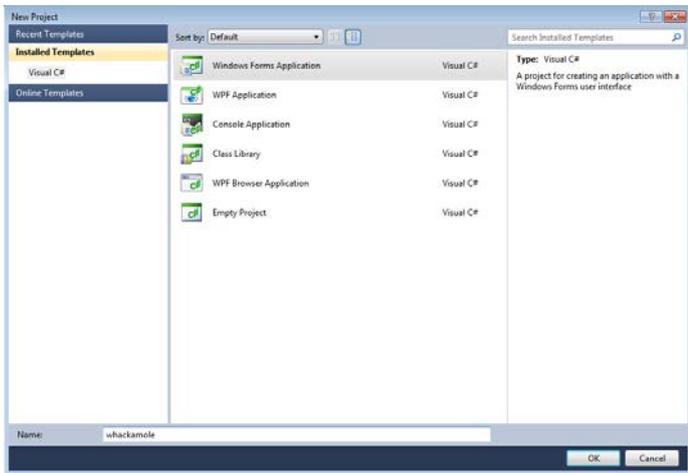


# C# Tutorial – Create a Whack a Mole Game in Visual Studio

Create a new project

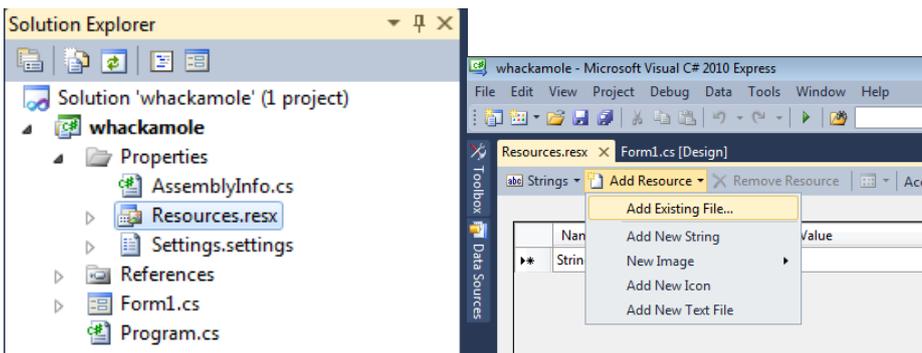
Windows Form Application

Name – whackamole



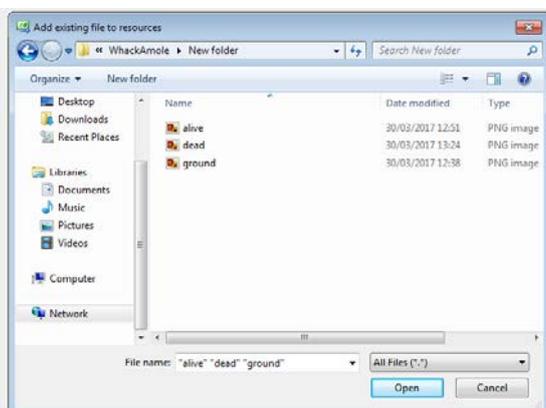
If you are using a different version of visual studio you may have more options but as long as you pick the C# and Windows form application it should be fine.

In the solutions explorer click on the properties arrow and double click on the **resources.resx**



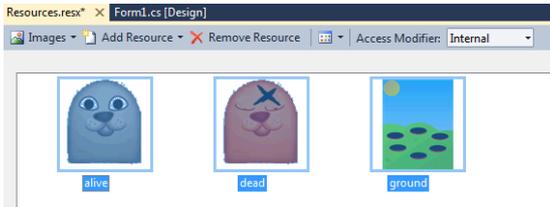
Click on the Add resources and click on Add Existing File

Find the images you saved from Mooict.com



Select all of them and click Open

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



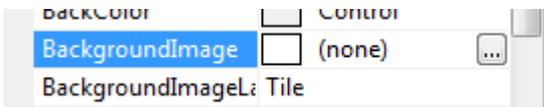
Ground is the background for the form

Make sure you click on File -> Save ALL for now. This will save all these files in the resources folder for this project.

