**ПРОГРАММИРОВАНИЕ**

**ПЕРЕМЕЩЕНИЯ КАМЕРЫ**

**НА СЦЕНЕ**

**СКРИПТ ДЛЯ ПЕРЕМЕЩЕНИЯ КАМЕРЫ**

**public class Scroll : MonoBehaviour**

**{**

**[SerializeField]**

**Transform targetPos;**

**[SerializeField]**

**int sensivity = 3;**

**void Update(){**

**if (Input.GetMouseButton(1))**

**{**

**float y= Input.GetAxis("Mouse X")\*sensivity;**

**if (y != 0) transform.RotateAround(targetPos.position, Vector3.up, y);**

**}**

**[SerializeField]**

**float scrollSpeed = 10f;**

**[SerializeField]**

**int maxdistance = 20;**

**[SerializeField]**

**int mindistance = 1;**

**bool ControlDistance (float distance)**

**{**

**if (distance > mindistance && distance < maxdistance) return true;**

**return false;**

**}**

**float x = Input.GetAxis("Horizontal")/sensivity; // клавиши A, D**

**if (x != 0)**

**{**

**Vector3 newpos = transform.position + transform.TransformDirection(new Vector3(x, 0, 0));**

**if (ControlDistance(Vector3.Distance(newpos, targetPos.position))) transform.position = newpos;**

**}**

**Float z = Input.GetAxis("Mouse ScrollWheel") \* scrollSpeed;**

**if z != 0)**

**{**

**Vector3 newpos = transform.position + transform.TransformDirection(Vector3.forward \*z);**

**if (ControlDistance(Vector3.Distance(newpos, targetPos.position))) transform.position = newpos;**

**}**

**}**

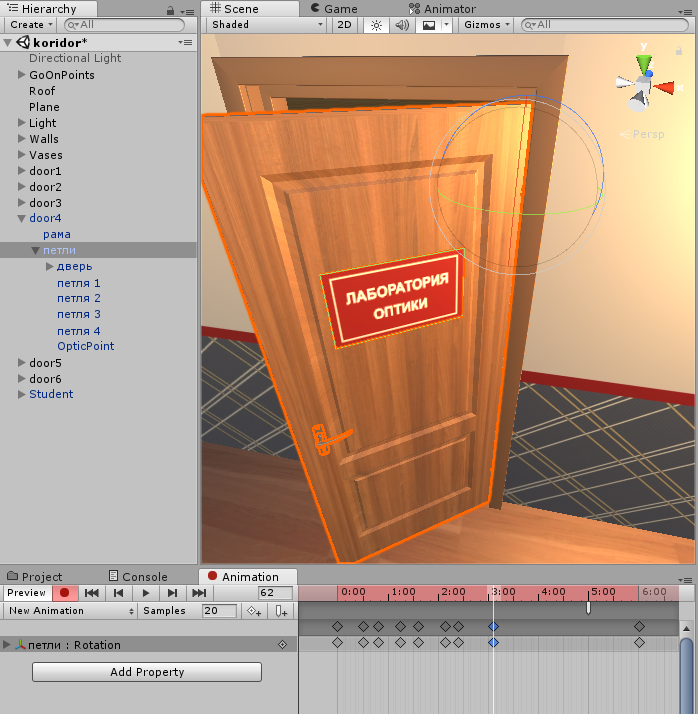
**}**

**СОЗДАНИЕ АНИМАЦИИ ОБЪЕКТА**

**В СРЕДЕ UNITY**

Для создания анимации в редакторе **Unity** нужно: выбрать в окне проекта **Project** папку **Assets**, выбрать объект, который будет анимироваться, открыть для него вкладку **Window/Animation/Animation,** задать в открывающемся окне имя для файла анимации с типом **\*.anim** и сохранить его в папке проекта **Assets**.

После сохранения файла анимации в нем создается анимация объекта. На временной шкале в нижней части открывшегося окна редактора **Unity** нажать кнопку с красным кружком (на рисунке) и в соответствующих ключевых кадрах записать требуемую анимацию для объекта.

