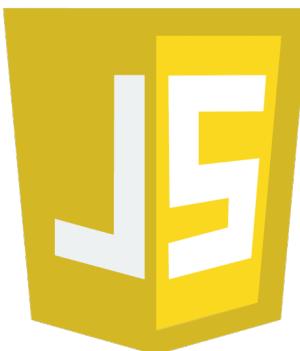
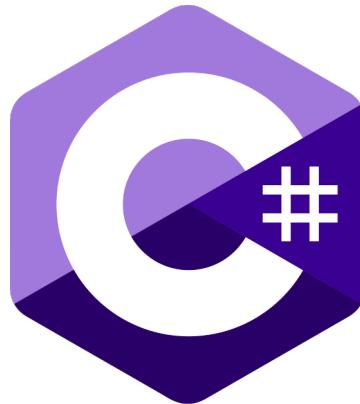


Архангельский колледж телекоммуникаций

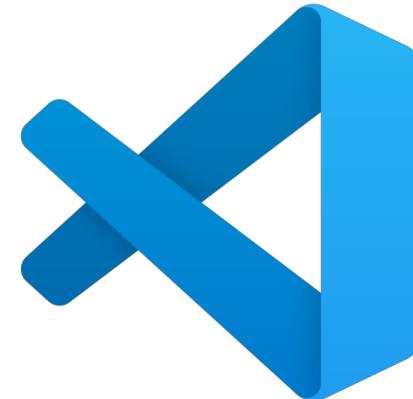
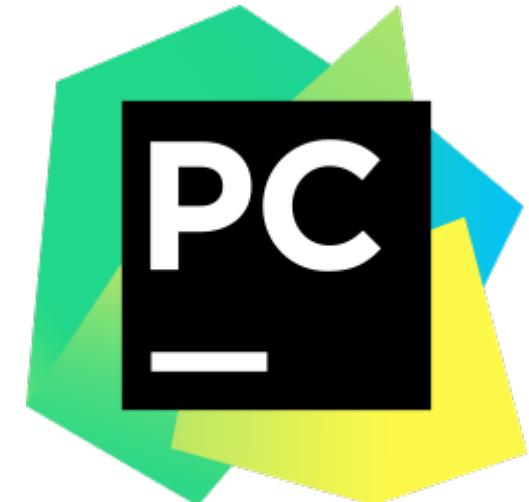
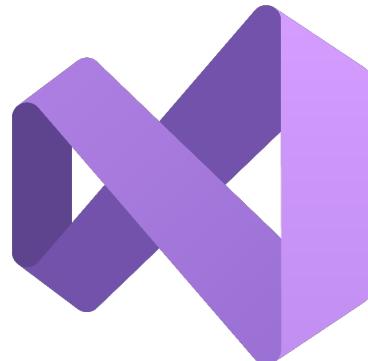
Разработка игры платформера на С#

Абрамова Полина Александровна

Языки программирования



Среды программирования



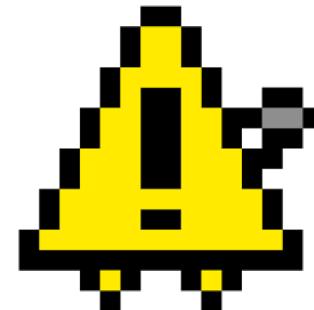
Visual Studio

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

Добавить новый проект

Последние шаблоны проектов

Консольное приложение (.NET Framework)

C#

Консольное приложение (Майкрософт)

C#

Приложение WPF (.NET Framework)

C#

Консольное приложение

C++

Приложение Windows Forms (.NET Framework)

C#

Приложение Windows Forms (Майкрософт)

C#

Приложение WPF (Майкрософт)

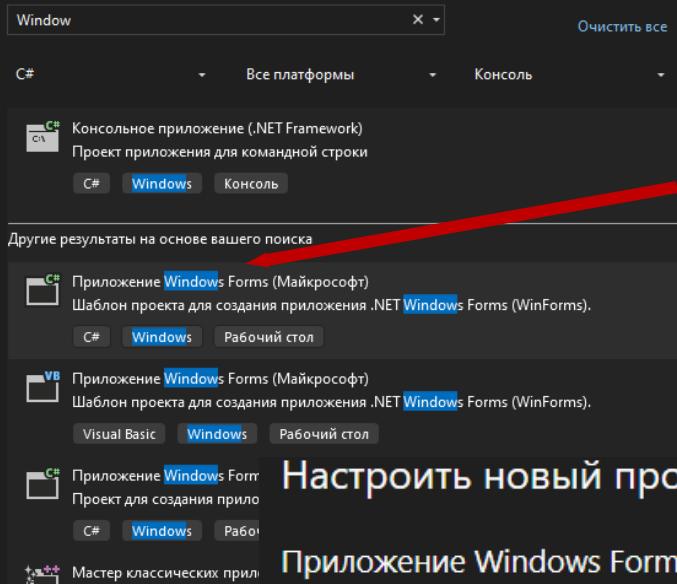
C#

Avalonia .NET App (AvaloniaUI)

C#

Веб-API ASP.NET Core (Майкрософт)

