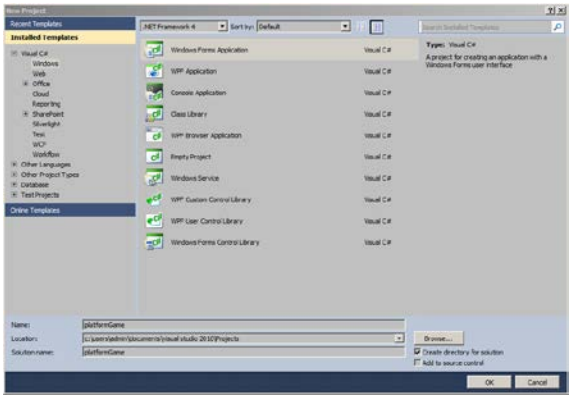
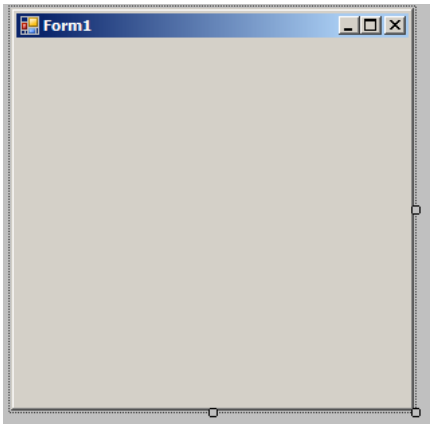
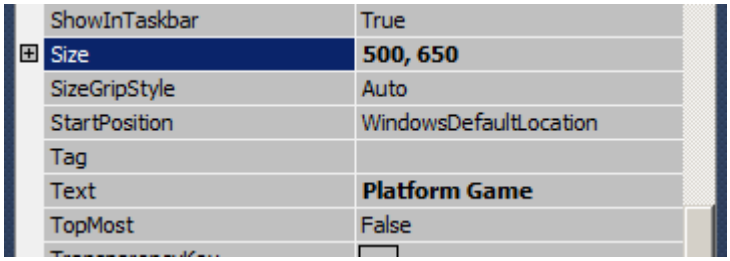
C# Tutorial - Create a simple Platform game in visual studio

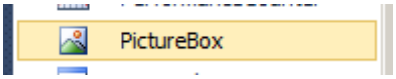
start a new project, Choose windows form application in C# and name it **platformgame**







In the properties window change the setting to Size - 500, 650 and Text to Platform Game

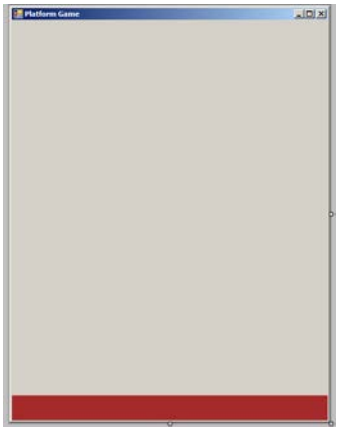


Add your first picture box to the form.

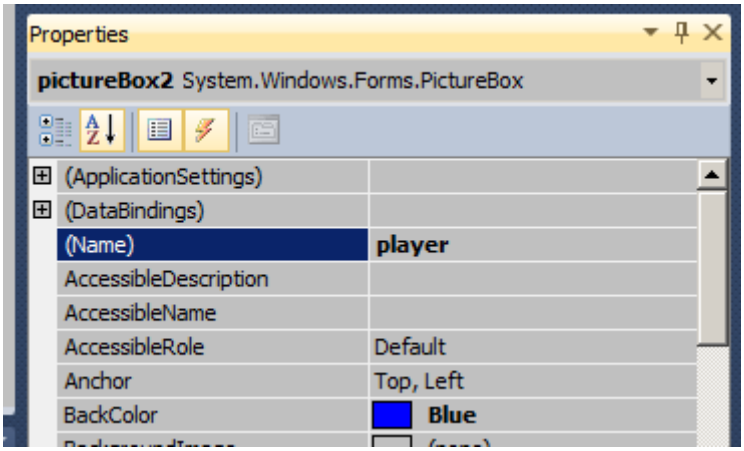
Change the back colour to Brown

Change TAG to platform

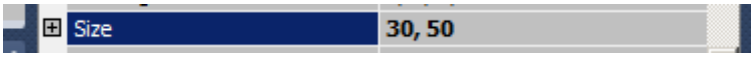
Now you can scale the picture to go across the floor



