Министерство науки и высшего образования Российской Федерации

Федерального государственного бюджетное образовательное учреждение высшего образования

«Российской экономический университет имени Г. В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ.**

ПРАКТИЧЕСКИЕ РАБОТЫ № 2

ДИСЦИПЛИНА: «Разработка программных модулей»

Специальность: 09.02.07 Информационные системы и программирование

Квалификация: Программист

Листов:

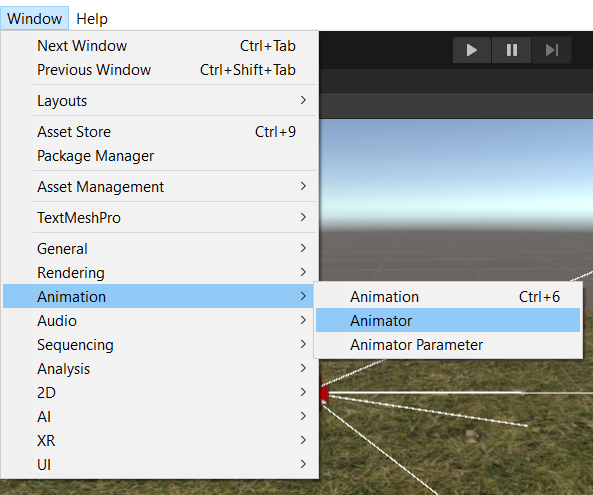
|  |  |  |
| --- | --- | --- |
| Выполнил студент |  | Проверил преподаватель |
| Группа П50-3-18 |  | \_\_\_\_\_\_\_\_\_\_\_О.Б.Солдатова |
| Пахомов Даниил Александрович |  | «\_­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­\_\_» \_\_\_\_\_\_\_\_2020 года |

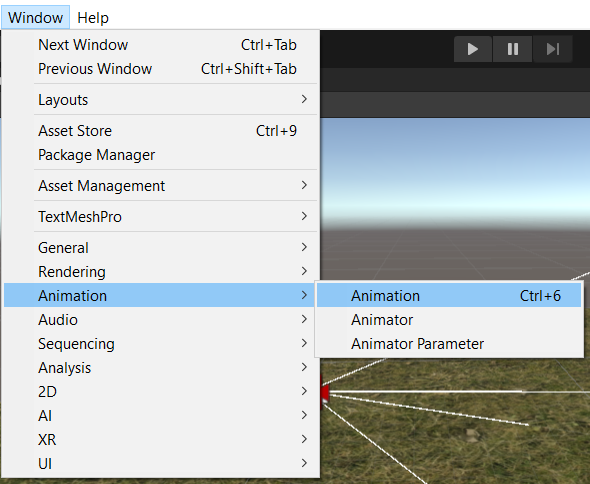
Москва, 2020

# Практическая работа №2

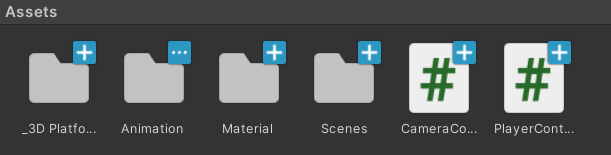
Цель: улучшить поведение игрока, добавить анимацию, создание Killzone, создание экземпляра и настройка респауна, возрождение с помощью сопрограмм, перемещение камеры при возрождении, удаление ошибки при анимации пряжка на респаун.

