

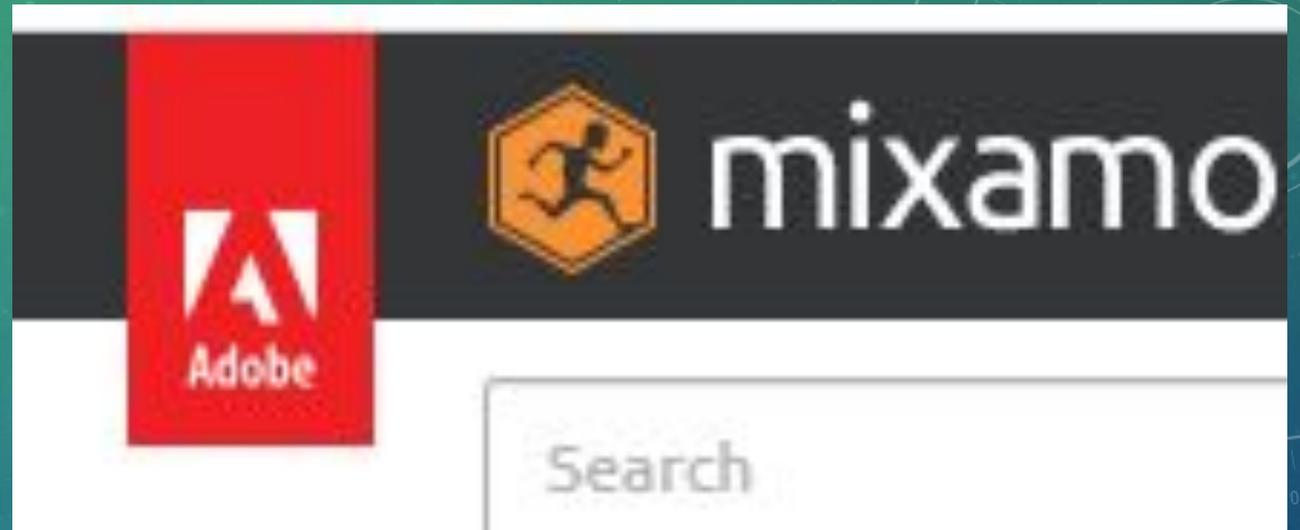


КУРС UNITY 3D

ВЗАИМОДЕЙСТВИЕ С ВРАГОМ. AI ЧАСТЬ

2

НАСТРОЙКА ПРОТИВНИКА



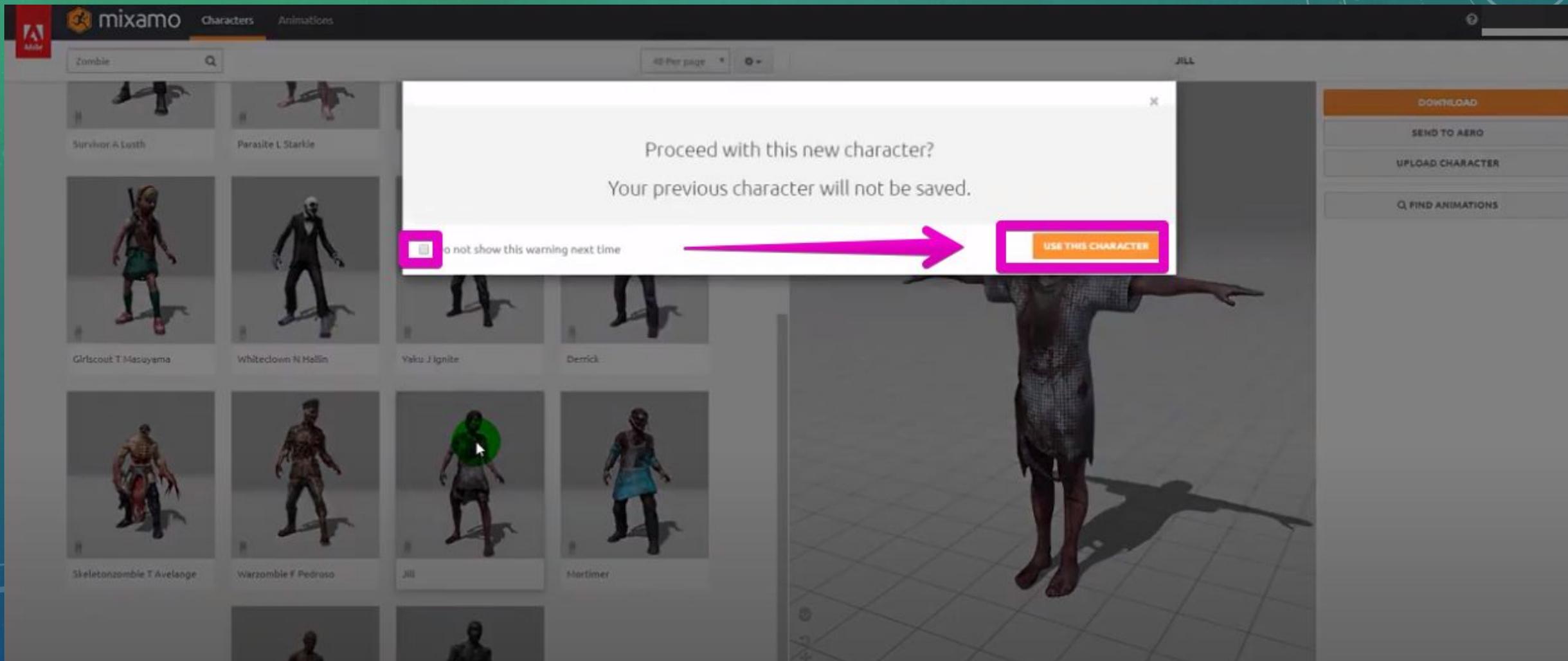
- Переходим на сайт Mixamo, и скачиваем несколько моделей противников. Лучше всего брать зомби, он более менее адаптирован для работы с нашим проектом. На нем мы потренируемся

НАСТРОЙКА ПРОТИВНИКА

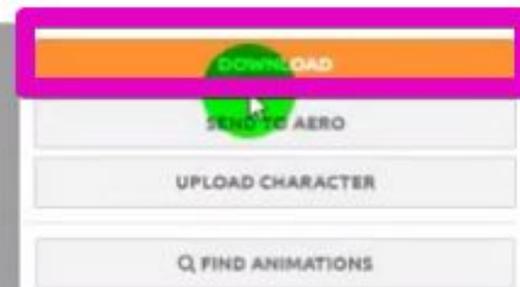
- Как только залогинимся, нам сразу же предстанет окно выбора

The screenshot displays the Mixamo website interface. At the top, there is a navigation bar with the Adobe logo, the Mixamo logo, and links for "Characters" and "Animations". A search bar is located below the navigation bar. The main content area is divided into two sections. On the left, there is a grid of animation thumbnails, each with a small icon of a character in a specific pose and a label below it. The labels include "Grabbing Ammo", "Drawing Gun", "Jumping Down", "Climbing", "Jumping Down", and "Walk To Stop". On the right, there is a large 3D preview window showing a character in a red and black suit performing a "Standing Torch Jump". To the right of the preview window is a configuration panel for the selected animation, titled "Standing Torch Jump". The panel includes several sliders and checkboxes: "Overdrive" (set to 50), "Character Arm-Space" (set to 50), "Trim" (set to 52 total frames), and a "Mirror" checkbox. The interface also features a "DOWNLOAD" button, a "SEND TO AERO" button, and an "UPLOAD CHARACTER" button. The bottom of the interface shows a progress bar indicating the current frame (4/51) and a time display (2:16).

НАСТРОЙКА ПРОТИВНИКА



НАСТРОЙКА ПРОТИВНИК А



НАСТРОЙКА ПРОТИВНИКА

DOWNLOAD SETTINGS

Format

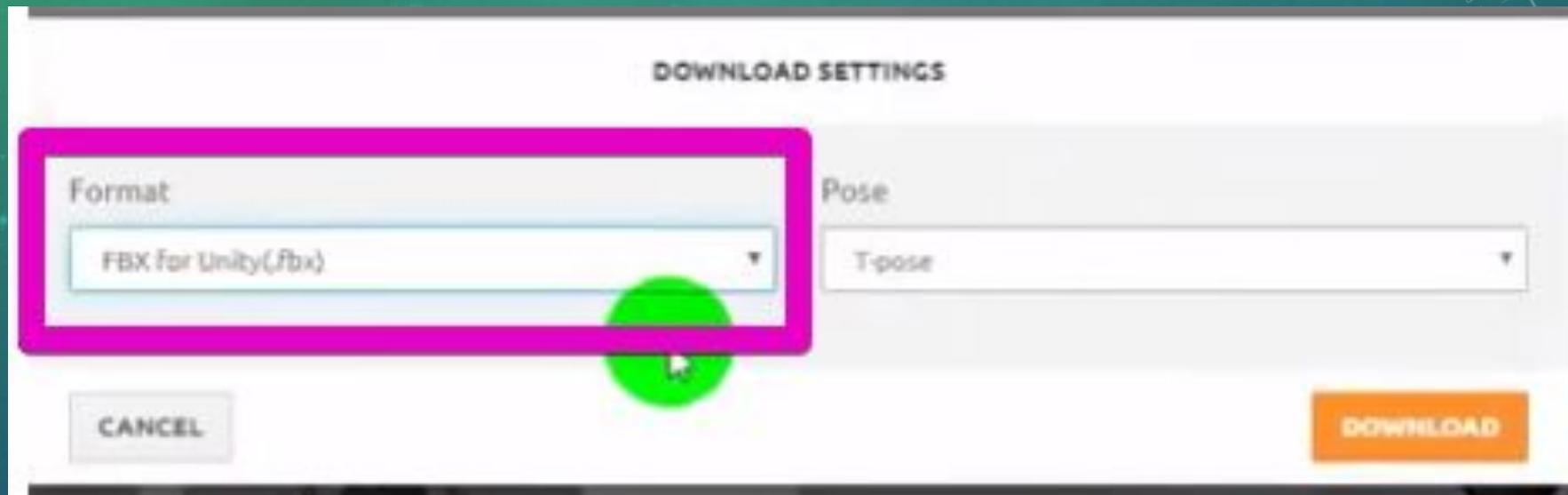
FBX for Unity(.fbx)

Pose

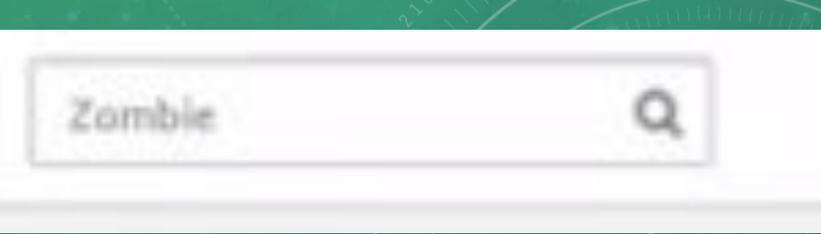
T-pose

CANCEL

DOWNLOAD



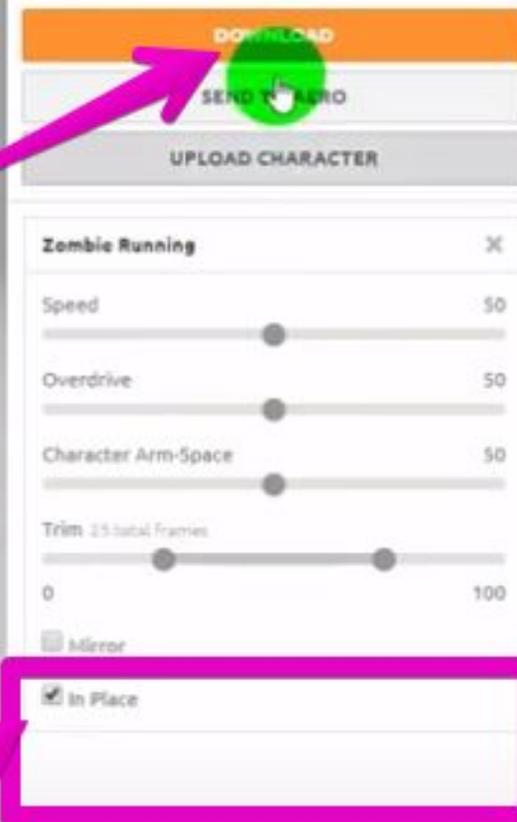
НАСТРОЙКА ПРОТИВНИКА



Zombie Running

После чего скачиваем

Обязательно
ставим
галочку в чек
бокс



НАСТРОЙКА ПРОТИВНИКА

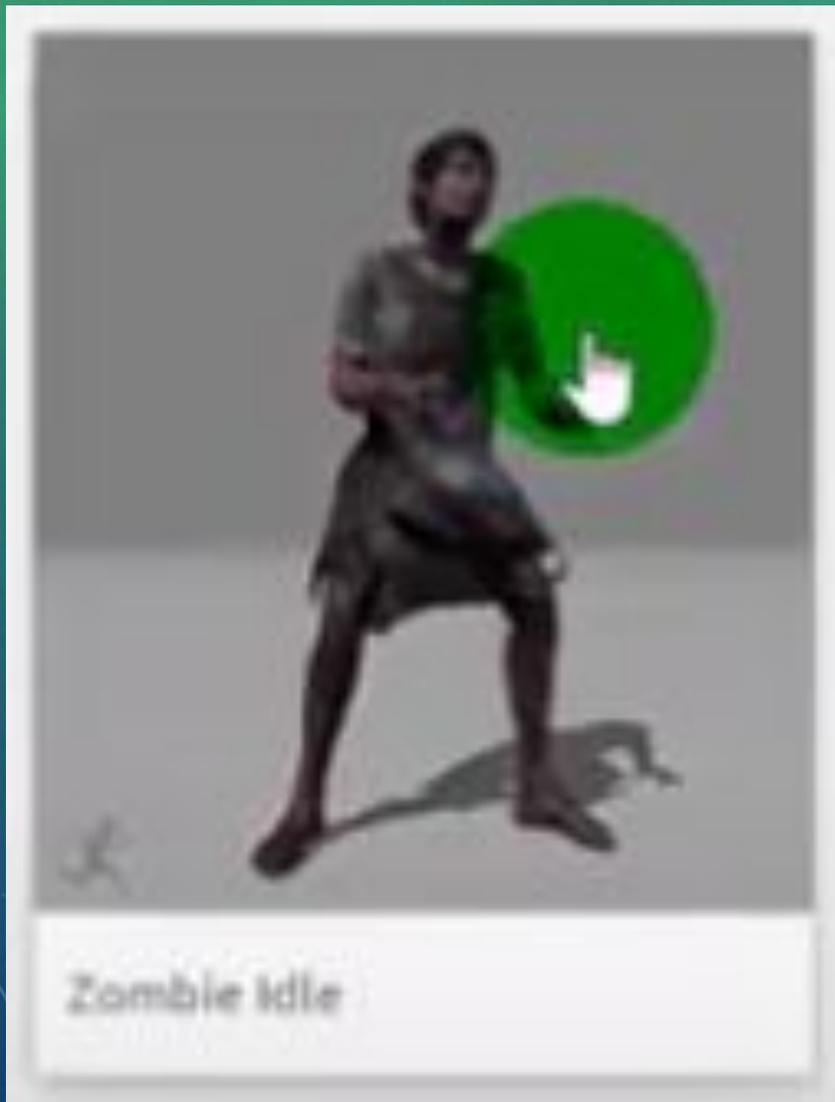
DOWNLOAD SETTINGS

Format:	Skin:
FBX for Unity(.fbx) ▼	Without Skin ▼
Frames per Second:	Keyframe Reduction:
30 ▼	none ▼

CANCEL **DOWNLOAD**



НАСТРОЙКА ПРОТИВНИКА



DOWNLOAD

SEND TO AERO

UPLOAD CHARACTER

Zombie Idle X

Overdrive 50

Character Arm-Space 50

Trim 131 total frames

0 100

Mirror

A control panel for the character. It features an orange 'DOWNLOAD' button, a green 'SEND TO AERO' button with a hand cursor, and a grey 'UPLOAD CHARACTER' button. Below these are three sliders: 'Overdrive' (set to 50), 'Character Arm-Space' (set to 50), and 'Trim' (set to 131 total frames). A 'Mirror' checkbox is also present.