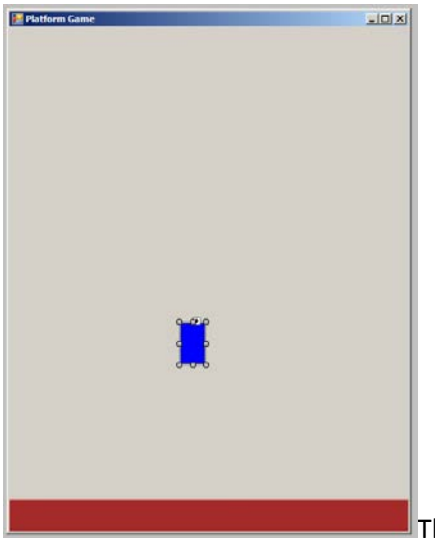
Lets add the second picture box to form this time we are going to add the player.



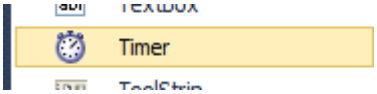
Change the name property to player and back colour to blue.



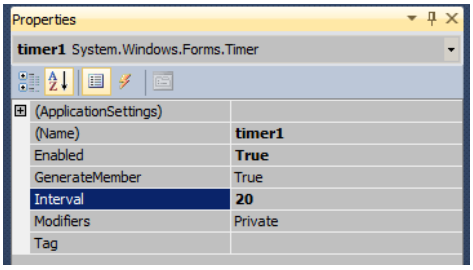
Change the size to 30, 50



This is the player



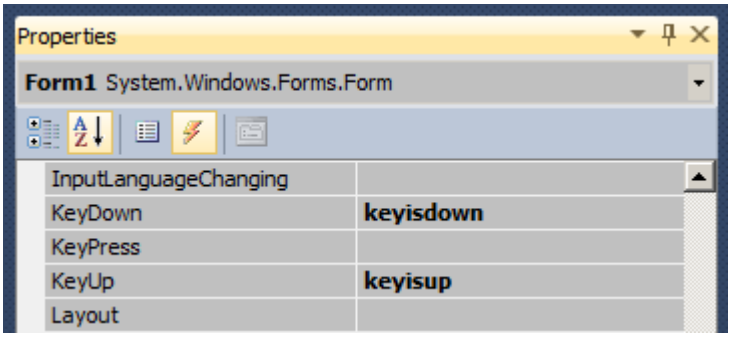
Now add a timer to the form



Change enabled = TRUE and Interval to 20.

Adding Events to the form

Click on the from and go to the events panel in the properties window



Find the key down - Type **keyisdown** and press enter

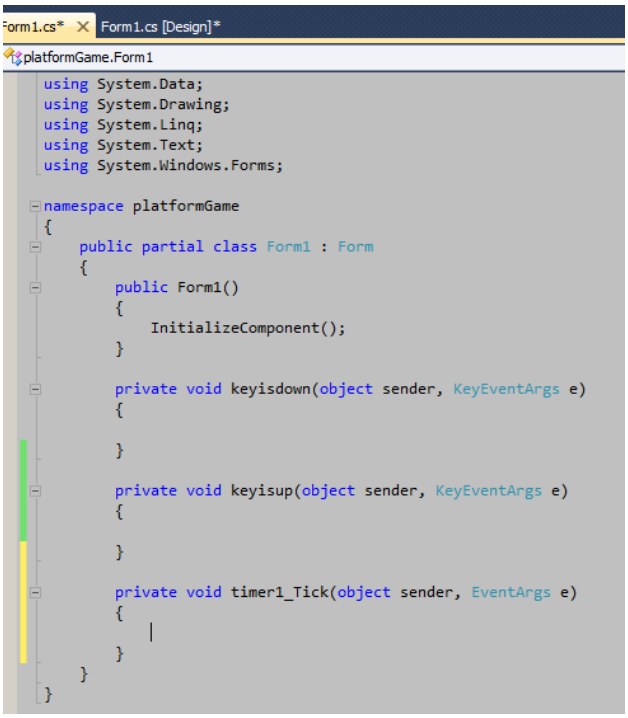
Find the key up - Type **keyisup** and press enter

Both of these actions will take you to the code view. We need to do one more thing before we can start coding for this game.

Double click on the timer

<-- double click this

Now this what the code view looks like



We have all 3 events done. Let's get coding

```
        bool goleft = false;
        bool goright = false;
        bool jumping = false;

        int jumpSpeed = 10;
        int force = 8;
        int score = 0;
```

```
public partial class Form1 : Form
{
    bool goleft = false;
    bool goright = false;
    bool jumping = false;

    int jumpSpeed = 10;
    int force = 8;
    int score = 0;

    public Form1()
    {
        InitializeComponent();
    }
```

These are variables we need for this game. Go left and go right and jumping are Booleans which will determine the direction of the player. Integer jumpSpeed is holding the value 10. Integer force is holding the value 8. Integer score is zero.	This is what is looks like in visual studio. Notice we entered the code above the public form() line. This is where it should be.