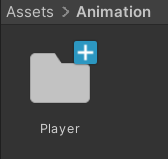
1. Добавление окон анимации



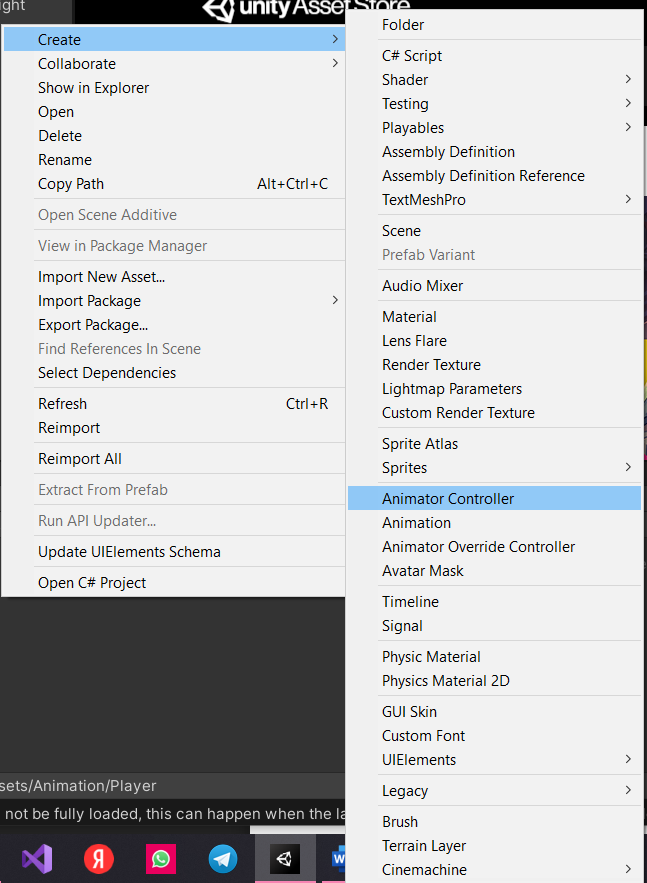


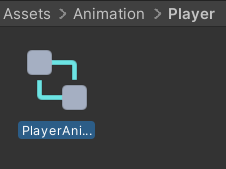
1. Создаем две папки Animation и Player



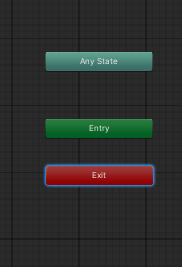


1. Создаем AnimatorController для контролировании анимации

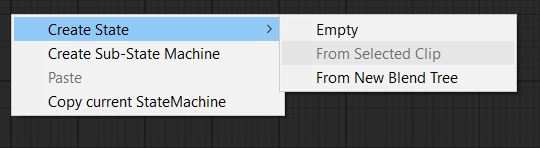




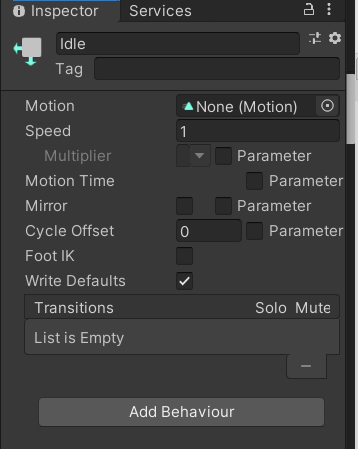
1. Базовые настройки анимации



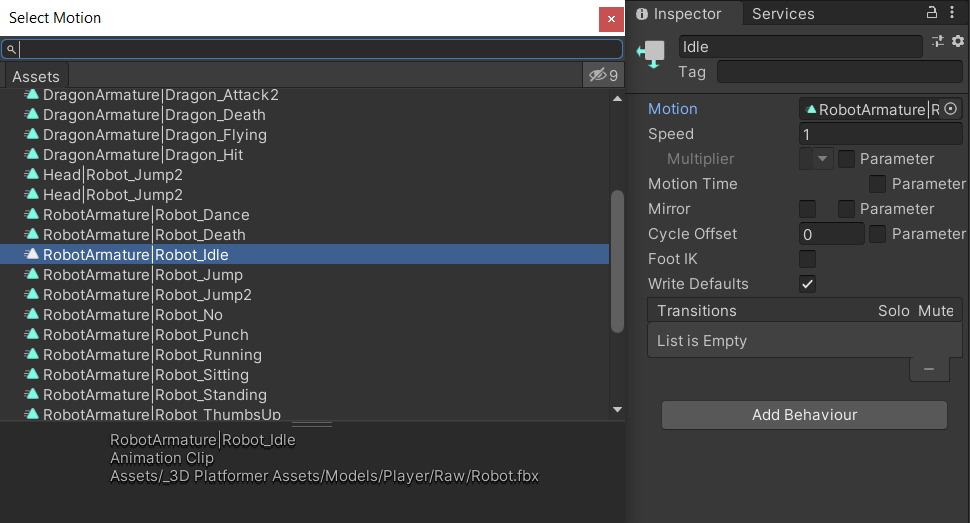
1. Создание нового действия



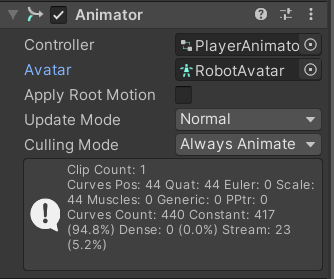
1. Переименовать ново созданный элемент



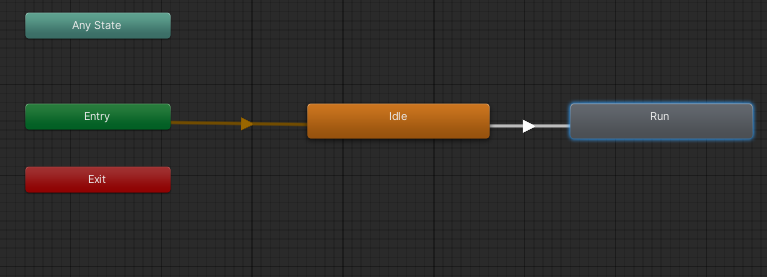
1. Выбрать уже созданную анимацию для робота и добавить его для Idle



