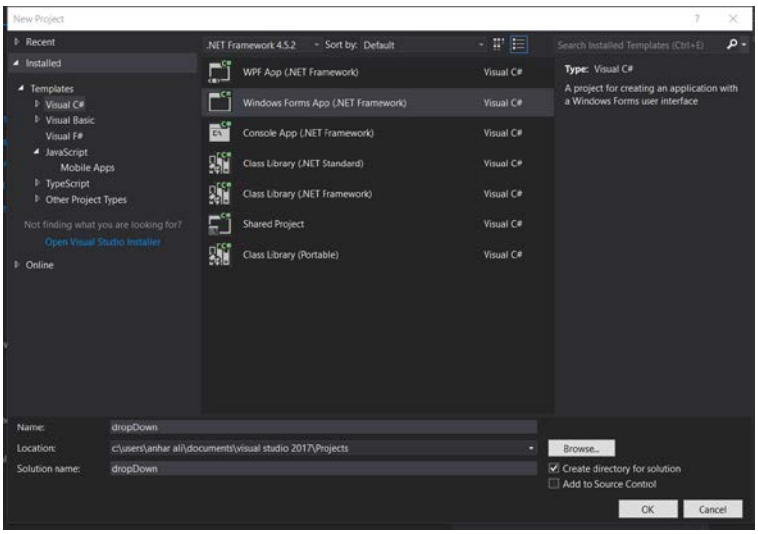
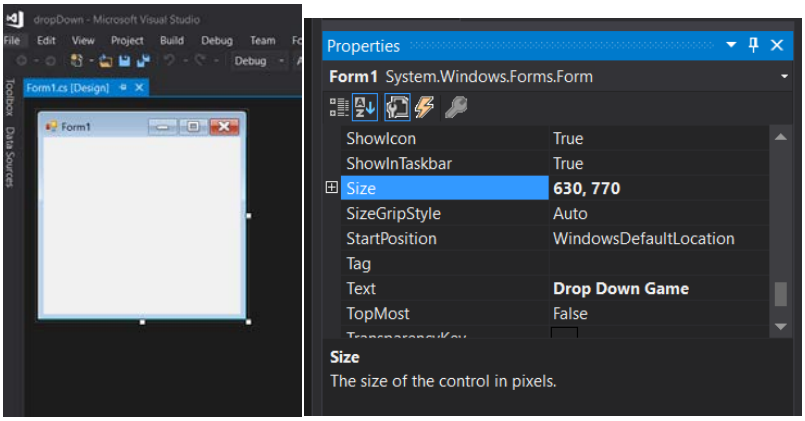
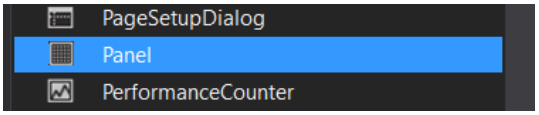
C# Tutorial – Create a Platform Drop-Down Game

Start Visual Studio under the C# programming language select Windows Form Application, name the project **DropDown** and click OK.

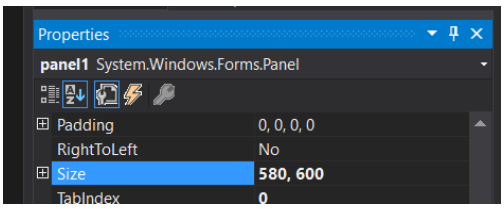




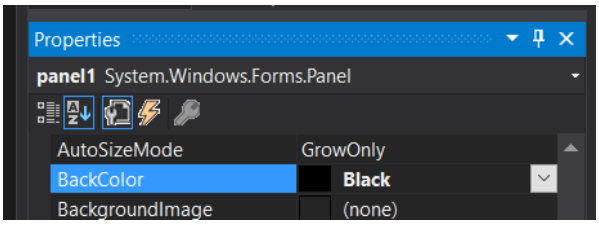
This is the initial windows form screen. If you click on the form and then find the properties window which is usually located in the bottom right of the window. If you cannot find it there then simply right click on the form and click on properties. Inside the properties window you will find an option called Size. Change the size of the form to 630, 770. This means 630 pixels width and 770 pixels height.