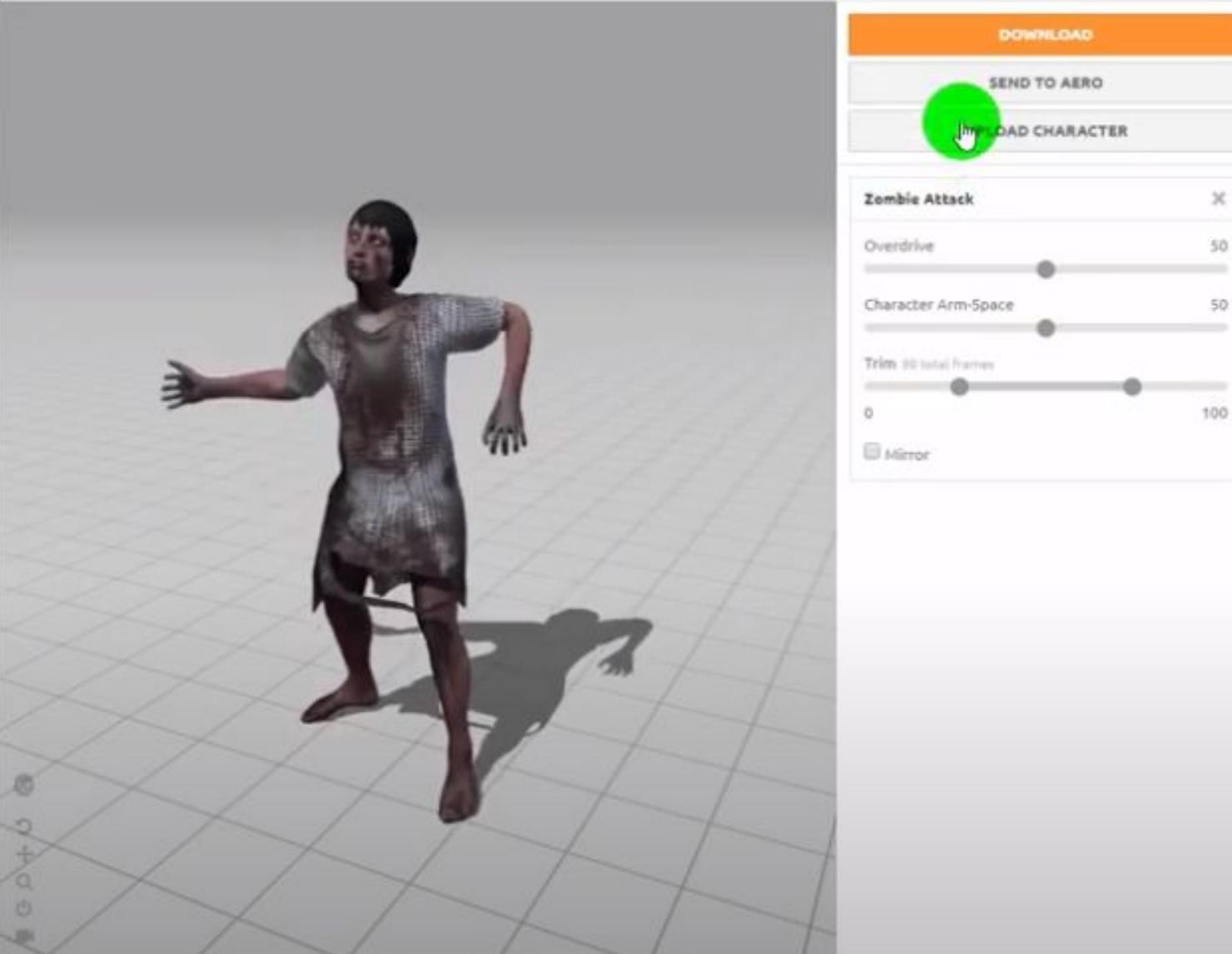
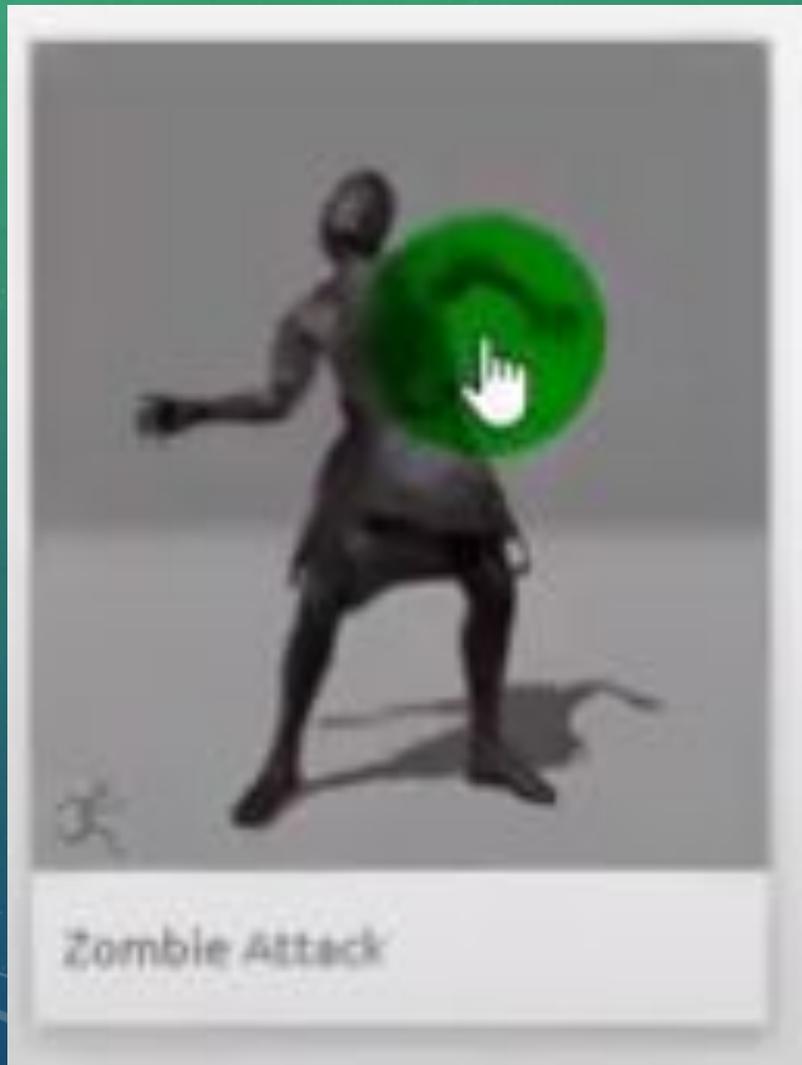
НАСТРОЙКА ПРОТИВНИКА

DOWNLOAD SETTINGS

Format	Skin
FBX for Unity(.fbx) ▼	Without Skin ▼
Frames per Second	Keyframe Reduction
30 ▼	none ▼

CANCEL **DOWNLOAD**

НАСТРОЙКА ПРОТИВНИКА



DOWNLOAD

SEND TO AERO

LOAD CHARACTER

Zombie Attack X

Overdrive 50

Character Arm-Space 50

Trim 30 total frames

0 100

Mirror

A main interface showing a 3D model of the zombie character in a light-colored, textured outfit. To the right of the model is a settings panel. The panel has an orange "DOWNLOAD" button at the top, followed by "SEND TO AERO" and "LOAD CHARACTER" buttons. Below these are three sliders: "Overdrive" (set to 50), "Character Arm-Space" (set to 50), and "Trim 30 total frames" (set to 30). At the bottom of the panel is a "Mirror" checkbox.

НАСТРОЙКА ПРОТИВНИКА

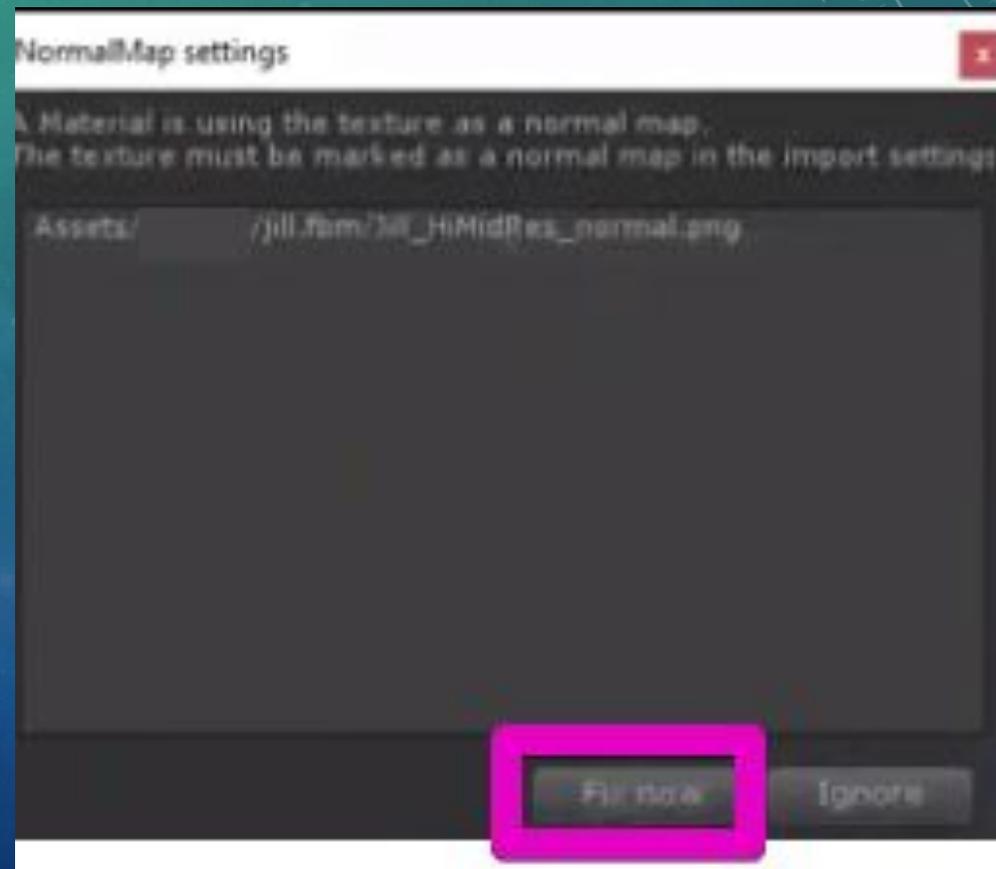
DOWNLOAD SETTINGS

Format	Skin
FBX for Unity(.fbx) ▼	Without Skin ▼
Frames per Second	Keyframe Reduction
30 ▼	none ▼

CANCEL **DOWNLOAD**

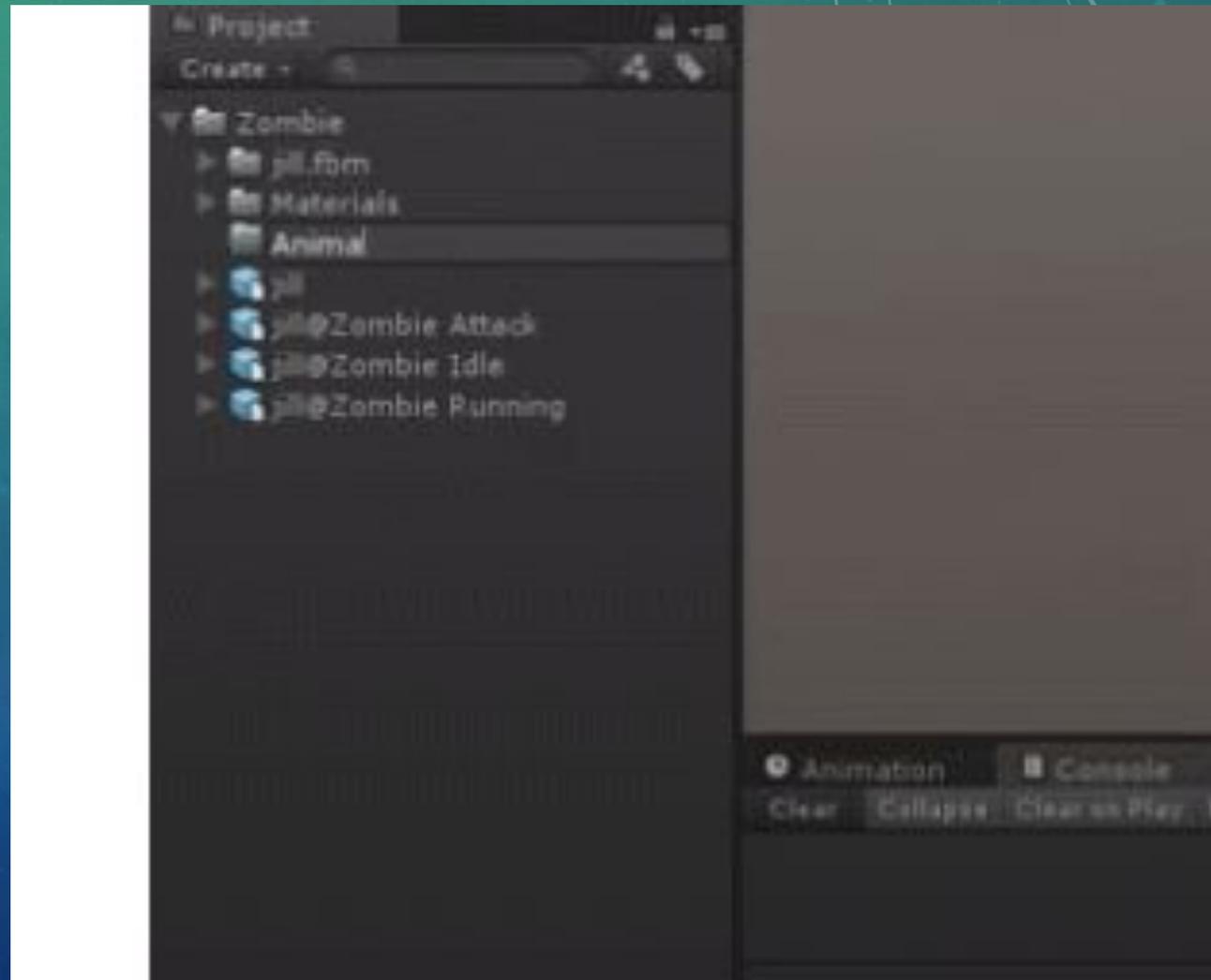
НАСТРОЙКА ПРОТИВНИКА

- Переносим все, что скачали, в папку Prefabs. На появившееся сообщение нажимаем Fix Now

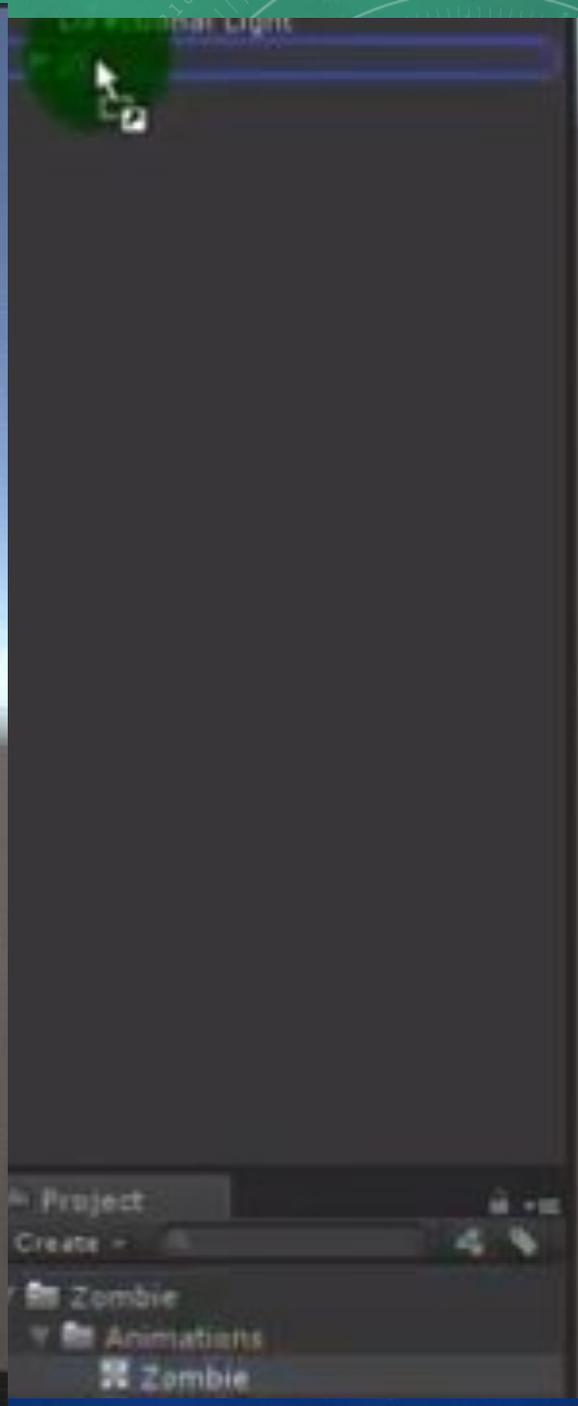
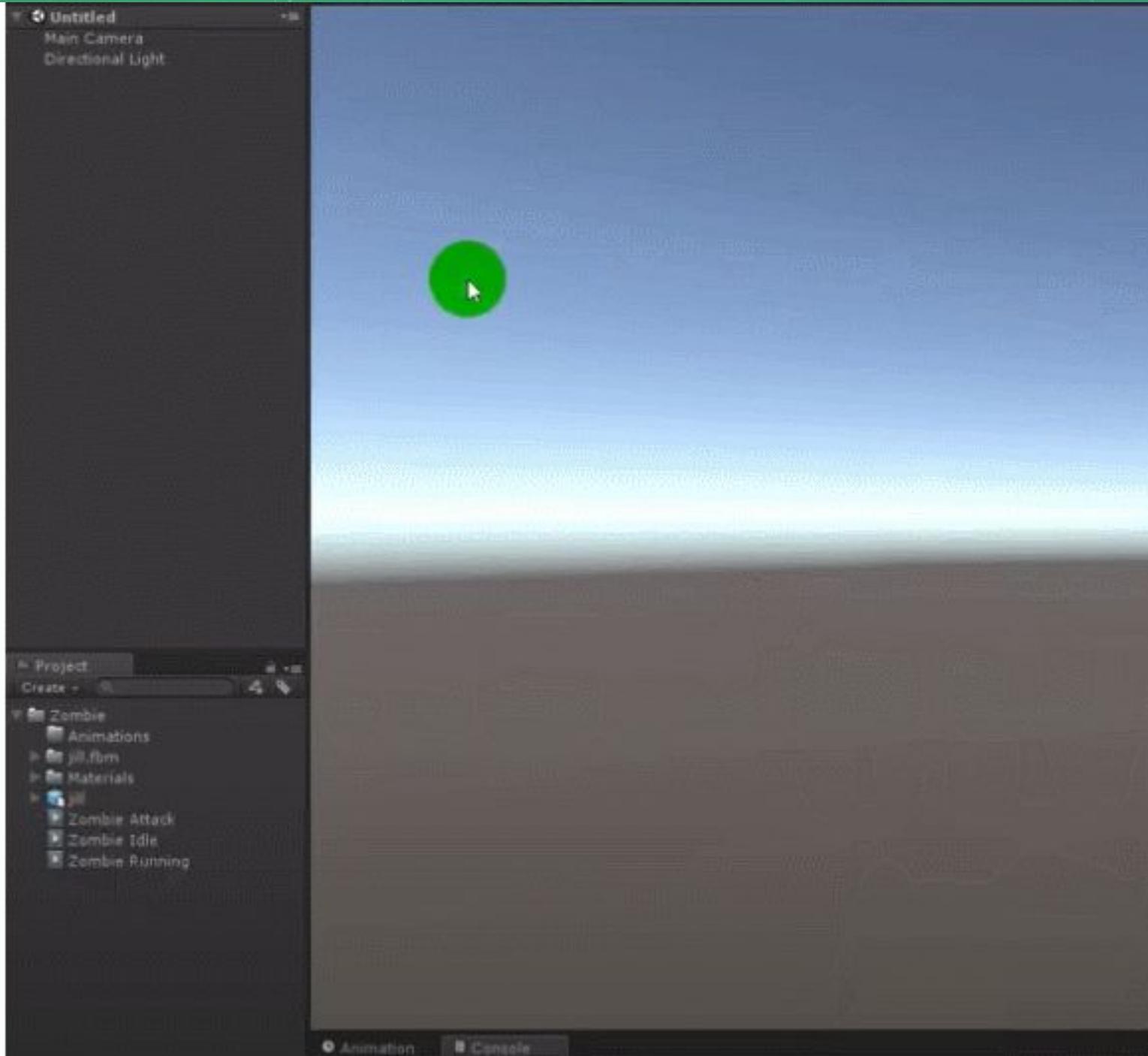