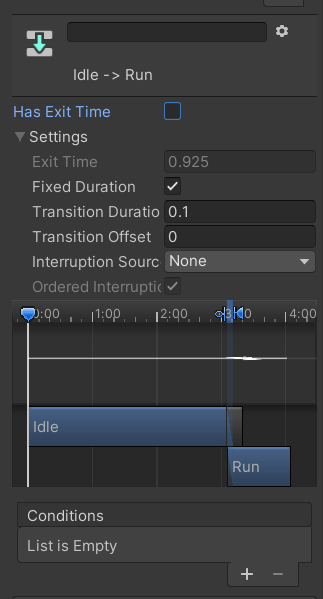
1. Назначить персонажу анимацию



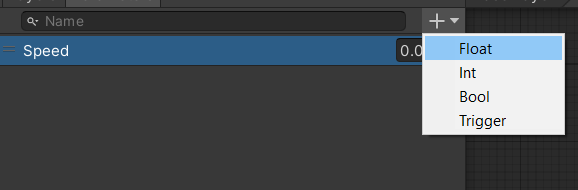
1. Создаем новое движение и связываем его с Idle

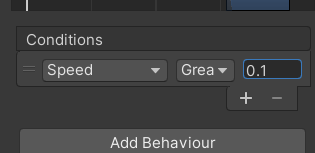


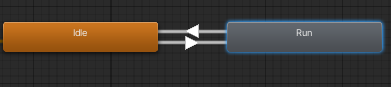
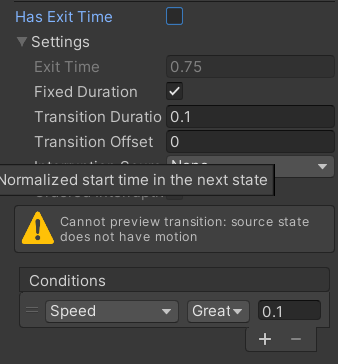
1. Настройка анимации



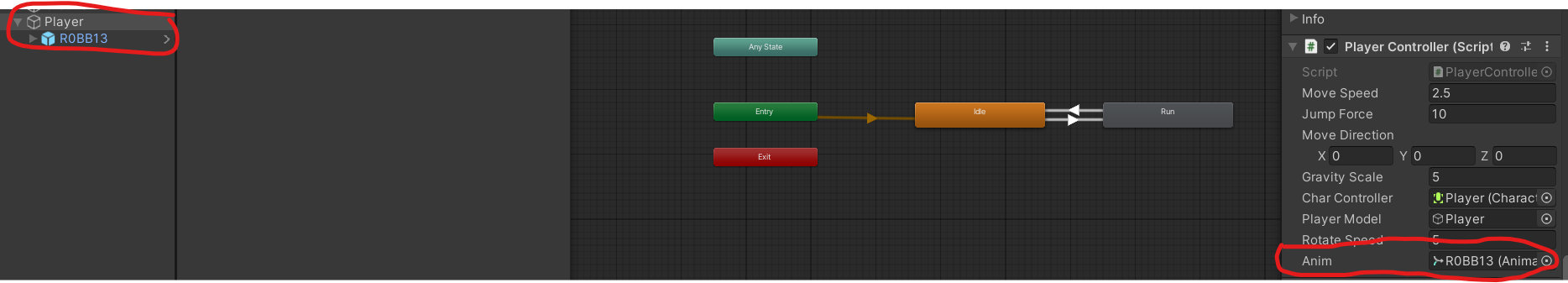
1. Создаем переменную, которая отвечает за скорость



1. Добавляем нашу созданную переменную к анимации
2. Повторяем те же самые действия с Элементом Run

