**9. AIWeaponComponent**

1. Что мы хотим сделать в данном уроке для нашего NPC (при стрельбе)? Что создаем, от кого наследуемся?

2. Что нужно куда перенести у наших BaseWeapon и WeaponComponent – почему и для чего?

3. Какие две функции (закономерно со вторым пунктом) определяем в WeaponAIComponent (и какой ЗФ надо не забыть подключить)?

4. Что нам необходимо переопределить в нашем AICharacter – синтаксис?

1. Сделаем так, чтобы когда у нас у одного оружия кончались патроны, NPC переключался на следующее **заряженное** оружие. Сделаем это с помощью модификации STUWeaponComponent, создав его наследника STUAIWeaponComponent.

2. У **BaseWeapon** нам необходимо перенести функцию **IsAmmoEmpty**() из секции **protected** в **public**.

У WeaponComponent:

* сделаем функции StartFire и NextWeapon виртуальными, чтобы переопределить их в AIWeaponComponent;
* массив оружий, указатель на текущее оружие и индекс текущего оружия переносим в секцию protected из private, в результате сможем получить к ним доступ в AIWeaponComponent;
* то же самое делаем с функциями CanFire, CanEquip, EquipWeapon.

3. В WeaponAIComponent переопределим функции StartFire и NextWeapon, не забыв подключить ЗФ для BaseWeapon:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Логика StartFire проста – если оружие разряжено – меняем на другое, в противном случае стреляем.

В функции NextWeapon мы будем в цикле искать заряженное оружие до тех пор, пока либо не дойдем до текущего индекса снова (то есть обойдем весь массив), либо найдем заряженное оружие (IsAmmoEmpty какого-то оружия вернет false). Если после цикла индексы не равны – меняем оружие.

4. Осталось переопределить WeaponComponent в AICharacter (не забываем подключить ЗФ AIWeaponComponent):

Изображение выглядит как текст

Автоматически созданное описание