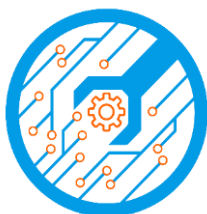
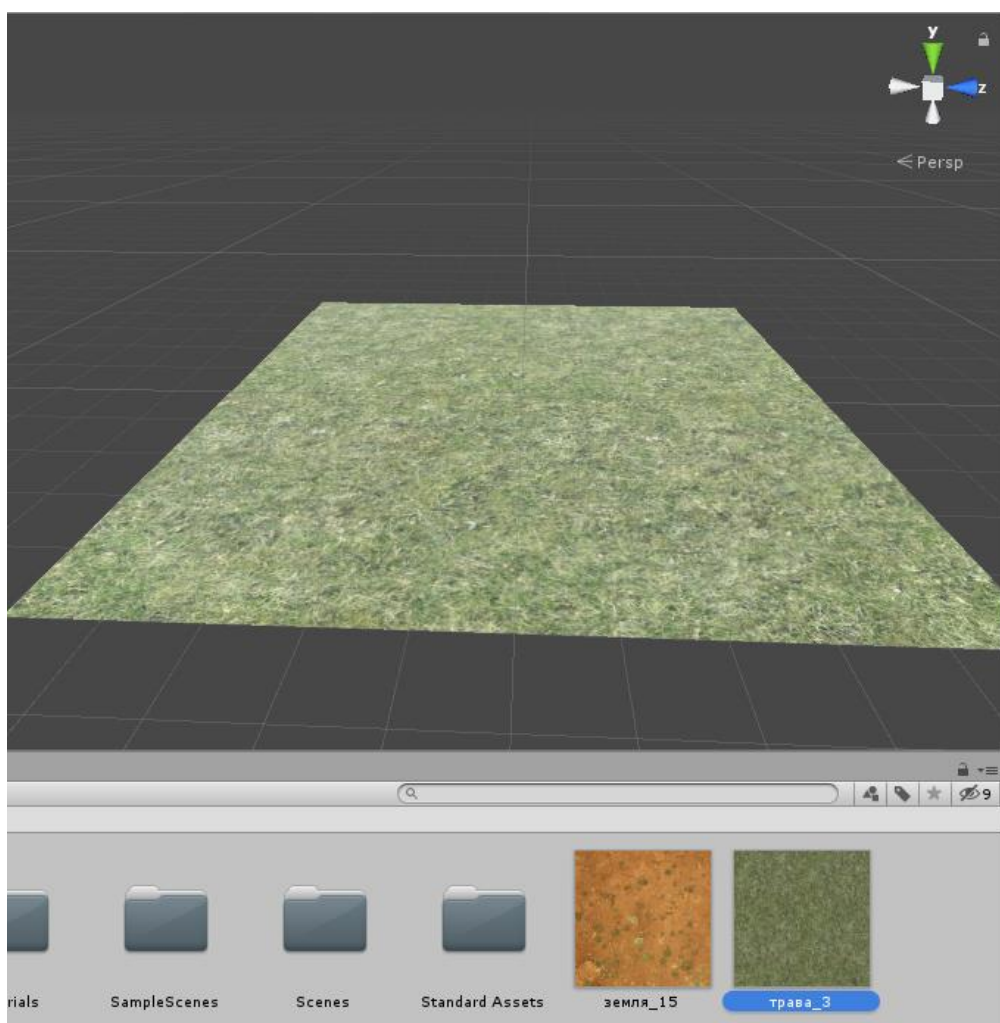


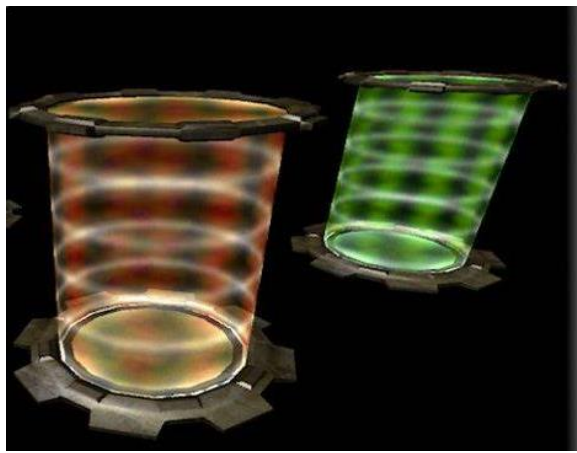
ТЕЛЕПОРТАЦИЯ

Создание сцены

для понимания принципа достаточно объектов:
Plane, Cube и префабов и Standard Assets

1. Импортируем Standart Assets;
2. Создаем площадку Plane;
3. Добавляем текстуру травы.