C#



Настройте 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#

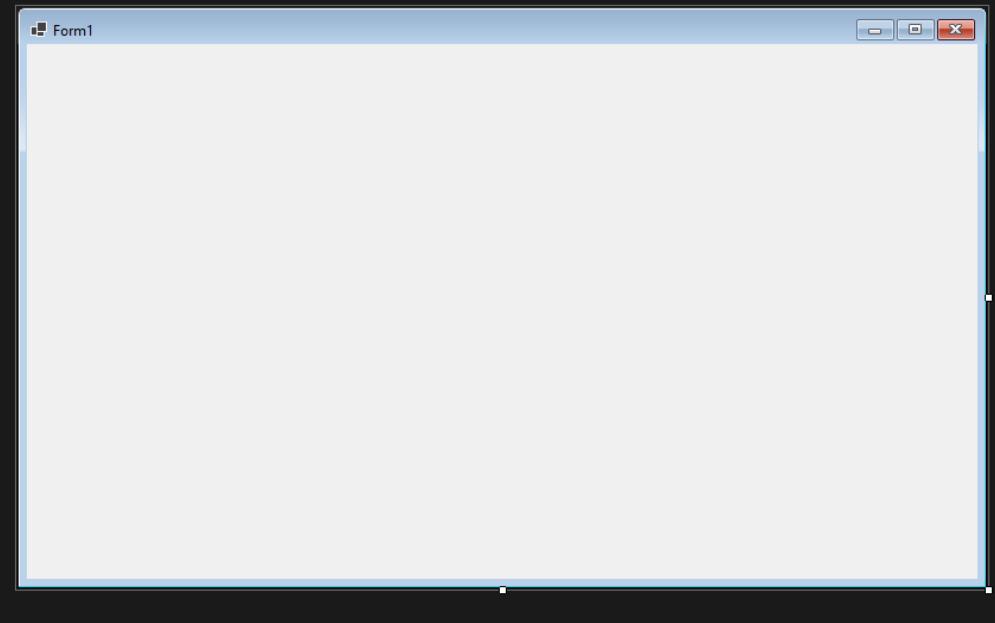
Windows

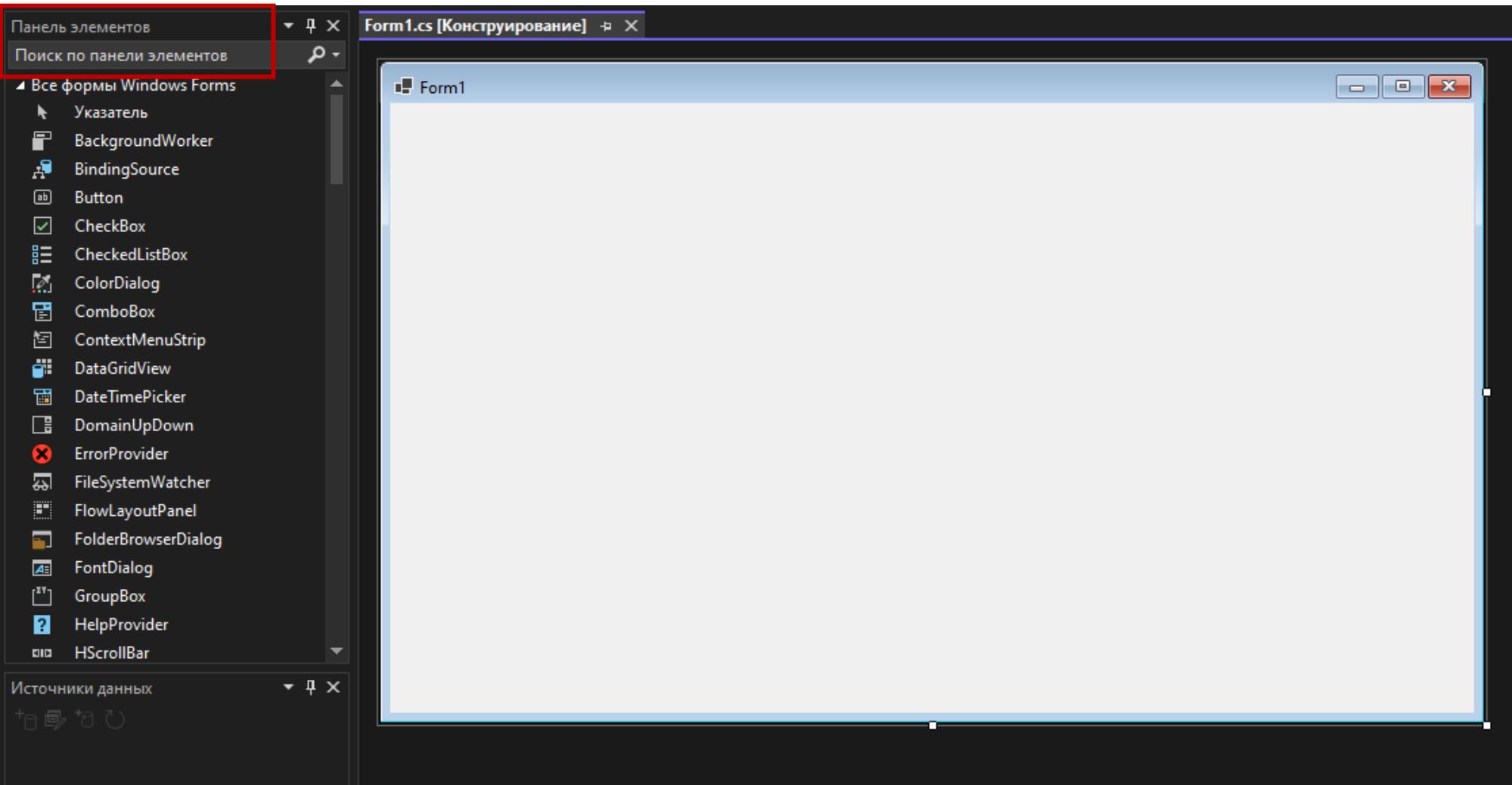
Рабочий стол

Платформа ⓘ

.NET 8.0 (долгосрочная поддержка)

