**8. FIRE SERVICE**

1. Какой создадим сервис в этом разделе (что будет реализовывать)?

2. Что добавляем нашему конструктору (по аналогии с FindEnemy-сервисом), только какая логика будет в функции?

3. Однако нам нужно подправить кое-что в нашем базовом классе оружия? Что, почему и на какое поведение?

4. Как настраиваем наше дерево решения теперь с новым сервисом?

1. Создадим сервис, отвечающий за стрельбу NPC. Так же наследуемся от BTService и назовем STUFireService.

2. Так же создаем конструктор класса (для определения имени), проперти для ключа с вражеским актором и переопределяем функцию TickNode (и подключаем 4 ЗФ):

Изображение выглядит как текст, снимок экрана, экран, серебряный

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Получили контроллер, блекборд.

Далее проверили, есть ли смысл стрелять вообще – с помощью проверки, есть ли ключ в EnemyActor. С одной стороны, данная проверка не нужна, так как мы поместим наш сервис в Sequence с названием Attack, а там уже есть декоратор с такой проверкой, но лучше делать класс заранее самодостаточным.

Далее мы получили WeaponComponent и вызываем функцию StartFire, если цель есть, и StopFire, если ее нет или она пропадает из области видимости.

3. Нам нужно подправить в BaseWeapon функцию GetPlayerViewPoint, так как в ней мы вызываем функцию контроллера GetPlayerViewPoint. Она находится в классе APlayerController, а в классе AIController ее не существует. В случае с NPC GetPlayerViewPoint (наша функция) вернет false, потому что GetPlayerController вернет nullptr:

Изображение выглядит как текст

Автоматически созданное описание

Изменим функцию и будем проверять, контролируется ли наш персонаж ИИ или живым игроком с помощью функции IsPlayerControlled. Если она возвращает true, то воспользуемся ранее написанной логикой, если же нет, то мы просто будем стрелять по направлению из дула:

Изображение выглядит как текст

Автоматически созданное описание

4. FireService можем добавить в ноду Selector, все будет корректно работать, потому что у нас есть проверка на EnemyActor, но логичнее, чтобы данный сервис не получал тик, когда у нас не идет атака, поэтому добавим к ноде Attack и ставим EnemyActor в EnemyActorKey:

