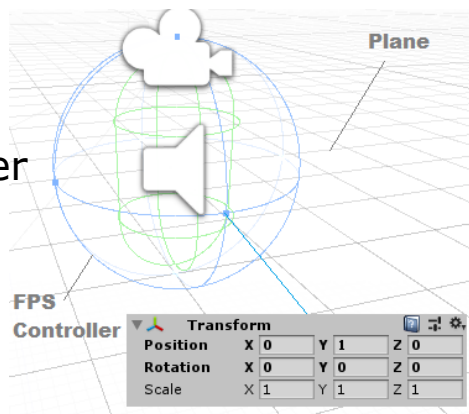


First-Person Shooter (FPS)

(часть 1: оружие и «огонь»)

1. Создаем пустой проект;
2. Добавляем и импортируем ассеты:
 - Standard Asset (для First Person Controller);
 - Weapon Master-SciFi (лайт бесплатная версия)
(<https://assetstore.unity.com/packages/3d/props/weapons/weapon-master-sci-fi-weapon-1-lite-134423>)
 - Modern Guns: Handgun
<https://assetstore.unity.com/packages/3d/props/guns/modern-guns-handgun-129821>
 - Terrain Tools Sample Asset Pack (не обязательно)
<https://assetstore.unity.com/packages/2d/textures-materials/terrain-tools-sample-asset-pack-145808>
3. Временно добавляем Plane с габаритами $X = 10$, $Y=1$, $Z =10$. Помещает данный Plane в нулевую позицию.
4. Заходим:
Asset -> Standard Assets ->
Characters -> FirstPersonCharacter
-> Prefabs и выбираем и добавляем на сцену:

FPSController



First-Person Shooter (FPS)

(часть 1: оружие и «огонь»)

4. Заходим:

Asset -> WM_SciFi_Weapon1_Lite -> Prefabs и выбираем и добавляем на сцену:

WM_SF_Wp01_a

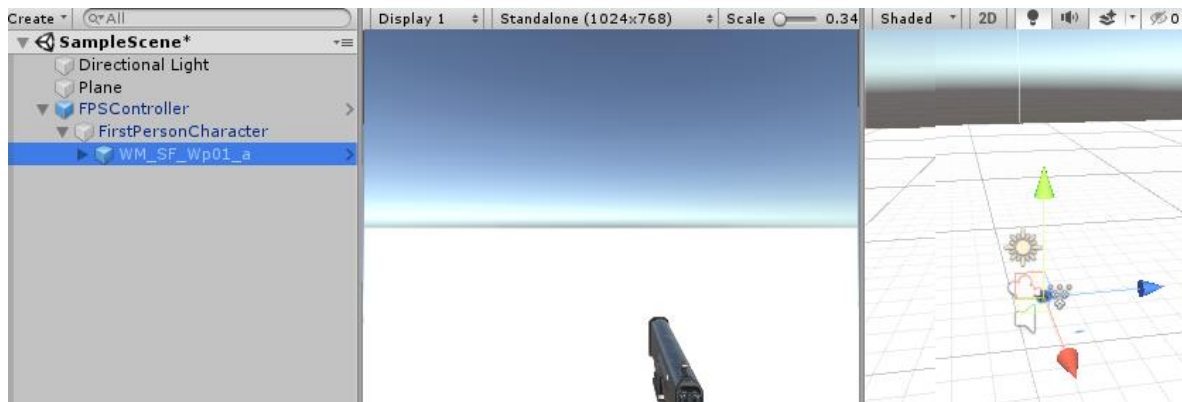
5. «Подчиняем» WM_SF_Wp01_a FPSController-у, как:

➡ FPSController

➡ FirstPersonCharacter

➡ WM_SF_Wp01_a

При такой иерархии видимая часть ствола будет перемещаться вместе с камерой.



6. Обнуляем позицию WM_SF_Wp01_a относительно FPSController. Далее располагаем окна Scene и Game так, чтобы видеть изменение положения WM_SF_Wp01_a.

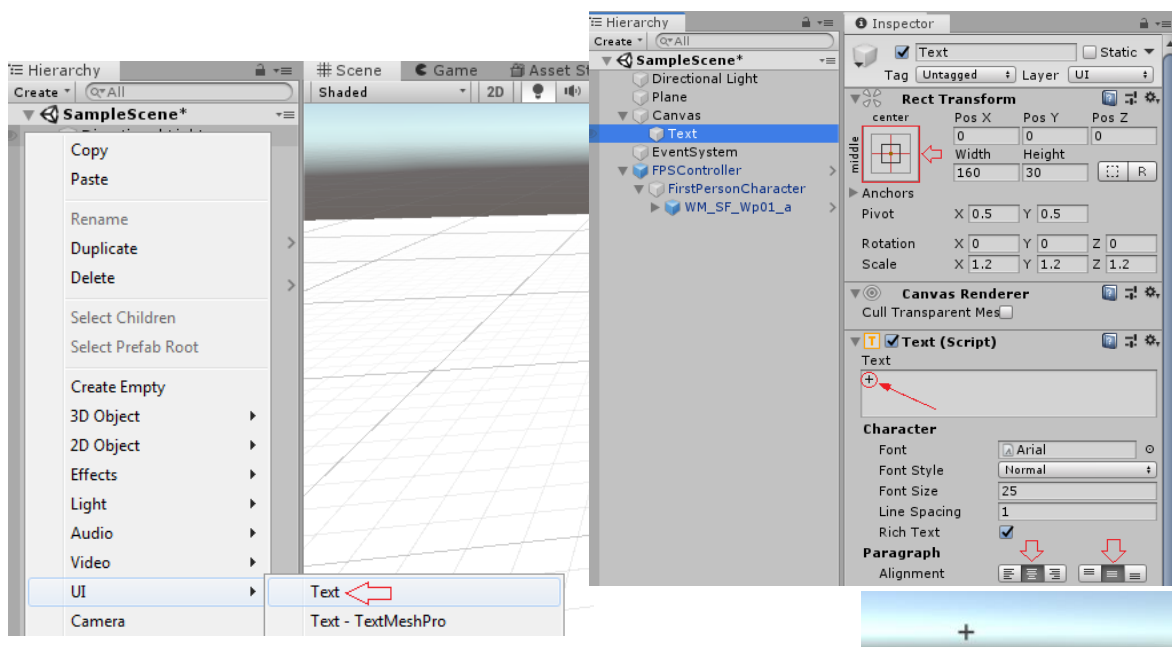


Изменяя координаты, добиваемся видимости ствола – как от первого лица.

