**Лабораторная работа № 2**

***Задание для тех, кто знаком с основами Unity смотрите в пункте 2***

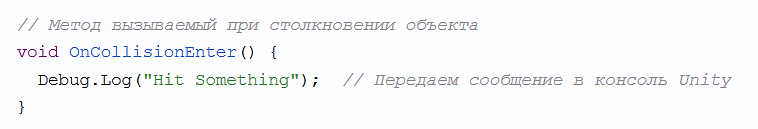
***Пункт 1:***

1. ***Обнаружение столкновений (collisions)***

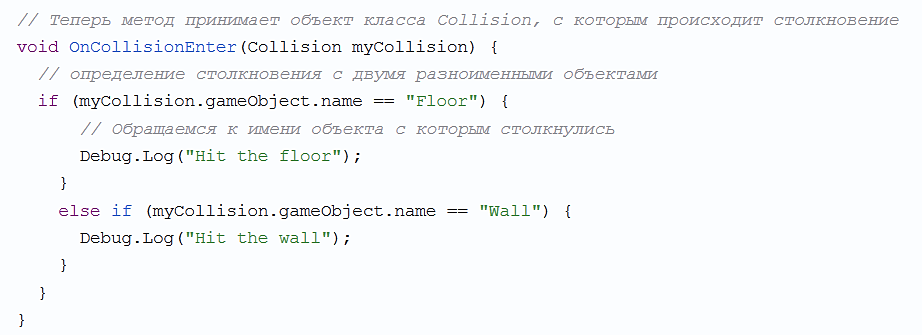
*Физический движок сообщит о столкновениях с объектом с помощью вызова функций событий в скрипте этого объекта. Функции OnCollisionEnter, OnCollisionStay и OnCollisionExit будут вызваны по началу, продолжению и завершению контакта.*

*В функцию передаётся параметр, предоставляющий дополнительную информацию о столкновении (координаты, “личность” входящего объекта и т.д.).*

*Пример показывает обнаружение столкновений*



*Пример показывает идентификацию объектов, с которыми произошло столкновение по параметру myCollision*



**Задание 1**

2.1 Создайте пол и две стены, в окне Hierarchy дайте этим компонентам соответствующие имена. Расположите над полом куб, слегка повернуть его, что бы при падении он падал на грань или угол, придайте ему упругость.

2.1. Запустите игру, куб должен ударяться об пол и стены

2.2 Реализуйте приведенный ранее пример с выводом сообщений в консоль.

1. ***Вращение***

*Следующий фрагмент кода вращает объект со скоростью 30 градусов в секунду, вокруг оси Y:*

transform.Rotate(Vector3.up \* 30 \* Time.deltaTime);

справка:<https://docs.unity3d.com/ru/530/ScriptReference/Transform.Rotate.html>

**Задание 2**

2.1 Разместите на сцене несколько кубов один над другим на некотором расстоянии друг от друга (как бы парящими в воздухе). Назначьте им компонент ***Rigidbody***, но сделайте так, чтобы на них не действовала сила гравитации. Запустите игру, кубы не должны падать.

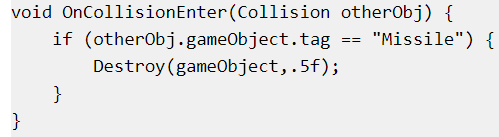
2.2 Назначьте кубам вращение (через скрипт). Итог – кубы парят в воздухе и вращаются.

2.3 Назначьте верхнему кубу скрипт, который при соприкосновении с объектом придаст объекту силу ***AddForce*** в горизонтальном направлении.

2.4. Включите у верхнего куба гравитацию, запустите игру. Кубы должны вылетать из-под верхнего **в сторону**.

1. ***Уничтожение объекта***

*Функция* ***Destroy*** *уничтожит объект после того, как загрузка кадра будет завершена или опционально после короткой паузы:*



*Второй параметр является не обязательным и указывает время, спустя которое будет уничтожен объект.*

*Функция Destroy может уничтожать отдельные компоненты без влияния на сам объект.*