НАСТРОЙКА ПРОТИВНИКА

- Далее создаем папку Animation, заходим в prefabs, куда закидывали наши анимации, выделяем через CTRL нужные анимации (idle, running, attack) и нажимаем CTRL + D. Все остальное из оставшегося - удаляем



НАСТРОЙКА ПРОТИВНИКА



С ПРОШЛОГО ЗАНЯТИЯ – ОСТАНОВИЛИСЬ НА АНИМАЦИИ

Выбираем тип переменной bool

```
graph TD; Idle[Idle] --> Run[Run]; Idle --> Attack[Attack]; Run --> Idle; Attack --> Idle; Run --> Attack; Attack --> Run;
```

Animation/Zombie_controller

С
ПРОШЛОГ
О
ЗАНЯТИЯ
—
ОСТАНОВ
ИЛИСЬ НА
АНИМАЦИ
И

Any State

Exit

Entry

Не забываем!
1. ПКМ - Make transition
для установления связей

2. Вначале от IDLE - RUN - ATTACK
Потом ATTACK - RUN - IDLE

3. В последний момент ATTACK - IDLE
IDLE - ATTACK

Run

Make Transition

Set as Layer Default State

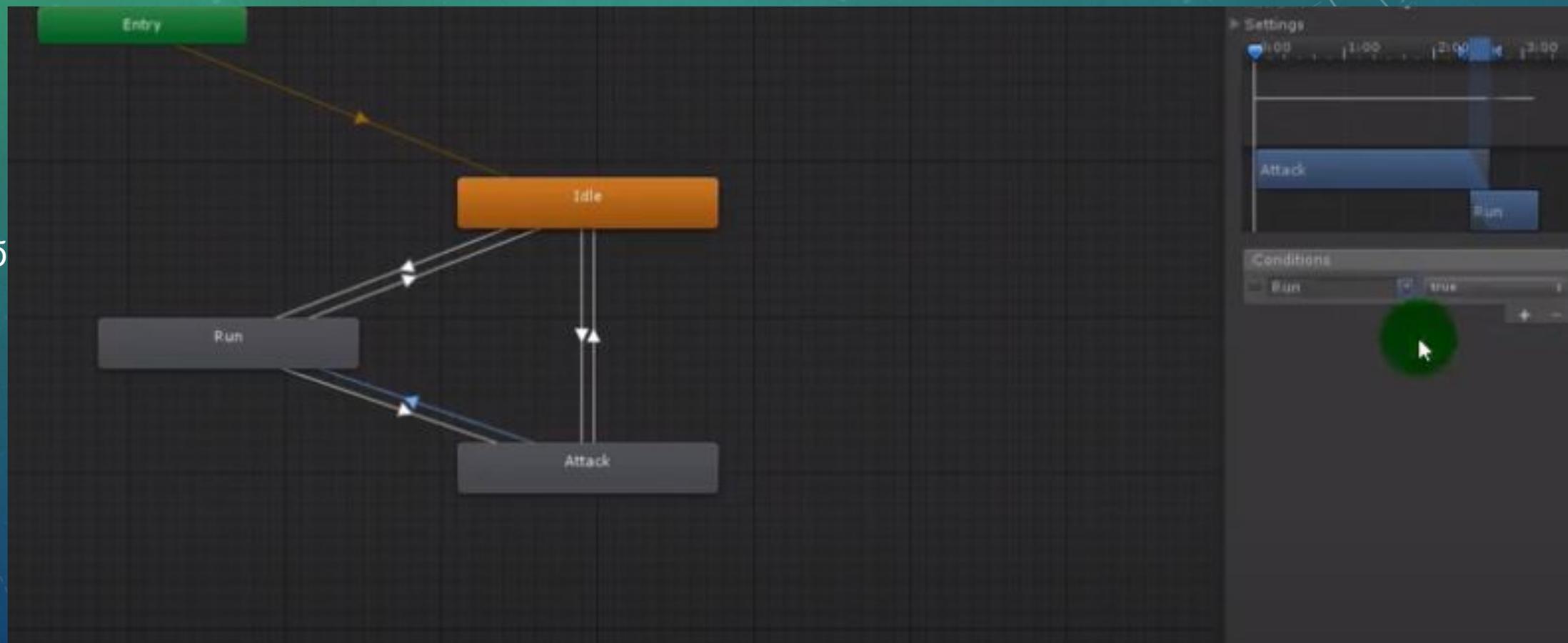
Create new BlendTree in State

Delete

Attack

НАСТРОЙКА ПРОТИВНИКА

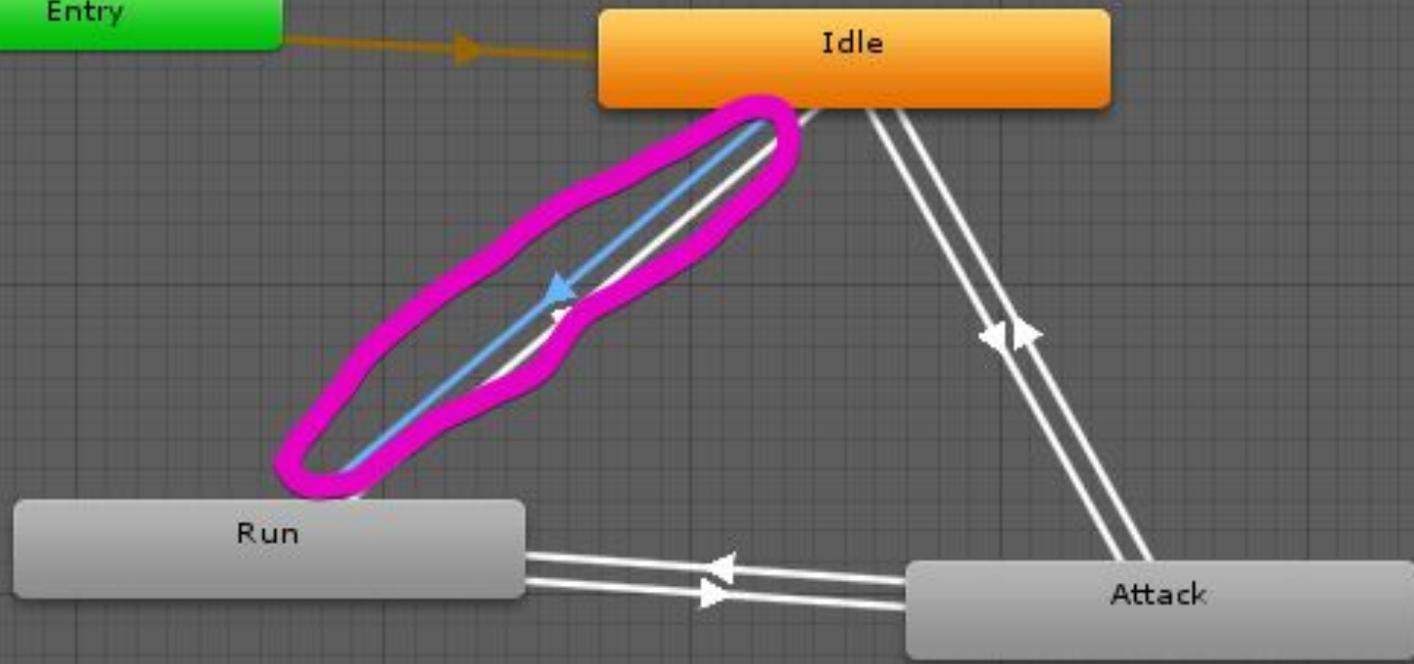
- Не заб



Any State

Exit

Entry

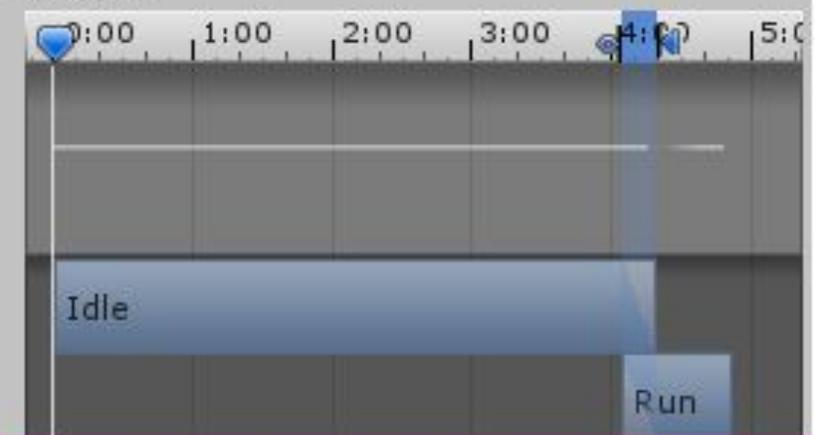


Idle -> Run
1 AnimatorTransitionBase

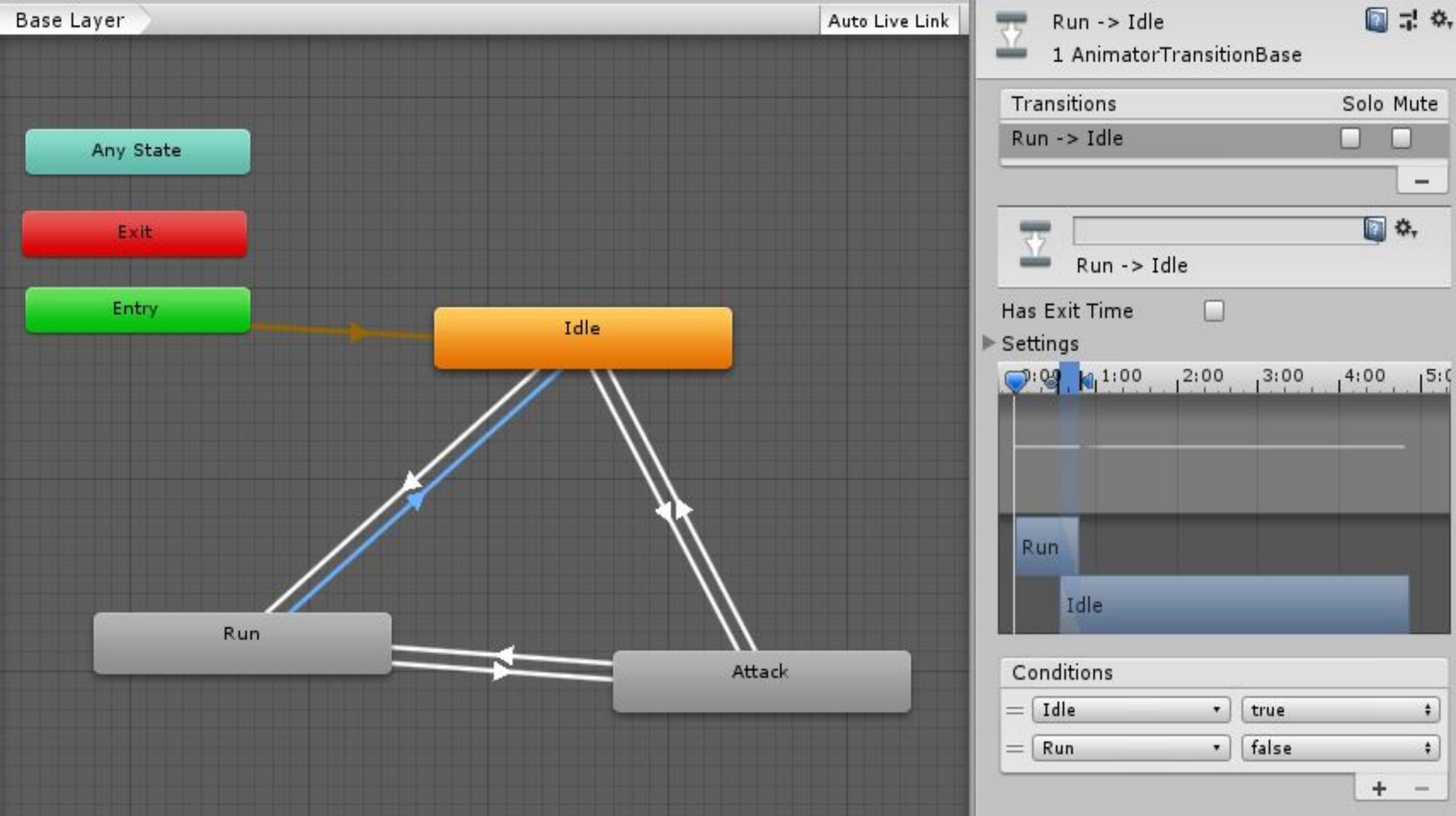
Transitions Solo Mute
Idle -> Run

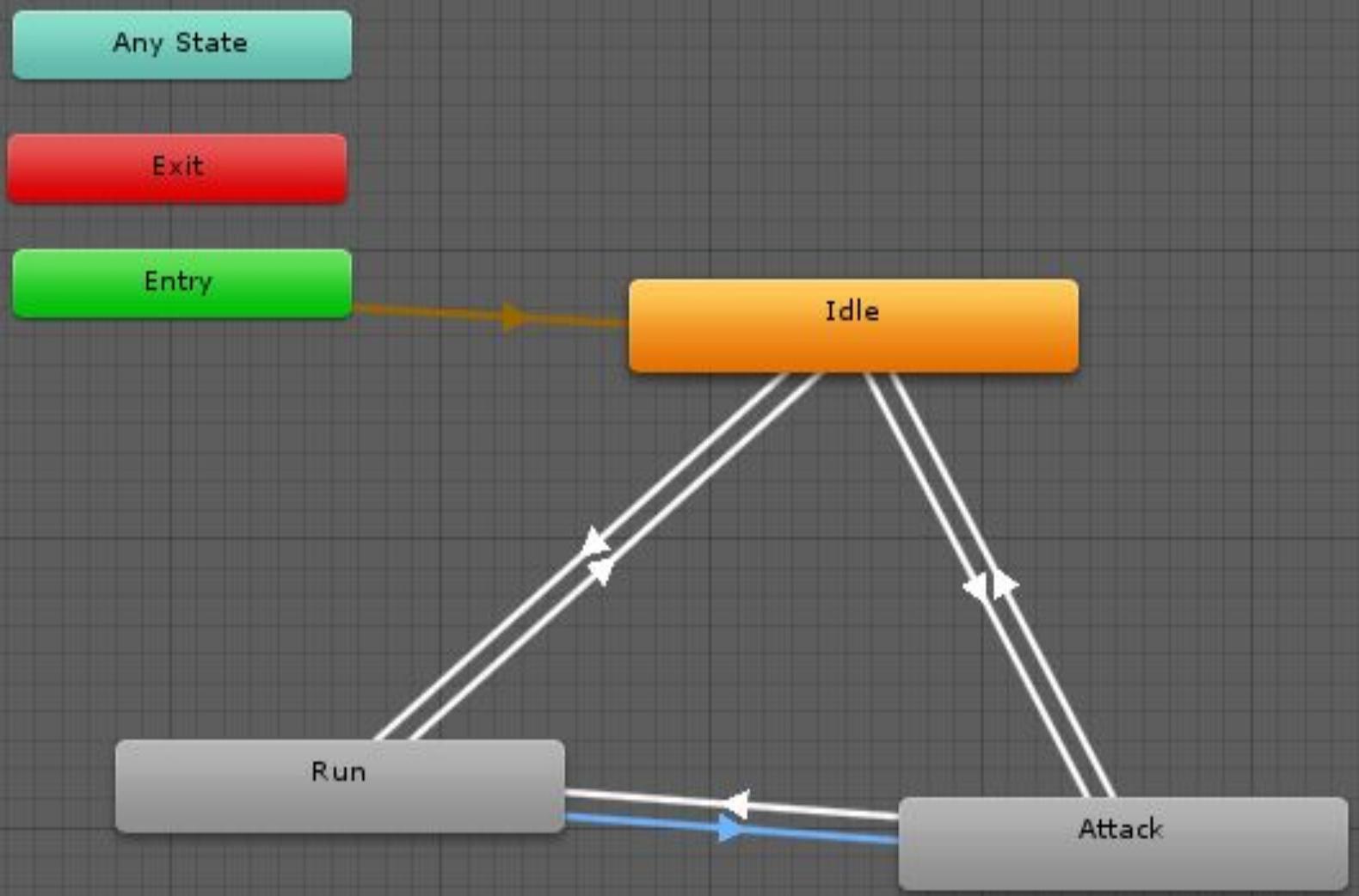
Idle -> Run

Has Exit Time
Settings



Conditions	
= Idle	false
= Run	true





Run -> Attack ? [] [] []
1 AnimatorTransitionBase

Transitions Solo Mute

Run -> Attack	<input type="checkbox"/>	<input type="checkbox"/>
---------------	--------------------------	--------------------------

