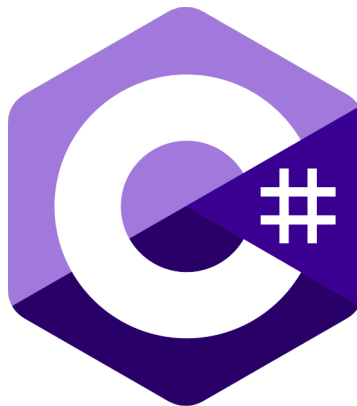


Архангельский колледж телекоммуникаций

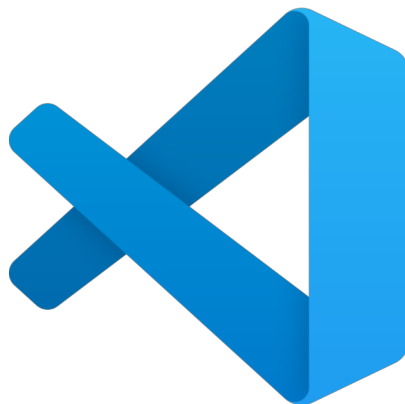
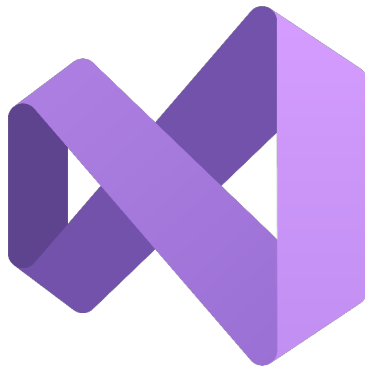
# **Разработка игры платформера на C#**

Абрамова Полина Александровна

# Языки программирования



# Среды программирования



# Visual Studio

Это комплексная интегрированная среда разработки (IDE), которую можно использовать для записи, редактирования, отладки и сборки кода, а затем развертывания приложения. Visual Studio включает компиляторы, средства завершения кода, управление версиями, расширения и многие другие функции для улучшения каждого этапа процесса разработки программного обеспечения.

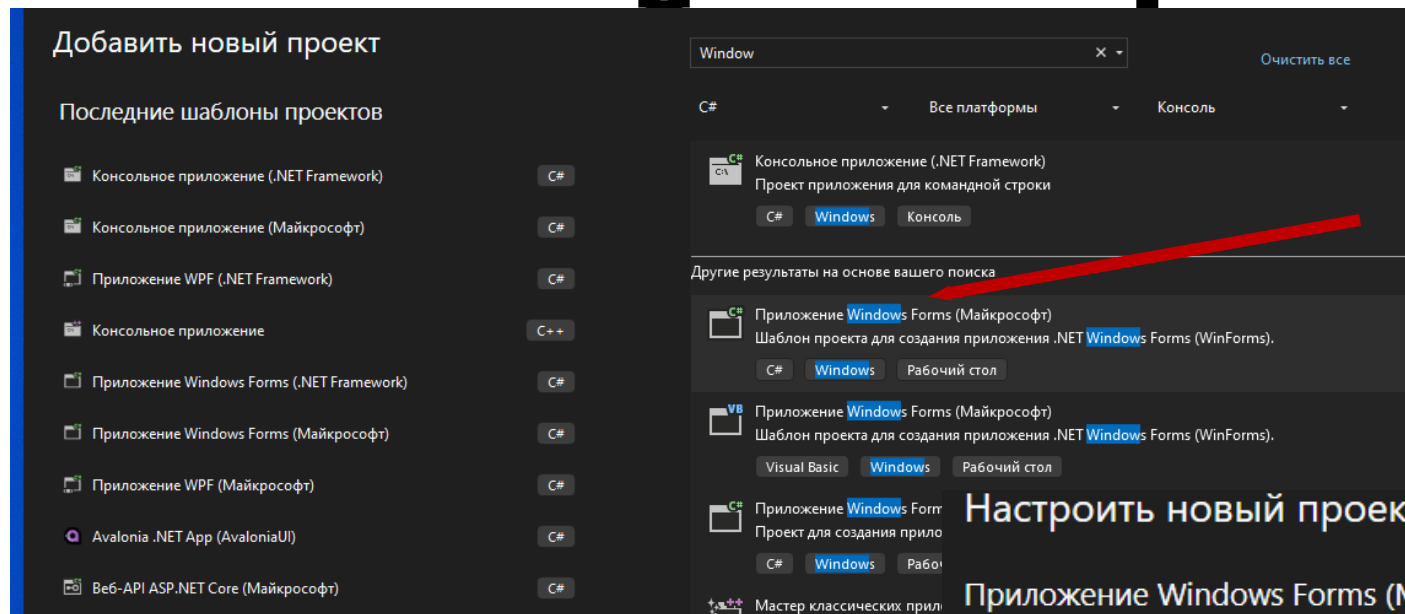
# **Жанр платформер**

В платформерах игровой процесс построен вокруг прыжков. Это может быть акробатика с видом сбоку или от третьего лица. Вы скачете по тем самым платформам, совершаете рывки, отталкиваетесь от стен, собираете бонусы. Запас жизней обычно ограничен, потеряли все — откатываетесь к старту уровня. Чтобы пополнить количество жизней, необходимо хватать какие-нибудь колечки и фрукты. Путь к цели преграждают пропасти и противники.

# Платформеры



# Создание проекта



## Настроить новый проект

### Приложение Windows Forms (Майкрософт)

C# Windows Рабочий стол

Имя проекта

Platformer

Расположение

C:\Users\221\source\repos\ConsoleApp10

Проект будет создан в "C:\Users\221\source\repos\ConsoleApp10\Platformer"

# Дополнительные сведения

Приложение Windows Forms (Майкрософт)

