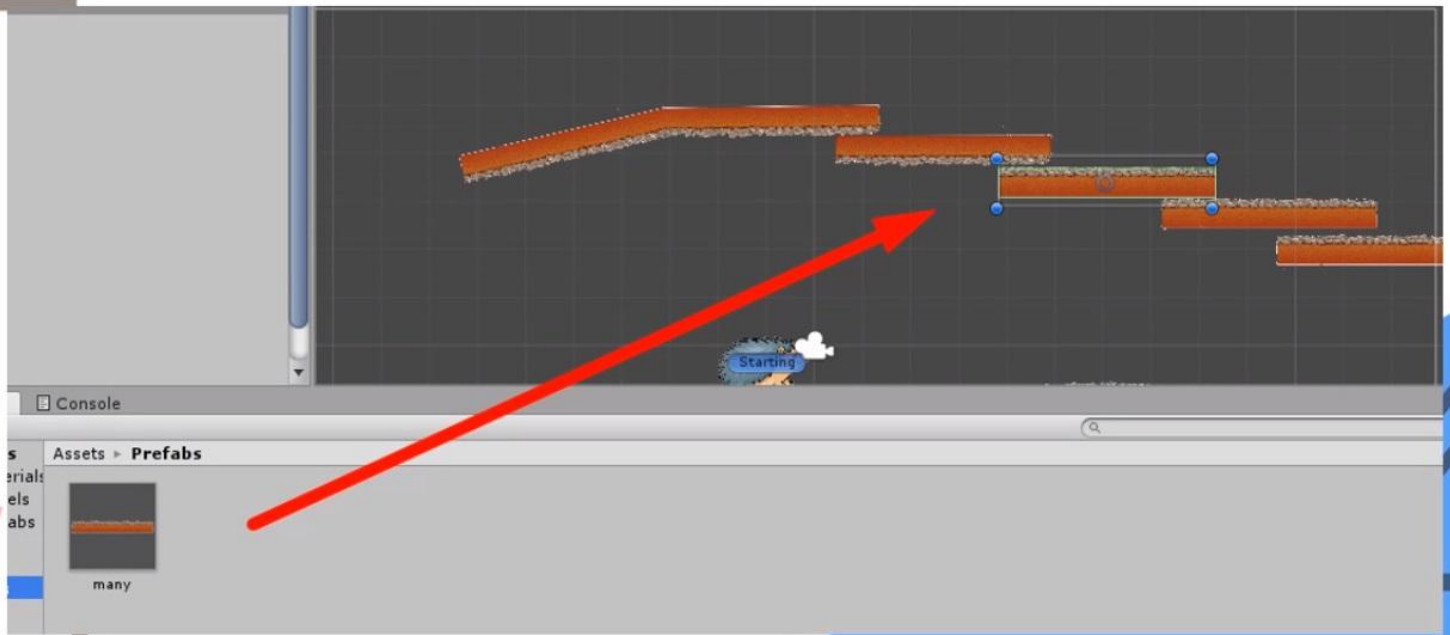


Unity 3D

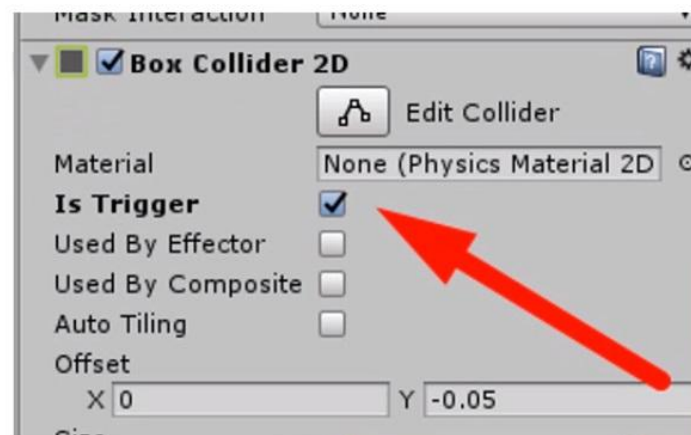
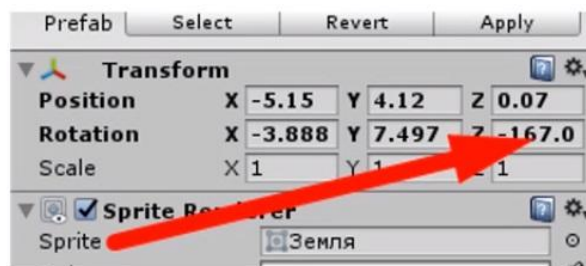
2D игра
Управление с клавиатуры
Урок 3



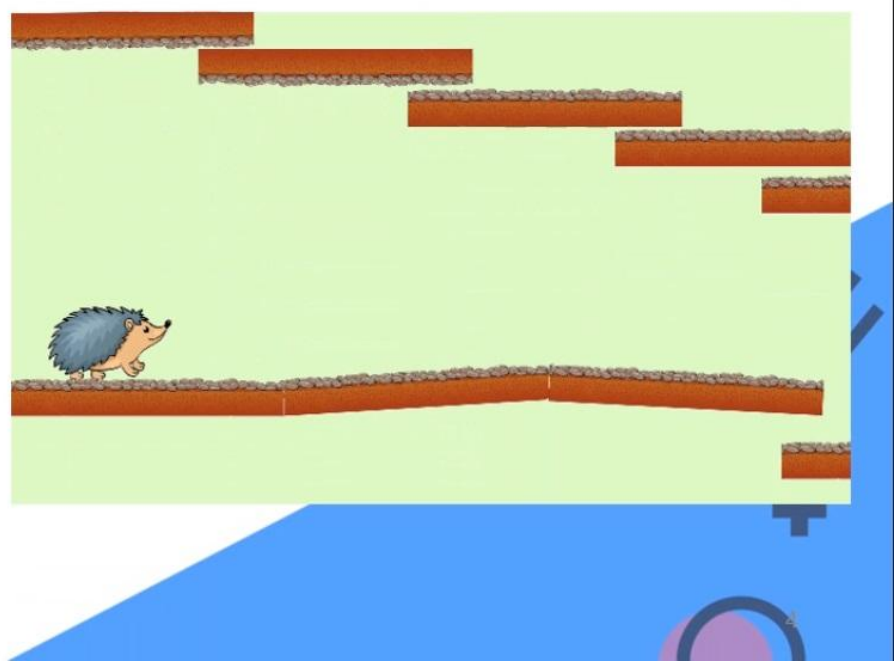
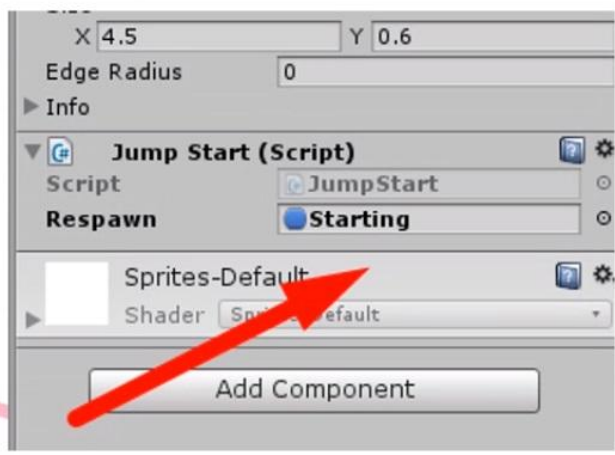
Домашнее задание



Домашнее задание

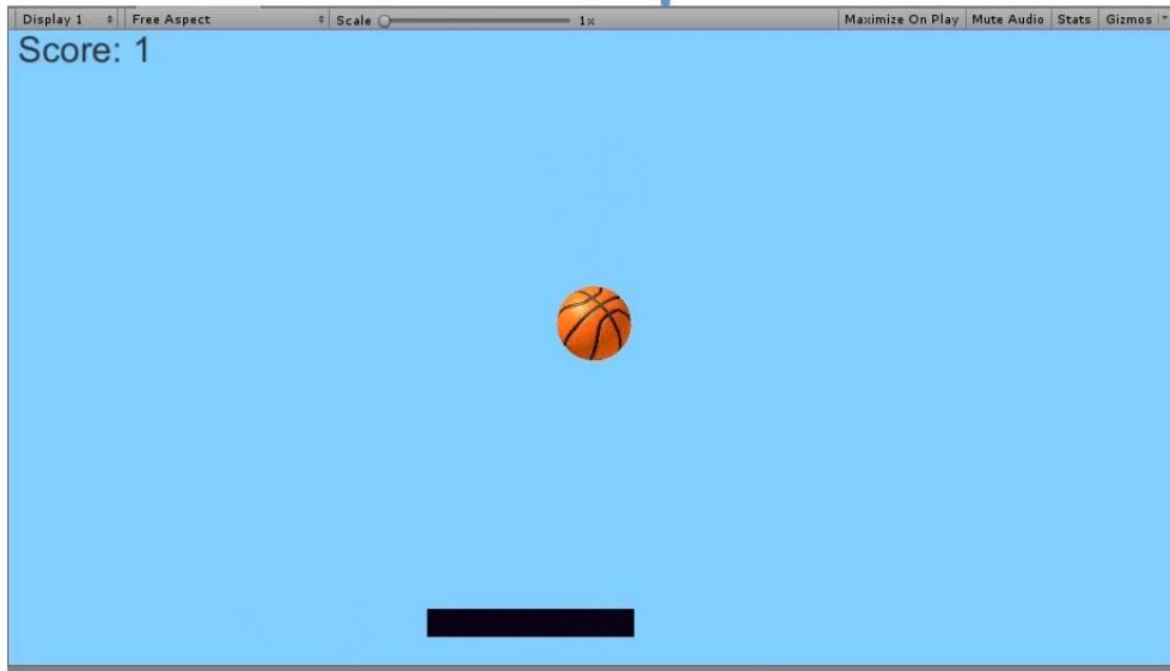


Домашнее задание



2D игра

5



Создаем проект

6

Projects

Learn

New

Open

On Disk

In the Cloud

Hedgehog

Path: C:\Users\Полиночка\Documents\Unity | Unity version: 2017.4.22

Platform

Path: C:\Users\Полиночка\Documents\Unity | Unity version: 2017.4.22 | unity_9dkud5r_s2flcq

for me

Path: C:\Users\Полиночка\Documents\Unity | Unity version: 2017.4.22 | unity_9dkud5r_s2flcq

Game with Owl_correct

Создаем проект

Project name: Platform

Location: C:\Users\ \Documents\Unit ...

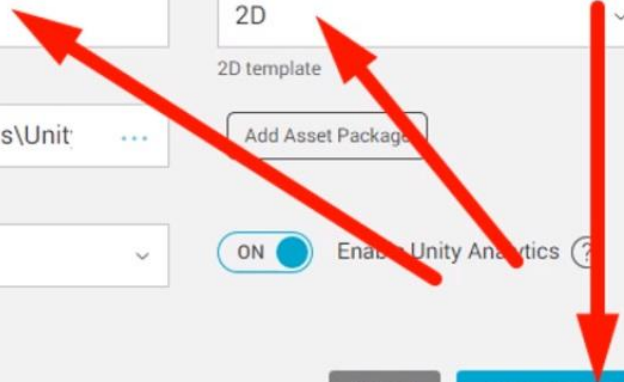
Organization: unity_9DKUd5r_s2fLcQ

Template: 2D

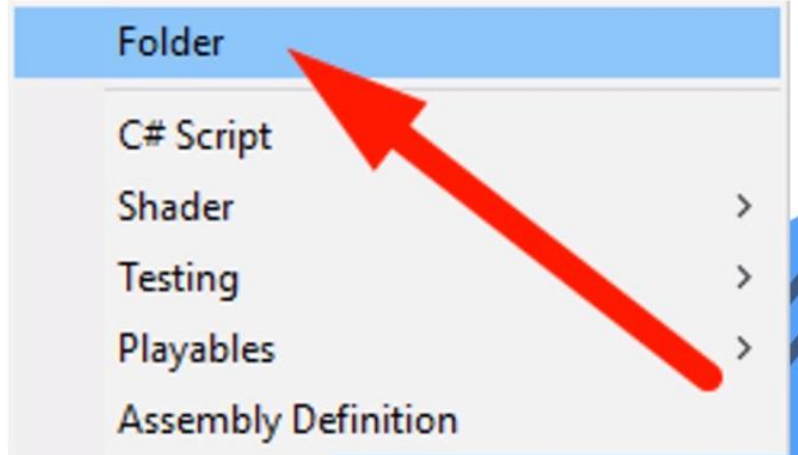
2D template: Add Asset Package

ON Enable Unity Analytics ?

