

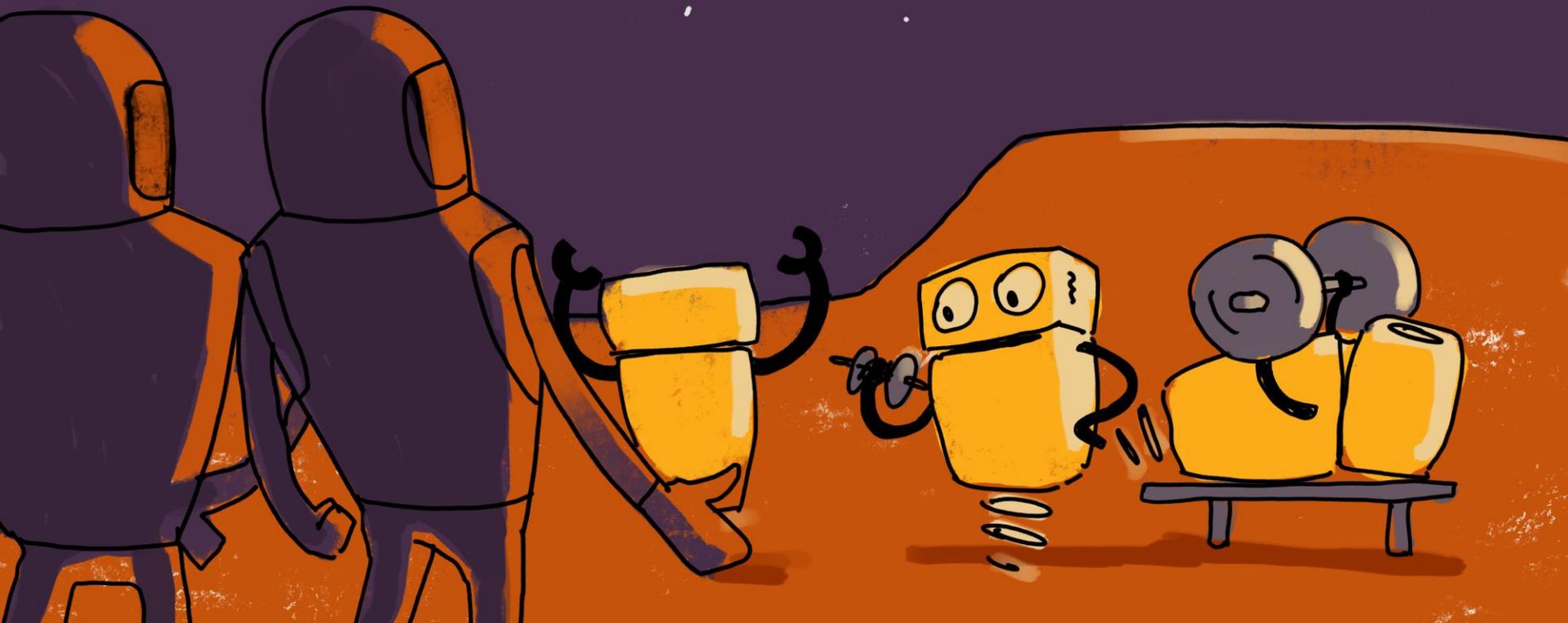
алгоритмика

Международная школа
программирования для детей

Unity 2D

Дополнения к уровням

Повтор



Сегодня на уроке...

Сегодня у нас будет множество небольших дополнений в нашу игру!

- Переход между уровнями
- Создание двери и ключа
- Дополнительно: Бонусы

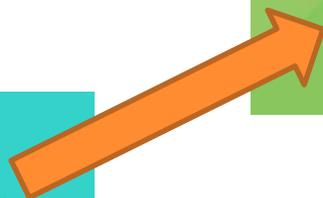
Новые
приключения
ждут!



Переход между уровнями

В конце игровых уровней всегда есть переход на следующий

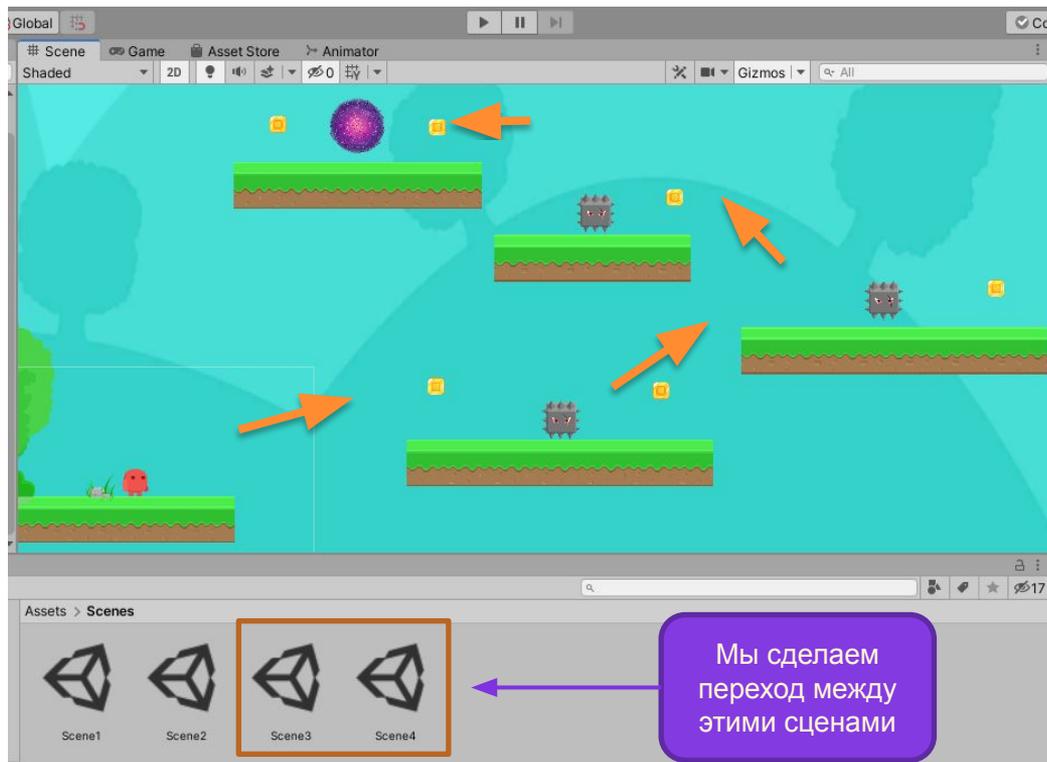
С реализации именно этой системы мы и начнём!