First-Person Shooter (FPS)

(часть 1: оружие и «огонь»)

7. Для добавления прицела, с помощью User Interface помещаем «+» в центр экрана. Для этого, выбрав UI -> Text, производим центровку, как текстового поля (Middle – по вертикали и Center – по горизонтали.), так и самого «+» внутри него.



Для достижения требуемого эффекта также подберем вид шрифта, его размер, Scale и цвет.



First-Person Shooter (FPS)

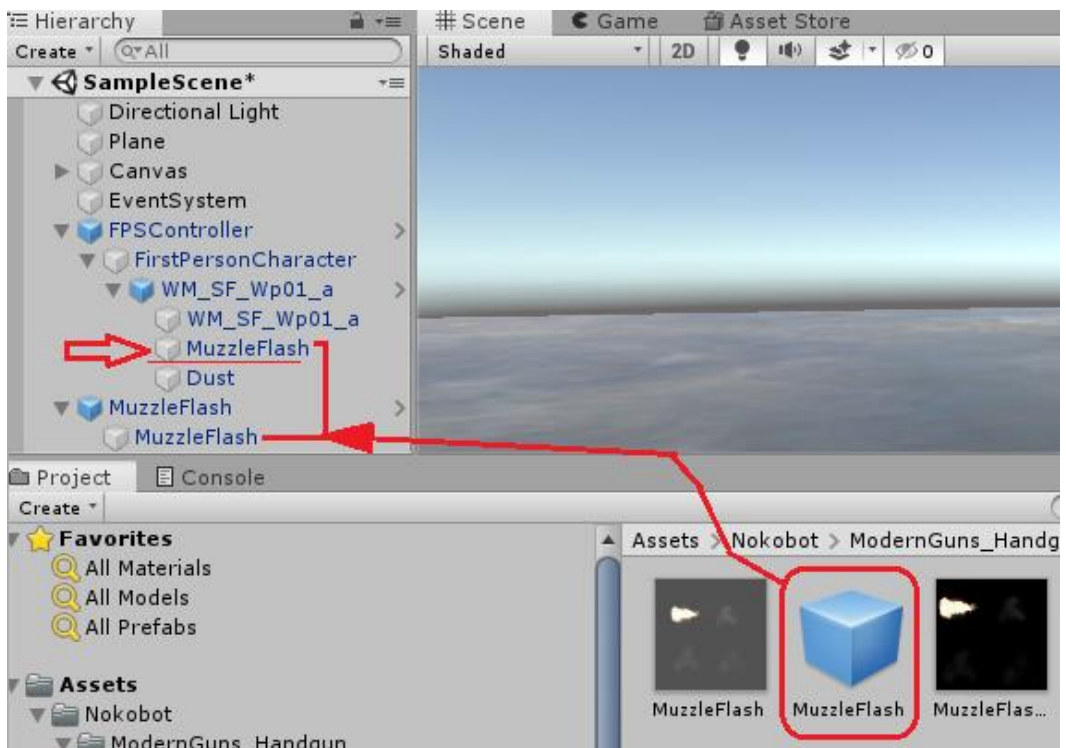
(часть 1: оружие и «огонь»)

8. Добавляем «дульное пламя» (**Muzzle Flash**).

Для этого заходим в уже импортированный Asset **Modern Guns**:

Asset -> Nokobot -> ModernGuns_Handgun -> Effects и перетаскиваем в окно иерархии **MuzzleFlash**.

Задача заключается в извлечении самого компонента Muzzle Flash из одноименного префаба и его подчинение префабу WM_SF_Wp01_a.

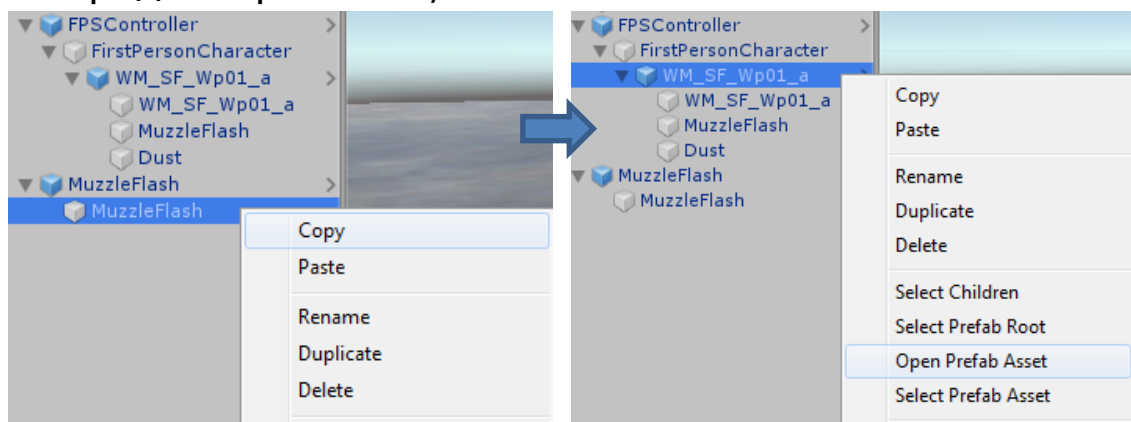


First-Person Shooter (FPS)