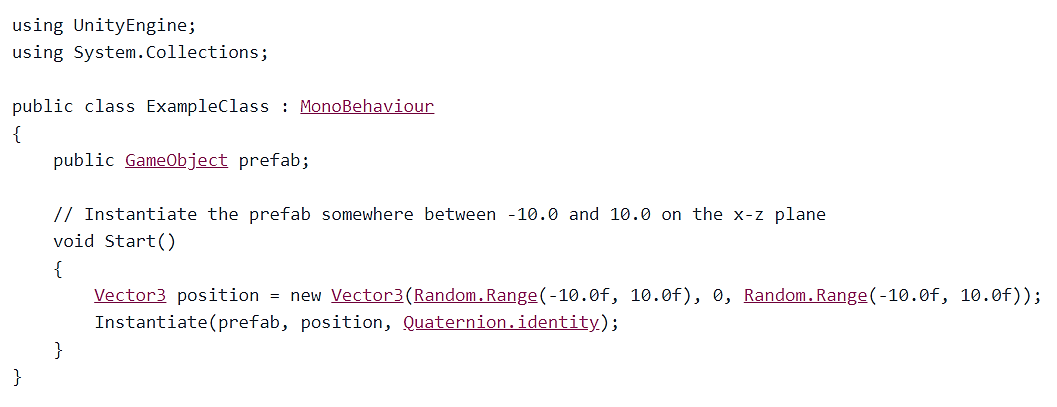
**Задание 3**

Создайте сцену с кубами, как в предыдущем задании, только теперь при соприкосновении верхнего куба с нижними, нижние кубы должны исчезнуть.

1. ***Создание объекта из существующего (или из префаба)***

*Игровой объект (GameObject) может быть создан при использовании функции* ***Instantiate****, которая делает копию существующего объекта. При копировании игрового объекта (GameObject), копируются все компоненты оригинального объекта.*

*Объект с которого берется копия не обязан присутствовать на сцене. Гораздо чаще используется* ***префаб****, который был объявлен в public переменной на панели Inspector. В следующем примере генерируются префабы в случайной точке между -10.0 и 10.0 на плоскости x-z. Данный скрипт необходимо назначить какому-либо объекту на сцене, либо, если такового нет, пустому объекту (GameObject> Create Empty).*



Код можно скопировать здесь: <https://docs.unity3d.com/ru/530/ScriptReference/Random.Range.html>

1. **Взаимодействие с триггером**

Объект *Rigidbody* можно сделать коллайдером, тогда он не будет обладать твердой границей и через него можно проходить. Взаимодействие обрабатывается методами OnTriggerEnter, OnTriggerStay и OnTriggerEnter.

Задание 5.

5.1. Создайте поверхность и сферу, управляемую клавишами (как в предыдущей лабораторной работе).

5.2. На поверхности разместите триггер в виде плоскости отличного от поверхности цвета.

5.3. Напишите скрипт в функции **FixedUpdate**, который при входе сферы в область триггера будет увеличивать размер сферы (можно менять другой параметр на свое усмотрение), а при повторном входе в зону триггера сфера уменьшится.

Задание 6 **Придумайте свою игру** с применением изученных функций (управляемое движение, различные взаимодействия, генерация объектов, уничтожение объектов и т.д.) или реализуйте предложенный вариант игры.

*Один из вариантов игры:*

* *Создайте поверхность с бортиками. На неё поместите управляемую клавишами капсулу.*
* *Создайте префаб (зеленый куб)*
* *Напишите скрипт, генерирующий кубы из этого префаба на поверхности в случайных координатах (см. код выше) при запуске игры. В итоге после запуска игры должна получиться поверхность с несколькими зелеными кубами, расположенными в случайном порядке*
* *Расположите на сцене несколько красных кубов, которые должны двигаться от стенки к стенке (при столкновении со стеной кубы меняют траекторию)*
* *Игровая логика: при соприкосновении капсулы с зелеными кубами, зеленые кубы должны исчезать, а при соприкосновении с красными кубами – исчезает сама капсула.*
* *Разместите на поверхности триггер, при входе сферы в который исчезнут красные кубы.*

**Пункт 2**

**Придумайте свою игру, в которой должны присутствовать следующие элементы:**

* Управляемый с клавиатуры объект (игрок)
* Генерация объектов при запуске в случайных местах сцены (например, стены или препятствия)
* На сцене помимо обычных объектов Rigitbody должны присутствовать объекты со статическими Rigitbody и триггеры;
* Механика игры должна включать различные результаты взаимодействий игрока с Rigitbody и с триггерами (например, при столкновении с одними скорость увеличивается, с другими –уменьшается, при входе в триггер генерируются новые объекты или изменятеся сам герой) (используйте методы OnTriggerEnter() и OnCollisionEnter() или их производные);
* Предусмотрите отсутствие взаимодействий между какими-либо группами объектов при помощи слоёв и матрицы столкновений (например, по сцене летают привидения, но взаимодействуют только с игроком, остальные объекты пролетают насквозь);