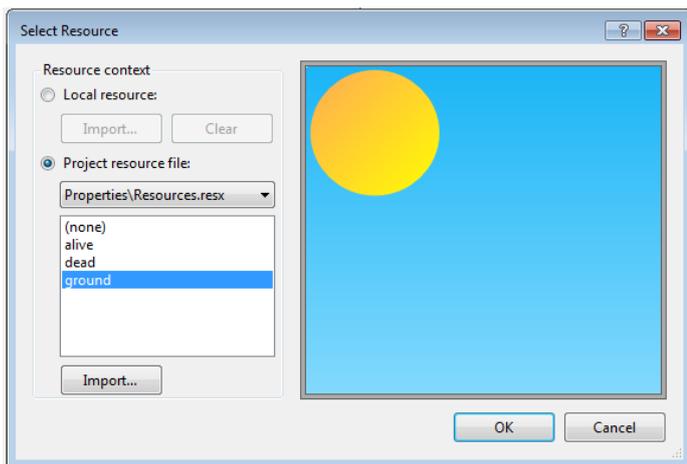
Go back to the form and in the properties window change the following values

**Size – 607, 586**

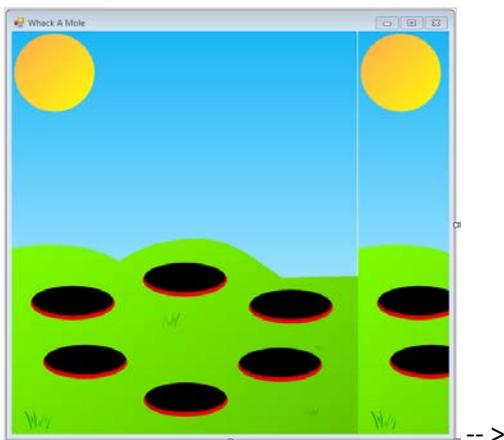
**Text – Whack a Mole**

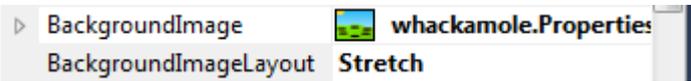


Find the Background Image option in the properties Window Click on the 3 dots

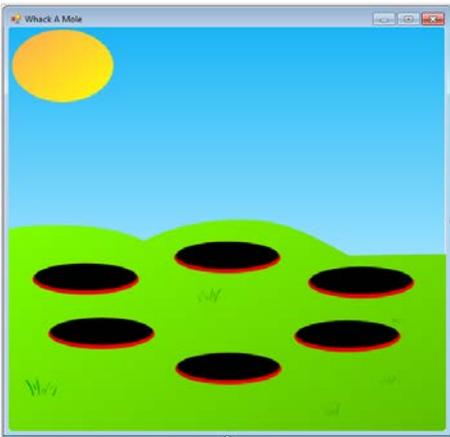


Chose Ground and click OK





Change the Background Image Layout option to Stretch



Now let's add 2 Labels to the screen

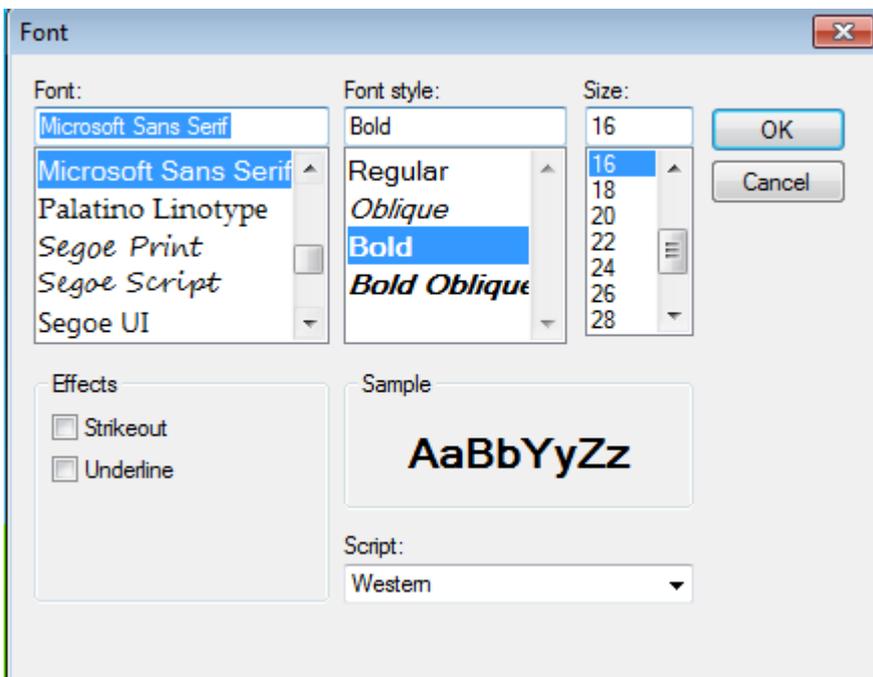


Drag and Drop the First one in the form from the Tool Box

Change the text to **Hit:**



Click on the 3 dots in the font option

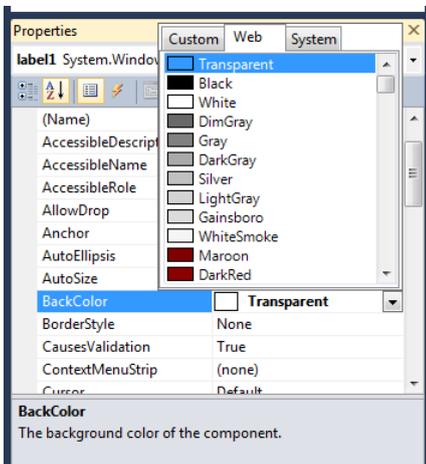


Change the options to BOLD and 16. You can change it to any size or font you want, in order to follow us properly don't make it too big.