Переход между уровнями

Для начала выберите 2 сцены, между которыми будет переход

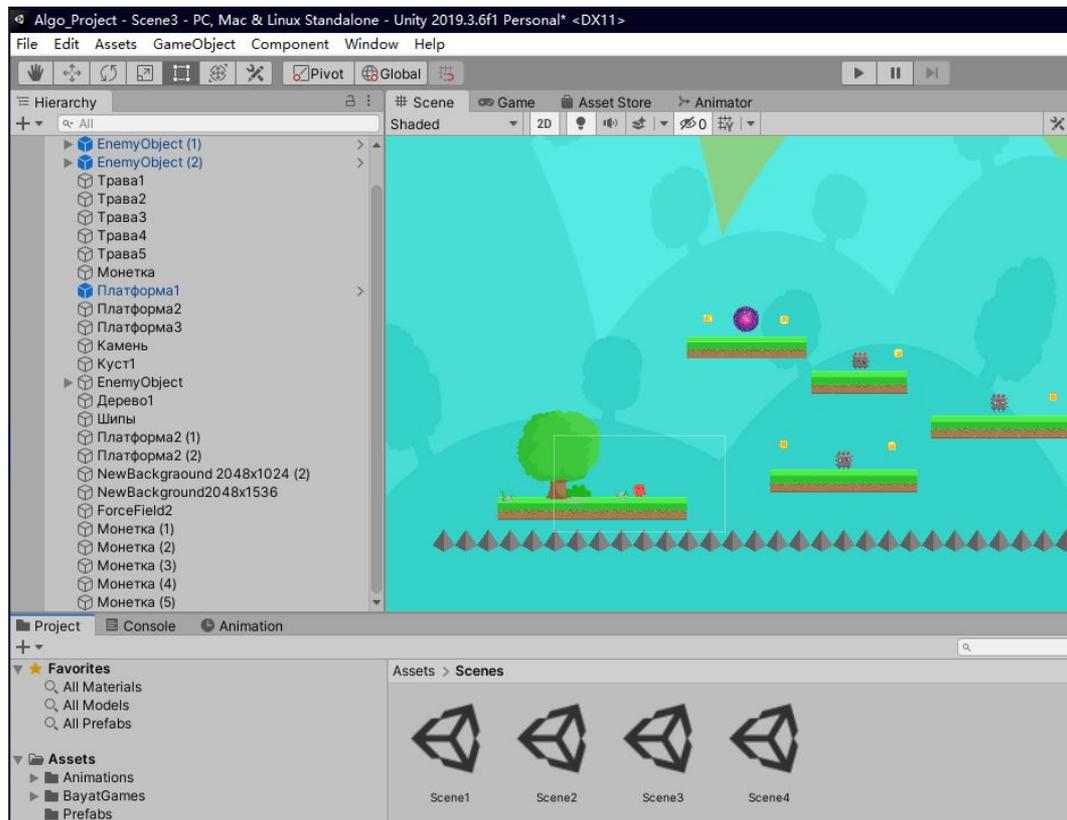
А так же разместите будущий телепорт где-нибудь на сцене



Переход между уровнями

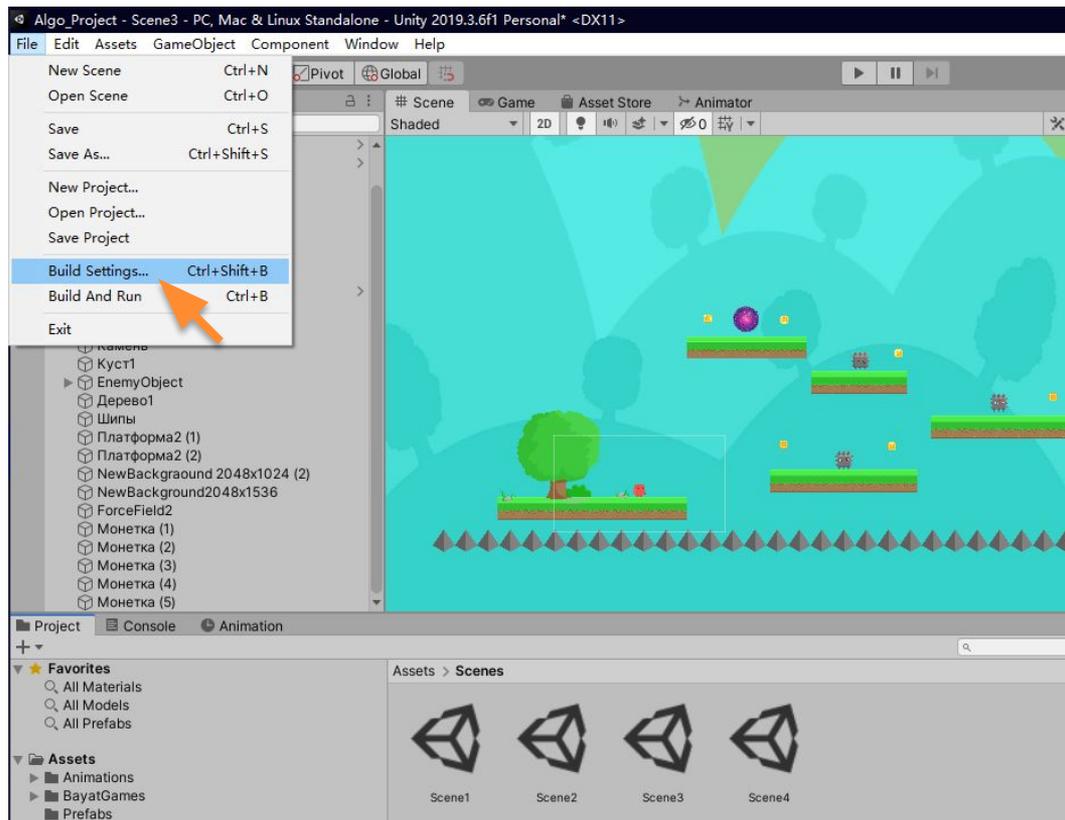
Далее, нам нужно выбрать, какие сцены будут в нашей игре

Все сцены, которые мы создаём, сами в игре не появятся. Мы добавляем их в игру вручную



Переход между уровнями

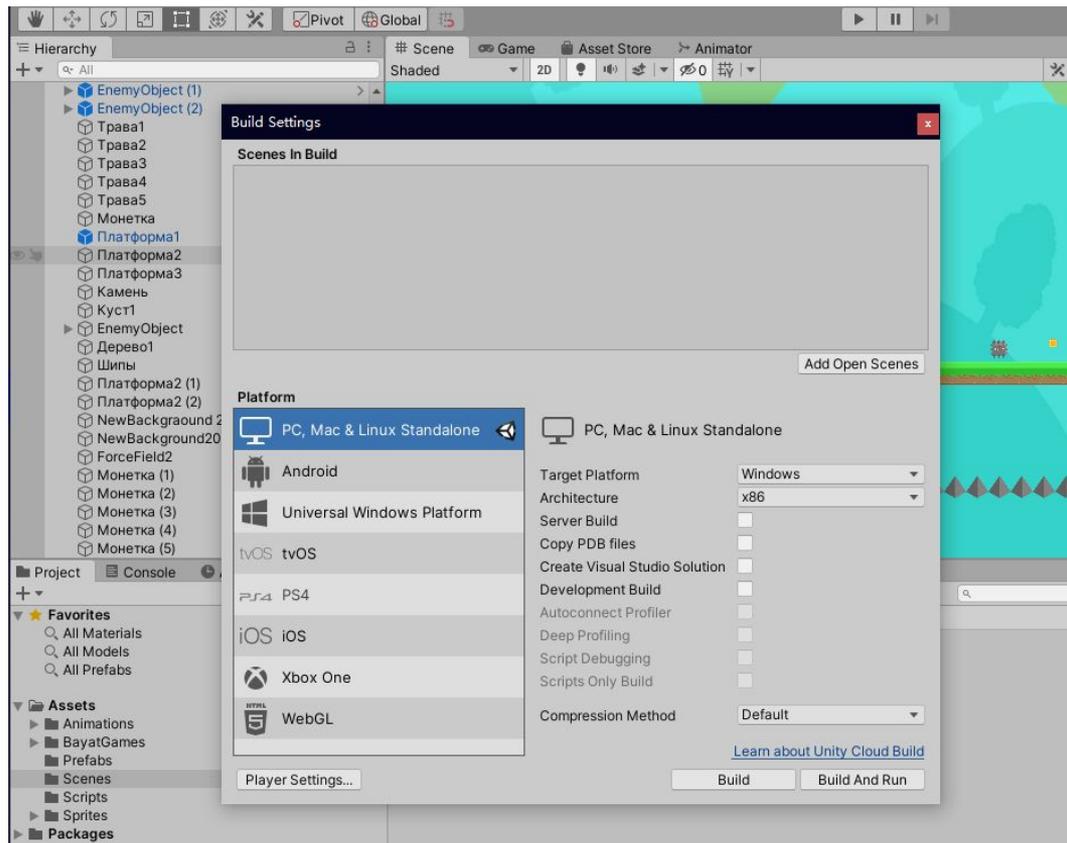
Нажмите **File** в верхнем левом углу экрана и выберите **Build Settings**