Cancel Create project



Создание папок

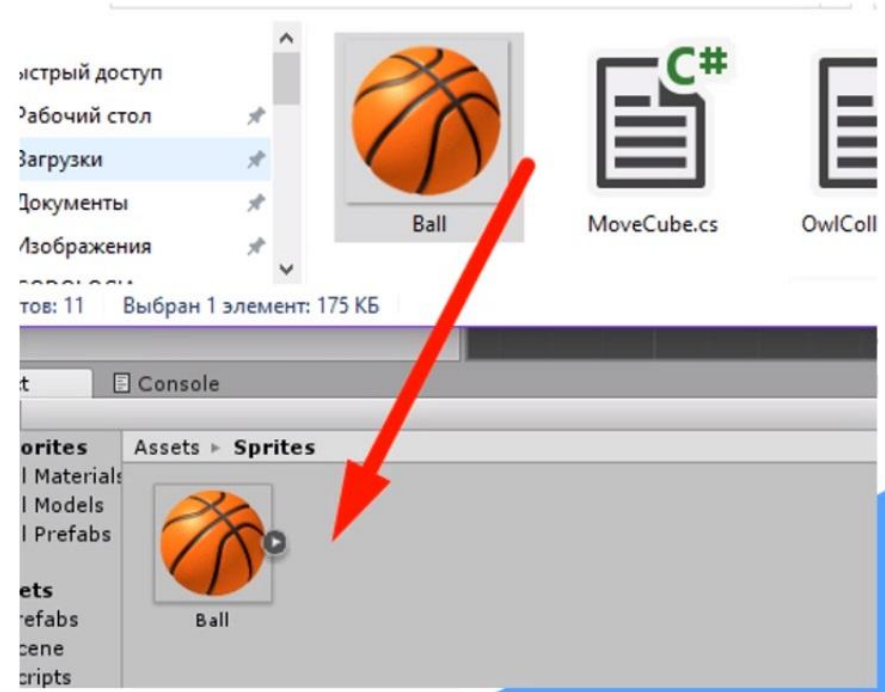


Задание

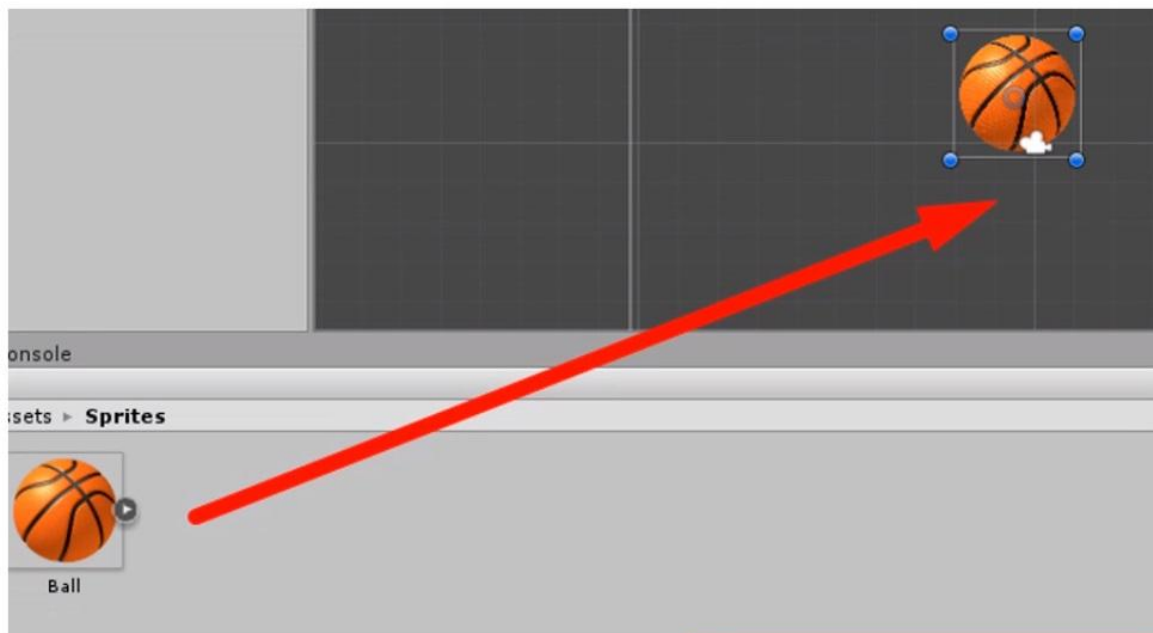


Для систематизации
создайте еще
несколько хранилищ.

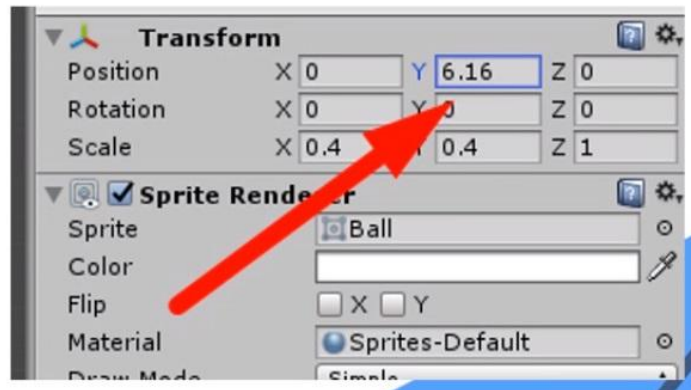
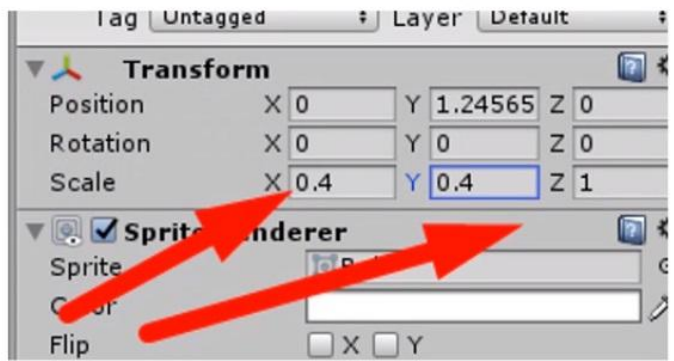
Добавление спрайта



Настройка спрайтов

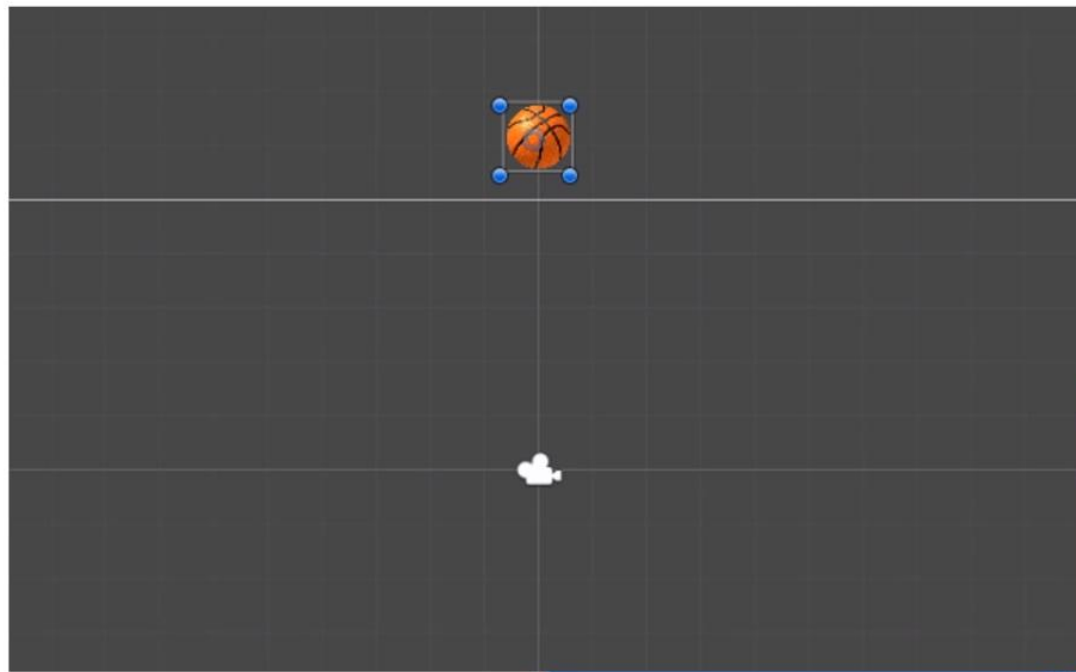


Настройка размера

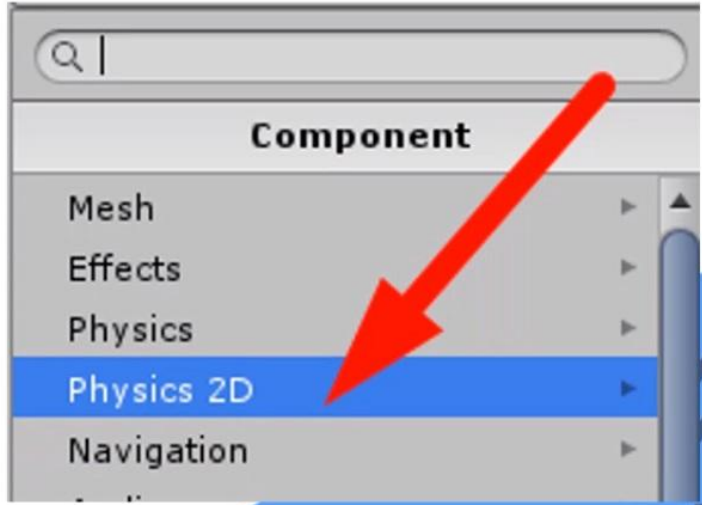
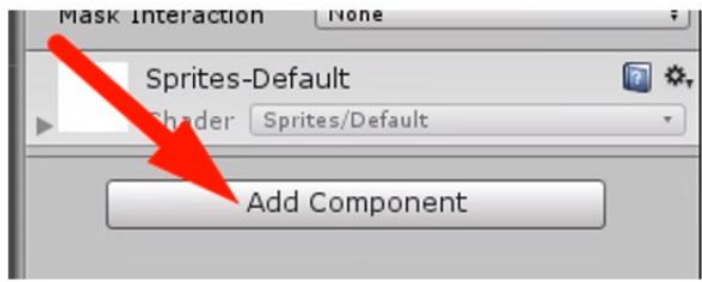


Настройка размера

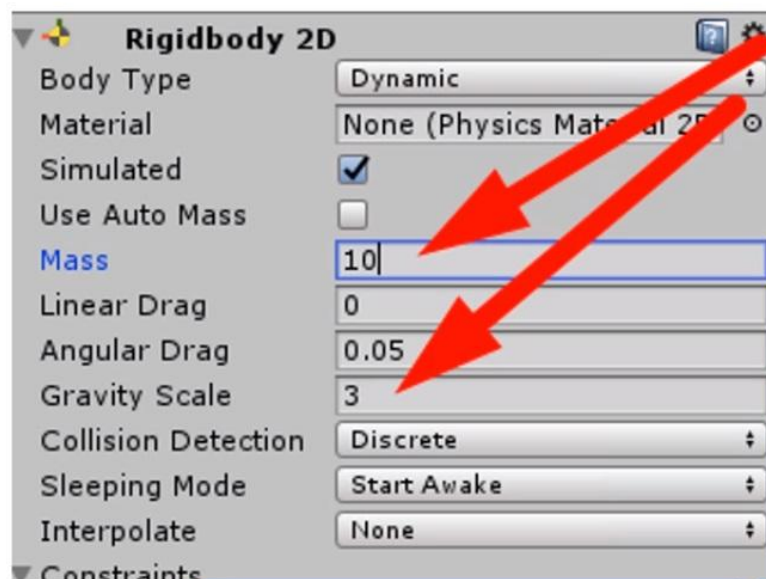
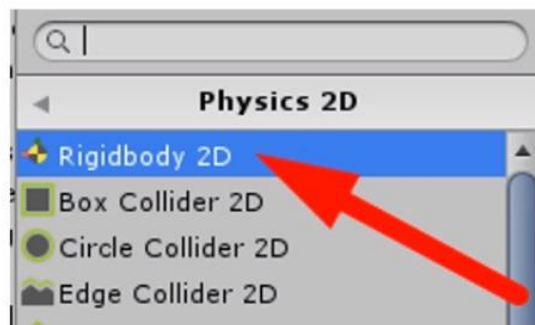
13



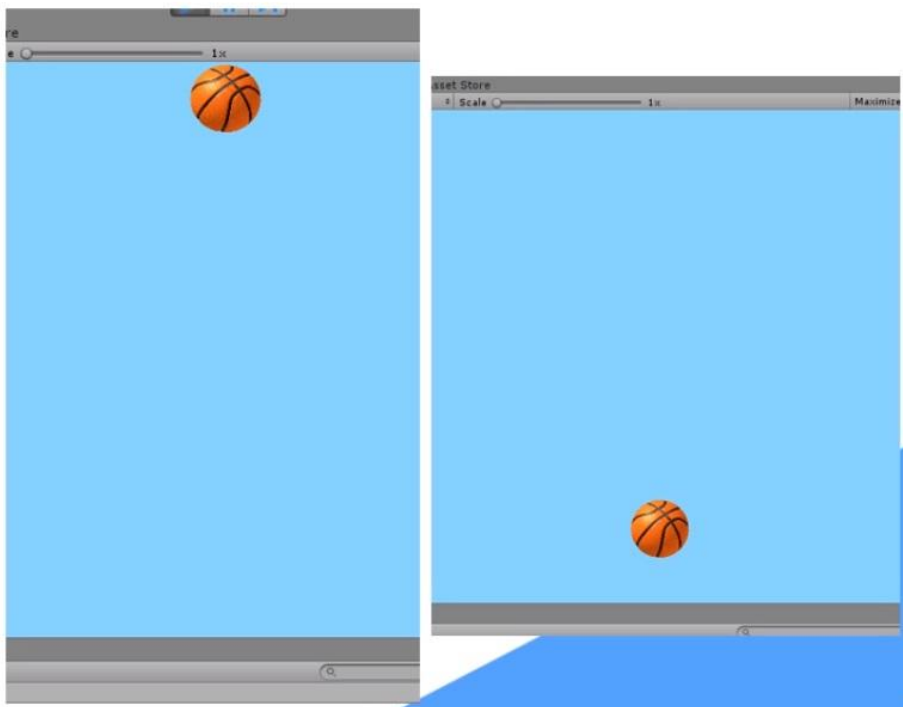
Добавление компонентов



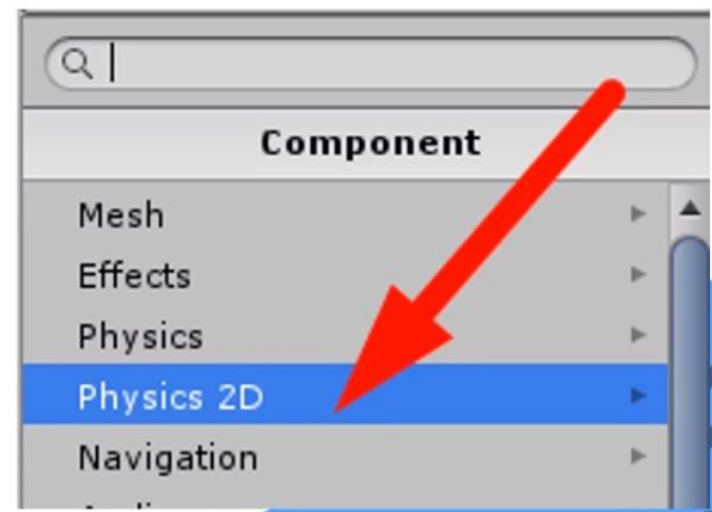
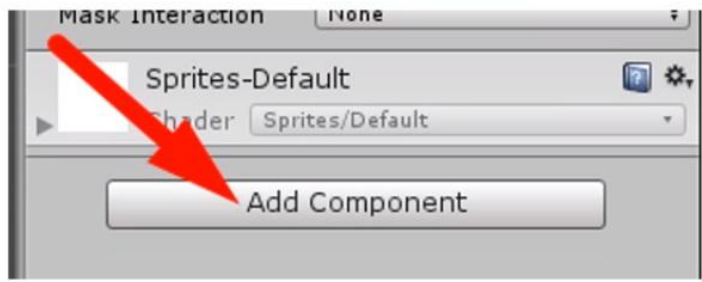
Добавление компонентов