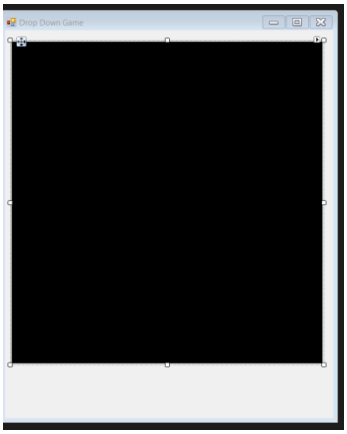
Find a panel control in the tool box and drag it to the form.



While the panel is selected on the form, go to the properties window and change the size to 580, 600. Remember we are changing the PANEL's height and width not the forms.

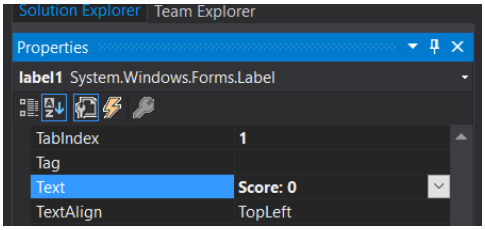


While the panel is selected, locate the Back Color option and change it to BLACK.

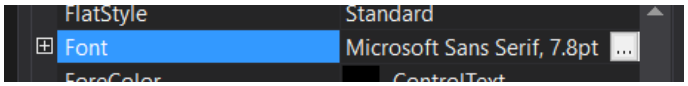


This is what the Game screen looks like right now.

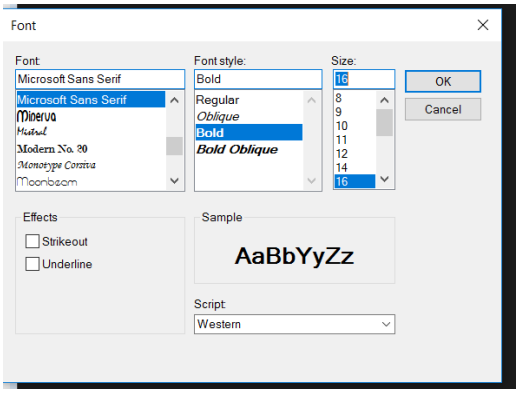
Select the label component from the tool box. Drag and drop it to the Panel



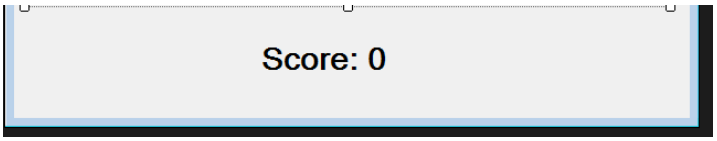
Change the text in the properties window for the label to Score: 0



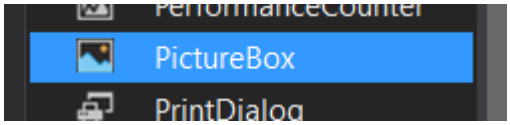
Click on the font option and now click on the three dotted button …



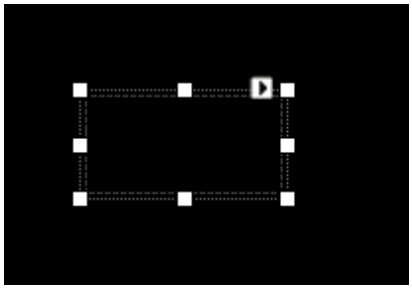
This box will come up, in this box we can change the font size and style. So pick BOLD and size 16. CLICK OK.