Переход между уровнями

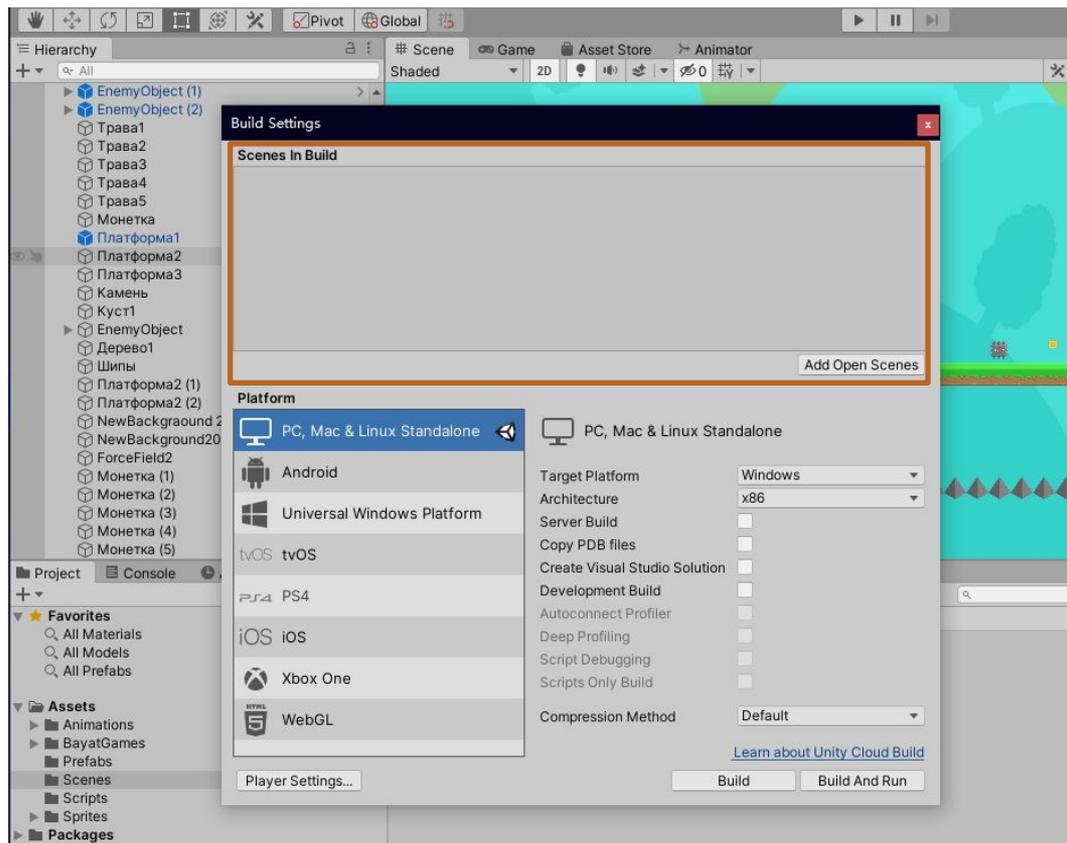
Нажмите **File** в верхнем левом углу экрана и выберите **Build Settings**

У вас откроется новое окно, которое отвечает за *платформу* для которой мы создаём игру и сцены в игре (**Scenes in build**)



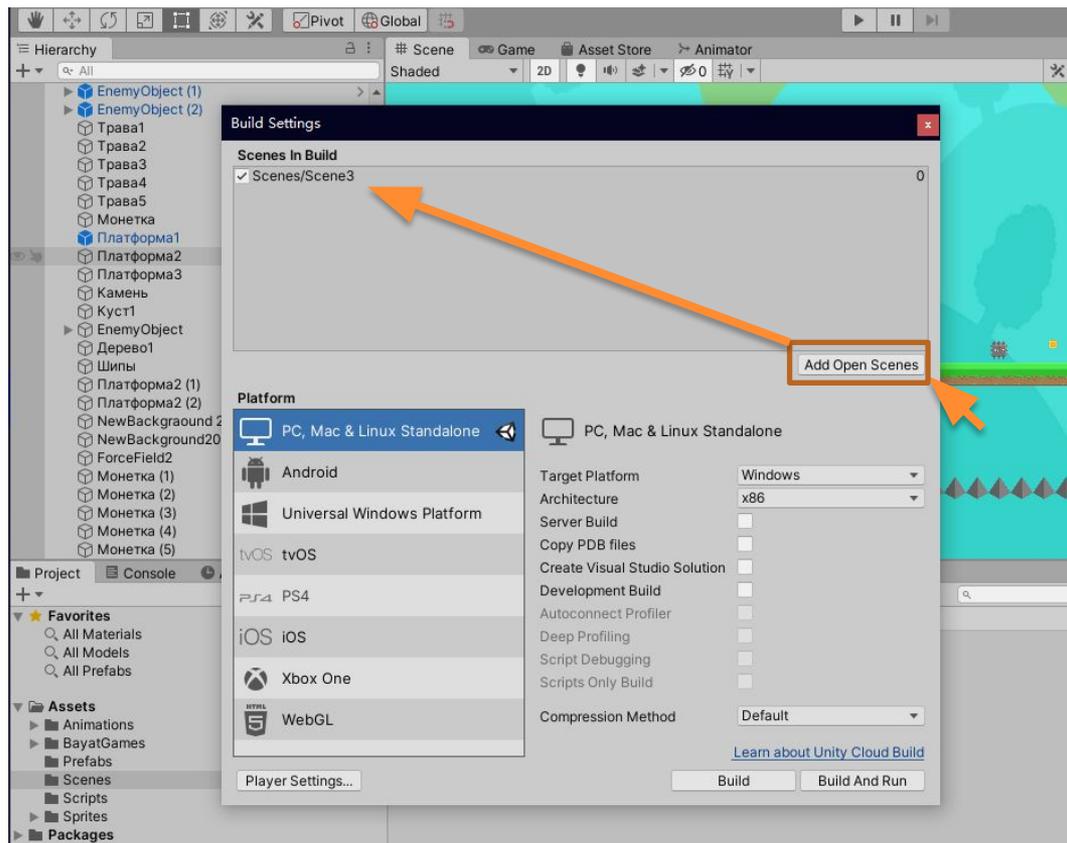
Переход между уровнями

Нас интересует поле **Scenes In Build**, сейчас в нём пусто, а это обозначает, что если мы соберем наш проект в приложение, то в нём не будет сцен и оно сразу выключится