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



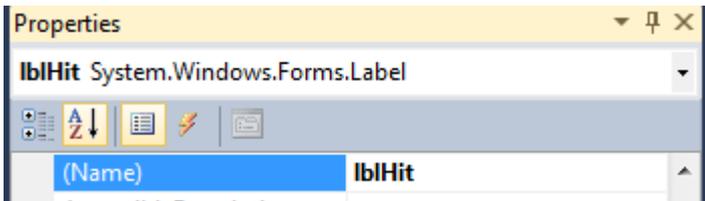
It looks nice but we want to make it transparent so it blends with the background not pop out of it.



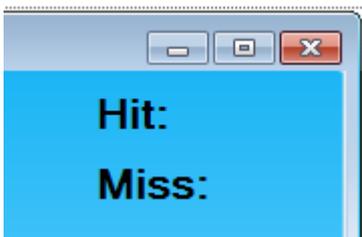
Find the **Back Color** option in the properties window, Click on WEB option on the top and CLICK on Transparent.



Finally change the name of label to **lblHit (its LBLHIT)**



Now copy and paste the HIT label and change the following

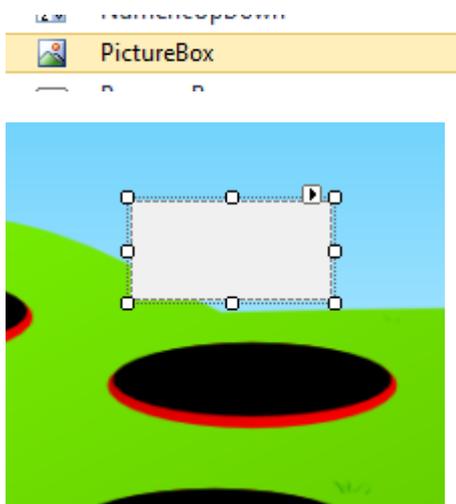


(Name) - lblMiss (ITS LBLMISS)

Text – Miss

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

Drag and Drop a Picture Box to the screen



The mole image will go into this picture box.

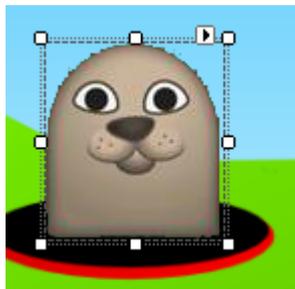
Change the following in the properties window for the picture box

