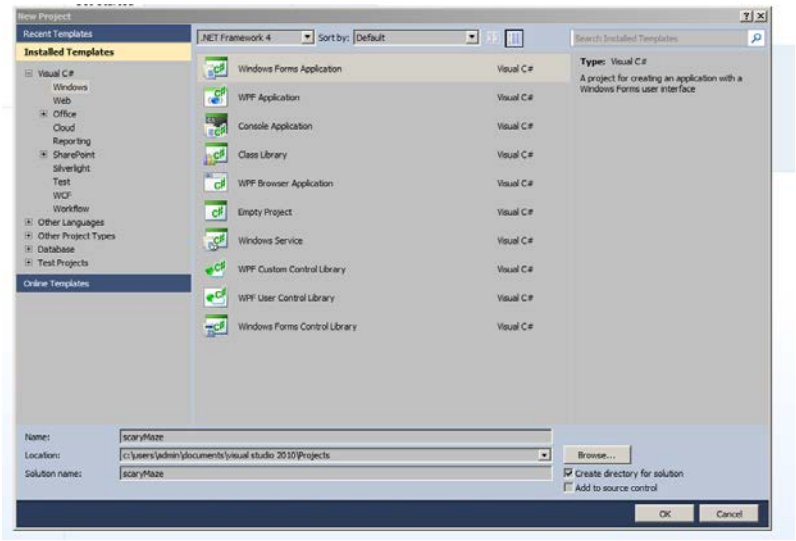
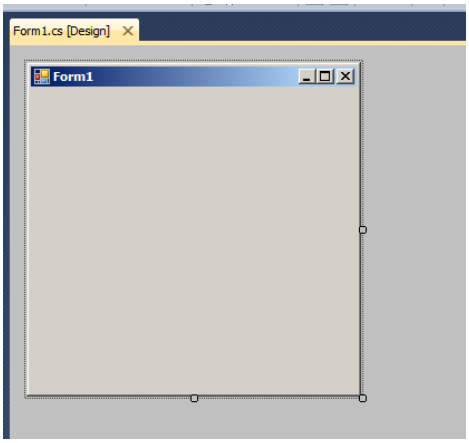
C# Tutorial - Create a Scary Maze Game in Visual Studio

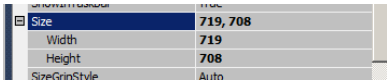
Start a new project in visual studio. Choose a Windows Form Application under the C# programming language.



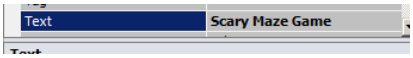
call the project scaryMaze. Click OK.



This is the first view of the project. We have only one Form in this project so far.

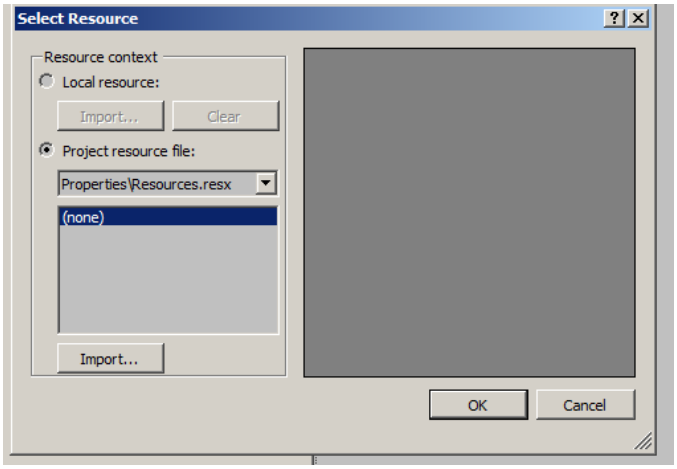


Check the properties window and set the size of the form to 719, 708.



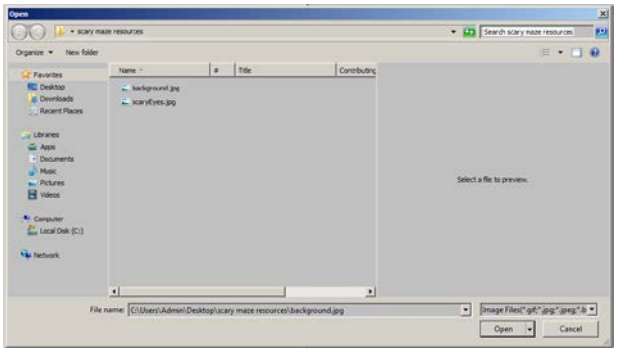
Change the text of the form to Scary Maze Game. This option will change the title of the form so instead of stating Form1 it will now say Scary Maze Game.

