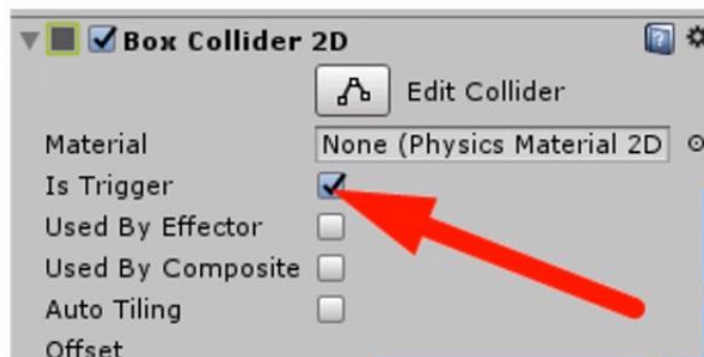
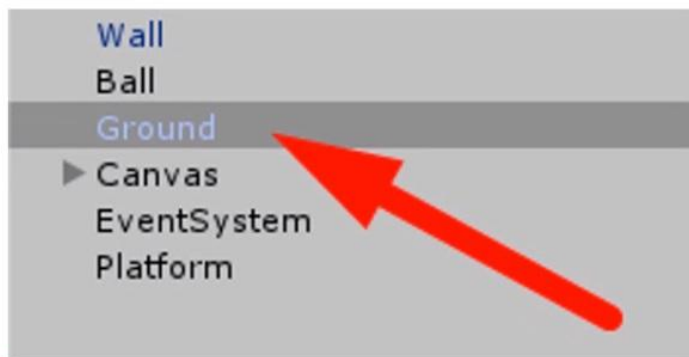


# Unity 3D

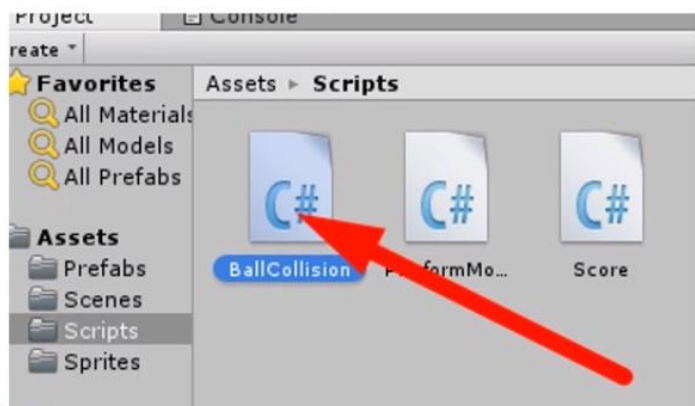
2D игра  
Массивы  
Урок 4



# Домашнее задание



## Домашнее задание



```
if (collision.gameObject.name == "Platform")  
{  
    direction = new Vector2(Random.Range(-1, 1), Ran  
    rbody.AddForce(direction * impulse, ForceMode2D.  
    score.GameScore += 1;  
}
```

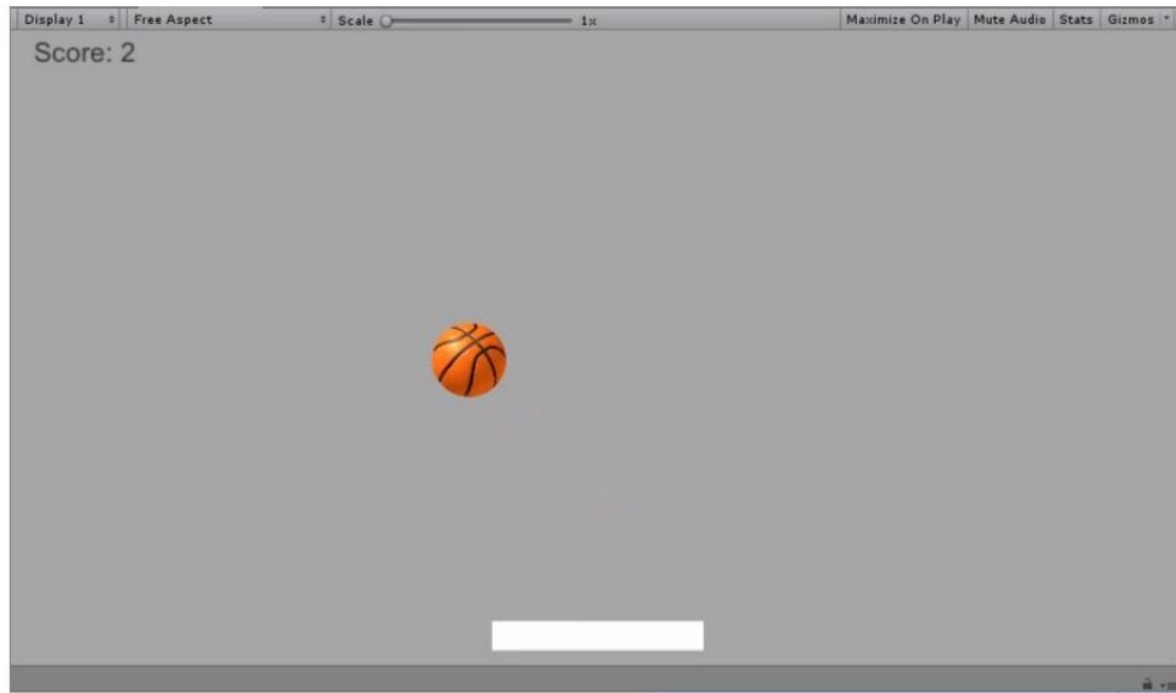
```
private void OnTriggerEnter2D(Collider2D collision)  
{  
}
```

## Домашнее задание

```
private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.gameObject.name == "Ground")
    {
        score.GameScore = 0;
    }
}
```

# Домашнее задание

5



# Игра

6



Score: 1

CODOLOGIA

# Создаем проект

7

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: ClickMe

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

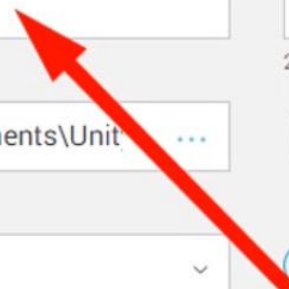
Organization: unity\_9DKUd5r\_s2fLcQ

Template: 2D

2D template: Add Asset Package

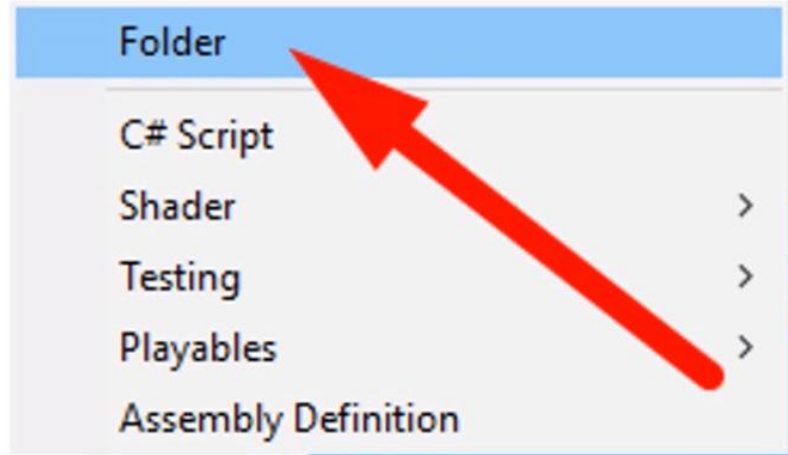
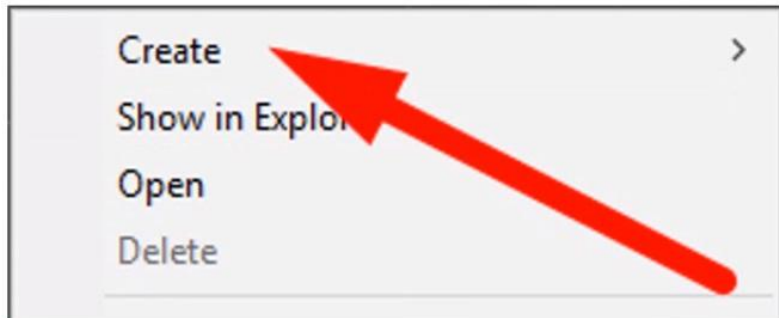
Enable Unity Analytics ?

Cancel Create project





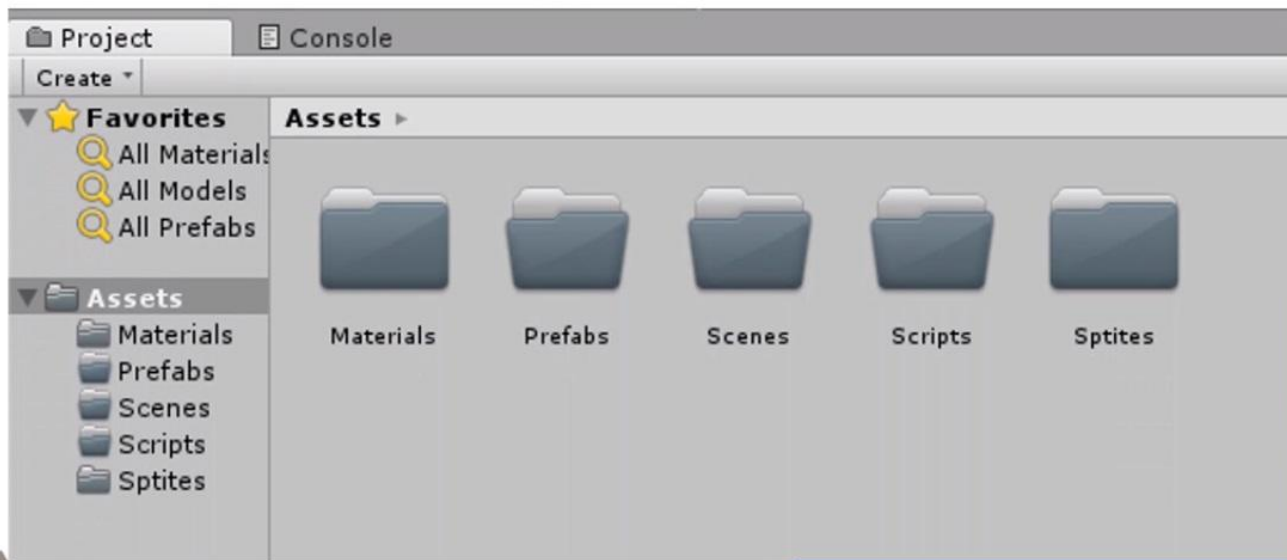
# Создание папок



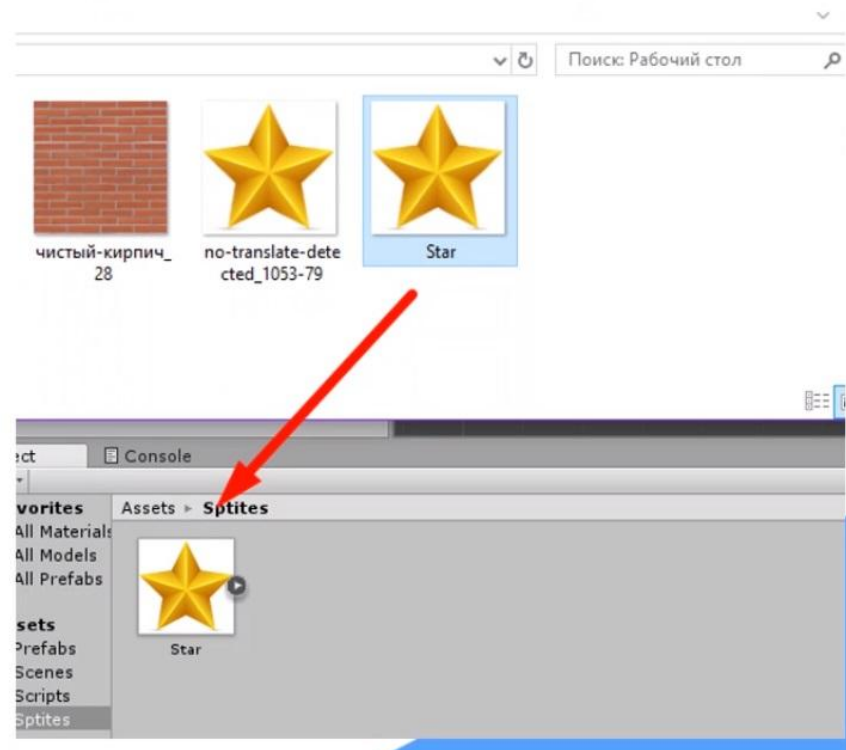
# Задание



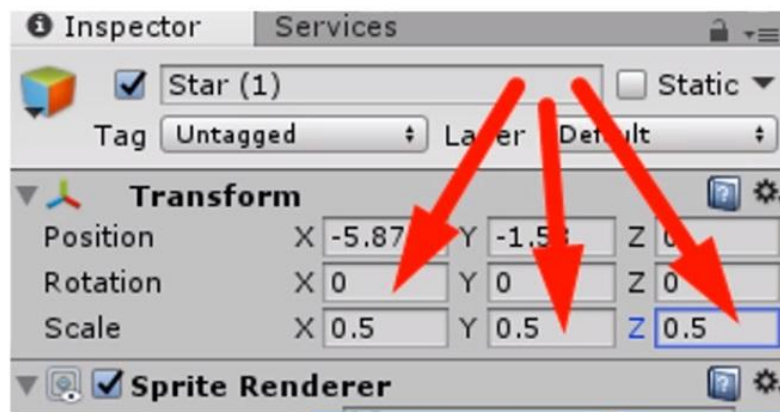
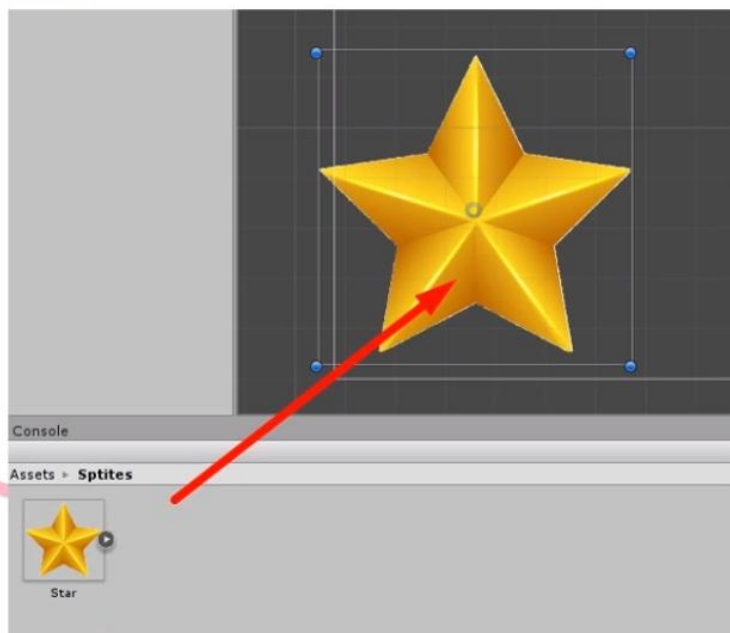
10



