**4. Ammo Pickup**

1. Добавляем два проперти – для его (логика)? В cpp подключаем 3 ЗФ и определяем поведение функции GivePickupTo, что она делает? Для чего проверять на то, мертв персонаж или нет?

2. Добавляем в WeaponComponent функцию – тут какая логика?

3. Добавляем в BaseWeapon функцию, к которой обращались в WeaponComponent – тут какая логика (тут три ветвления)? Добавили еще функцию IsAmmoFill – какая логика у нее и для чего?

4. Что надо не забыть настроить в BP (подсказка: первый пункт)? Какое поведение наблюдается в четырех случаях: с полным арсеналом, с неполным магазином, с неполным магазином и обоймами, с пустым магазином и 0 обойм?

5. Однако у нас наблюдается одна проблема при отстреле всего и замене на другое оружие – какой? Как исправили это поведение (делегат + колбек (логика последнего))?

1. Добавляем в AmmoPickup проперти, которое будет отвечать за то количество магазинов, которые получит персонаж при подборе нашего пикапа, а второе – тип оружия, по нему будем определять, к какому оружию нужно добавлять данные магазины.

Изображение выглядит как текст

Автоматически созданное описание

В cpp подключаем 3 ЗФ. В функции нашего пикапа GivePickupTo нам надо будет добавить количество магазинов соответствующему типу оружия.

Изображение выглядит как текст

Автоматически созданное описание

Проверка на жив или нет нужна для фильтрации события, когда персонаж мертвый упадет на пикап.

Далее получаем компонент оружия и, если он есть, то вызываем будущую функцию и передаем туда количество магазинов и тип оружия.

2. Добавляем в WeaponComponent эту функцию. Функция будет возвращать булевое значение, потому что даже при наличии оружия и живого персонажа у нас не получится добавить арсенал определенному оружию:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

В данной функции уже вызываем TryToAddAmmo самого оружия.

3. Добавляем данную функцию (в public) для BaseWeapon и функцию (в секцию protected) для проверки заполненности нашего арсенала:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

В функции TryToAddAmmo проверка ClipsAmount <= 0 нужна из-за того, что она открытая и ее можно вызвать из любого клиента, который использует наш класс (а не только из нашего пикапа).

В первом случае (IsAmmoEmpty) мы будем проигрывать анимацию.

Во втором случае у нас возможно несколько вариантов: нам нужно проверить, сможем ли мы с помощью значения ClipsAmount пополнить как магазин в боеприпасе, так и текущий магазин. Для этого создали дополнительную локальную переменную. Если у нас разница больше 0, значит мы не сможем заполнить наш текущий магазин, в противном случае – можем.

В последнем случае мы просто дополняем нашу текущую неполную обойму до максимального значения патронов.

4. Не забываем настроить у AmmoPickup WeaponType в тип нашего лаунчера.

Если мы пересекаем с полным арсеналом – ничего не происходит.

Если будет неполный текущий магазин – у нас восстанавливается значение патронов до полного.

Если будет неполный текущий магазин и одна целая обойма – пополнятся оба значения.

Если отстрелять все, то у нас заполнится 0/6, затем сразу (мы даже не увидим 0/6) станет 3/5 и проиграется анимация перезарядки (только в этом случае).

5. Однако у нас наблюдается одна проблема – если мы расстреляем весь боекомплект (и патроны, и обоймы), возьмем другое оружие и пробежим через наш пикап, то у нас при переключении обратно будет значение 0/6 и перезарядки не будет.

Это происходить из-за того, что в WeaponComponent мы всегда перезаряжаем CurrentWeapon. Чтобы это исправить, добавим к нашему делегату еще один параметр ASTUBaseWeapon\* - то есть при вызове Broadcast делагата мы будем передавать оружие, которое хотим перезарядить:



Broadcast’ы делегата и сигнатуру колбека надо поменять:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание



Изображение выглядит как текст

Автоматически созданное описание

Логика колбека такая – если текущее оружие — это то, которое мы хотим перезарядить – делаем это как обычно (вызывая анимацию и т.д.), если же нет – то вызываем функцию перезарядки самого оружия (она не проигрывает анимацию).