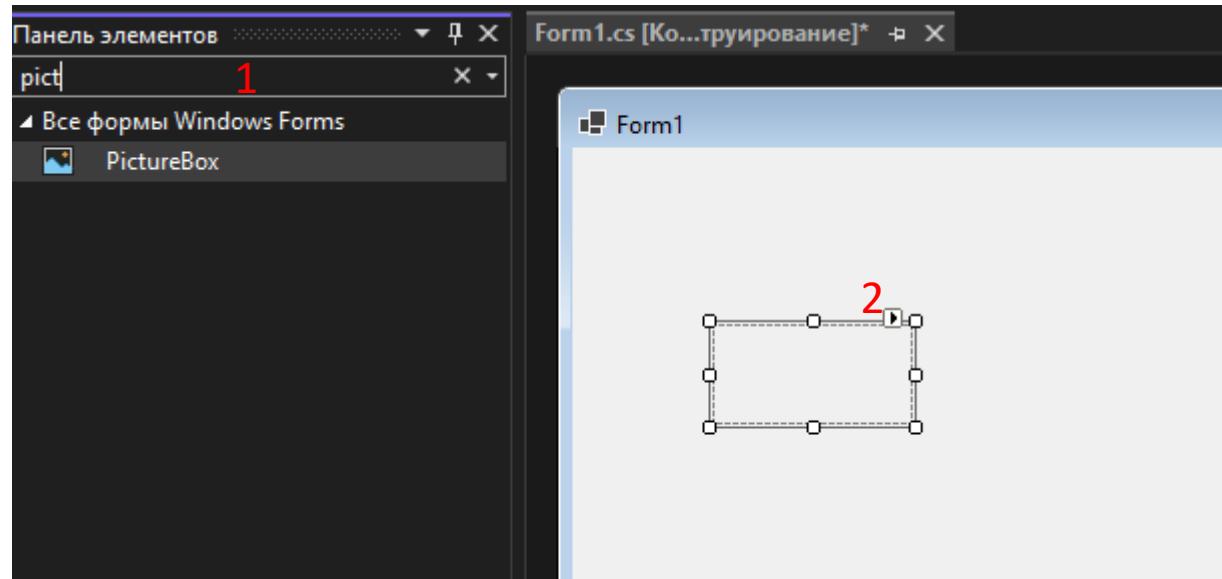
Form1.cs [Конструирование] + X





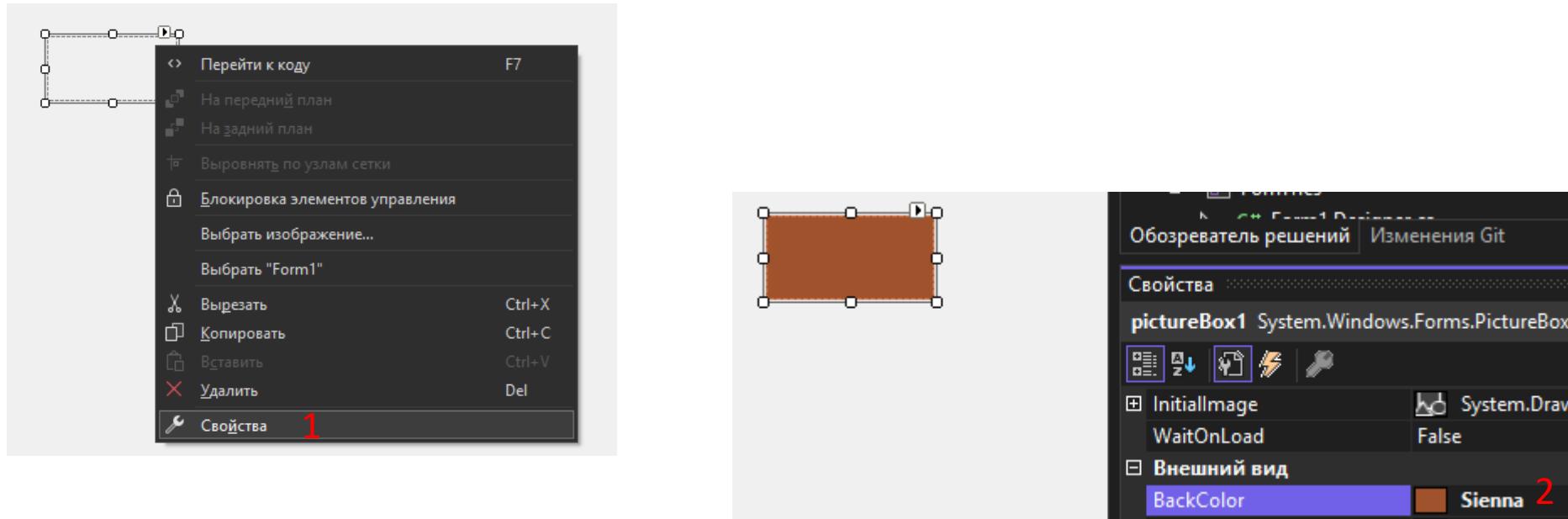
Платформа

Добавить на форму PictureBox



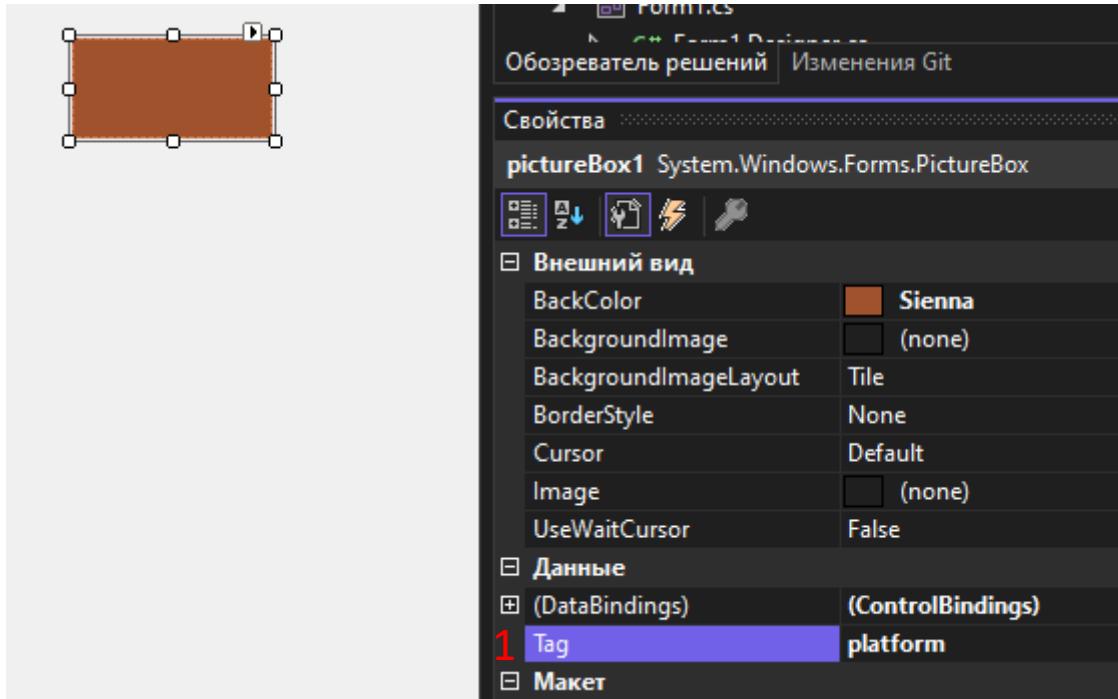
Платформа

Изменить фоновый цвет элемента