Key down function

`` private void keyisdown(object sender, KeyEventArgs e) { if (e.KeyCode == Keys.Left) { goleft = true; } if (e.KeyCode == Keys.Right) { goright = true; } if (e.KeyCode == Keys.Space && !jumping) { jumping = true; } } ```	This is the key is down function. This will trigger each time the buttons are pressed. When the left button is pressed we can go left to true. When the right button is pressed we can change the go right to true. When the space bar is pressed AND the character is not jumping we change the jumping to true.

key is up function

`` private void keyisup(object sender, KeyEventArgs e) { if (e.KeyCode == Keys.Left) { goleft = false; } if (e.KeyCode == Keys.Right) { goright = false; } if (jumping) { jumping = false; } } ```	This function will turn all of Booleans back to false when the users releases each key for example when the LEFT is up we change go left back to false. Same for right and SPACE bar.

Timer event (player movement)

Below are the first part of the timer event codes, enter them inside the function.

`` player.Top += jumpSpeed; if (jumping && force < 0) { jumping = false; } if(goleft) { ```	First we are starting up with play.Top += jumpSpeed; This line we continuosly drop the player towards the floor. We want it because we want the game to have a gravity effect. ``` if (jumping && force < 0) { jumping = false; } ```

```
            player.Left -= 5;
        }

        if(goright)
        {
            player.Left += 5;
        }

        if (jumping)
        {
            jumpSpeed = -12;
            force -= 1;
        }
        else
        {
            jumpSpeed = 12;
        }
```

This line above is checking if the player is jumping and force of the jump is less than 0 if so then we can change jump back to false.

```
if(goleft)
{
player.Left -= 5;
}
```

If go left is true then we can push the character towards left of the screen.

```
if(goright)
{
player.Left += 5;
}
```

If go right variable is true then we are going to move the player left towards the right.

```
if (jumping)
{
jumpSpeed = -12;
force -= 1;
}
```

If jumping is true then we change the jump speed integer to minus 12 which means it will thrust the player towards the top and we decrease the force by 1. If we don't do this then character can fly over everything we need to give the jump a limit.

```
else
{
jumpSpeed = 12;
}
```

ELSE the character is not jumping then we can keep the jump speed on 12.

Timer event code continued

```
foreach (Control x in this.Controls)
            {
if (x is PictureBox && x.Tag == "platform")
                {
   if (player.Bounds.IntersectsWith(x.Bounds) && !jumping)
                    {
          force = 8;
          player.Top = x.Top - player.Height;
                    }
                }
            }
```

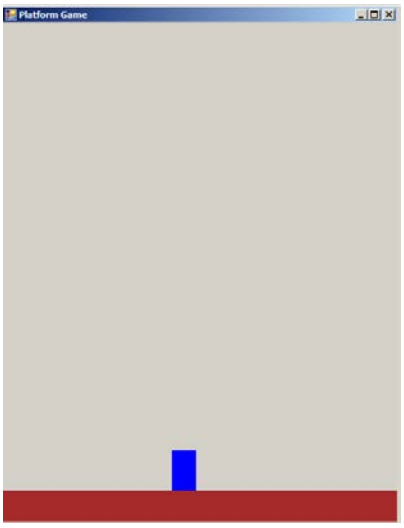
In this piece of code we will scan the whole form to find the picture boxes with this once our player interacts it will have to land on top of it.

For each control x in this.Controls means for each of the windows component we create a variable called X and give it a type of controls.

IF X is a type of picturebox AND tag is equals to "platform"

IF play bounds intersect with the bounds of the platform and out character is not jumping then we change the force integer back to 8 and player will be above the platform.

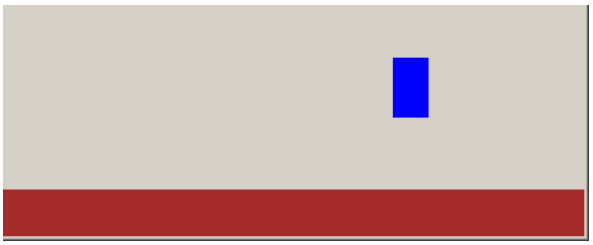
Try the game out now.