Переход между уровнями

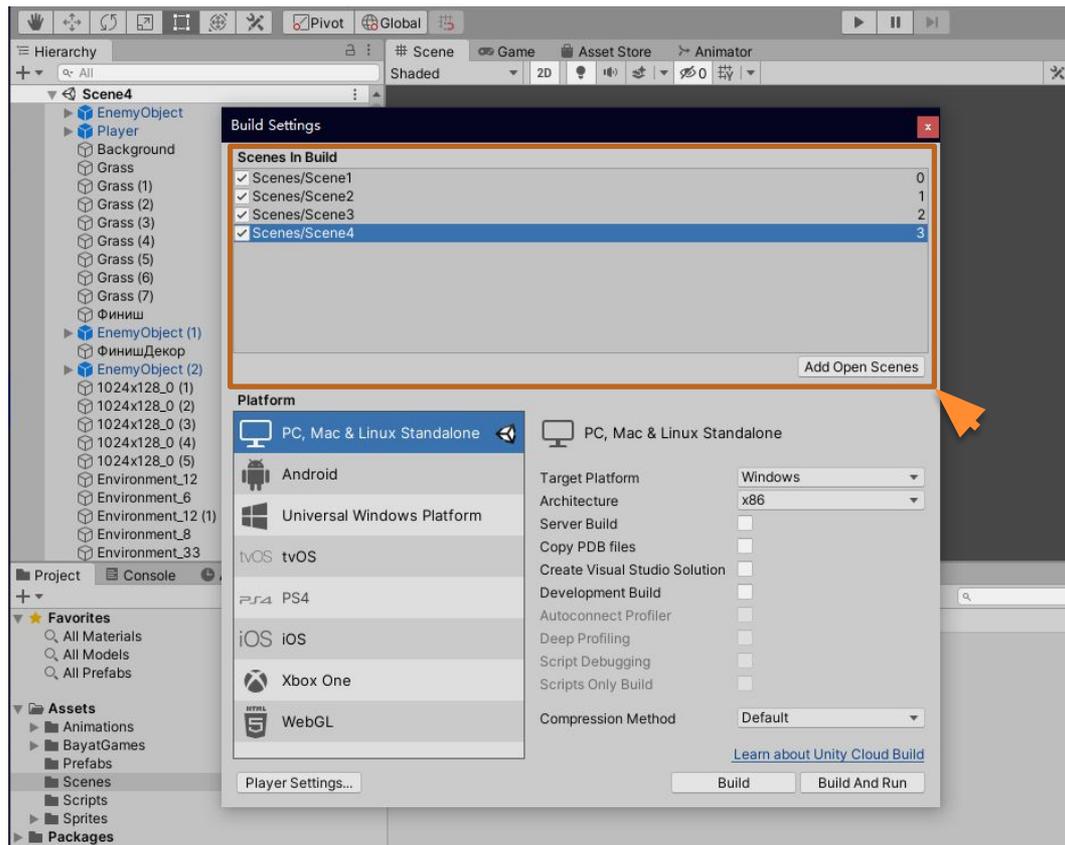
Нажав на кнопку **Add Open Scenes** вы добавите текущую открытую сцену в финальную сборку вашей игры



Переход между уровнями

Самостоятельная работа

- Добавьте все сцены, которые у вас есть в финальную сборку игры
- Отсортируйте сцены по названию (зажмите и перетащите)