C#

Windows

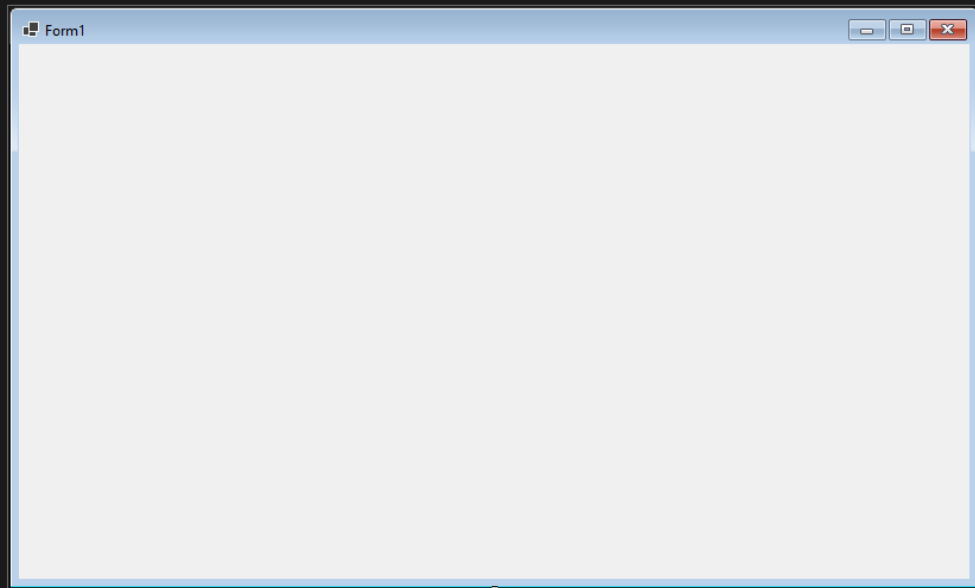
Рабочий стол

Платформа ⓘ

.NET 8.0 (долгосрочная поддержка)

Form1.cs [Конструирование] ✕

Form1





Панель элементов

Поиск по панели элементов

Все формы Windows Forms

- Указатель
- BackgroundWorker
- BindingSource
- Button
- CheckBox
- CheckedListBox
- ColorDialog
- ComboBox
- ContextMenuStrip
- DataGridView
- DateTimePicker
- DomainUpDown
- ErrorProvider
- FileSystemWatcher
- FlowLayoutPanel
- FolderBrowserDialog
- FontDialog
- GroupBox
- HelpProvider
- HScrollBar

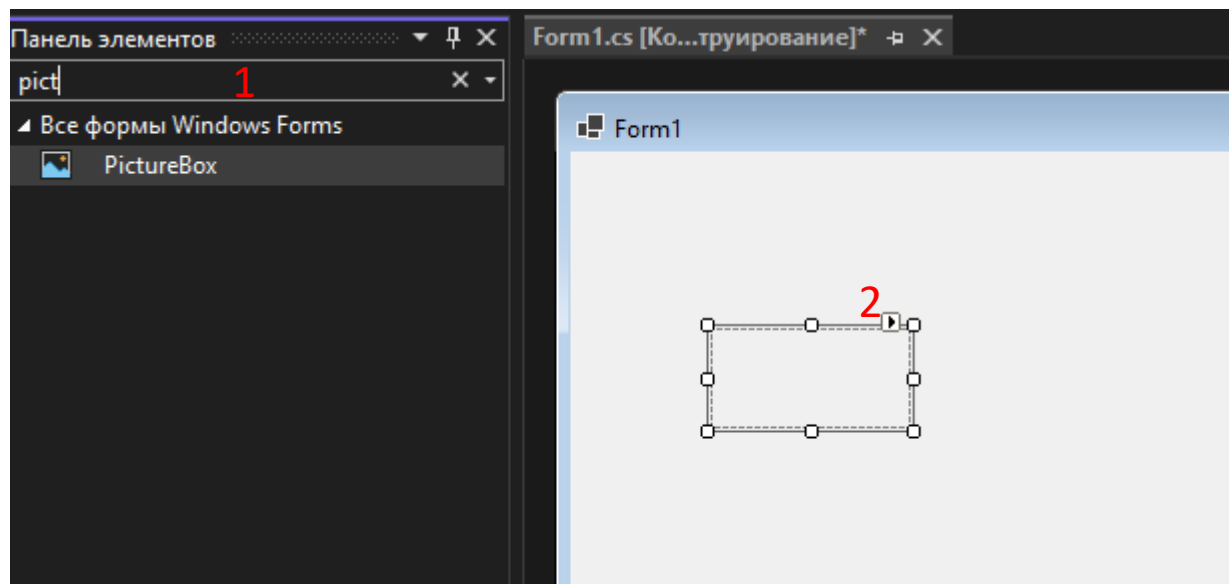
Источники данных

Form1.cs [Конструирование]

Form1

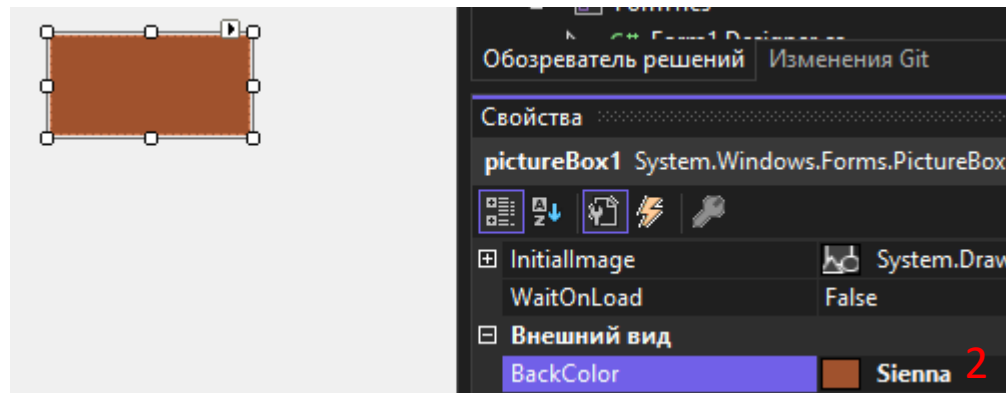
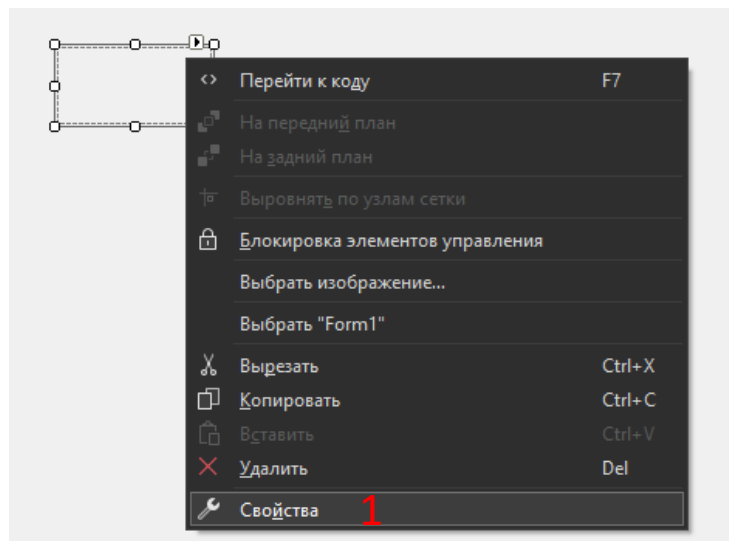
# Платформа