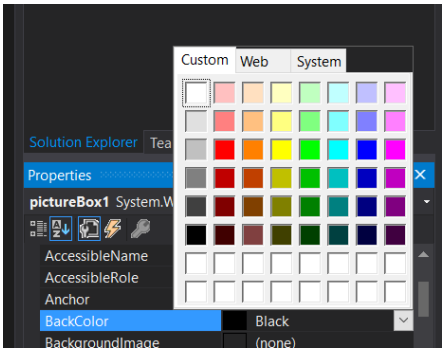
This is what the code label looks like now.



Go back to the form and click and drag a picture box to the panel. Make sure the picture box is inside the panel

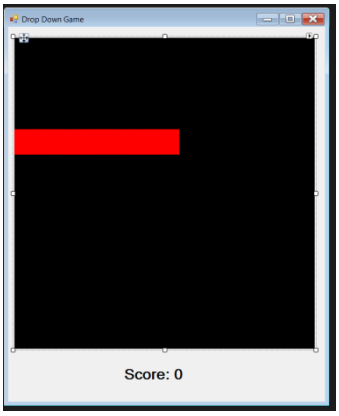


Click on the picture box and it will be selected in the panel.

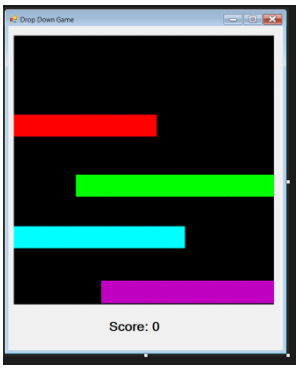


Check the back colour property in the properties window and change the back colour to RED.

Inside the properties window find the option called tag and type platform. Since we are planning to have multiple platforms we can use the tag option to identify which are the platforms in this game.



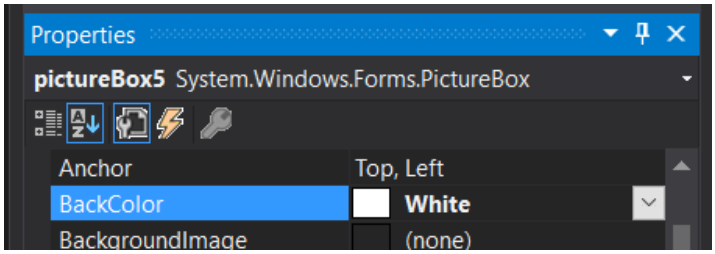
This is the first picture box done in this panel. Notice we have stretched it so covers just about half of the width of the original pane. Make sure the left side of it is touching the left part of the panel.



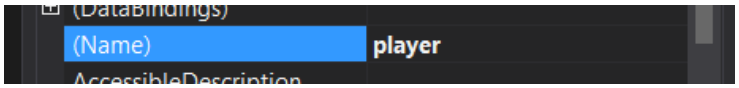
Now add the another 3 picture boxes, use the same steps as the first one but you can chose different size and back colour for each of them. We have done so with ours check the image above.

Adding the Player–

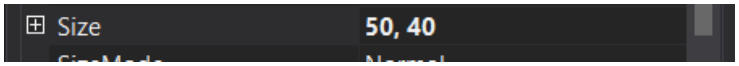
Add another picture box from the tool box, this picture box will be used as our player on the screen.



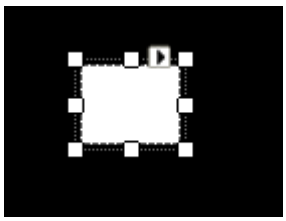
Choose a white back colour for this picture box.