Check the background image option and click on the three dots ...

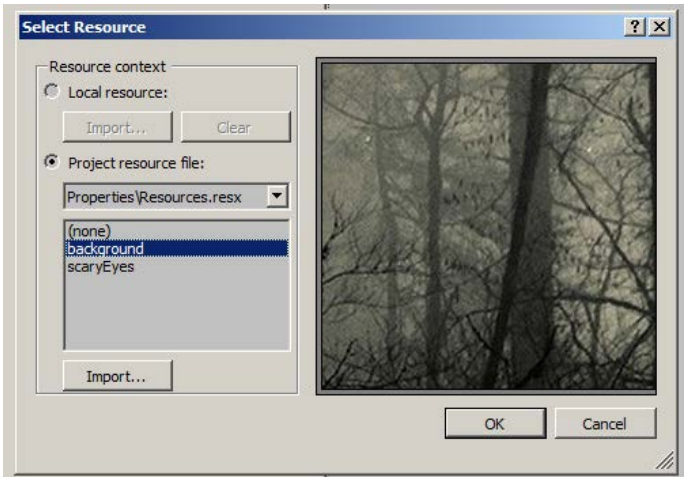


This window will show up. While you have project resource file option selected click on Import.

If you have downloaded the resources from **mooict.com** website then we can continue to the next step, if you haven't click on the link above this tutorial and download the require resources.

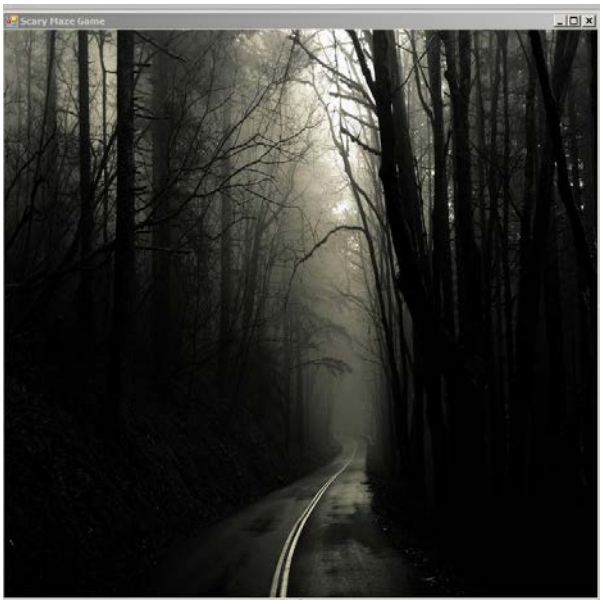


We have two images here, first one is for the background of this form and scary eyes image will be used later in this project. For now select both and click on open.

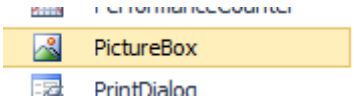


From the list select background and click ok

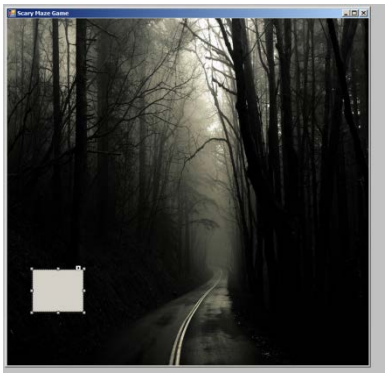
Now go back to the properties window and check the option background image layout and change it to stretch. This will stretch the image to the size of this form.