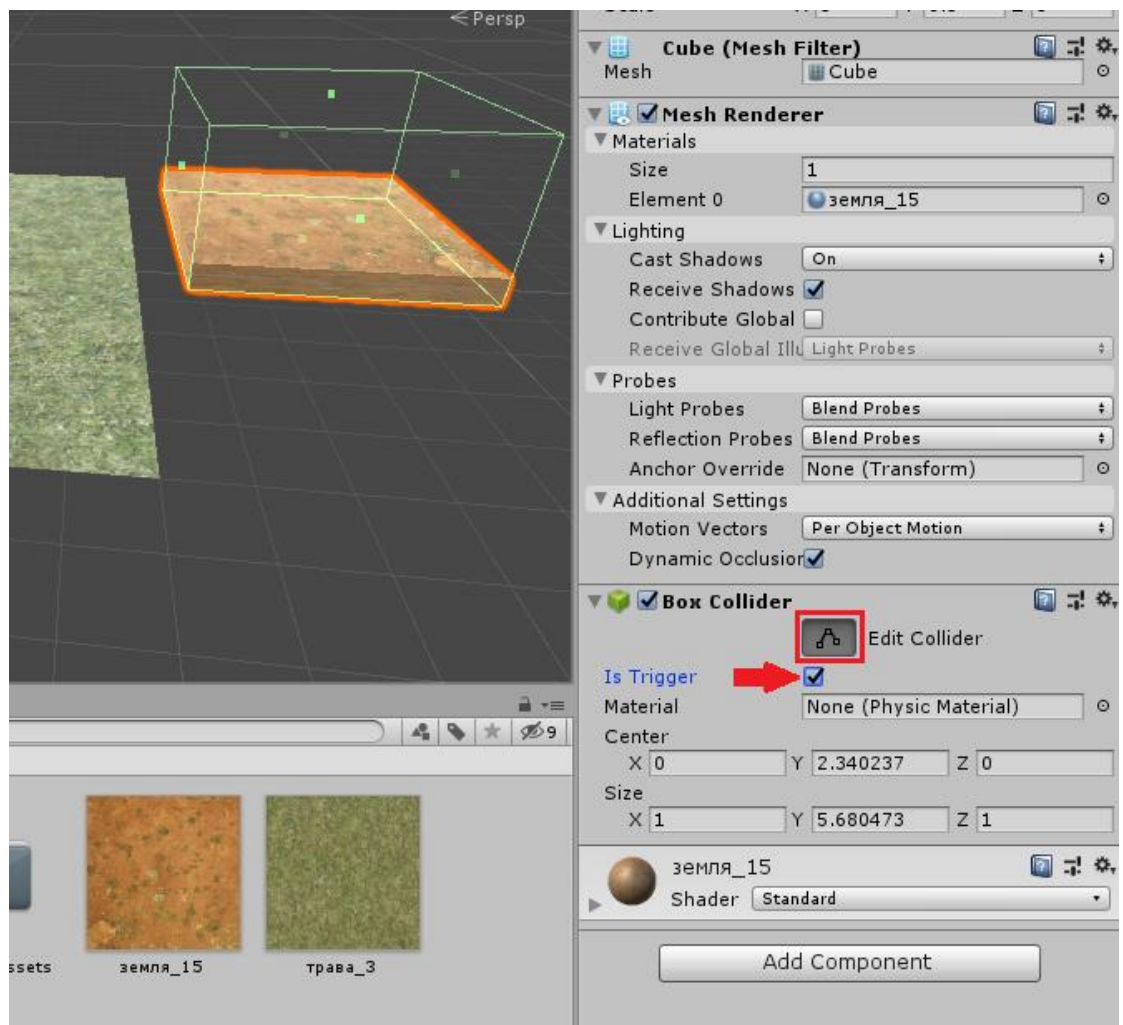


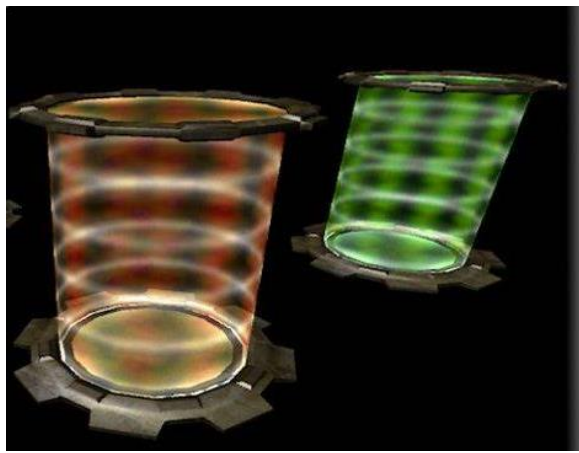


ТЕЛЕПОРТАЦИЯ

Создание сцены Платформа - телепорт

1. Импортируем Cube;
2. Придаем вид платформы;
3. Добавляем текстуру земли;
4. Расширяем Collider и делаем его Is Trigger.





ТЕЛЕПОРТАЦИЯ

Добавляем скрипт

В окне Assets нажатием правой кнопки мыши выбираем: Create → C# Script и переименовываем в Teleporter. Код скрипта (заменяем стандартный):

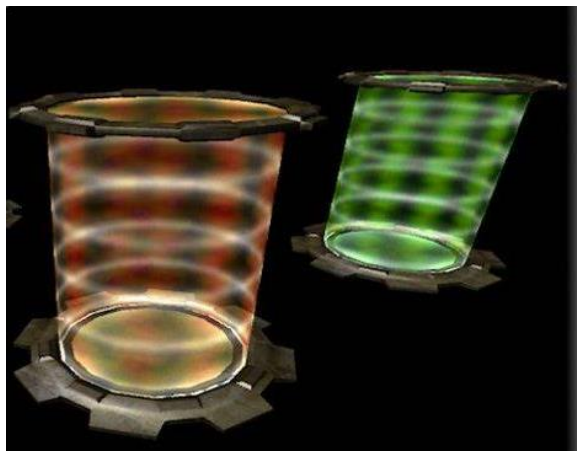
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Teleporter : MonoBehaviour
{
    // Подключаем сюда конечную точку телепорта
    public Transform Target;