**(Name)** - Mole

**Back Color** – Transparent

**Image** – alive (click on the 3 dots choose alive from the list of images)

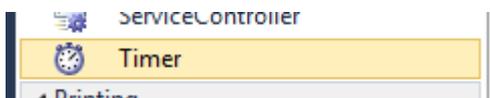
**Size** – 90, 100



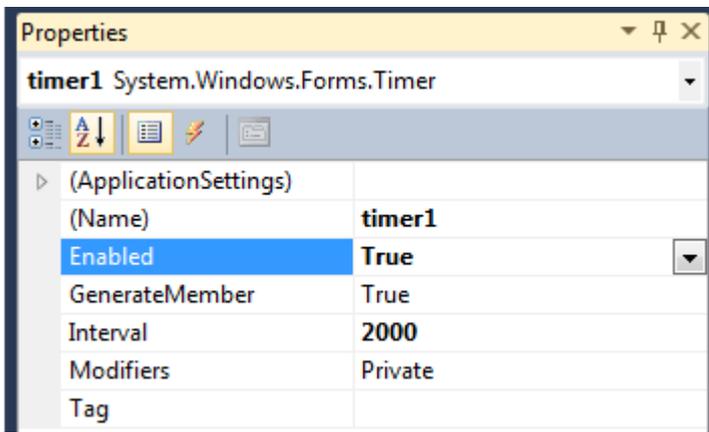
This is the result.

We only need 1 Mole for this game.

Finally let's add the timer to the form



Change the following in the timer properties window.



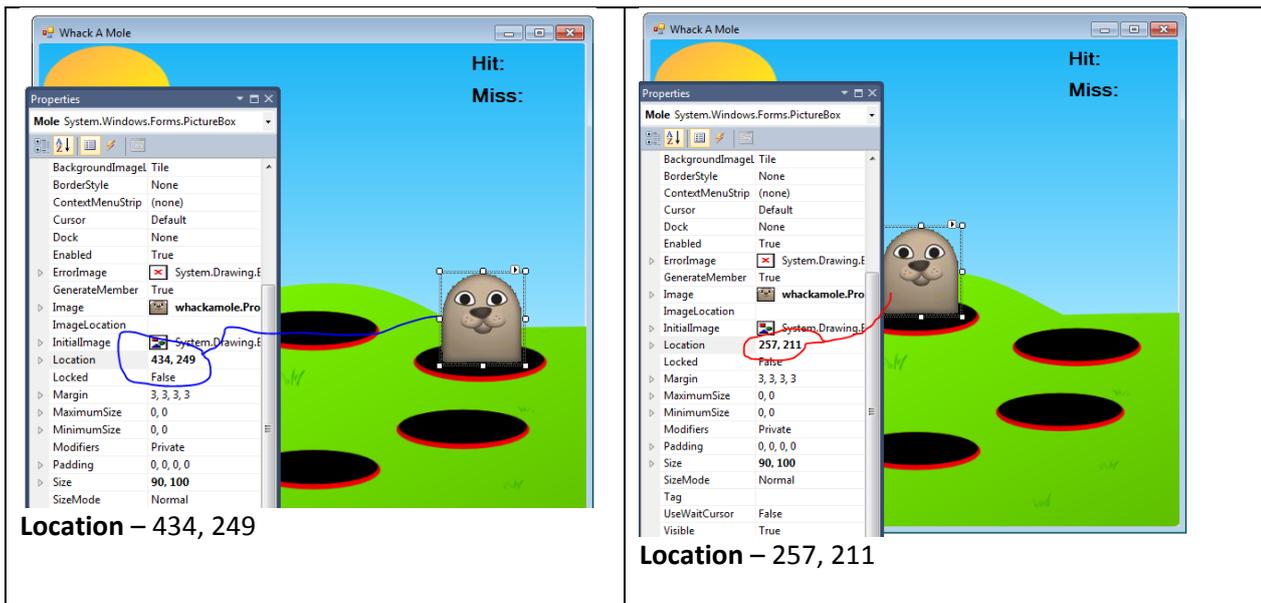
**Enabled** – True

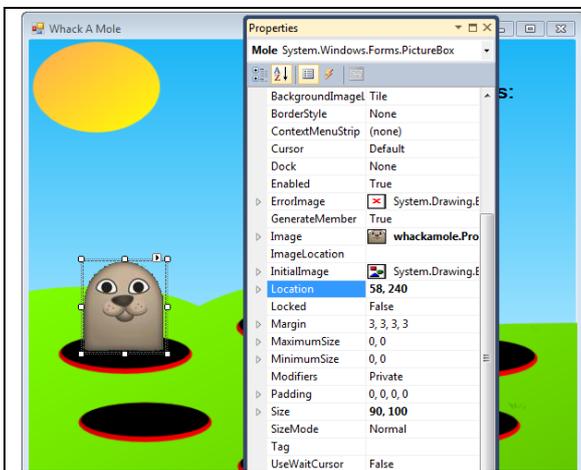
**Interval** – 2000

We want this timer to run every two seconds. (2000 milliseconds = 2 seconds)