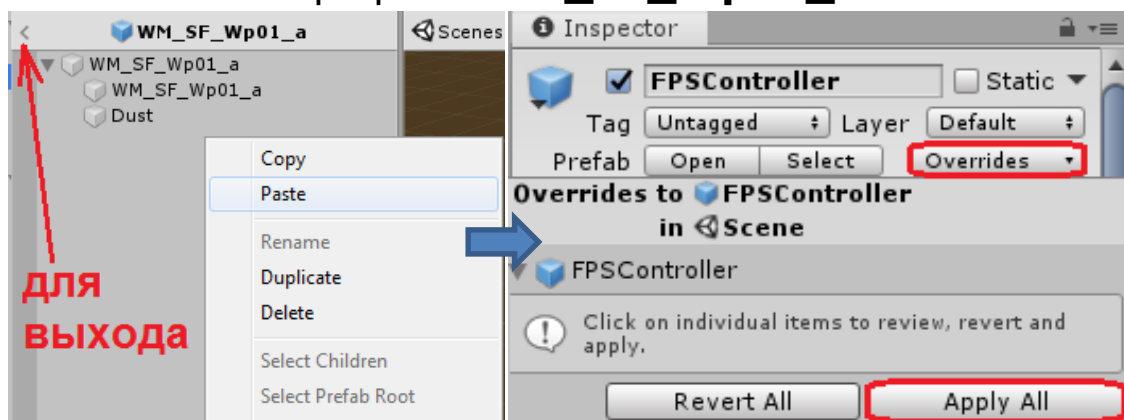
(часть 1: оружие и «огонь»)

Для этого необходимо выполнить пять операций:

- Скопировать интересующий нас компонент;
- Открыть префаб **WM_SF_Wp01_a** на редактирование;



- Поместить **Muzzle Flash** в **WM_SF_Wp01_a**;
- Выйти из префаба **WM_SF_Wp01_a**.

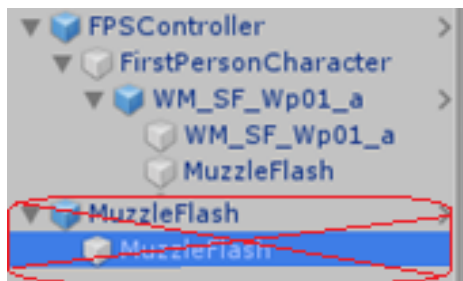


- Выбрать в Иерархии **FPSController**, затем в Inspector выбрать Overrides и нажать Apply All.

First-Person Shooter (FPS)

(часть 1: оружие и «огонь»)

Далее необходимо удалить сам префаб Muzzle Flash, использованный в качестве источника.

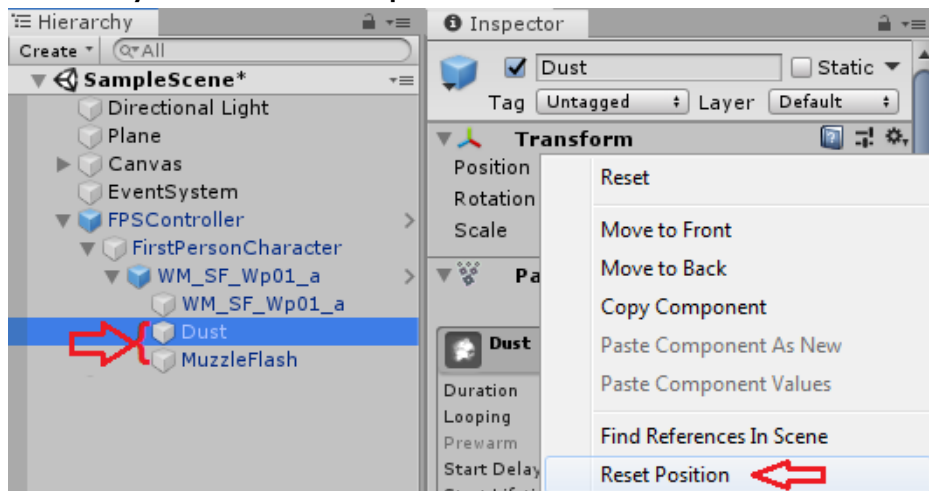


Аналогично необходимо добавить компонент **Dust** (пыль) из префаба **Explosion**, находящегося в Particles Systems ассета Standard Assets.

Если все было сделано верно, иерархия FPSController будет иметь следующий вид.



Как для Dust, так и для MuzzleFlash необходимо выполнить Reset Position. Так они окажутся на одной условно обнуленной координате.

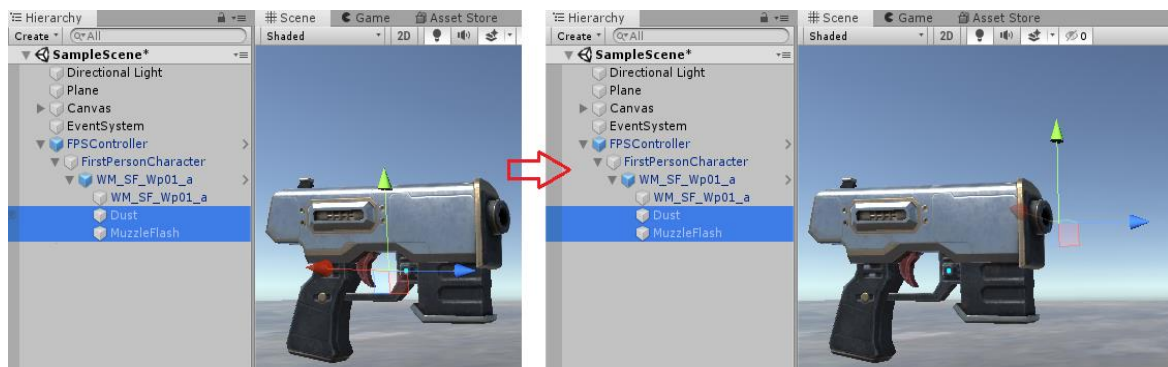


First-Person Shooter (FPS)

(часть 1: оружие и «огонь»)

Обнуление положения (Reset Position) приведет к тому, что центр компонентов **Dust** и **MuzzleFlash** окажется в условном центре **WM_SF_Wp01_a**. Теперь оба эти компонента необходимо переместить так, чтобы сместить их положение непосредственно к выходу из ствола.