    // При входе объекта в триггер (этот объект -
    игрок, то есть other)
    void OnTriggerEnter(Collider other)
    {
        // Телепортировать объект на конечную точку
        other.transform.position =
Target.transform.position;
    }
}
```



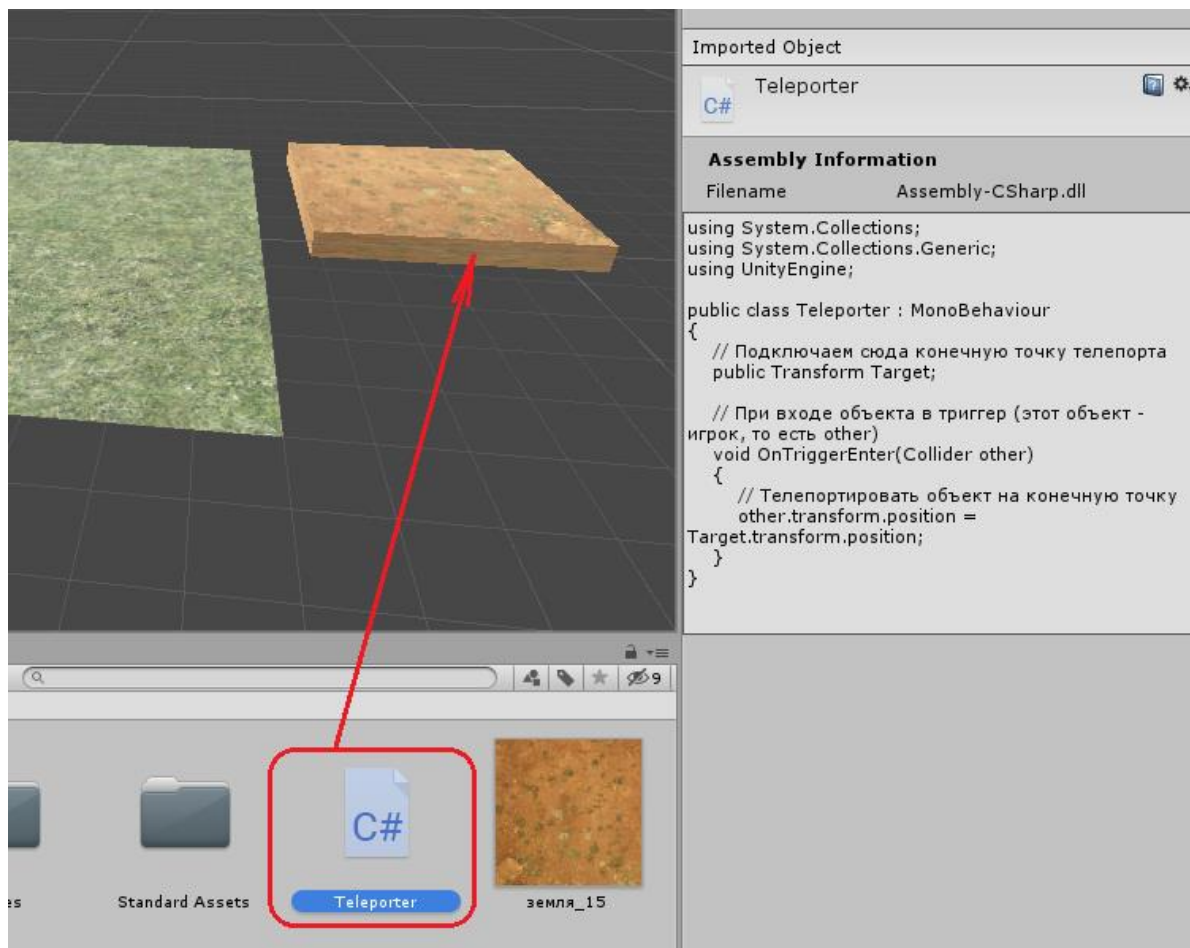
Таким образом создан класс, в котором переменная выступает точкой, куда телепортируется игрок, а функция срабатывает при взаимодействии с Collider.



ТЕЛЕПОРТАЦИЯ

Создаем префаб

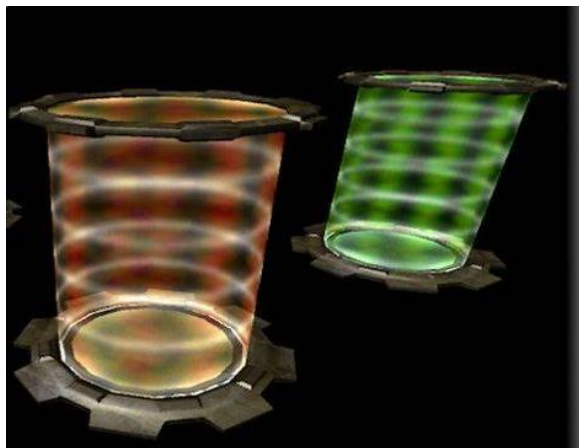
1. Перетаскиваем скрипт на Cube



2. В окне сцен создаем Game Object (Create Empty), переименовываем его в Teleport.

3. Помещаем в него наш Cube со скриптом.