Run -> Attack ? [] []

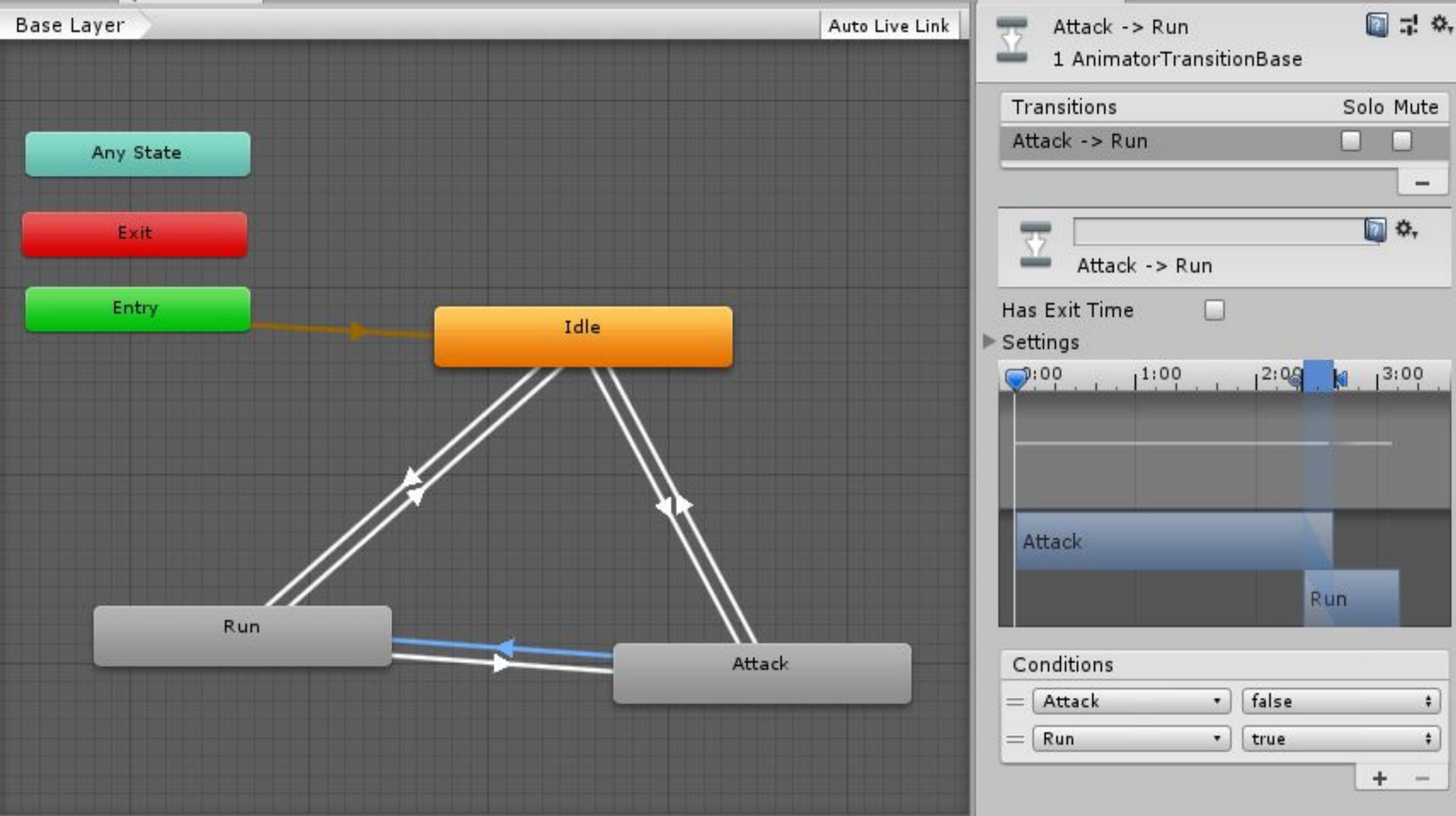
Has Exit Time

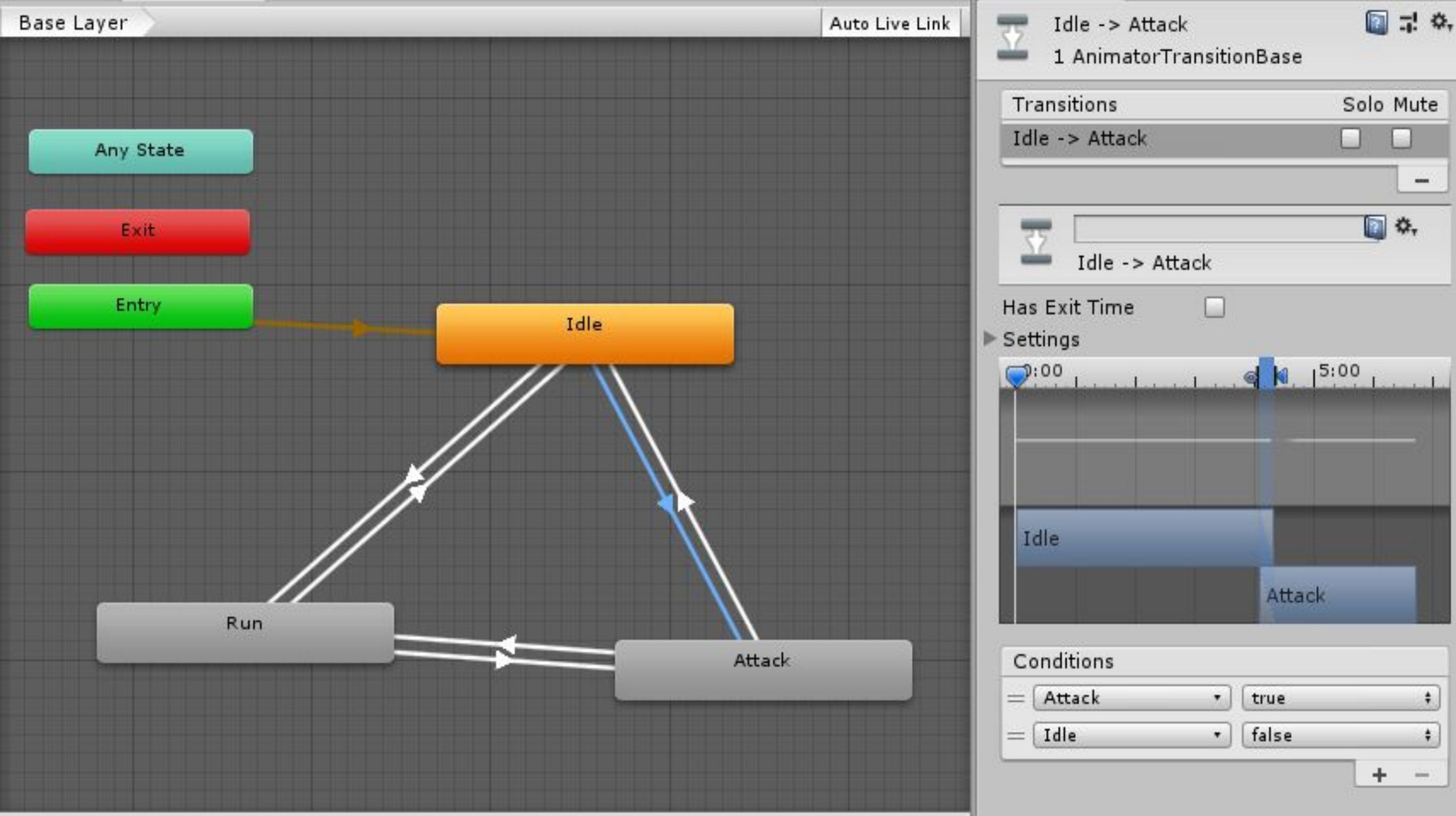
Settings

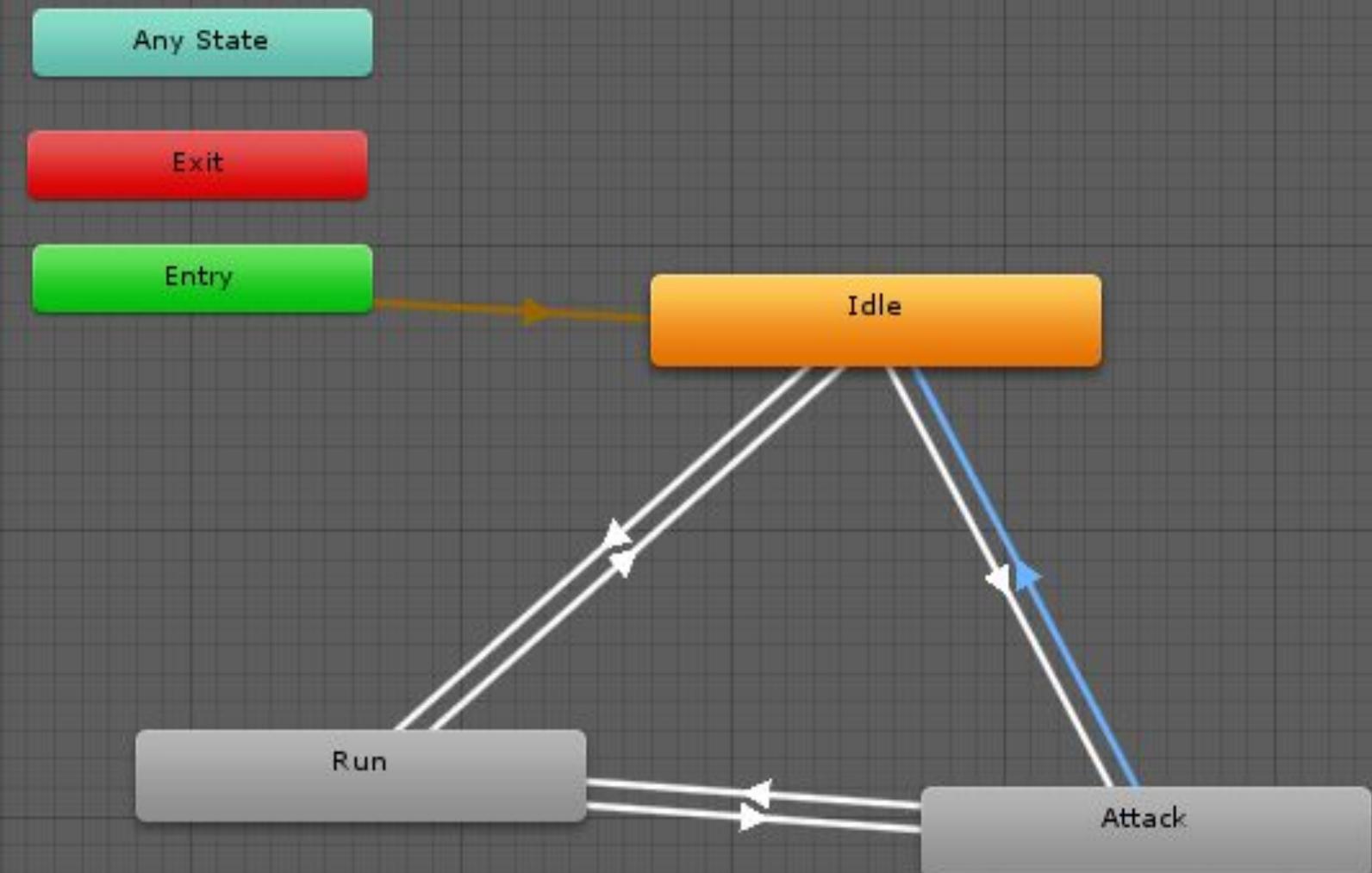
Conditions

=	Run	▼	false	⬆
=	Attack	▼	true	⬆

+ -







Transitions Solo Mute

Attack -> Idle

Attack -> Idle

Has Exit Time

Settings



Conditions

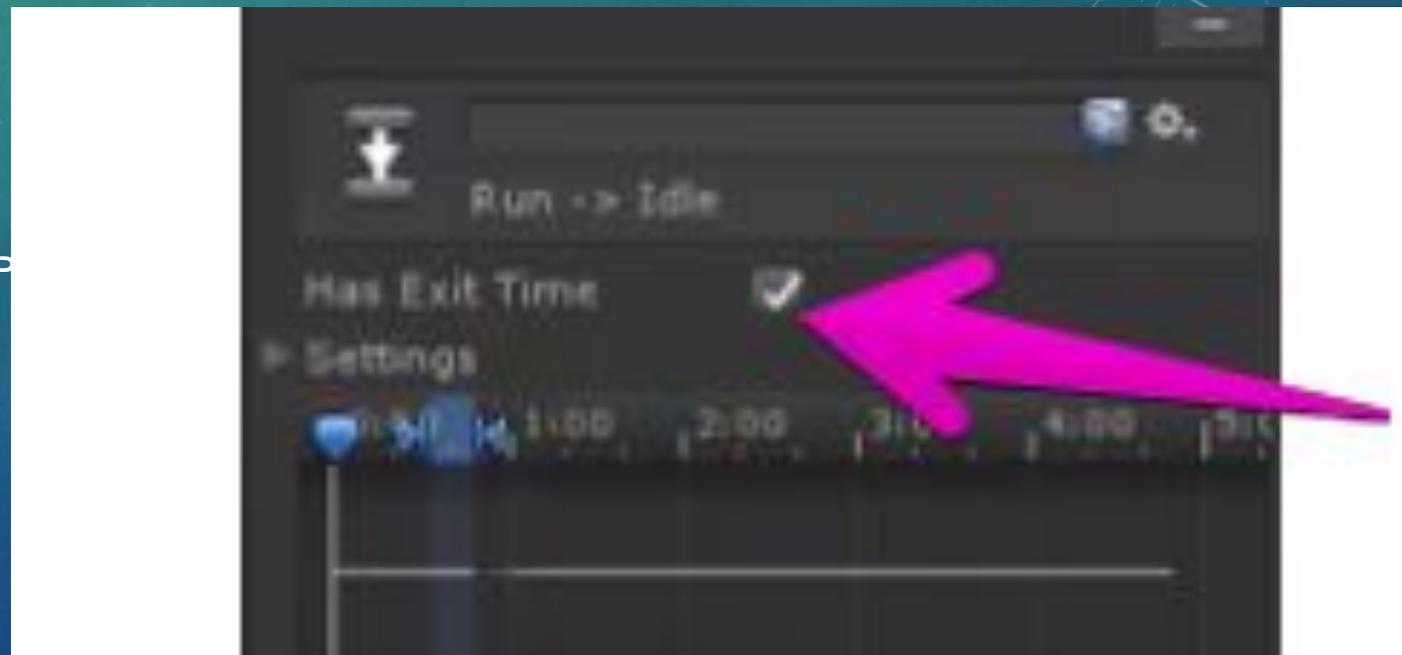
= Idle true

= Attack false

+ -

НАСТРОЙКА ПРОТИВНИКА

- Убираем галку Has Exit Time – для прерыв



КОД ДЛЯ АНИМАЦИИ ВРАГА

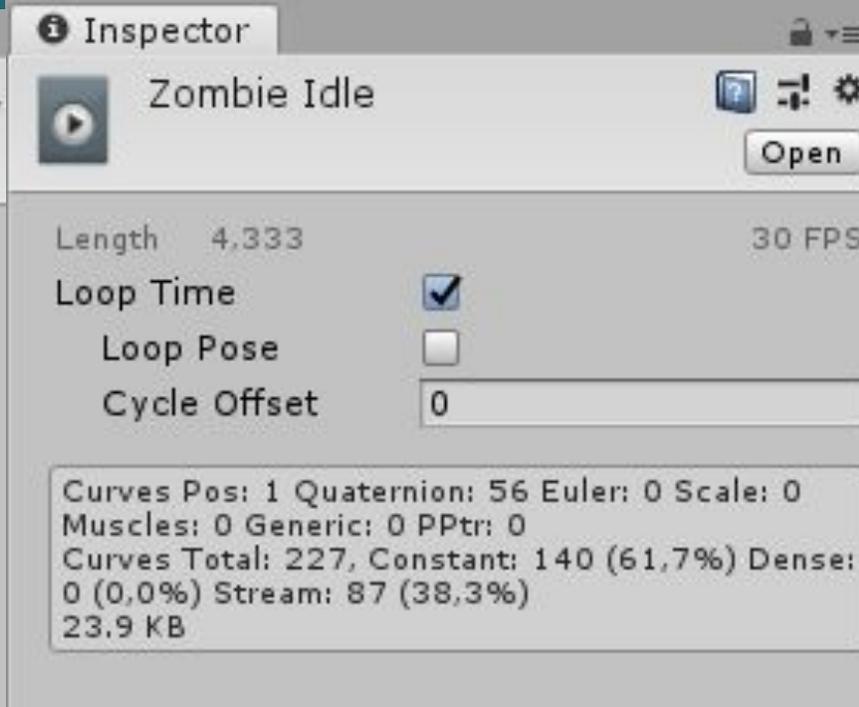
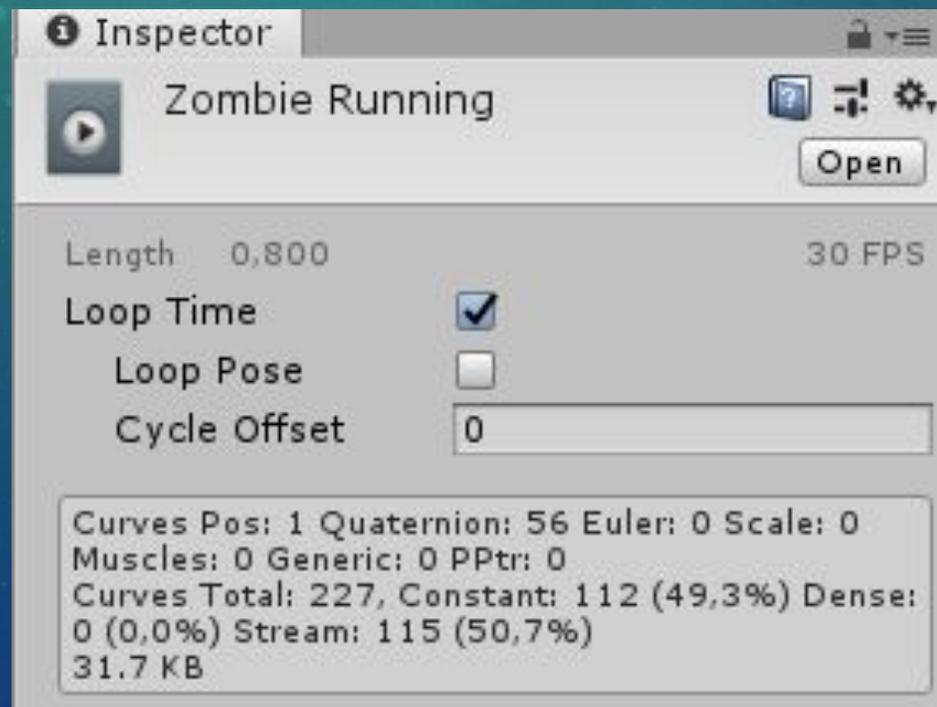
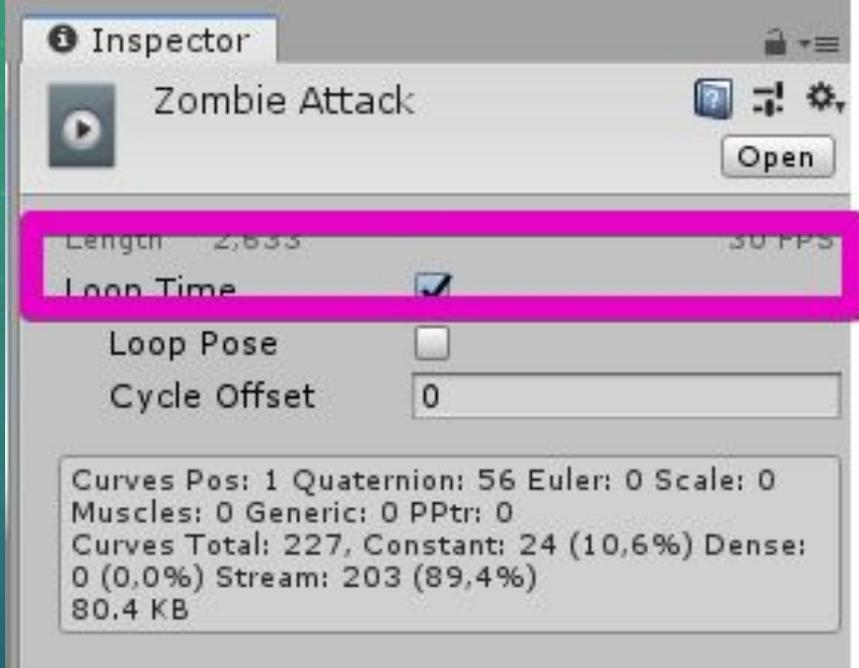
- Создаем новый скрипт, вешаем на врага и открываем в визуал студио

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.AI;
5
6 [RequireComponent (typeof(NavMeshAgent))]
7 public class Enemy : MonoBehaviour {
8
9     public Transform target;
10    float distance;
11    NavMeshAgent myAgent;
12    Animator myAnim;
13
14    void Start ()
15    {
16        myAnim = GetComponent <Animator> ();
17        myAgent = GetComponent <NavMeshAgent> ();
18    }
19
```

