false; // set go left to false
goright = false; // set go right to false
gameTimer.Start(); // start the game timer

```

We are setting the default values to the variables and lastly starting the timer because without it nothing can be happen EVER.



Now its time to debug click the start button from tool bar



So the eggs are coming down I can move the chicken, it doesn't cross the boundary either from left or right. Scores going up, misses are going up as well and when I missed more than 5 game stopped also you can see the splash eggs are showing up on the screen.

Game restarts and everything goes back to the default values. If you have followed this tutorial thus far, then well done and keep learning. Moo Out.

## Full Source Code –

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SavetheEggs
{
    public partial class Form1 :Form
    {
        bool goleft; // this boolean will be used to check if the player can move left
        bool goright; // this boolean will be used to check if the player can move right
        int speed = 8; // this is the default speed for the Eggs dropping
        int score = 0; // this is the default score value
        int missed = 0; // this is the default miss value
        Random rndY = new Random(); //this will be used to generate a random Y location
        Random rndX = new Random(); // this will be used to generate a random X location
        PictureBox splash = new PictureBox(); // create a new splash picture box, this will added dynamically

    public Form1()
    {
        InitializeComponent();
        reset(); // invoke the reset game function
    }

    private void keyisdown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Left)
        {
            // if the left key is pressed change the go left to true
            goleft = true;
        }
        if (e.KeyCode == Keys.Right)
        {
            // if the right key is pressed change the go right to true
            goright = true;
        }
    }

    private void keyisup(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Left)
        {

```

```

    // if the left key is up then change the go left to false
    goleft = false;
}
if (e.KeyCode == Keys.Right)
{
    // if the right key is up then change the go right to false
    goright = false;
}
}

private void gameTick(object sender, EventArgs e)
{
    label1.Text = "Eggs Caught: " + score; // show the score on Eggs Caught label
    label2.Text = "Eggs Missed: " + missed; // Show the misses on Eggs Missed label

    // if the go left boolean is true AND chickens left is greater than 0
    if (goleft == true && chicken.Left > 0)
    {
        // then we move the chicken 12 pixels to the left
        chicken.Left -= 12;
        //checken image will be change to the one moving left
        chicken.Image = Properties.Resources.chicken_normal2;
    }
    //if the go right is true AND chickens width and left is less than form width
    // meaning the chicken is still within the frame of the game
    if (goright == true && chicken.Left + chicken.Width < this.ClientSize.Width)
    {
        // move the chicken 12 pixels to the right
        chicken.Left += 12;
        // change the chicken image to the one moving right
        chicken.Image = Properties.Resources.chicken_normal;
    }

    //below for loop will check the eggs dropping from the top
    // for each Control we are calling X in this form
    foreach (Control X in this.Controls)
    {
        // if X is a type of picture box AND it has the tag of Eggs
        if (X is PictureBox && X.Tag == "Eggs")
        {
            // then move X towards the bottom
            X.Top += speed;

            // if the EGGS [X] reaches bottom of the screen
            if (X.Top + X.Height > this.ClientSize.Height)
            {
                // if the egg hit the floor then we show the splash image
                splash.Image = Properties.Resources.splash; // set the splash image
                splash.Location = X.Location; // splash shows up on the same location as the egg
                splash.Height = 59; // set the height
                splash.Width = 60; // set the width
                splash.BackColor = System.Drawing.Color.Transparent; // apply transparent background to the picture box

                this.Controls.Add(splash); // add the splash picture to the form

                X.Top = rndY.Next(80, 300) * -1; // position the eggs to a random Y location
                X.Left = rndX.Next(5, this.ClientSize.Width - X.Width); // position the eggs to a random X location
                missed++; // add 1 to the missed integer
                chicken.Image = Properties.Resources.chicken_hurt; // change the chicken image to hurt image
            }

            // if the eggs bound with the chicken image
            // if both picture boxes collide
            if (X.Bounds.Intersects(chicken.Bounds))
            {
                X.Top = rndY.Next(100, 300) * -1; // position the eggs to a random Y location
                X.Left = rndX.Next(5, this.ClientSize.Width - X.Width); // position the eggs to a random X location
                score++; // add 1 to the score
            }

            // if the score is equals to or greater than 20
            if (score >= 20)
            {
                speed = 16; // increase the eggs speed to 20
            }

            // if the missed number is greater than 5
            // we need to stop the game
            if (missed > 5)
            {
                gameTimer.Stop(); // stop the game timer
                // show the message box to say game is over.
                MessageBox.Show("Game Over!! We lost good Eggs" + "\r\n" + "Click OK to restart");
                // once the players clicks OK we restart the game again
                reset();
            }
        }
    }
}

private void reset()
{
    // check all of the controls with this loop
    // create a control called X and check it in the form components
}

```

```

foreach (Control X in this.Controls)
{
    /// if X is a picture box and it has a tag of Eggs
    if (X is PictureBox && X.Tag == "Eggs")
    {
        /// we move it to top of the screen
        X.Top = rndY.Next(80, 300) * -1; // give it a random y location
        X.Left = rndX.Next(5, this.ClientSize.Width - X.Width); // give it a random x location
    }
}

chicken.Left = this.ClientSize.Width / 2; // reset the chicken to middle of the form
chicken.Image = Properties.Resources.chicken_normal2; // change the chicken picture to face left

score = 0; // set score to 0
missed = 0; // set missed to 0
speed = 8; // set speed to 8

goleft = false; // set go left to false
goright = false; // set go right to false
gameTimer.Start(); // start the game timer
}
}
}

```