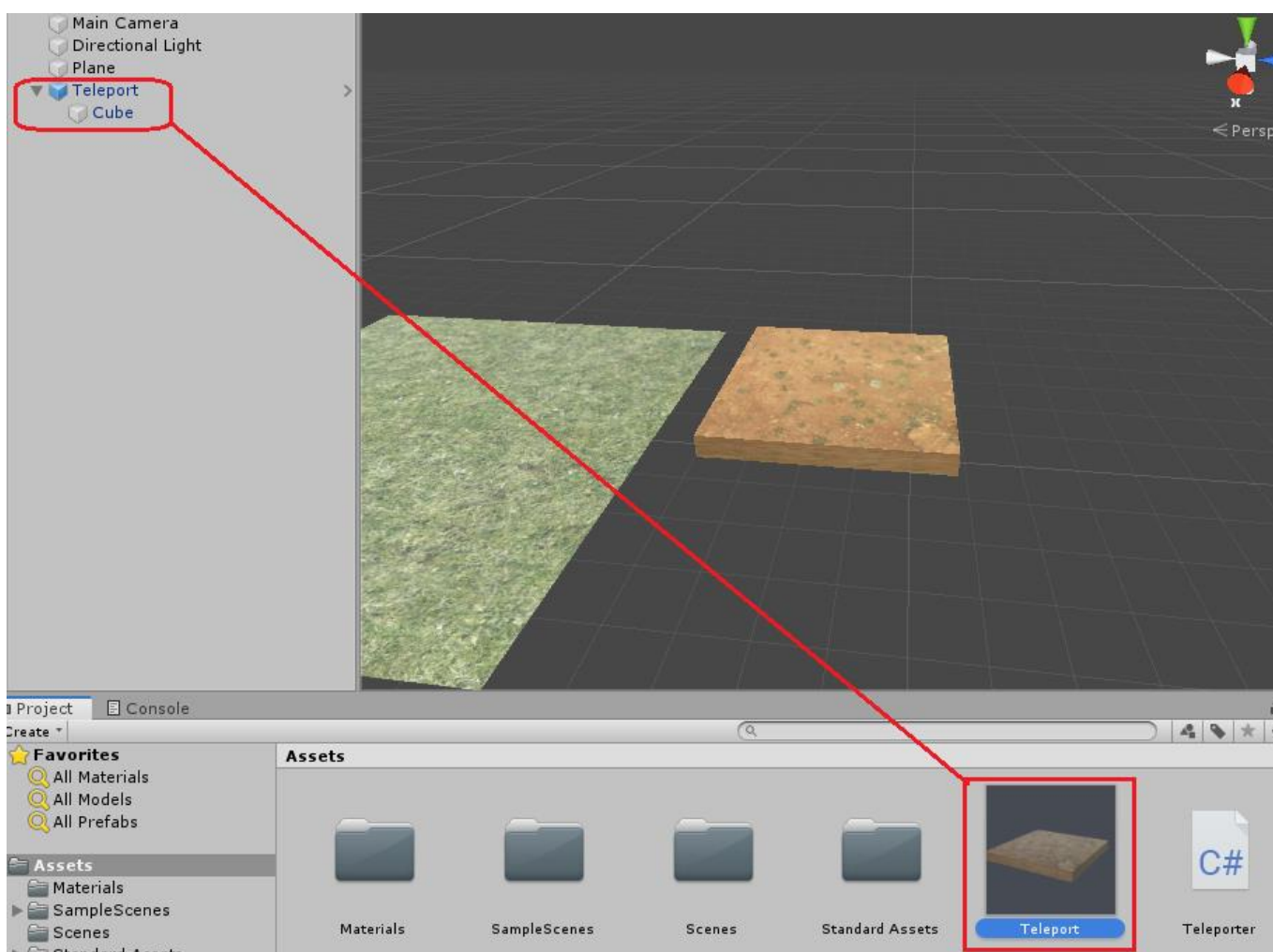




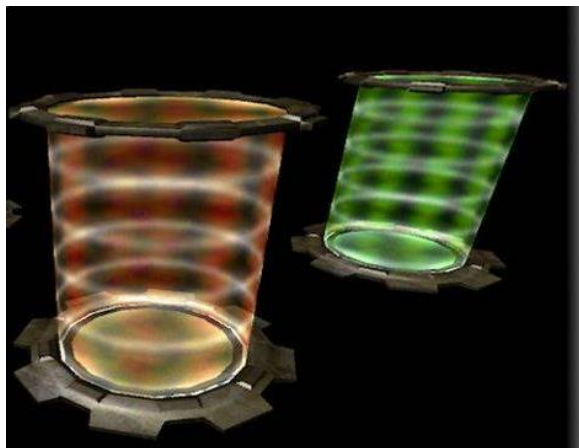
ТЕЛЕПОРТАЦИЯ

Создаем префаб

4. Сохраняем как Prefab



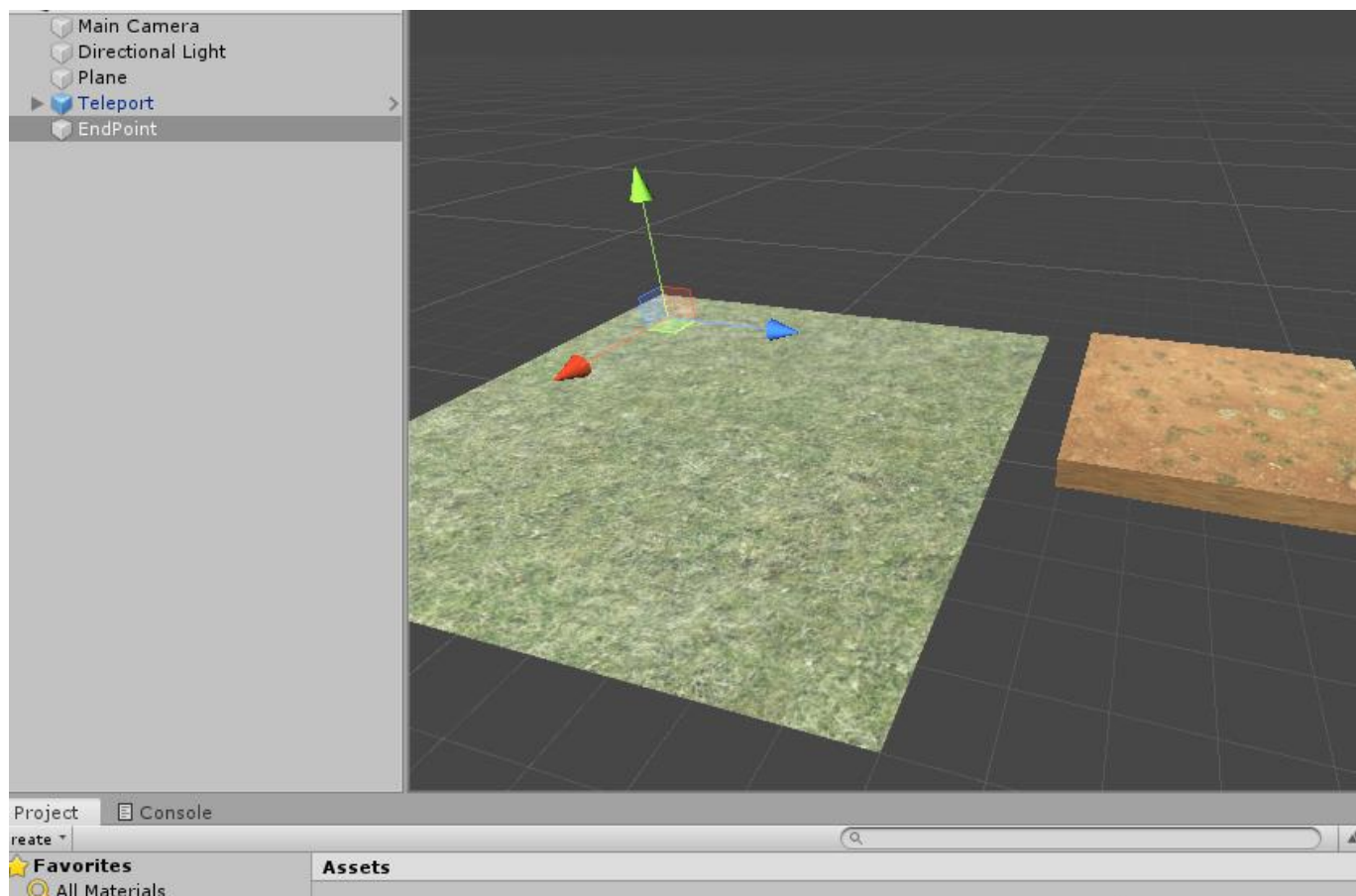
Созданный Prefab представляет собой портал (Teleport), при попадании на который игрок перемещается в другую точку. Осталось ее добавить



ТЕЛЕПОРТАЦИЯ

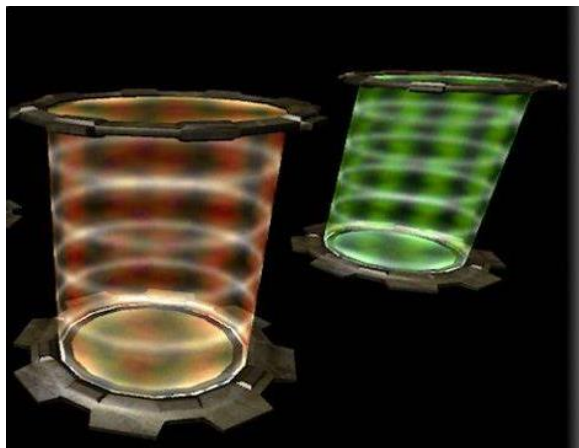
Конечная точка телепортации

Для добавления конечной точки создаем еще один Game Object (Create Empty) и переименовываем его как EndPoint. Расположим его на нашем Plane.



Остается только привязать эту точку к скрипту нашего префаба.

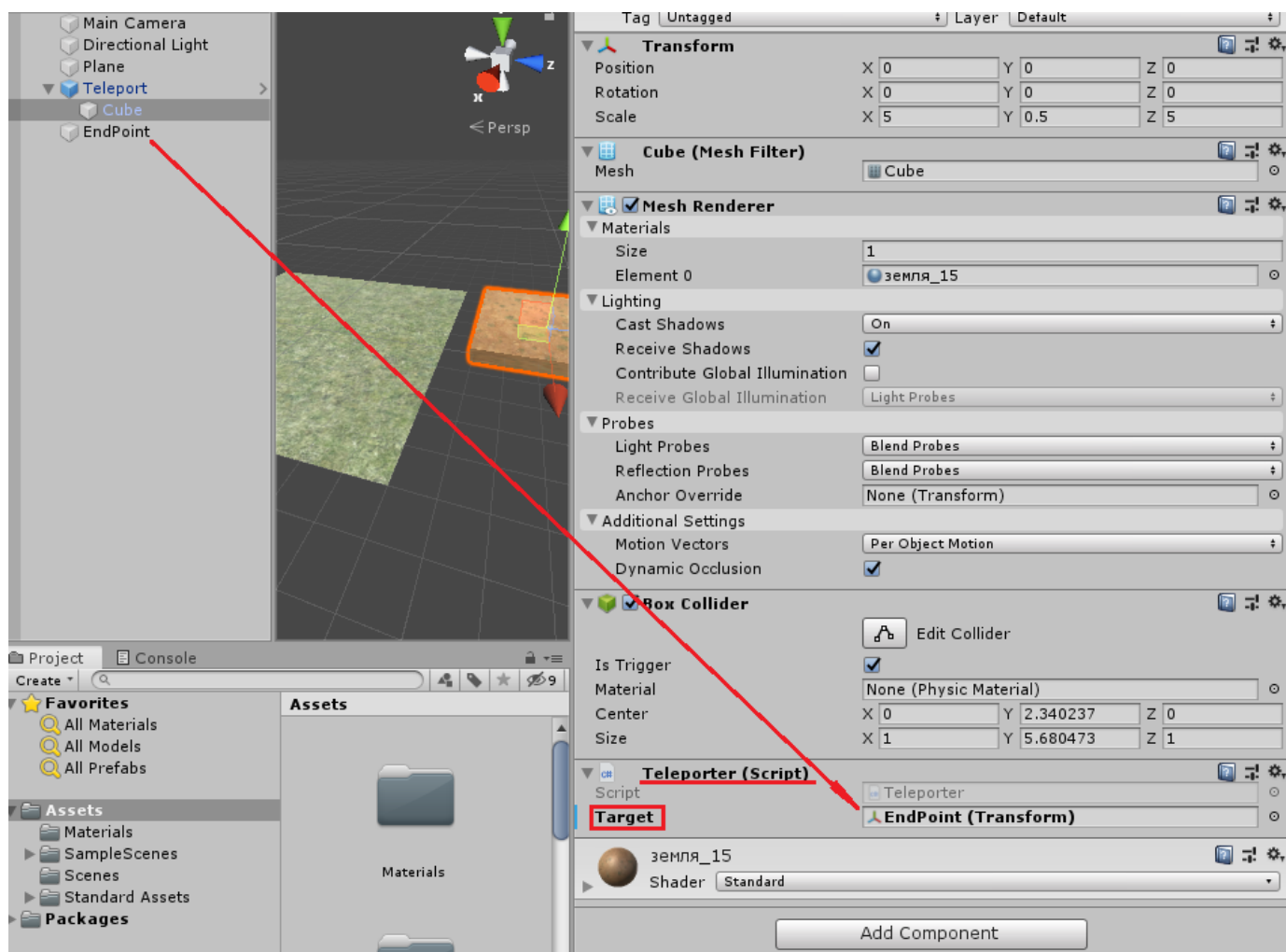