```
20 void Update ()
21 {
22     distance = Vector3.Distance (transform.position, target.transform.position);
23     if (distance > 10) {
24         myAgent.enabled = false;
25         myAnim.SetBool ("Idle",true);
26         myAnim.SetBool ("Run",false);
27         myAnim.SetBool ("Attack",false);
28     }
29
30     if (distance <= 10 & distance > 1.5f) {
31         myAgent.enabled = true;
32         myAgent.SetDestination (target.position);
33         myAnim.SetBool ("Idle",false);
34         myAnim.SetBool ("Run",true);
35         myAnim.SetBool ("Attack",false);
36     }
37
38     if (distance <= 1.5f) {
39         myAgent.enabled = false;
40         myAnim.SetBool ("Idle",false);
41         myAnim.SetBool ("Run",false);
42         myAnim.SetBool ("Attack",true);
43     }
44 }
```

КОД ДЛЯ АНИМАЦИИ ВРАГА

- Не забываем настроить Loop Time в каждой из анимаций!



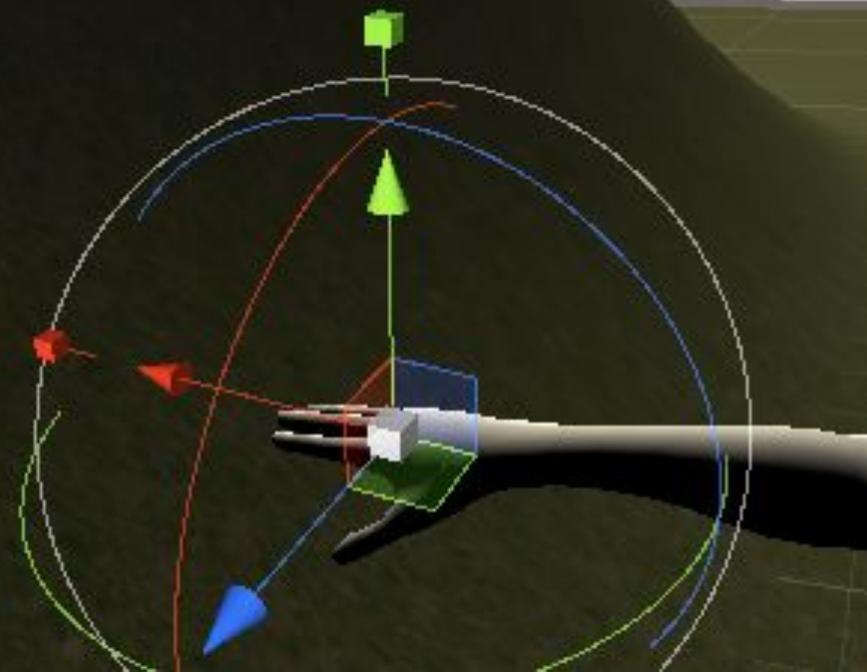
СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

- После того, как мы с вами создали анимацию, необходимо будет заскриптовать врага на отъём жизни, а также создать возможность нанести ответный урон

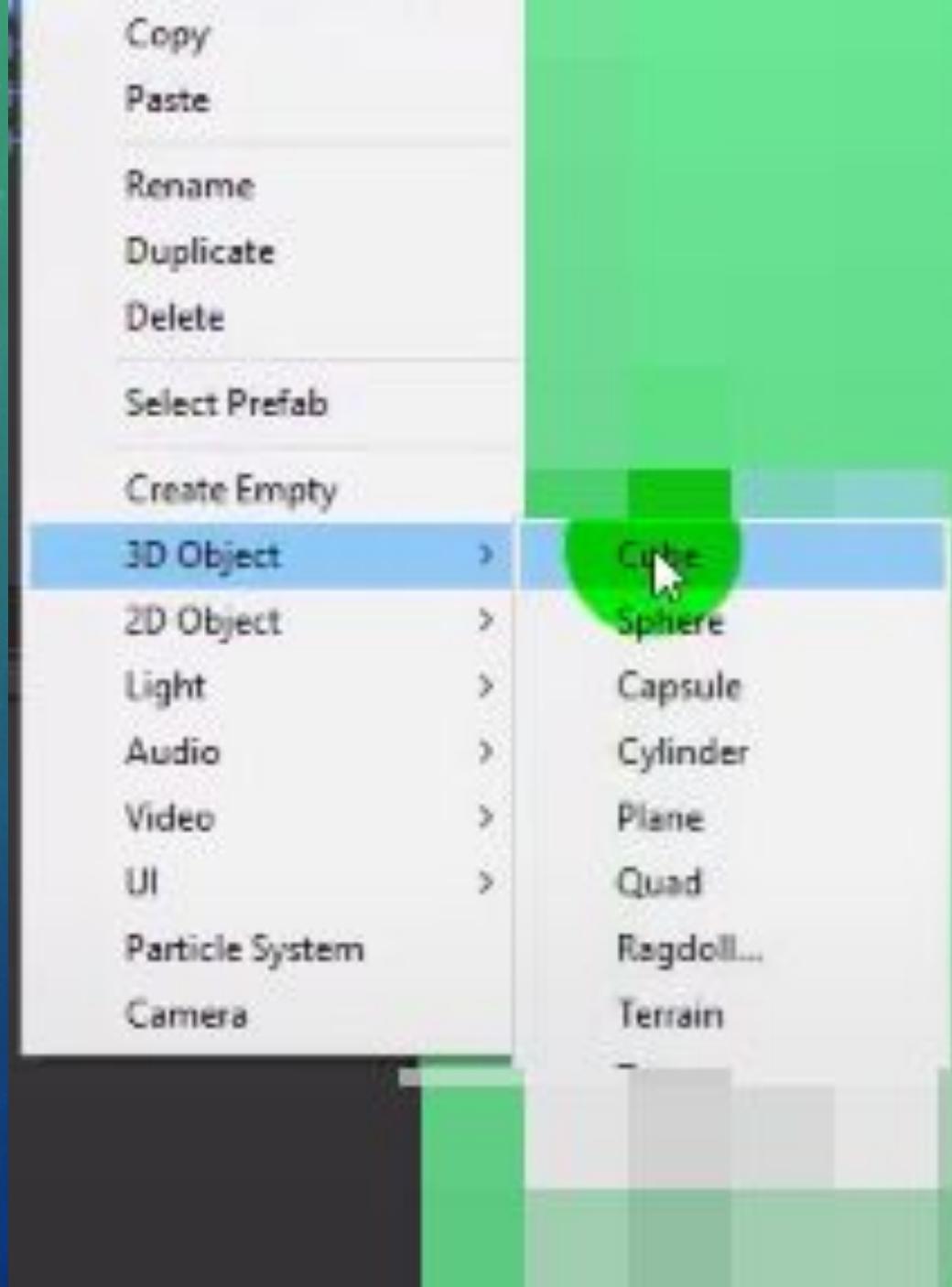
СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

Вначале создадим полосу жизни самому врагу, поскольку нам будет необходимо отнимать жизни в обе стороны.

СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

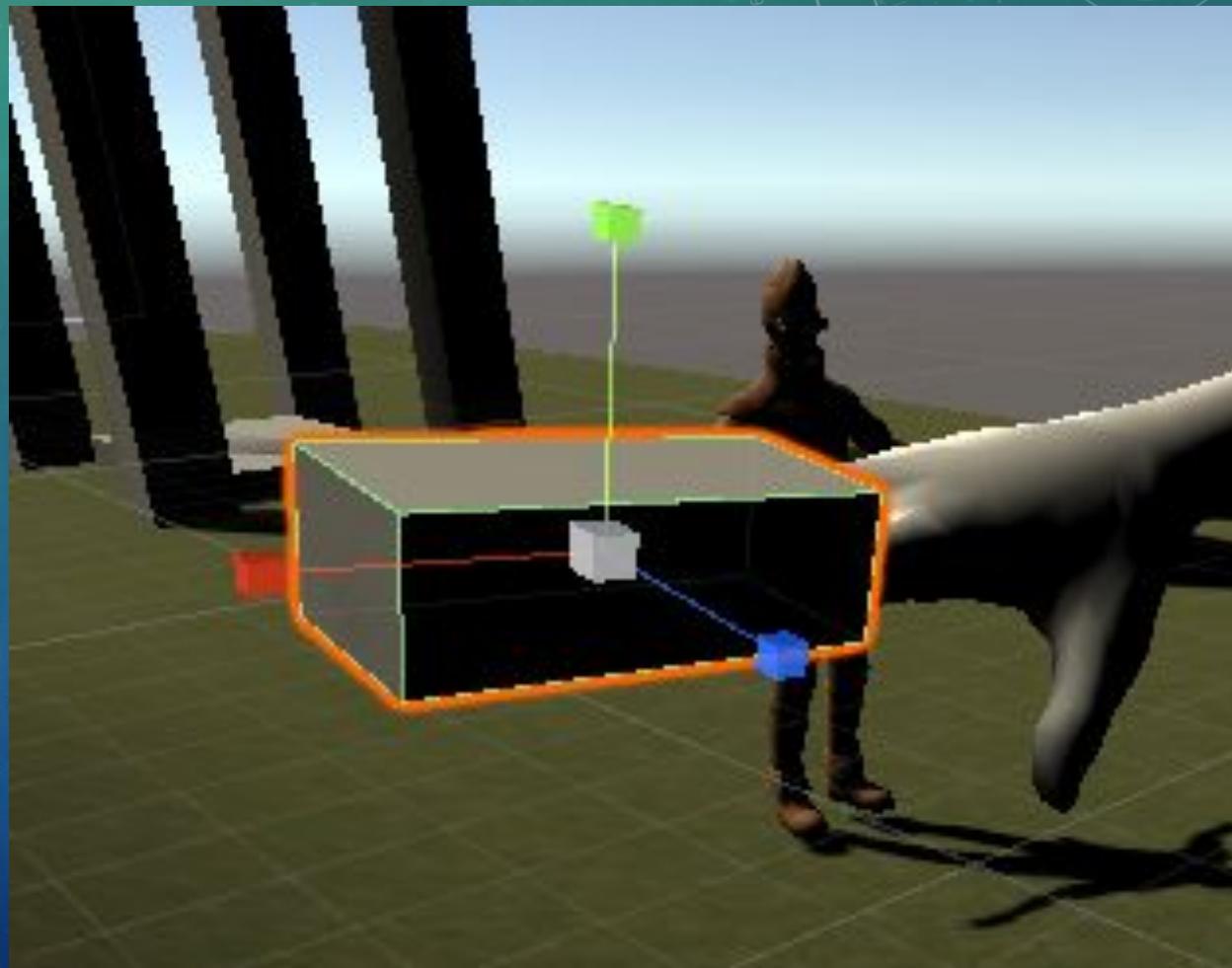


СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ



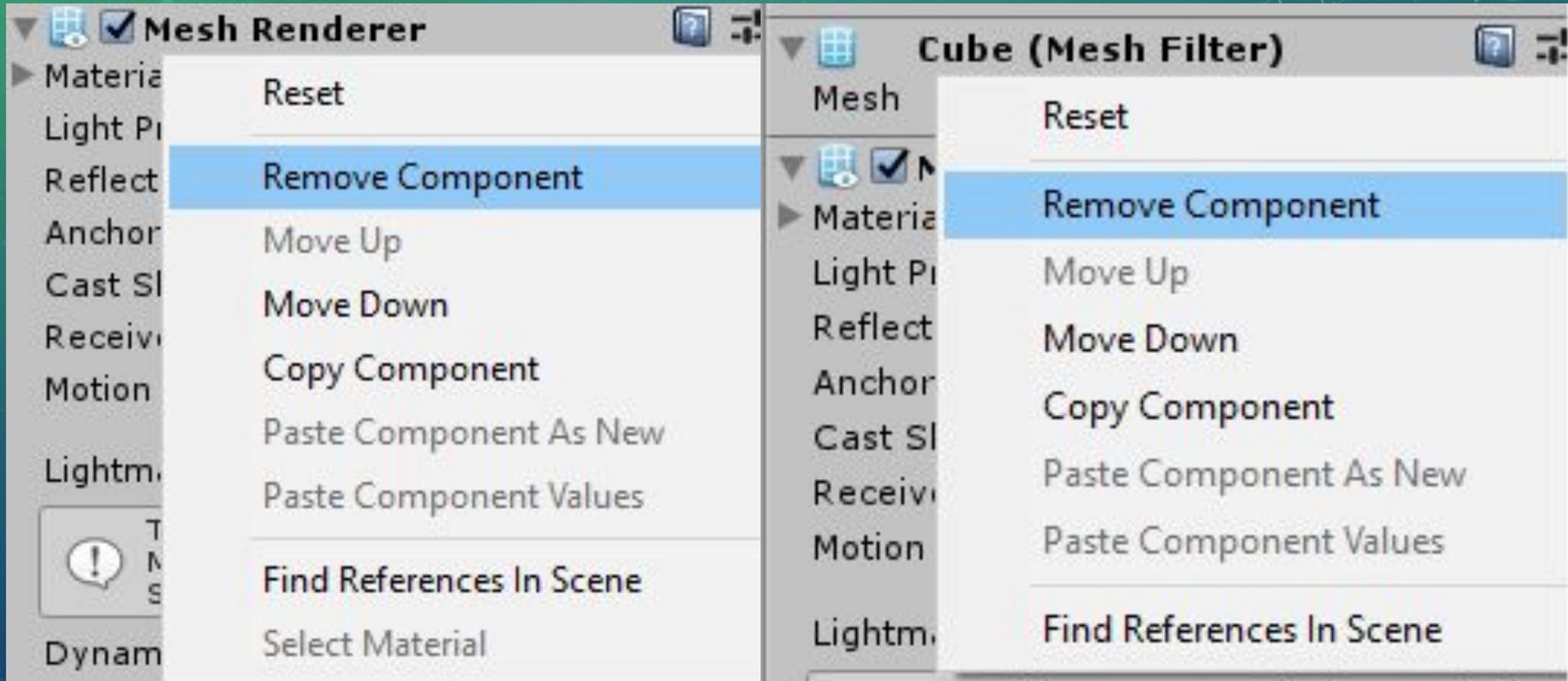
СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