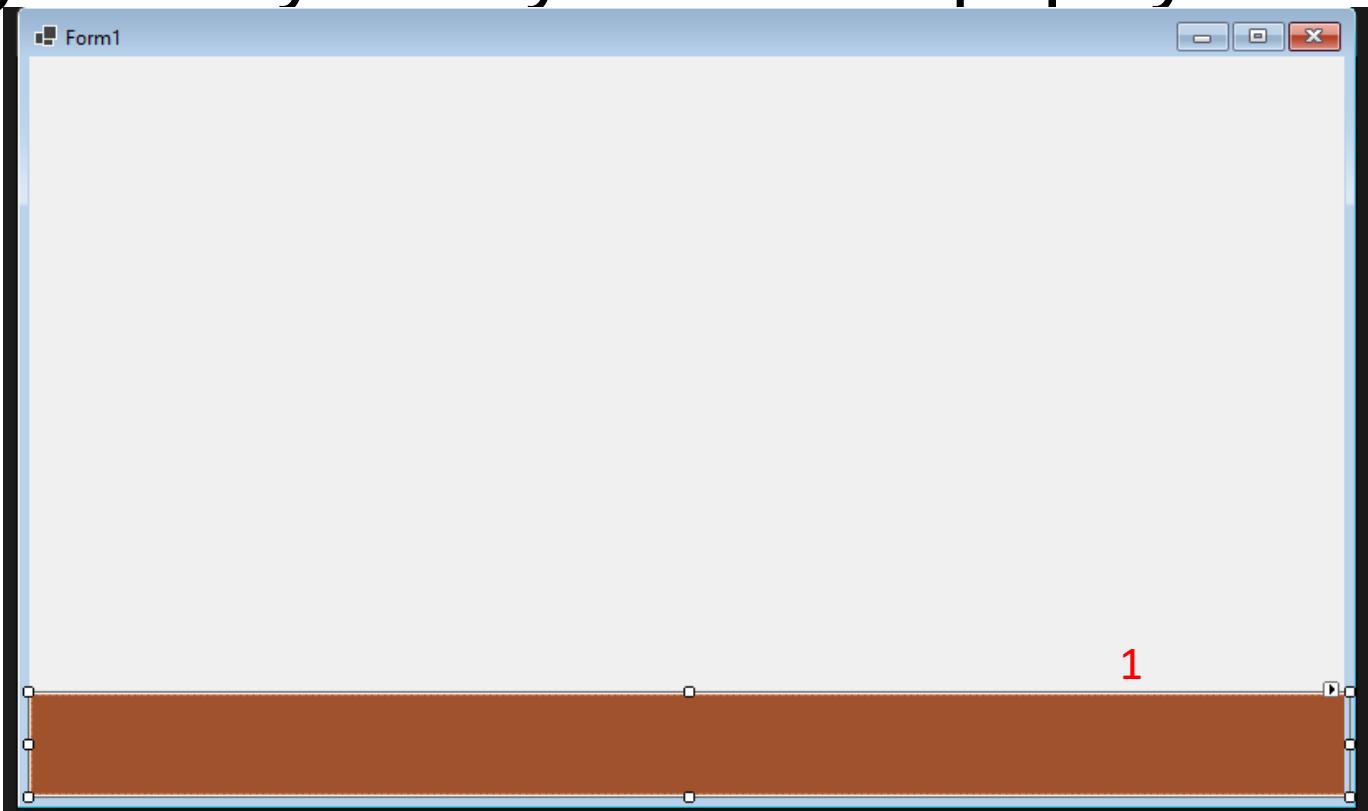
Платформа

Установить Tag = platform



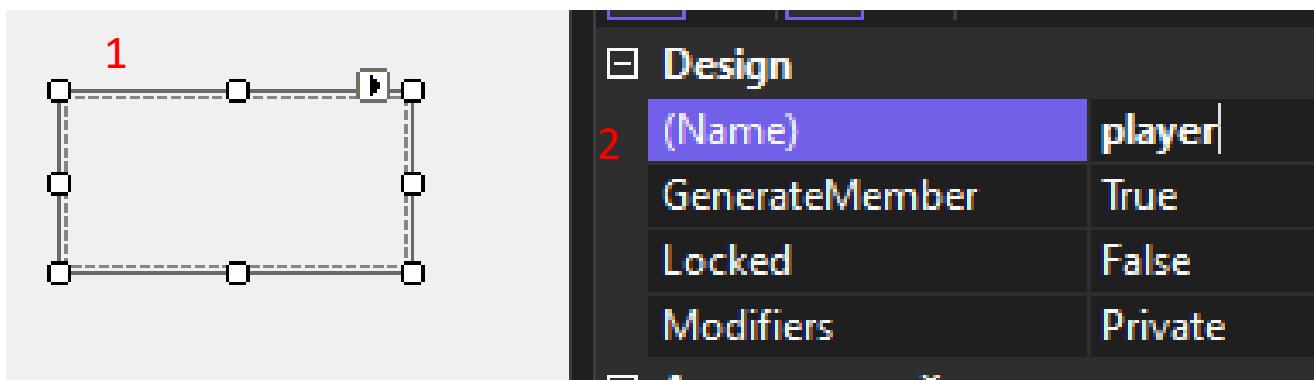
Платформа

Растянуть получившуюся платформу



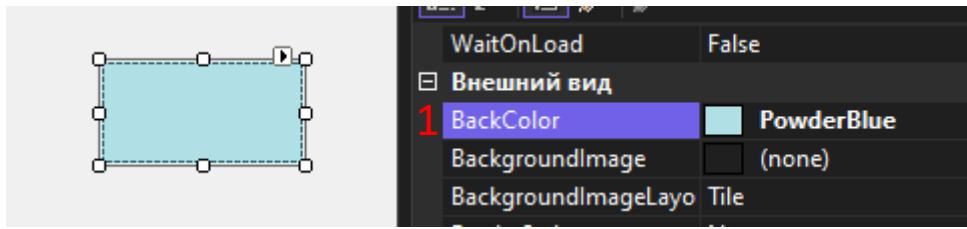
Персонаж