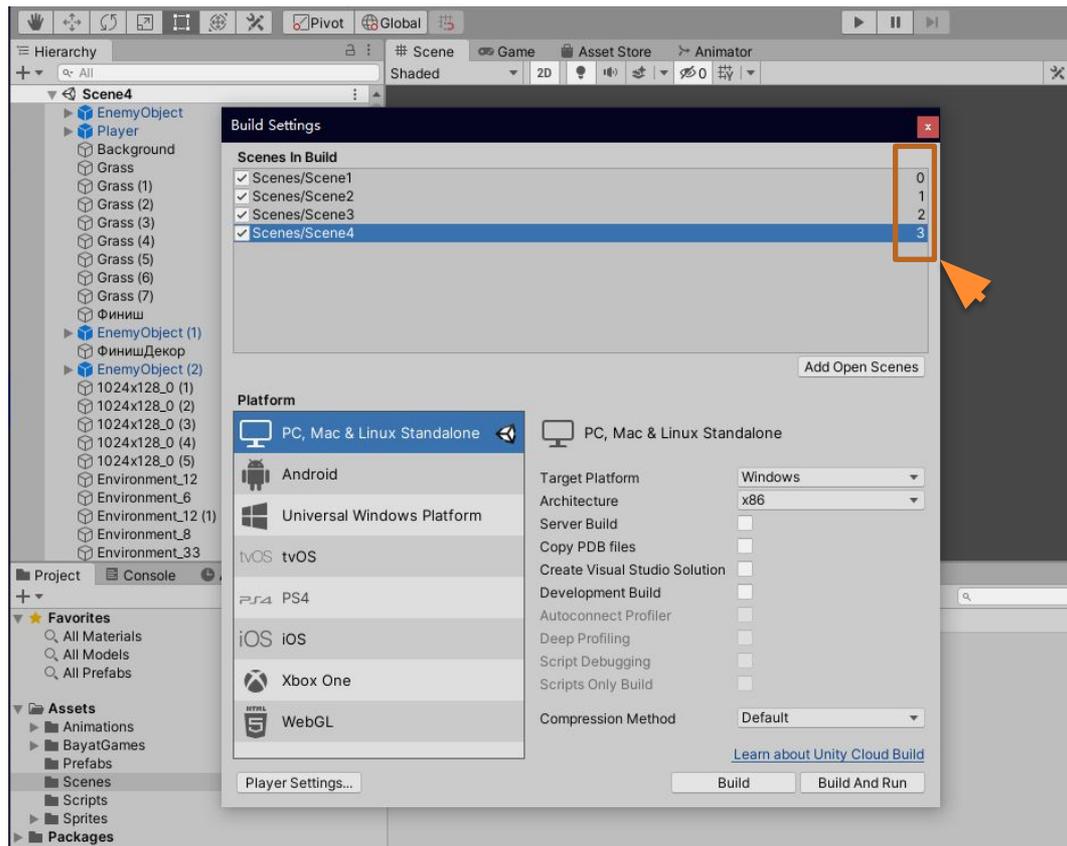


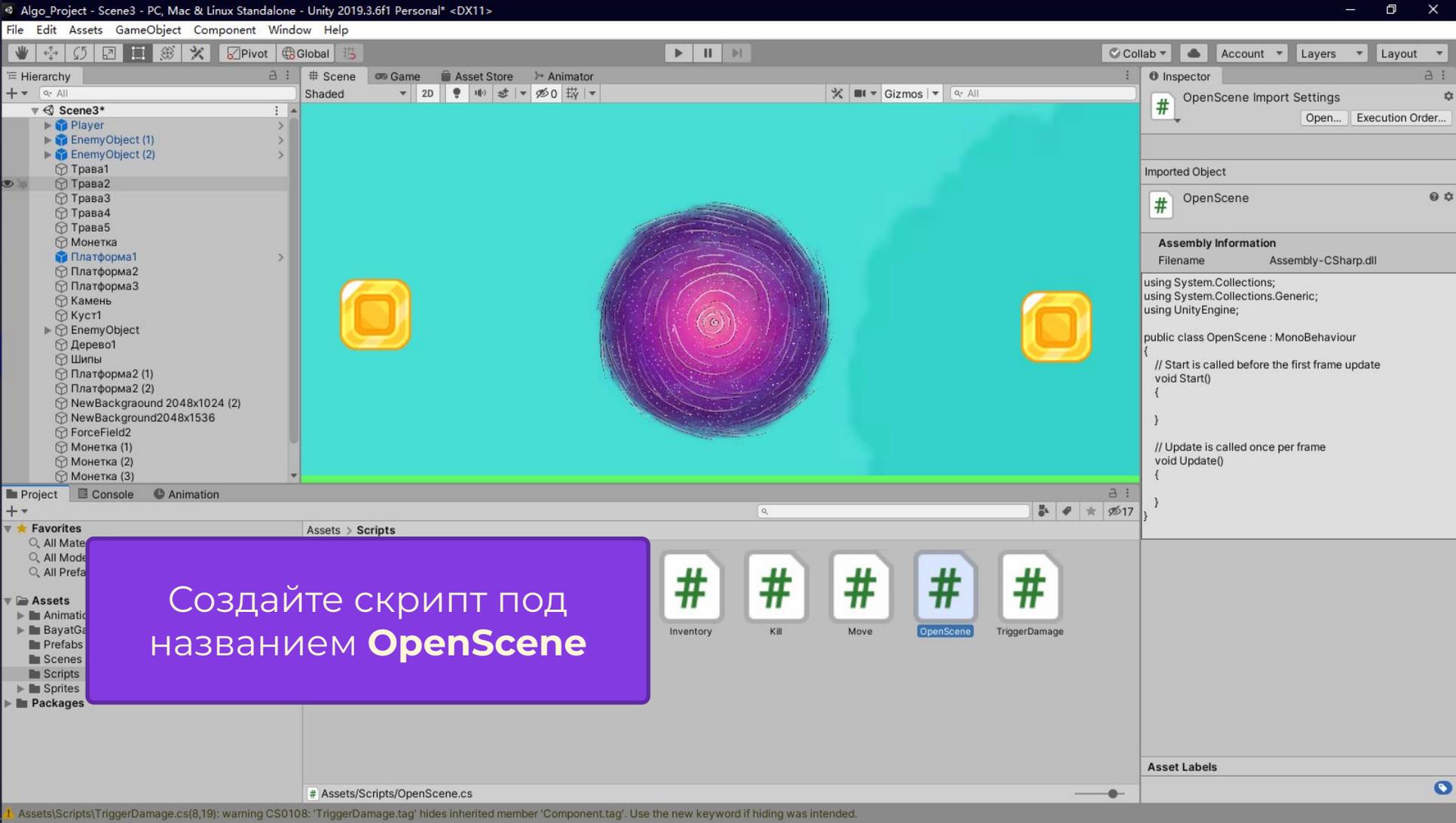
Переход между уровнями

Справа от сцен
расположены **id**(номер)
сцен.

При запуске приложения
(игры) всегда запускается
сцена с id 0

Зная **id** мы можем
отрывать их через скрипт





Создайте скрипт под названием **OpenScene**

```
OpenScene Import Settings
Open... Execution Order...

Imported Object
OpenScene

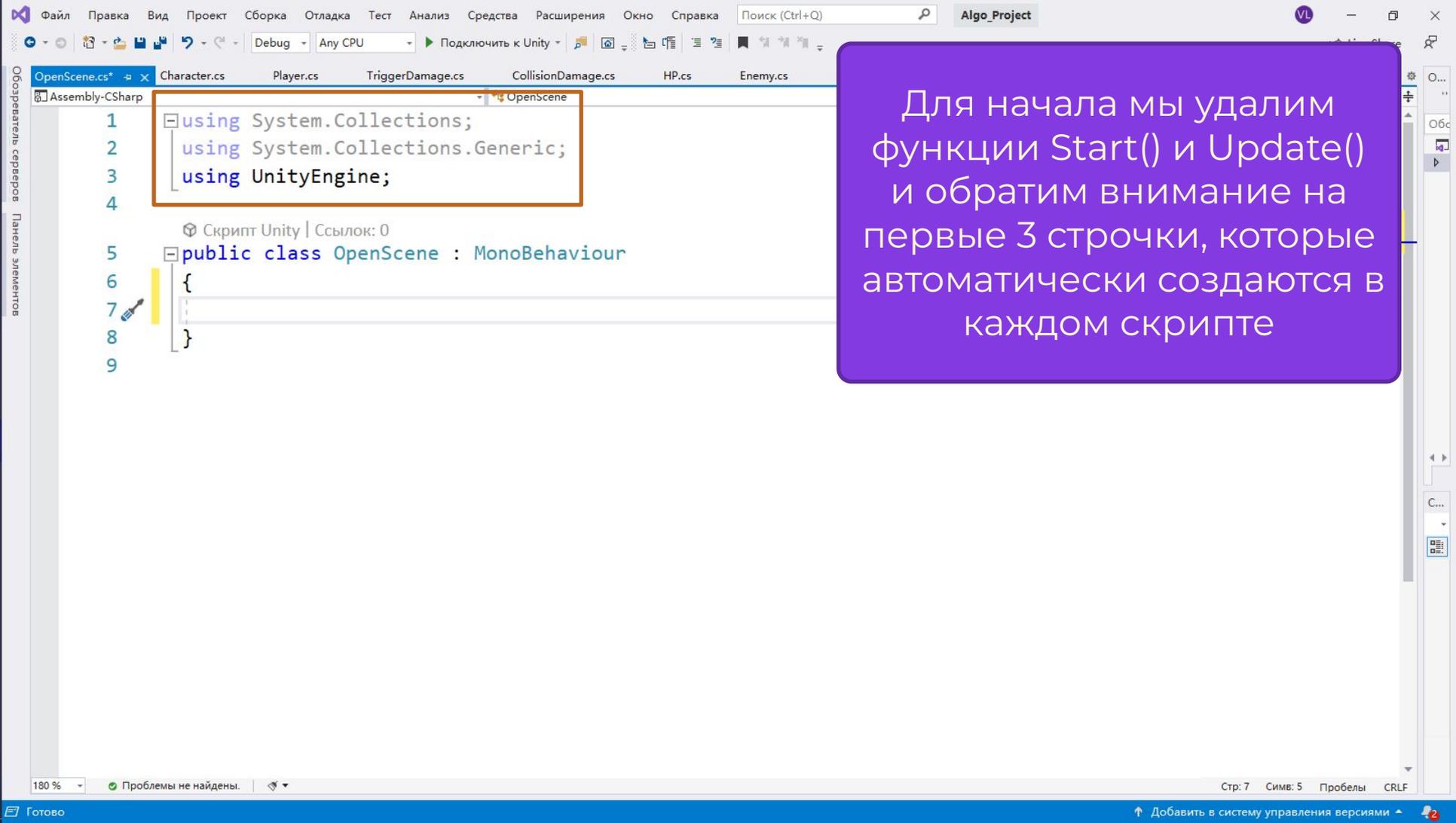
Assembly Information
Filename Assembly-CSharp.dll

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class OpenScene : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
    }
}
```