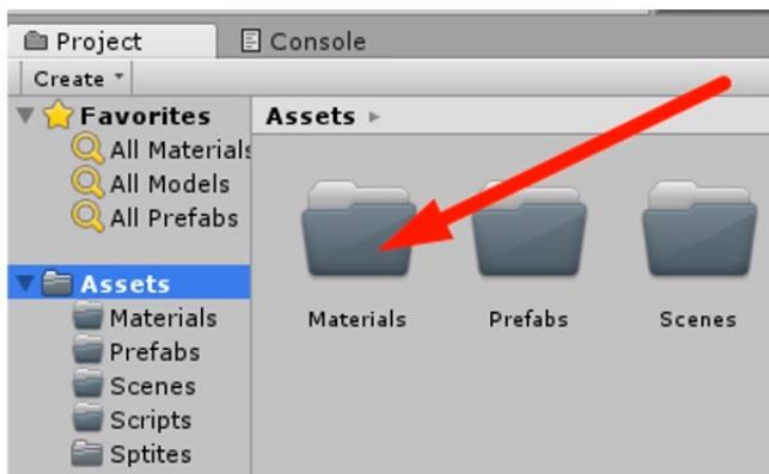
# Добавление спрайта



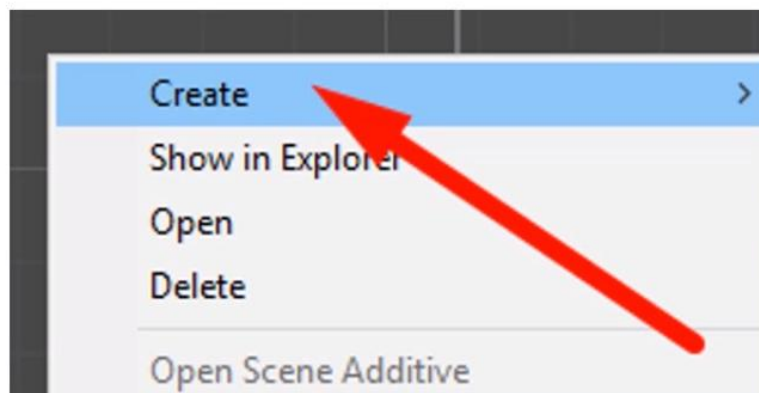
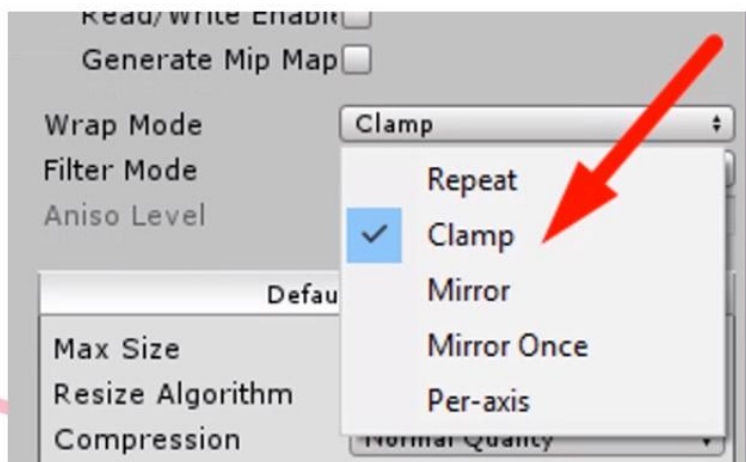
## Настройка спрайтов



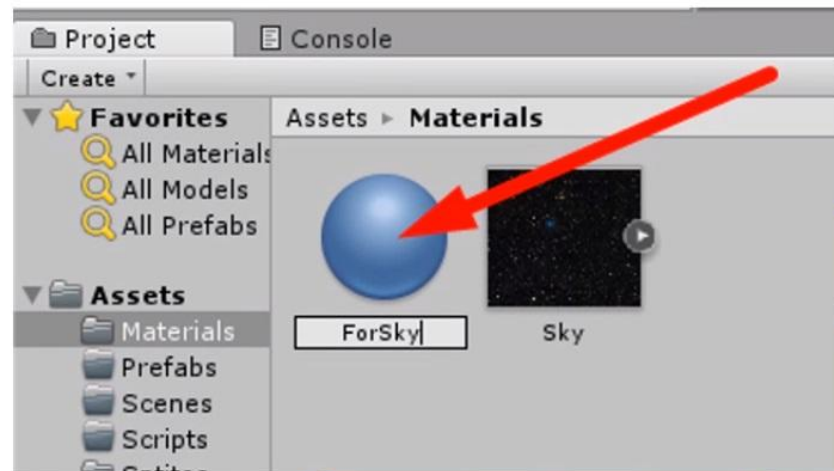
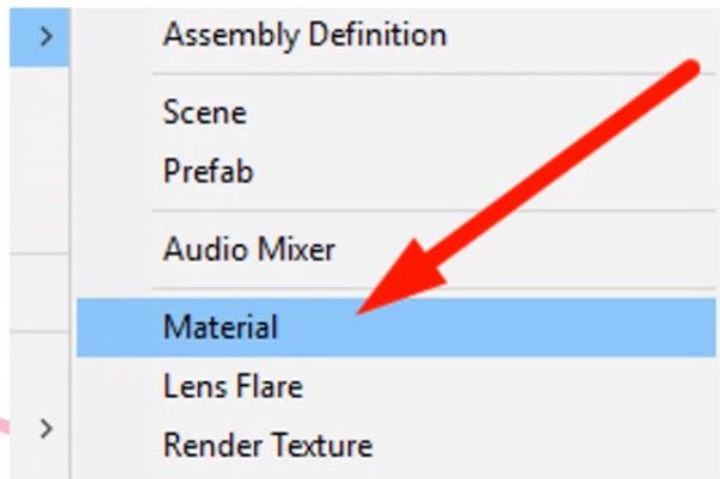
# Настройка материала



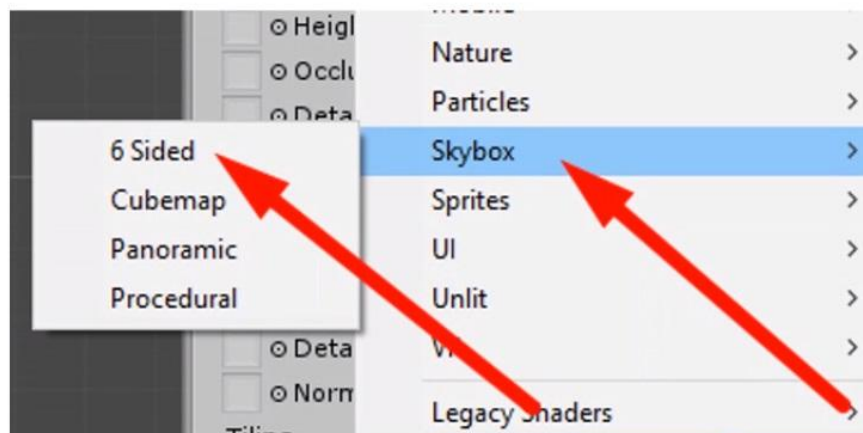
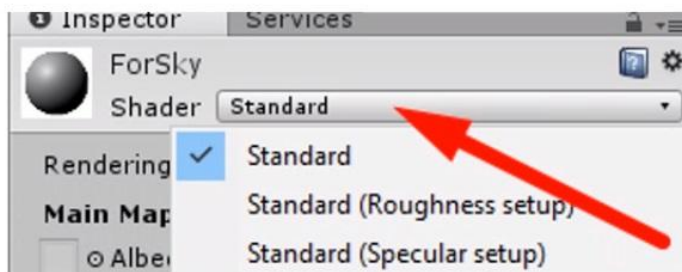
## Создание материала