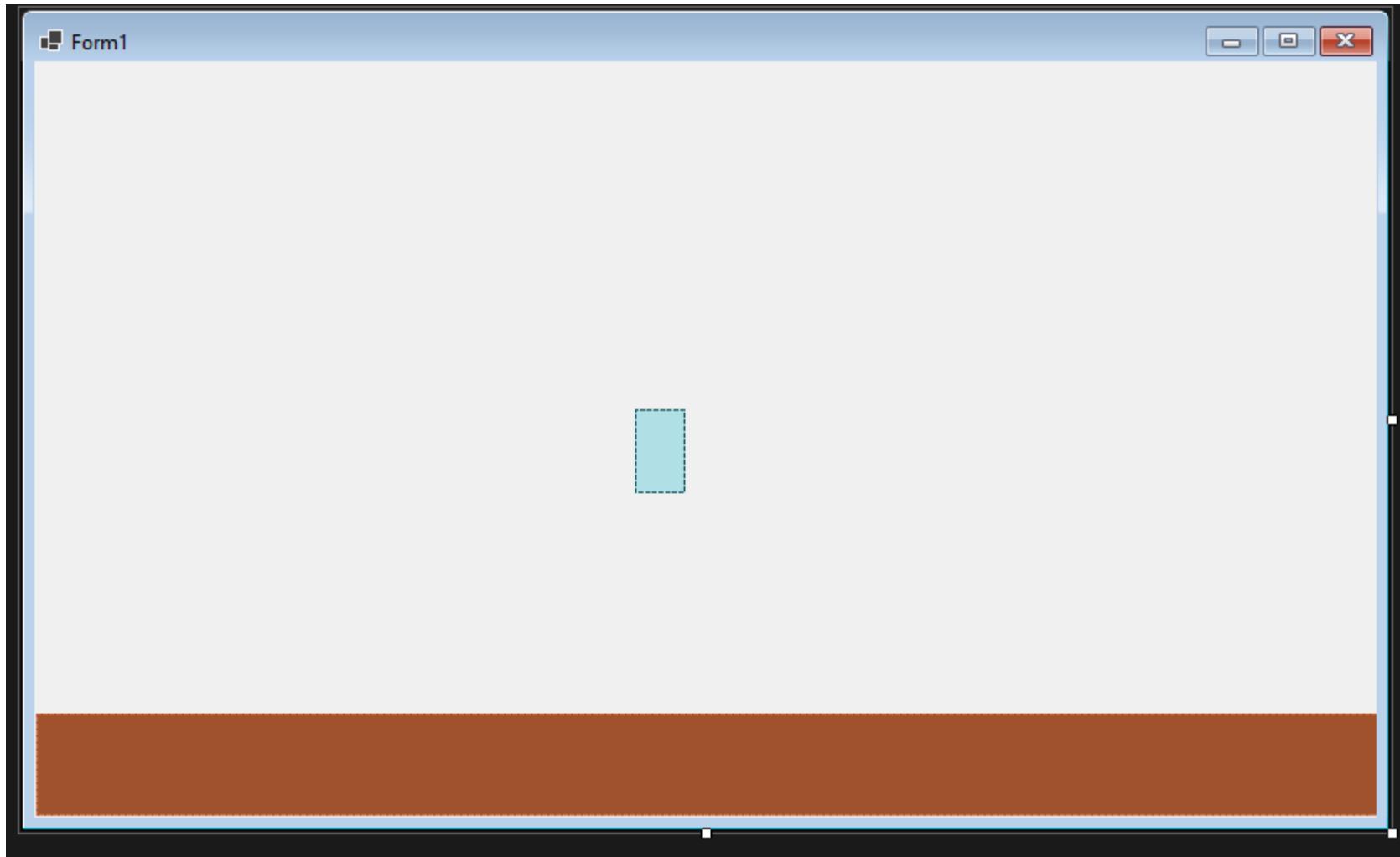
Свойство Name = player



Персонаж

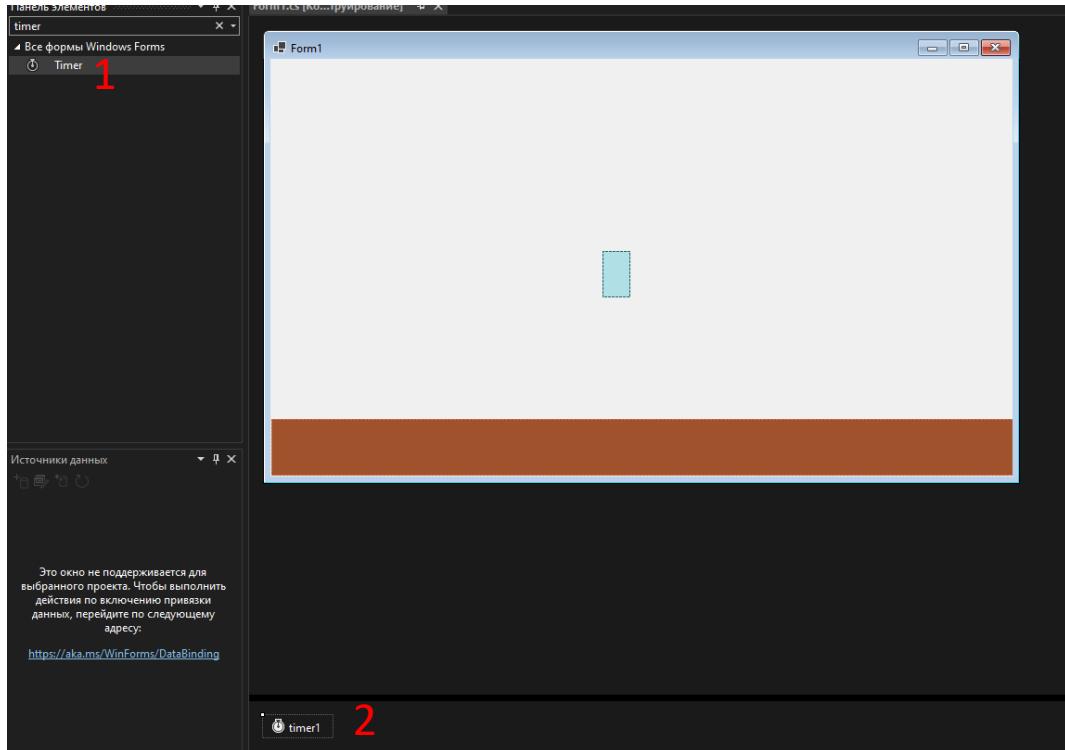
Цвет установить любой, размер 30x50.





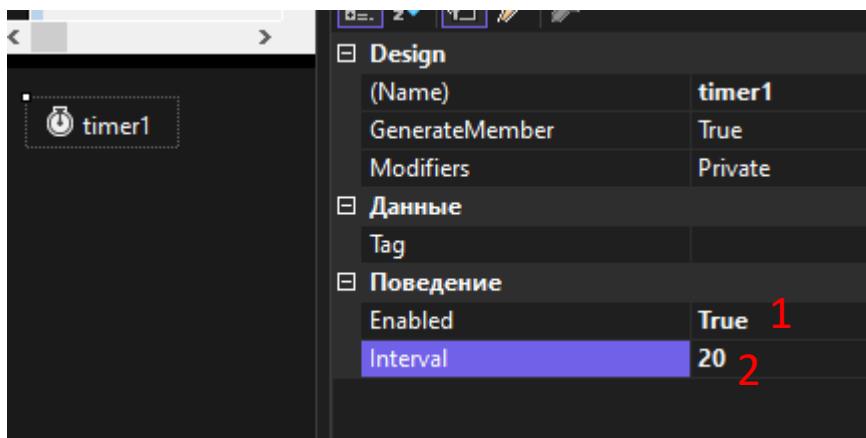
Таймер

Перетащить на форму таймер (отобразится снизу)