Для начала мы удалим функции Start() и Update() и обратим внимание на первые 3 строчки, которые автоматически создаются в каждом скрипте

```
File  Правка  Вид  Проект  Сборка  Отладка  Тест  Анализ  Средства  Расширения  Окно  Справка  Поиск (Ctrl+Q)
OpenScene.cs*  Character.cs  Player.cs  TriggerDamage.cs  CollisionDamage.cs  HP.cs  Enemy.cs
Assembly-CSharp  OpenScene
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  Скрипт Unity | Ссылок: 0
6  public class OpenScene : MonoBehaviour
7  {
8  }
9
using System.Collections;
using System.Collections.Generic;
//using UnityEngine;

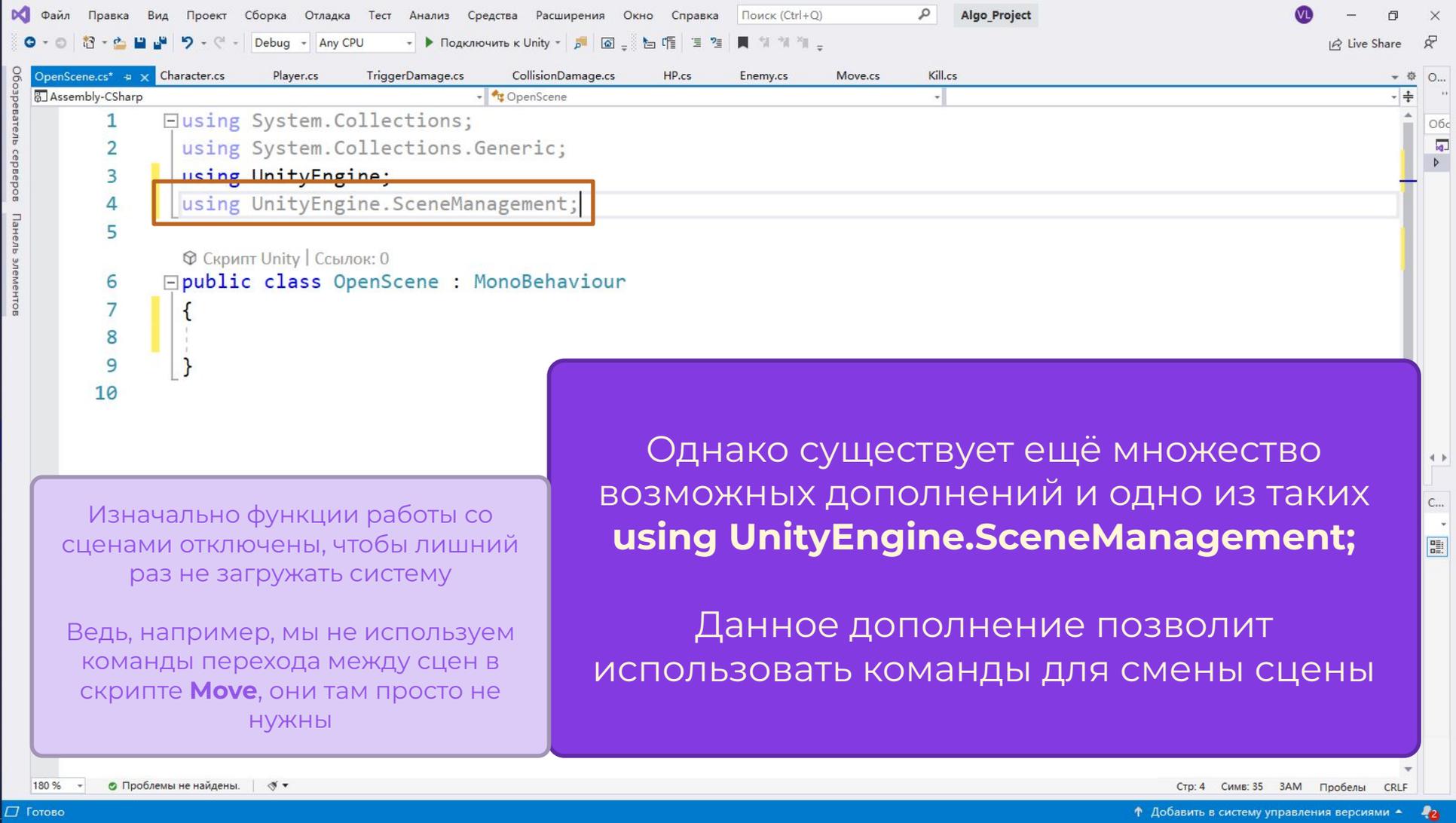
Ссылок: 0
public class OpenScene : MonoBehaviour
{
    Ссылка: 0
    private void Start()
    {
        Destroy(gameObject);
    }
}
```

using – команда, которая подключает дополнительные функции к нашему коду

Например **using UnityEngine** позволяет использовать все функции в ЮНИТИ (transform.Translate, Destroy, Start, Update...)

Программа без **using UnityEngine** выдаёт много ошибок

Обзор серверов Панель элементов



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class OpenScene : MonoBehaviour
7 {
8     ...
9 }
10
```

Изначально функции работы со сценами отключены, чтобы лишний раз не загружать систему

Ведь, например, мы не используем команды перехода между сцен в скрипте **Move**, они там просто не нужны

Однако существует ещё множество возможных дополнений и одно из таких **using UnityEngine.SceneManagement;**

Данное дополнение позволит использовать команды для смены сцены

Часть кода для нас уже
знакомая

При касании с объектом,
проверяем его тег, и если
тег «Player», то мы
загружаем нужную нам
сцену

В этой переменной
мы будем
указывать id сцены,
которую нужно
загрузить

```
public int sceneId;
```

Загружаем сцену
по id, который мы
указали в
переменной
sceneId

```
SceneManager.LoadScene(sceneId);
```

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class OpenScene : MonoBehaviour
7 {
8     public int sceneId;
9
10 private void OnCollisionEnter2D(Collision2D collision)
11 {
12     if (collision.gameObject.CompareTag("Player"))
13     {
14         SceneManager.LoadScene(sceneId);
15     }
16 }
17
18 }
19
```

После, добавляем на наш телепорт написанный скрипт, а так же Collider

В нашем компоненте OpenScene устанавливаем **id** нужной сцены, вспоминая **Build Settings**

Scenes In Build	id
<input checked="" type="checkbox"/> Scenes/Scene1	0
<input checked="" type="checkbox"/> Scenes/Scene2	1
<input checked="" type="checkbox"/> Scenes/Scene3	2
<input checked="" type="checkbox"/> Scenes/Scene4	3

Мы переходим из сцены Scene3 в Scene4, поэтому указываем **id Scene4 = 3**

Open Scene (Script)

- Script:
- Scene Id:

Circle Collider 2D

- Edit Collider:
- Material: None (Physics Material 2D)
- Is Trigger:
- Used By Effector:
- Offset: X 0 Y 0
- Radius: 5.12

Info

- Attached Body: None (Rigidbody 2D)
- Friction: 0.4
- Bounciness: 0
- Shape Count: 1
- Bounds: Center X 9.97 Y 10.46 Z 0, Extent X 0.5084 Y 0.5084 Z 0

Contacts

- Sprites-Default
- Shader: Sprites/Default

Переход между уровнями

Сделали всё правильно?

Тогда теперь ваш персонаж может переходить на следующие уровни!