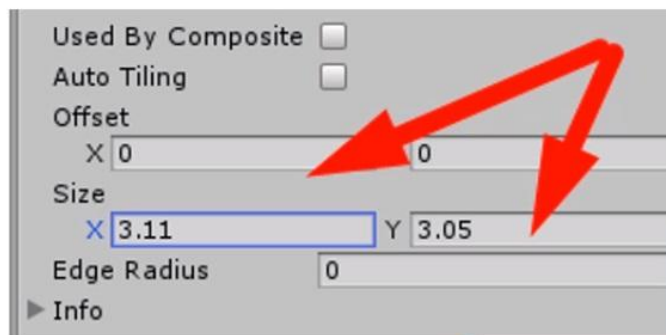
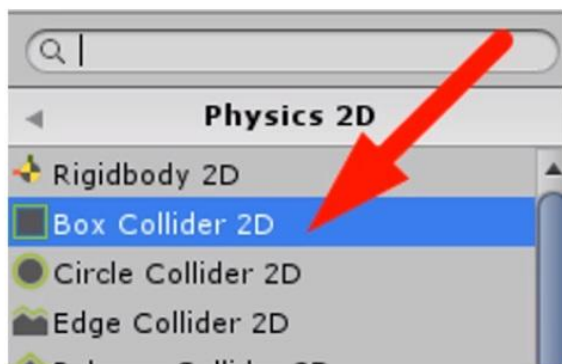
Запуск игры



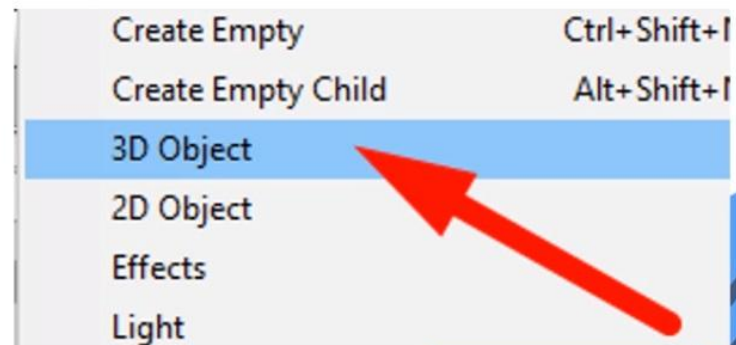
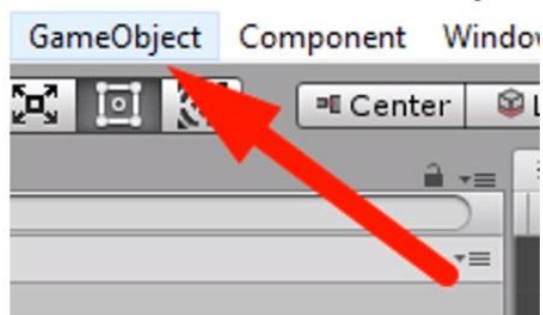
Добавление компонентов



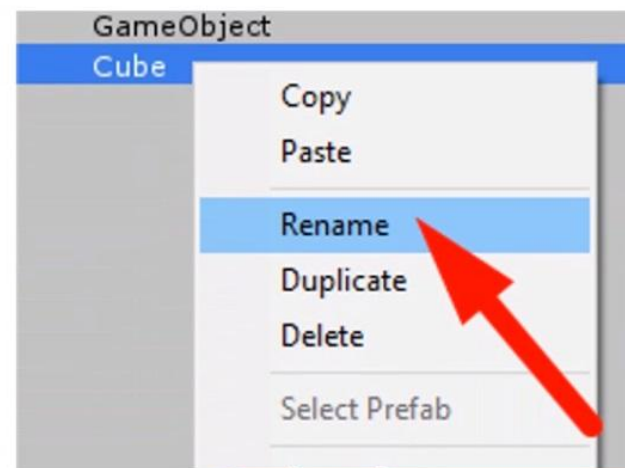
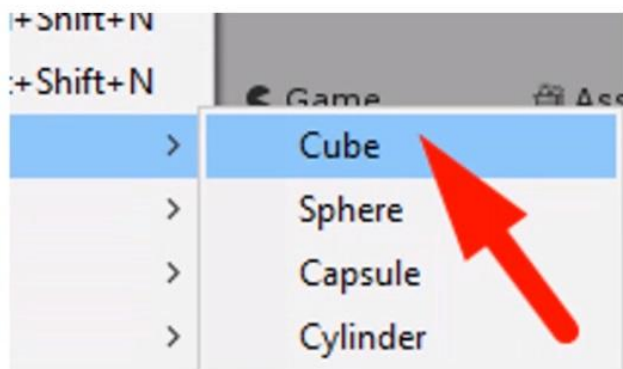
Добавление компонентов



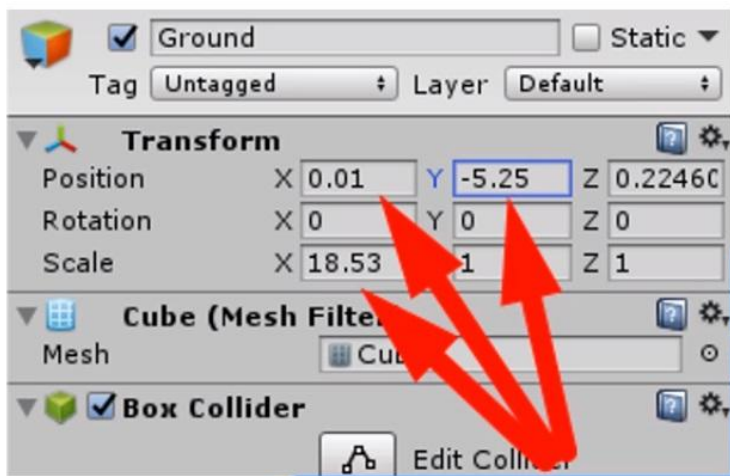
Добавление спрайта



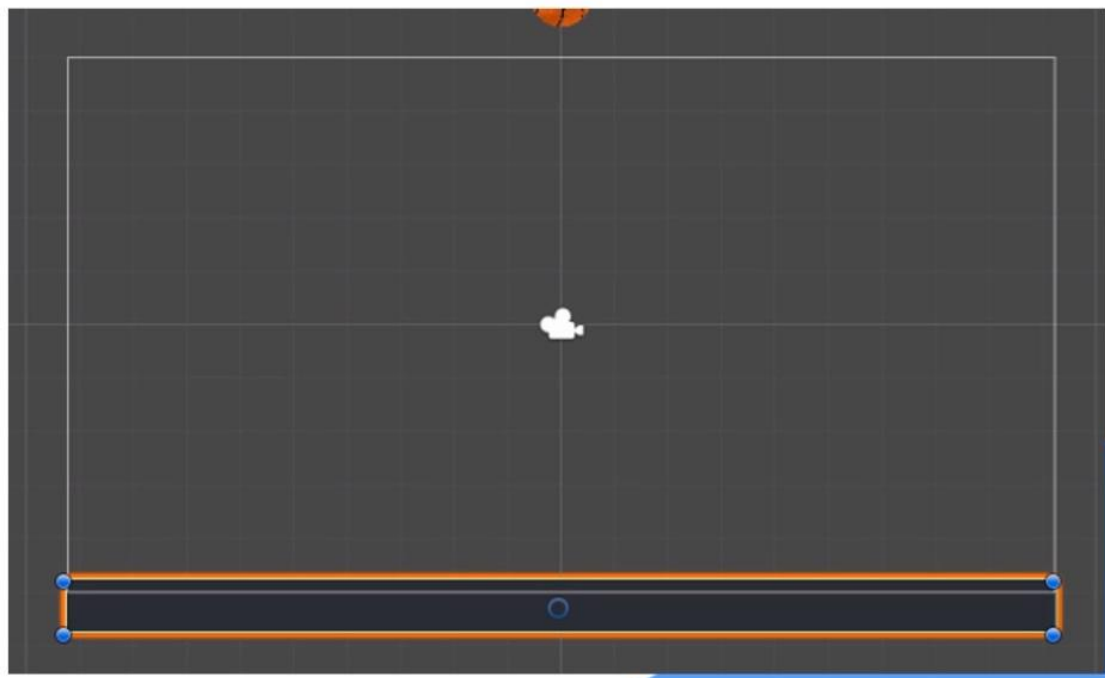
Добавление спрайта



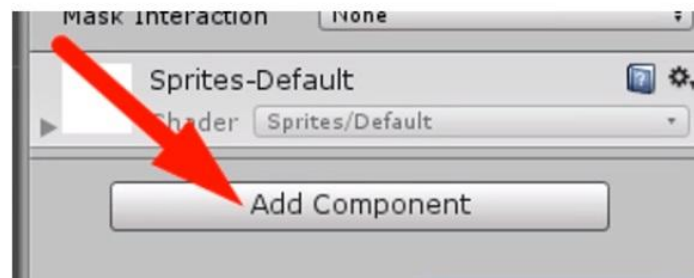
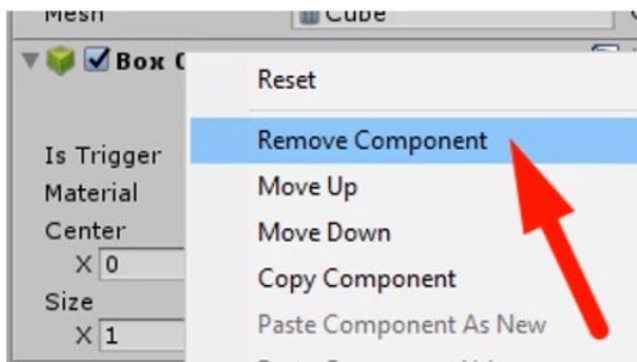
Добавление компонентов



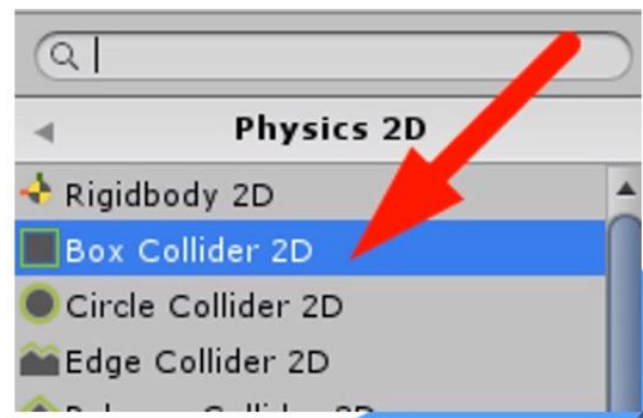
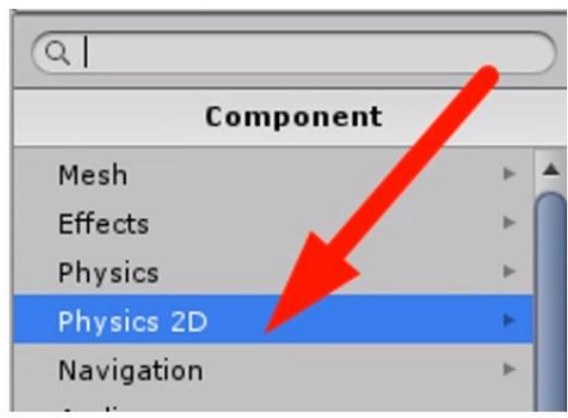
Добавление компонентов



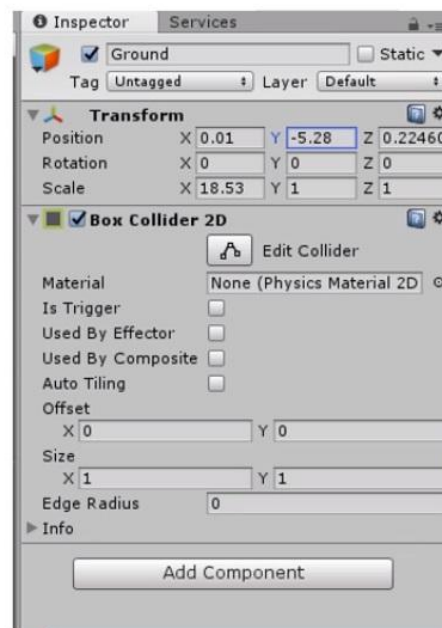
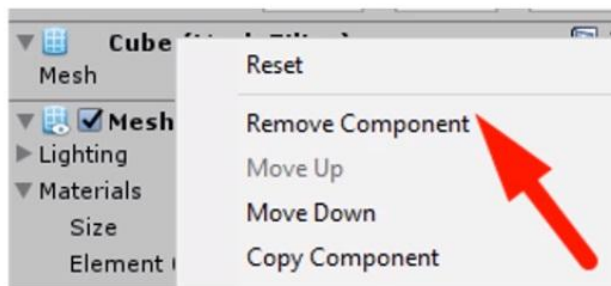
Добавление компонентов