## Создание материала

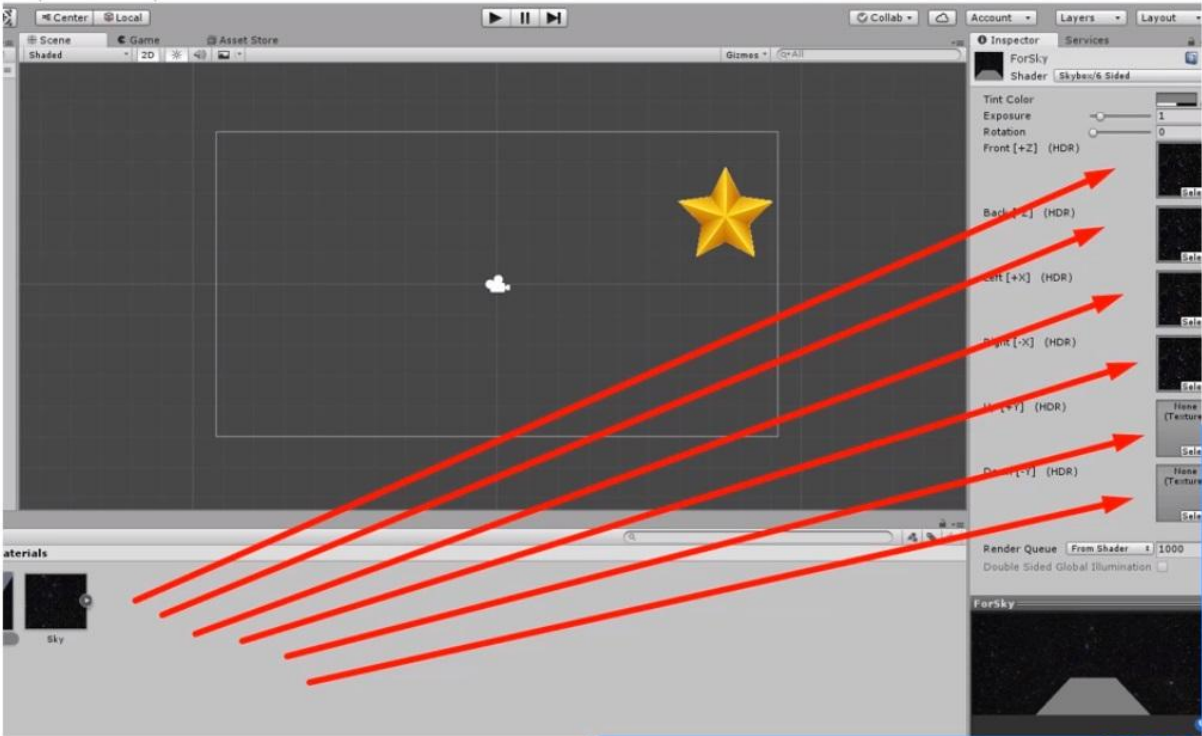


## Настраиваем SkyBox

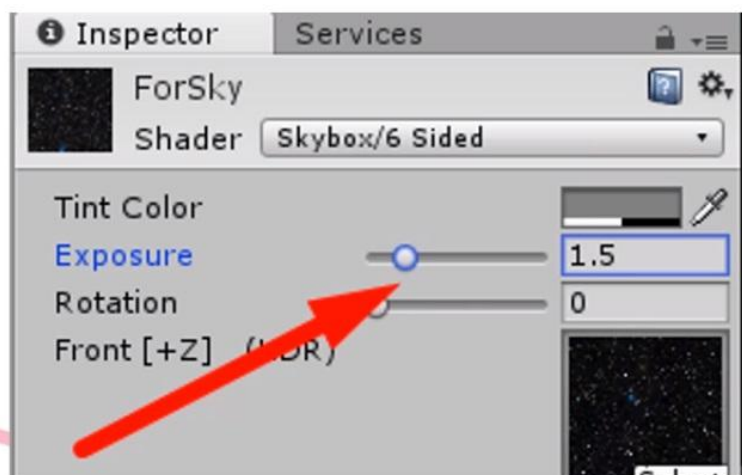




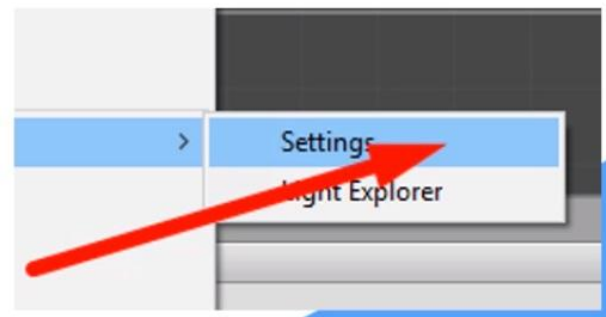
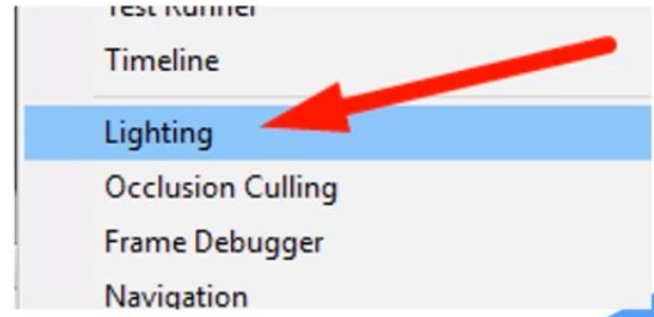
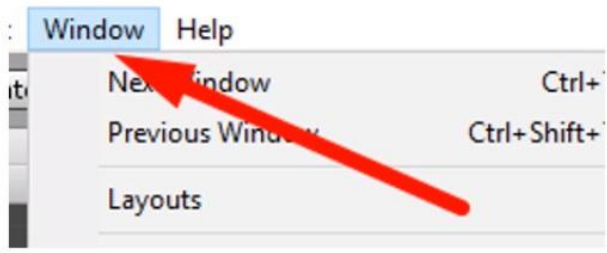
# Настраиваем SkyBox



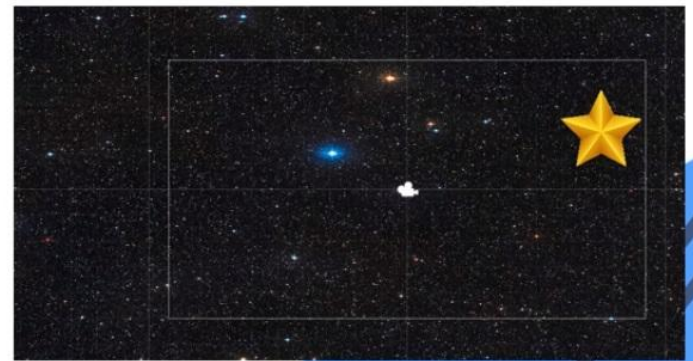
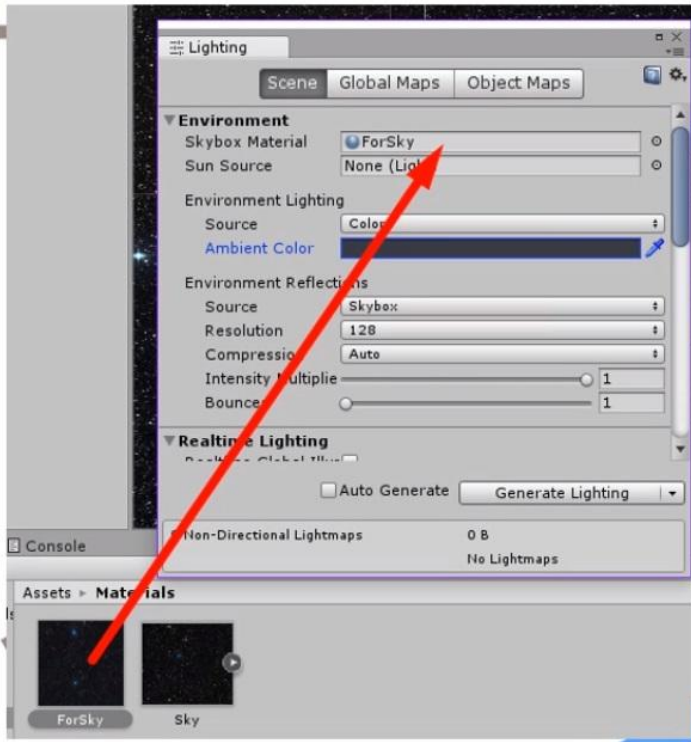
## Настраиваем SkyBox



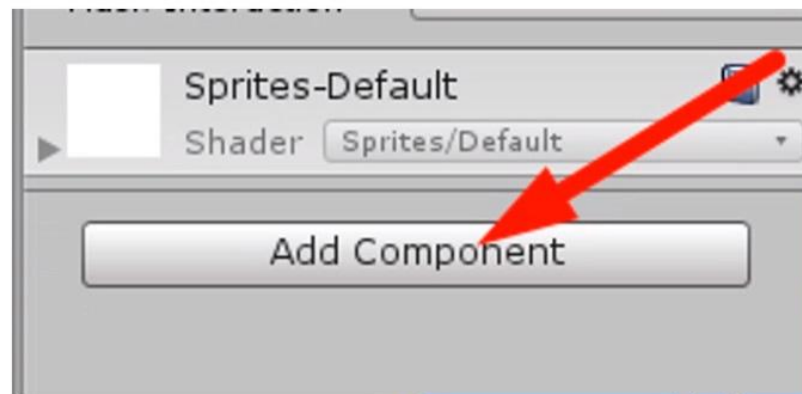
# Добавляем SkyBox на сцену



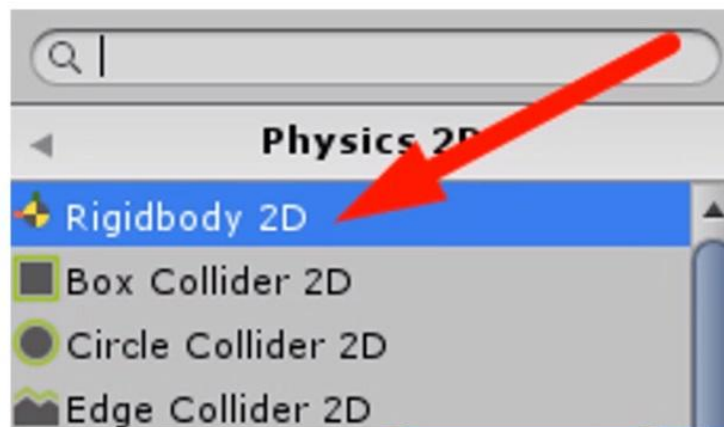
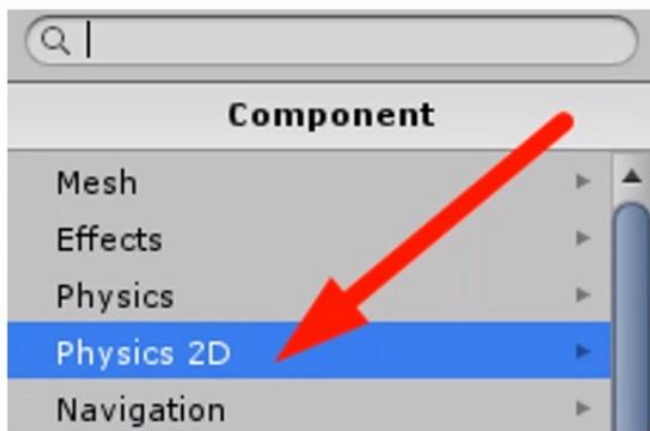
# Добавляем SkyBox на сцену



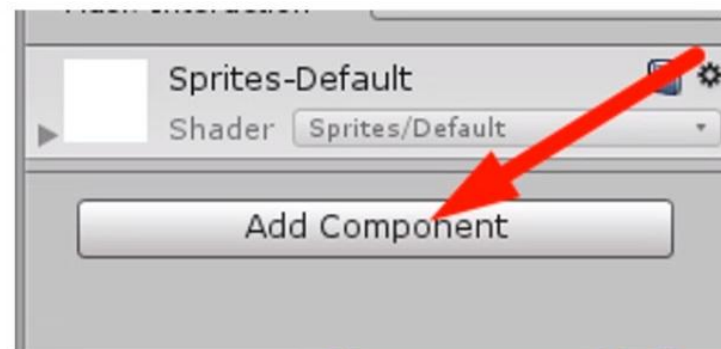
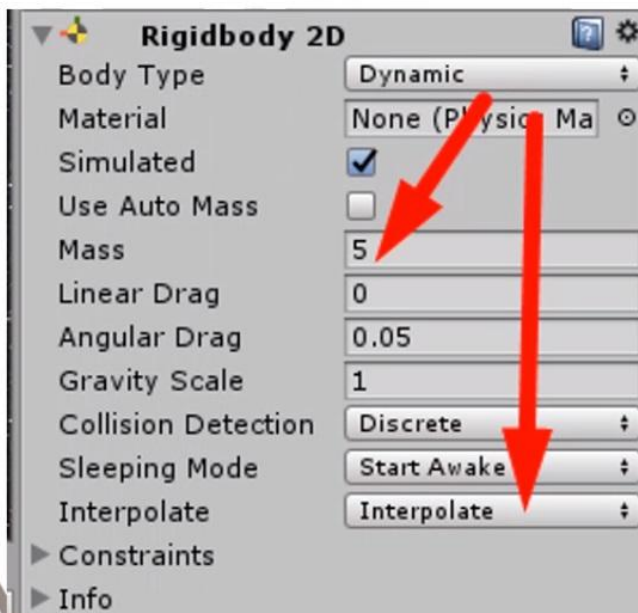
## Добавление компонентов



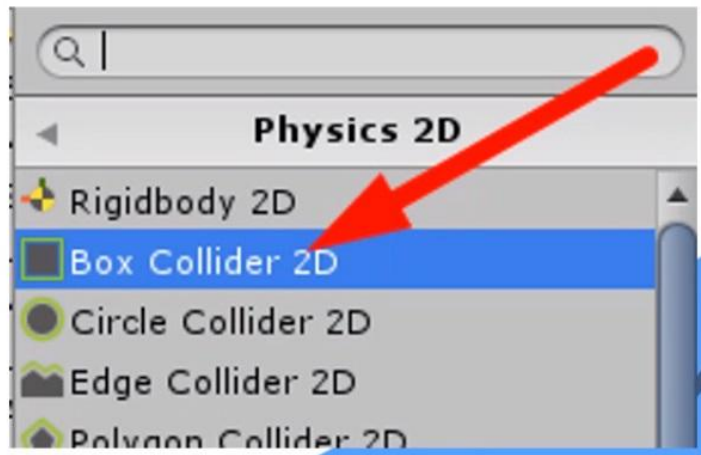
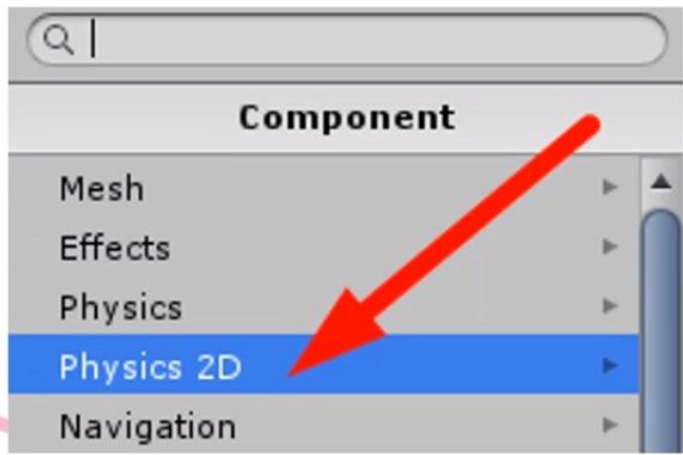
# Добавление компонентов