Добавить на форму PictureBox



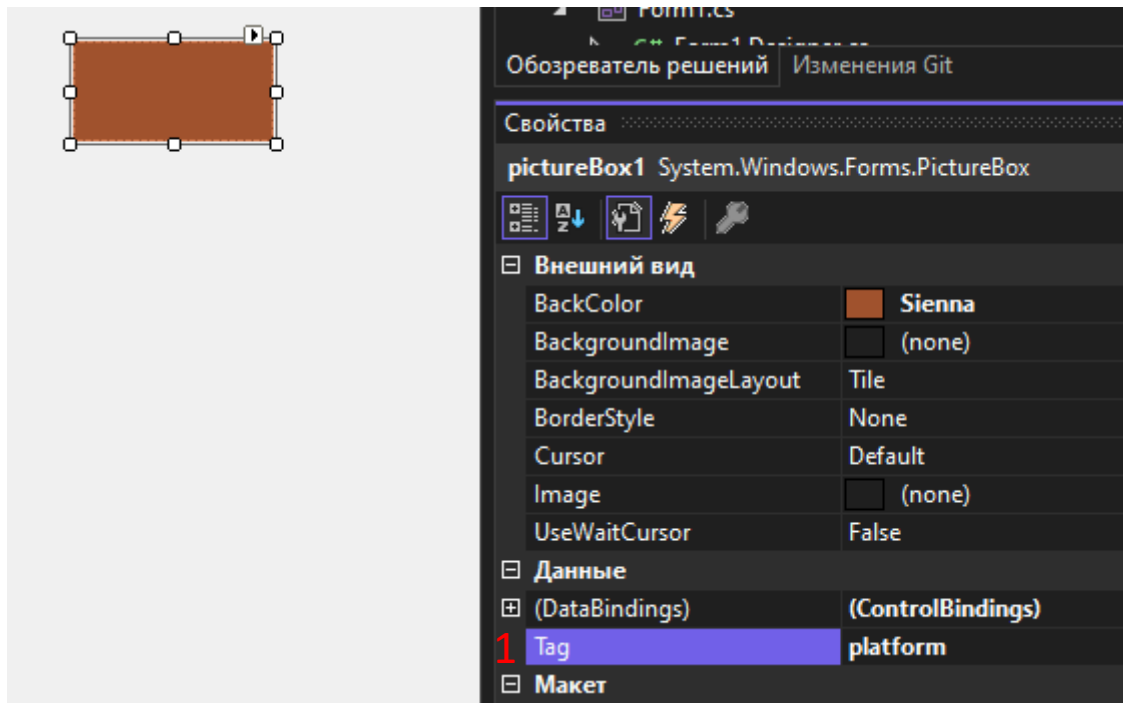
# Платформа

## Изменить фоновый цвет элемента



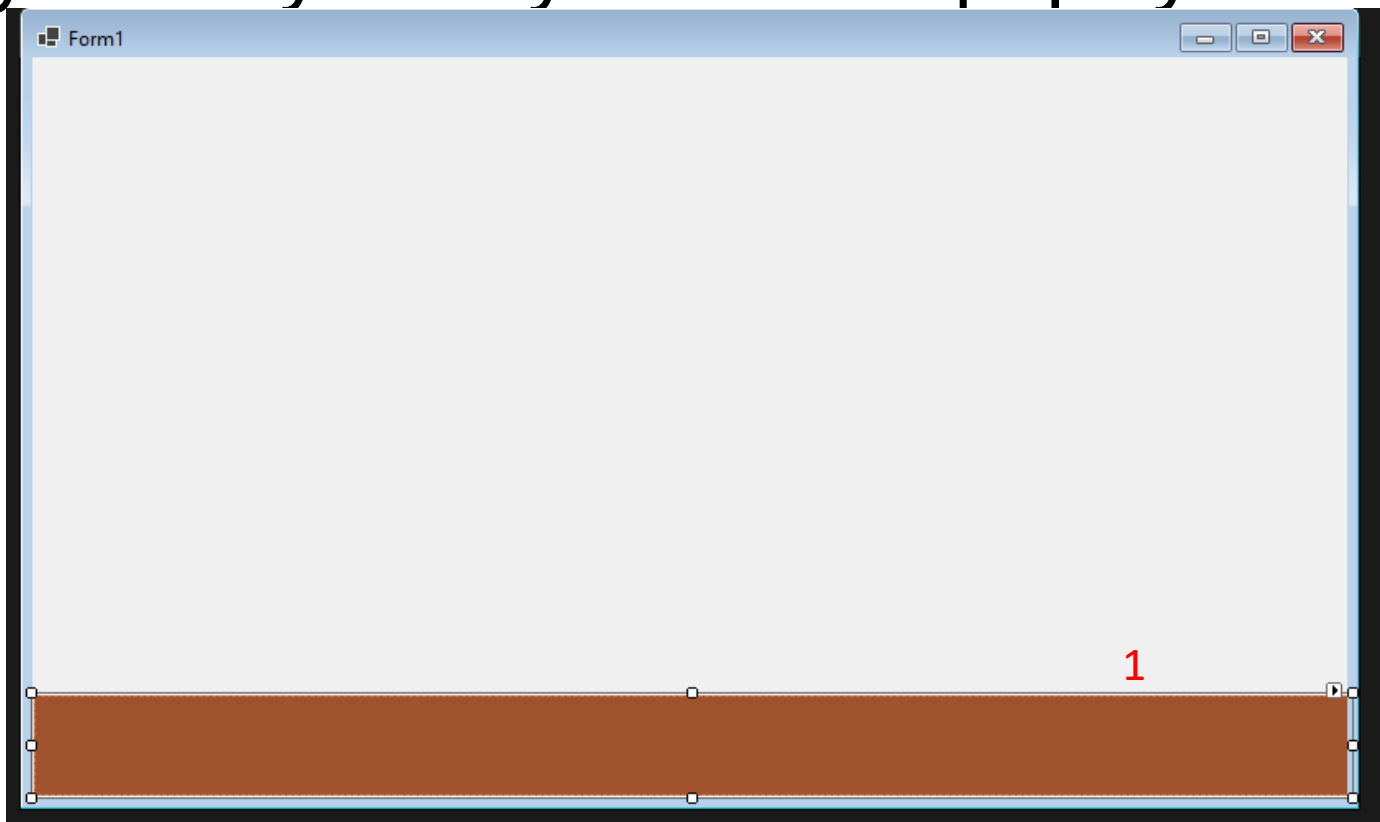
# Платформа

Установить Tag = platform



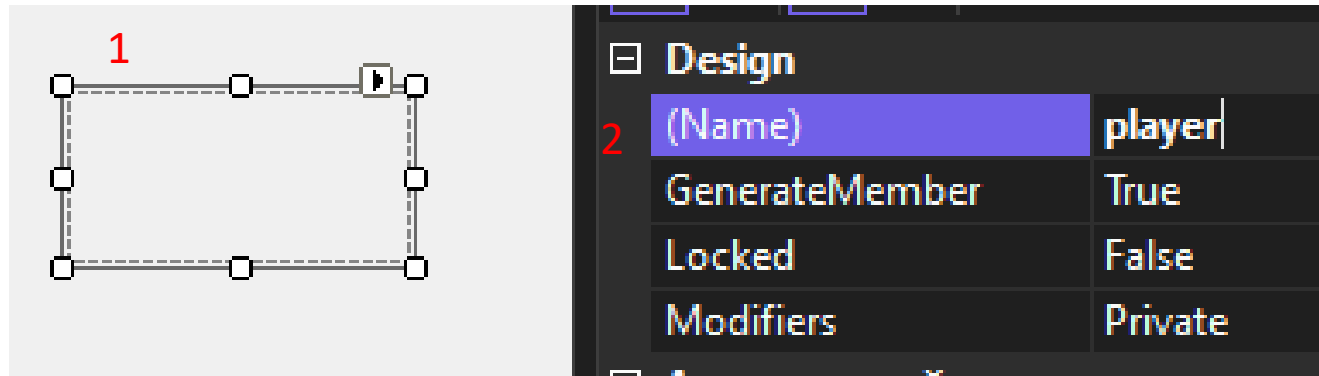
# Платформа

Растянуть получившуюся платформу