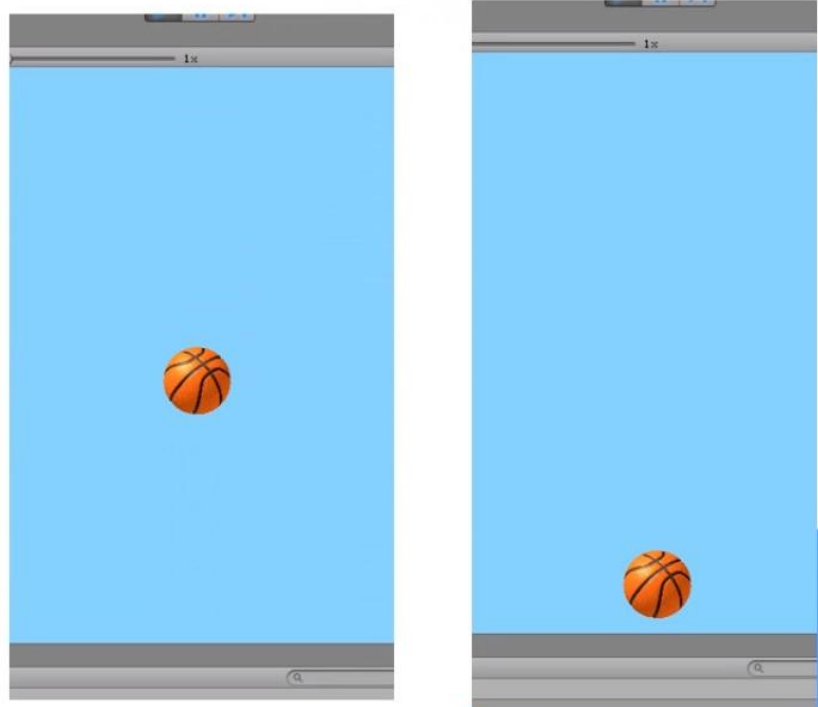
Добавление компонентов



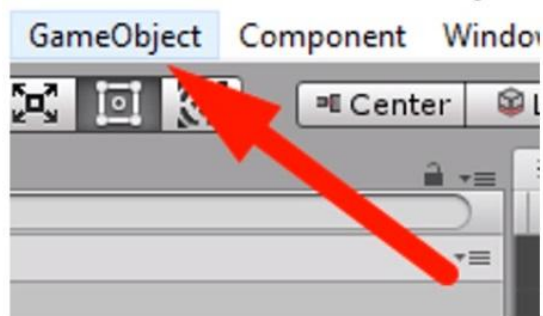
Удаление компонентов



Запуск игры

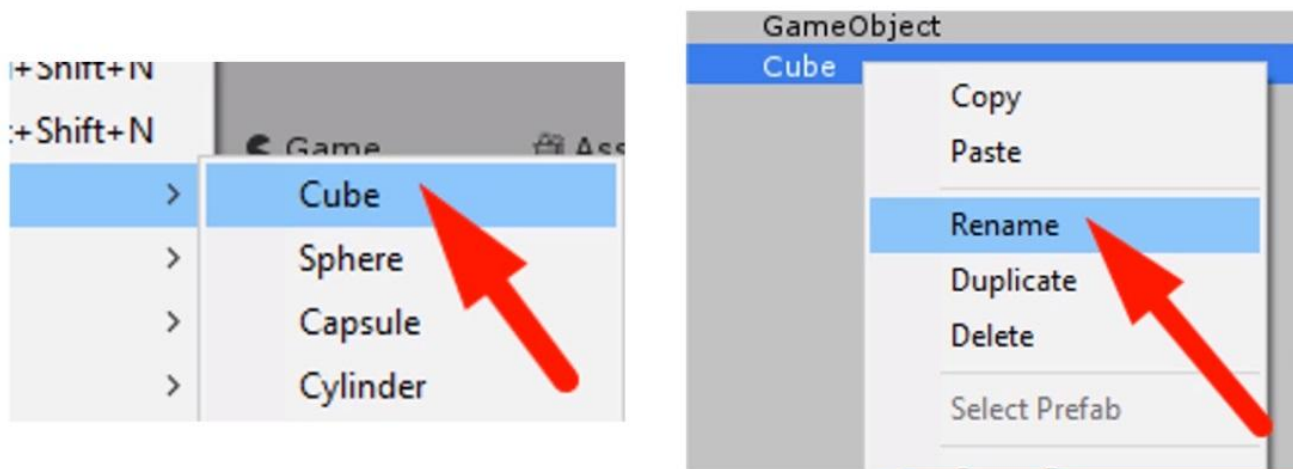


Добавление спрайта

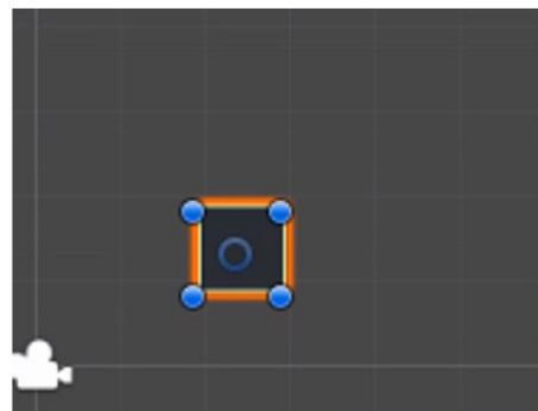


- Create Empty Ctrl+Shift+I
- Create Empty Child Alt+Shift+I
- 3D Object**
- 2D Object
- Effects
- Light

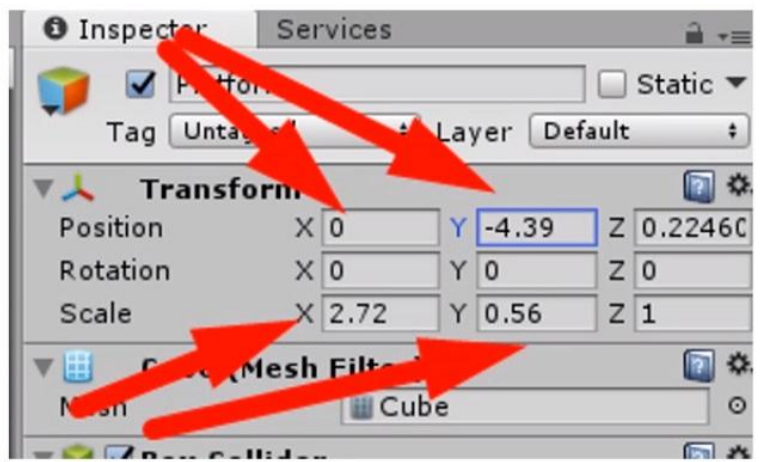
Называем



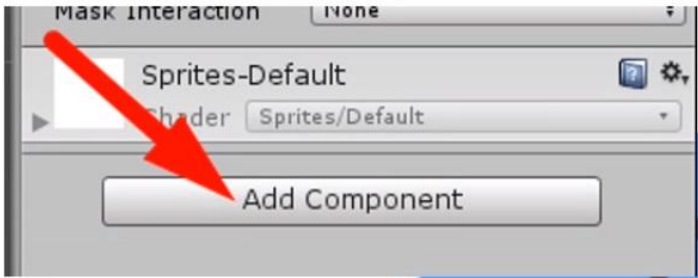
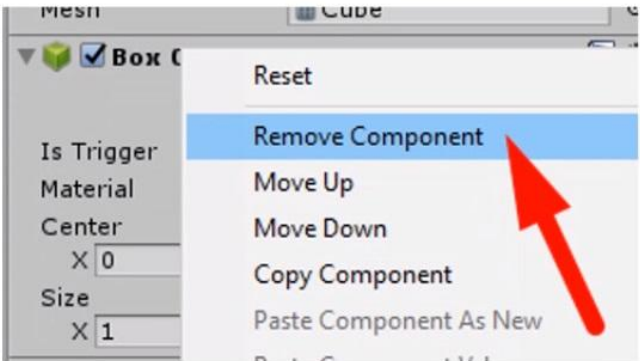
Называем



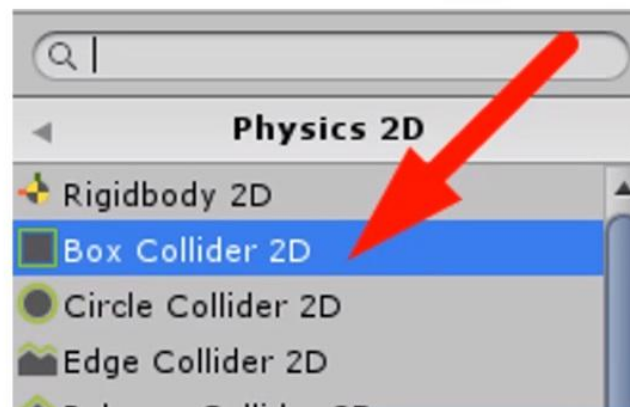
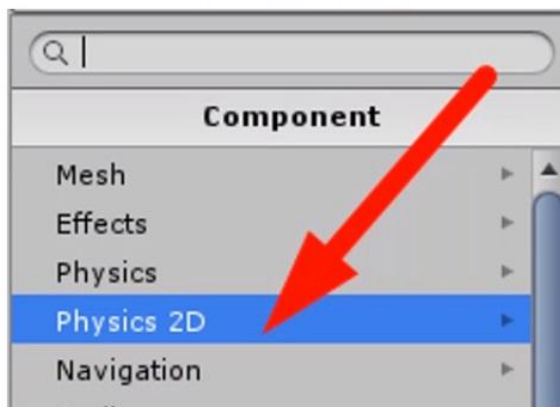
Настройка размера



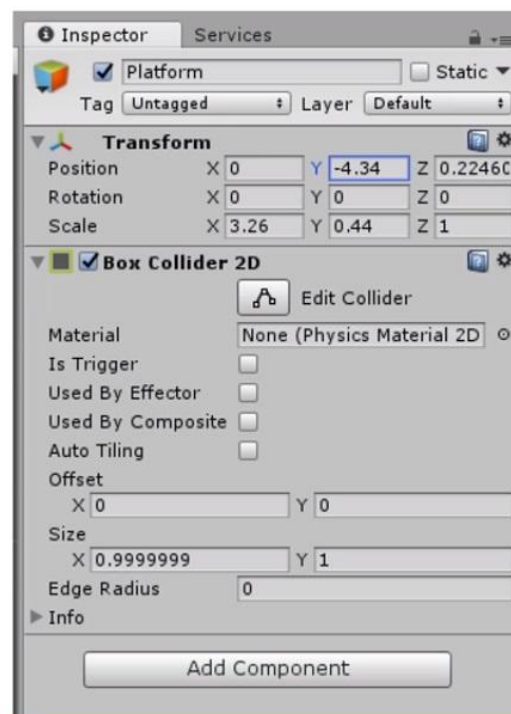
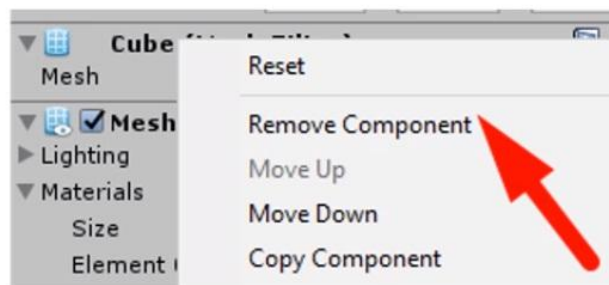
Добавление компонентов



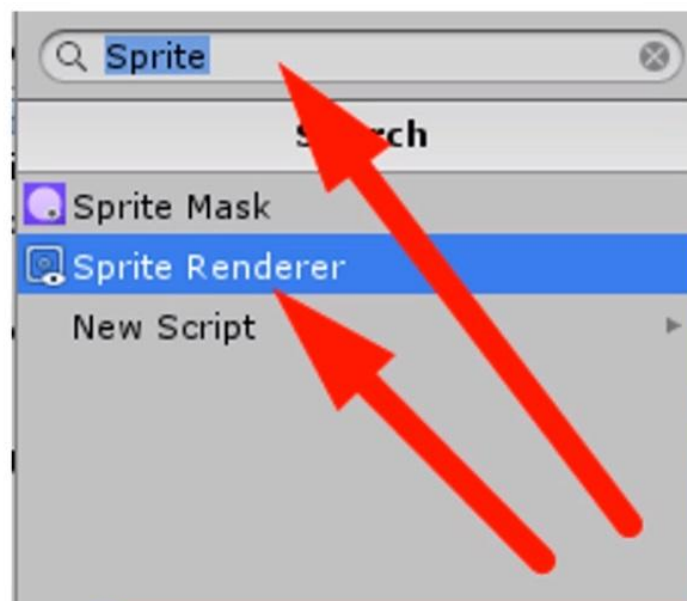
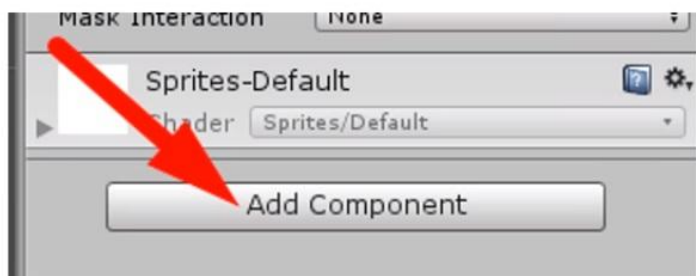
Добавление компонентов