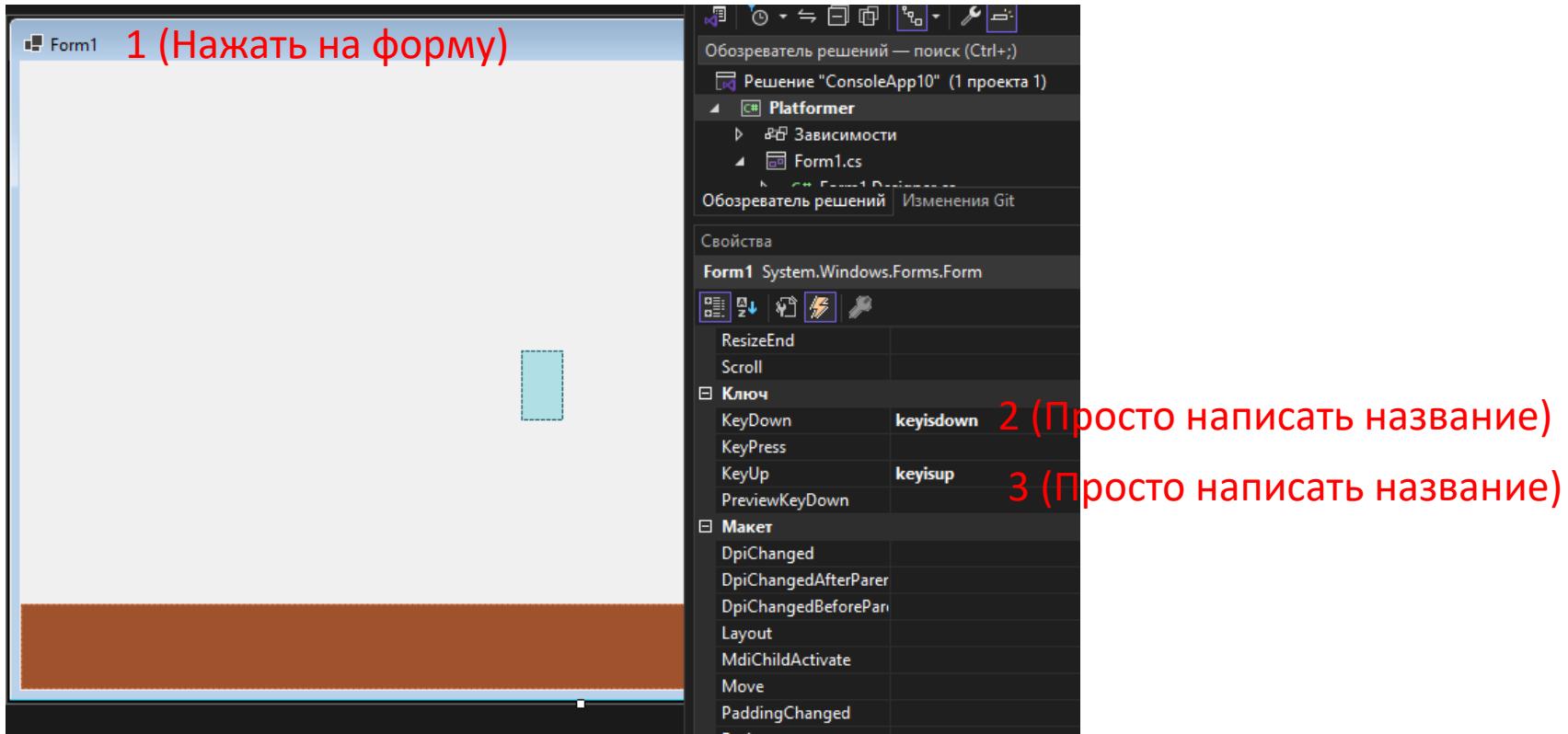
Таймер

Свойства Enabled = True Interval = 20.



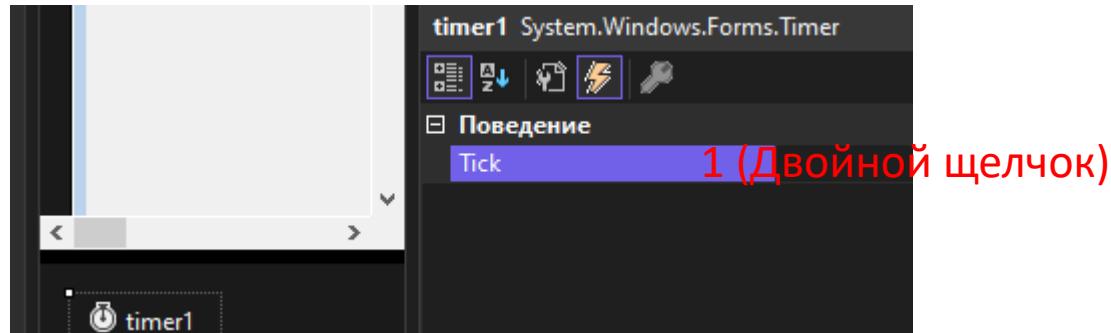
Нажатия на клавиши

Добавить обработчики событий



Тики таймера

Добавить обработчики таймера



Итог

На данном этапе код будет выглядеть так:

Код → Конструктор

```
Form1.cs*  Form1.cs [Конструирование]*  Platformer
namespace Platformer
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void keyisdown(object sender, KeyEventArgs e)
        {

        }

        private void keyisup(object sender, KeyEventArgs e)
        {

        }

        private void timer1_Tick(object sender, EventArgs e)
        {

    }
}
```

