**3. Take Damage**

1. Функция для получения урона в UE, какие аргументы принимает, подробно о втором аргументе (какие виды есть)

2. Как получить информацию о том, что урон был нанесен персонажу – ищем ответы внутри функции для получения урона, какие там есть делегаты, чем отличаются, какой мы используем?

3. Находим сигнатуру делагата и создаем функцию, что принимают

4. Где и как подписываемся на наш делегат

5. Переносим функцию и подписываемся на делегат (тут поподробнее как это сделать) в компоненте здоровья

1. В UE существует специальный набор функций, с помощью которых можно осуществлять нанесение урона.

Начнем с функции TakeDamage(). Будем вызывать данную функцию в Tick(), то есть персонаж будет наносить урон сам себе.

Изображение выглядит как текст

Автоматически созданное описание

Первый аргумент – сколько урона мы хотим передать.

Второй аргумент – DamageEvent – передадим объект с дефолтным конструктором.

Через F12 попадаем во внутреннюю кухню, данная структура отвечает за дополнительную информацию, которую можно передать с уроном.

У него есть специальная PROPERTY DamageType, с помощью которой мы можем определять какого типа урон был нанесен персонажу (взрыв, огонь, лава и т.д.) и в зависимости от данной информации персонаж может издавать соответствующие звуковые эффекты, проигрывать специальные анимации.

Можно унаследоваться от данного DamageEvent и создать свой собственный. В UE есть два более конкретных DamageEvent. Первый – это PointDamageEvent, с помощью него можем описывать точечный урон персонажу, будь то пуля, режущее оружие, то есть мы можем передать информацию, в какую часть тела был нанесен данный урон, какое было направление, и в зависимости от данных параметров мы можем физически верно проигрывать анимацию.

Второй тип DamageEvent называется RadialDamageEvent – он описывает урон, который может прийти от взрыва гранаты (например). В данном ивенте мы можем передать параметры сферы – ее положение в пространстве, радиус и каким образом подсчитывается финальный ущерб персонажу.

Третий параметр – указатель на контроллер персонажа, **КОТОРЫЙ** наносят ущерб. Передача данного параметра важна, потому что с помощью этого указателя мы можем определить, находится ли игрок в противоположной команде (либо в нашей) – то есть можем получить информацию о персонаже, который нанес нам ущерб. В данном случае мы передаем наш собственный контроллер, который находится в переменной Controller.

Последний параметр – указатель на актора, который ответственен за ущерб – это может быть граната, пуля, колющее/режущее оружие и т.д. В данном случае передаем this.

2. Теперь нам надо разобраться, как получить информацию о том, что урон был нанесен текущему актору. Для этого прыгаем в TakeDamage через F12. А далее в Super::TakeDamage (функцию актора). В данном функции в зависимости от ивента, который приходит в TakeDamage, происходит проверка, является ли данный ивент PointDamage ивентом и затем вызывается делегат OnTakePointDamage, на который мы можем подписаться.

Изображение выглядит как текст

Автоматически созданное описание

В нашем случае мы вызываем функции с просто Damage ивентом, в данную ветку мы не попадем. Далее проверяется, является ли ивент RadialDamage ивентом, и, если является, вызывается делегат OnTakeRadialDamage. Мы можем на него подписаться и получать соответственный callback при возникновении данного события. В нашем случае мы опять сюда не попадаем.

Изображение выглядит как текст

Автоматически созданное описание

И наконец, в самом конце функции, вызывается делегат OnTakeAnyDamage, данное событие вызовется при любом раскладе – то есть при любом типе ивента. Как раз на него мы и будем подписываться.

Изображение выглядит как текст

Автоматически созданное описание

3. Через F12 попадаем к определению делегата OnTakeAnyDamage. Видим, что в заголовочном файле объявлены 3 делегата. Плюс они являются UPROPERTY и мы можем подписаться на них в блюпринтах.

Изображение выглядит как текст

Автоматически созданное описание

Находим сигнатуру нашего делагата и копируем ее, в класс нашего персонажа в ЗФ пишем функцию OnTakeAnyDamageHandle, в которую и копируем всю нашу сигнатуру:

Изображение выглядит как текст

Автоматически созданное описание

Первый параметр – актор, которому нанесен урон, второй – количество урона, третий – DamageType – то есть в callback’е приходит не весь ивент, а лишь тип урона, далее контроллер, ответственный за ущерб, и в конце – актор, который нанес ущерб.

То есть что мы тут сделали – это взяли сигнатуру делегата, а ей должна соответствовать сигнатура нашей функции, и вставили данную сигнатуру в определении нашей функции.

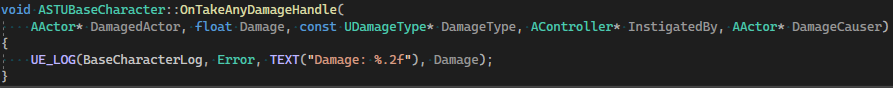
4. Далее в BeginPlay() подписываемся на наш делегат:

Изображение выглядит как текст

Автоматически созданное описание

И создаем локальную категорию логирования, чтобы проверить работу функции:





Получается, что мы на каждом тике наносим себе небольшой урон, из-за чего вызывается функция, подписанная на наш делегат OnTakeAnyDamage.

5. Теперь мы можем перенести данную функцию в HealthComponent и подписаться на делегат OnTakeAnyDamage внутри HealthComponent. Таким образом вся логика обработки здоровья персонажа будет находиться в одном месте (тут просто перенесли функцию в наш компонент). (**НЕ ЗАБЫТЬ убрать подписку на делегат в классе персонажа**!)

Теперь подпишемся на делегат OnTakeAnyDamage в BeginPlay(). У каждого компонента существует функция GetOwmer(), которая возвращает указатель на владельца компонента.

И реализуем простой декремент здоровья в функции.

Изображение выглядит как текст

Автоматически созданное описание