## Добавление компонентов



# Добавление компонентов





# Настройка компонентов

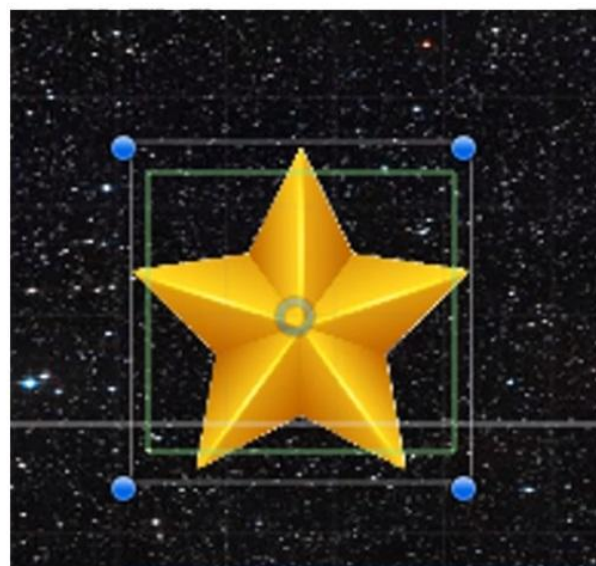
Auto Tiling

Offset  
X  Y

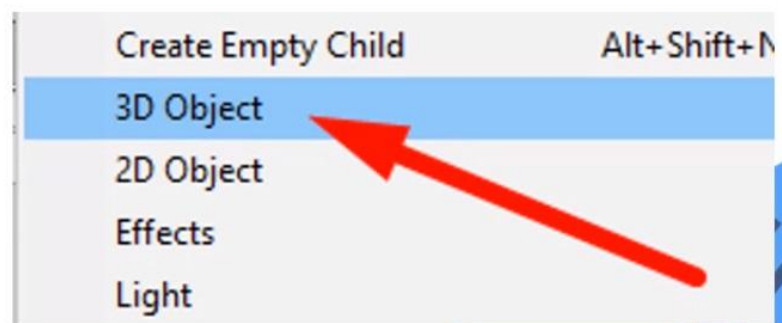
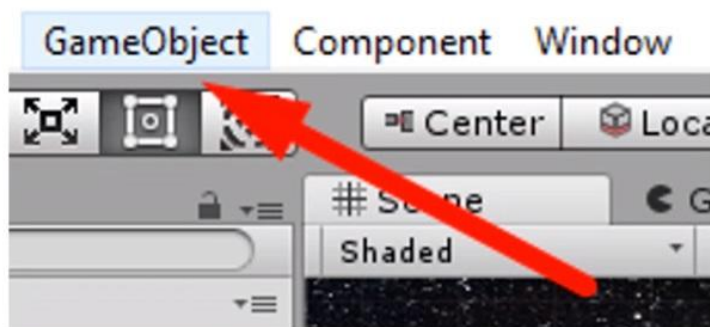
Size  
X  Y

Edge Radius

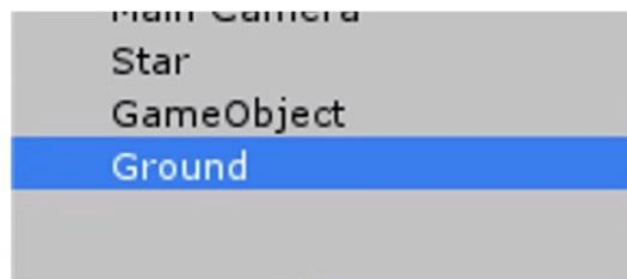
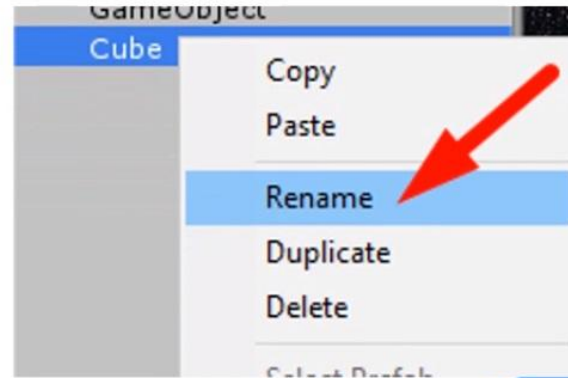
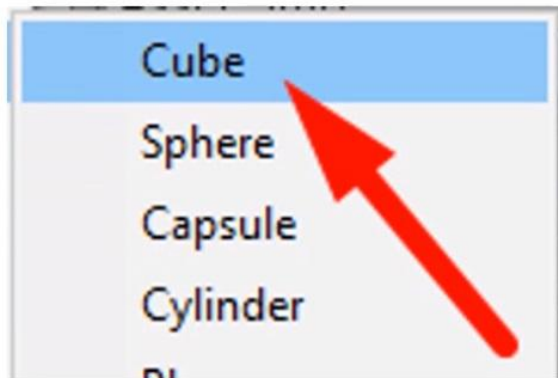
▶ Info



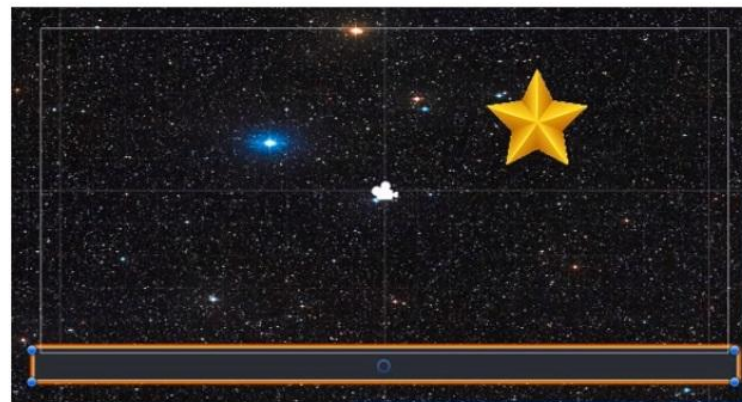
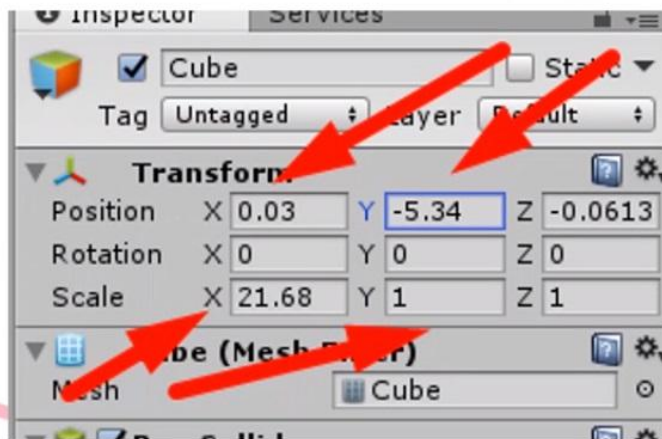
# Создание спрайта



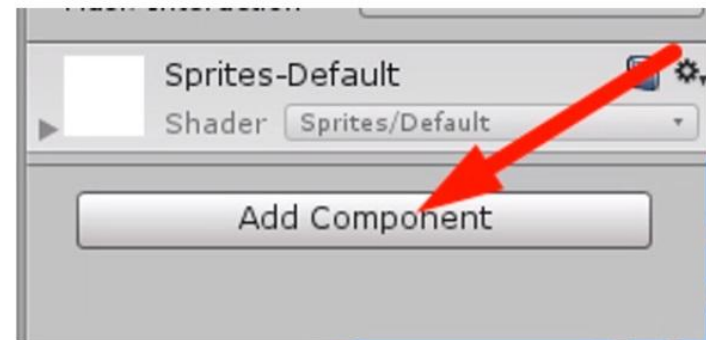
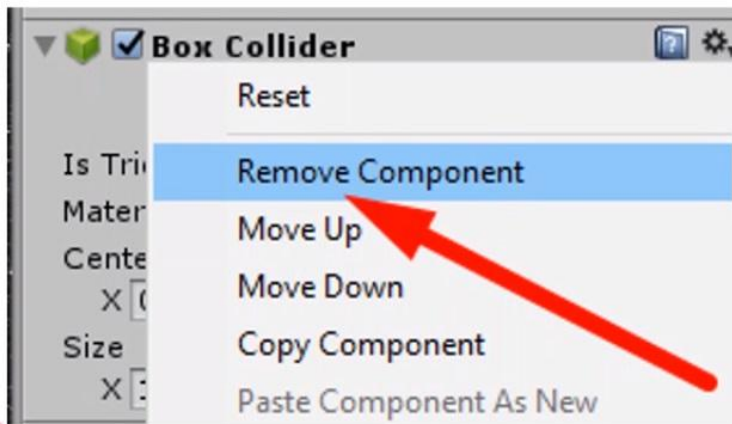
# Создание спрайта



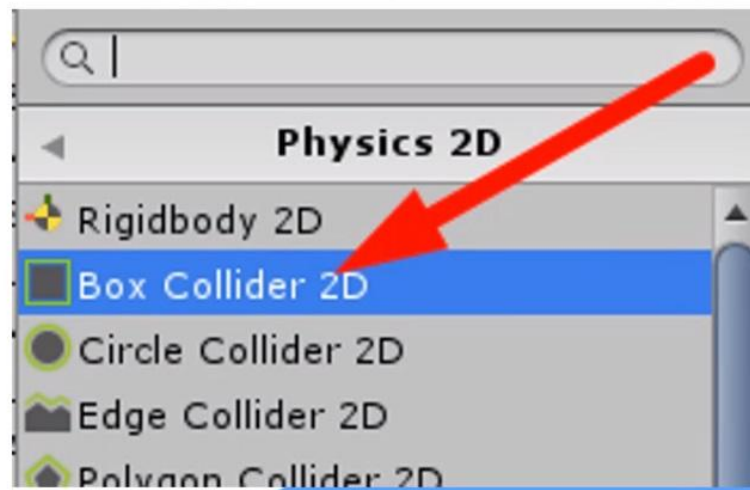
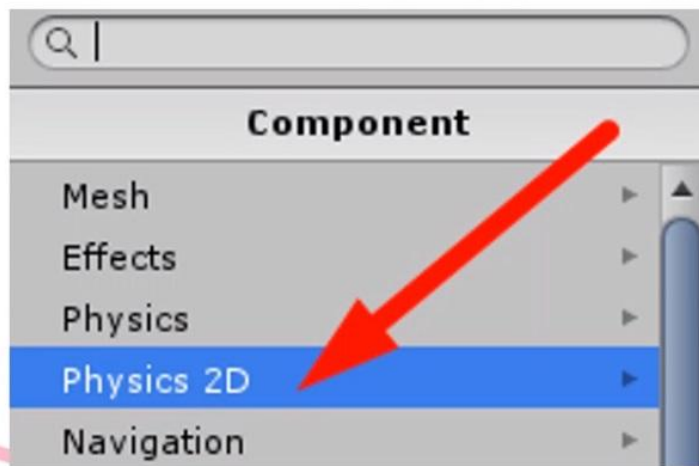
# Настройка компонентов



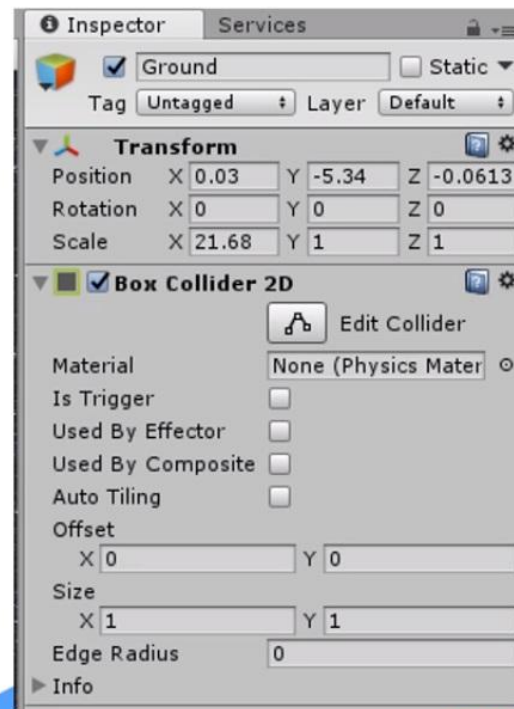
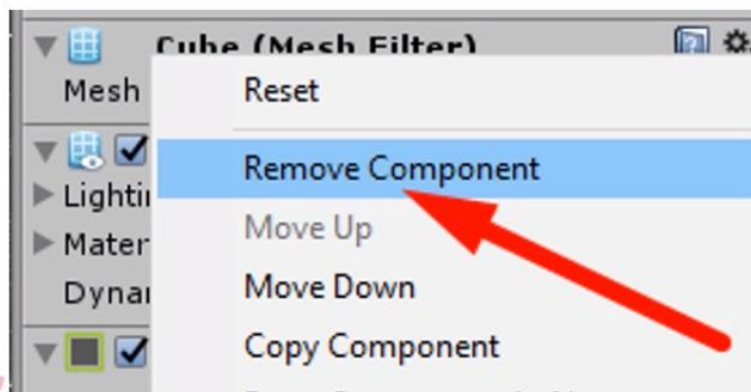
## Настройка компонентов