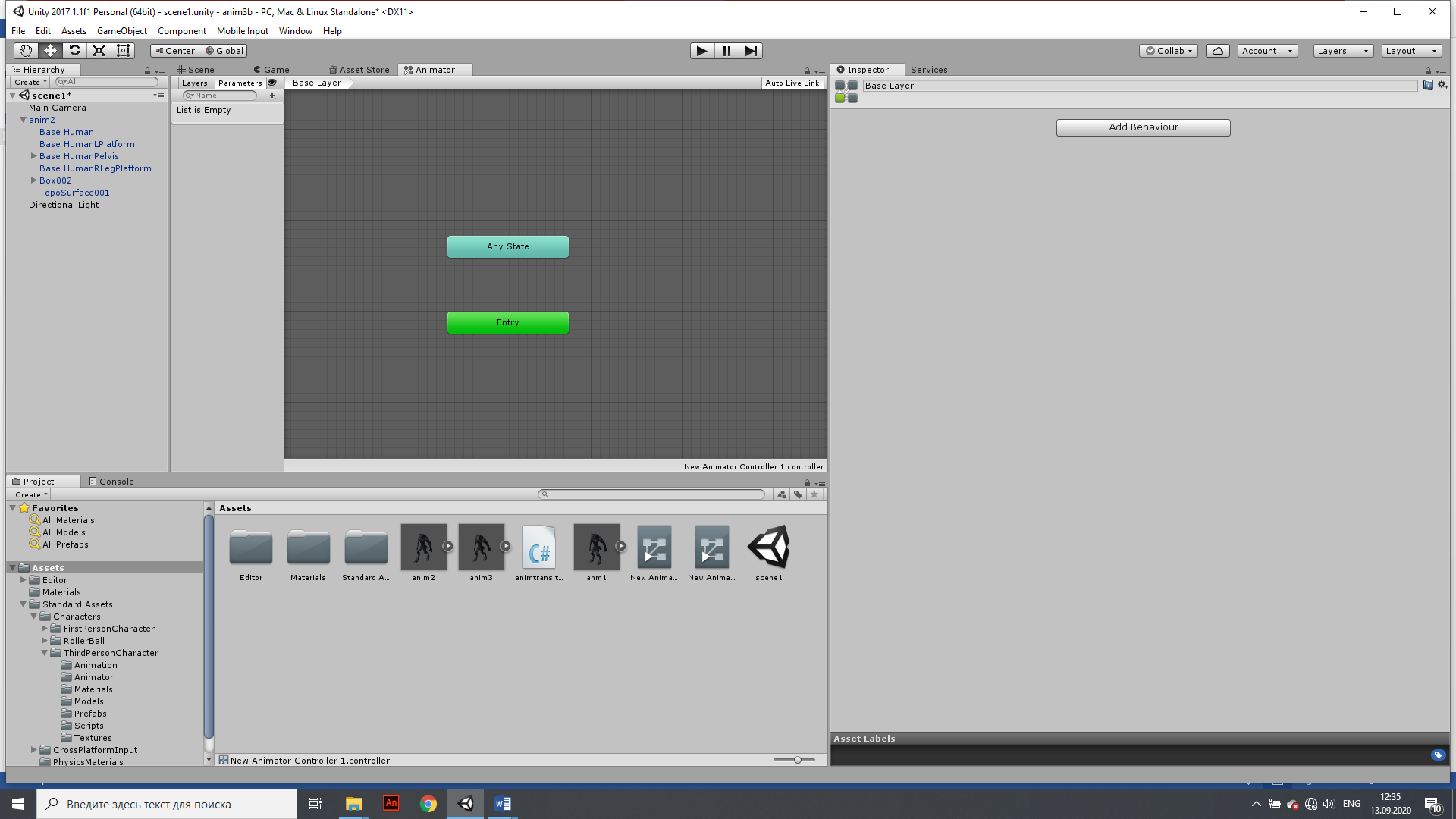


*При создании анимации необходимо учесть, что, если объект создан в среде* ***3dsMAX,*** *то при его анимируемом повороте, например открытии дверки духового шкафа, точка центра вращения* ***Pivot*** *по умолчанию будет расположена по центру. Чтобы объект вращался правильно вокруг края дверки, нужно в* ***Unity*** *создать пустой объект типа* ***Empty Object*** *(невидимый при проигрывании) и в окне* ***Hierarchy*** *сделать анимируемую дверку дочерним элементом этого пустого объекта, при этом разместить предполагаемую точку вращения в центре пустого объекта.*