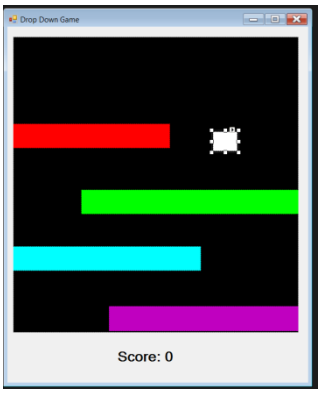
Change the name to player in the properties window



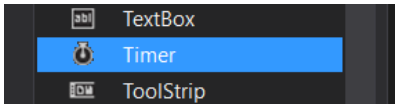
Change the size to 50, 40



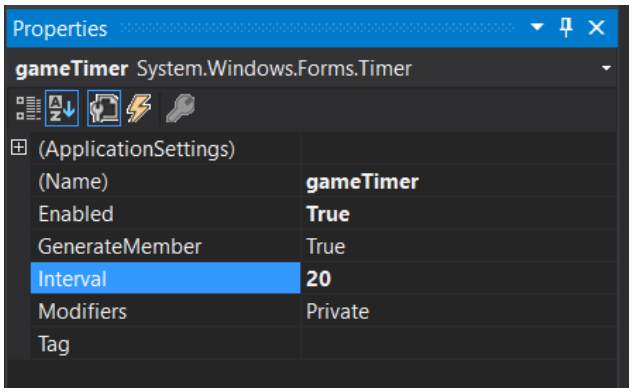
This should be the final view of the player picture box



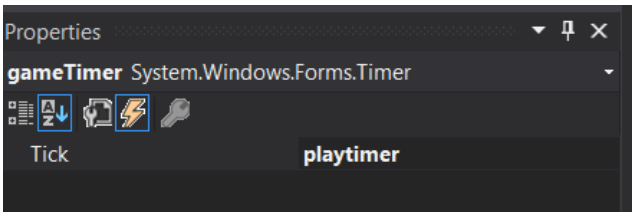
You can place the character anywhere in the screen, we chose to place it on top of the green one (2nd platform)



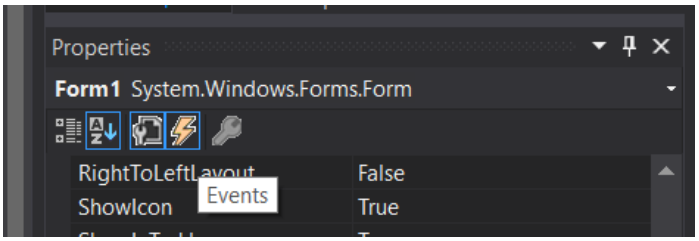
Find the timer object in the tool box and add it to the form.



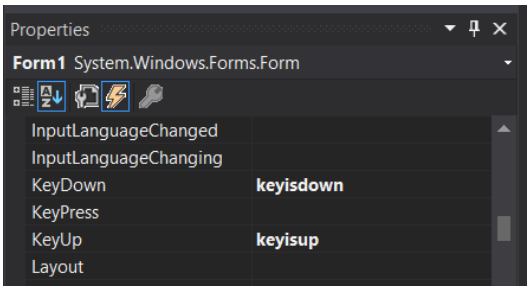
In the properties window for the timer object, name the timer gameTimer, change Enabled to True and change the Interval to 20. All this is for the properties of the timer.



Now go to the events window for the timer, its right next to the properties icon in the Window. In the Tick option type playtimer this will be name of the event which controls the timer. All of the game instructions will be mainly focused inside the events.



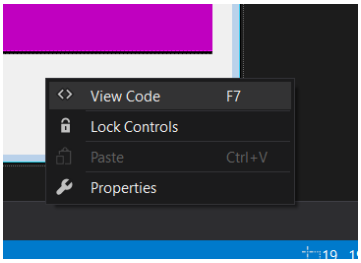
Now click on the form and click on the events window.