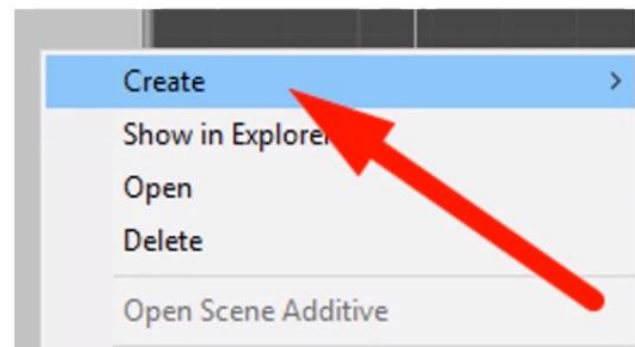
Удаление компонента



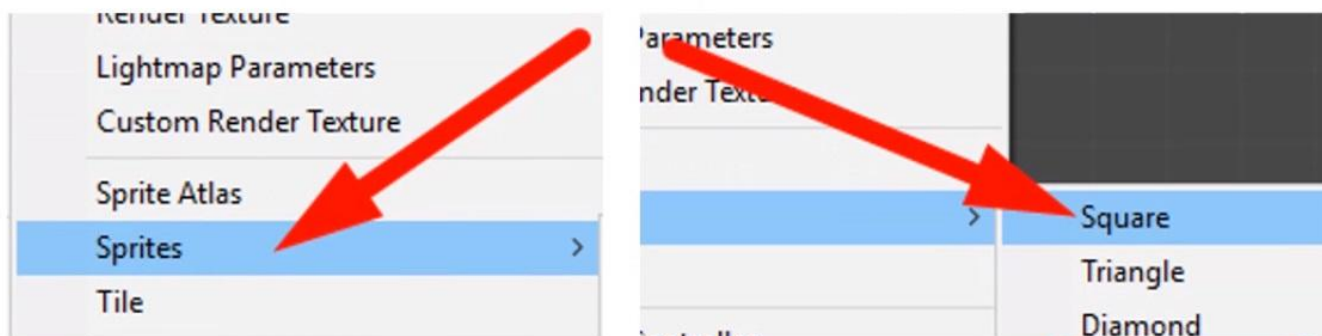
Добавление компонента



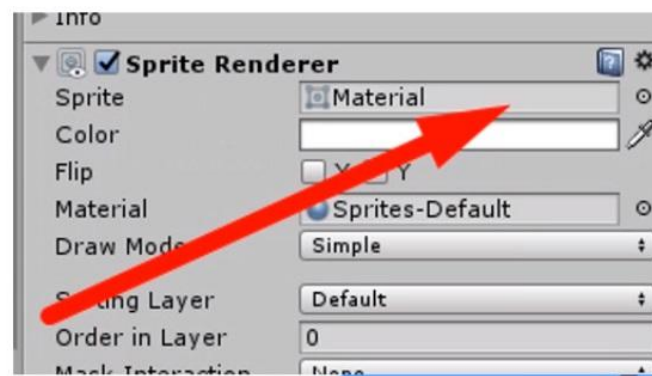
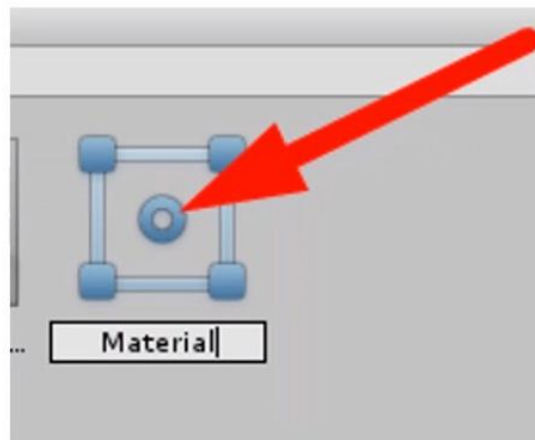
Создание материала



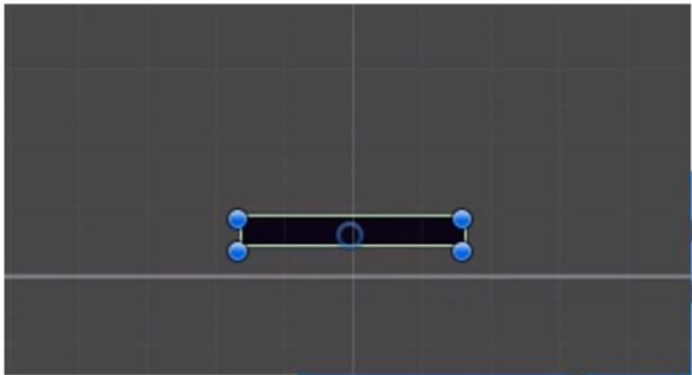
Настраиваем материал



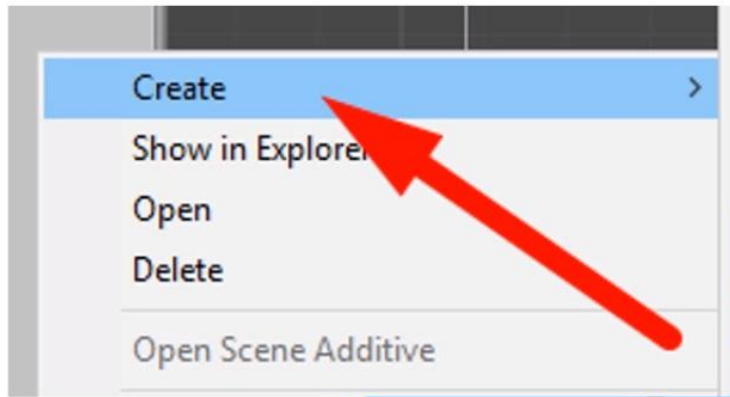
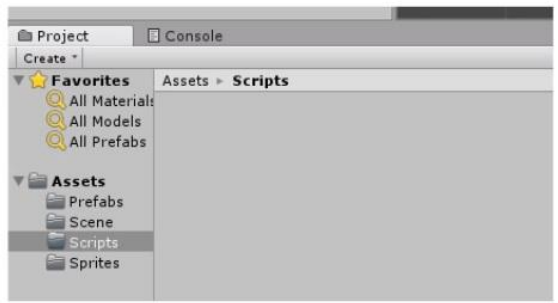
Настраиваем материал



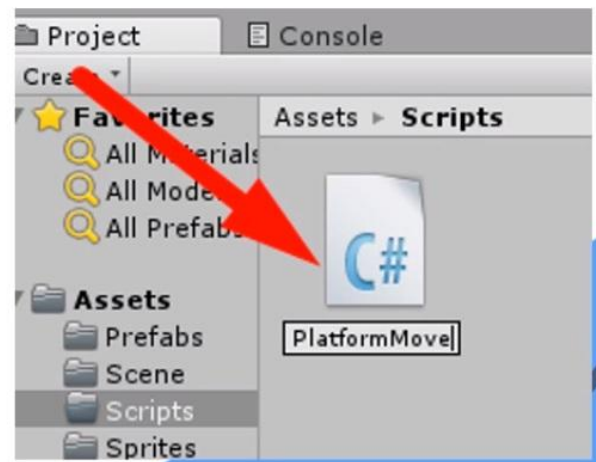
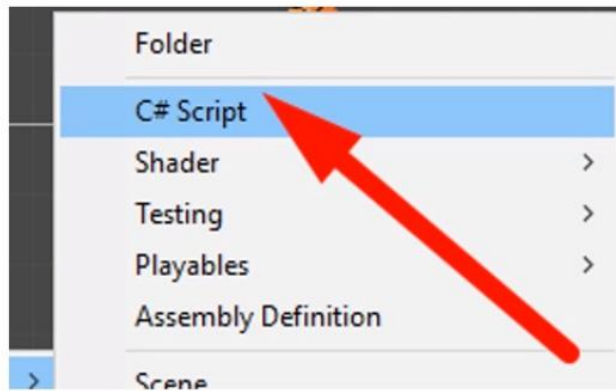
Выбираем цвет



Создаем скрипт



Создаем скрипт



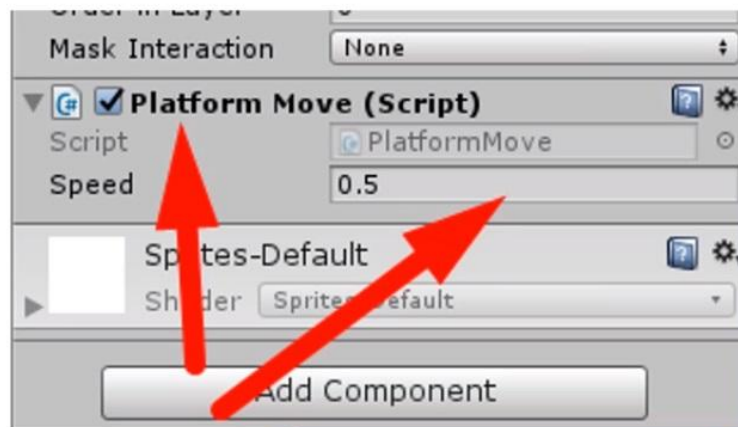
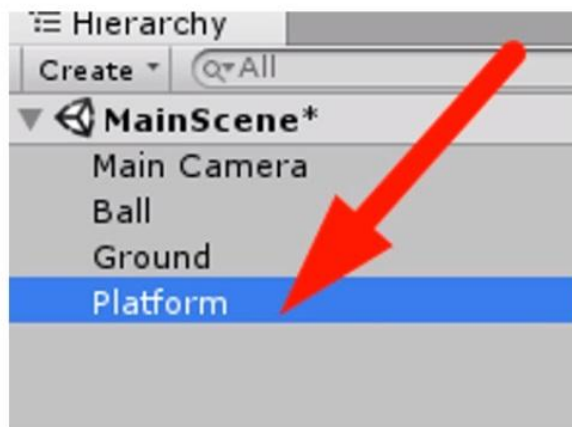
Пишем код

```
public class PlatformMove : MonoBehaviour {  
    [SerializeField] private float speed;  
  
    void Start ()  
    {  
    }  
  
    void FixedUpdate ()  
    {  
    }  
}
```

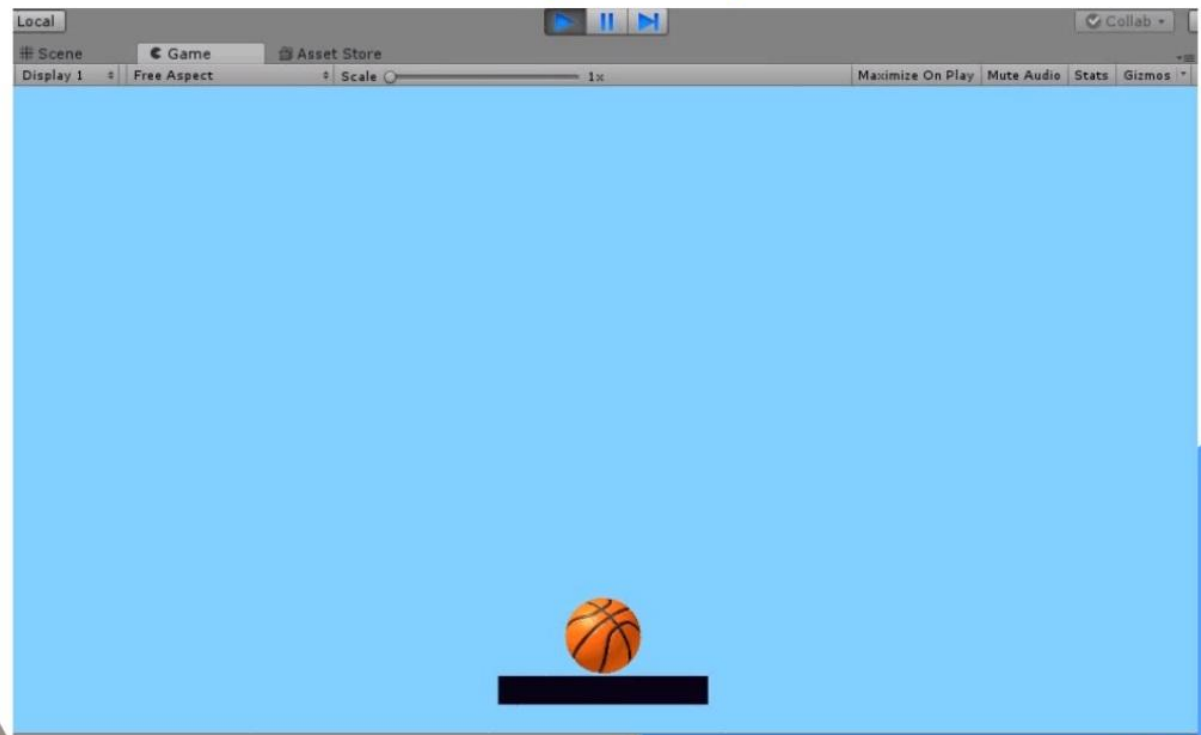
Написание скрипта

```
void FixedUpdate ()  
{  
    if (Input.GetKey(KeyCode.RightArrow))  
    {  
        transform.Translate(new Vector3(speed, 0, 0));  
    }  
    else if (Input.GetKey(KeyCode.LeftArrow))  
    {  
        transform.Translate(new Vector3(-speed, 0, 0));  
    }  
}
```

Прикрепляем скрипт



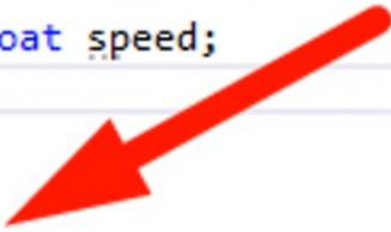
Запуск игры



Пишем код

```
public class PlatformMove : MonoBehaviour {  
    [SerializeField] private float speed;  
    private float leftboard;  
    private float rightboard;  
    private float xPlatform;  
    void Start ()  
    {  
    }
```

private float leftboard;
private float rightboard;
private float xPlatform;



Пишем код

```
void Start ()  
{  
    var renderer = GetComponent<SpriteRenderer>();  
  
    leftboard = Camera.main.ViewportToWorldPoint(new Vector2(0, 0)).x;  
    rightboard = Camera.main.ViewportToWorldPoint(new Vector2(1, 0)).x;  
}
```

Пишем код

```
void Start ()  
{  
    var renderer = GetComponent<SpriteRenderer>();  
  
    leftboard = Camera.main.ViewportToWorldPoint(new Vector2(0, 0)).x;  
    rightboard = Camera.main.ViewportToWorldPoint(new Vector2(1, 0)).x;  
  
    xPlatform = renderer.bounds.size.x;  
}
```

Написание скрипта

```
void FixedUpdate ()
{
    if (transform.position.x + xPlatform > leftboard)
    {
        if (transform.position.x - xPlatform < rightboard)
        {
            if (Input.GetKey(KeyCode.RightArrow))
            {
                transform.Translate(new Vector3(speed, 0, 0));
            }
            else if (Input.GetKey(KeyCode.LeftArrow))
            {
                transform.Translate(new Vector3(-speed, 0, 0));
            }
        }
    }
}
```

Two red arrows are present. One points from the right side of the code block towards the innermost 'if' statement. The other points from the right side towards the outermost 'if' statement.