- Уменьшаем примерно до вот такого размера



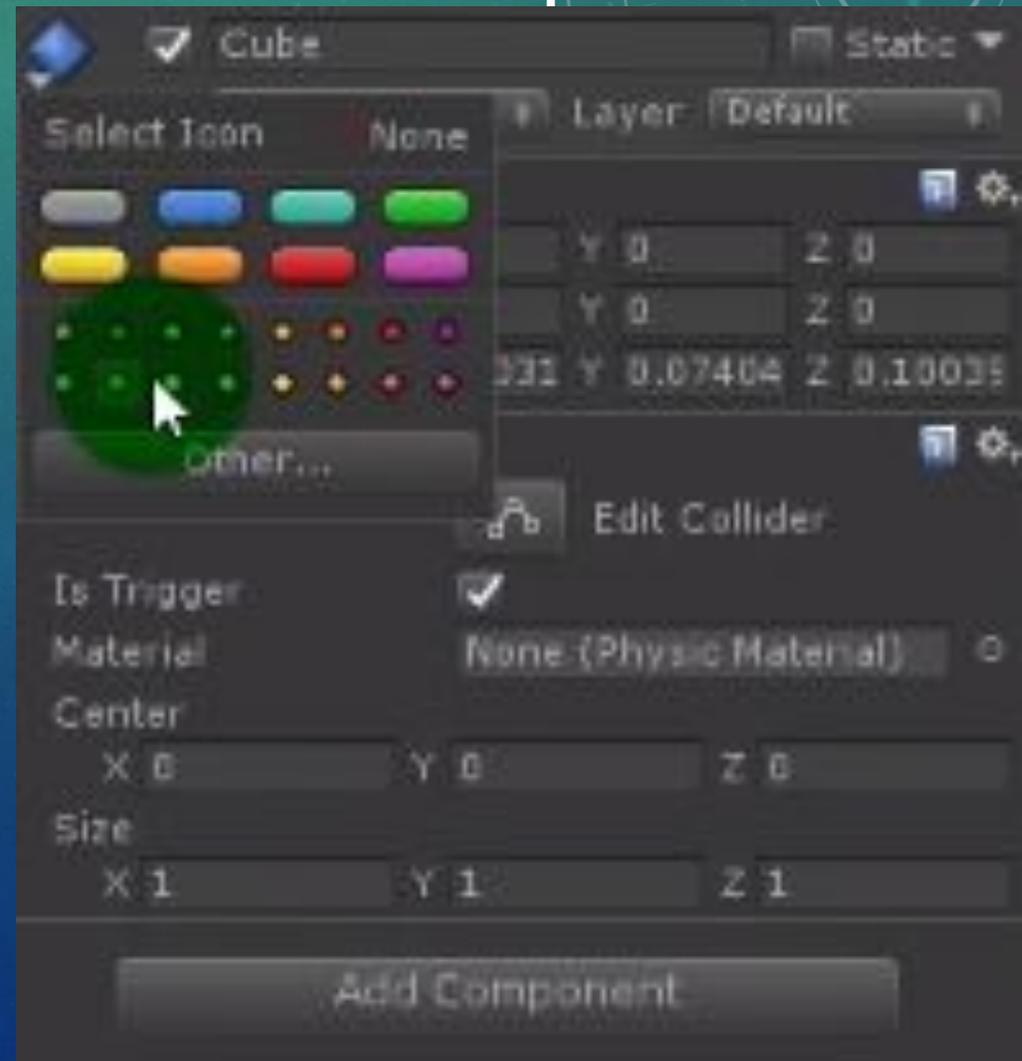
СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

- Mesh фильтр и mesh renderer куба сразу же удалим

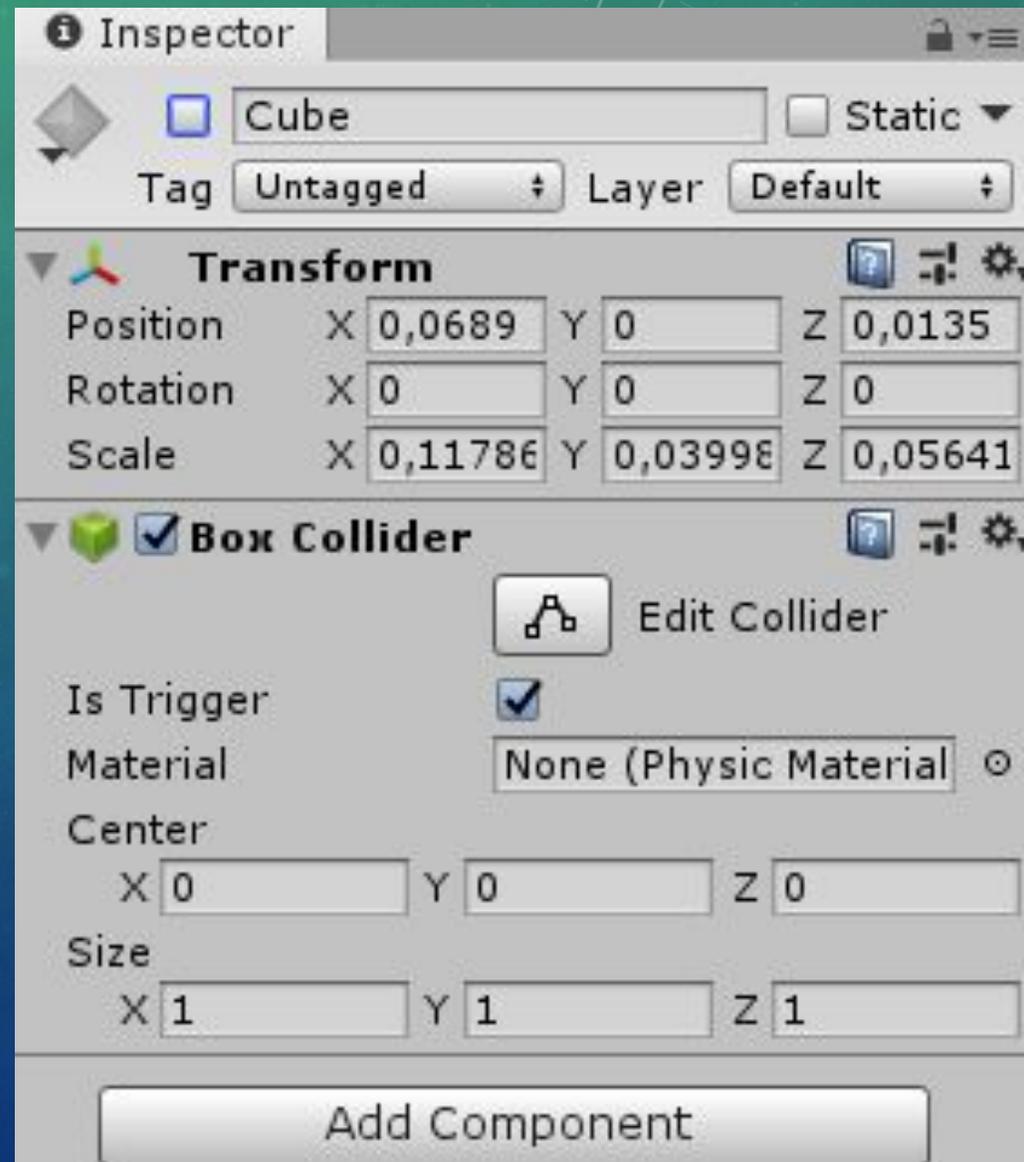


СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

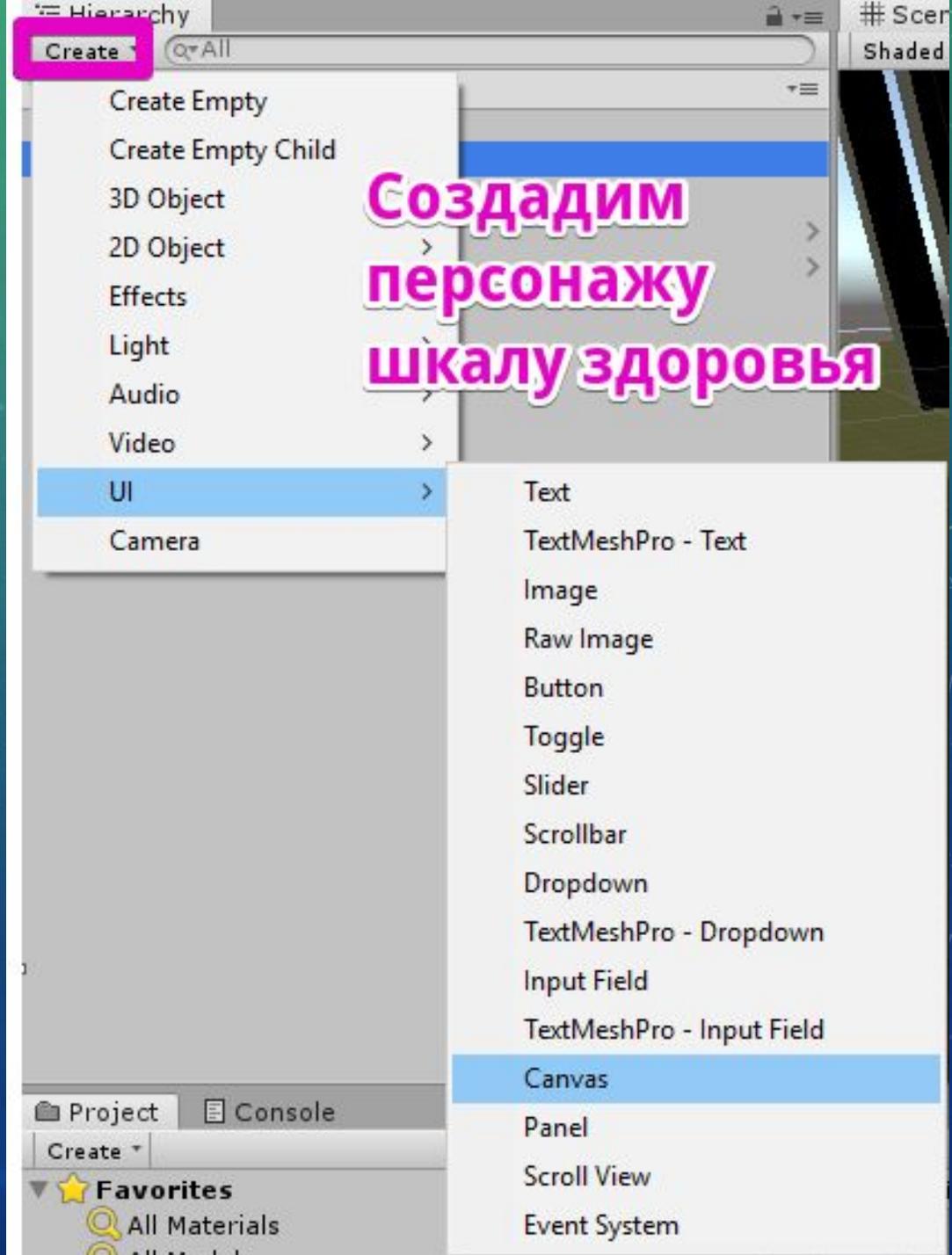
- Ставим галочку «IsTrigger»
нашему BoxCollider, выберем
иконку кубу и выключим
данный объект



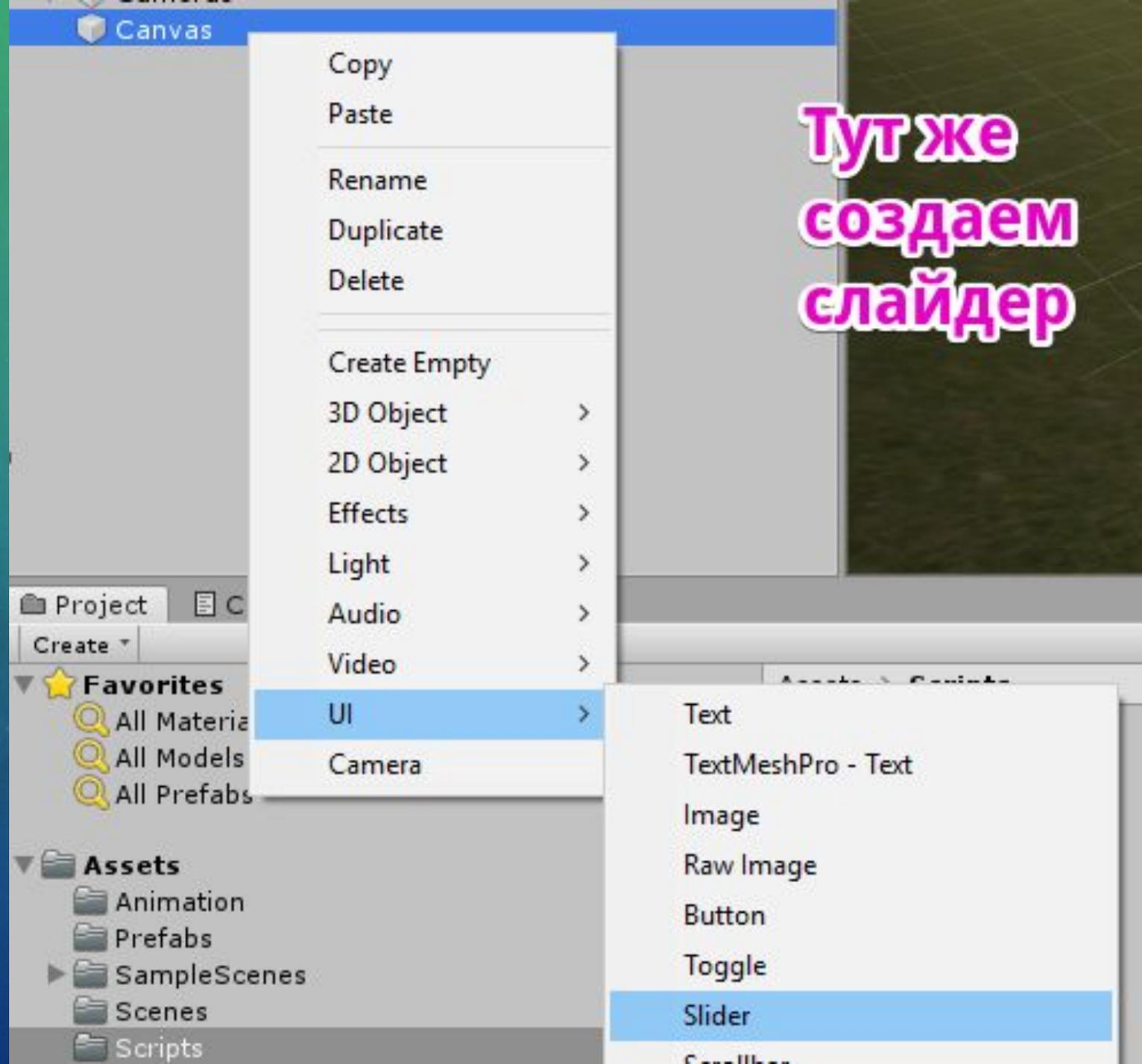
СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ



СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ



СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ



Тут же
создаем
слайдер

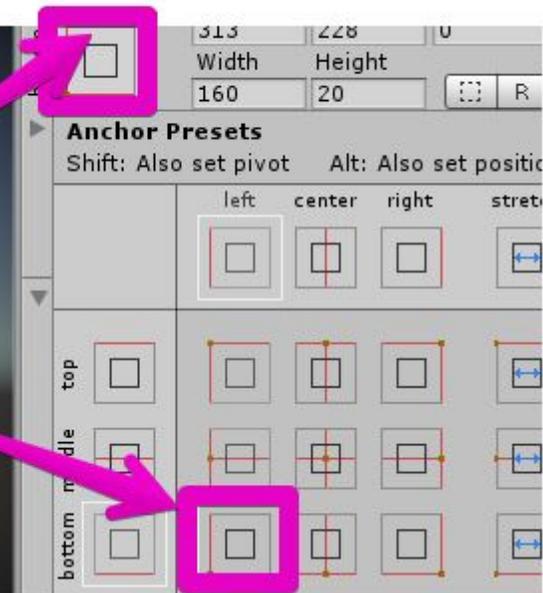
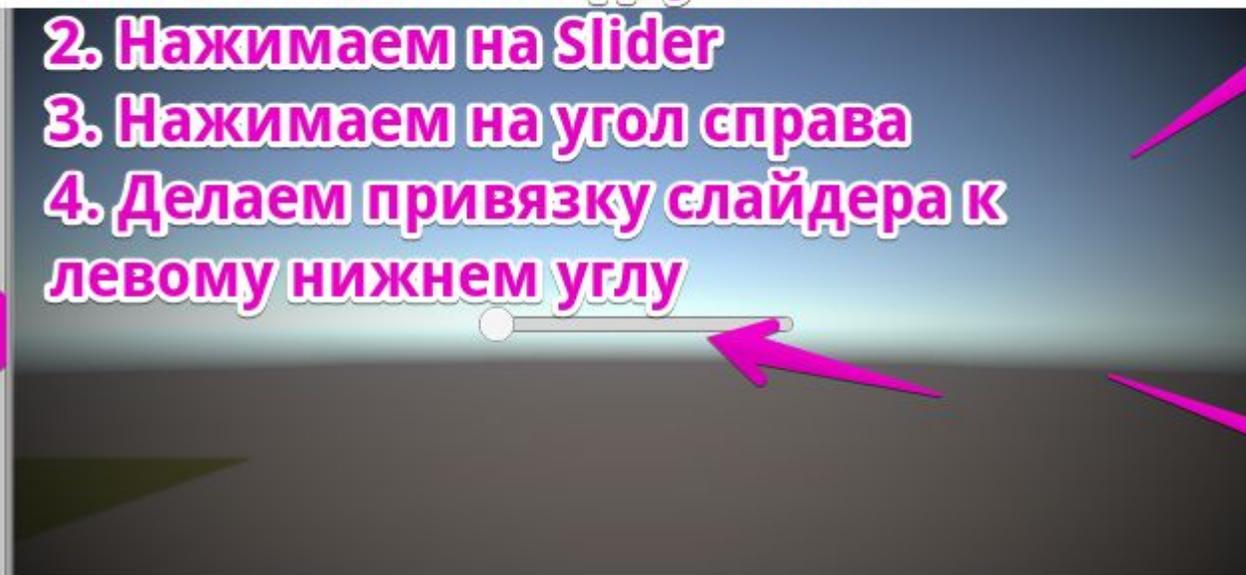
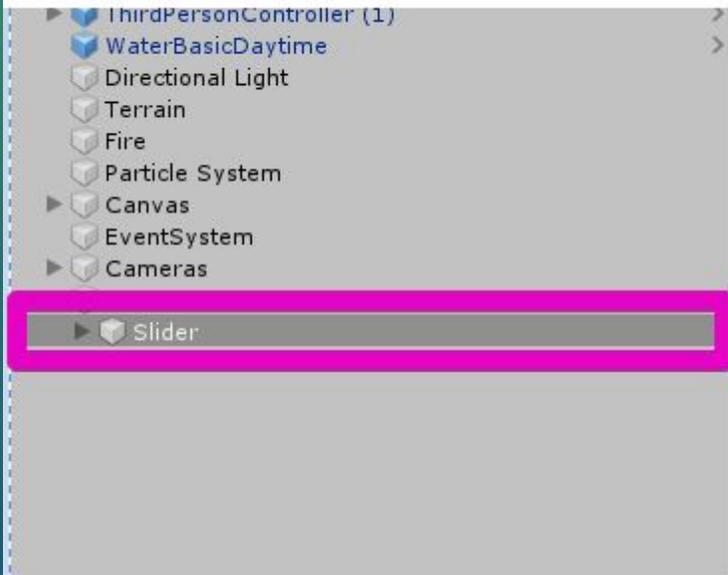
СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

1. Нажимаем вкладку Game

2. Нажимаем на Slider

3. Нажимаем на угол справа

4. Делаем привязку слайдера к левому нижнем углу



Slider Static
Tag Untagged Layer UI

Rect Transform

left	Pos X	Pos Y	Pos Z
bottom	84,9	30,5	0
	Width	Height	
	160	20	

Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Slider (Script)

Interactable

Transition Color Tint

Target Graphic Handle (Image)

Normal Color

Highlighted Color

Pressed Color

Disabled Color

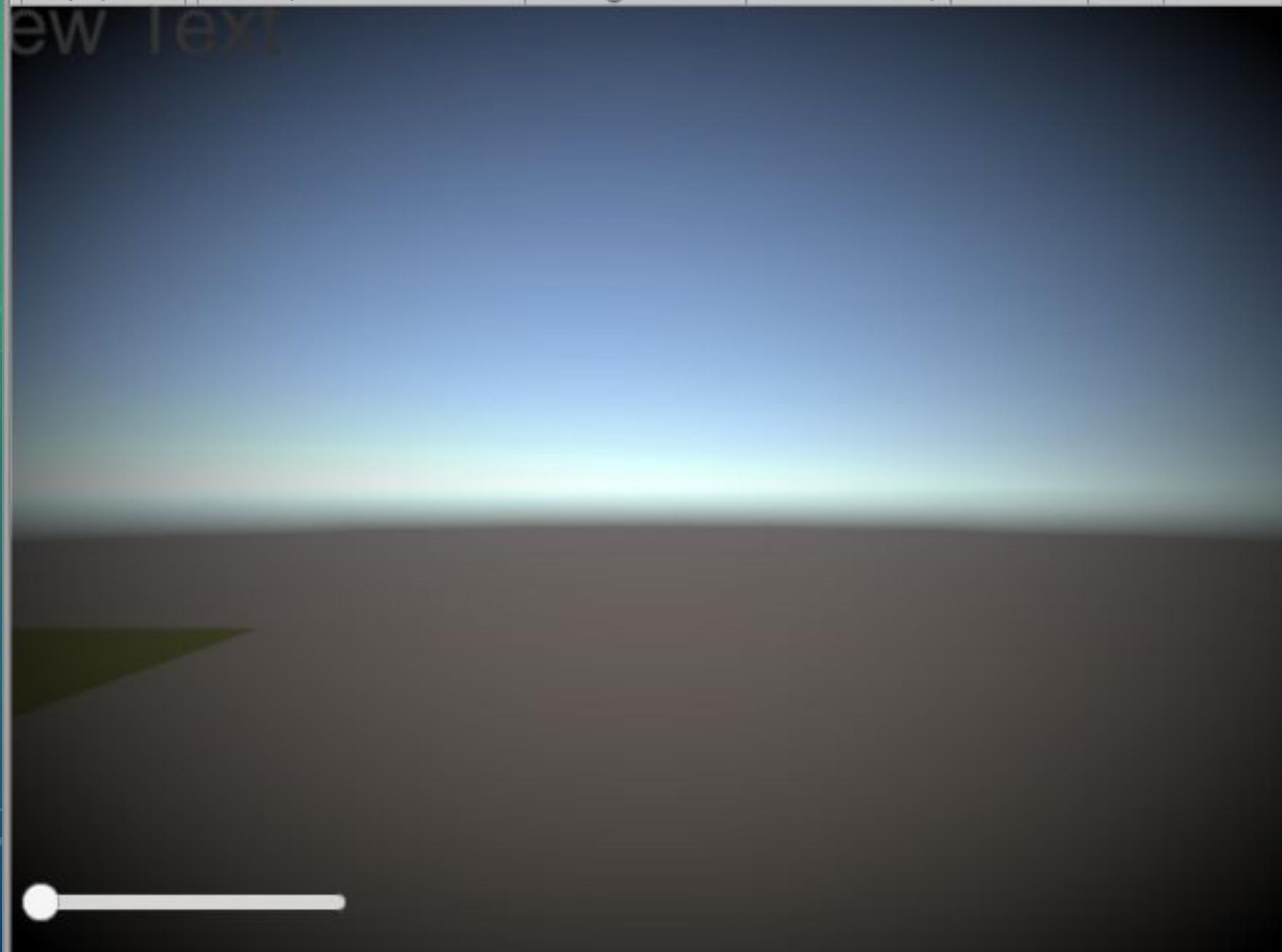
Color Multiplier 1

Fade Duration 0,1

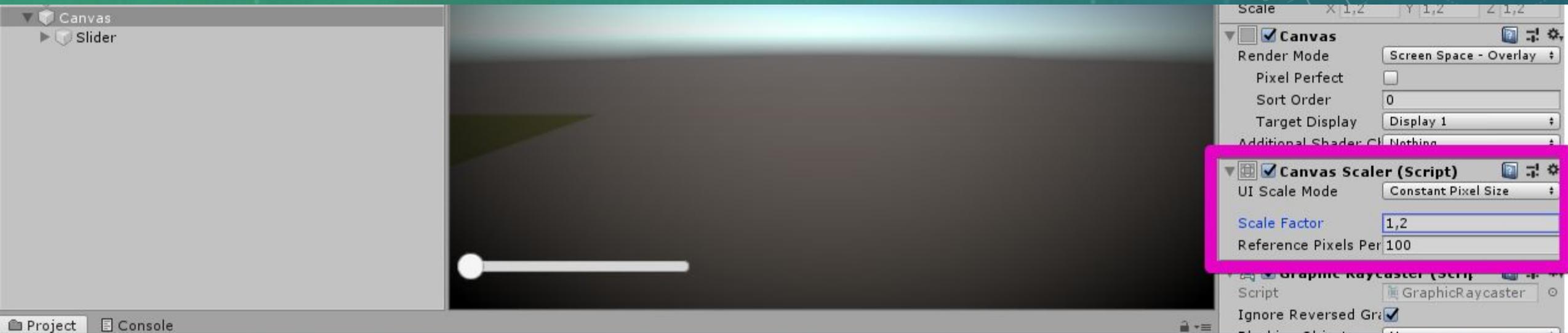
Navigation Automatic

Visualize

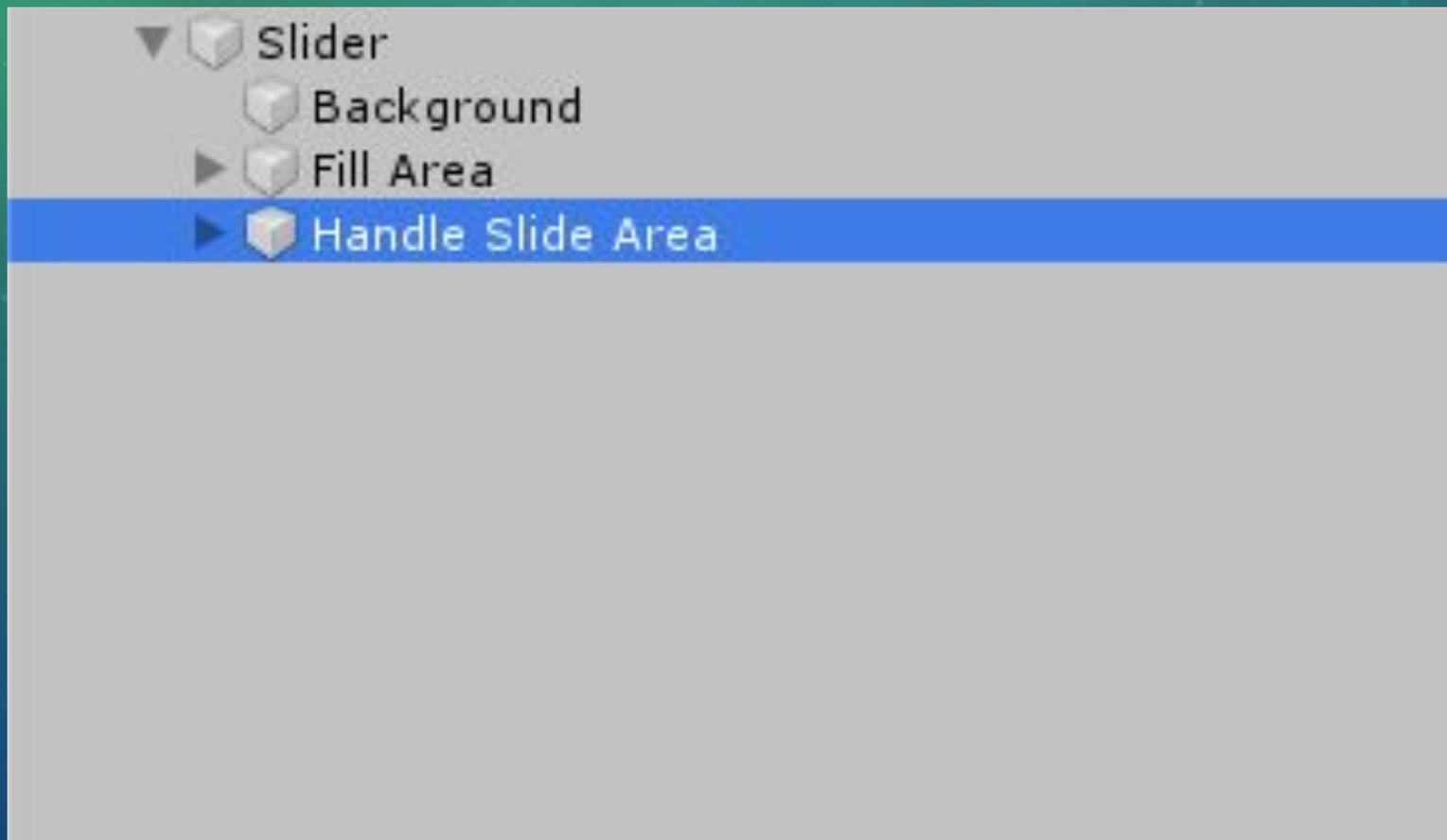
Fill Rect Fill (Rect Transfo)