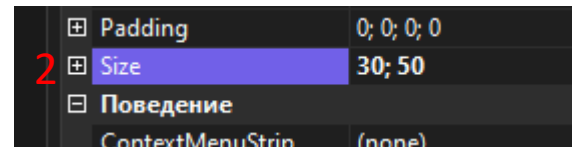
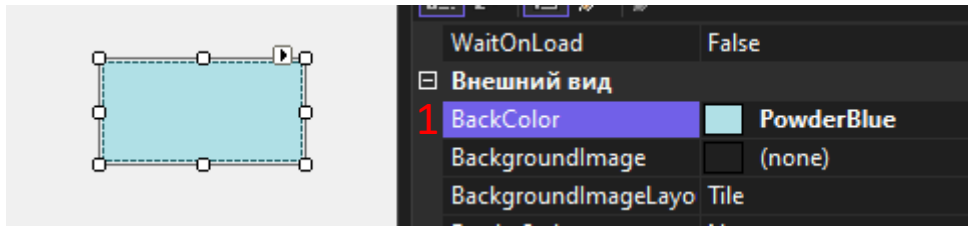
# Персонаж

Свойство Name = player



# Персонаж

Цвет установить любой, размер 30x50.

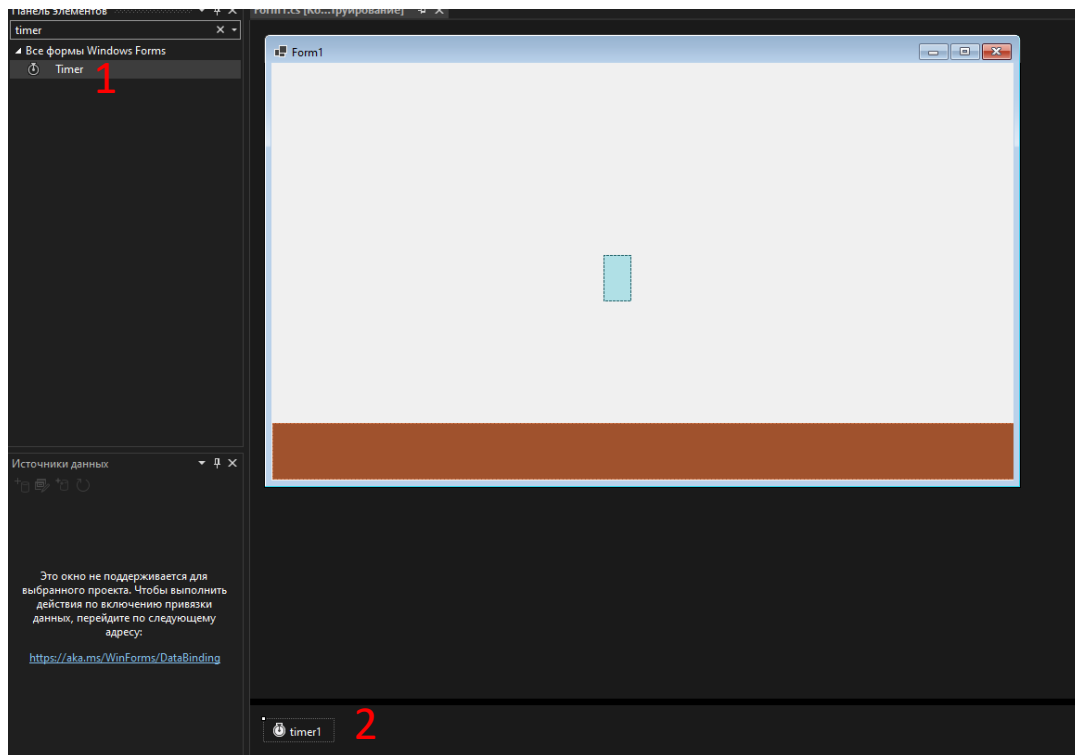






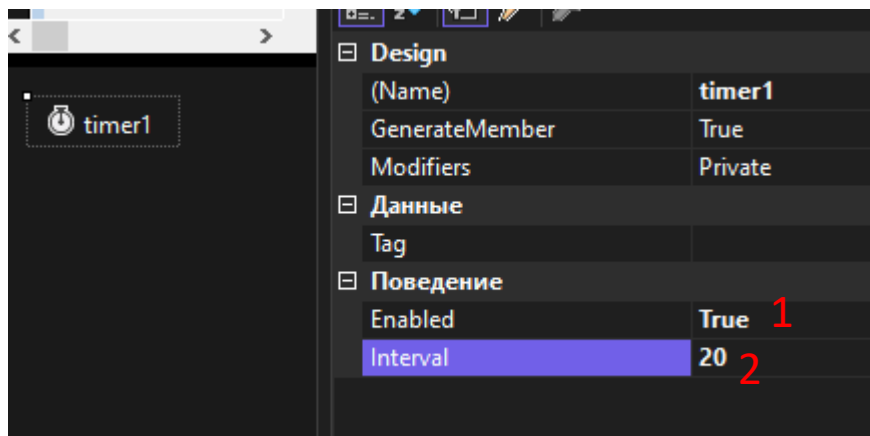
# Таймер

Перетащить на форму таймер (отобразится снизу)