Самостоятельная работа:

- Сделайте порталы на каждом игровом уровне
- Доп.: создайте анимацию для портала

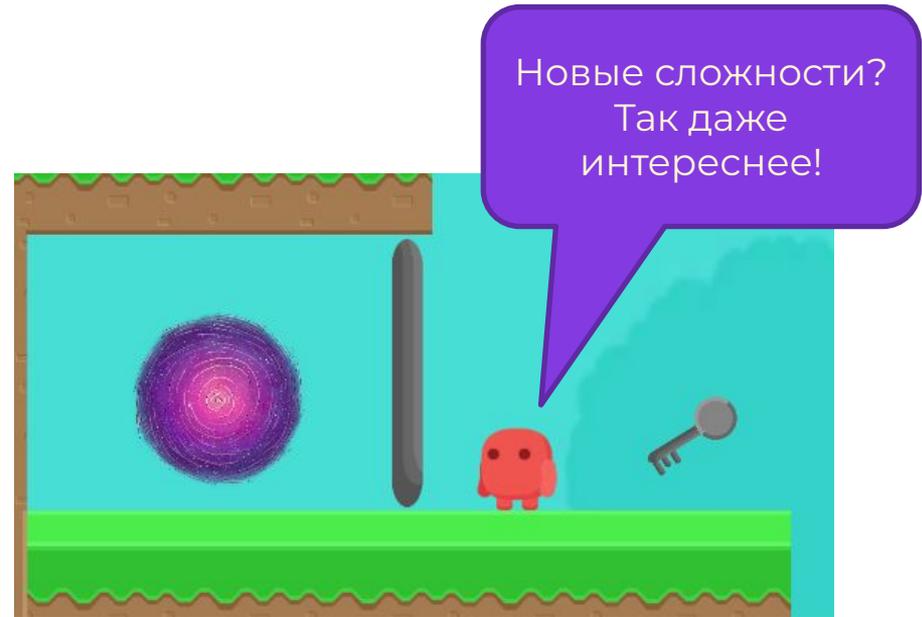


Дверь и ключ

Каждый ваш новый уровень должен быть сложнее предыдущего

Однако помимо сложности, вы должны ещё и разнообразить игру, добавляя разные интересные механики и головоломки

Одна из таких механик – это система дверей, которые открываются с помощью ключа

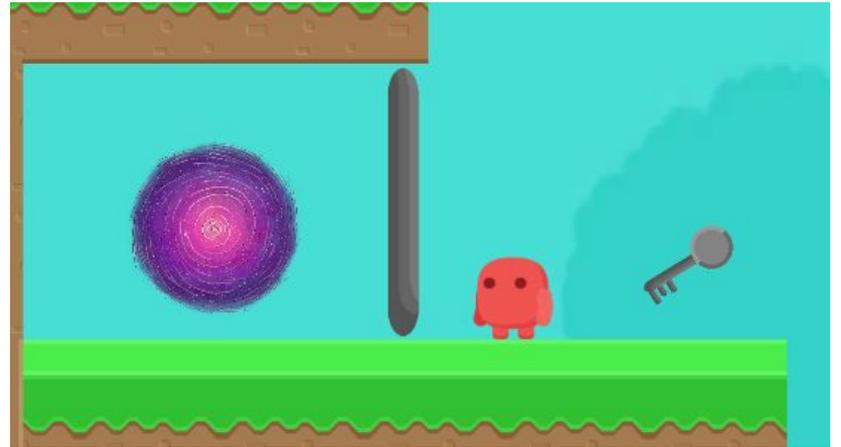


Новые сложности?
Так даже
интереснее!

Дверь и ключ

Существует множество способов реализовать данную механику:

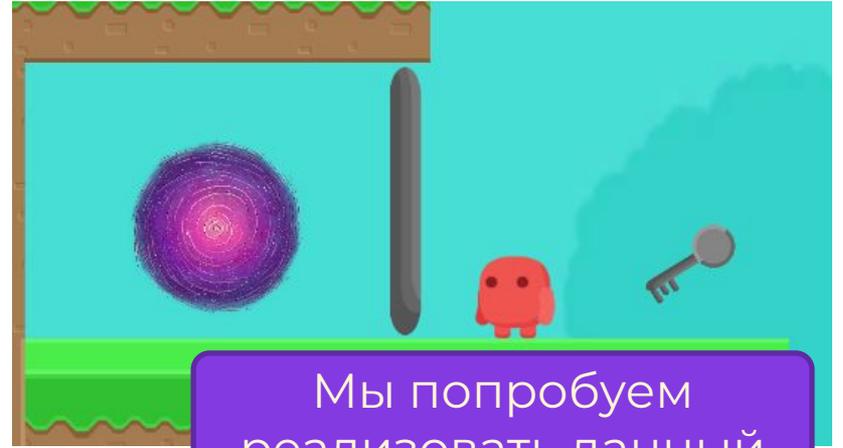
- Каждый ключ открывает определённую дверь
- Один ключ открывает только одну дверь, после использования пропадает
- Один ключ открывает все двери



Дверь и ключ

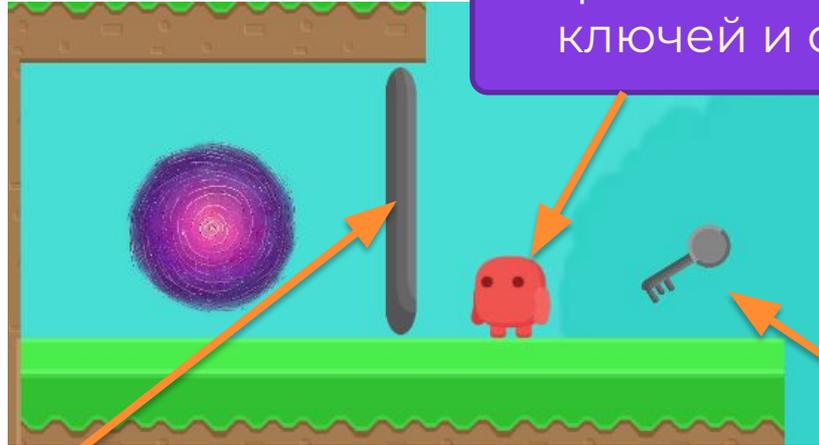
Существует множество способов реализовать данную механику:

- Каждый ключ открывает определённую дверь
- Один ключ открывает только одну дверь, после использования пропадает
- Один ключ открывает все двери
- И т.д.



Мы попробуем реализовать данный способ, так как такие двери можно будет использовать очень интересным способом

Дверь и ключ



Для персонажа мы создадим скрипт инвентаря, в котором будем хранить подобранные ключи, а так же в нём пропишем логику поднятия ключей и открытия дверей

Для двери мы создадим отдельный тег "Door"

Для ключей мы также создадим отдельный тег «Key»

Создайте новый скрипт и назовите его **Inventory**

Если у вас, так же как и у нас этот скрипт уже создан, то просто откройте его

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Inventory : MonoBehaviour
6 {
7
8
9     Сообщение Unity | Ссылки: 0
10    private void OnCollisionEnter2D(Collision2D collision)
11    {
12        if (collision.gameObject.CompareTag("Coin"))
13        {
14            Destroy(collision.gameObject);
15        }
16    }
17
```

У нас тут уже реализована система подбора монеток, не обращайте на неё внимания

Для начала создадим переменную **keys**, которая будет сохранять кол-во ключей, которые есть у нашего персонажа

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Inventory : MonoBehaviour
6 {
7     public int keys = 0;
8
9     private void OnCollisionEnter2D(Collision2D collision)
10    {
11        if (collision.gameObject.CompareTag("Coin"))
12        {
13            Destroy(collision.gameObject);
14        }
15    }
16 }
17
```

```
Inventory.cs* x OpenScene.cs Character.cs Player.cs TriggerDamage.cs CollisionDamage.cs HP.cs Enemy.c
Assembly-CSharp - Inventory
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Inventory : MonoBehaviour
6 {
7     public int keys = 0;
8
9     private void OnCollisionEnter2D(Collision2D collision)
10    {
11        if (collision.gameObject.CompareTag("Coin"))
12        {
13            Destroy(collision.gameObject);
14        }
15    }
16 }
17
```

Далее создайте функцию **OnCollisionEnter2D**, для отслеживания коллизии (касания) с ключами и дверьми