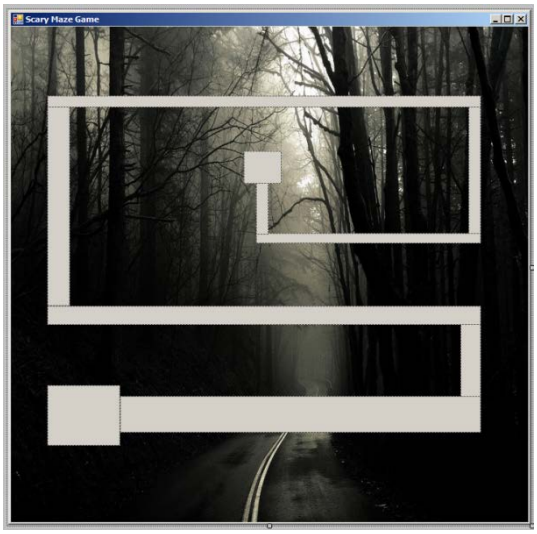
This is what the new form now looks like. Well done.



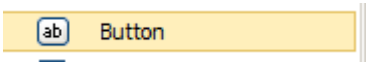
From the toolbox select the picture box picture and drag it to the screen.



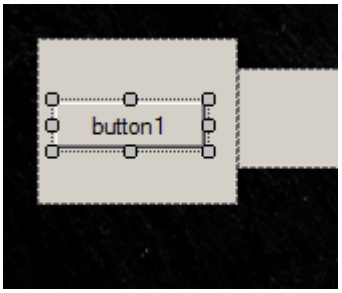
This is the new picture box on the screen.



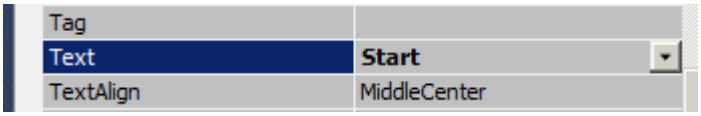
You can copy and paste the same picture box and make the shapes as you see above, we simply used the same picture and stretched in various places to make it look like a puzzle. Note - you can make up your own puzzle, but make sure they are all picture boxes. Let's move on to add another component to the game.



Find the button component in the tool box.

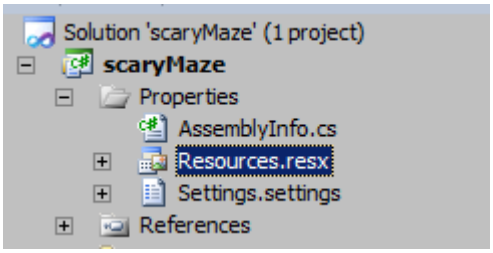


drag and on the first picture we added to the form.

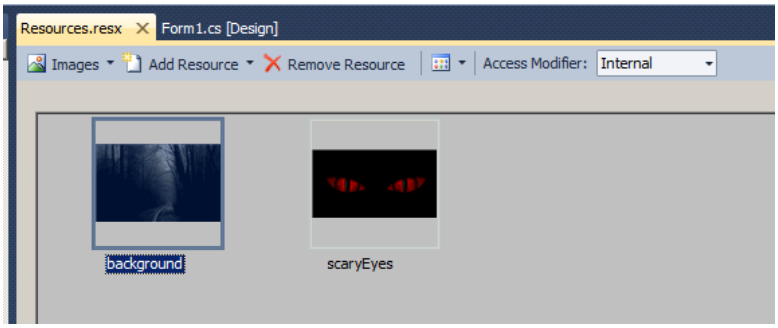


while the button is selected go to the properties window and change the Text of the button to Start.

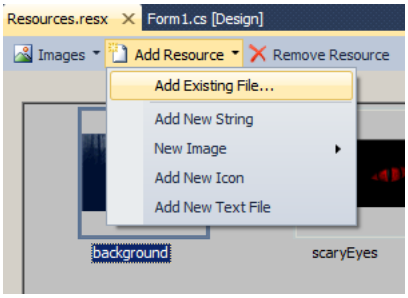
Now you might of noticed we have 2 audio files in the resources you downloaded earlier. We will add them to this project now.



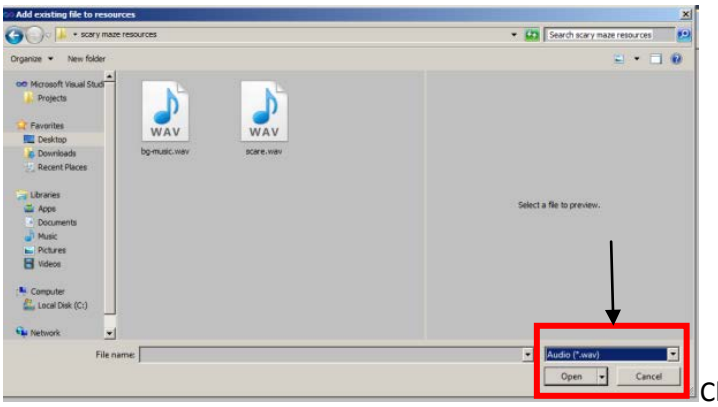
Look at the solutions explorer, under the properties options you will find Resources.resx this is where all the resources are kept. Lets double click on it.