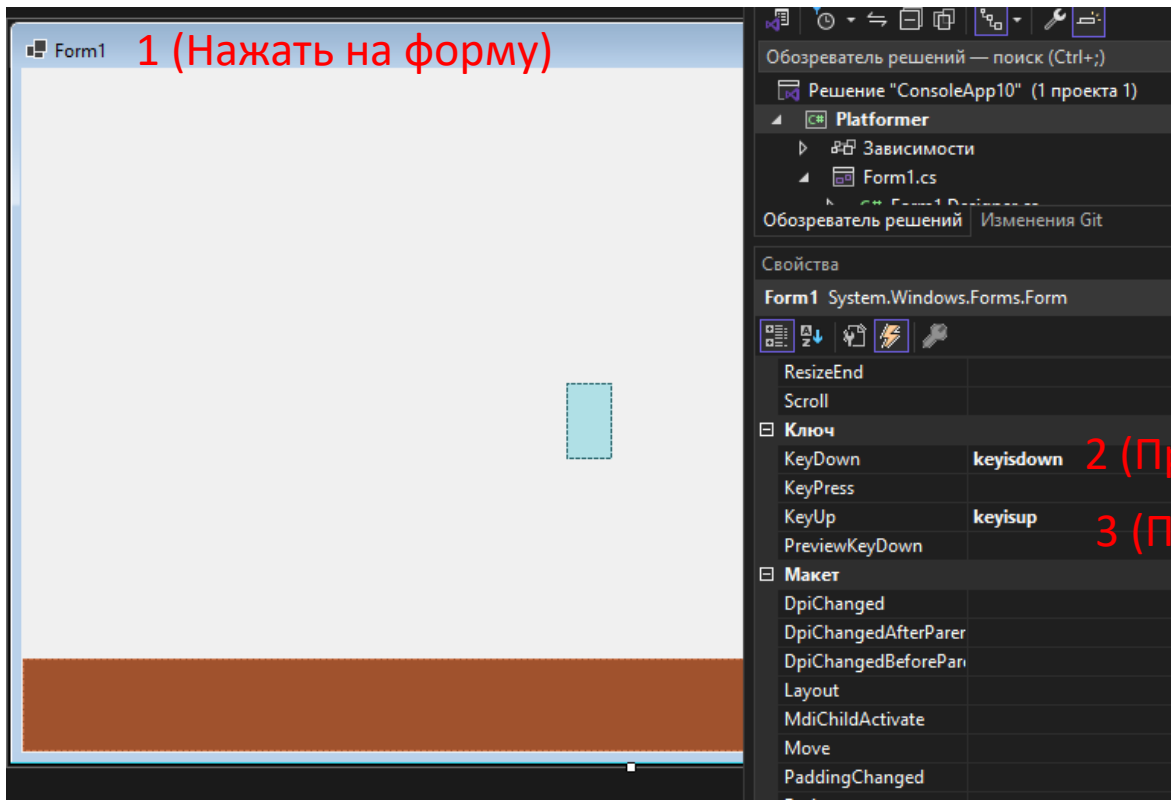
# Таймер

Свойства Enabled = True Interval = 20.



# Ножотия но клавиши

## Добавить обработчики событий



The screenshot shows the Visual Studio IDE with a Windows Form named 'Form1' on the left and the 'Solution Explorer' and 'Properties' windows on the right. The form has a light blue background and a small blue rectangle in the center. The 'Solution Explorer' shows the project structure, including 'Form1.cs'. The 'Properties' window shows the 'Form1' object of type 'System.Windows.Forms.Form'. The 'Events' tab is selected, showing a list of events and their handlers. The 'KeyDown' event is highlighted, and the handler 'keyisdown' is entered. The 'KeyUp' event is also highlighted, and the handler 'keyisup' is entered. Red annotations are present: '1 (Нажать на форму)' points to the form, '2 (Просто написать название)' points to 'keyisdown', and '3 (Просто написать название)' points to 'keyisup'.

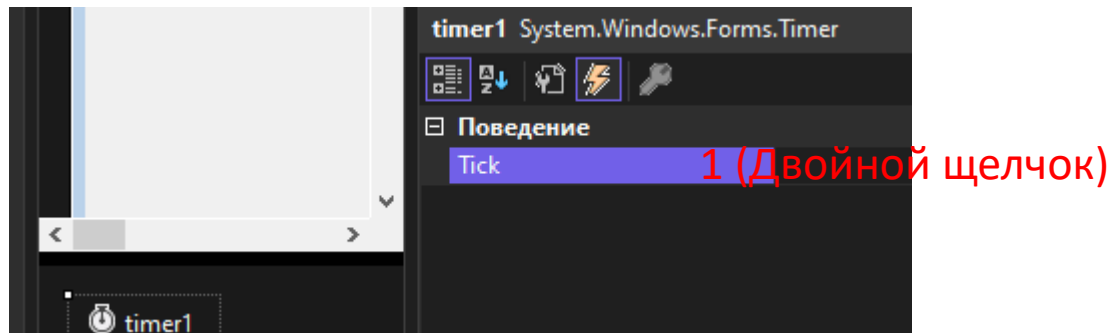
1 (Нажать на форму)

2 (Просто написать название)

3 (Просто написать название)