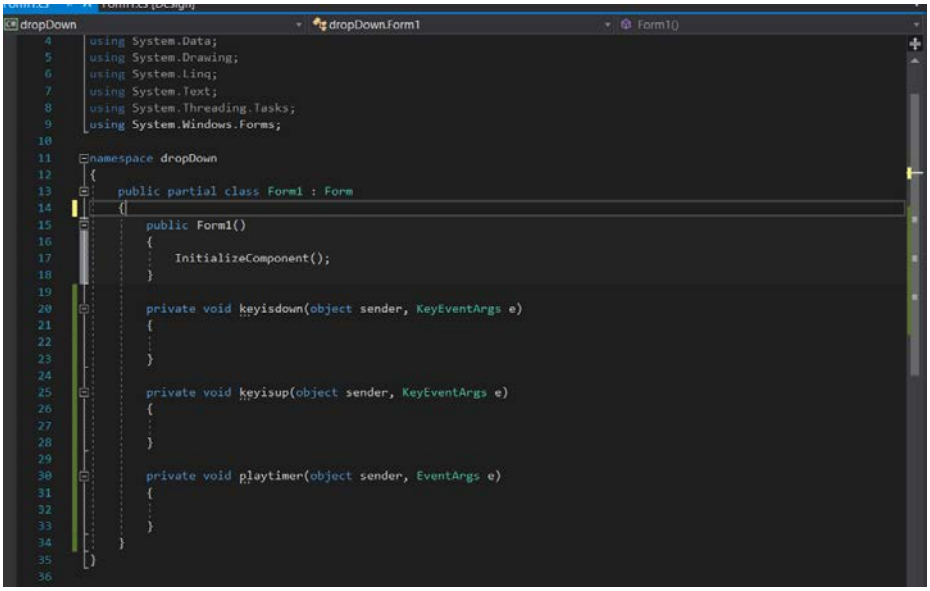
Find the key down and the key up events in the list.

Write **keyisdown** for the key down events.

Write **keyisup** for the key up events.

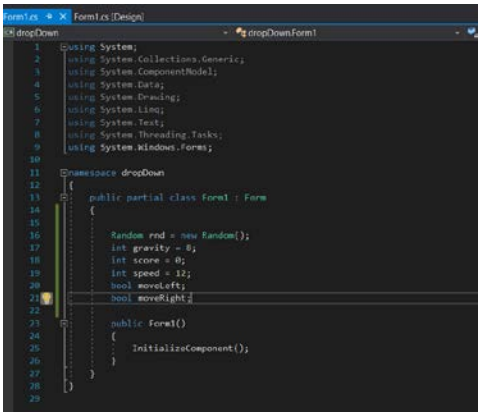


Now let's go to the Code view for the game. Right click on the form and you will see the code view. You can also press F7.



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

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

        private void keyisdown(object sender, KeyEventArgs e)
        {

        }

        private void keyisup(object sender, KeyEventArgs e)
        {

        }

        private void playtimer(object sender, EventArgs e)
        {

        }
    }
}
```

This is what the code view looks like so far, we have added 3 events to this code and we will adding out variables next.



```
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 dropDown
{
    public partial class Form1 : Form
    {

        Random rnd = new Random();
        int gravity = 8;
        int score = 0;
        int speed = 12;
        bool moveLeft;
        bool moveRight;

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

Add the variables before the public Form1() line. Make space before them and then add the variables as seen below.

Variables

```
Random rnd = new Random();
int gravity = 8;
int score = 0;
int speed = 12;
bool moveLeft;
bool moveRight;
```

`Random rnd = new Random();`

In this line is creating a new random instance. We are calling it RND and it will help us to generate random numbers.

`int gravity = 8;`

This line above is the integer called gravity and we will give it a value of 8.

`int score = 0;`

this line above we are creating a score variable and it will have 0 value in it.

`int speed = 12;`

The integer above is a speed variable which will hold the value 12.

`bool moveLeft;`

Above we are creating a Boolean called moveLeft. By default it has a value of False. Next is the moveRight Boolean with the same default value of False.

Key Down Event

```
Private void keyisdown (object sender, KeyEventArgs e)
        {
if (e.KeyCode == Keys.Left)
            {
moveLeft = true;
            }
if (e.KeyCode == Keys.Right)
            {
moveRight = true;
            }
        }
```

In the function above we have mapped out two key presses for the game, Left and Right.