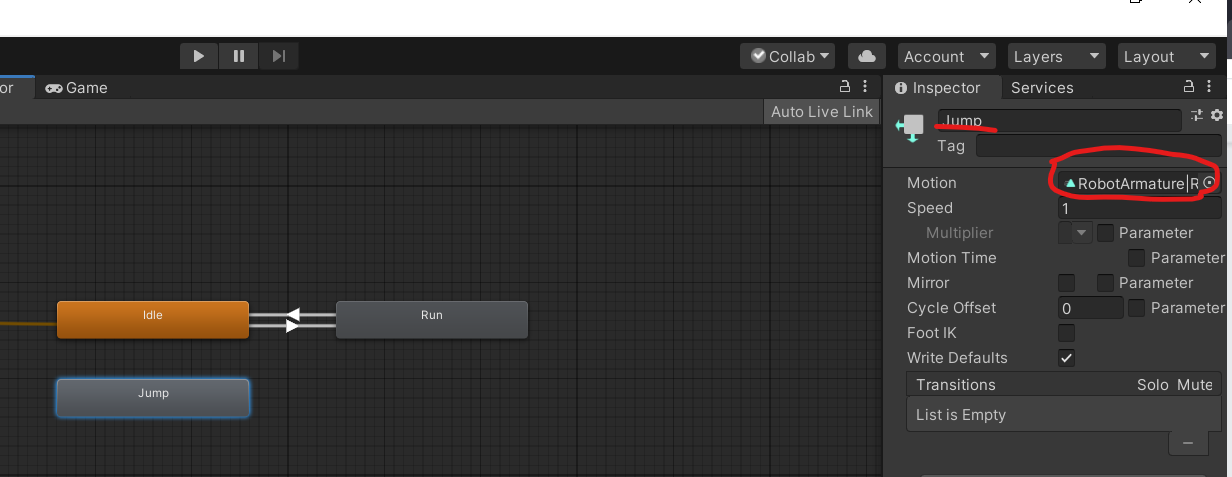


1. Связь анимации с персонажем

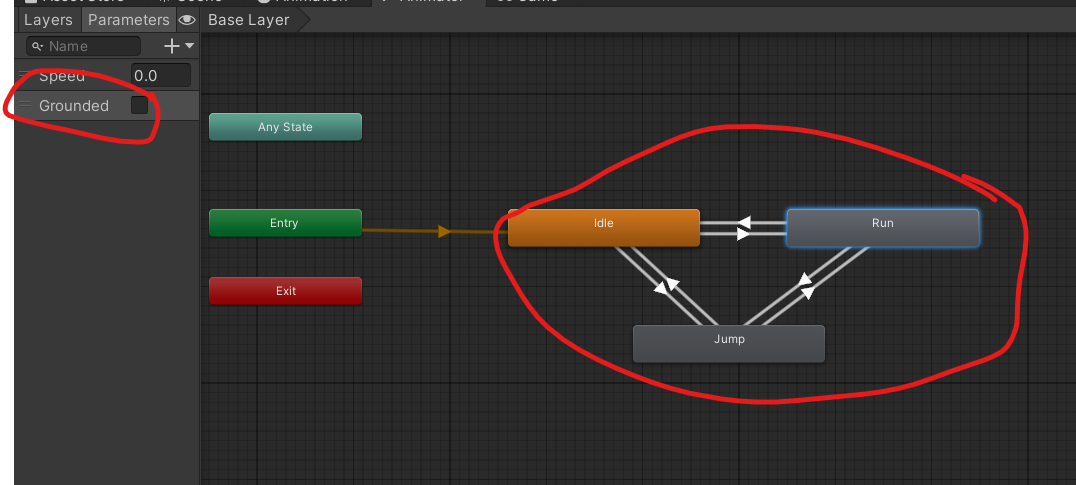




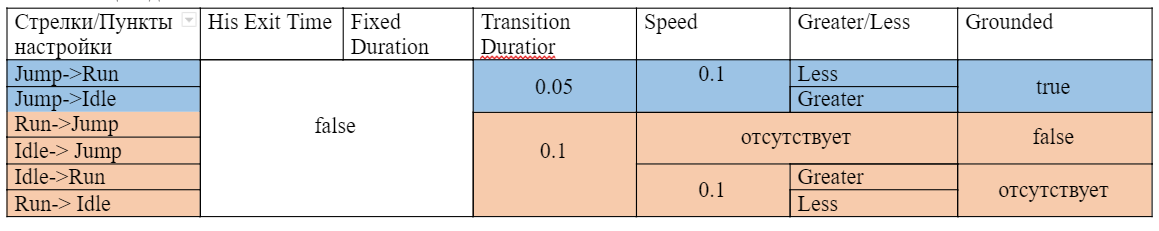