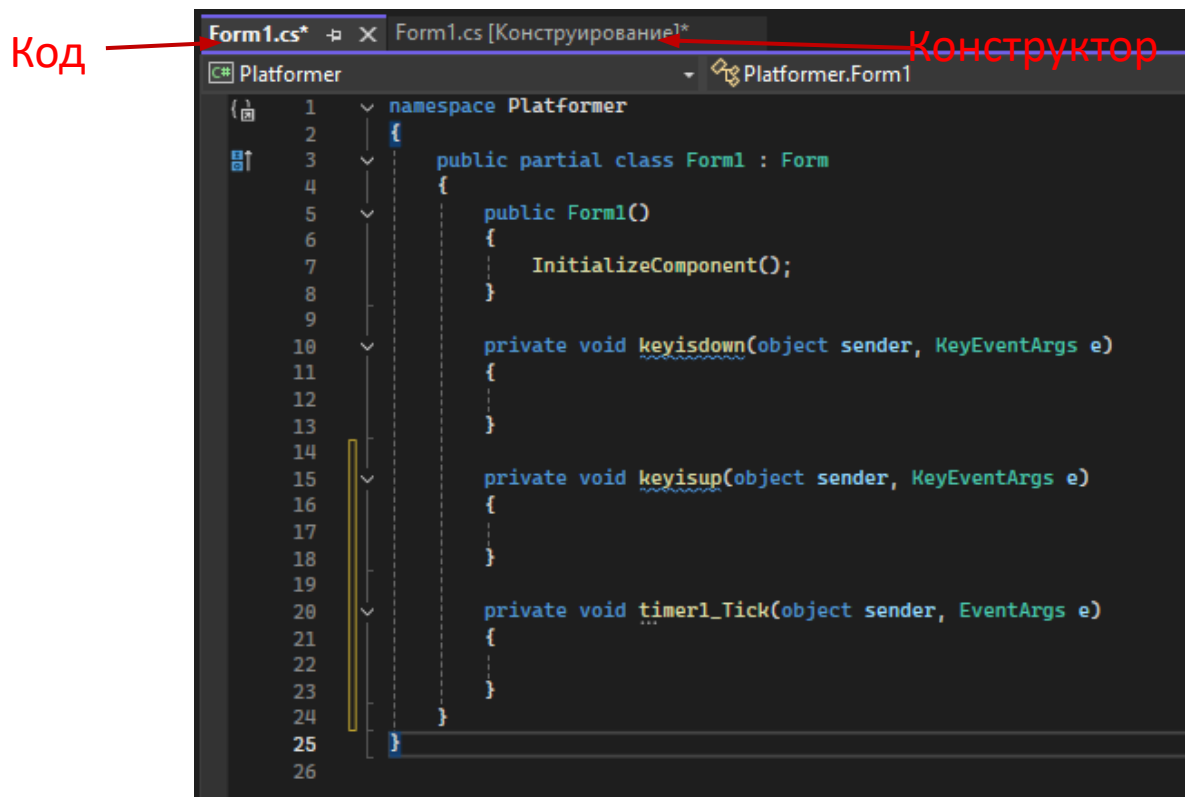
# Тики таймера

Добавить обработчики таймера



# Итог

На данном этапе код будет выглядеть так:



# Переменные

Добавить в код следующие переменные:

```
bool goleft = false;  
bool goright = false;  
bool jumping = false;  
int jumpSpeed = 10;  
int force = 8;  
int score = 0;
```

# Передвижение персонажа

Внутри метода `keyisdown` добавить:

```
if (e.KeyCode == Keys.Left) {  
    goleft = true;  
}  
  
if (e.KeyCode == Keys.Right) {  
    goright = true;  
}  
  
if (e.KeyCode == Keys.Space && !jumping) {  
    jumping = true;  
}
```

# Передвижение персонажа

Внутри метода `keyisup` добавить:

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



# Передвижение персонажа

В timer1\_Tick прописать:

```
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;  
}
```

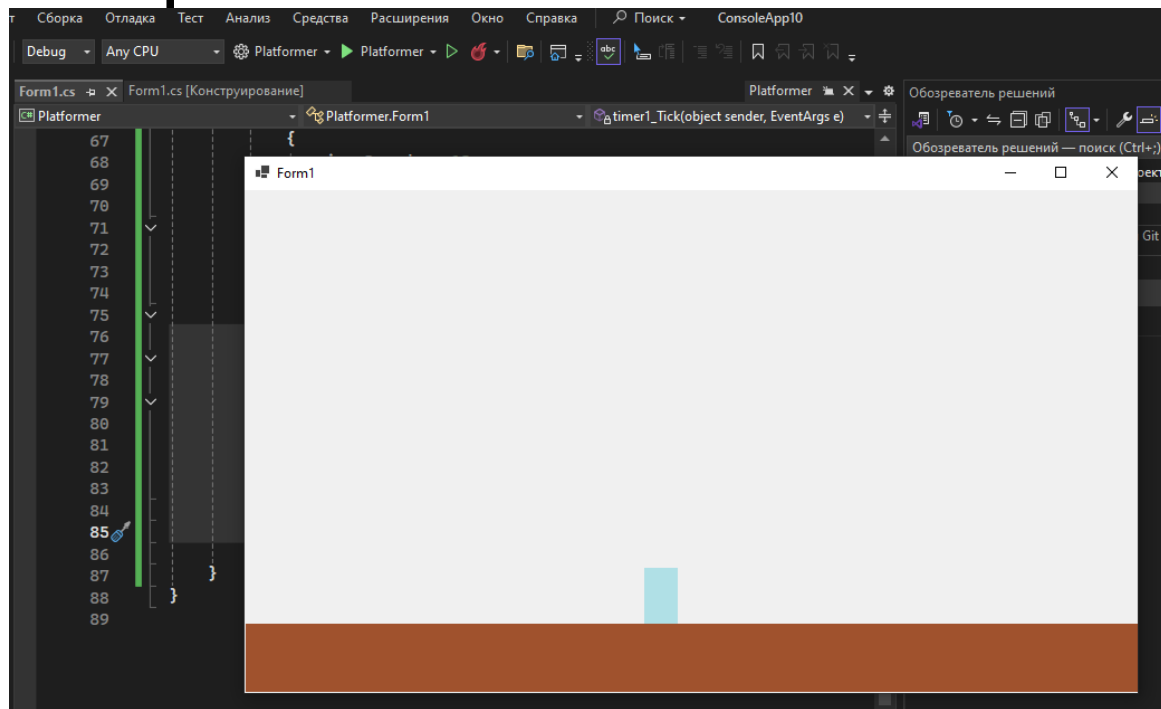
# Передвижение персонажа

В timer1\_Tick также добавить:

```
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;  
        }  
    }  
}
```