В дальнейшем координаты центров **Dust** и **MuzzleFlash** можно настроить отдельно.



Далее необходимо скачать один из звуков выстрела. Коллекцию звуков можно найти на сайте:

<https://noisefx.ru/skachat-zvuki-pistoleta.html>

Выбрав необходимый звук, необходимо с помощью online-сервиса <https://mp3cut.net/ru/> выбрать требуемый фрагмент звука, длительностью порядка одной секунды (1 сек.) и сохранить в файл Shot.mp3.

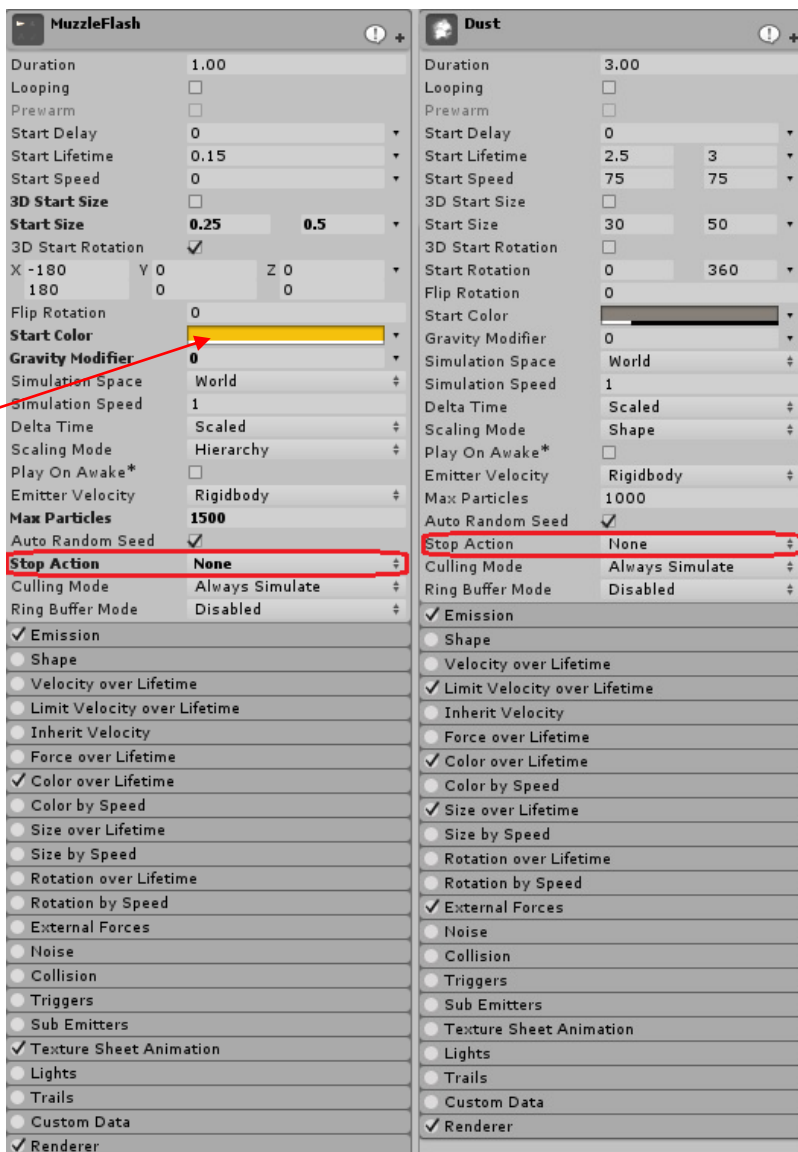


First-Person Shooter (FPS)

(часть 1: оружие и «огонь»)

Настройки компонентов системы частиц **Dust** и **MuzzleFlash** примерно следующие. Не забудем убрать Play on Awake.

подбор цвета пламени



Важно: установить Stop Action в значение None.

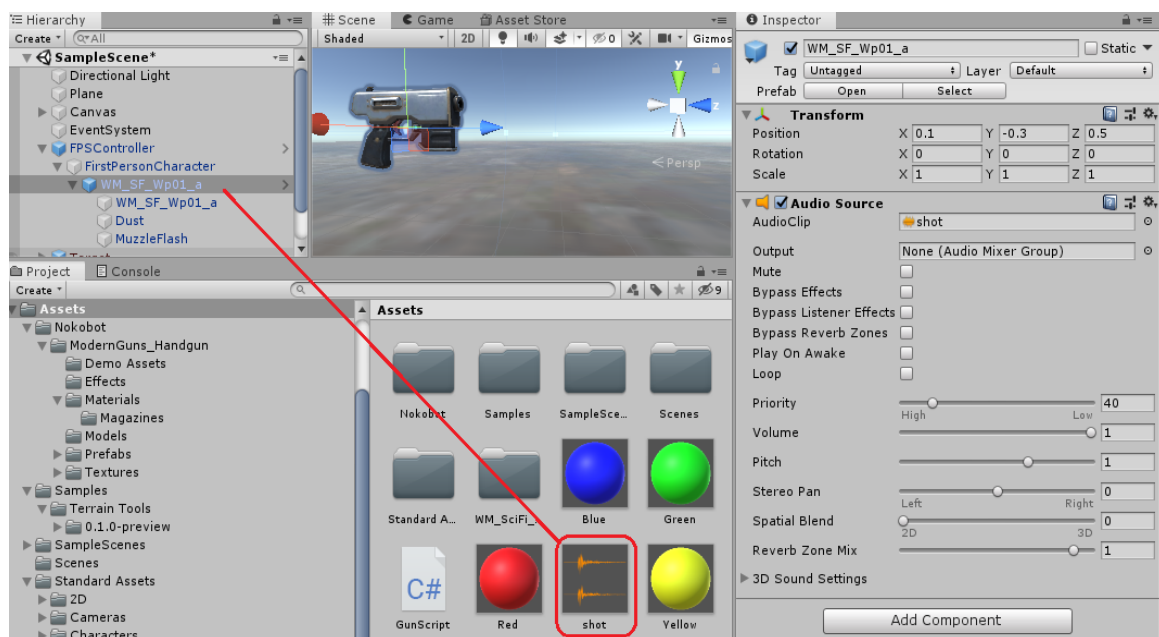


First-Person Shooter (FPS)

(часть 1: оружие и «огонь»)

9. Добавляем звук выстрела Shot.mp3 перетаскиванием в окно Assets.