Now here you can see the two images we imported earlier.

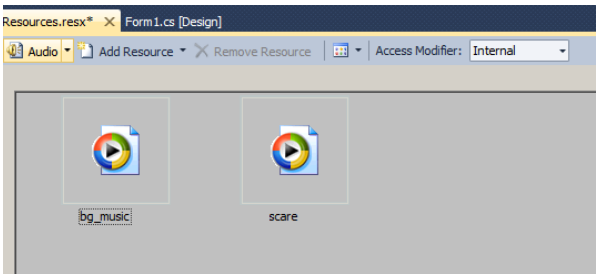


Click on the add resource drop down and click on add existing file.

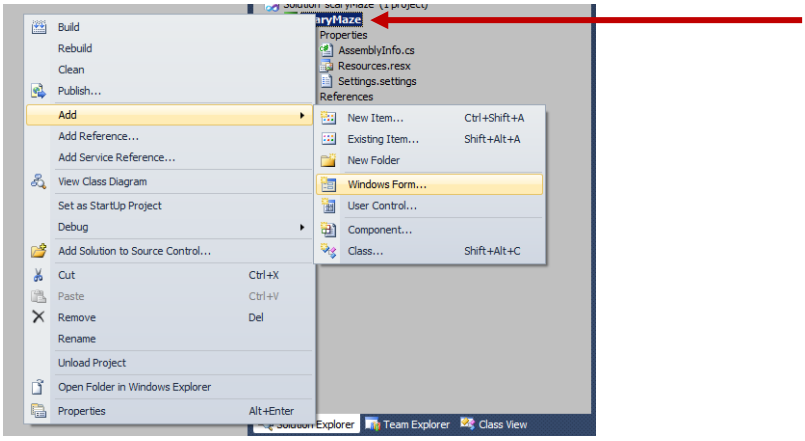


Change the file type to Audio and you will be able to find both audio files in the folder.

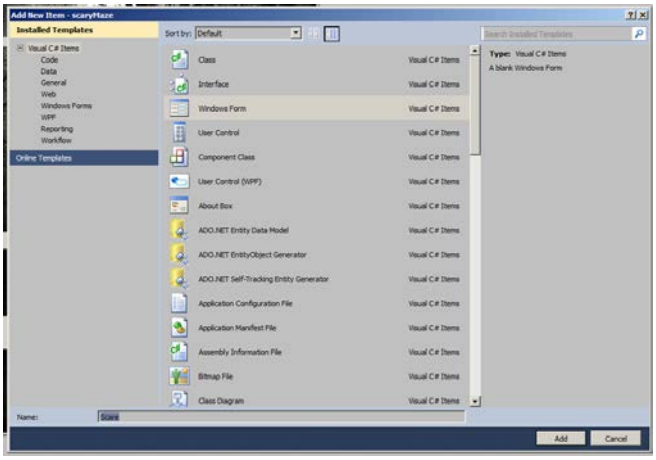
Highlight them both and click open.



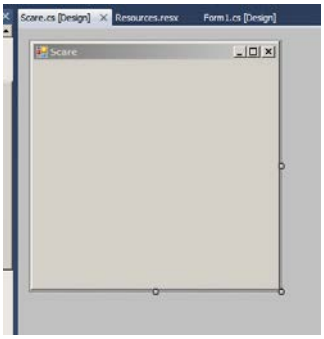
Now they both been added to the resources. Press **CTRL + S** to save the resources file.