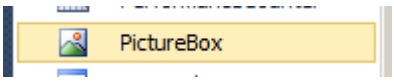
It stopped when it dropped to the platform


Move left

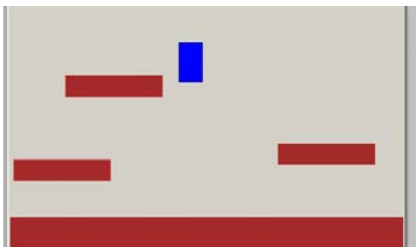

Move right


Press SPACE to JUMP

Now off course we can't just have a single platform on a platform game.

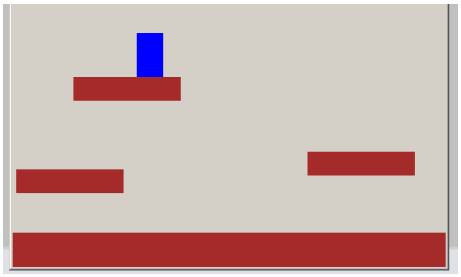
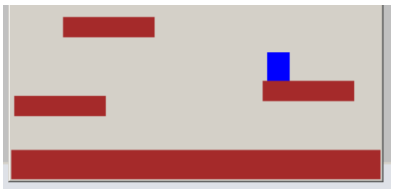
Now time to add some more picture boxes to form. 

You can choose any back colour you want. For this example we will use the BROWN one.



Note make sure you change the TAG for each of the picture boxes to platform.

Run the game and check.

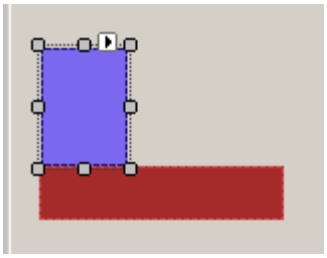



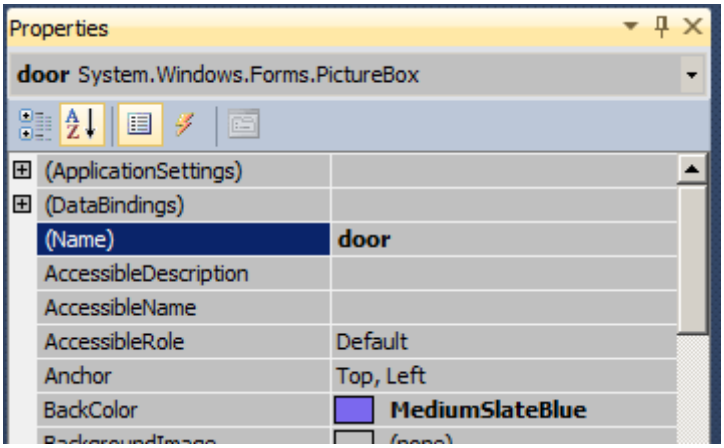
It worked.

Now you can get creative with the whole thing.

We need to have a way for the player to win or lose the game in case you want to go further with this.

WIN the game

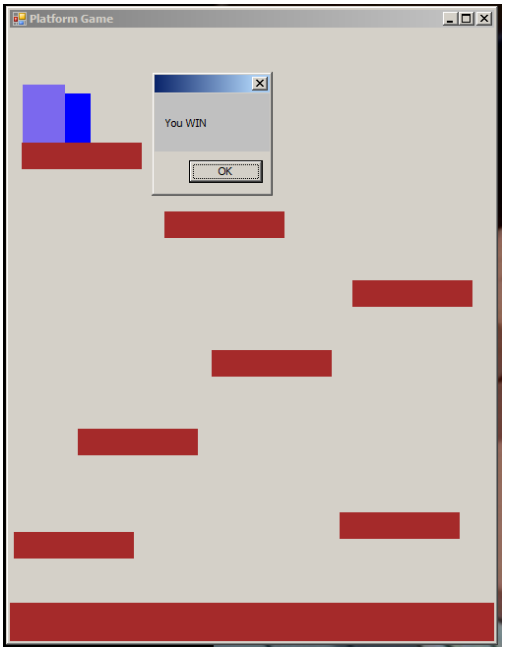

This is the new picture box I've added to the screen.



name it door.

Lets add a small code so when we reach the door we can end the game

`if (player.Bounds.IntersectsWith(door.Bounds)) { timer1.Stop(); MessageBox.Show("You WIN"); }`	Inside the timer event we will add this line which is checking if the player touches the door we can stop the timer and show a message showing that you won the game.