1. Создание новой анимации



1. Создание переменной и связей



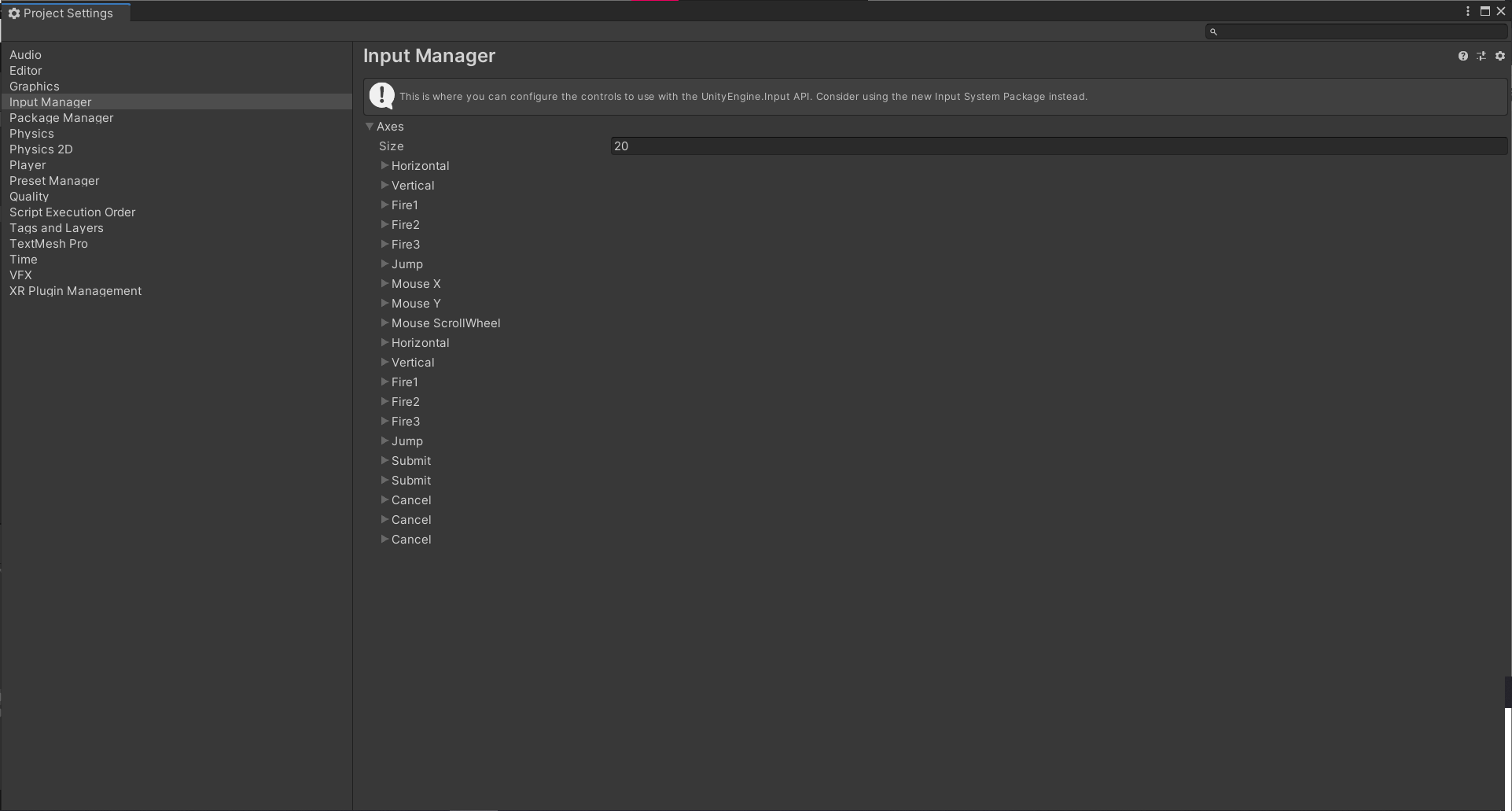
Все связи были настроены по данной таблице.