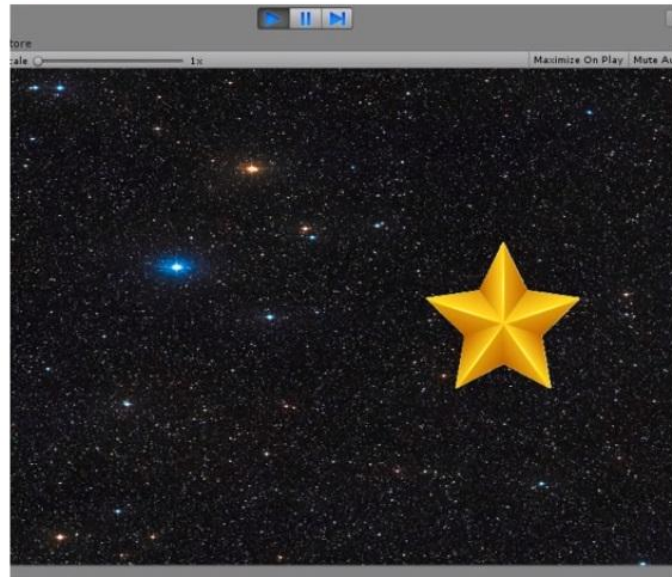
## Добавляем компоненты



## Удаляем компоненты

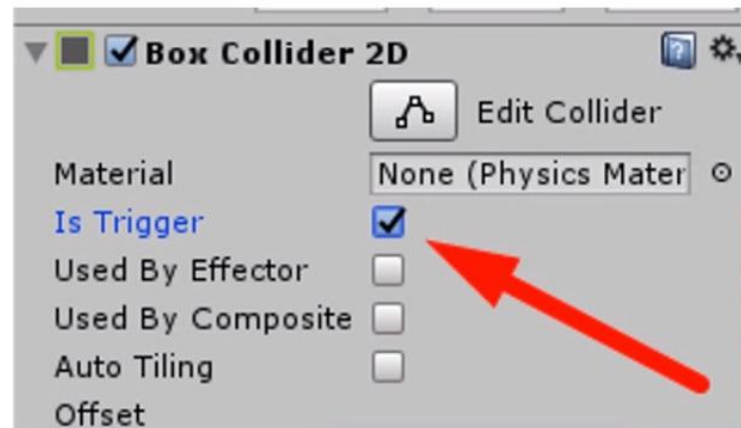


# Запускаем игру

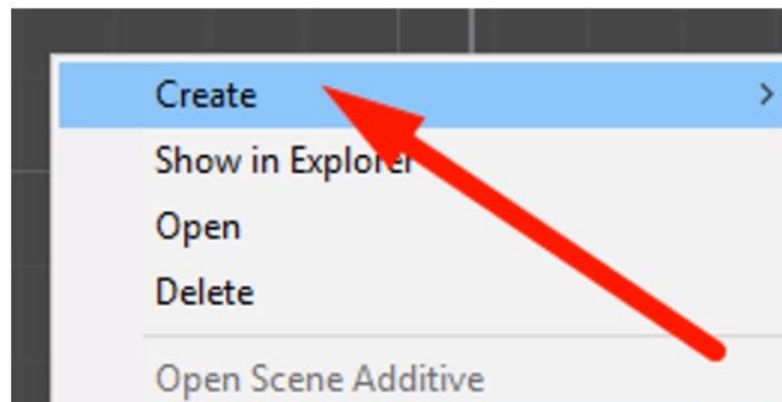
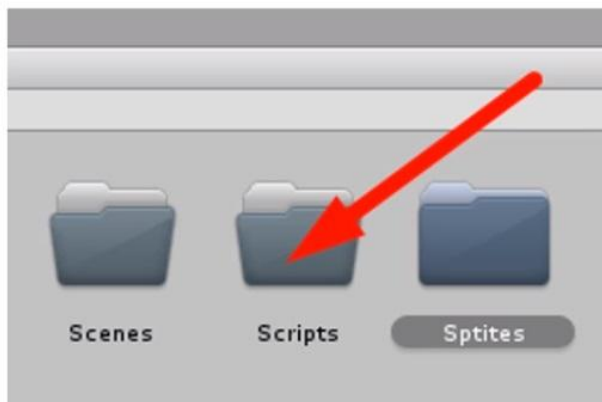




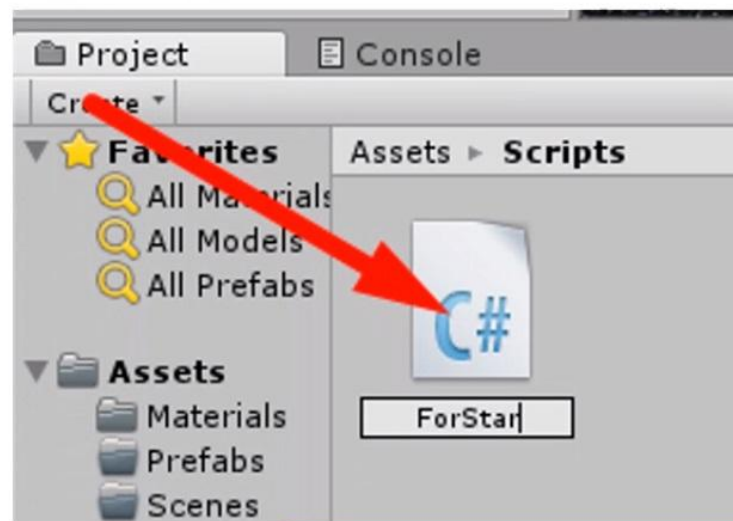
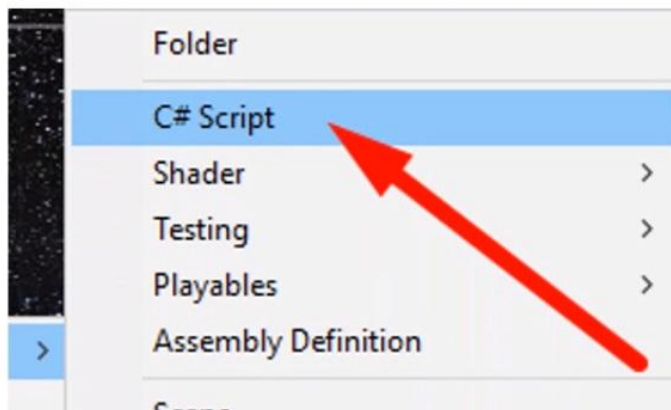
## Добавление компонентов



# Создание скрипта



## Создание скрипта

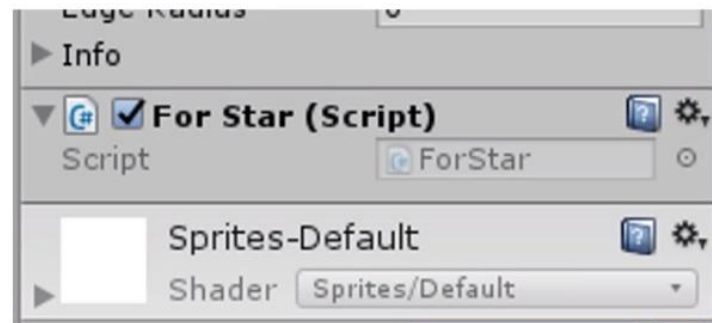


# Пишем код

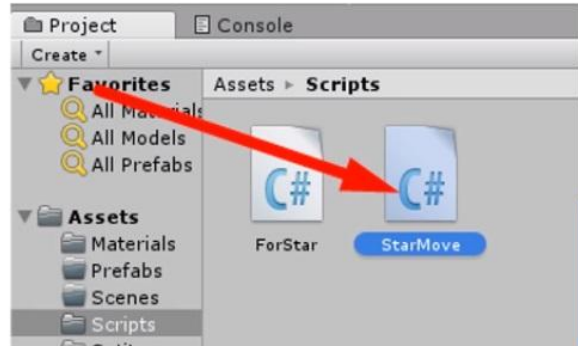
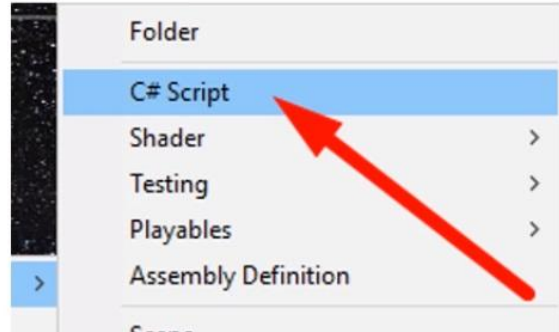
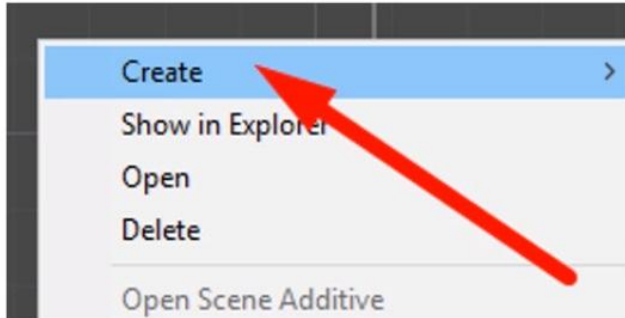
```
public class ForStar : MonoBehaviour {  
    private void OnMouseDown()  
    {  
        }  
}
```

```
public class ForStar : MonoBehaviour {  
    private void OnMouseDown()  
    {  
        Destroy(gameObject);  
    }  
}
```

## Прикрепляем скрипт



# Создаем скрипт



## Пишем код

```
public class StarMove : MonoBehaviour {  
  
    [SerializeField] private GameObject star;  
    [SerializeField] private float TimeStars;  
  
    GameObject[] Stars;  
    public int numberOfStars;  
}
```

# Массивы





# Пишем код

```
void Start()  
{  
    Stars = new GameObject[numberOfStars];  
}
```

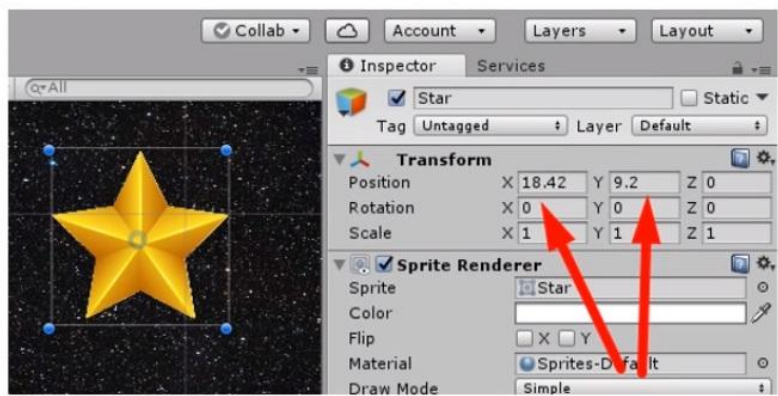
# Написание скрипта

```
IEnumerator SpawnNyushes()  
{  
    for (int i = 0; i < numberOfStars; i++)  
    {  
    }  
}
```

## Пишем скрипт

```
IEnumerator SpawnMyushes()  
{  
    for (int i = 0; i < numberOfStars; i++)  
    {  
        Stars[i] = Instantiate(star, new Vector3(), Quaternion.identity);  
        yield return new WaitForSeconds(TimeStars);  
    }  
}
```

# Задание



Найдите координаты  
возможного появления звезд.

## Пишем код

```
es()  
  
< numberOfStars; i++)  
  
stantiate(star, new Vector3(Random.Range(-9f, 10f), Random.Range(6f, 10f), 0), Quaternion.identity);  
ew WaitForSeconds(TimeStars);
```



## Пишем код

```
IEnumerator SpawnNyushes()
{
    for (int i = 0; i < numberOfStars; i++)
    {
        Stars[i] = Instantiate(star, new Vector3(Random.Range(-9f, 10f), Random.Range(6f, 10f),
        yield return new WaitForSeconds(TimeStars));
    }
}

while (true)
{
    for (int i = 0; i < numberOfStars; i++)
    {
        // ...
    }
}
```

## Пишем код

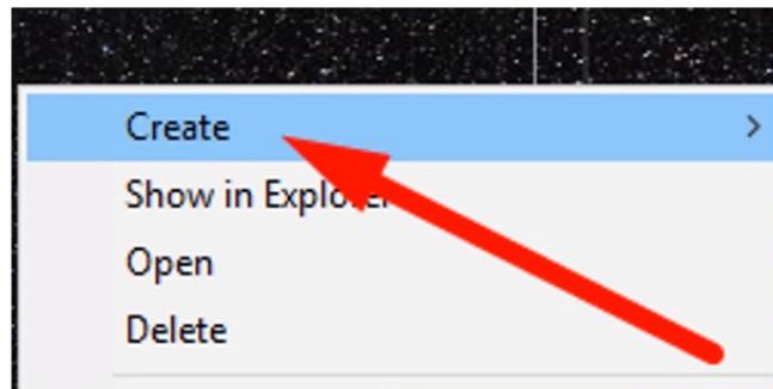
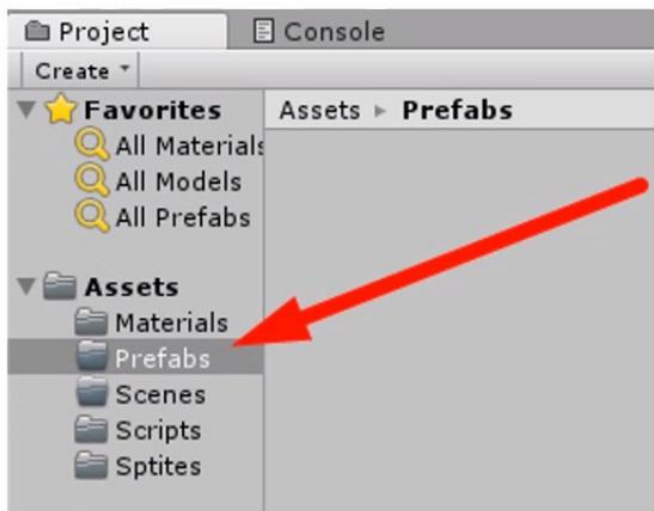
```
IEnumerator SpawnNyushes()
{
    for (int i = 0; i < numberOfStars; i++)
    {
        Stars[i] = Instantiate(star, new Vector3(Random.Range(-9f, 10f), Random.Range(6f, 10f), 0), Quaternion.identity);
        yield return new WaitForSeconds(TimeStars);
    }
    while (true)
    {
        for (int i = 0; i < numberOfStars; i++)
        {
            Stars[i].transform.position = new Vector3(Random.Range(-9f, 10f), Random.Range(6f, 10f), 0);
            yield return new WaitForSeconds(TimeStars);
        }
    }
}
```