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

Внутрь 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

Внутрь 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;
}
```

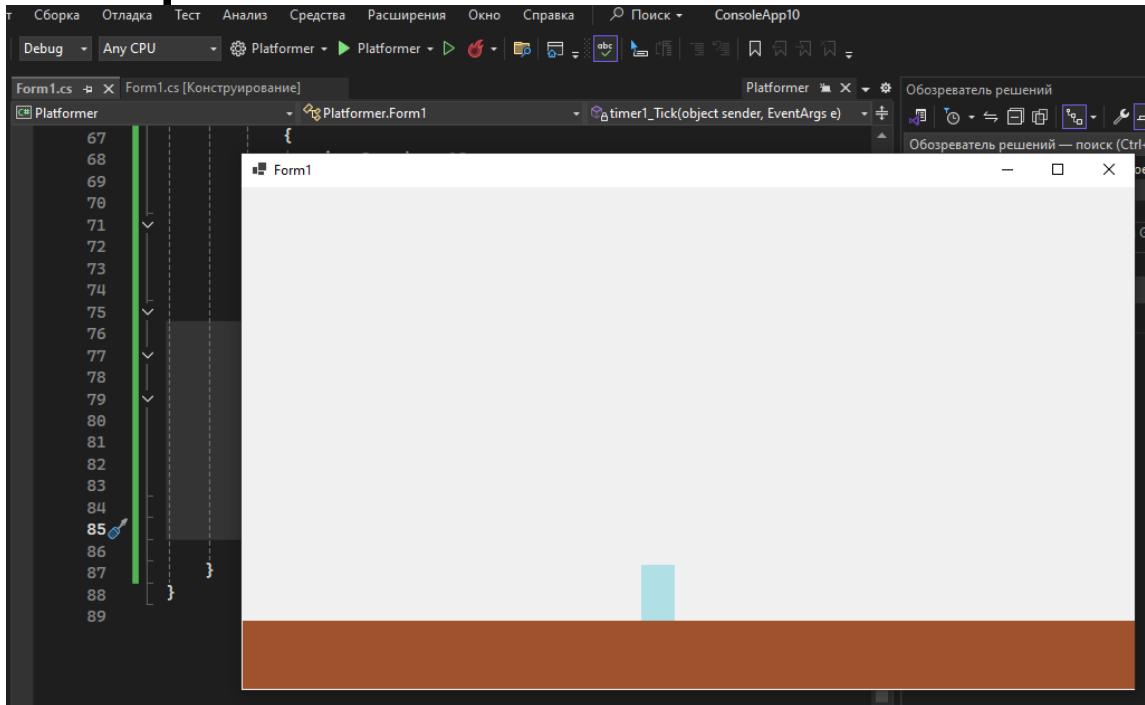
Передвижение персонажа

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

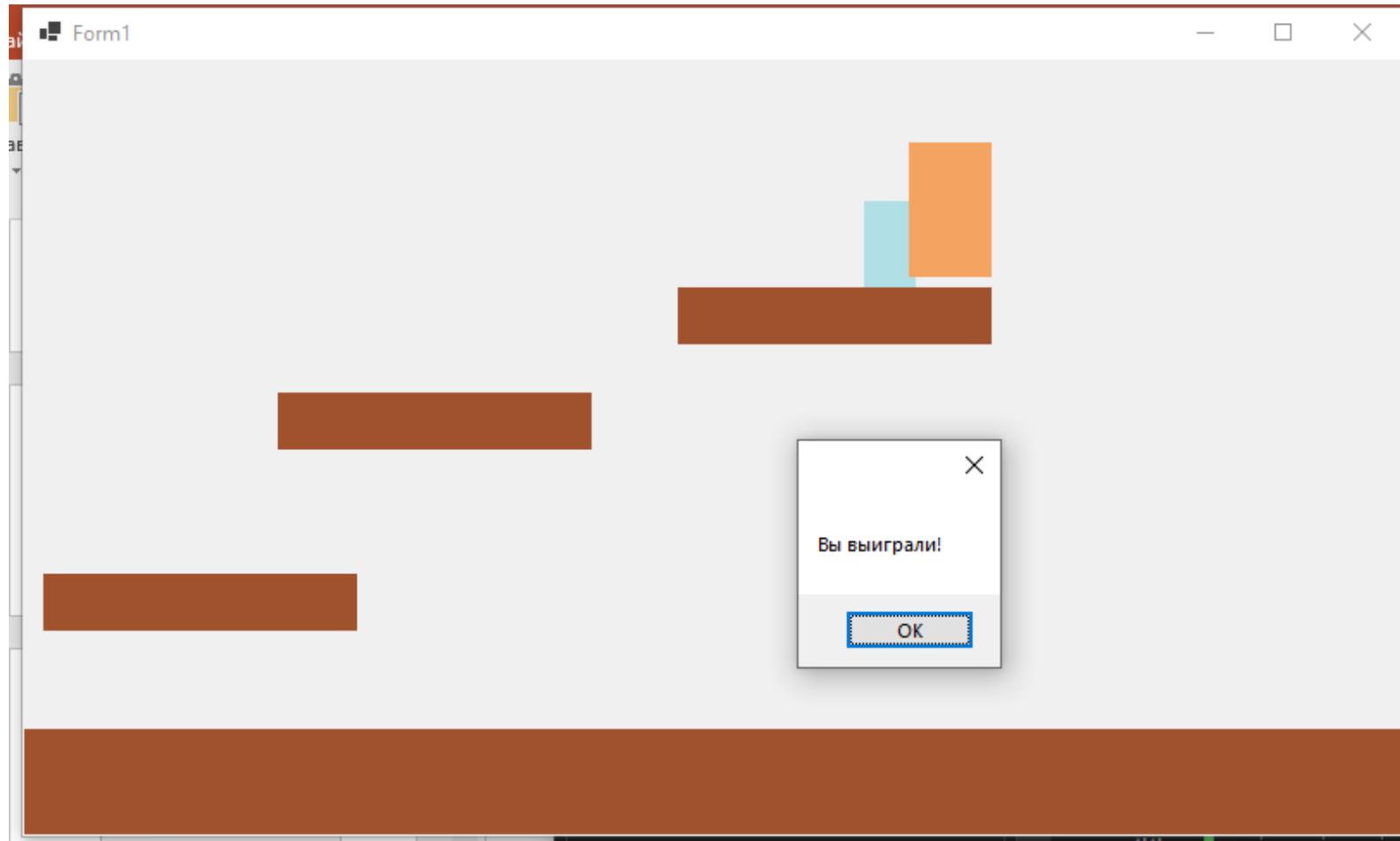
Добавьте еще несколько платформ (можно копировать).



Завершение уровня

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

Завершение уровня

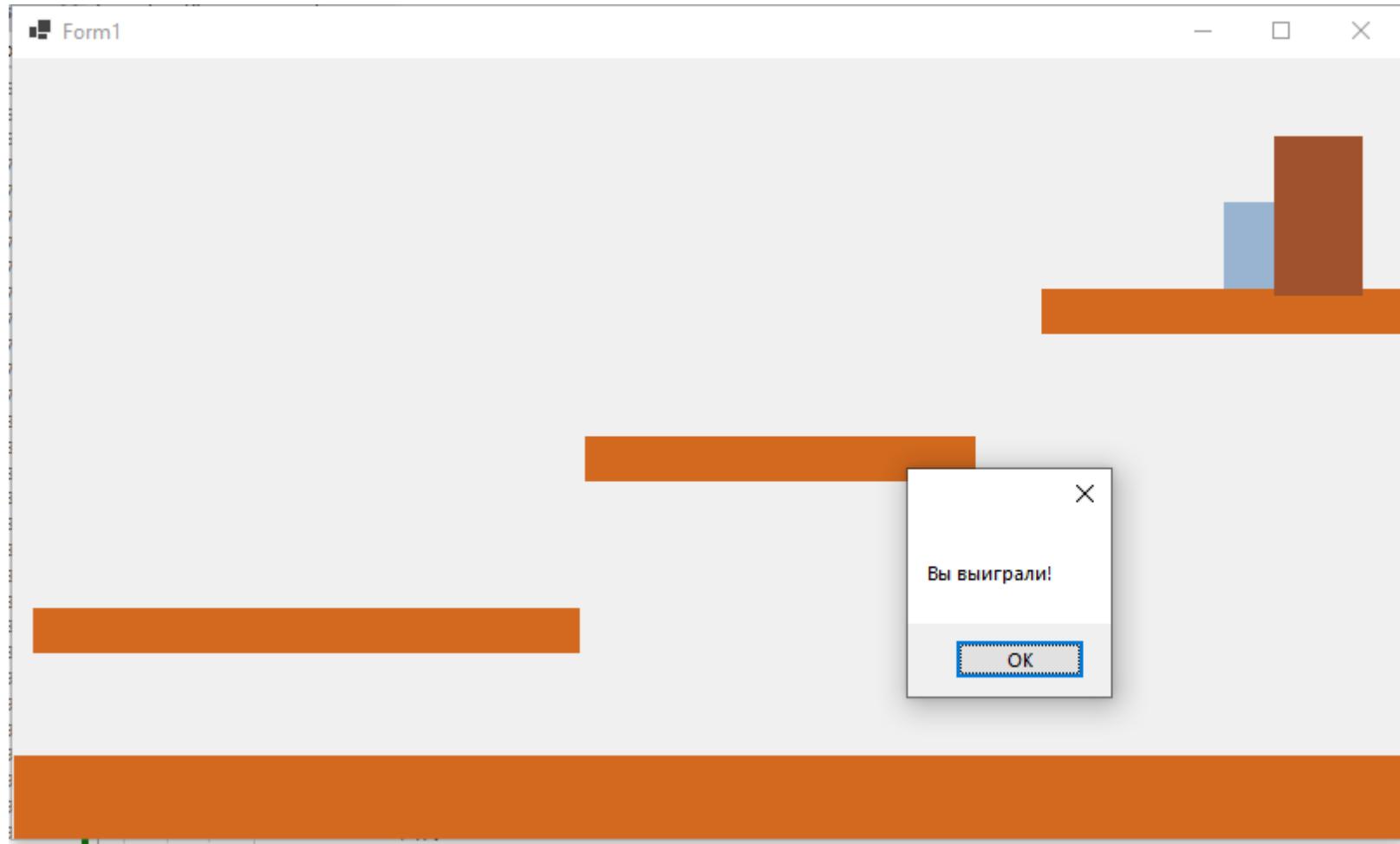


Завершение уровня

В timer1_Tick добавить:

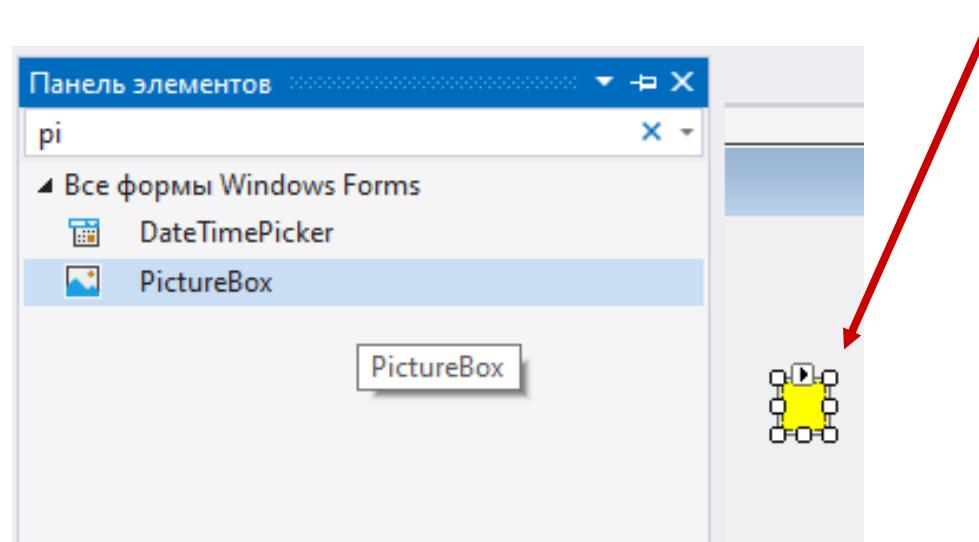
```
if(player.Bounds.IntersectsWith(door.Bounds))  
{  
    timer1.Stop();  