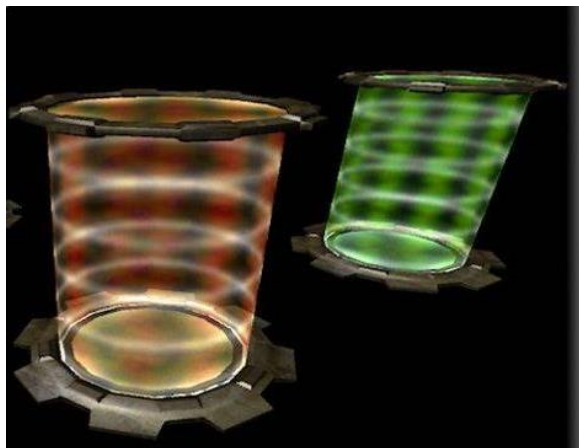
ТЕЛЕПОРТАЦИЯ

Привязка конечной точки к префабу (скрипту)

Для привязки конечной точки телепортации к нашему префабу со скриптом, перетаскиваем созданный EndPoint в поле Target скрипта.



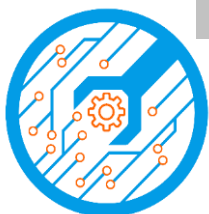
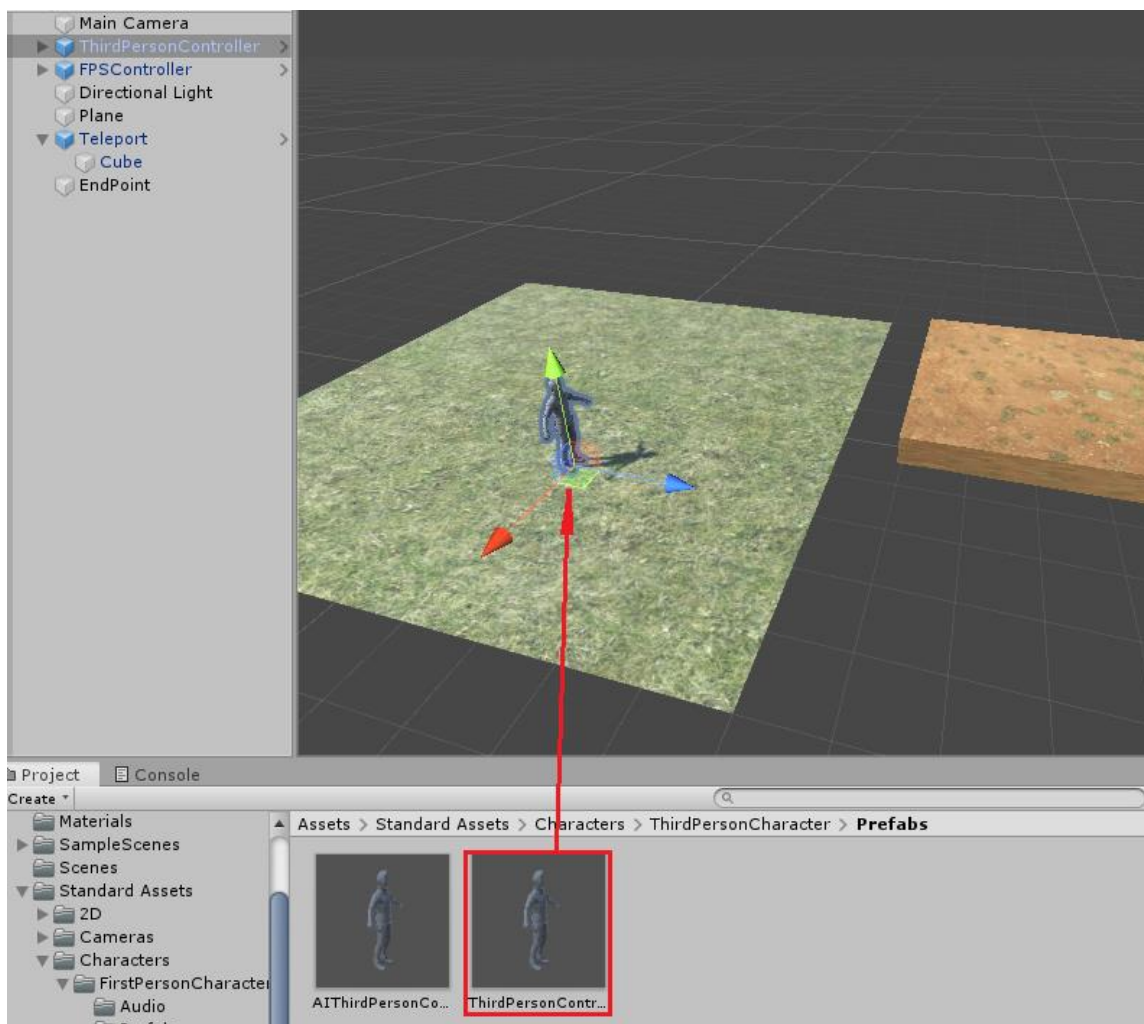
Теперь нужно добавить игрока и камеру из Standard Assets.

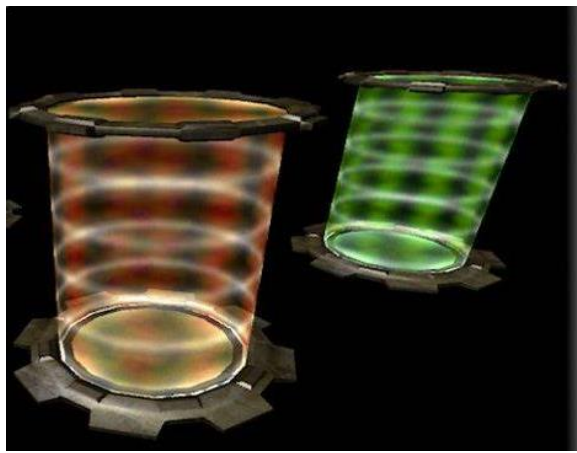


ТЕЛЕПОРТАЦИЯ

Добавляем игрока

В окне Assets выбираем: Assets → Standart Assets → Characters → ThirdPersonCharacter → Prefabs → ThirdPersonController и ставим его на Plane.

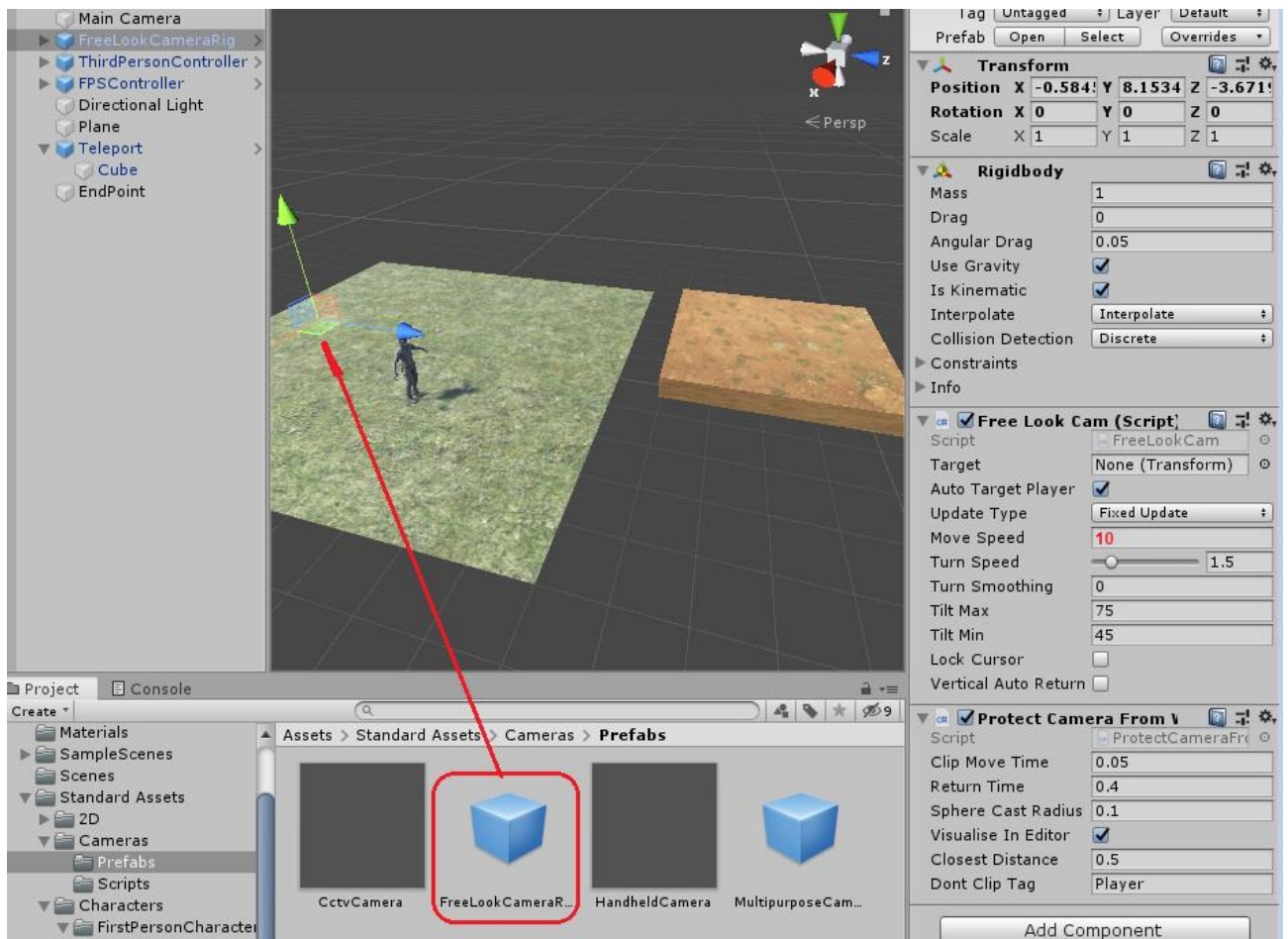




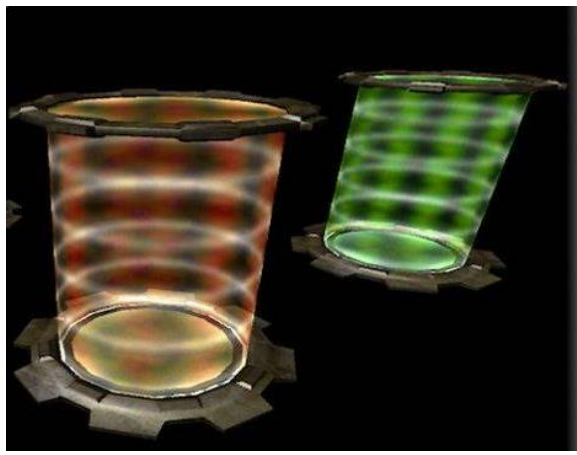
ТЕЛЕПОРТАЦИЯ