После чего перетягиваем появившийся в Asset значок файла Shot на **WM_SF_Wp01_a**.



После чего, при выборе **WM_SF_Wp01_a**, в Inspector-е появятся свойства Audio Source. Обязательно уберем «галку» в CheckBox **Play on Awake**.

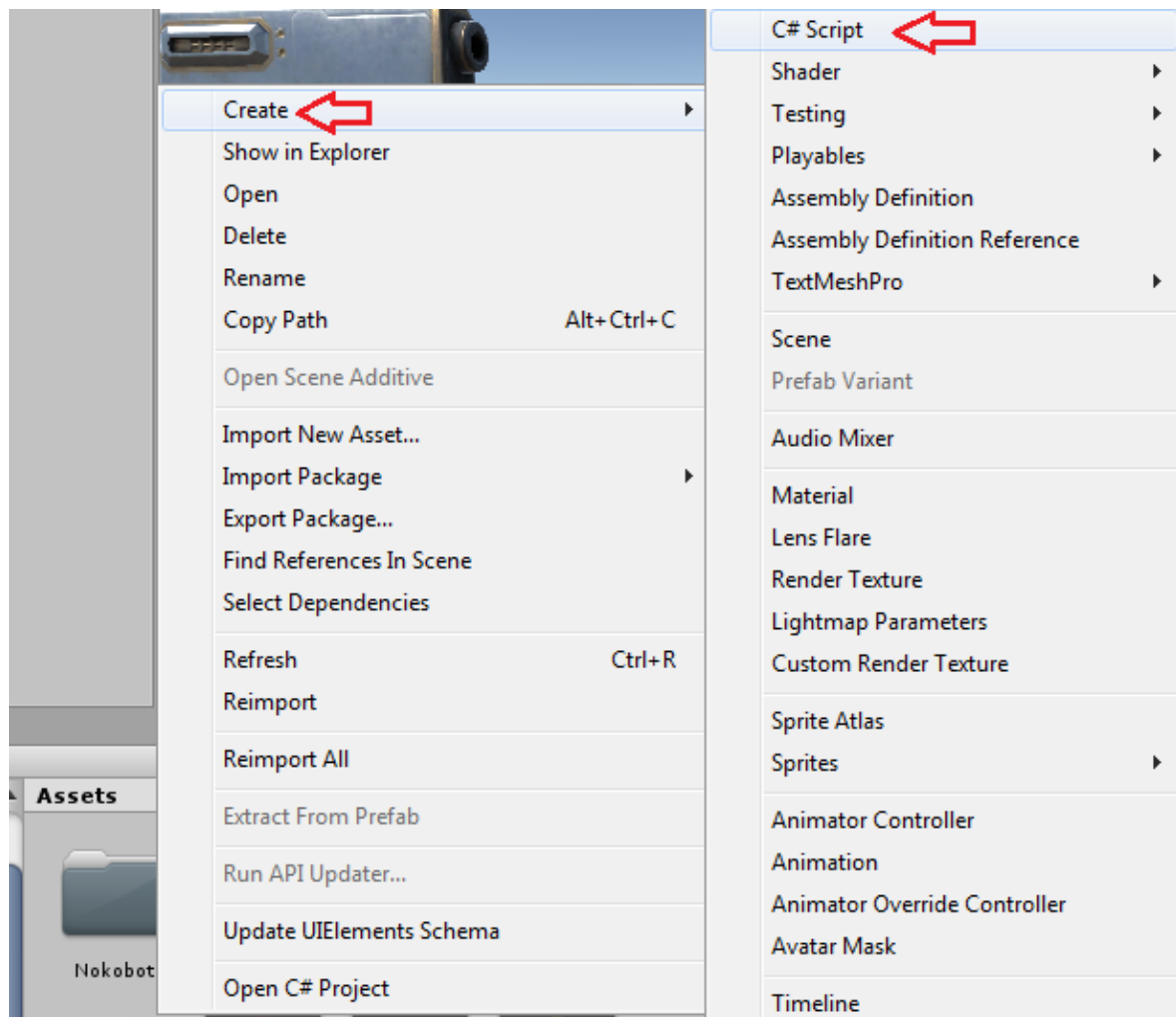
Осталось добавить скрипт.



First-Person Shooter (FPS)

(часть 1: оружие и «огонь»)

10. Добавляем скрипт



Переименовываем скрипт в **GunScript**.



First-Person Shooter (FPS)

(часть 1: оружие и «огонь»)

11. Код скрипта

```
/*подключаемые библиотеки*/  
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
/*созданный класс GunScript */  
public class GunScript : MonoBehaviour  
{  
    public float range = 100f;  
  
    float normalSize;           // переменные для  
    CharacterController controller; // функции «присесть»  
  
    public ParticleSystem particlesDust; // дым выстрела  
    public ParticleSystem muzzleFlash; // дульное пламя  
    public GameObject particleExplosionPrefab; // взрыв  
    private GameObject getExpired; // от попадания пули  
    private Camera gameCam; // камера  
    public AudioClip FireVoice; // подключаемый файл  
    AudioSource audioSource; // звука выстрела
```



First-Person Shooter (FPS)