    MessageBox.Show("Вы выиграли!");  
}
```

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.IntersectsWith(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

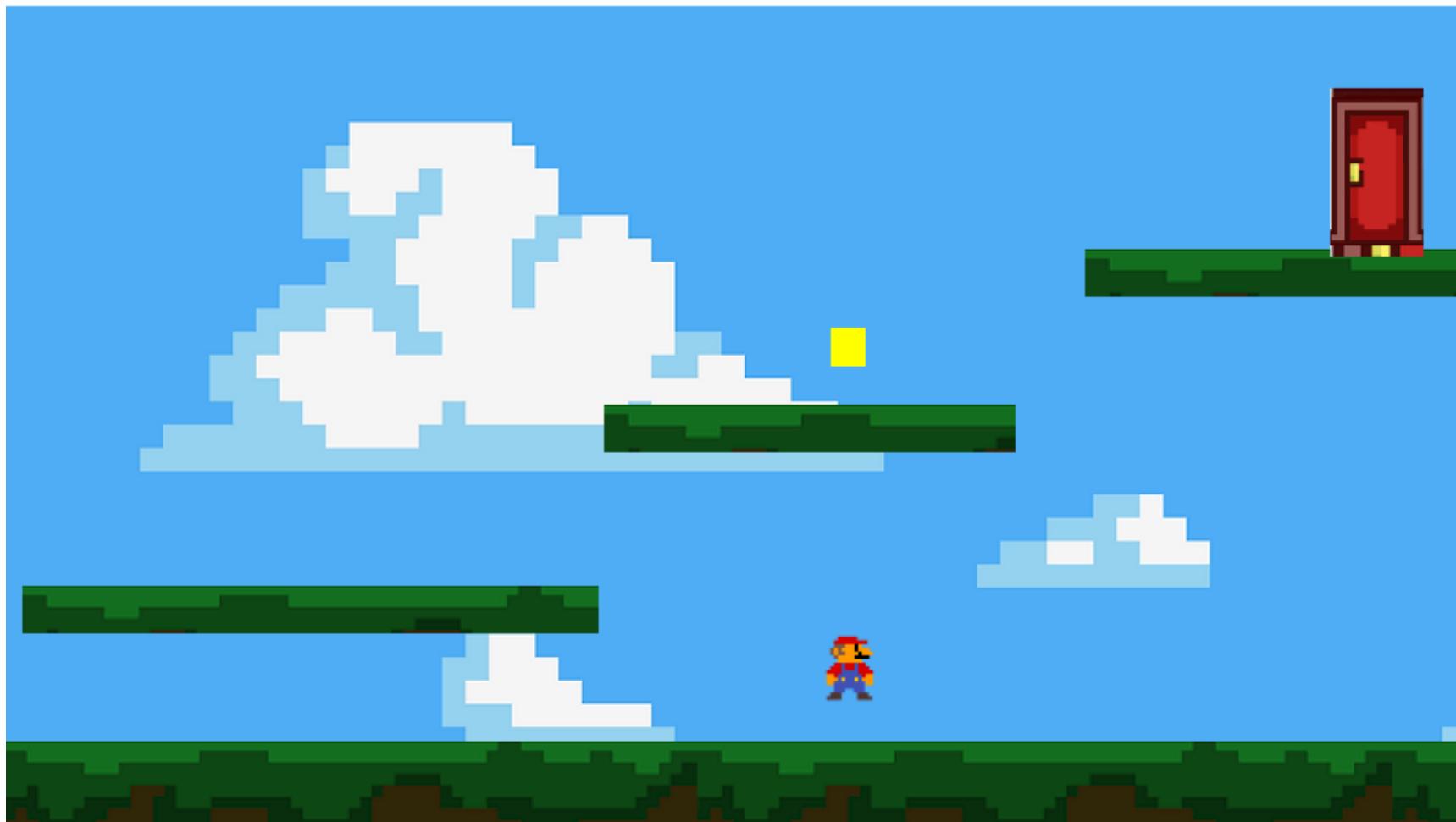
```
private void timer1_Tick(object sender, EventArgs e)
{
    player.Top += jumpSpeed;
    if (jumping && force < 0) ...
    if (goleft) ...
    if (goright) ...
    if (jumping) ...
    else ...
    foreach (Control x in this.Controls)
    {
        if (x is PictureBox && x.Tag == "platform")
    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

Тест игры

Запустите игру, проверьте правильность работы.

Form1

- □ ×



Что можно улучшить?

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