Лабораторная работа №2. УПРАВЛЕНИЕ 3D-МОДЕЛЬЮ ТАНКА В СРЕДЕ UNITY

1. В созданном ранее проекте Unity3D с внедренной моделью танка в окружающую среду *Terrain* задать в Инспекторе компонент **Rigidbody** танку и задать ему массу *(Mass)* около 1000.
2. Раскрыть иерархию вложенных в танк объектов в окне *Hierarchy* проекта и проверить правильность вложений: в танк должен быть вложен корпус, а затем башня, а в башню вложен ствол. Кроме того, при программном управлении движением 3D-объектов необходимо учесть разную направленность осей координат в Unity (вертикальная ось – Y, а в горизонтальной плоскости – оси X-Z) и 3ds MAX (вертикальная ось – Z, а в горизонтальной плоскости – оси X-Y), что, как правило, приводит к необходимости использовать требуемое направление (Vector3.forward, Vector3.right).
3. Создать файл сценария, выполнив в инспекторе ресурсов *Assets* необходимо выполнить команду **Create/C# Script** и дать файлу соответствующее имя *(например,* ***tank****)*.
4. Добавить скрипт как компонент в модель танка (то есть на верхний объект иерархии танка, в нашем случае это его корпус) – перетянуть файл из окна *Assets* на объект танка в окне *Hierarchy* (или нажать кнопку *Add Component* и найти в списке созданный начальный скрипт). В инспекторе ресурсов *Assets*, кликнув на значке скрипта два раза, открыть файл в *Visual Studio*.
5. Задать в области объявления класса **MonoBehaviour:** переменные для скорости **float MoveSpeed** **= 0.1f;** и поворота танка **float RotateSpeed** **= 1f**;.
6. Добавить в метод **void Update()** движение танка вперед-назад, каждый кадр прибавляя к текущему положению танка смещение в сторону направления его локальной оси z **(Vector3.forward)**, с применением преобразования из локальных координат в глобальные **(TransformDirection)**, скорость **(MoveSpeed)** и ввод с клавиатуры **Input.GetAxis("Vertical")** // клавиши ws. В итогеперемещение реализуется следующим образом: **transform.position += transform.TransformDirection (Vector3.forward \* MoveSpeed \* Input.GetAxis("Vertical"));**
7. Добавить в метод **Update()** выполнение вращения танка вокруг вертикальной оси Y влево-вправо при условии нажатия соответствующих клавиш через метод *transform.Rotate:*

**transform.Rotate(0f, Input.GetAxis("Horizontal")\*RotateSpeed, 0f);** // ("Horizontal")-клавиши AD

1. Создать новый скрипт (**bashnya.cs**) для вращения башни вокруг вертикальной оси, добавив переменную для хранения скорости (**float bashnyaSpeed**), а в методе **Update()** добавить метод *Rotate*, с использованием перемещения мыши влево-вправо через **Input.GetAxis("Mouse X"),** и скорости: **transform.Rotate(0f, Input.GetAxis("Mouse X") \*** **bashnyaSpeed, 0f);.** Поместить данный скрипт в качестве компонента на башню.
2. Создать новый скрипт для вращения ствола (**stvol.cs**) при перемещении мыши вверх-вниз с ограничением по диапазону угла поворота. Установить ограничение угла в заданных пределах: объявить переменные **float min** (минимальный угол), **float max** (максимальный угол), которые нужно определить самостоятельно, а также **float angle = 0;** – текущий угол смещения. В метод **Update()** добавить следующие условные конструкции:

// перемещаем мышь если смещение не вышло за установленные пределы  
**if (Input.GetAxis("Mouse Y") > 0 && angle > min)  
        {  
            transform.localRotation = Quaternion.Euler(angle--, 0, 0);**// поворачиваем ствол со смещением **}  
        if (Input.GetAxis("Mouse Y") < 0 && angle < max)  
        {  
            transform.localRotation = Quaternion.Euler(angle++, 0, 0);   
        }**