(часть 1: оружие и «огонь»)

```
void Start() // запускается при старте игры
{
    gameCam = Camera.main; // инициализация камеры

    /*получение габаритов высоты игрока*/
    controller = GetComponentInParent <CharacterController>();
    audioSource = GetComponent<AudioSource>();
    normalSize = controller.height;

} //End Start()

void Update() // выполняется каждую смену кадра игры
{
    if (Input.GetButtonDown("Fire1")) // левая кнопка мыши
    {
        Shoot(); // «огонь»
    }
    if (Input.GetButtonDown("Left Control")) // клавиша L. Ctrl
    {
        controller.height = normalSize * 0.5f; // присесть
    }
    if (Input.GetButtonUp("Left Control")) // отпускаем L. Ctrl
    {
        controller.height = normalSize;
    }
} //end Update()
```



First-Person Shooter (FPS)

(часть 1: оружие и «огонь»)

```
private void Shoot()
{
    muzzleFlash.Play(); // дульное пламя
    particlesDust.Play(); // дым
    audioSource.PlayOneShot(FireVoice, 0.9f); // Play Shot

    RaycastHit hit; // направление условной пули

    if (Physics.Raycast(gameCam.transform.position,
        gameCam.transform.forward, out hit, range))
    {
        getExpired = Instantiate(particleExplosionPrefab, hit.point,
            Quaternion.identity); // взрыв в точке попадания пули
        Destroy(getExpired, 5.0f); // уничтожаем клон взрыва
    } end if
} //end Shoot()
} // end class GunScript
```

Краткое пояснение к скрипту:

RaycastHit - рейкаст отправляет воображаемый «лазерный луч» вдоль взгляда камеры (куда смотрит прицел), до тех пор, пока он не встретит на пути коллайдер.

gameCam.transform.position – точка расположения камеры;

gameCam.transform.forward – луч направлен вперед;

out hit – точка столкновения по пути луча;

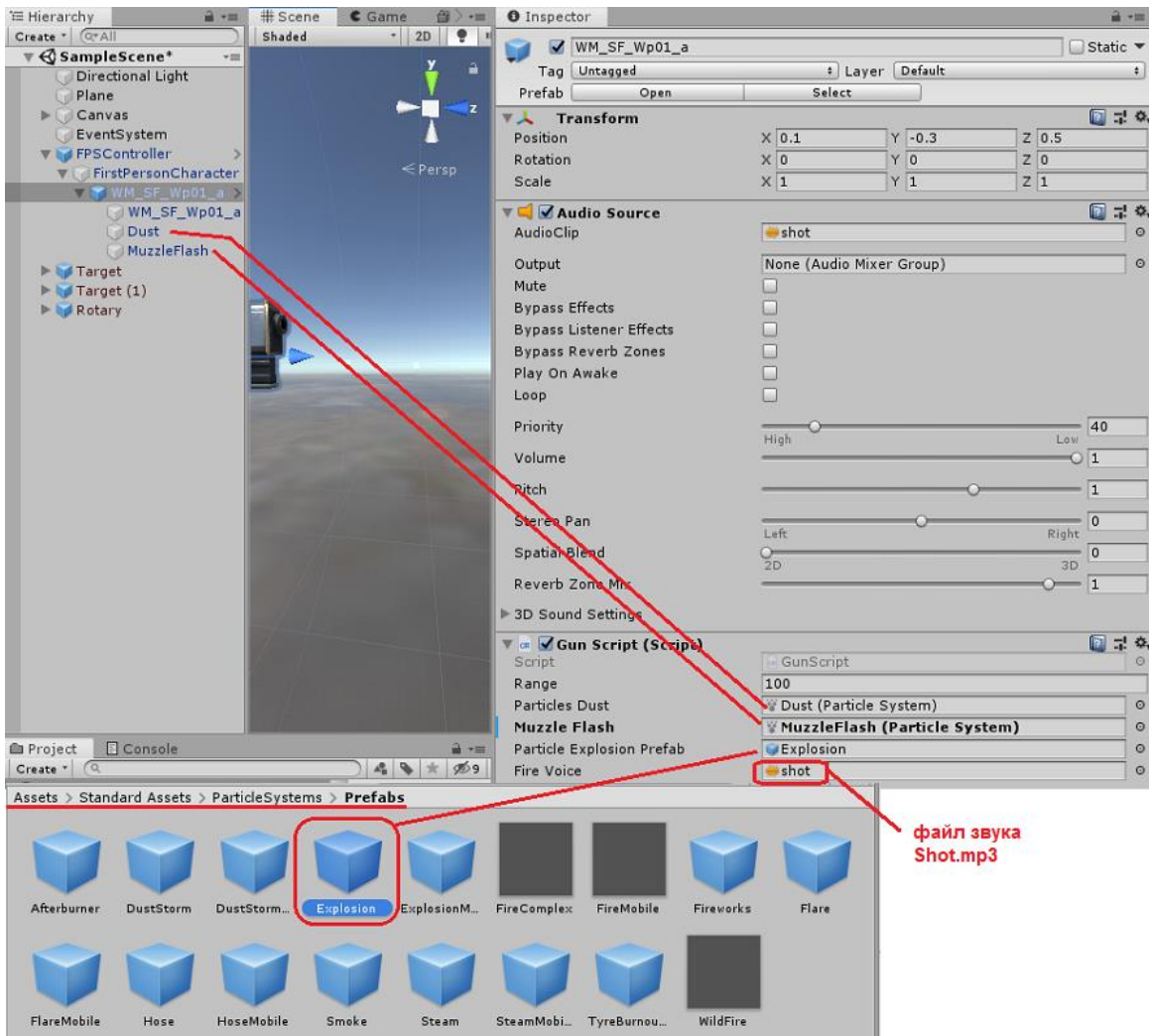
range - максимальное расстояние, куда пуля «долетает».



First-Person Shooter (FPS)

(часть 1: оружие и «огонь»)

11. Распределим переменные



Explosion из Standard Assets в окно переменной скрипта добавляется «как есть». Это взрыв в месте попадания пули.