Go back to the solutions explorer and right click on the scaryMaze -> Add -> Windows Form



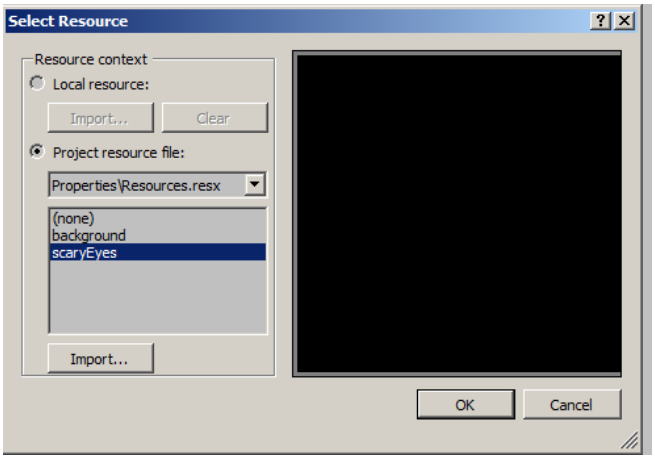
call this form Scare and click add.



This is the new form.

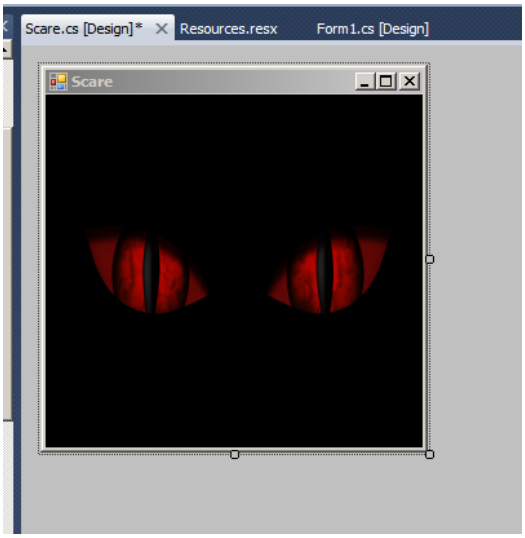
Change the text option in the properties window to Scare.

Same as before, look at the properties window and click on the three dots on the background image option.

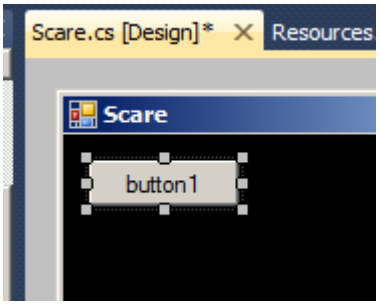


Click on the scary eyes image and click on.

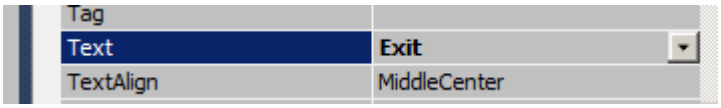
Under the background image layout and change it to Stretch.



This is what the form looks like. Now we will make this form full screen when we want to scare the player for now leave it this size. Now find a BUTTON in the toolbox and drag to the form.