## Пишем код

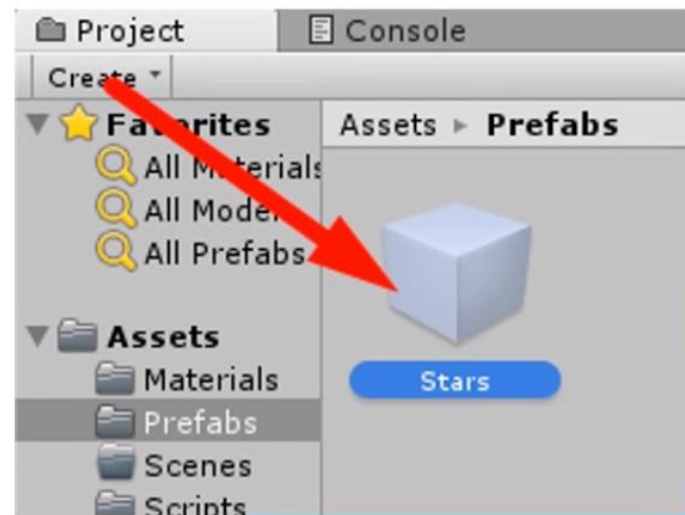
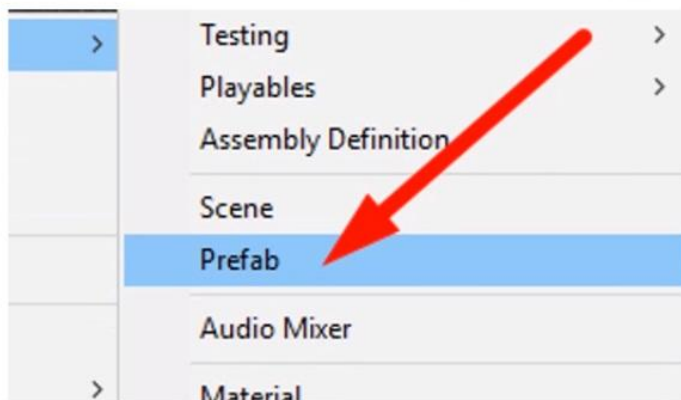
```
void Start()  
{  
    Stars = new GameObject[numberOfStars];  
    StartCoroutine(SpawnNyushes());  
}  
  
IEnumerator SpawnNyushes()  
{  
    for (int i = 0; i < numberOfStars; i++)  
    {
```



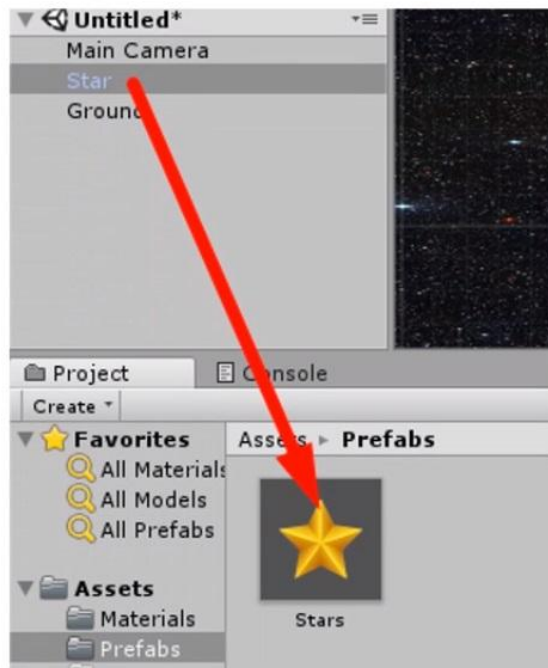
# Создаем префаб



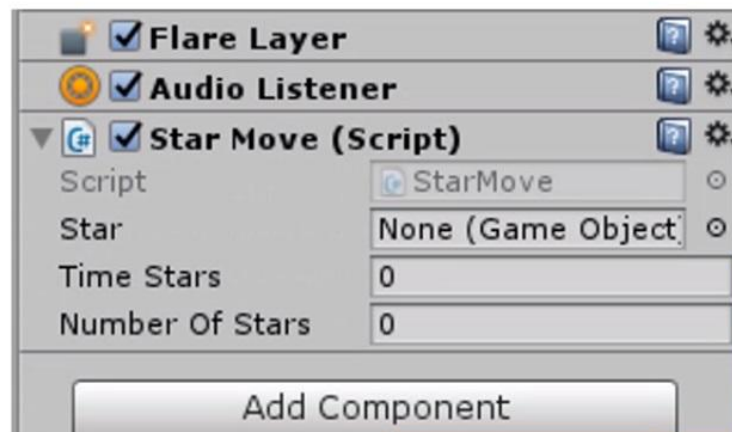
## Создаем префаб



# Создаем префаб



## Прикрепляем скрипт

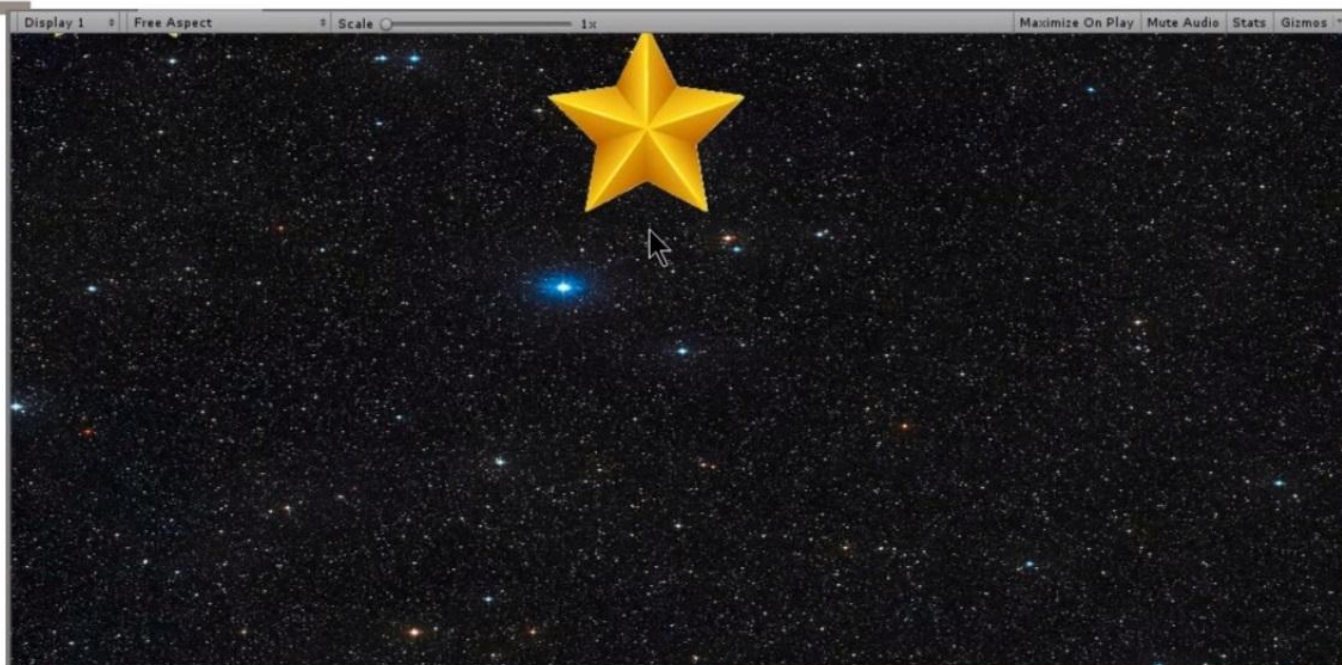


## Настраиваем значения

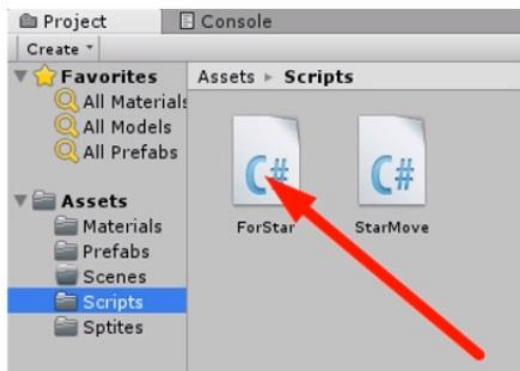
The image displays two side-by-side screenshots of the Unity Inspector window, illustrating the configuration of the 'Star Move (Script)' component. In the left screenshot, the 'Time Stars' field is set to 0. In the right screenshot, the 'Number Of Stars' field is set to 5. Red arrows highlight the input fields for 'Time Stars' and 'Number Of Stars' in both screenshots.

Field	Value
Script	StarMove
Star	Stars
Time Stars	0
Number Of Stars	5

## Запускаем игру

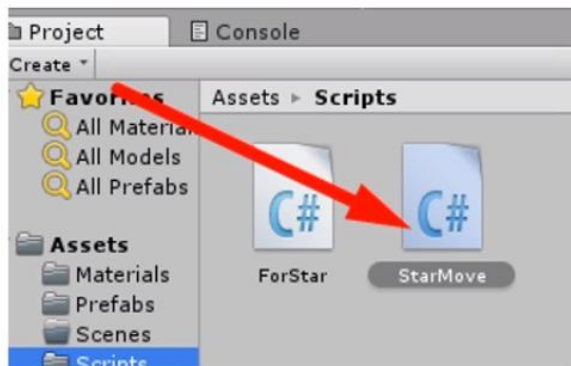


## Пишем код



```
public class ForStar : MonoBehaviour {  
  
    private Rigidbody2D rbody;  
    Vector3 start = new Vector3(0, -15f, 0);  
  
    private void OnMouseDown()  
    {  
        transform.position = start;  
    }  
}
```

# Пишем код



```
public class StarMove : MonoBehaviour {  
  
    [SerializeField] private GameObject stars;  
    [SerializeField] private float timeStars;  
  
    Vector3 velocity = new Vector3(0, 0, 0);  
    float position;  
  
    GameObject[] Stars;  
    public int numberOfStars;  
}
```




## Пишем код

```
IEnumerator SpawnMyushes()
{
    for (int i = 0; i < numberOfStars; i++)
    {
        Stars[i] = Instantiate(star, new Vector3(Random.Range(-9f, 10f), Random.Range(6f, 10f), 0), Quaternion.i
        position = Random.Range(0.3f, 0.6f);
        Stars[i].transform.localScale = new Vector3(position, position, position);
        yield return new WaitForSeconds(TimeStars);
    }
    while (true)
    {
        for (int i = 0; i < numberOfStars; i++)
```

## Пишем код

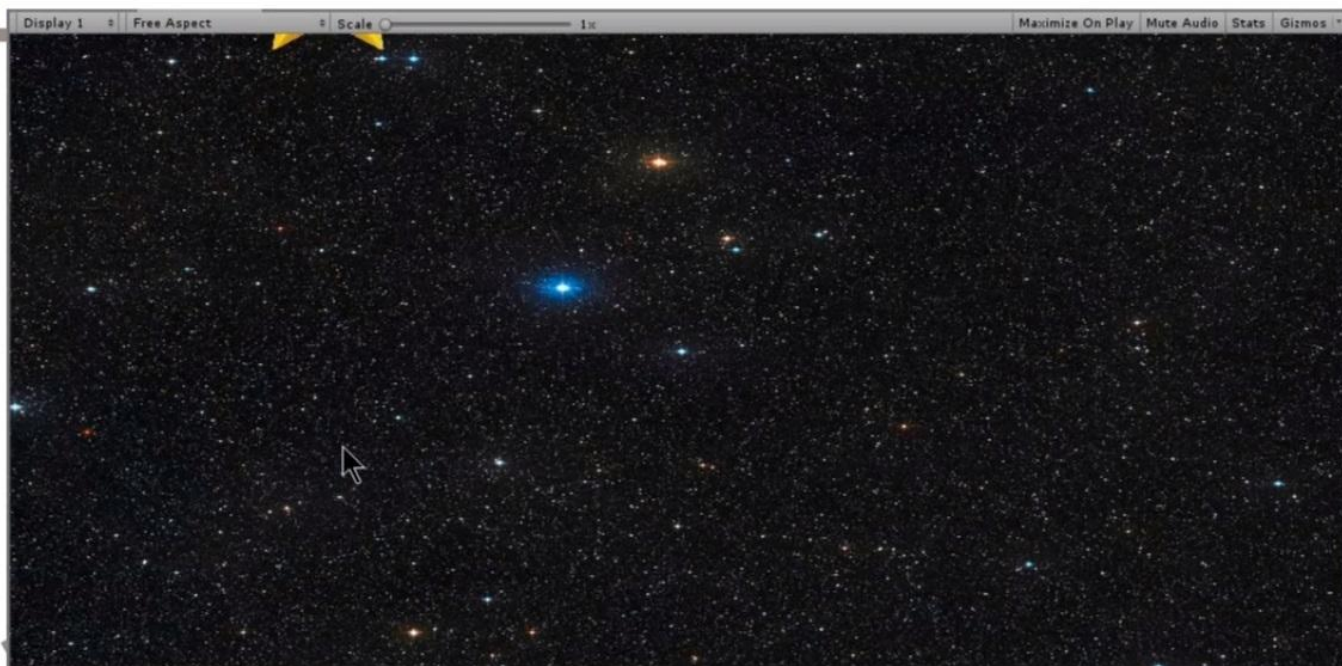
```
yield return new WaitForSeconds(TimeStars);  
}  
while (true)  
{  
    for (int i = 0; i < numberOfStars; i++)  
    {  
        position = Random.Range(0.3f, 0.6f);  
        Stars[i].transform.localScale = new Vector3(position, position, position);  
        Stars[i].transform.position = new Vector3(Random.Range(-9f, 10f), Random.Range  
        yield return new WaitForSeconds(TimeStars);  
    }  
}
```