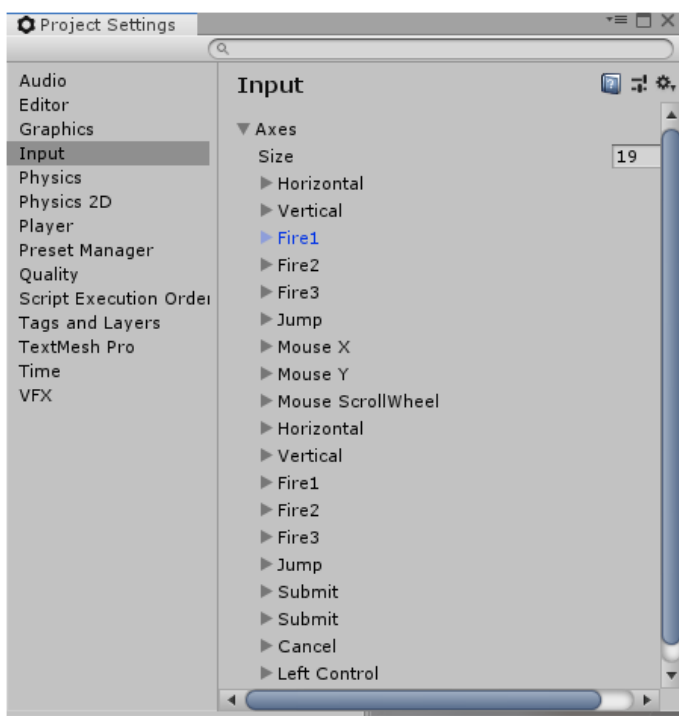
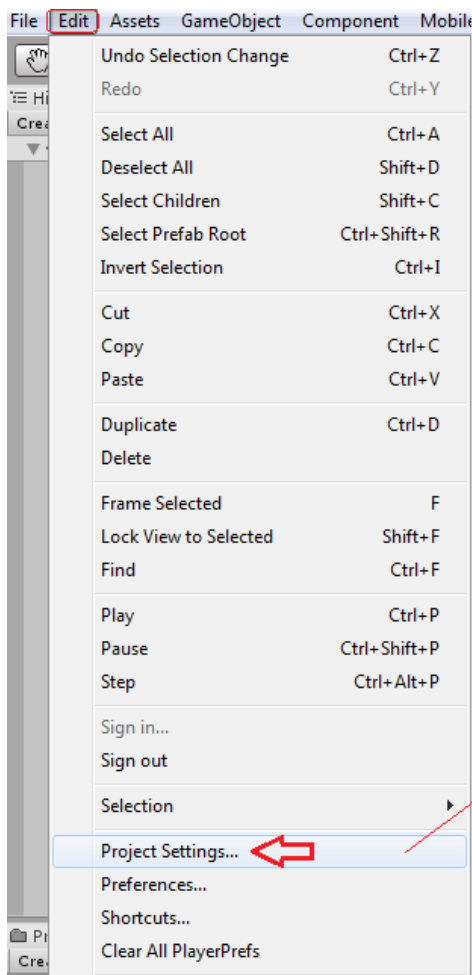
First-Person Shooter (FPS)

(часть 1: оружие и «огонь»)

12. Настройка клавиш управления

Стандартные и пользовательские клавиши управления игроком, «огнем» и т.д. Можно найти выбором пункта:

меню: **Edit -> Project Setting**



Выбираем Input. Откроются Axes (в данном случае кнопки управления).



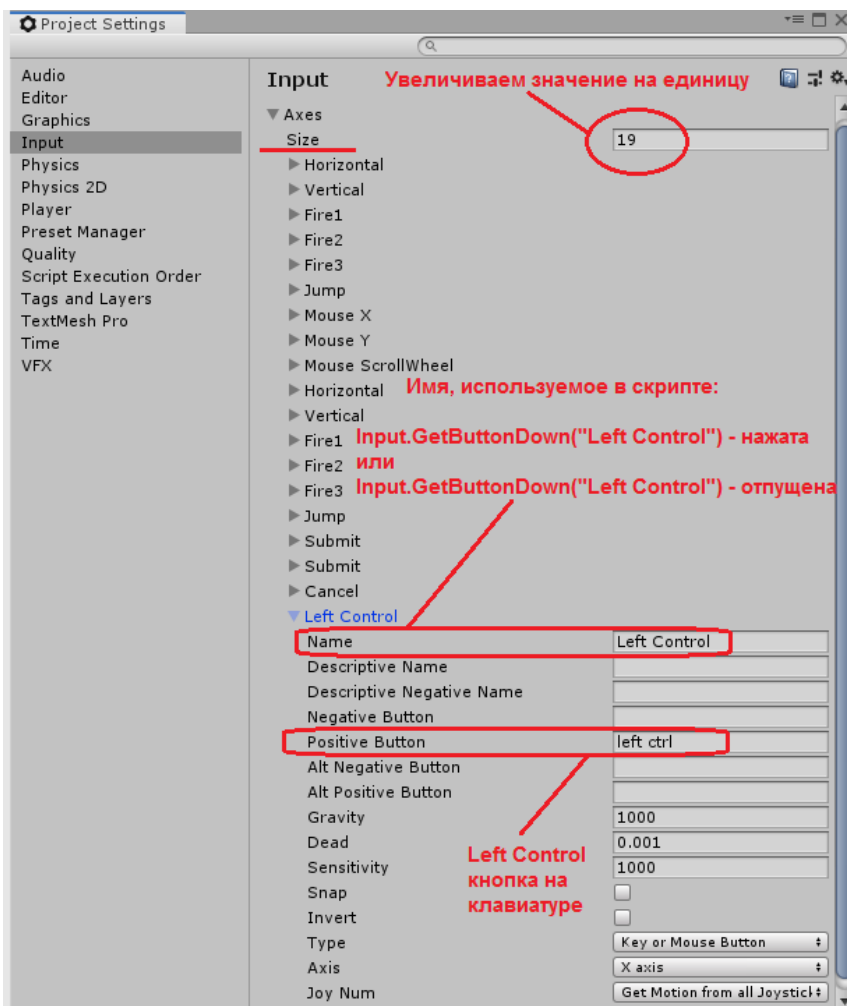
Здесь можно увидеть и управление «Огнем» и перемещения с прыжками.