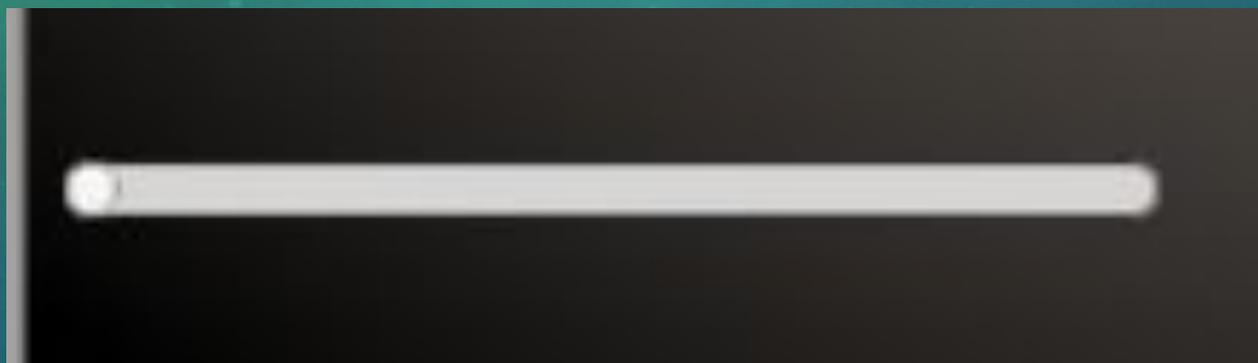
СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ



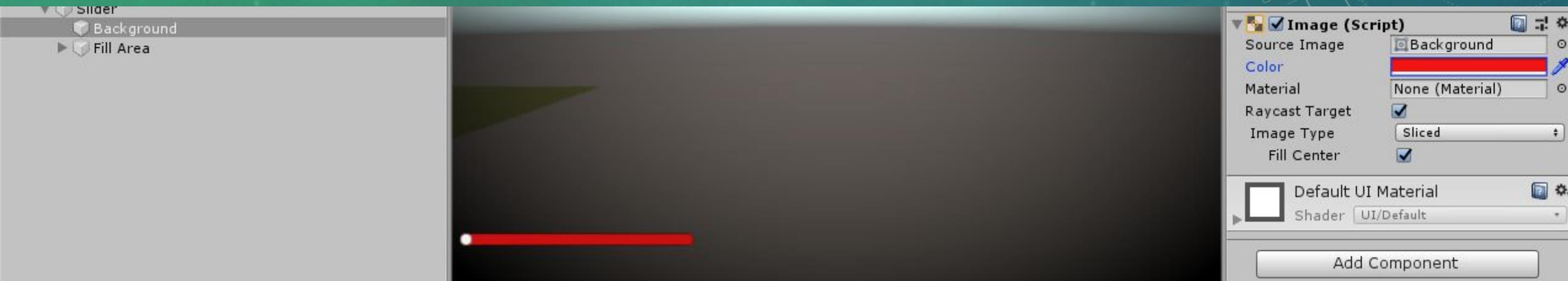
СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ



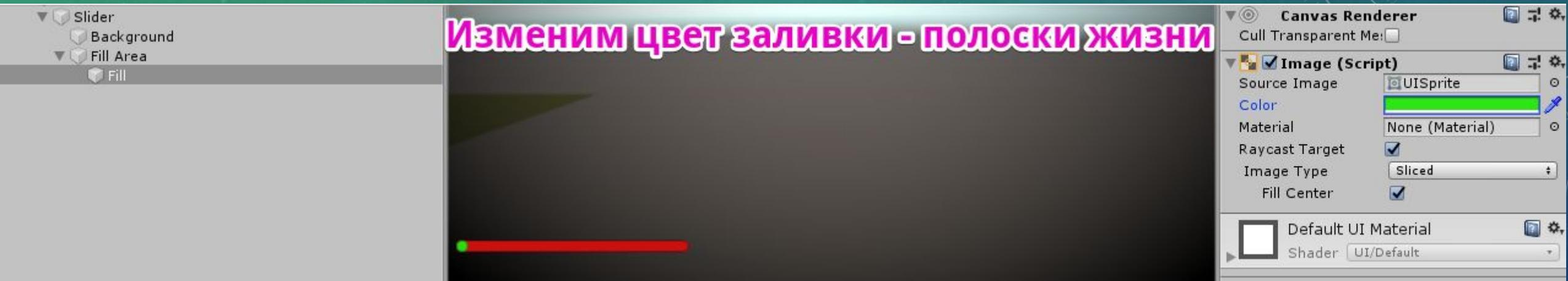
СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ



СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

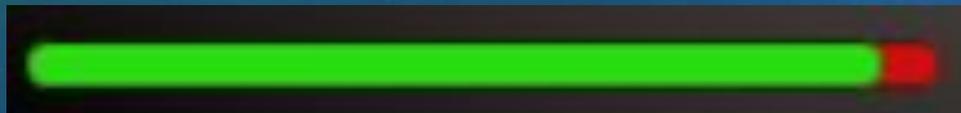


СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

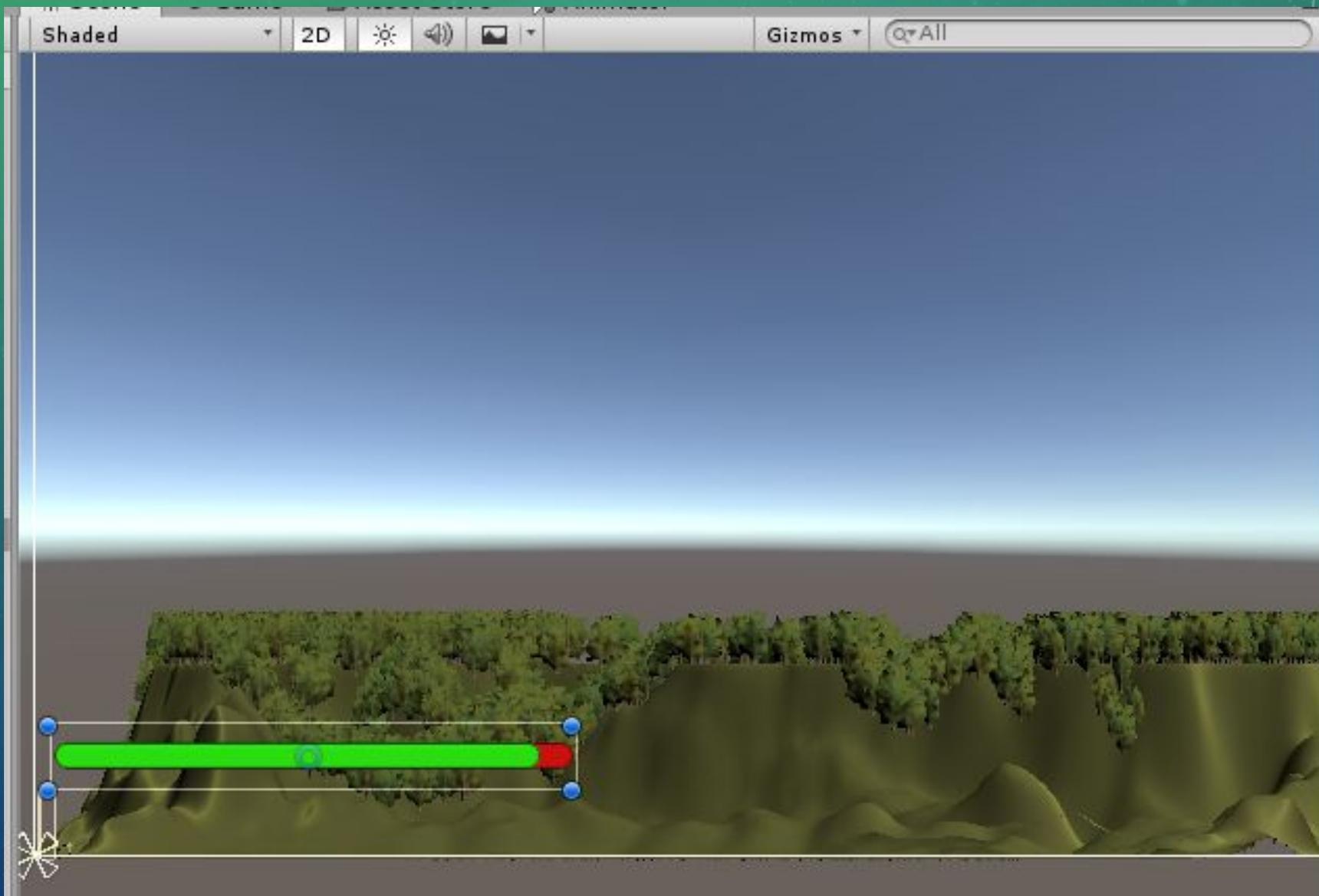


СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

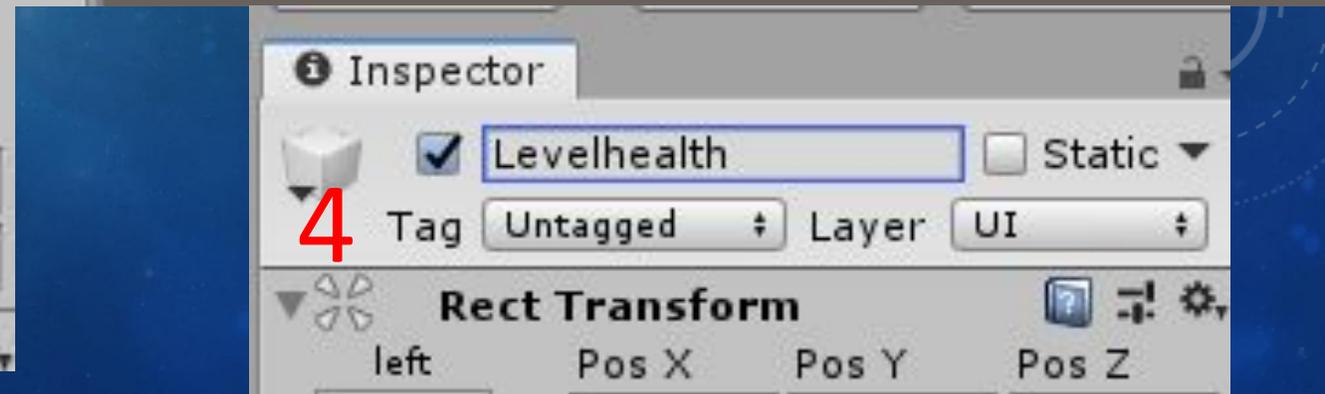
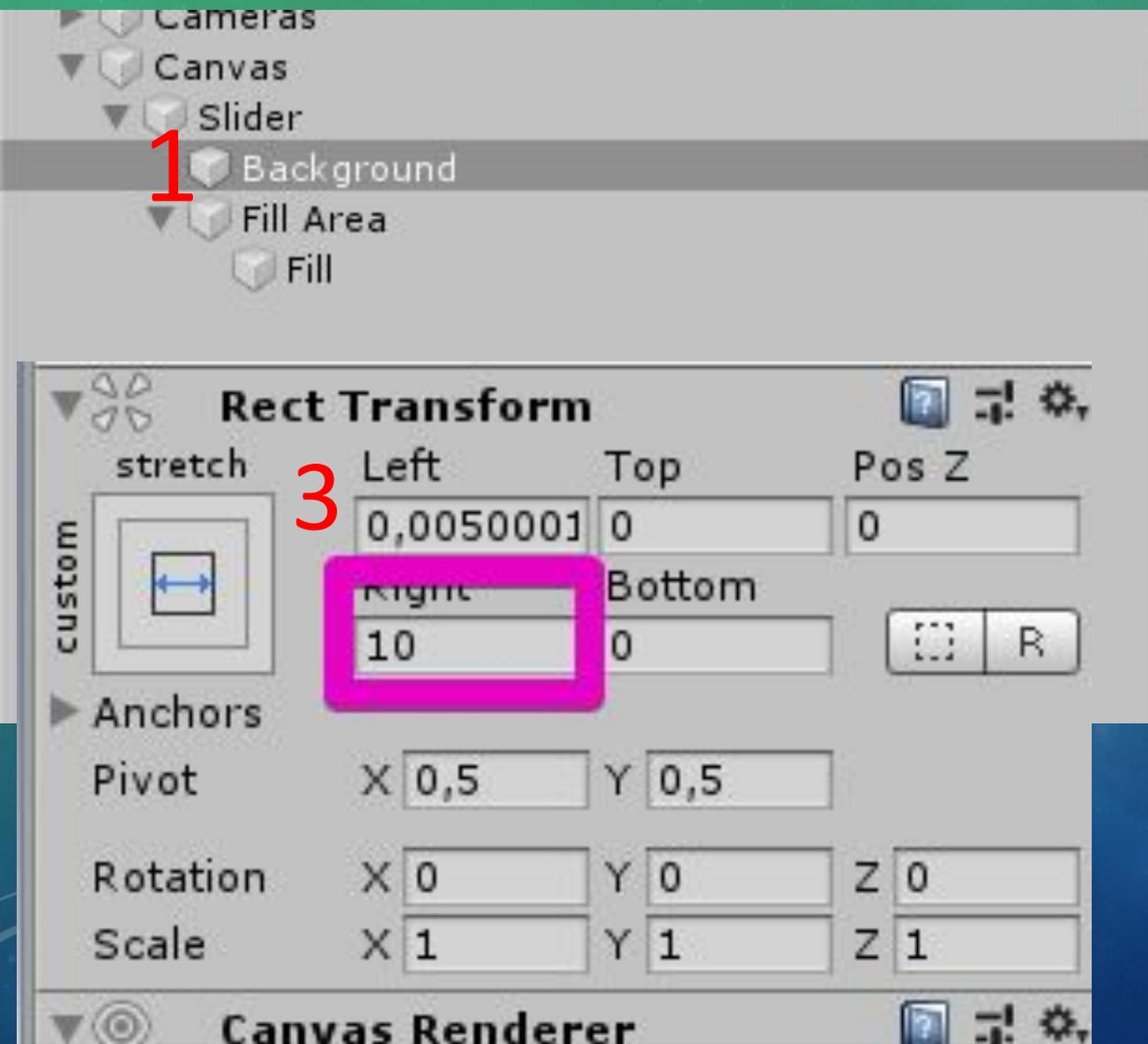
- Переходим на слайдер и выбиваем максимальное значение в блоке slider в инспекторе. Сразу поменяется полоска жизни.



СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

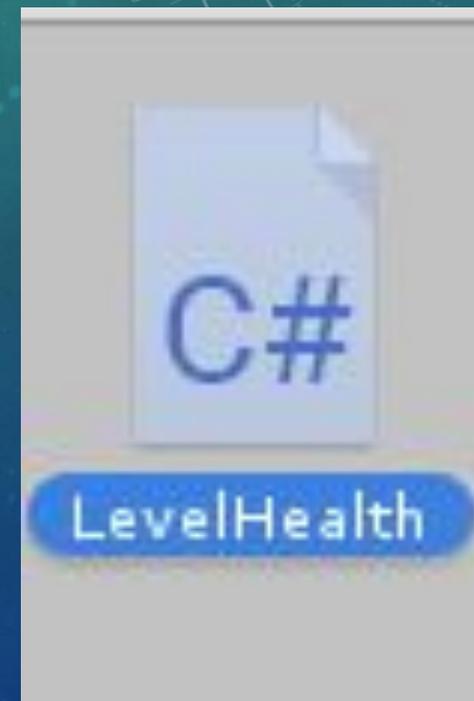


СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ



СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

- Создаем скрипт LevelHealth, именно так, с большой буквы, иначе будут сложности, синтаксис C# он такой. И открываем его в VisualStudio

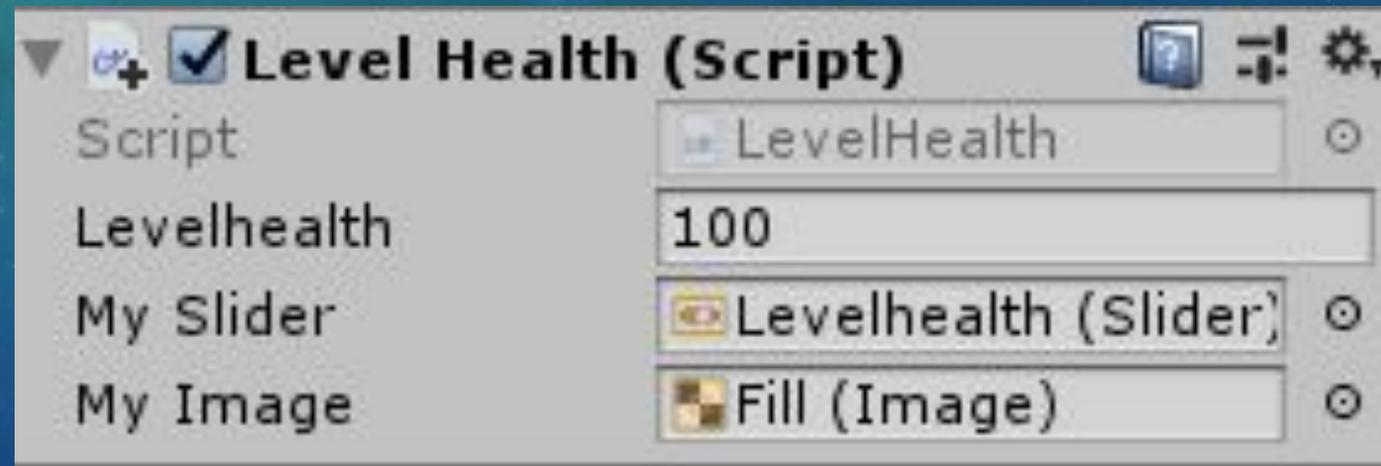


СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class LevelHealth : MonoBehaviour {
7
8     public int levelHealth = 100;
9     public Slider mySlider;
10    public Image myImage;
11
12    void Update () {
13        mySlider.value = levelHealth;
14        if (levelHealth < 10) {
15            myImage.enabled = false;
16        } else {
17            myImage.enabled = true;
18        }
19    }
20 }
```