Если она у вас уже есть, то пропустите этот шаг, одной функции достаточно

```
Файл Правка Вид Проект Отладка Тест Анализ Средства Расширения Окно Справка Поиск (Ctrl+Q)
Debug Any CPU Подключить к Unity
Inventory.cs* OpenScene.cs Character.cs Player.cs TriggerDamage.cs CollisionDamage.cs HP.cs
Assembly-CSharp Inventory
4
5 public class Inventory : MonoBehaviour
6 {
7     public int keys = 0;
8
9     Сообщение Unity | Ссылка: 0
10 private void OnCollisionEnter2D(Collision2D collision)
11 {
12     if (collision.gameObject.CompareTag("Key"))
13     {
14     }
15     if (collision.gameObject.CompareTag("Door"))
16     {
17     }
18 }
19     if (collision.gameObject.CompareTag("Coin"))
20     {
21         Destroy(collision.gameObject);
22     }
23 }
24 }
25
```

Затем, создаём 2 проверки, на касание объектов с тегом "Key" и с тегом "Door"

```
Файл Правка Вид Проект Отладка Тест Анализ Средства Расширения Окно Справка Поиск (Ctrl+Q)
Debug Any CPU Подключить к Unity
Inventory.cs* OpenScene.cs Character.cs Player.cs TriggerDamage.cs CollisionDamage.cs HP.cs
Assembly-CSharp
Скрипт Unity | Ссылка: 0
5 public class Inventory : MonoBehaviour
6 {
7     public int keys = 0;
8
9     Сообщение Unity | Ссылка: 0
10    private void OnCollisionEnter2D(Collision2D collision)
11    {
12        if (collision.gameObject.CompareTag("Key"))
13        {
14            keys += 1;
15            Destroy(collision.gameObject);
16        }
17        if (collision.gameObject.CompareTag("Door"))
18        {
19        }
20        if (collision.gameObject.CompareTag("Coin"))
21        {
22            Destroy(collision.gameObject);
23        }
24    }
25 }
26
```

При касании с ключом, мы будем прибавлять к переменной с ключами +1 и удалять его из игры

```
Файл  Правка  Вид  Проект  Сборка  Отладка  Тест  Анализ  Средства  Расширения  Окно  Справка  Поиск (Ctrl+Q)
Inventory.cs  OpenScene.cs  Character.cs  Player.cs  TriggerDamage.cs  CollisionDamage.cs  HP.cs  Enemy.cs
Assembly-CSharp
Скрипт Unity | Ссылка: 0
5  public class Inventory : MonoBehaviour
6  {
7      public int keys = 0;
8
9      Сообщение Unity | Ссылка: 0
10 private void OnCollisionEnter2D(Collision2D collision)
11 {
12     if (collision.gameObject.CompareTag("Key"))
13     {
14         keys += 1;
15         Destroy(collision.gameObject);
16     }
17     if (collision.gameObject.CompareTag("Door"))
18     {
19         if (keys > 0)
20         {
21             keys -= 1;
22             Destroy(collision.gameObject);
23         }
24     }
25     if (collision.gameObject.CompareTag("Coin"))
26     {
27         Destroy(collision.gameObject);
28     }
29 }
```

При касании с дверьми, мы сначала проверим, есть ли у нас хотя бы 1 ключ (если ключей больше 0), и если это так, то мы вычитаем 1 ключ из переменной, а также удаляем дверь, которую коснулись

Будьте внимательны!

Обращайте внимание на отступы и фигурные скобки

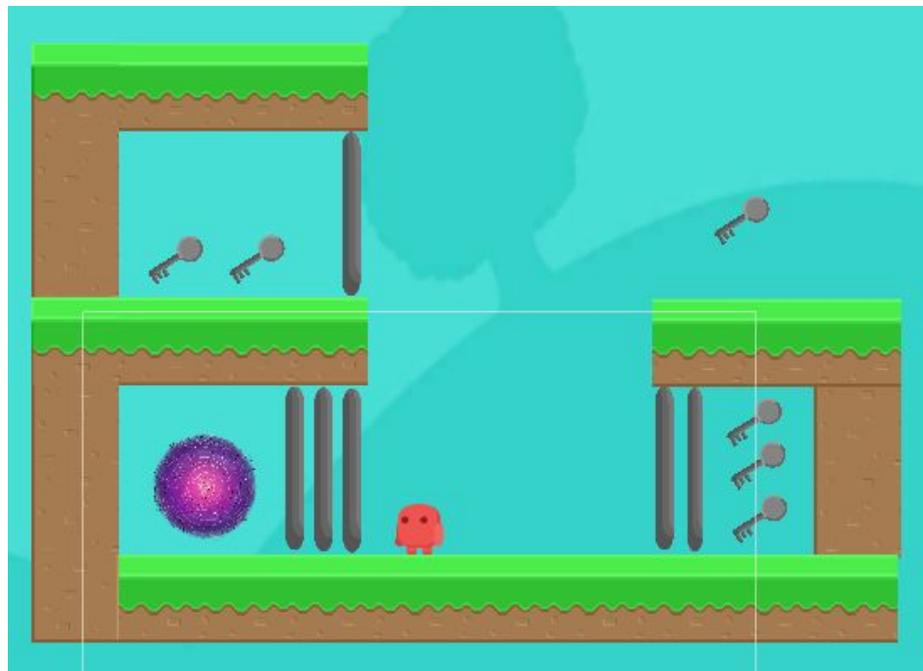
Далее останется только передать этот скрипт игроку, а так же расставить на сцене двери с ключами с соответствующими тегами

Самостоятельная работа

Добавьте на ваши уровни двери и ключи к ним, чтобы разнообразить геймплей!

Попробуйте сделать небольшую головоломку с ключами

Не забывайте разнообразить ваш уровень и врагами с монетками!



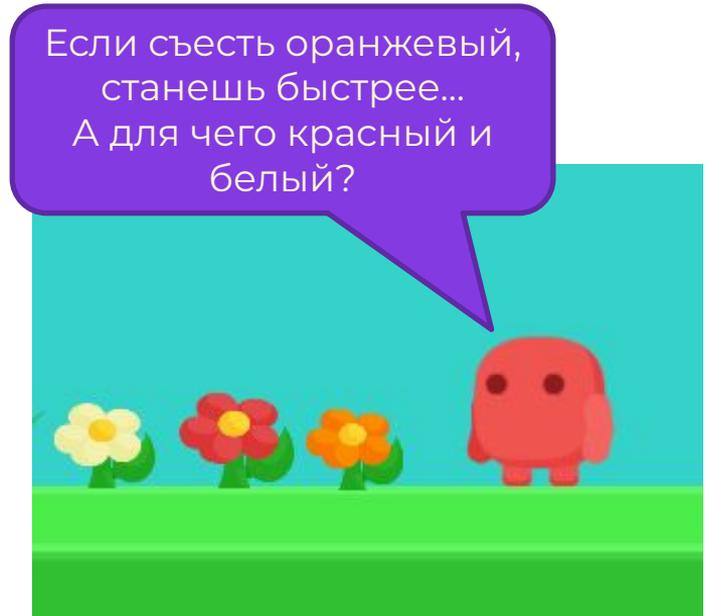
Дополнительное задание

Нужны ли бонусы в вашей игре?

Например: временное повышение скорости, или усиленный прыжок?

Если есть желание попробовать реализовать систему бонусов, то смело переходите по [ссылке](https://drive.google.com/drive/folders/1U95-oDpYpaGc0uAQKVYslcA1wLlpWt7Q?usp=sharing):

<https://drive.google.com/drive/folders/1U95-oDpYpaGc0uAQKVYslcA1wLlpWt7Q?usp=sharing>



**До встречи на
следующем уроке!**

Спасибо!