If key code equals to left we change the move left Boolean from false to true. This way the program will be notified when the left key has been pressed. The same logic is inputted for the right key.

Key Up Event

```
Private void keyisup (object sender, KeyEventArgs e)
        {
if (e.KeyCode == Keys.Left)
            {
moveLeft = false;
            }
if (e.KeyCode == Keys.Right)
            {
moveRight = false;
            }
        }
```

In the function we are changing the move left and right Booleans back to false when the keys are up. When either of these Booleans are turned from true to false the program will be notified and it will give us a chance adjust the game according this.

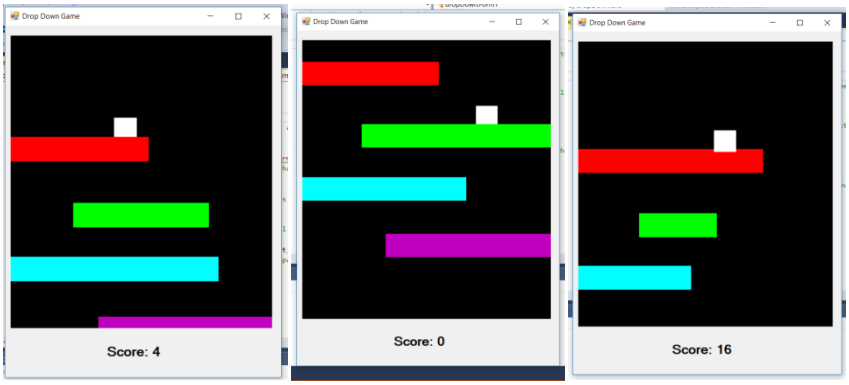
Timer Event

```
Private void playtimer(object sender, EventArgs e)
        {
            player.Top += gravity; // drop the player from the top
            label1.Text = "Score: " + score; // show the score on the label

          if (moveLeft && player.Left > 1)
           // if move left is true and player left is greater than 1
            {
              player.Left -= speed;
              //move player to the left
            }
           if (moveRight && player.Left + player.Width < panel1.Width)
          //if player right is true and player left plus player width is greater than panel1s width
            {
              player.Left += speed;
             //move player to the right
            }

      foreach (Control x in panel1.Controls)
     // this for loop has a control called x and will be used to search for components
     // in the panel 1
            {
            if (x is PictureBox && x.Tag == "platform")
           // if x is type of picture box and has a tag string platform
                {
                  x.Top -= 5;
                  // move the x to the top by 5 pixels

                   if (x.Top < panel1.Top - x.Height)
                   // if x's top is less then panel 1 top minus x's height
                    {
                      x.Top = panel1.Height + x.Height;
                     // re-position x top to bottom of panel 1

                       x.Width = rnd.Next(100, 400);
                       // give x a random width

                       score++;
                       // increase score by one
                    }

                if (player.Bounds.IntersectsWith(x.Bounds))
                 // if player collides with the x
                     {
                         gravity = 0;
                         // change gravity value to 0
                         player.Top = x.Top - player.Height;
                        //stop the player from going through the picture box and position it on top
                     }
                else
                     {
                         gravity = 8;
                        // if not then change gravity to 8
                     }
                 }
            }

                  if (player.Top + player.Height < 0 || player.Top > panel1.Height)
                     //if player top reaches top of the panel OR
                     // player drops to the bottom of the panel
                   {
                     gameTimer.Stop();
                    // stop the timer
                   }


        }
```

As you can see the timer event is Long so we have commented on the code to ensure you go through each line of code and see what they do. Few things to remember

First make sure you follow the code thoroughly and second make sure you follow the curly brackets for example the initial program has a open and closing curly brackets {} and then we have the if statements which also have two and the for loop has two as well. So make sure you follow it. Also ensure the semi colons are placed end of the lines.

Screen Shots



As you can see the game is working perfectly. Now you can try to add your own spin to the program and see how far you can go with it.