**19. Reload animation, anim montage, anim notify**

1. Какие у нас есть анимации для перезарядки? Что от них создали, как настроили? Добавили биндинг.

2. Перезарядки отличаются – встает вопрос – где их хранить? Какие были варианты и как мы решили хранить? Как теперь у нас хранятся оружия? А еще надо добавить один проперти в связи с добавлением нового параметра для оружий

3. В компоненте оружия надо подправить функцию спауна оружий – как?

4. В функции EquipWeapon так же внесем изменения – что нужно поменять после смены оружия? Какой есть плохой вариант и правильный установки данного значения?

5. Добавляем функцию для биндинга перезарядки и пишем тело функции и биндим. Остается только настроить в блюпринтах наш массив оружий

6. У нас опять наблюдается проблема, как и для смены оружия. Создаем новый нотифай, а так же базовый, для чего, как переносится куда туды-сюды?

7. Аналогично смене оружия, добавляем проперти и две функции, какой в них код и как меняются существующие функции для смены оружия?

8. Переходим в функцию InitAnimation – что нам надо тут сделать (по аналогии с уже имеющимся функционал), как мы решаем написать функцию для поиска нужного нотифая?

1. У нас есть две анимации – для каждого типа оружия. Создадим из обоих AnimMontage’ы и заменим имя слота на UpperBody (по аналогии с перезарядкой).

А так же забиндим клавишу R на смену анимации и назовем биндинг Reload.

2. У нас перезарядки лаунчера и винтовки отличаются. Встает вопрос – где их хранить? С точки зрения ООП хранить его в самих оружиях неправильно, так как это анимация персонажа. Можем создать проперти в BaseWeapon и указать в соответствующих BP нужную нам анимацию. Поэтому мы добавим в WeaponComponent дополнительную структуру WeaponData и в данной структуре будет хранить класс оружия и анимацию перезарядки.

Изображение выглядит как текст

Автоматически созданное описание

И теперь будем хранить не массив оружий, а массив таких структур, из которых мы будем брать всю необходимую нам информацию:

Изображение выглядит как текст, внутренний, снимок экрана

Автоматически созданное описание

А так же добавим еще одно проперти – указатель на AnimMontage, который будет указывать на текущую анимацию перезарядки (чтобы когда мы будем менять оружие, менять и анимацию).

Изображение выглядит как текст

Автоматически созданное описание

3. В компоненте оружия надо подправить функцию спауна оружий:

Изображение выглядит как текст

Автоматически созданное описание

4. В функции EquipWeapon так же внесем изменения – после того, как мы меняем оружие, нам надо поменять анимацию перезарядки. Можно было бы брать по индексу из массива наш элемент и обращаться к полю с AnimMontage. Это сработает, так как создавали акторов просто проходя по массиву WeaponData и добавляли их в массив Weapons. Однако никто не гарантирует, что на не понадобится сортировка в будущем одного из этих массивов и тогда мы будем брать анимацию не для того оружия. Поэтому лучше искать нужный нам класс и брать анимацию, если нашли:

Изображение выглядит как текст

Автоматически созданное описание

5. Добавляем функцию для биндинга перезарядки:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

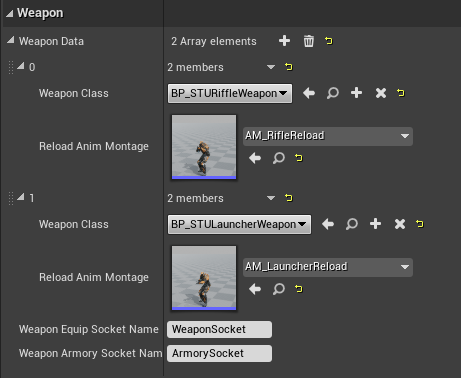
Автоматически созданное описание

И биндим ее (в классе персонажа):

Изображение выглядит как текст

Автоматически созданное описание

Остается только настроить наш массив WeaponData у блюпринта нашего персонажа в компоненте оружия.



6. У нас опять наблюдается проблема – если мы перезаряжаемся, то мы можем снова запустить анимацию перезарядки и стрелять.

Создадим Notify (как и для Equip) для маркировки конца перезарядки (STUReloadFinishedAnimNotify), только создадим еще базовый класс (STUAnimNotify), в который перенес функционал из старого Notify для Equip.

Соответственно, нужно будет перенести делегат и переопределение функции Notify в базовый класс и изменить наследование у AnimNotify для Equip.

Мы все это отплесали для того, чтобы не дублировать функциональность, связанную с делегатом. Теперь оба Notify умеют получать клиента, который подписался на делегата OnNotified.

7. Аналогично тому, как мы делали при смене оружия, нужно сделать все блокировке при перезарядке (добавляем флаг, callback и CanReload):

Изображение выглядит как текст

Автоматически созданное описание

Функционал OnReloadFinished аналогичен функции OnEquipFinished:

Изображение выглядит как текст

Автоматически созданное описание

В CanReload надо добавить переключение флага и проверку:

Изображение выглядит как текст

Автоматически созданное описание

В CanFire и CanEquip добавляем дополнительное условие (то есть мы не можем стрелять/менять оружие, когда у нас идет перезарядка:

Изображение выглядит как текст

Автоматически созданное описание

Функция CanReload будет совпадать с функцией CanFire:

Изображение выглядит как текст

Автоматически созданное описание

8. Переходим в функцию InitAnimation – нам осталось добавить callback на все нотифаи, которые находятся в анимациях перезарядки. Поиск будет абсолютно такой же. Поэтому, чтобы его не дублировать, можем обернуть данный код в собственную шаблонную функцию в компоненте оружия:

Изображение выглядит как текст

Автоматически созданное описание

Тогда в InitAnimations код станет:

Изображение выглядит как текст

Автоматически созданное описание

Первая часть относится к биндингу на делегат окончания смены оружия, а вторая – к биндингу на делагат окончания перезарядки (так как у нас массив, то нужно пройти по всем его элементам).

Остается только добавить соответствующие Notify в AnimMontage для обеих перезарядок.