**Лабораторная работа №9**

**Тема: Создание скриптов. Переменные. Обращение к компонентам. Работа с методами интерфейса Input.**

***Работа с консолью Unity***

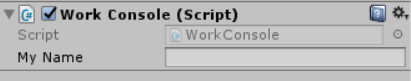
*Вывод в консоль:*

*Debug.Log("...") или print("...")*

**Задание 1**: в окне **Project** при помощи контекстного меню создайте новый скрипт. По умолчанию в скрипте определены функции Start и Update. В функции Start напишите вывод в консоль сообщения "Hello World", а в функции Update напишите вывод в консоль любого другого сообщения. Т.к. выполняются только те скрипты, которые присоединены к игровому объекту, то наш скрипт надо привязать к какому-либо объекту. Создайте на сцене пустой объект Create>Empty и перетяните на него написанный скрипт. Перейдите в консоль и запустите режим игры.

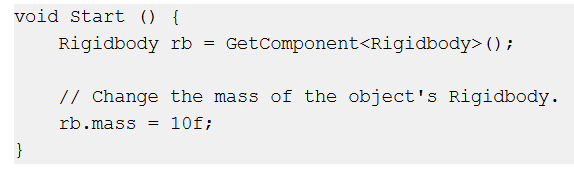
***Public переменные***

*Когда скрипт присоединяется к объекту он становится компонентом объекта и отображается на панели Inspector. При инициализации в скрипте переменных типа* ***pubic***  *в соответствующем компоненте на панели Inspector появляется редактируемое поле для ввода или изменения этой переменной, например поле My Name на рисунке. Unity позволяет изменять значение переменных скрипта в запущенной игре. Это очень полезно чтобы увидеть эффекты от изменений сразу же, без остановки и перезапуска.*



***Обращение к компонентам объекта***

*Для обращения к компонентам GameObject необходимо получить ссылку на экземпляр компонента, с которым вы хотите работать. Это делается с помощью функции GetComponent.*



***Перемещение объекта***

*Следующая запись мгновенно перемещает объект в новую точку, т.е. задается новая позиция:*

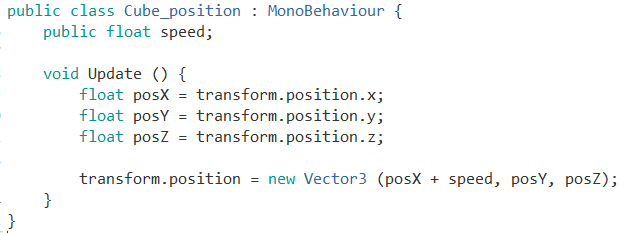
gameObject.GetComponent<**Transform**>().position = new Vector3(2,2,2)

можно записать в кратком виде

**transform**.position= new Vector3(2,2,2)

*Для плавного перемещения необходимо изменять координаты объекта на какое-либо приращение допустим каждый кадр. Для обозначения этого приращения объявим переменную* ***speed*** *типа* ***public****, чтобы иметь возможность изменять ее во время режима игры.*

**Задание 2:** создайте скрипт следующего вида и присоедините его к кубу на сцене. При запуске вводите различные значения переменной speed на панели **Inspector**, чтобы добиться нужной скорости.

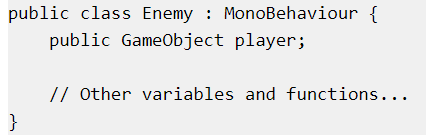


*Замечания: в устройствах с различной производительностью кадры будут меняться с разной частотой, поэтому, чтобы убрать зависимость скорости от производительности устройства, скорость рекомендуется умножать на переменную* ***Time.deltaTime****.*

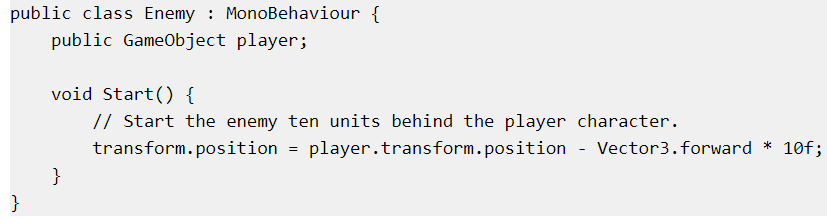
***Обращение к другим объектам***

*Один из способ - это связывание объектов через переменные*

*Добавим в скрипт переменную типа GameObject с уровнем доступа public:*

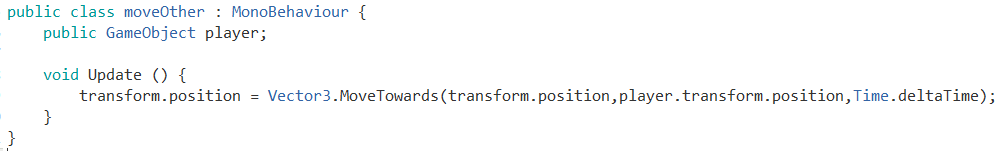


*Поле для ввода переменной появится в окне Inspector, и теперь нужно перетащить объект со сцены или из панели Hierarchy в это поле в окне Inspector. Теперь функция GetComponent и доступ к переменным компонента доступны для связанного объекта:*