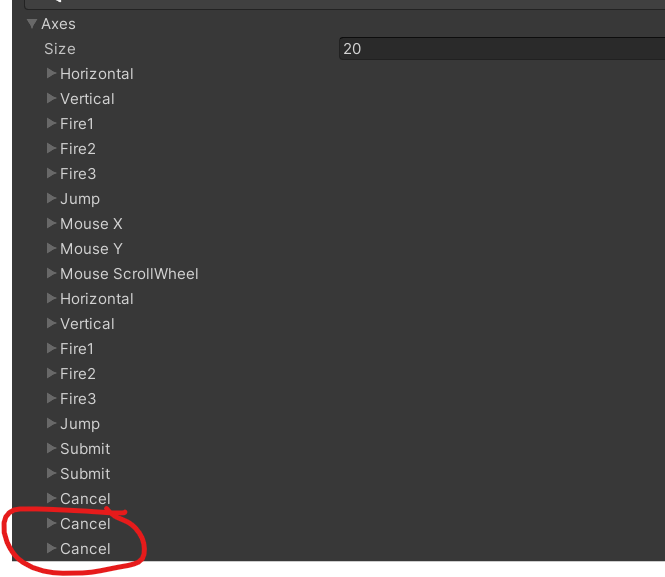
1. Листинг кода, проверка анимации

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class PlayerController : MonoBehaviour  {  public float moveSpeed;// Создание скорости передвижения  public float jumpForce; //Создание высоты прыжков  public Vector3 moveDirection;  public float gravityScale = 5f;  public CharacterController charController;  private Camera theCam;  public GameObject playerModel;  public float rotateSpeed;  public Animator anim;  private void Start()  {  theCam = Camera.main;  }  private void Update()  {  float yStore = moveDirection.y;  moveDirection = (transform.forward \* Input.GetAxisRaw("Vertical")) + (transform.right \* Input.GetAxisRaw("Horizontal"));  moveDirection.Normalize();  moveDirection = moveDirection \* moveSpeed;  moveDirection.y = yStore;  if (charController.isGrounded)  {  if (Input.GetButtonDown("Jump"))  {  moveDirection.y = jumpForce;  }  }  moveDirection.y += Physics.gravity.y \* Time.deltaTime \* gravityScale;  charController.Move(moveDirection \* Time.deltaTime);  if (Input.GetAxisRaw("Horizontal") != 0 || Input.GetAxisRaw("Vertical") != 0)  {  transform.rotation = Quaternion.Euler(0f, theCam.transform.rotation.eulerAngles.y, 0f);  Quaternion newRotation = Quaternion.LookRotation(new Vector3(moveDirection.x, 0f, moveDirection.z));  playerModel.transform.rotation = Quaternion.Slerp(playerModel.transform.rotation, newRotation, rotateSpeed \* Time.deltaTime);  }  anim.SetFloat("Speed", Mathf.Abs(moveDirection.x) + Mathf.Abs(moveDirection.z));  anim.SetBool("Grounded", charController.isGrounded);  }  } |

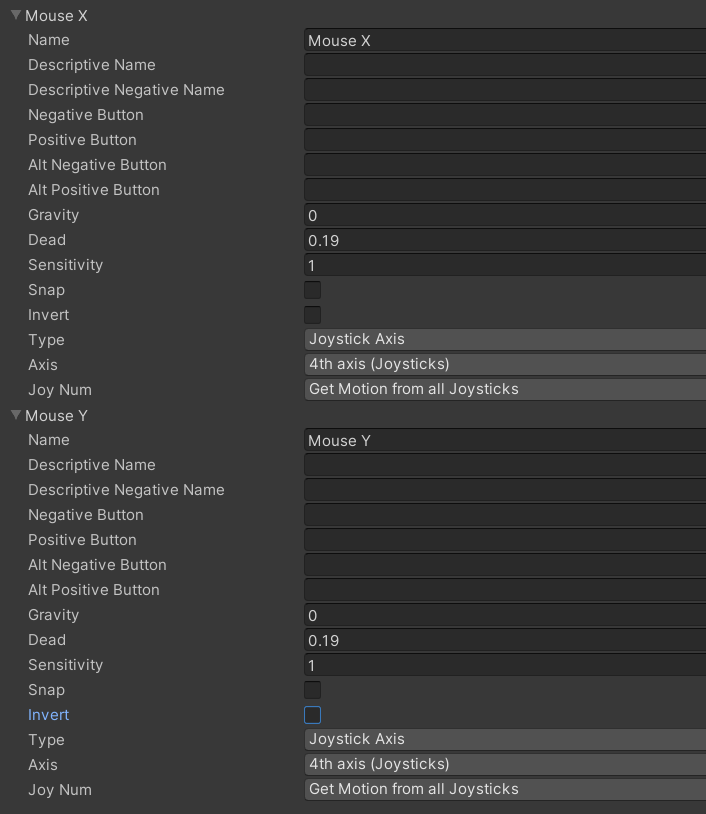
1. Движение при помощи джойстика
   1. Изменение количеств кнопок



* 1. Добавленные кнопки



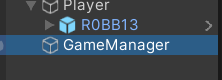
* 1. Настройка кнопок



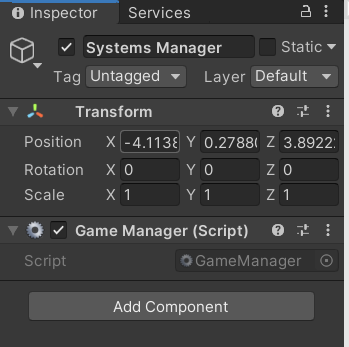
1. Следование за курсором мыши
   1. Создаем скрип GameManager