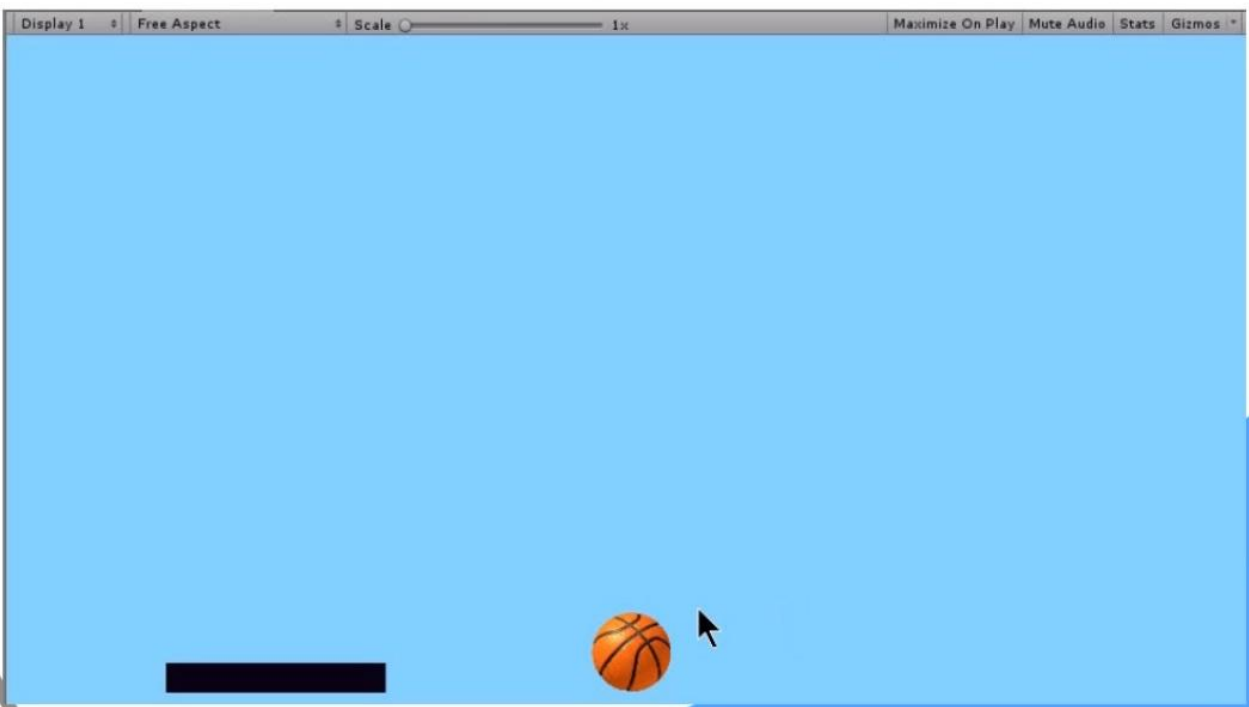
Написание скрипта

```
void FixedUpdate ()  
{  
    if (transform.position.x + xPlatform > leftboard)  
    {  
        if (transform.position.x - xPlatform < rightboard)  
        {  
            if (Input.GetKey(KeyCode.RightArrow))  
            {  
                transform.Translate(new Vector3(speed, 0, 0));  
            }  
            else if (Input.GetKey(KeyCode.LeftArrow))  
            {  
                transform.Translate(new Vector3(-speed, 0, 0));  
            }  
        }  
        else  
        transform.position = new Vector3(leftboard, -4.54f, 0);  
    }  
}
```

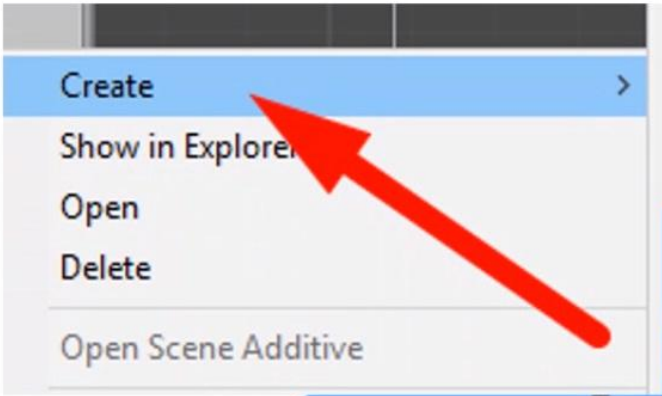
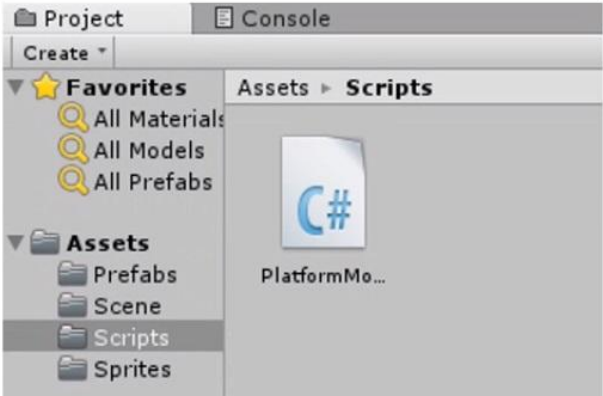
Пишем код

```
1 (transform.position.x < leftboard &amp; transform.position.x > rightboard)  
{  
    if (Input.GetKey(KeyCode.RightArrow))  
    {  
        transform.Translate(new Vector3(speed, 0, 0));  
    }  
    else if (Input.GetKey(KeyCode.LeftArrow))  
    {  
        transform.Translate(new Vector3(-speed, 0, 0));  
    }  
}  
else  
    transform.position = new Vector3(leftboard, -4.54f, 0);  
else  
    transform.position = new Vector3(rightboard, -4.54f, 0);
```

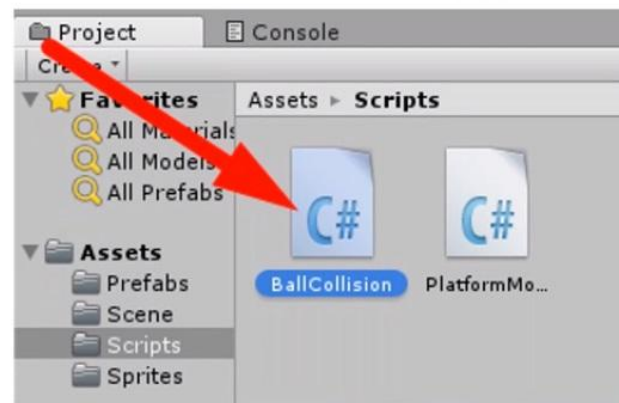
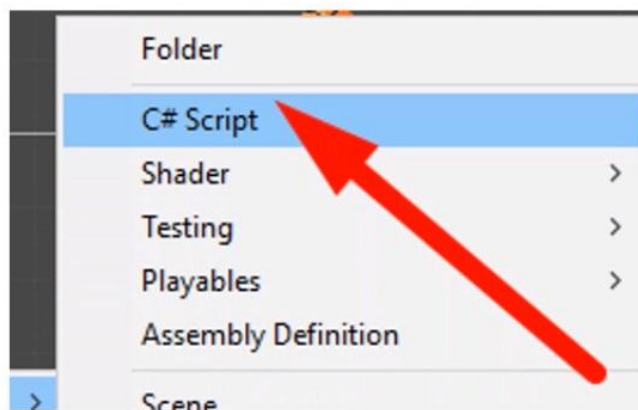
Запуск игры



Создаем скрипт



Создаем скрипт



Пишем код


```
public class BallCollision : MonoBehaviour {  
    [SerializeField] float impulse;  
    private Rigidbody2D rbody;  
  
    void Start ()  
    {  
        ...  
    }  
}
```

[SerializeField] float impulse;
private Rigidbody2D rbody;



Пишем код

```
public class BallCollision : MonoBehaviour {  
  
    [SerializeField] float impulse;  
    private Rigidbody2D rbody;  
  
    void Start ()  
    {  
        rbody = GetComponent<Rigidbody2D>();  
    }  
}
```



Пишем код

```
void Start ()  
{  
    rbody = GetComponent<Rigidbody2D>();  
}  
  
private void OnCollisionEnter2D(Collision2D collision)  
{  
    if (collision.gameObject.name == "Platform")  
    {  
    }  
}
```


Пишем код

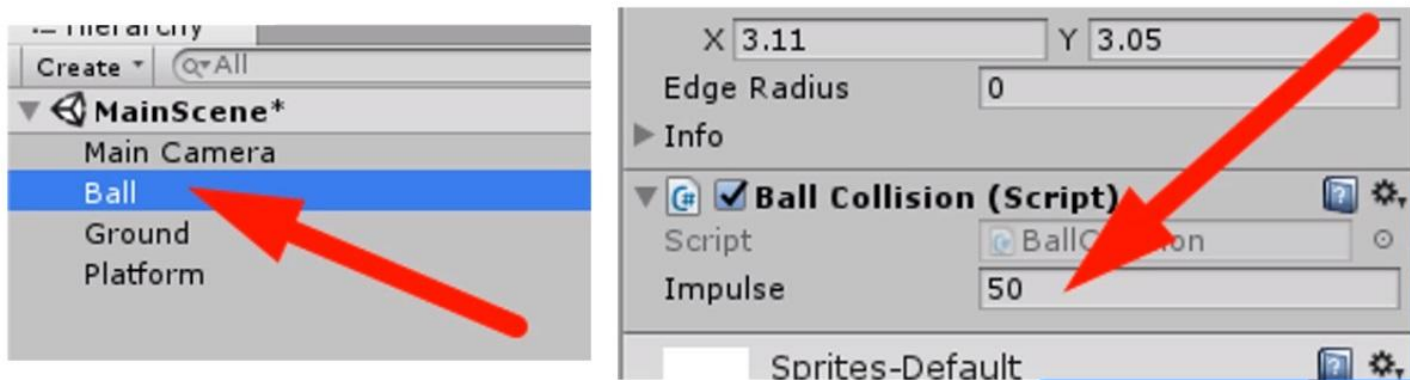
```
public class BallCollision : MonoBehaviour  
{  
    [SerializeField] float impulse;  
    private Rigidbody2D rbody;  
    private Vector2 direction;  
  
    void Start ()  
    {  
        rbody = GetComponent<Rigidbody2D>();  
    }  
}
```



Пишем код

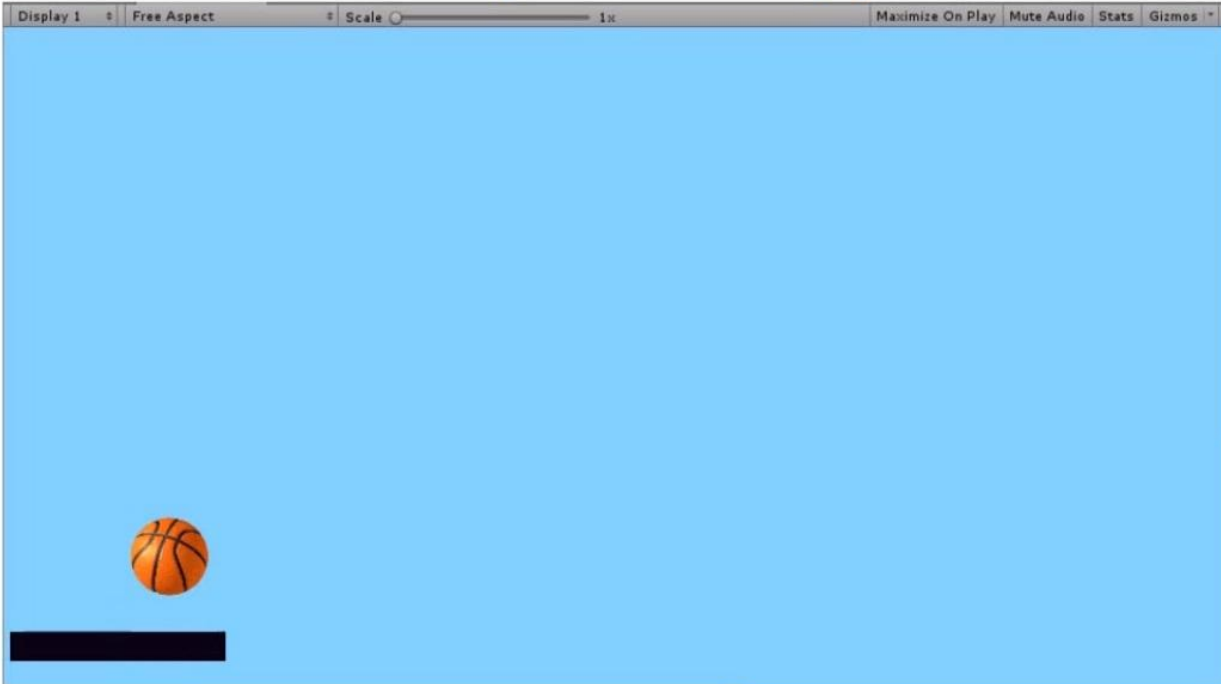
```
private void OnCollisionEnter2D(Collision2D collision)
{
    if (collision.gameObject.name == "Platform")
    {
        direction = new Vector2(Random.Range(-1, 1), Random.Range(3, 5));
        rbody.AddForce(direction * impulse, ForceMode2D.Impulse);
    }
}
```