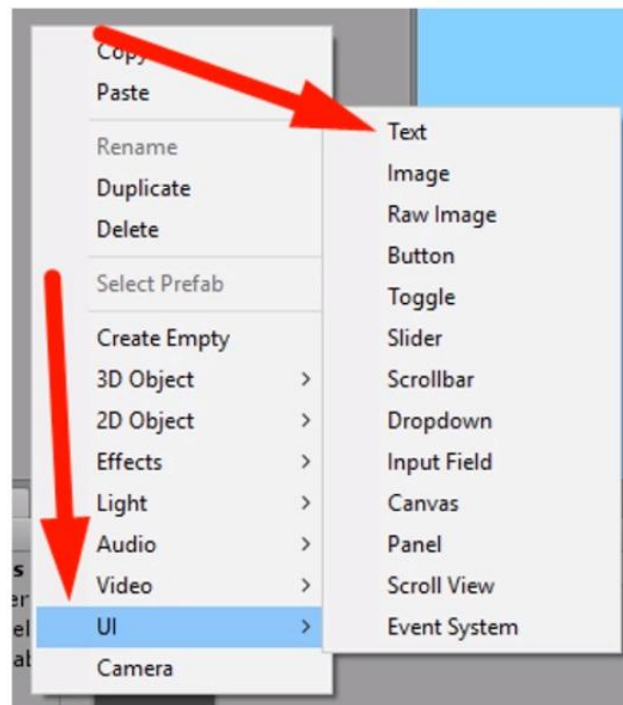
## Пишем код

```
}  
while (true)  
{  
    for (int i = 0; i < numberOfStars; i++)  
    {  
        position = Random.Range(0.3f, 0.6f);  
        Stars[i].transform.localScale = new Vector3(position, position, position);  
  
        Stars[i].GetComponent<Rigidbody2D>().velocity = new Vector3(0, 0, 0);  
  
        Stars[i].transform.position = new Vector3(Random.Range(-9f, 10f), Random.Range(6f, 10f), 0);  
        yield return new WaitForSeconds(TimeStars);  
    }  
}
```

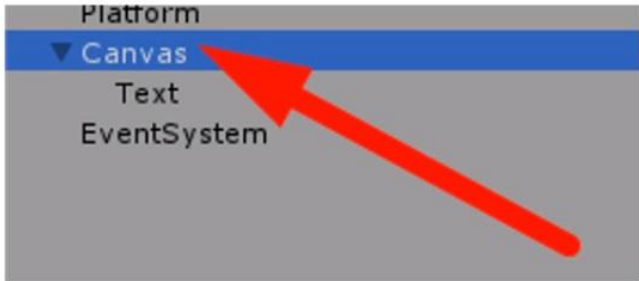
## Запускаем игру



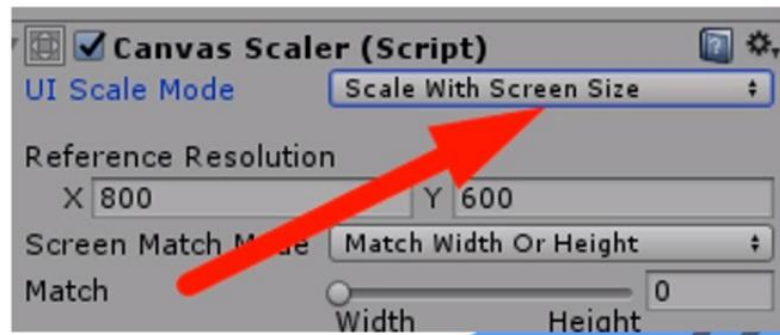
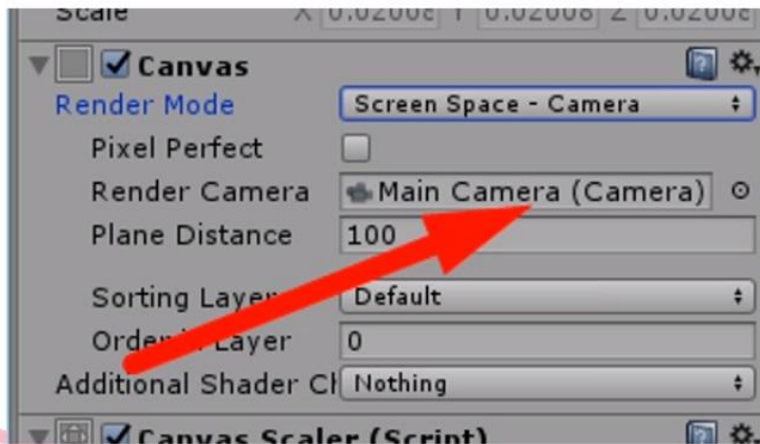
## Добавляем текст



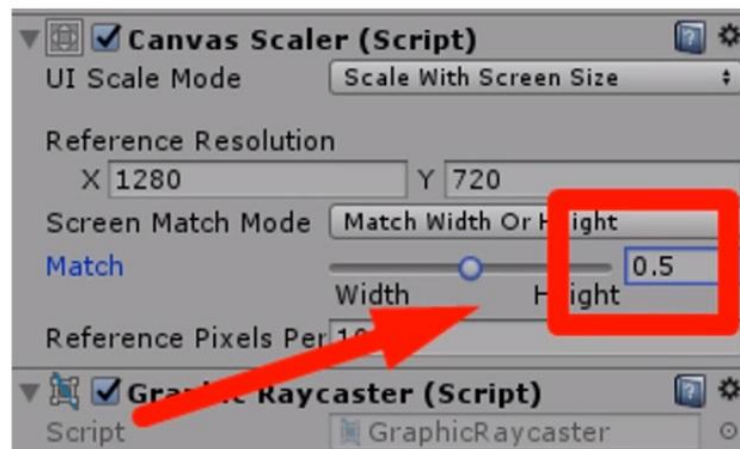
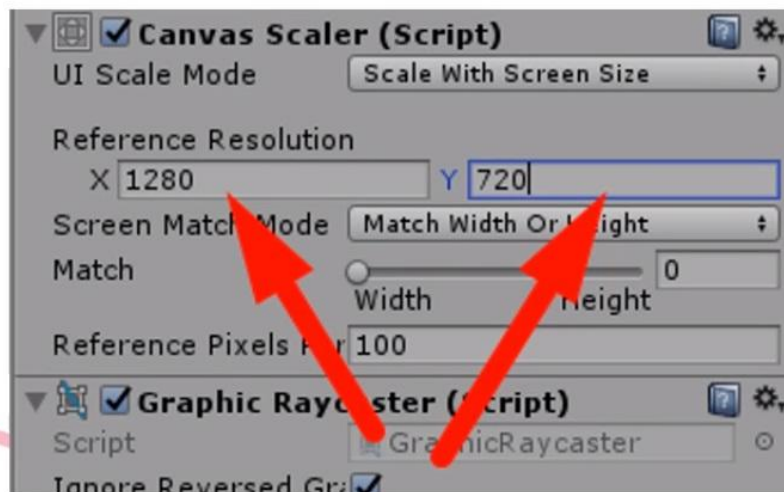
# Настройка текста



# Настройка текста

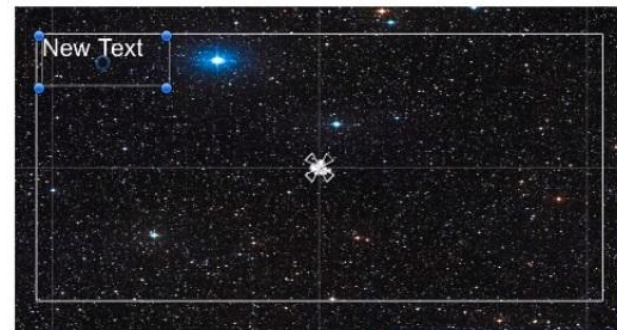
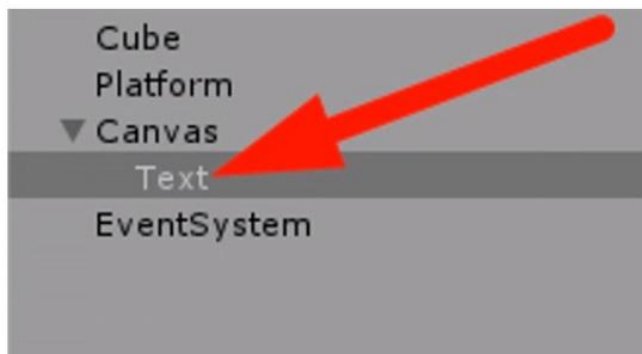


## Настройка текста

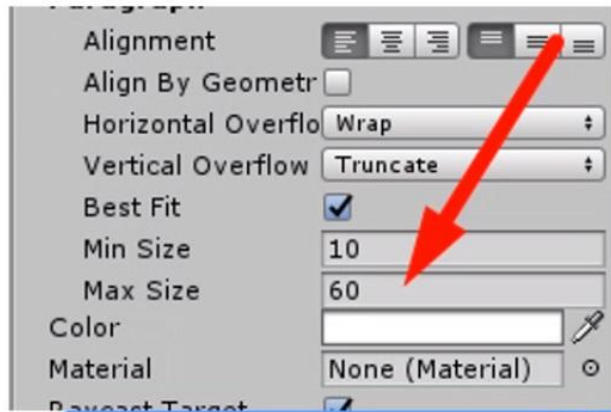
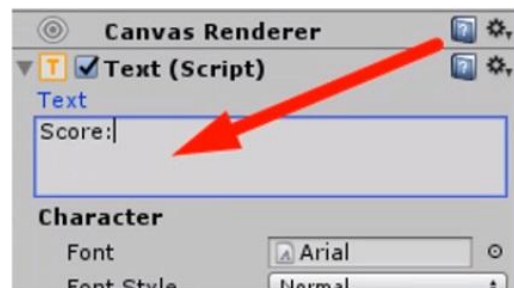
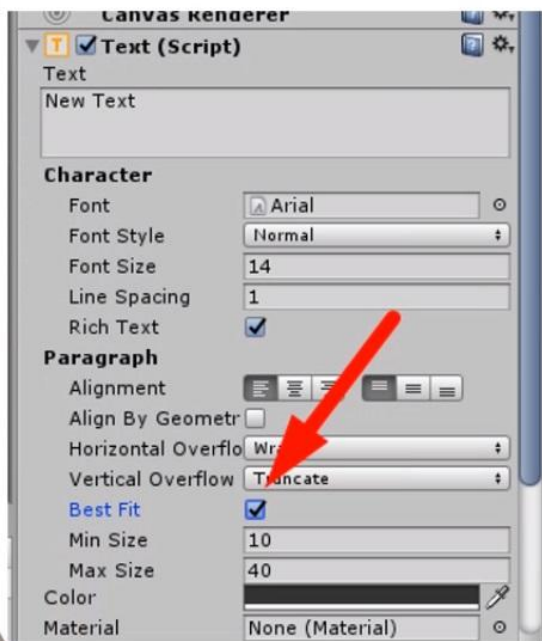




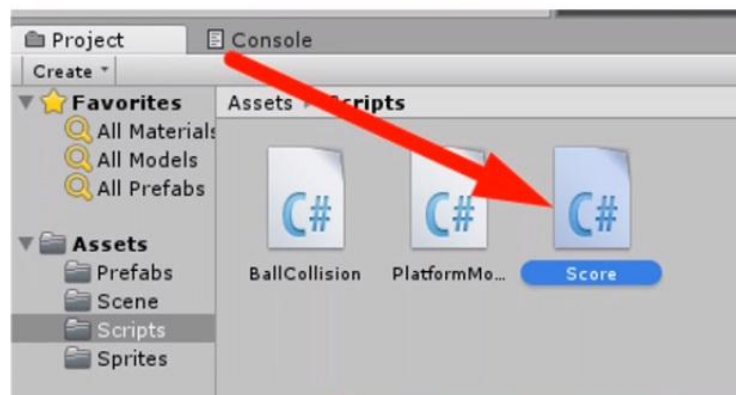
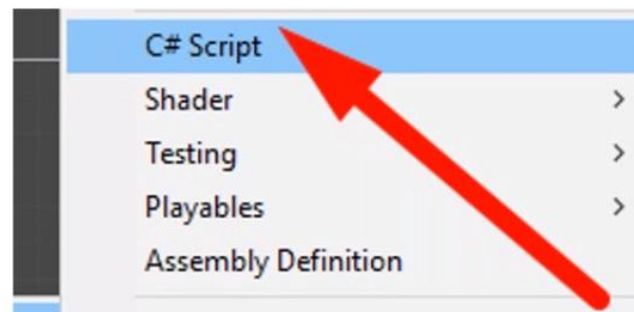
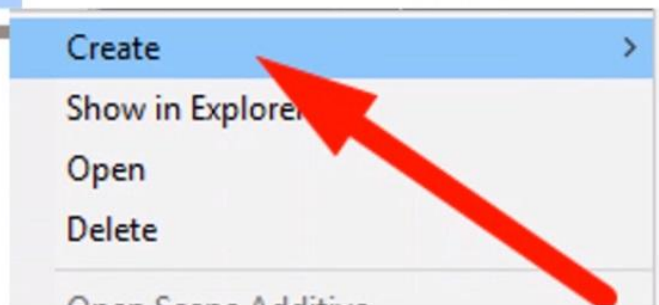
# Местоположение



