



#ШПАРГАЛОЧКИ

СОЗДАНИЕ ИГР UNITY

Материалы подготовлены отделом методической разработки

Больше полезных материалов и общения в нашем комьюнити
в Telegram: https://t.me/hw_school





```
[Header("Танк:")]  
[SerializeField] float speed;  
[SerializeField] float rotationSpeed;
```

```
rb.AddForce(transform.right * speed * v,  
ForceMode2D.Impulse);
```

```
rb.AddTorque(rotationSpeed * h,  
ForceMode2D.Impulse);
```

```
Bullet bul = Instantiate(bullet, bulletPoint.position,  
Quaternion.identity);  
bul.SetDirection(bulletPoint.right);
```

Создание заголовка для сериализованных полей в юнити.

Если у полей есть заголовок, то даже если они одного типа, их все равно нужно писать на отдельных строках.

Прикладываем импульсную силу к твердому телу объекта. Физически «пинаем» его направо (или налево) с заданной силой

Прикладываем крутящий момент к твердому телу объекта. Заставляем его вращаться вокруг своего центра с указанной скоростью.

Ссылку на инстанцируемый объект можно сохранять в переменной, когда мы хотим с ним еще что-то сделать.



`bul.transform.Rotate(transform.rotation.eulerAngles);` Поворачиваем инстанцированную пулю в том же направлении, куда смотрит танк. В `transform.rotation` хранится вращение объекта в кватернионах, но функция `Rotate()` принимает аргумент в градусах, так что используем `transform.rotation.eulerAngles`

`anim = GetComponentInChildren<Animator>();`

Функция получения компонента из игрового объекта, находящего внутри данного игрового объекта.

С этой функцией нужно быть осторожным, так как если у нескольких дочерних объектов есть компонент искомого типа – то получим только один из них случайным образом.

`if (collision.name.StartsWith("Brick"))`
`{`
`...`

Условие выполнится, если имя начинается с указанной в аргументе строки ("Brick")