Добавляем камеру

В окне Assets выбираем: Assets → Standard Assets → Cameras → Prefabs → FreeLookCameraRig.



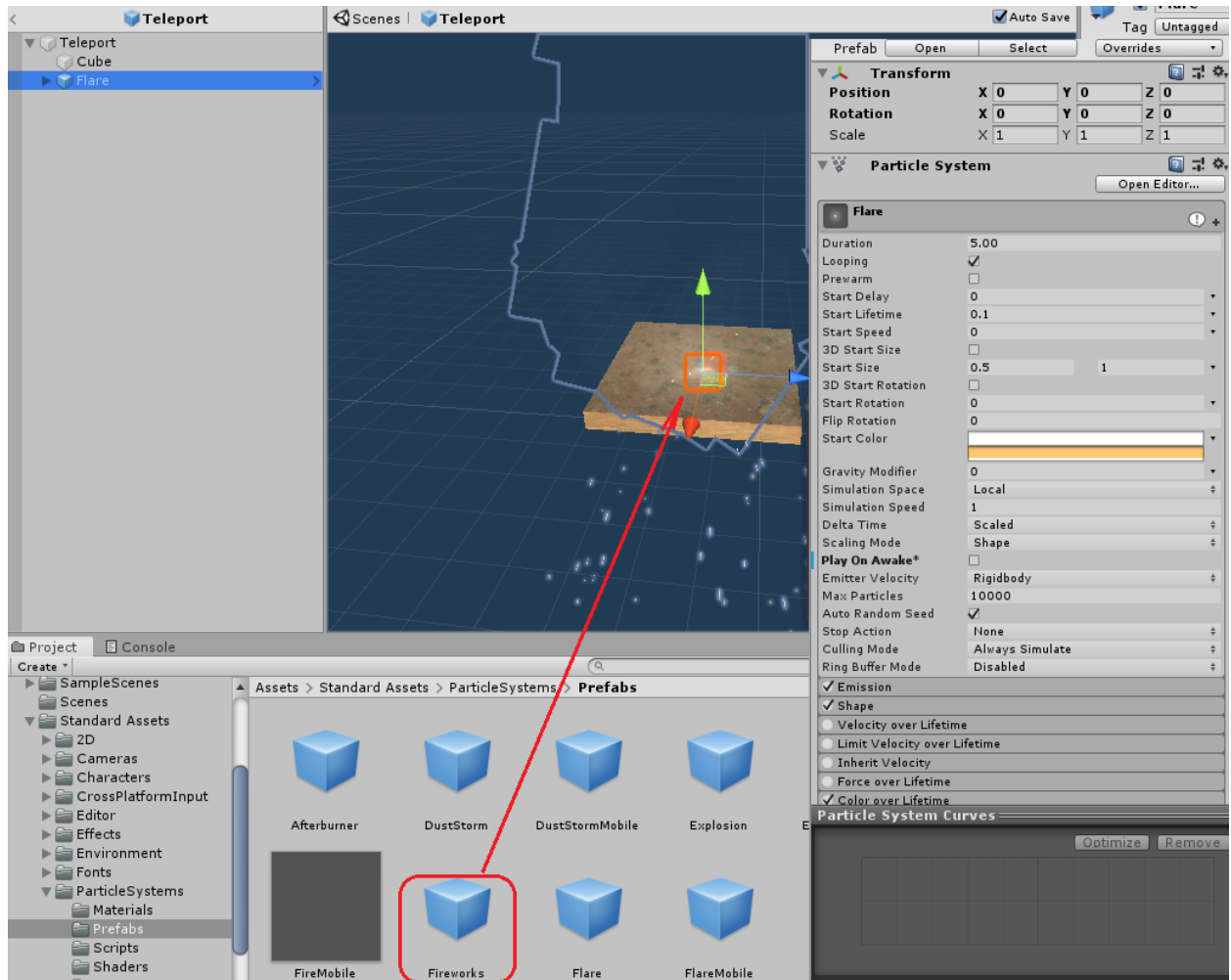
1. Значение Move Speed для камеры можно установить равным 10.
2. Удаляем стандартную камеру.
3. Проверяем работу игры



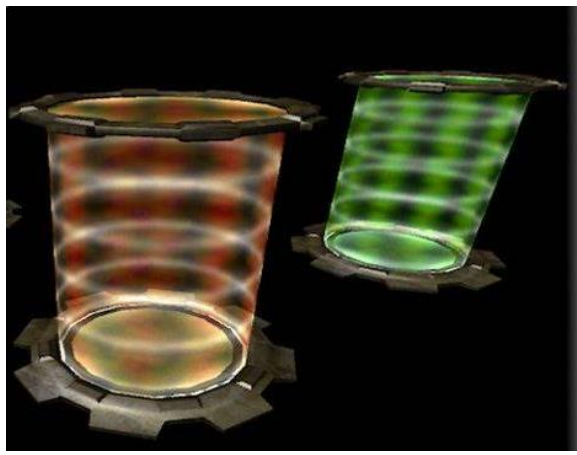
ТЕЛЕПОРТАЦИЯ

Система частиц

Для эффектного появления в конечной точке телепортации добавляем фейерверк: Assets → ParticleSystems → Prefabs → Flare.



Для этого открываем префаб на редактирование и добавляем в него Flare, выключив Play on Awake, чтобы не было запуска при старте.



ТЕЛЕПОРТАЦИЯ

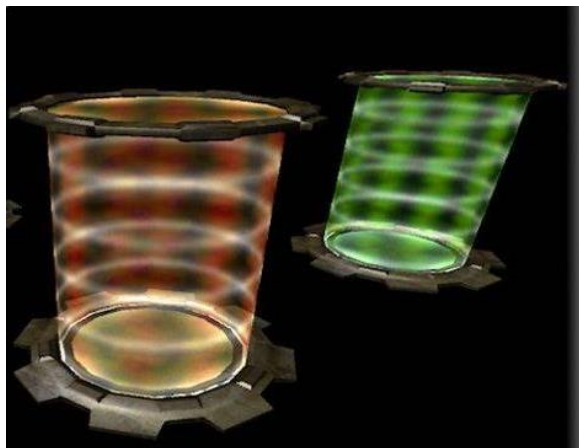
Добавление фейерверка в скрипт