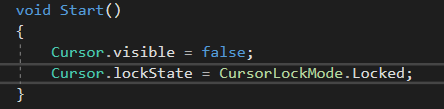
* 1. Создаем новый GameObject



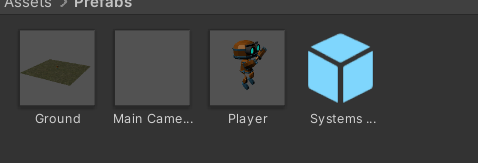
* 1. Привязываем наш созданный скрип к объекту



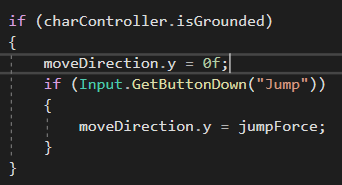
* 1. Листинг кода



1. Создание прифабов



1. Листинг кода



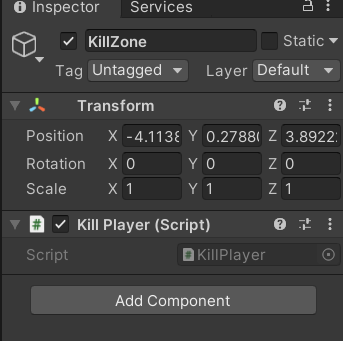
1. Создание KillZone



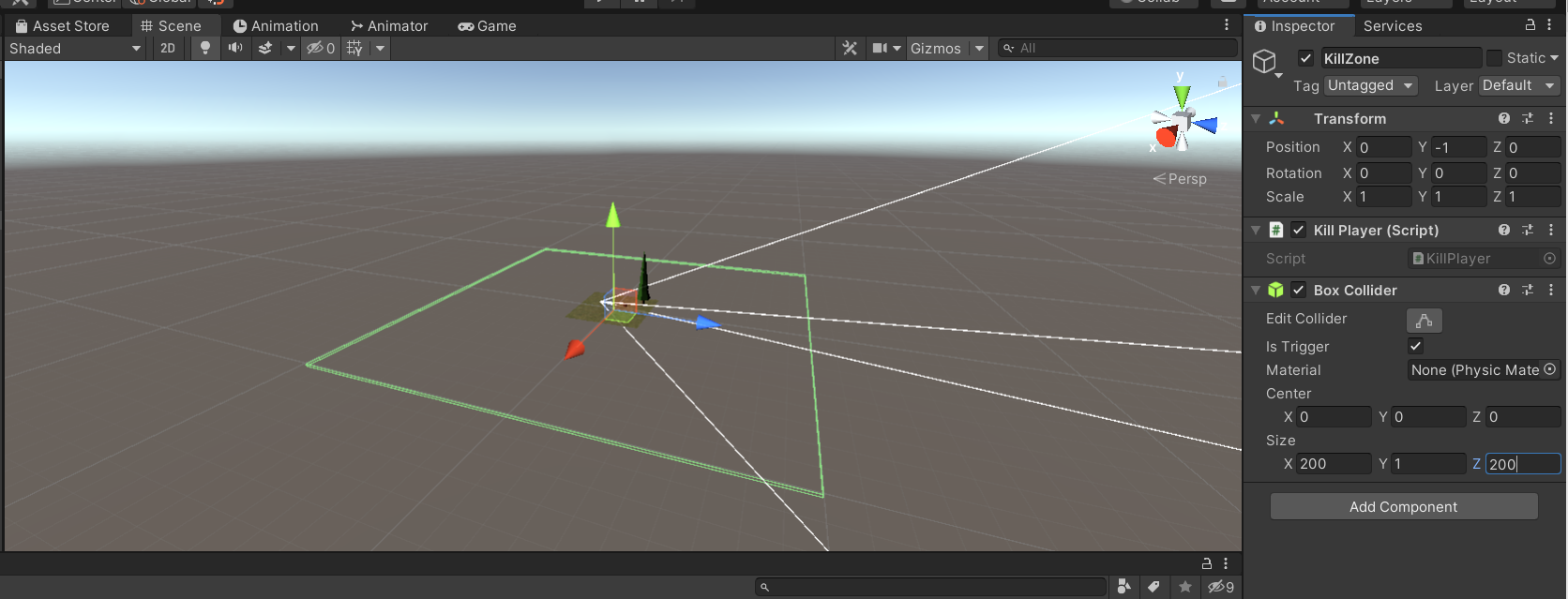
1. Создаем скрип