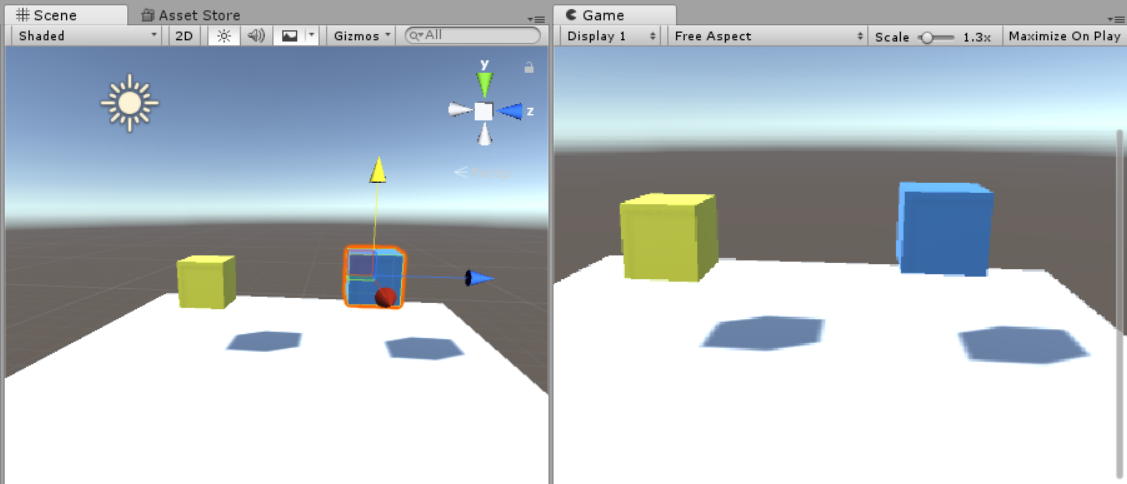


*Vector3 – структура, которая манипулирует тремя числами.*

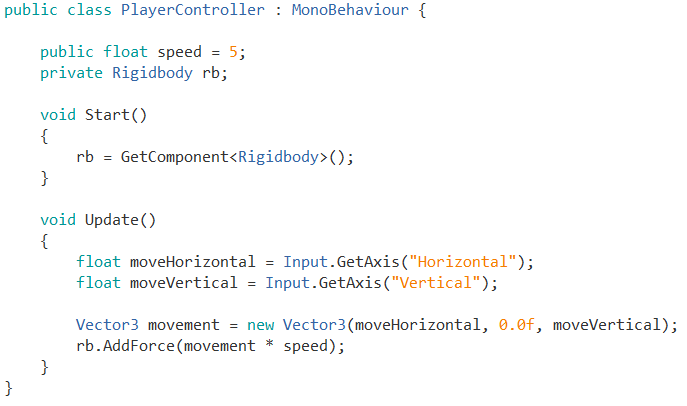
**Задание 3:** опробуйте приведенный ниже скрипт. Для этого разместите на сцене два куба, одному назначьте скрипт, а другой куб задайте в качестве переменной.

**

Для выполнения этого примера расположите окна Scene и Game как на рисунке, и в процессе выполнения двигайте куб-цель по сцене.



**Задание 5**. Следующий скрипт двигает объект стрелками на клавиатуре. Реализуйте его. Создайте новую сцену. На сцене разместите плоскость и сферу. Назначьте сфере скрипт.



**Задание 6.** В созданную в пятом задании сцену добавьте возможность изменения цвета сферы по нажатию на произвольную клавишу. Доступ к свойству **color** осуществляется через свойство **material** компонента **Renderer**. Функции обработки действий с клавиатуры находится в интерфейсе **Input** (см. документацию).

**Задание 7.** Добавьте в сцену куб, который будет «догонять» сферу, т.е. куб будет отслеживать позицию сферы и перемещаться в эту позицию.

**Задание 8.** Реализуйте следующий скрипт. Объясните код. Добавьте генерацию сфер по нажатию на ЛКМ (Input.GetMouseButtonDown(0)). Оптимизируйте скрипт, переместите в ф-ю **Start** те строки кода, которые по логике не должны быть в **Update**.