```
public class Teleporter : MonoBehaviour
{
    // Подключаем сюда конечную точку телепорта
    public Transform Target;

    // Подключаем ссылку на частицы
    public ParticleSystem Particles;

    // При входе объекта в триггер (этот объект -
    // игрок, то есть other)
    void OnTriggerEnter(Collider other)
    {
        // Телепортировать объект на конечную точку
        other.transform.position =
Target.transform.position;
        // Перемещаем частицы к игроку
        Particles.transform.position =
Target.transform.position;
        // Запускаем их
        Particles.Play();
    }
}
```

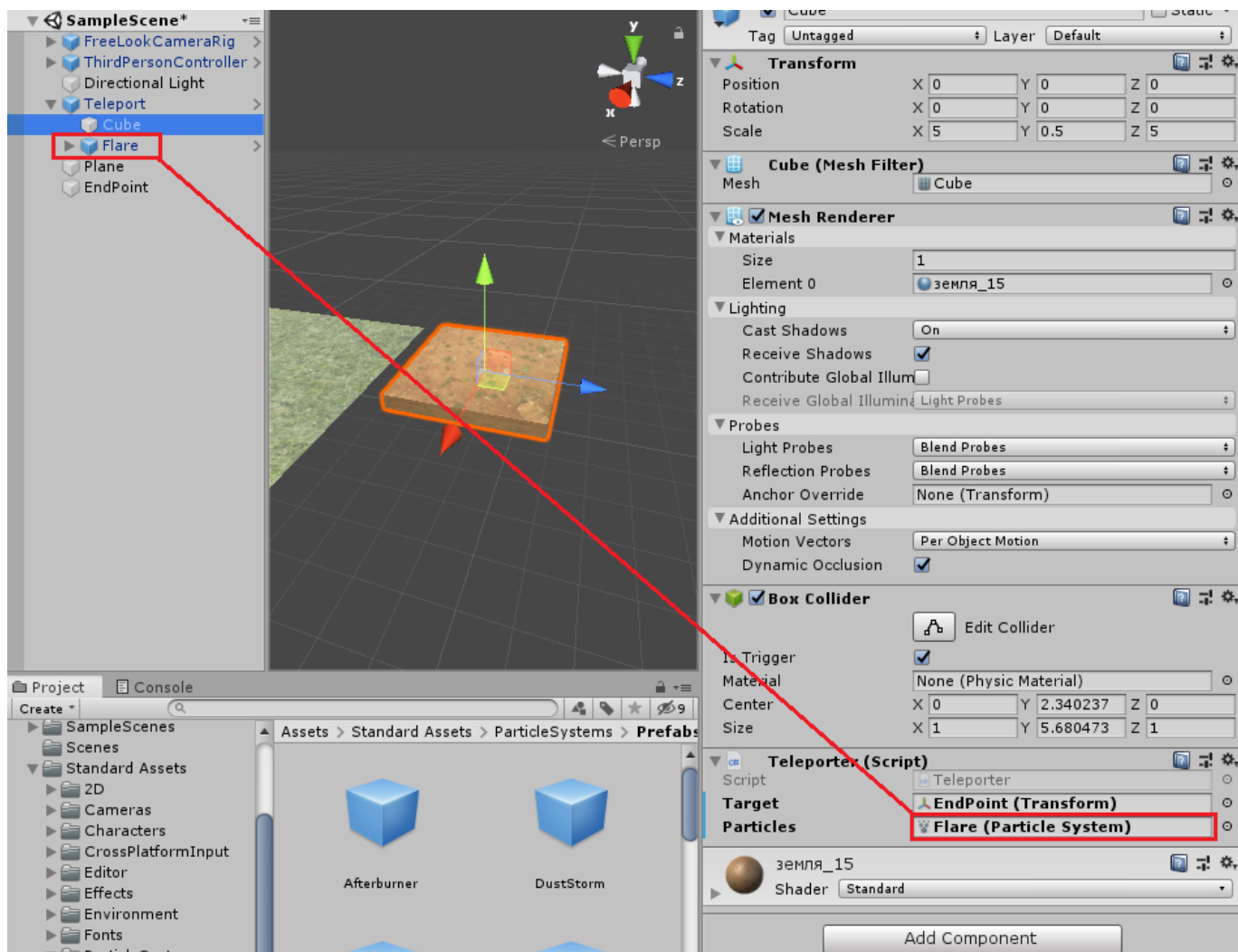




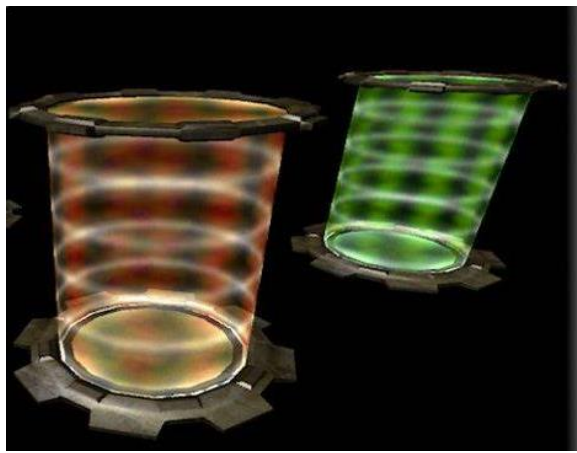
ТЕЛЕПОРТАЦИЯ

Система частиц

Перетаскиваем Flare в появившуюся в окне скрипта созданную нами public переменную.



Проверяем работу игры.

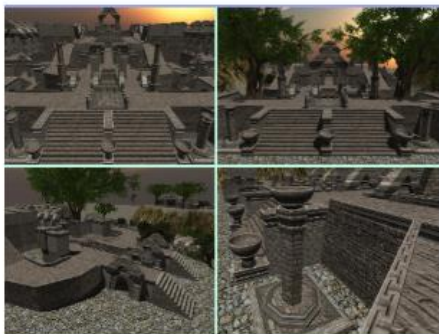


ТЕЛЕПОРТАЦИЯ

Самостоятельная работа

Импортируем бесплатный Asset

<https://assetstore.unity.com/packages/3d/environments/landscapes/p-w-temple-edition-33637>)



AQUARIUS MAX

(P&W) Temple Edition

★★★★★ 2 user reviews

As this is a very old pack of mine (one of my first, created some 5 or so years ago), i will be deprecating this soon. Until then, I've decided to make this FREE! 192 prefabs can always be used one way or another.

These models scales and their textures are not UV unwrapped that well compared to modern games, and at least are great for early prototyping. Enjoy this new freebie, until its removed.. Thanks!

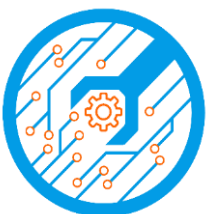
This package includes 192 prefabs!

[View Full Details](#)

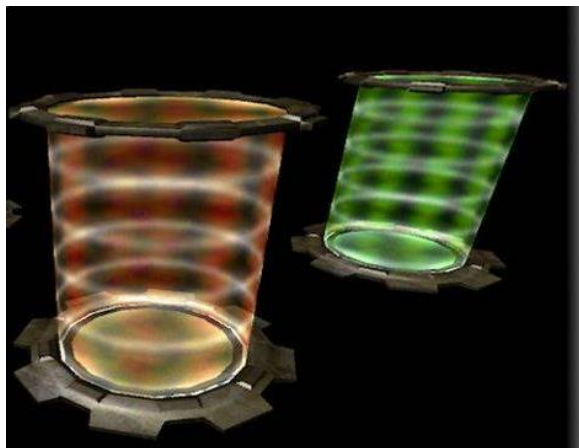
Package contents	15.0 MB
Releases	current ver. 1.1
Supported Unity versions	4.3.4 or higher

[Support website](#)

[Publisher website](#)



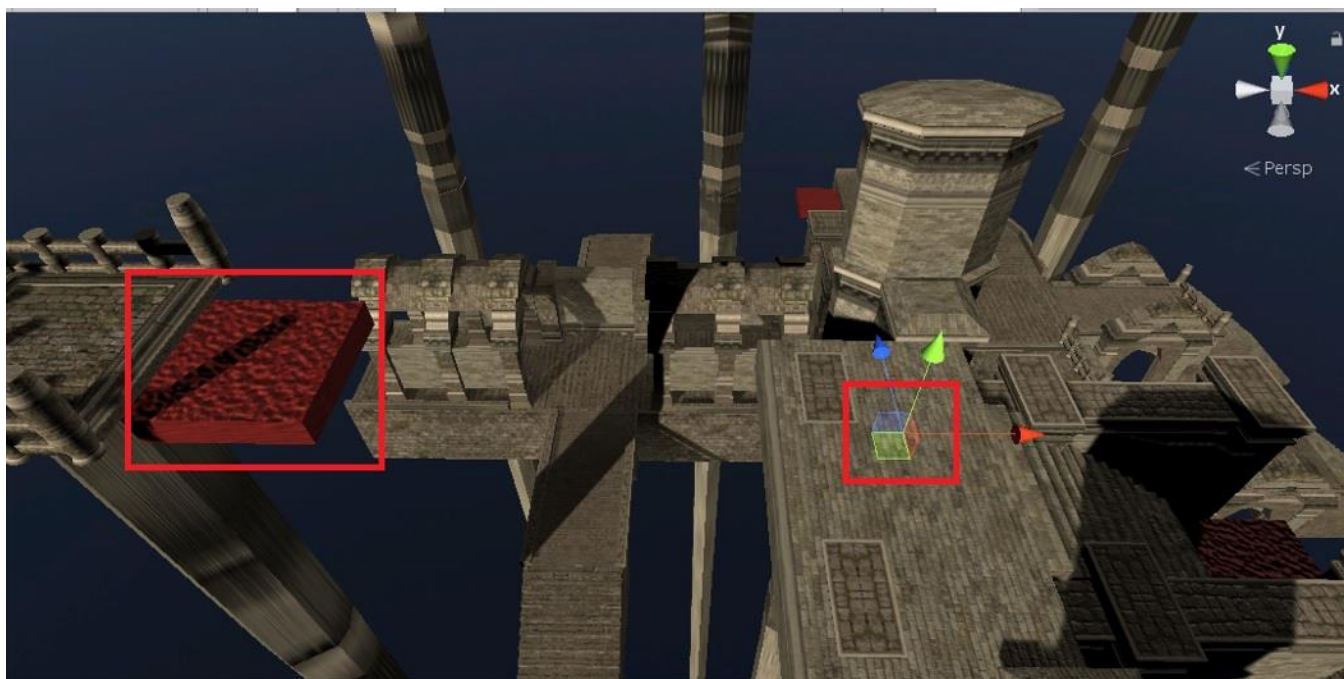
Создаем с помощью его элементов красивый уровень.



ТЕЛЕПОРТАЦИЯ

Самостоятельная работа

Расставляем телепорты (Teleport) и конечные точки (EndPoints) от 1 до N и связываем их (стр. 7 и 12):



Teleport1 → EndPoints1

Teleport2 → EndPoints2

Teleport3 → EndPoints3

Teleport4 → EndPoints4

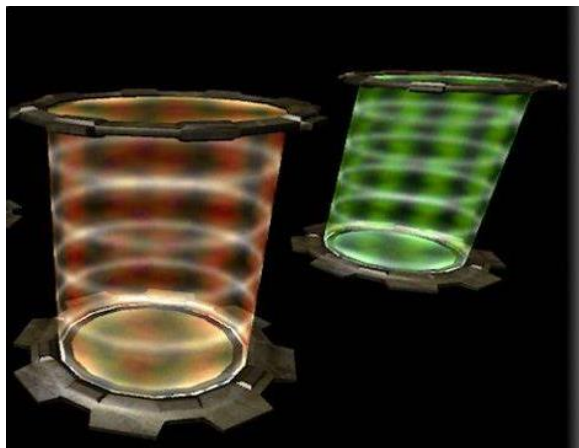
Teleport5 → EndPoints5

Teleport6 → EndPoints6

.....

TeleportN → EndPointsN

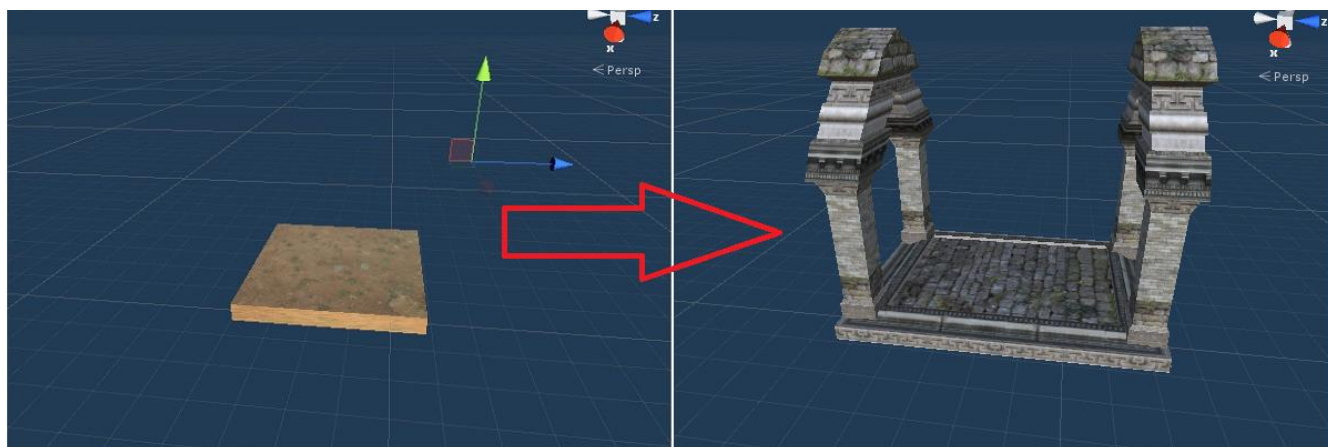




ТЕЛЕПОРТАЦИЯ

Самостоятельная работа создаем красивый телепорт

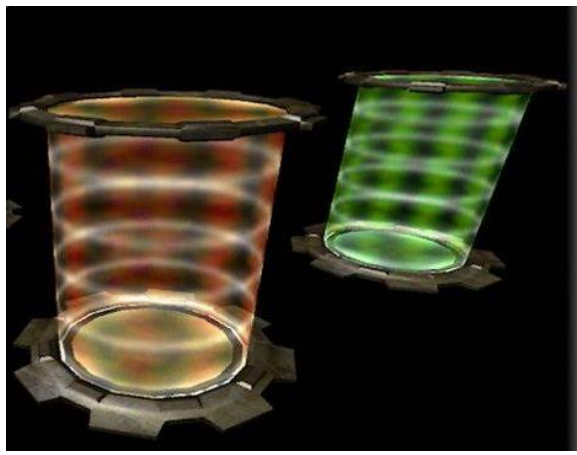
Чтобы не редактировать каждый портал (Teleport) отдельно, заходим в режим редактирования префаба (стрелочка возле объекта либо двойной клик по префабу в окне Project)



Поскольку к нашему Cube привязаны скрипты, просто оставляем его внутри.

Сохраняем изменения. Если все сделано верно, то все расставленные префабы изменятся.





ТЕЛЕПОРТАЦИЯ

ТАЙМЕР

Есть возможность считать время и телепортировать игрока к исходной точке по достижению определённого времени. Создаем новый скрипт с именем `LeverRestarter`.

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