**СОЗДАНИЕ КОНТРОЛЛЕРА АНИМАЦИИ**

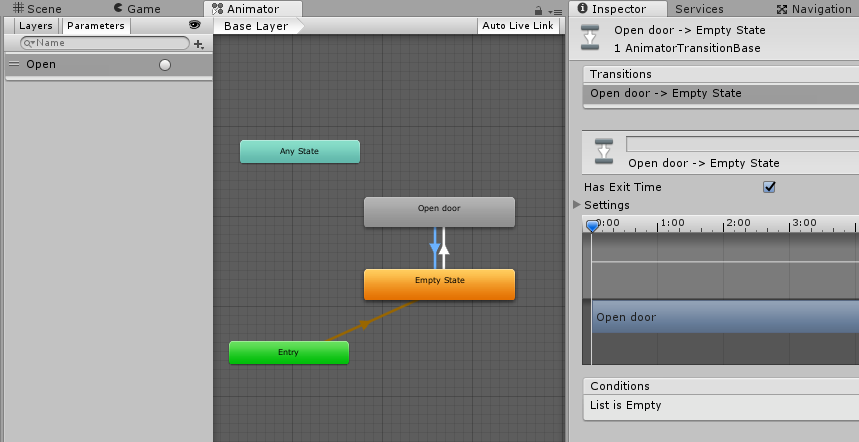
Затем нужно добавить к объекту, контроллер анимации **Animator** – кликнуть правой кнопкой мыши по свободной области в папке **Assets**, выполнить команду **Create->Animator Controller** и дать имя контроллеру. Дважды кликнув на созданный контроллер, перейти в окно редактора.

В начале есть два блока, это **Entry**, с которого будут начинать проигрываться анимации и **Any State** – особое состояние, которое всегда существует и позволяет перейти от любого состояния к конкретному.



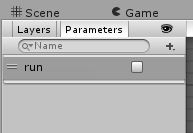
Создать **Empty State** как состояние по умолчанию, щелкнув правой кнопкой мыши в свободной части окна аниматора и выполнив команду **Create State/Empty,** в качестве состояния покоя у двери и добавить созданную анимацию отдельным состоянием.

Переходы к состоянию анимации открытия двери и обратно должны происходить по булевой переменной **Bolean**, причем обратно - по условию установки режима **Has Exit Time**. При этом после создания анимации двери необходимо *снять зацикливание* в контроллере анимации.

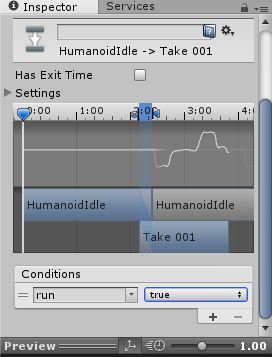


**УПРАВЛЕНИЕ АНИМАЦИЕЙ В UNITY НА ЯЗЫКЕ C#**

Для программного управления анимацией непосредственно в окне редактора необходимо создать программный код на языке C# и, прежде всего, в окне аниматора **создать переменную** открыввкладку **Parameters,** нажать плюс (добавить), выбрать тип создаваемой переменной, затем дать имя переменной, например **run**.



После этого выбрать в окне редактора щелчком мыши переход от анимации ожидания к анимации движения. Затем в Инспекторе для перехода в списке **Conditions** и добавляем в него нашу переменную **run** со значением **true**, нажав кнопку **+**.



Нажать правой кнопкой мыши на анимацию ходьбы и выбрать **Make transition**, чтобы создать обратную связь на анимацию по умолчанию. Выбрать новый переход и опять снять галочку **Has exit time**. Затем в список **Conditions** добавляем нашу переменную **run**, но со значением **false**.

**ДОБАВИТЬ СКРИПТ ДЛЯ ПЕРЕКЛЮЧЕНИЯ МЕЖДУ АНИМАЦИЯМИ**

В окне инспектора **Inspector** добавить **Add Component** для объекта новый скрипт **New Script** и в редакторе кода создатьследующий код:

**Animator anim;** //переменная типа **Animator** для ссылки на анимацию

**void Start()**

**{**

**anim = GetComponent<Animator>();** //инициализация контроллера анимации

**}**

**void Update()**

**{**

**If(Input.GetKeyDown(KeyCode.Q))** //если нажата клавиша **q**

**{**

**anim.SetBool(“run”, true);** // переменная, отвечающая за переход имеет значение **true**

**}**

**If(Input.GetKeyDown(KeyCode.W))** //если нажата клавиша **w**

**{**

**anim.SetBool(“run”, false);** // переменная, отвечающая за переход имеет значение **false**

**}**

**}**

**ПРОИГРАТЬ АНИМАЦИЮ ИЗ ЛЮБОГО СОСТОЯНИЯ**

Чтобы проиграть какую-либо анимацию можно было из любого состояния, необходимо:

1. создать переход от **Any State** к этой анимации, например с именем **Hit**. Для возвращения в состояние ожидания необходимо создать переход от новой анимации **Hit** к анимации по умолчанию.
2. создать еще одну переменную типа **Trigger** (например **hitten**), которая будет отвечать за щелчок мышью для запуска анимации. Добавим к переходу от **Any State** эту переменную, а для обратного перехода оставим галочку **Has Exit Time**, чтобы переход совершался автоматически (без использования переменной) один раз после проигрывания анимации.
3. добавить в метод **Update()** код проигрывания новой пользовательской анимации по клику мыши:

**if (Input.GetMouseButtonDown(0))  {  anim.SetTrigger("hitten"); }**