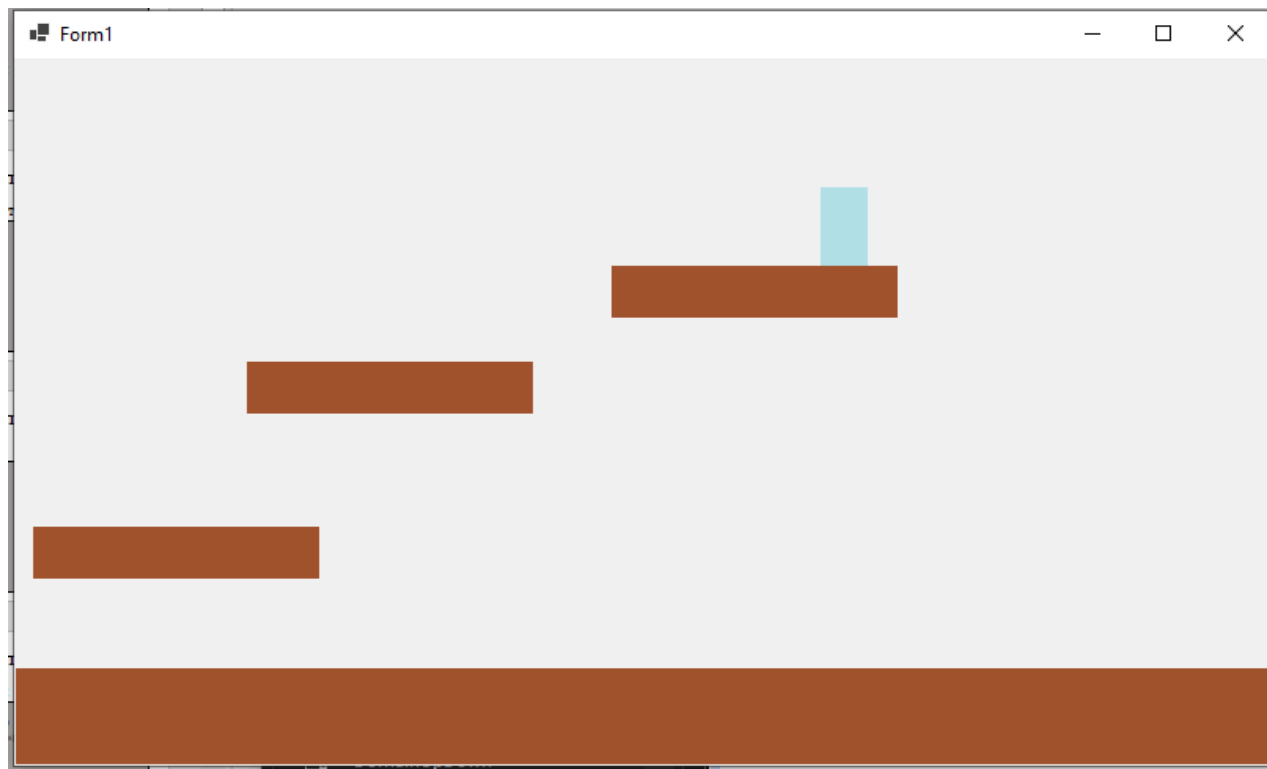
# Тест программы

Запустите программу (Ctrl + F5) и попробуйте управлять персонажем



# Усложнение игры

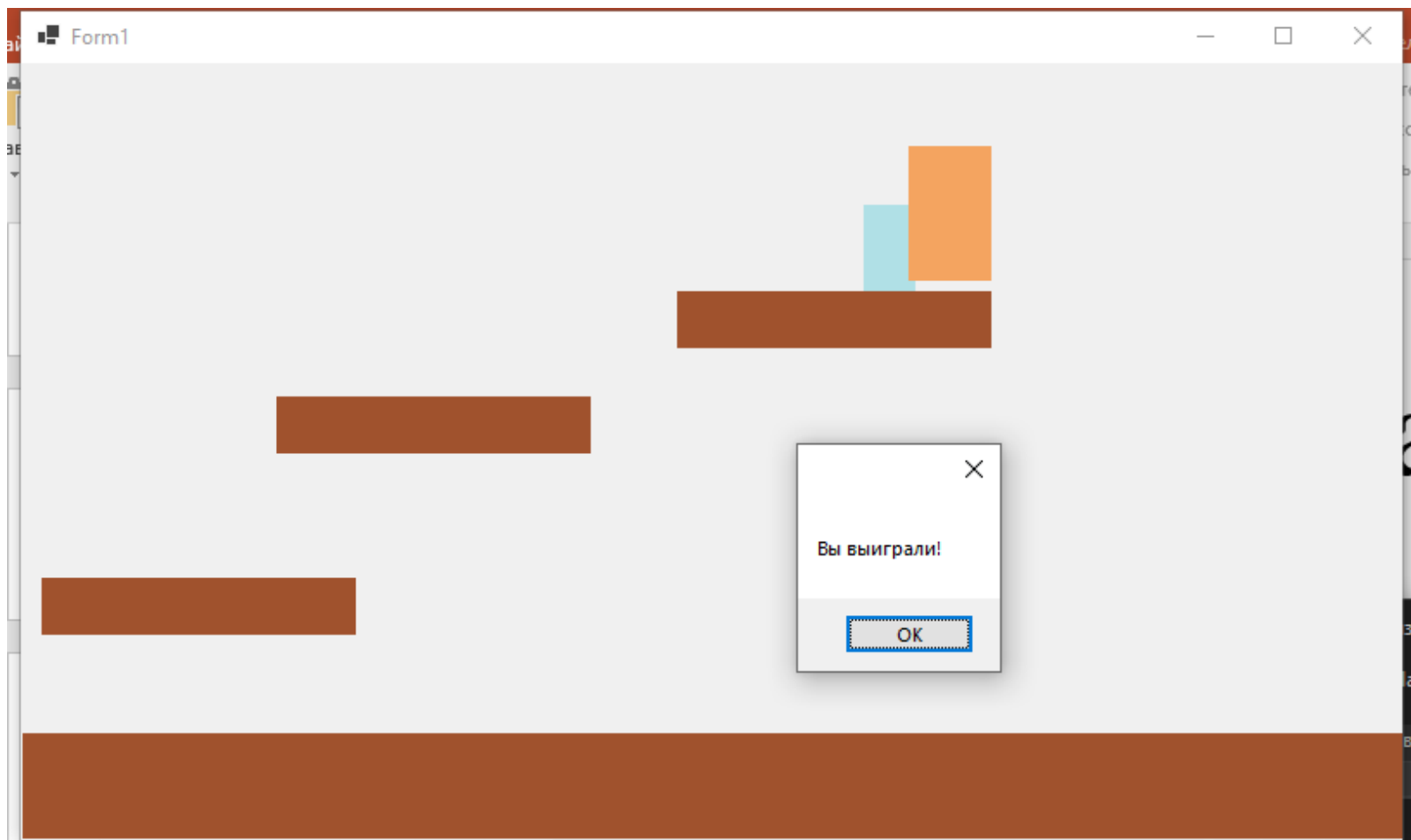
Добавьте еще несколько платформ (можно копировать).



# Завершение уровня

Добавьте еще один pictureBox, Name = door, BackColor = любой.