```
public class LevelRestarter : MonoBehaviour
{
    public float MaxTime; // Максимальное возможное время игры
    private Vector3 StartPoint; // Стартовая точка

    private float currentTime; // Текущее время игры

    void Start ()
    {
        // Устанавливаем стартовую точку в том месте, где находимся в начале игры
        StartPoint = transform.position;

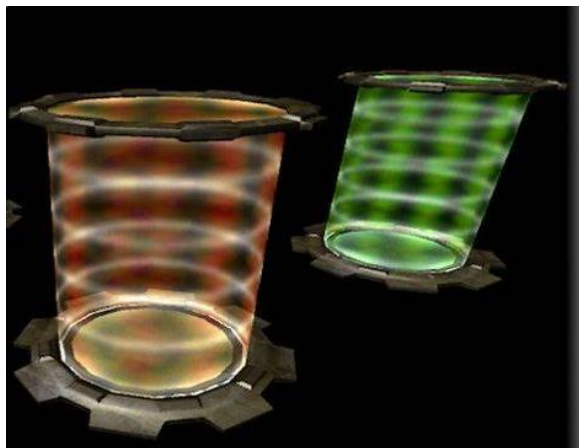
        currentTime = 0; // Обнуляем время в начале игры
    }

    void Update()
    {
        // Обновляем время игры
        currentTime += Time.deltaTime;

        if (currentTime > MaxTime) // Если время игры стало равно максимальному возможному
        {
            Debug.Log("Время вышло, рестарт..."); // Говорим, что время вышло

            // Перезагружаемся
            // transform.position - позиция объекта, на котором висит скрипт
            transform.position = StartPoint;

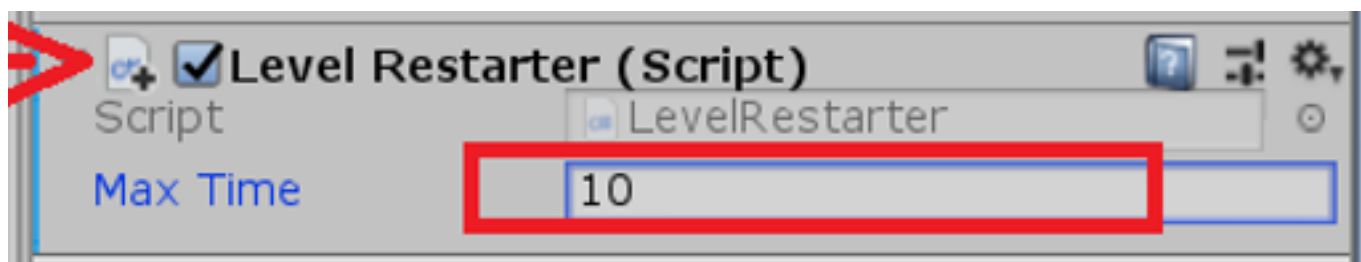
            // Обнуляем время
            currentTime = 0;
        }
    }
}
```

ТЕЛЕПОРТАЦИЯ

ТАЙМЕР

1. Перетаскиваем скрипт на ThirdPersonController.
2. Устанавливаем максимальное время.



Проверяем работу игры.

УДАЧИ!!!