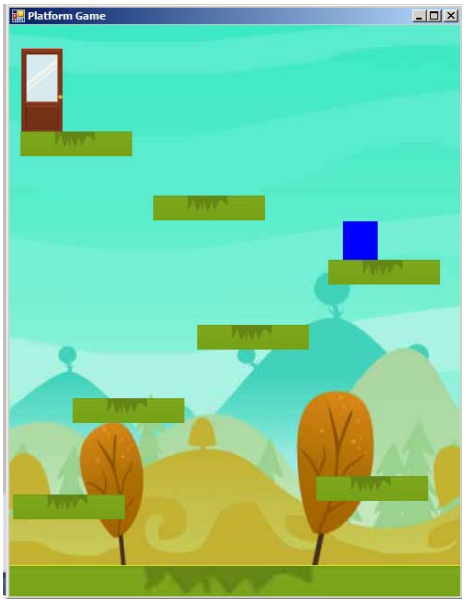
There you go Job done.

It's a simple

You can also do a another picture box which will end the game.

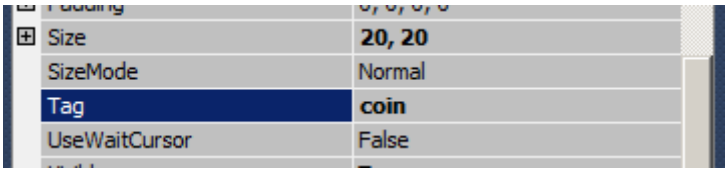
Since all of them are picture boxes we can design the level to make it look more game such see the example below.

You make your own or use some images from the internet and experiment with making your own platform game.



Lets Add some points to the game

Lets add another picture box to form. Give this one a size of 20, 20 and back colour of yellow. Make sure it has TAG of coin.



Now lets add the following code inside the for loop where we have written about the player and platform collision before.

`foreach (Control x in this.Controls)` `{` `if (x is PictureBox && x.Tag == "platform")` `{` `if` `(player.Bounds.IntersectsWith(x.Bounds) && !jumping)` `{` `force = 8;` `player.Top = x.Top - player.Height;` `}` `}` `}`	This is current for loop we will add the instructions for the coin here now.
`foreach (Control x in this.Controls)` `{` `if (x is PictureBox && x.Tag == "platform")` `{` `if (player.Bounds.IntersectsWith(x.Bounds) && !jumping)` `{` `force = 8;` `player.Top = x.Top - player.Height;` `}` `}` `if (x is PictureBox && x.Tag == "coin")` `{` `if (player.Bounds.IntersectsWith(x.Bounds) && !jumping)` `{` `this.Controls.Remove(x);` `score++;` `}` `}` `}`	We are using the same principle as before except this time we are checking if the picture box has a TAG of coin. If so and it's in the bounds of the player then we can remove it from the form and add 1 to the score variable. Clever right. ☺ ☻♥!

Test Run


It works. Lets copy and paste the coin all over the level now.


We collected all of the points and completed the level.

Happy learning. Now you can try to build on this yourself and see if you add more points or enemies on the screen and interact with the player. Make sure to add label to screen which can used to show the score. We can do everything for YOU.

Below are the full code for this 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 platformGame
{
    public partial class Form1 : Form
    {
        bool goleft = false;
        bool goright = false;
        bool jumping = false;

        int jumpSpeed = 10;
        int force = 8;
        int score = 0;

        public Form1()
        {
            InitializeComponent();
        }

        private void keyisdown(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Left)
            {
                goleft = true;
            }
            if (e.KeyCode == Keys.Right)
            {
                goright = true;
            }
            if (e.KeyCode == Keys.Space && !jumping)
            {
                jumping = true;
            }
        }

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

            if (e.KeyCode == Keys.Right)
            {
                goright = false;
            }
            if (jumping)
            {
                jumping = false;
            }
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            player.Top += jumpSpeed;

            if (jumping && force < 0)
            {
                jumping = false;
            }

            if (goleft)
            {
                player.Left -= 5;
            }

            if (goright)
            {
                player.Left += 5;
```

```
            }

            if (jumping)
            {
                jumpSpeed = -12;
                force -= 1;
            }
            else
            {
                jumpSpeed = 12;
            }

            foreach (Control x in this.Controls)
            {
                if (x is PictureBox && x.Tag == "platform")
                {
                    if (player.Bounds.IntersectsWith(x.Bounds) && !jumping)
                    {
                        force = 8;
                        player.Top = x.Top - player.Height;
                    }
                }

                if (x is PictureBox && x.Tag == "coin")
                {
                    if (player.Bounds.IntersectsWith(x.Bounds) && !jumping)
                    {
                        this.Controls.Remove(x);
                        score++;
                    }
                }
            }

            if (player.Bounds.IntersectsWith(door.Bounds))
            {
                timer1.Stop();
                MessageBox.Show("You WIN");
            }

        }
    }
}
```