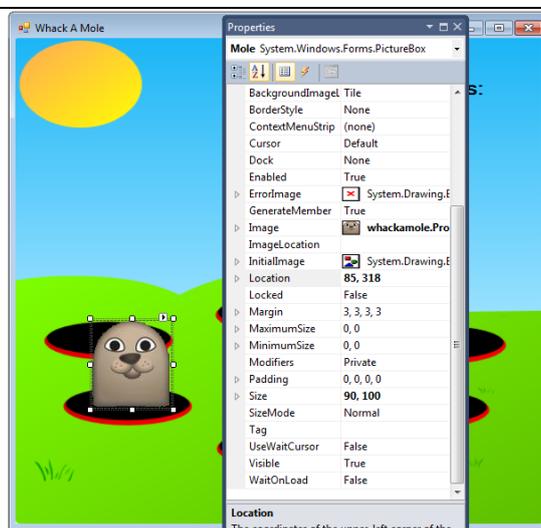
Logic for this game

We will physically move the Mole over each hole and note down the X and Y value so we can use it, move between those spaces randomly in the game.

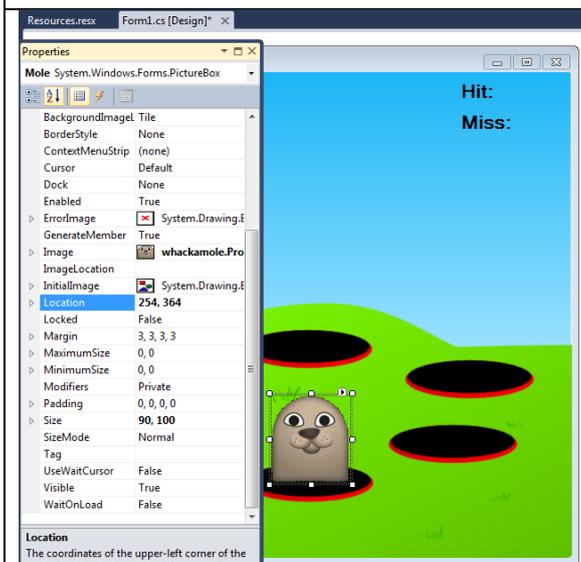




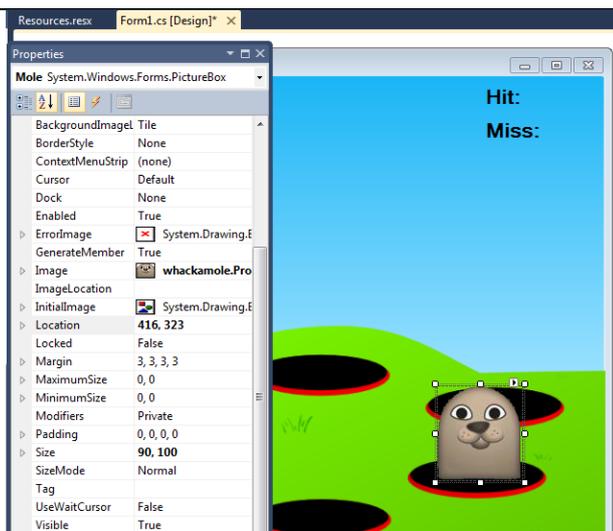
Location – 58, 240



Location - 85, 318



Location - 254, 364



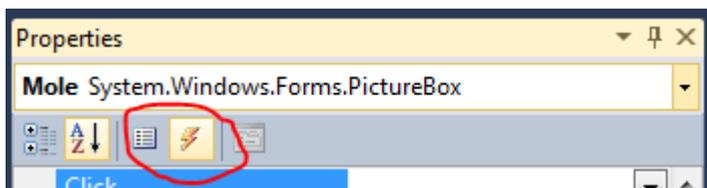
Location – 416, 323

It's important to take notes of each of the locations because we will create a function which will move the picture box between these locations.