Прикрепляем скрипт



Запускаем игру

60

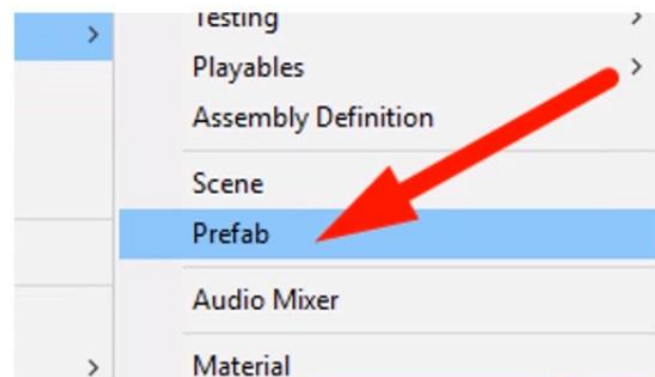
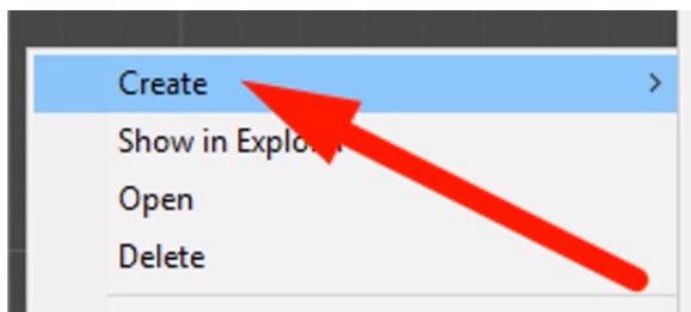


Задание

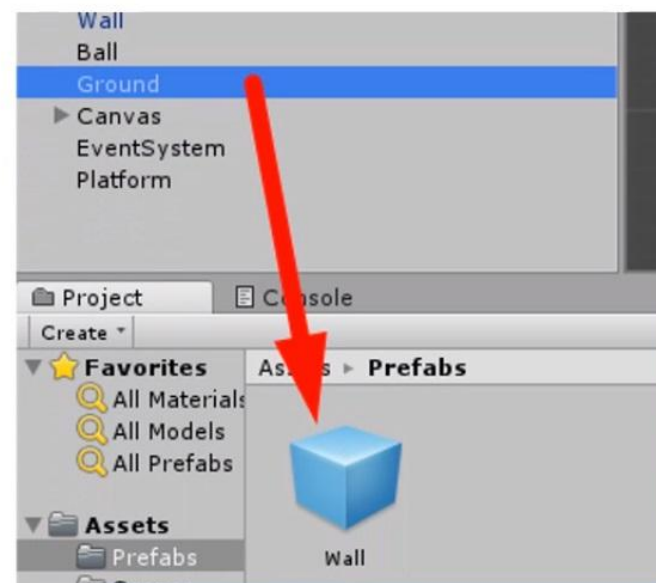
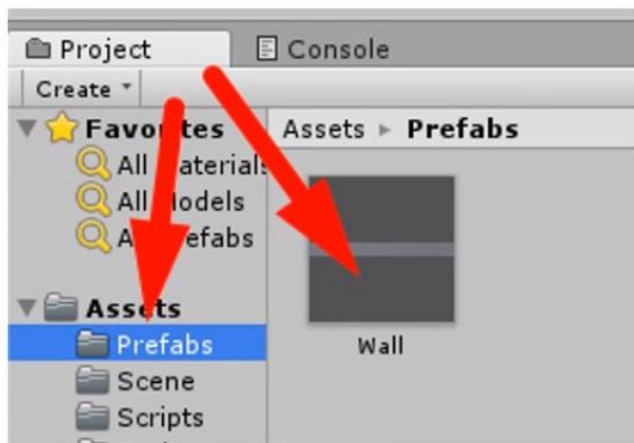


Чтобы мяч не улетал, нужно по бокам установить коллайдеры. Сделать это нужно также, как мы создавали объект Ground. Чтобы клонировать объект используйте префабы.

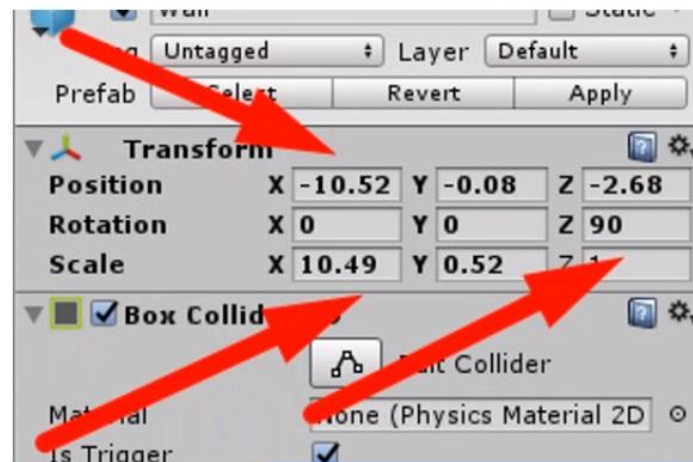
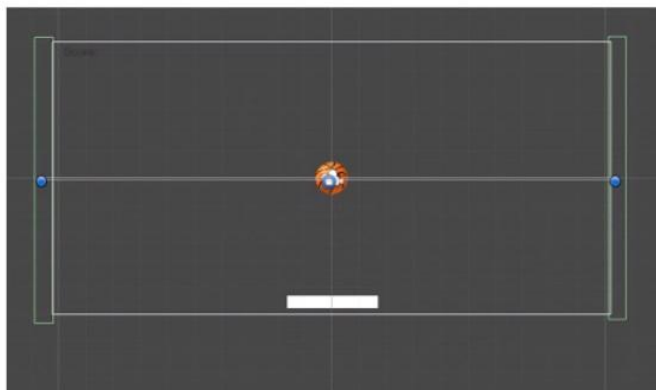
Добавляем префаб



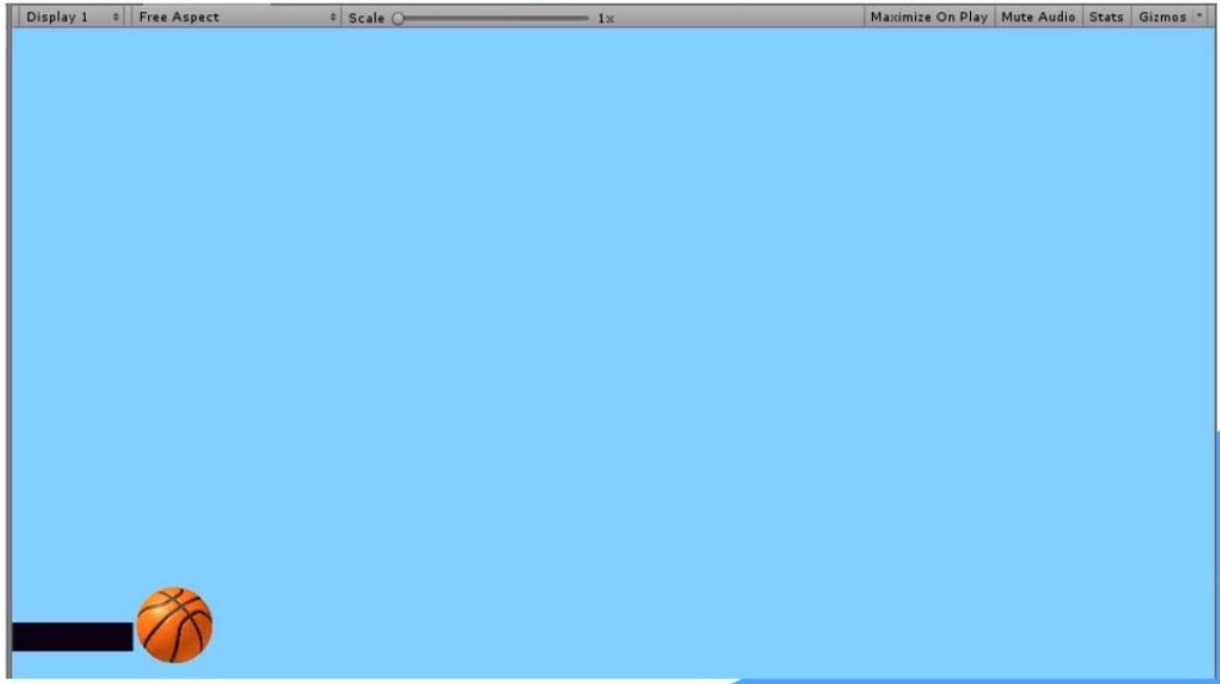
Добавляем префаб



Настраиваем



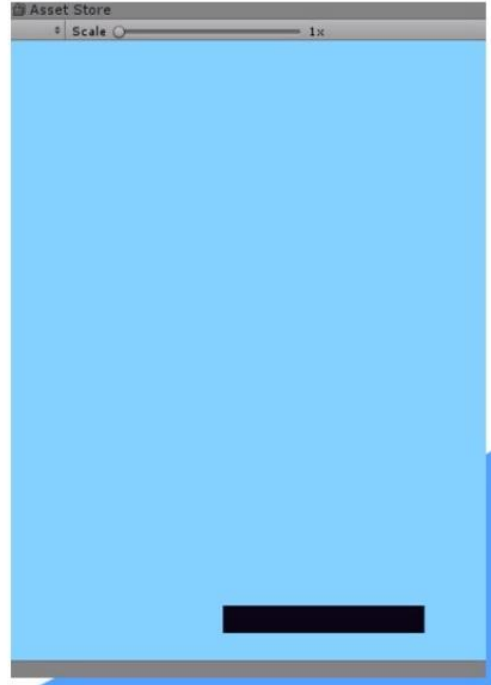
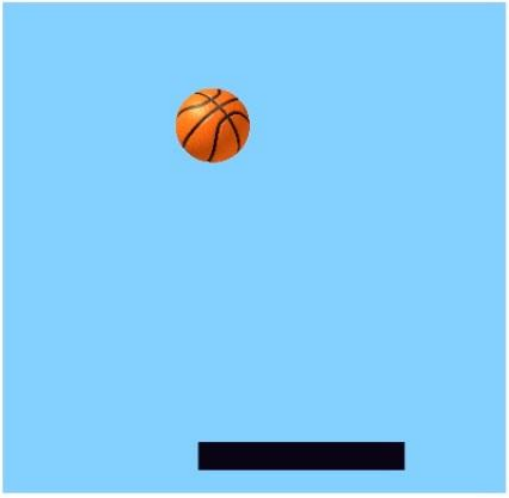
Запуск игры



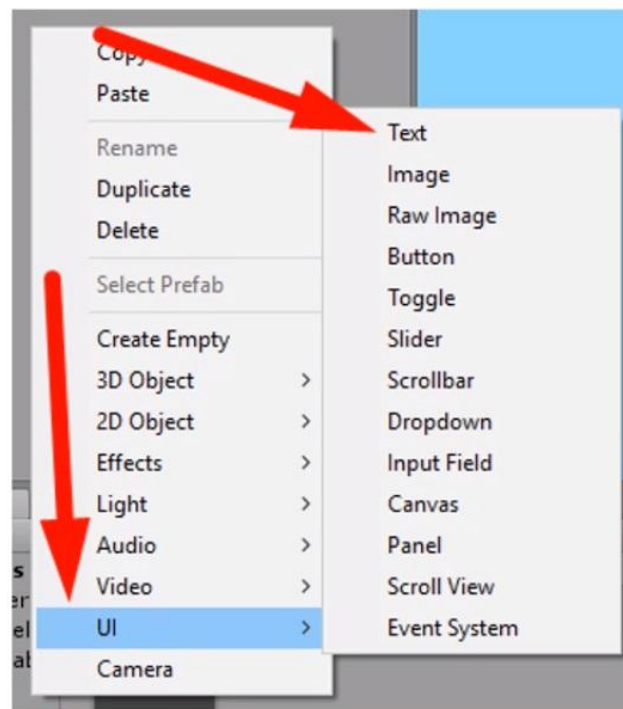
Настройка компонентов



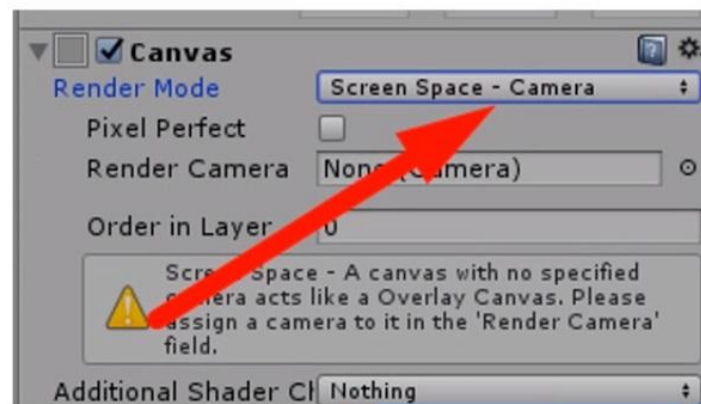
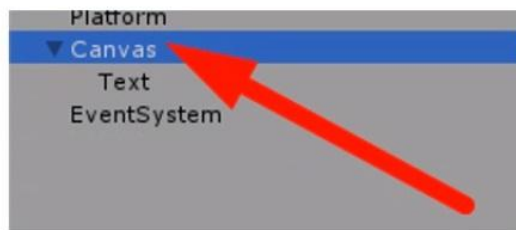
Запуск игры