# Завершение уровня

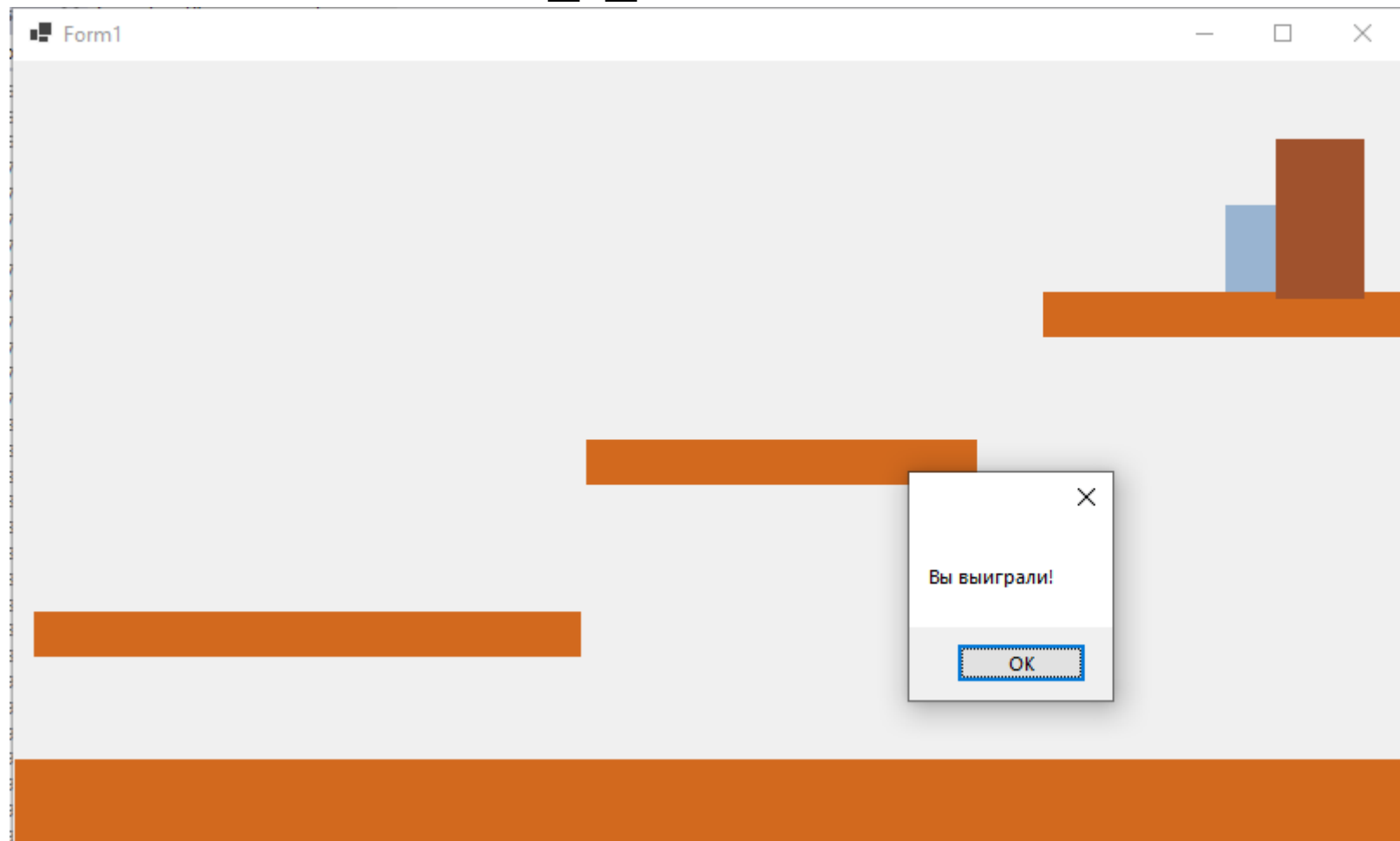


# Завершение уровня

В timer1\_Tick добавить:

```
if(player.Bounds.IntersectsWith(door.Bounds))  
{  
    timer1.Stop();  
    MessageBox.Show("Вы выиграли!");  
}
```

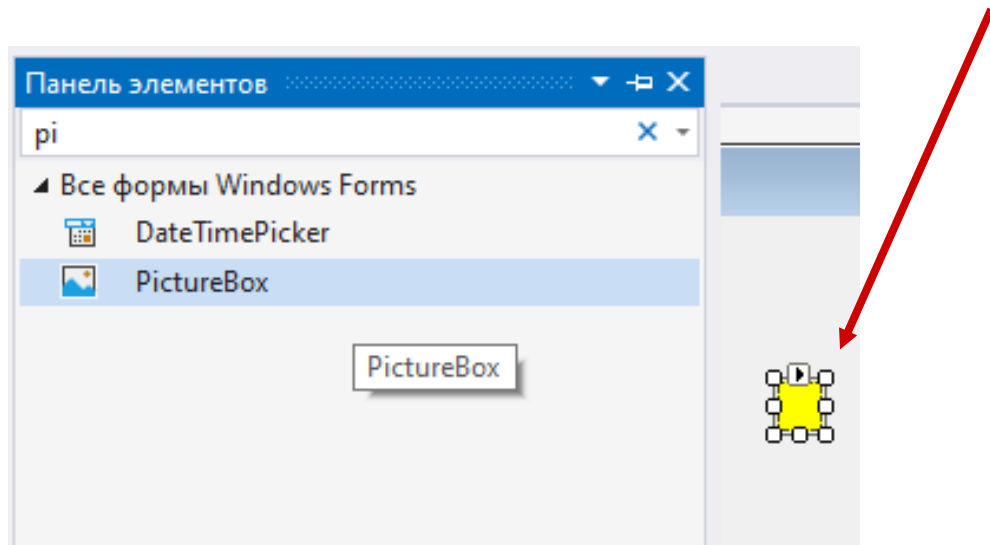
# Результат





# Монетки

Перетащить на форму pictureBox, BackColor = желтый, Tag = coin.



# Монетки

Внутри цикла foreach в timer1\_Tick добавить:

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

# Монетки

timer1\_Tick

Код выглядит так:

Цикл foreach (в него  
вставляем код)

Синим выделен код,  
который нужно  
вставить

```
private void timer1_Tick(object sender, EventArgs e)
{
    player.Top += jumpSpeed;
    if (jumping && force < 0) ...
    if (goleft) ...
    if (goright) ...
    if (jumping) ...
    else ...
    foreach (Control x in this.Controls)
    {
        if (x is PictureBox && x.Tag == "platform")
        {
            if (player.Bounds.Intersects(x.Bounds) && !jumping)
            {
                force = 8;
                player.Top = x.Top - player.Height;
            }
        }

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

        if (player.Bounds.Intersects(door.Bounds))
        {
            ...
        }
    }
}
```

# Тест игры

Запустите игру, проверьте правильность работы.



# **Что можно улучшить?**

- Враги
- Здоровье
- Различные препятствия
- Способности
- Новые уровни