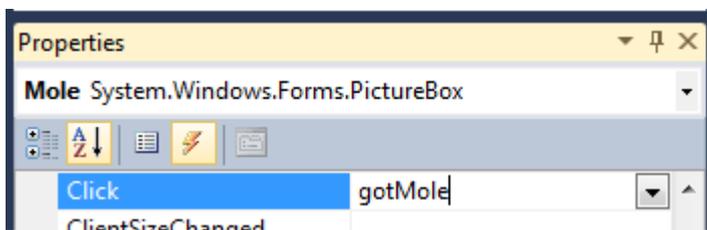
Notice there are two parts to this number sequence for example 10, 10. The first part is the LEFT or the X value of the object and after the comma the second value is the TOP or the Y value of the object.

Now let's start adding events to this program.

Click on the picture box and click on that little lightning bolt icon next to the properties icon.



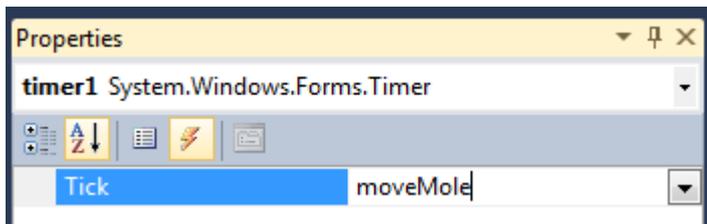
Find the click event in the events window.



Type in **gotMole** and press enter

This will take you to the code view, come back to the design view and add one more event.

Click on the timer and go to the events window again



Type in move mole and press enter

```
Form1.cs x Resources.resx x Form1.cs [Design]
whackamole.Form1
moveMole(object sender, EventArgs e)

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

        private void gotMole(object sender, EventArgs e)
        {

        }

        private void moveMole(object sender, EventArgs e)
        {
            |
        }
    }
}
```

This is my code so far. As you can see I have nothing but two empty events in here. So time to add some code for this game.

Add the highlighted code above the line **public Form1()**

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

```

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 whackamole
{
    public partial class Form1 : Form
    {
        Random rnd = new Random();
        int locationNum = 0;
        int score = 0;
        int misses = 0;
        bool isHit = false;

        public Form1()
        {
            InitializeComponent();
        }

        private void gotMole(object sender, EventArgs e)
        {

        }

        private void moveMole(object sender, EventArgs e)
        {

        }
    }
}

```

**Random rnd = new Random();**

This line is creating a new instance of the random class. We will use this class to generate a random number later in the program. So it's important to define it now.

**int locationNum = 0;**

This is an integer. This only holds 0 for now.

**int score = 0;**

This is the integer for score

**int misses = 0;**

this integer will count how many times we missed the mole

**bool isHit = false;**

This is a Boolean which will come handy when we need to determine whether we hit the mole or we missed it. To begin the game it will be set to false.

Before we add any instructions inside the events functions we need to create a custom function.

```

private void moveMole()
{
    isHit = false;
    Mole.Enabled = true;
    Mole.Image = Properties.Resources.alive;
    Mole.BackColor = System.Drawing.Color.Transparent;

    locationNum = rnd.Next(1, 7);

    switch (locationNum)
    {
        case 1:
            Mole.Left = 434;
            Mole.Top = 249;
            break;

        case 2:
            Mole.Left = 257;
            Mole.Top = 211;
            break;

        case 3:
            Mole.Left = 58;
            Mole.Top = 240;
            break;
    }
}

```

**private void moveMole()**

{

This is the header for move mole function. We are using a private function since we are not going to use it outside this CLASS. Remember the open curly brackets end of the name and closing curly brackets end of the function.

**isHit = false;**

We are setting the is hit Boolean to false.

**Mole.Enabled = true;**

This line is enabling the mole picture box.

**Mole.Image = Properties.Resources.alive;**

This line is changing the image of the mole to alive version.

Only when we click on the picture box we are going to set it to DEAD.

**Mole.BackColor = System.Drawing.Color.Transparent;**

Back Color option is being set here in the code to transparent.