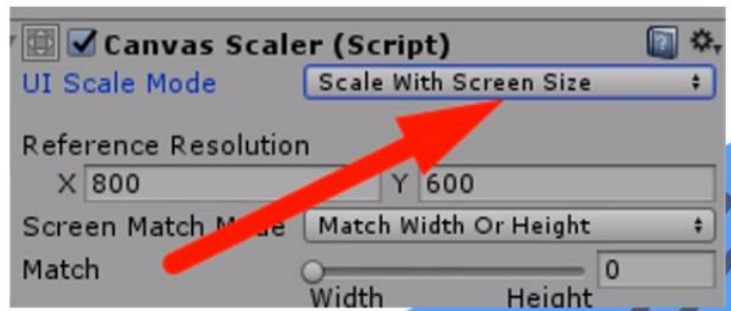
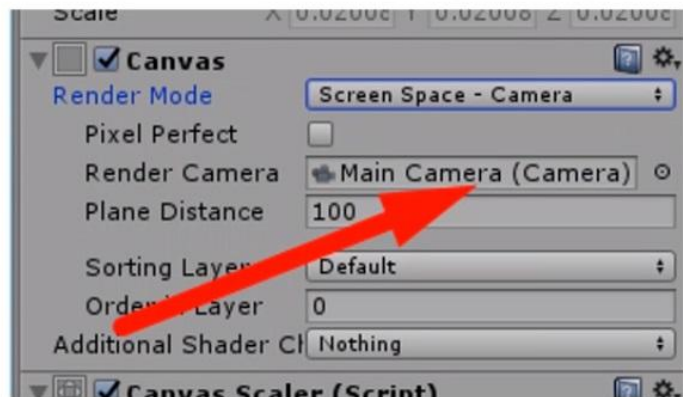
Добавляем текст



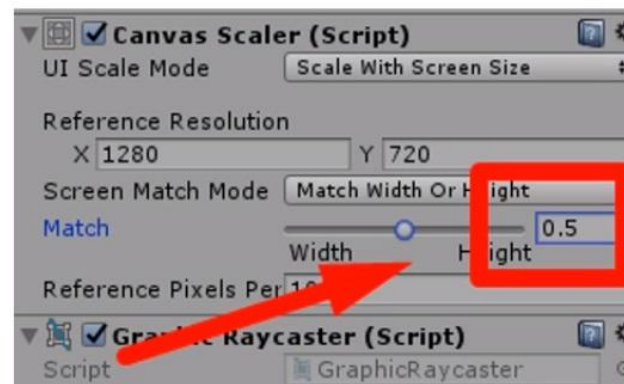
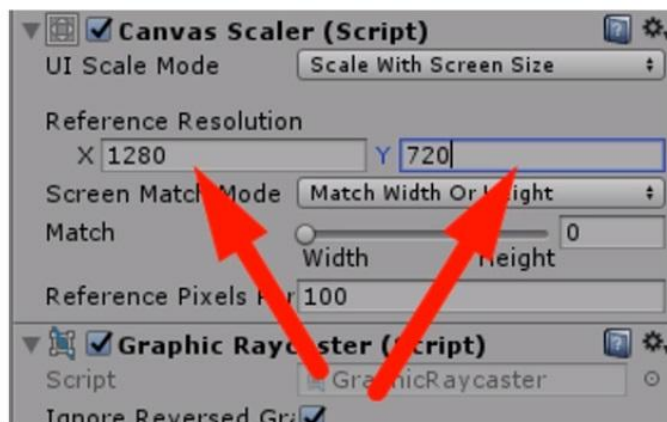
Настройка текста



Настройка текста

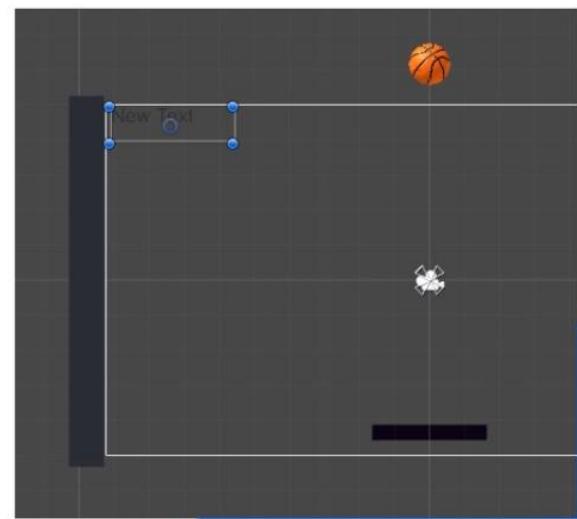



Настройка текста

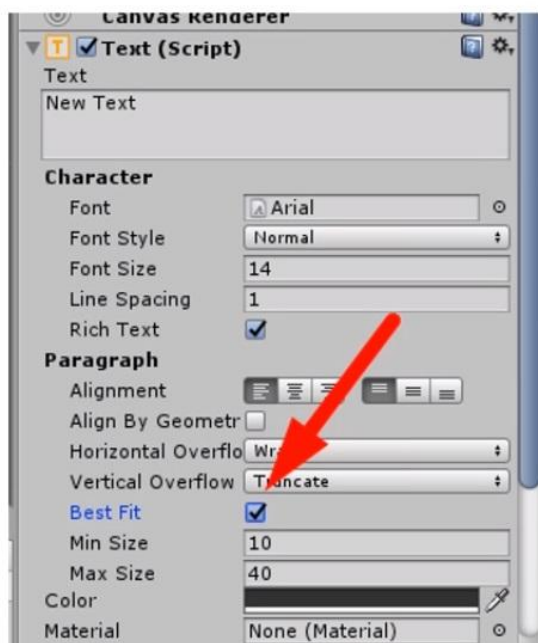


Местоположение

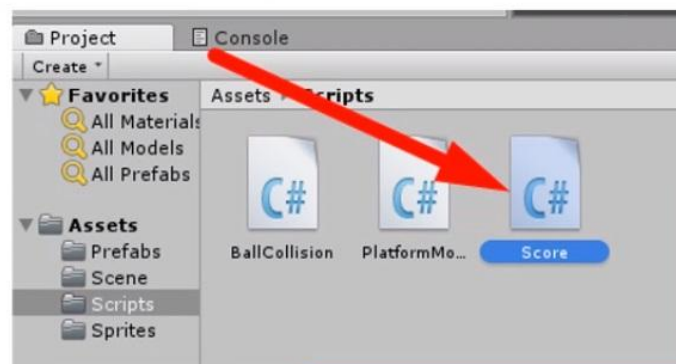
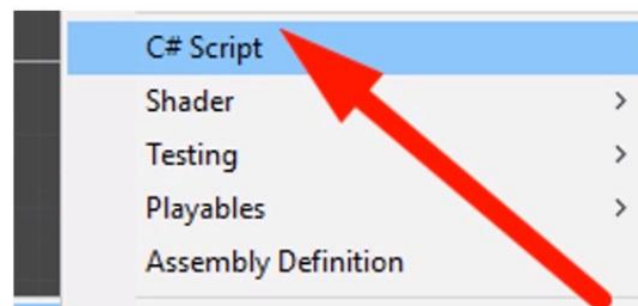
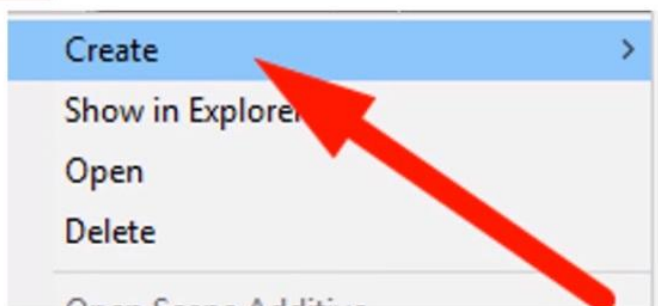
- Cube
- Platform
- ▼ Canvas
 - Text
- EventSystem



Настройка размера



Создание скрипта



Пишем код

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Score : MonoBehaviour {

    [SerializeField] private int score;
    [SerializeField] private Text scoreText;

    public int GameScore
    {
        get
        {
            return score;
        }
        set
        {
            score = value;
            scoreText.text = "Score: " + score;
        }
    }
}
```

Скрипт

76

```
public class BallCollision : MonoBehaviour {  
  
    [SerializeField] float impulse;  
    private Rigidbody2D rbody;  
    private Vector2 direction;  
  
    private Score score;
```

Скрипт

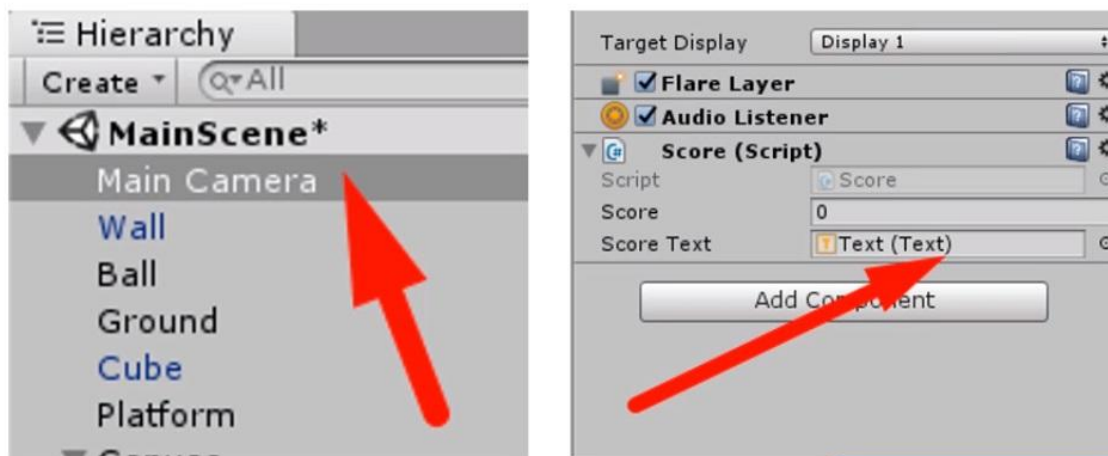
77

```
private Score score;  
  
void Start ()  
{  
    rbody = GetComponent<Rigidbody2D>();  
    score = Camera.main.GetComponent<Score>();  
}
```

Скрипт

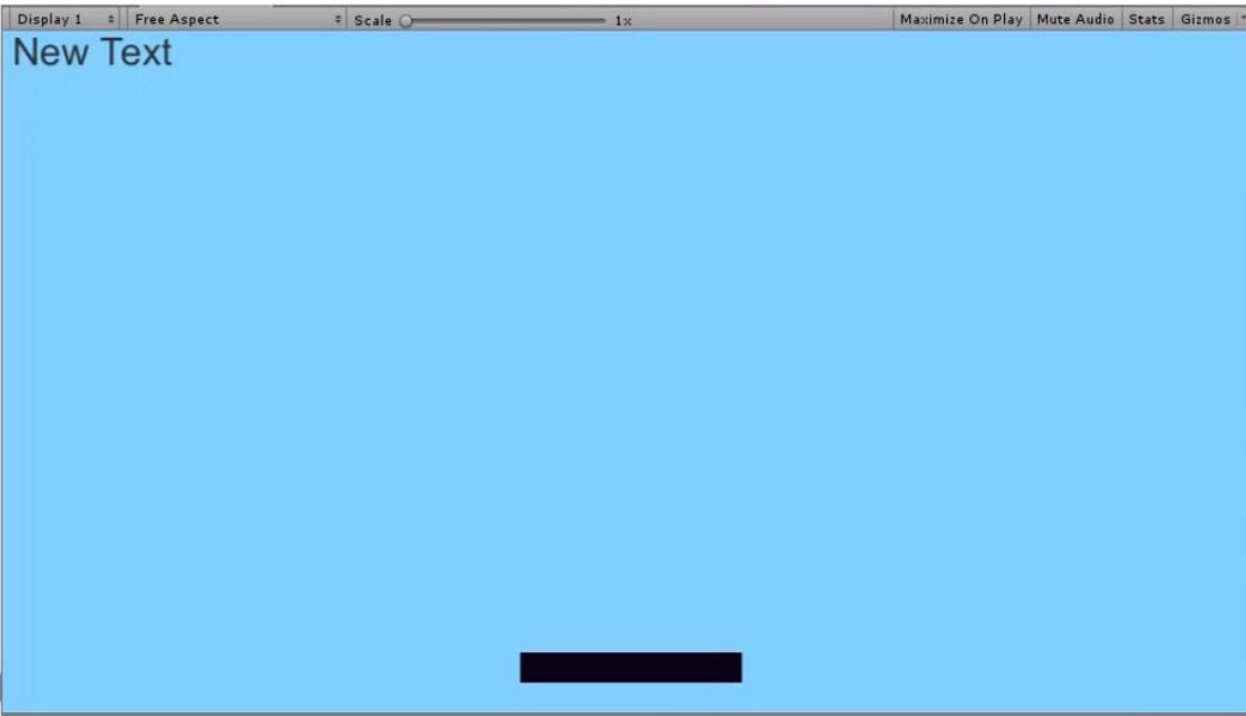
```
private void OnCollisionEnter2D(Collision2D collision)
{
    if (collision.gameObject.name == "Platform")
    {
        direction = new Vector2(Random.Range(-1, 1), Random.Range(3, 5));
        rbody.AddForce(direction * impulse, ForceMode2D.Impulse);
        score.GameScore += 1;
    }
}
```

Прикрепление скрипта



Запуск игры

80



Домашнее задание

Score: 3



Напишите код программы, который при падении мяча будет обнулять счет. Используйте метод `OnTriggerEnter2D`, так как мяч падает через объект `Ground`, у которого стоит галочка `Is Trigger`.