First-Person Shooter (FPS)

(часть 1: оружие и «огонь»)

13. Добавление кнопки управления.

Новая кнопка добавляется путем увеличения значения **Size** на единицу. При этом продублируется последняя кнопка в списке. Переименовываем ее так, как используем в скрипте.

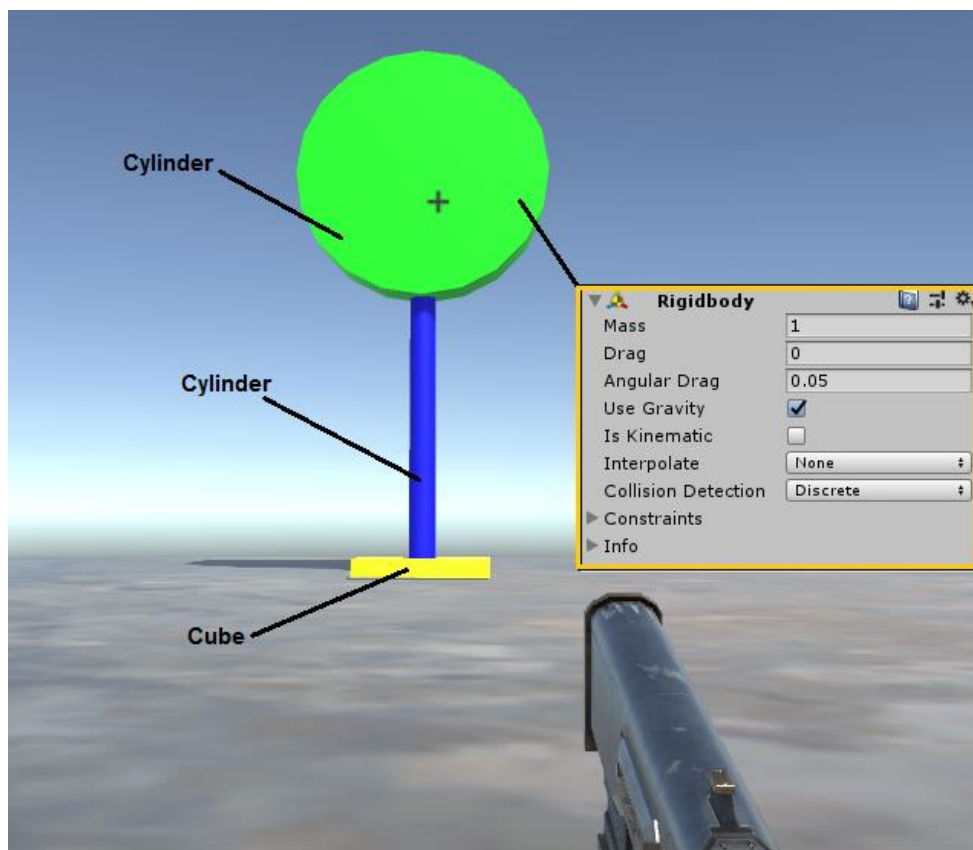


First-Person Shooter (FPS)

(часть 1: оружие и «огонь»)

14. Добавление мишени.

Мишень для проверки работы скрипта может быть изготовлена из двух примитивов: Cube и Cylinder.



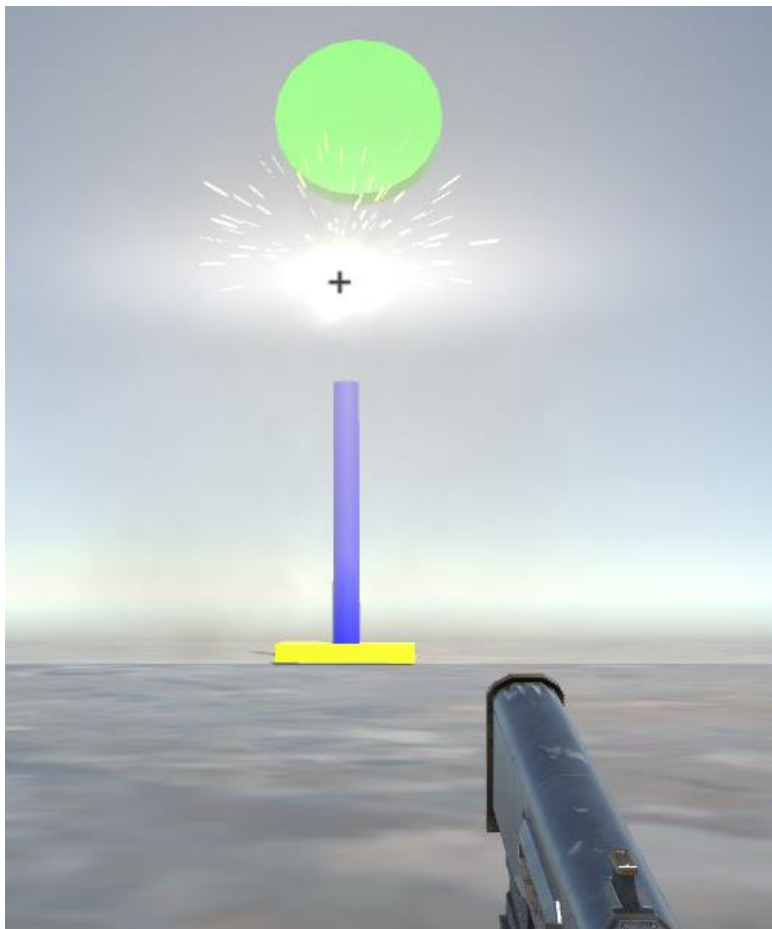
Для взаимодействия с «пулей» к мишени должен быть добавлен компонент **Rigidbody**.



First-Person Shooter (FPS)

(часть 1: оружие и «огонь»)

Теперь, вместо **Plane**, можно добавить вполне полноценный уровень с естественными и реалистичными объектами: либо с использованием **Terrain**, либо с помощью пакета **ProBuilder**.



САМОСТОЯТЕЛЬНО!!!