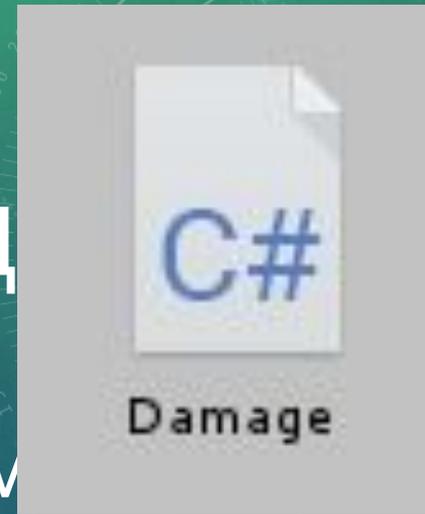
СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

- После чего поставим следующие блоки в настройки скрипта



СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИ

- Создаем новый скрипт в папке Scripts, называем Damage (урон) и переходим в Visual Studio. Как вы понимаете, будем использовать максимально простой метод – OnTriggerEnter, через теги.



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Damage : MonoBehaviour
6 {
7     private void OnTriggerEnter(Collider mycol)
8     {
9         if (mycol.tag == ("Player"))
10        {
11            mycol.GetComponent<LevelHealth>().levelhealth -= 10;|
12        }
13    }
14
15
16 }
17
```

СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

- Перетягиваем на наш куб, называем его Damage. Проверяем. Если все верно – урон получает враг. Безусловно, тут не очень с логикой. Ведь при ударе урон должны получать мы. Однако все дело в том, что у нас уже есть своя система получения урона, поэтому предлагаю немного уравновесить шансы. Открываем для правки скрипт HPVar. И допишем в нем строчку для получения урона при соприкосновении с врагом.

```
1 using UnityEngine;
2 using System.Collections;
3 using UnityEngine.UI;
4
5
6 public class HPBar : MonoBehaviour
7 {
8     //Выводим публичные переменные для установки стартовых значений жизней
9     public int HP = 100;
10    public Text HP_Bar;
11    public GameObject HP_Plus_Model;
12
13    void OnTriggerEnter(Collider other) //устанавливаем взаимодействие с объектом через триггер
14    {
15        if (other.tag == "Fire")
16        {
17            HP = HP - 25;
18        }
19
20        if (other.tag == "Enemy")
21        {
22            HP = HP - 15;
23        }
24
25        if (other.tag == "HP_Plus")
26        {
27            HP = HP + 50;
28            HP_Plus_Model.SetActive(false);
29        }
30    }
31 }
```

СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

Теперь сохраним сцену и создадим новую. В ней добавим канвас и два юи текста, первый будет гейм овер, второй - кнопка эскейп
Создадим скрипт, который сработает после окончания игры и позволит выйти из проекта

Game over

Нажми Escape чтобы выйти



СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class GameEscape : MonoBehaviour {
6
7     void Update () {
8         if (Input.GetKeyDown (KeyCode.Escape)) {
9             Application.Quit ();
10        }
11    }
12 }
```

СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИ



СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class GameOver : MonoBehaviour {
7
8     void Update () {
9         if (GetComponent <LevelHealth> ().levelHealth < 10) {
10             SceneManager.LoadScene ("1");
11         }
12     }
13 }
14
```

**ТУТ НАЗВАНИЕ
ВАШЕЙ СЦЕНЫ
ПРОИГРЫША**

СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

- Теперь выгребаем и мы и персонаж. Однако мы с вами просто касаемся персонажа, а я предлагаю вам добавить ещё и оружия, которое мы с вами настроим уже на следующем занятии.

Home > 3D > Props > Weapons > Apocalyptic Weapon Pack

Apocalyptic Weapon Pack

 ShadowOfTheTree

(not enough ratings) | [1 Reviews](#)

 RoyS

 a month ago

Excellent

This is a very good weapon pack. There is quite a variety of home made weaponry from

FREE

Download

СКРИПТИНГ ПОСЛЕ СОЗДАНИЯ АНИМАЦИИ

- Продолжение – в части 3