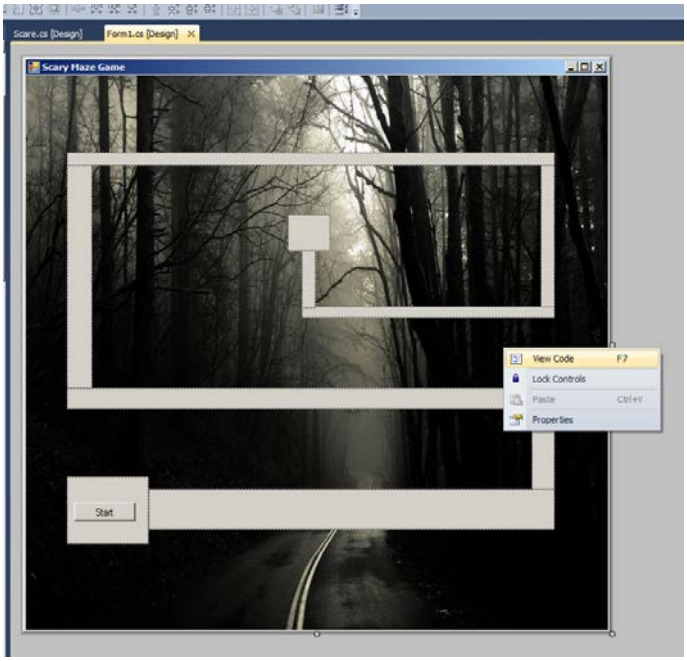


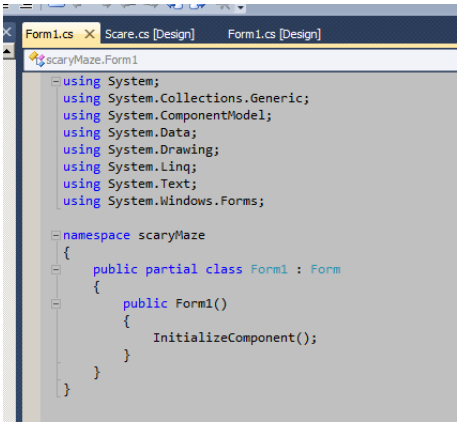
This is the button we dragged to the Scare Form.



Change the text of this button to Exit. Since this form will be full screen we need a way for the user to go back to the puzzle.

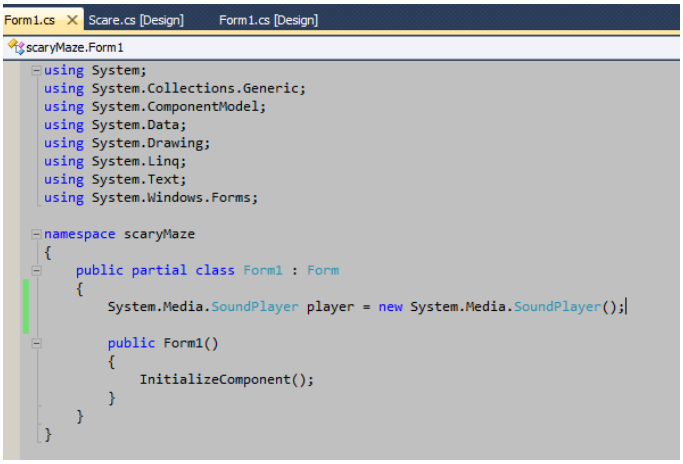
Time to add the code for this game. Let's go back to the Form1, right click on the form and click on View Code or F7





This is the first view of the code. These codes are done by default and we will add our own in there, follow the tutorial thoroughly.

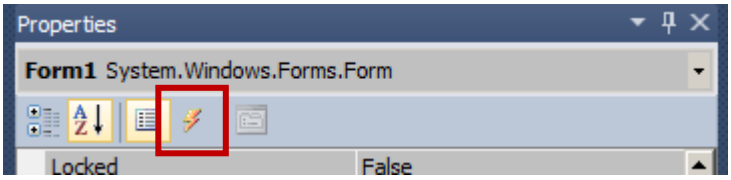
Lets start by adding our sound player to the code.




```
System.Media.SoundPlayer player = new System.Media.SoundPlayer();
```

This line code is using the system sound player. We are creating new instance of this sound player and calling it player. We will be able to direct the sound files we imported earlier in the project and play it on the form. This same sound file will be used to play the background music and the scream sound.

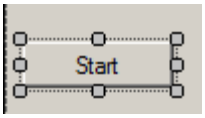
Lets go back to the form and add a few events



Click on that little lightning bolt icon in the properties window. This will open the events list. These events can be added to the project. For now we are interested in the Mouse Hover event.

Name this event endGame and press enter.

Come back to the form and click on the start button.

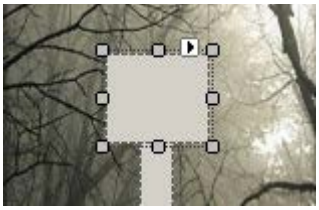


Go back to the events properties by clicking on the lightning bolt icon.

find the click properties and type start and press enter.

Let's go back to the form once more and add the last event for this game.

This mouse over event will be added to the last picture box of the puzzle, when the play is on top of that one we will bring out the scare form.



This is our last picture box in the puzzle.

add the event called ScarePlayer and press enter.