## Настройка размера



## Создание скрипта



## Пишем код

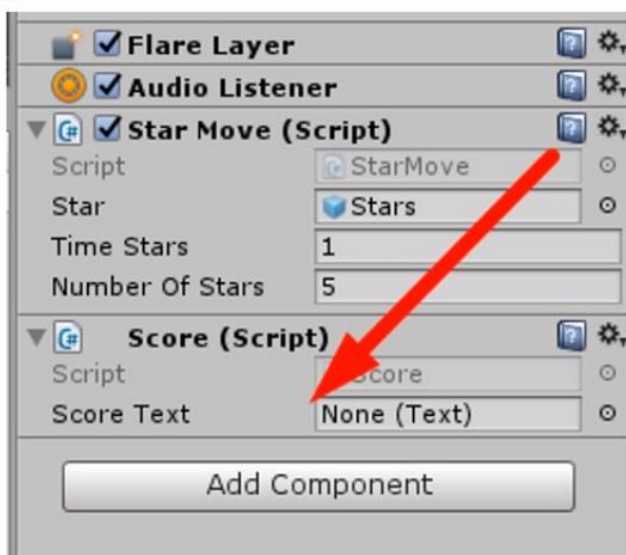
```
using UnityEngine;
using UnityEngine.UI;

public class Score : MonoBehaviour {

    [SerializeField] private Text scoreText;
    private int _gameScore;

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

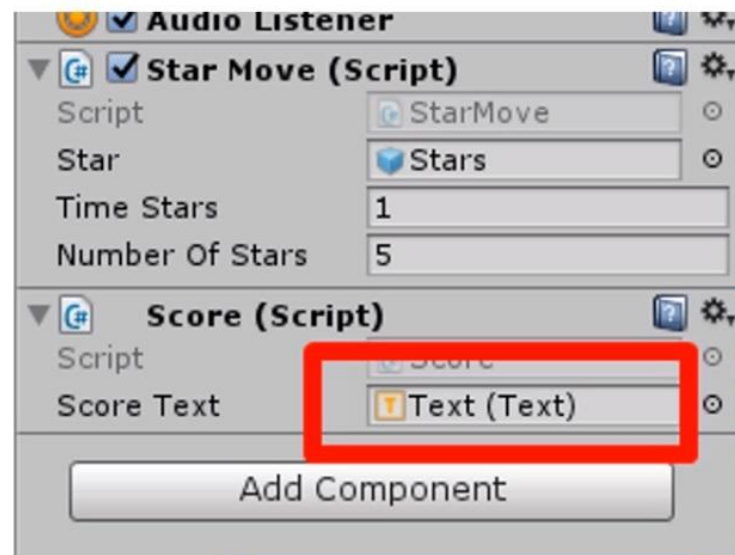
## Настраиваем скрипт



Inspector view showing the configuration for the **Star Move (Script)** component. The **Score** dropdown menu is highlighted with a red arrow, indicating it is set to **None (Text)**.

Property	Value
Script	StarMove
Star	Stars
Time Stars	1
Number Of Stars	5
Score	None (Text)

Add Component



Inspector view showing the configuration for the **Score (Script)** component. The **Score Text** dropdown menu is highlighted with a red box, indicating it is set to **Text (Text)**.

Property	Value
Script	Score
Score Text	Text (Text)

Add Component


# Запуск игры



## Задание



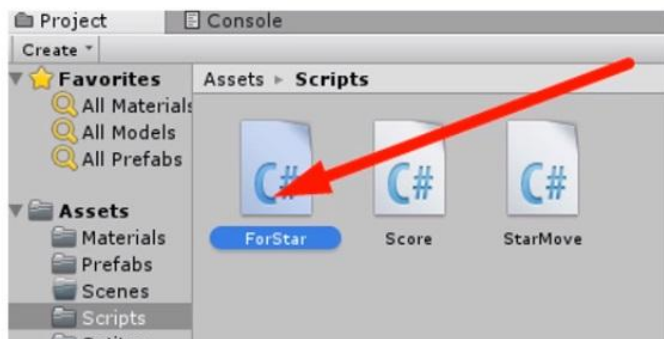
72

A screenshot of a game interface showing a score of 2. The text "Score: 2" is displayed in white on a black background with a starry space pattern. A red arrow points from the bottom right towards the number "2".

Score: 2

Добавьте в скрипт строку, которая прибавит к переменной `gameScore` +1, когда мышшь кликнет по звезде.

## Пишем код



```
public class ForStar : MonoBehaviour {  
  
    private Score score;  
  
    private Rigidbody2D rbody;  
    Vector3 start = new Vector3(0, -15f, 0);  
}
```



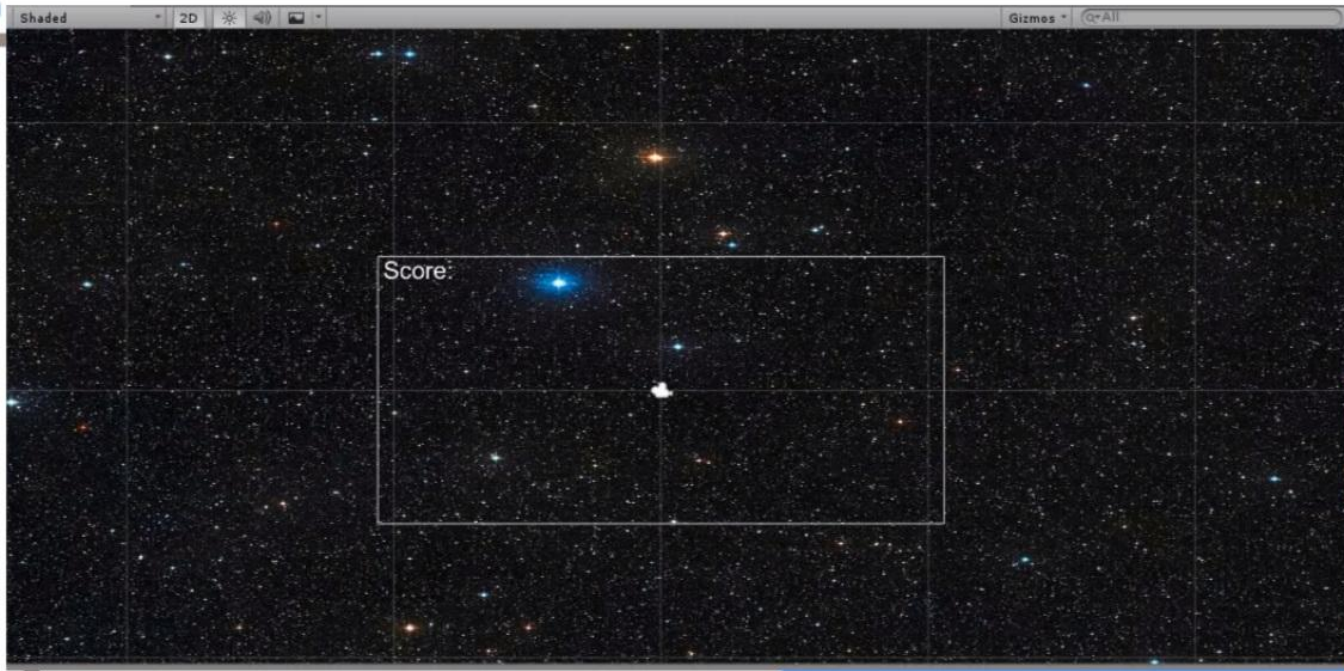
## Пишем код

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

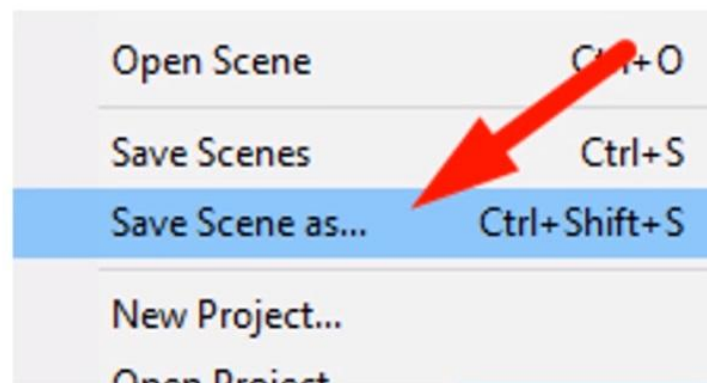
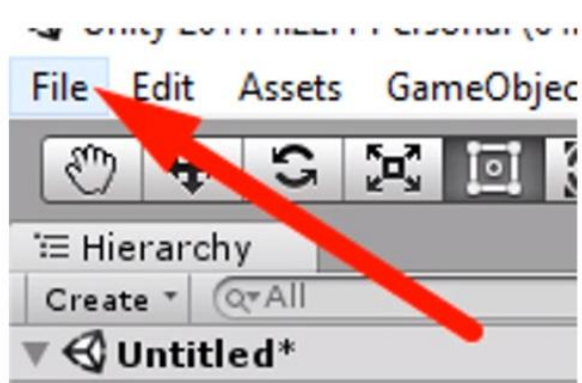
```
private void OnMouseDown()  
{  
    ...  
}
```

```
private void OnMouseDown()  
{  
    transform.position = start;  
    score.GameScore += 1;  
}
```

# Запуск игры



## Сохранение сцены



## Сохранение сцены

<< документы > Unity > Сценки > Assets

Новая папка

Имени

- Materials
- Prefabs
- Scenes
- Scripts
- Sprites

- > Объемные объ
- > Рабочий стол
- > Локальный дис

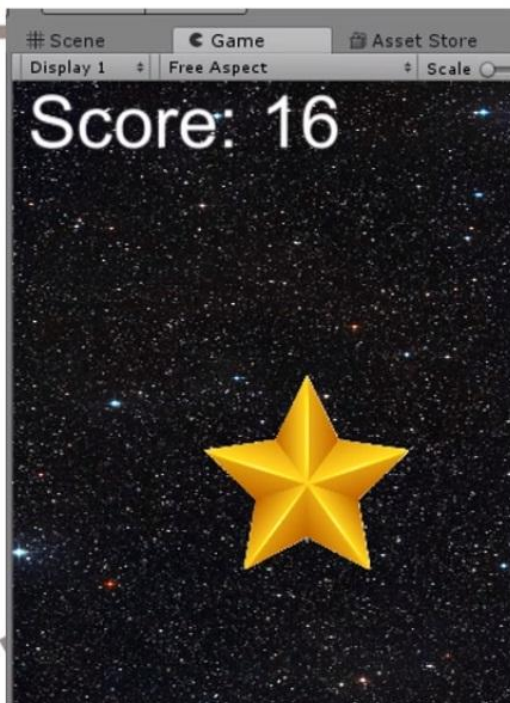
Имя файла: MainScene

Тип файла: unity

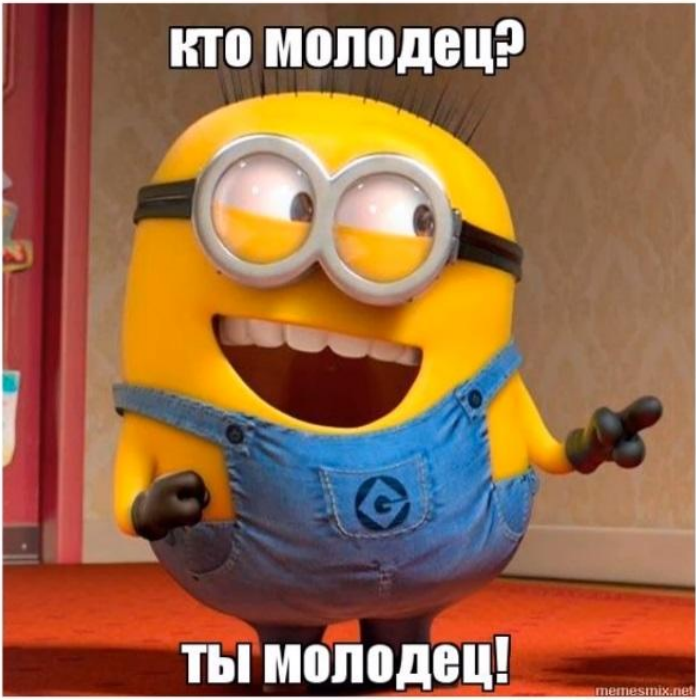
Сохранить

Отмена

## Домашнее задание



Напишите код программы, который при падении звезды (если игрок не успел по ней щелкнуть мышью) из счета вычитает минус 1. Подсказка: прежде чем улететь, звезда пролетает через триггер Ground.



Все молодцы!