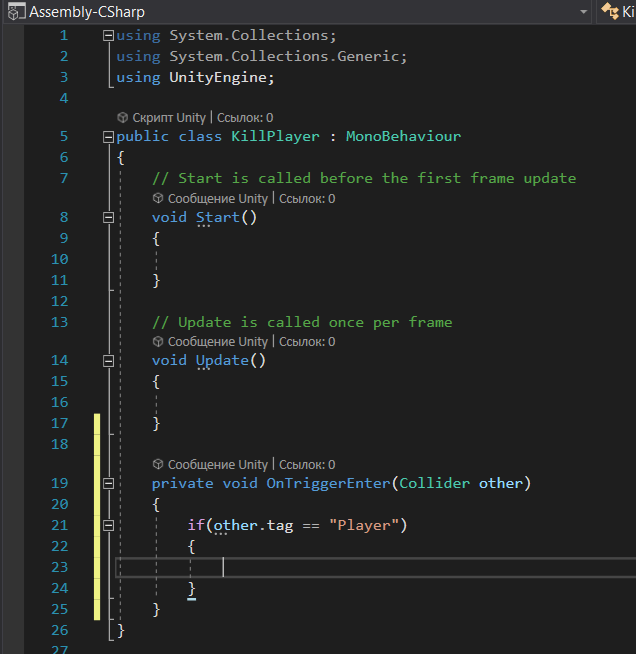
1. Связываем скрип и объект



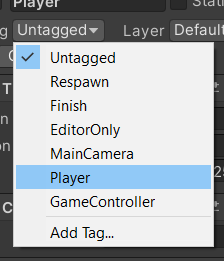
1. Настройка KillZone

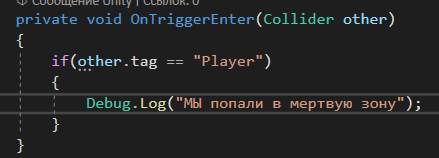


26.Листинг кода

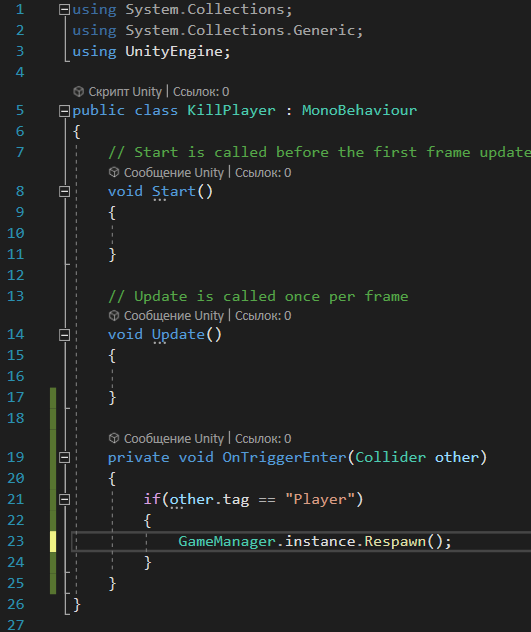


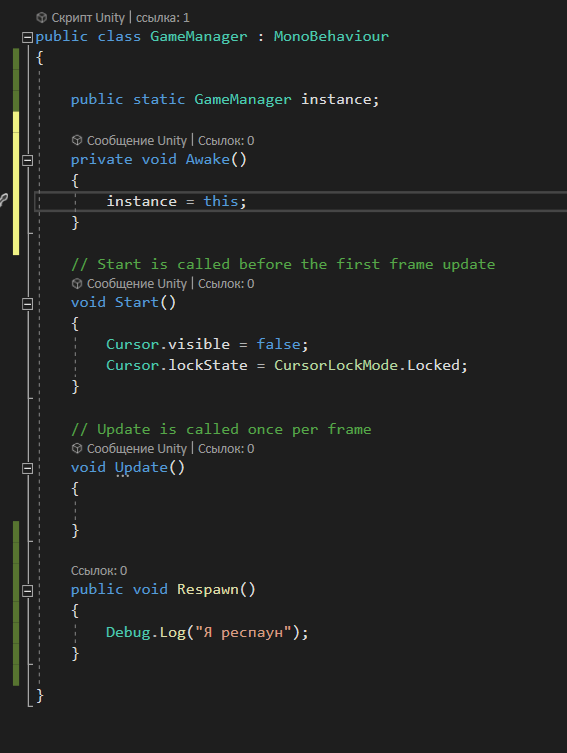
1. Настройка выхода за границы поверхности





1. Создание экземпляра и настройка респауна
   1. Листинг кода





1. Листинг кода