Below is the code so far in the code view.

```
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 scaryMaze
{
    public partial class Form1 : Form
    {
        System.Media.SoundPlayer player = new System.Media.SoundPlayer();

        public Form1()
        {
            InitializeComponent();
        }

        private void endGame(object sender, EventArgs e)
        {

        }

        private void start(object sender, EventArgs e)
        {

        }

        private void ScarePlayer(object sender, EventArgs e)
        {

        }
    }
}
```

This is the code so far. you can see that both of the events are added to the code. Lets add some instructions inside them now.

```
        private void endGame(object sender, EventArgs e)
        {
            reset();
        }
```

this is the code for end game event. This will run when the player hovers over the form and not the picture box. This will run a separate function called reset. If there are red lines under the code so far don't panic we are going to declare this function soon in the code, follow along.

```
        private void start(object sender, EventArgs e)
        {
            button1.Enabled = false;
            StarGame();
        }
```

This is the start event from the button. This function will disable the button when its clicked and run the Start Game function.

```
        private void StarGame()
        {
            foreach (Control x in this.Controls)
            {
                if (x is PictureBox)
                {
                    x.Enabled = true;
                    x.BackColor = System.Drawing.Color.SkyBlue;
                }
```

```
            }
            // play the back ground music when it starts
            player.Stream = Properties.Resources.bg_music;
            player.PlayLooping();
        }
```

This is the Start Game function.

We are running a for each loop in this function to indentify the picture boxes. When we find these picture boxes we will enable them and then change their back colour to sky blue.

We will also then play our background music. notice the player.Stream option being declared in this function. This line of code refers back to when we identified the sound player earlier in the code. We are linking this player to the music file from the resources folder and then we are playing this sound file in loop.

```
        private void reset()
        {
            button1.Enabled = true;
            player.Stop();

            foreach (Control x in this.Controls)
            {
                if (x is PictureBox)
                {
                    x.Enabled = false;
                    x.BackColor = System.Drawing.Color.Black;
                }
            }
        }
```

This is the reset function, this function will run the when the player has gone off the puzzle, this function will enable the start button, stop the music and then run a loop which will disable all of the picture boxes on the form and change their background colours to black.

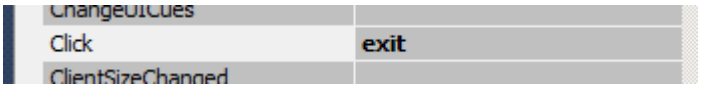
```
        private void ScarePlayer(object sender, EventArgs e)
        {
            Scare fullScreen = new Scare();
            fullScreen.Show();
        }
```

This is the final event in this code. This mouse hover event will run when the player hovers over the last picture box and it will open the Scare form we created earlier. When this function runs it will create an instance of the scare form we are calling it full screen and then we are allowing the full screen form to show on display.

Double check the code here to make sure you haven't missed anything and then let's move to the Scare form.

In this scare form we have one exit button, lets add a exit event to this button.

Click on the Exit button



Find the Click option in the events properties, type exit and press enter.

```
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 scaryMaze
{
    public partial class Scare : Form
    {
        public Scare()
        {
            InitializeComponent();
        }

        private void exit(object sender, EventArgs e)
        {

        }
    }
}
```

This is the view of the scare form code so far.

Public Scare() function add the following code inside it.

```
        public Scare()
        {
            InitializeComponent();

            // make this form full screen
            this.WindowState = FormWindowState.Normal;
            this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None;
            this.Bounds = Screen.PrimaryScreen.Bounds;

            // play the scart sound
            System.Media.SoundPlayer player = new System.Media.SoundPlayer();
            player.Stream = Properties.Resources.scare;
            player.Play(); // play this once
        }
```

First we are making space after the line Intialize Component () this is a very important function for windows form so lets have that run first and then we add our own.

Full Screen Code Explanation:

```
this.WindowState = FormWindowState.Normal;
```

This means we are aiming at Scare window. So by citing THIS the following will only be applied to the form we currently working on. Window State we are leaving as Normal.

```
this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None;
```

We are changing the border style of the form to NONE, this means it will not have the minimize, maximize or CLOSE option on top. Hence why we need that button.

```
this.Bounds = Screen.PrimaryScreen.Bounds;
```

We are checking the primary screen size height and width and we will cover it with this form.

Playing sound -

```
System.Media.SoundPlayer player = new System.Media.SoundPlayer();
player.Stream = Properties.Resources.scare;
player.Play(); // play this once
```

These three lines you have seen in the last form, only different here is instead of using PlayLooping we are playing it only once with the Play() option. We have also changed the background music to scare sound for this.

```
private void exit(object sender, EventArgs e)
{
    this.Close();
}
```

In this exit function enter this line. This.Close() will run the close function which will close the form. Since it doesn't have the windows close on the top right corner we will use our own to allow the user to close this form.