**locationNum = rnd.Next(1,7);**

```

    case 4:
        Mole.Left = 85;
        Mole.Top = 318;
        break;

    case 5:
        Mole.Left = 254;
        Mole.Top = 364;
        break;

    case 6:
        Mole.Left = 416;
        Mole.Top = 323;
        break;

    default:
        break;
}
}
}

```

We have six possible places on the screen where the mole can spawn. In this location num variable we are generating random number between 1 and 7 since the limit is 7 it will only go up to 6 when a number is generated.

```

switch (locationNum)
{

```

This is the beginning part of the switch statement. The logic here is we are going to instruct this program to run a specific location for the mole depending on the number that is generated. Since there are 6 holes in the picture and we have calculated the location of each its easier to place the mole image on them depending on the number that is generated.

For example if the number comes back 5 Mole will be placed on the

248 pixels from the left and 377 pixels from the TOP. We define such activities as case in switch statements.

```

case 1:
    Mole.Left = 428;
    Mole.Top = 264;
    break;

```

in CASE 1 being the random number generated is 1 we are placing the mole

The same method will follow until case 6. Each location are noted before, you can check the previous pages to ensure you have the right number, or check your notes.

after the mole is placed on the scene it's important that we break the condition so it doesn't run further, hence the BREAK; keyword.

```

default:
    break;

```

We left the default

}

end with the closing curly brackets

Now lets add the timer event codes

```

private void moveMole(object sender, EventArgs e)
{
    lblHit.Text = "Hit: " + score;
    lblMiss.Text = "Miss: " + misses;

    if (isHit == false)
    {
        misses++;
    }

    if (score > 15)
    {
        timer1.Stop();
        Mole.Enabled = false;
        MessageBox.Show("You Win");
    }
    else if (misses > 10)
    {
        timer1.Stop();
        Mole.Enabled = false;
        MessageBox.Show("You Lose");
    }
}

```

```

lblHit.Text = "Hit: " + score;

```

This line we are connecting the score integer to the lblHit label.

```

lblMiss.Text = "Miss: " + misses;

```

lblMiss label will show how many times we missed the mole.

```

if (isHit == false)
{
    misses++;
}

```

In order to calculate how we are going to miss the Mole, if the isHIT is false when the timer is runs then we can add 1 to the misses integer. doing misses++ means adding one to it each time it runs.

```

    }
    moveMole();
}

```

```

if (score > 15)
{
timer1.Stop();
Mole.Enabled = false;
MessageBox.Show("You Win");
}

```

If the score is greater than 15 then we can stop the timer, disable the mole and show a message stating you've won the game.

```

else if (misses > 10)
{
timer1.Stop();
Mole.Enabled = false;
MessageBox.Show("You Lose");
}

```

Else if the misses value is greater than 10, we will once again stop the timer, disable the picture box and show a message stating you've lost.

Add the following in the **gotMole** click event. This event is directly linked to the picture box so each time we click it the following instructions will execute.

```

private void gotMole(object sender, EventArgs e)
{
    score++;
    Mole.Image = Properties.Resources.dead;
    isHit = true;
    Mole.Enabled = false;
}

```

```
score++;
```

This will add 1 to the score

```
Mole.Image = Properties.Resources.dead;
```

The mole picture box image will change from the alive picture to the dead mole picture.

```
isHit = true;
```

We will change the is hit Boolean to true.

```
Mole.Enabled = false;
```

This will disable the picture box so if the player keeps clicking on the dead mole no points will be added to it.

Now test the game, debug it and find out how you done, if there is an error make sure to go back to tutorial and look through the steps you've taken.

Note - we have set the timer to 2 seconds you can make it faster to make the game more challenging.

## Final screens



Full Code for the game.

```

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 whackamole
{
    public partial class Form1 : Form
    {
        Random rnd = new Random();
        int locationNum = 0;
        int score = 0;
        int misses = 0;
        bool isHit = false;

        public Form1()
        {
            InitializeComponent();
        }

        private void gotMole(object sender, EventArgs e)
        {
            score++;

            Mole.Image = Properties.Resources.dead;

            isHit = true;

            Mole.Enabled = false;
        }

        private void moveMole(object sender, EventArgs e)
        {
            lblHit.Text = "Hit: " + score;
            lblMiss.Text = "Miss: " + misses;

            if (isHit == false)
            {
                misses++;
            }

            if (score > 15)
            {
                timer1.Stop();
                Mole.Enabled = false;
                MessageBox.Show("You Win");
            }
        }
    }
}

```

