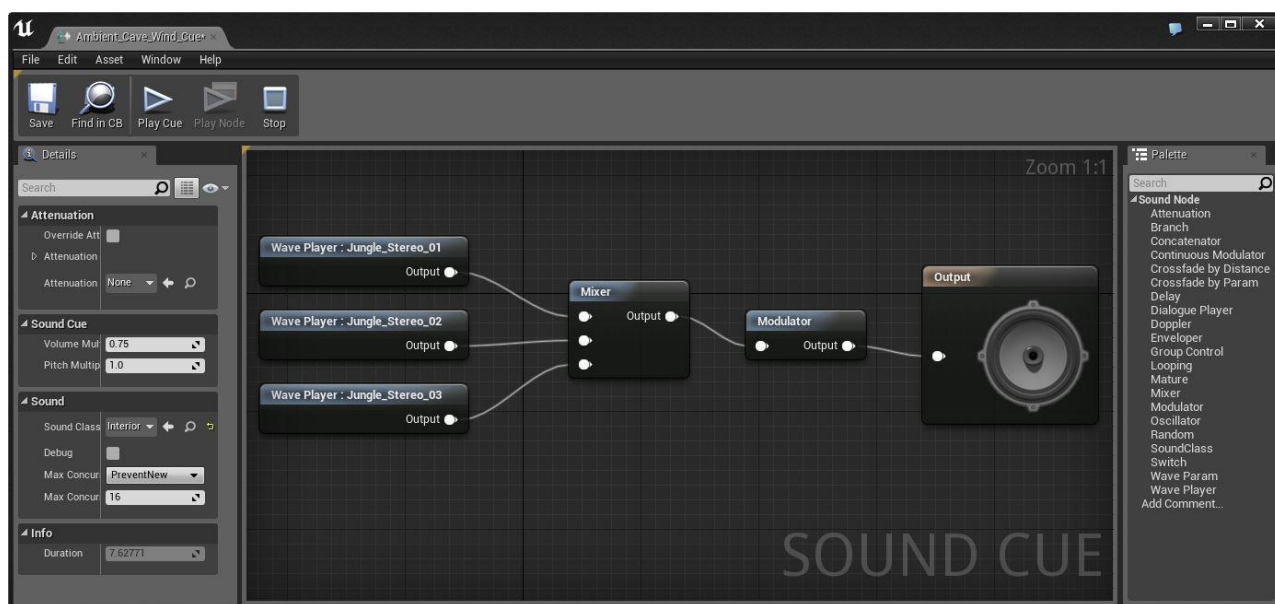


Тutorial по Unreal Engine. Часть 7: звук



В видеоиграх под понятием «звук» подразумеваются музыка, диалоги и звуковые эффекты. В наше время игра без звука будет казаться небрежной и незавершённой.

Кроме того, звук позволяет усилить погружение игрока в игру. Музыка провоцирует эмоциональный отклик. Диалоги раскрывают персонажей и сюжет. Звуковые эффекты обеспечивают обратную связь и правдоподобность. Все они могут превратить хорошую игру в отличную.

В этой части tutorials вы научитесь следующему:

- Воспроизводить музыку и заикливать её
- Воспроизводить звуковой эффект в определённых точках анимации
- Модулировать тон звука при каждом воспроизведении
- Перемещать их регулировать громкость звука в зависимости его местоположения в 3D-пространстве
- Управлять с помощью UI громкостью музыки и звуковых эффектов по отдельности

Приступаем к работе

Скачайте <https://koenig-media.raywenderlich.com/uploads/2017/10/SkywardMuffinAudioStarter.zip> и распакуйте её. Откройте проект, перейдя в папку проекта и запустив *SkywardMuffin.uproject*.

Нажмите на *Play*, чтобы начать игру. Цель игры — коснуться как можно больше облаков, не упав вниз. Нажмите на *левую клавишу мыши*, чтобы подпрыгнуть к первому облаку.

Игра расслабляет, не так ли? Чтобы усилить чувство расслабления, мы начнём с того, что будем воспроизводить спокойную пианинную музыку.

Воспроизведение музыки

Перейдите в Content Browser и найдите папку *Audio*. Здесь находятся все звуки, которые будут использоваться в этом tutorialе. Их можно прослушать, *наведя курсор* на их *значок*, а затем нажав на появившийся *значок воспроизведения*.



Воспроизвести музыку просто: достаточно перетащить звуковой ассет во Viewport. Однако музыка воспроизведётся только *один раз*. Зацикленность необходимо включать в ассете вручную. *Дважды щёлкните* на *S_Music*, чтобы открыть его.



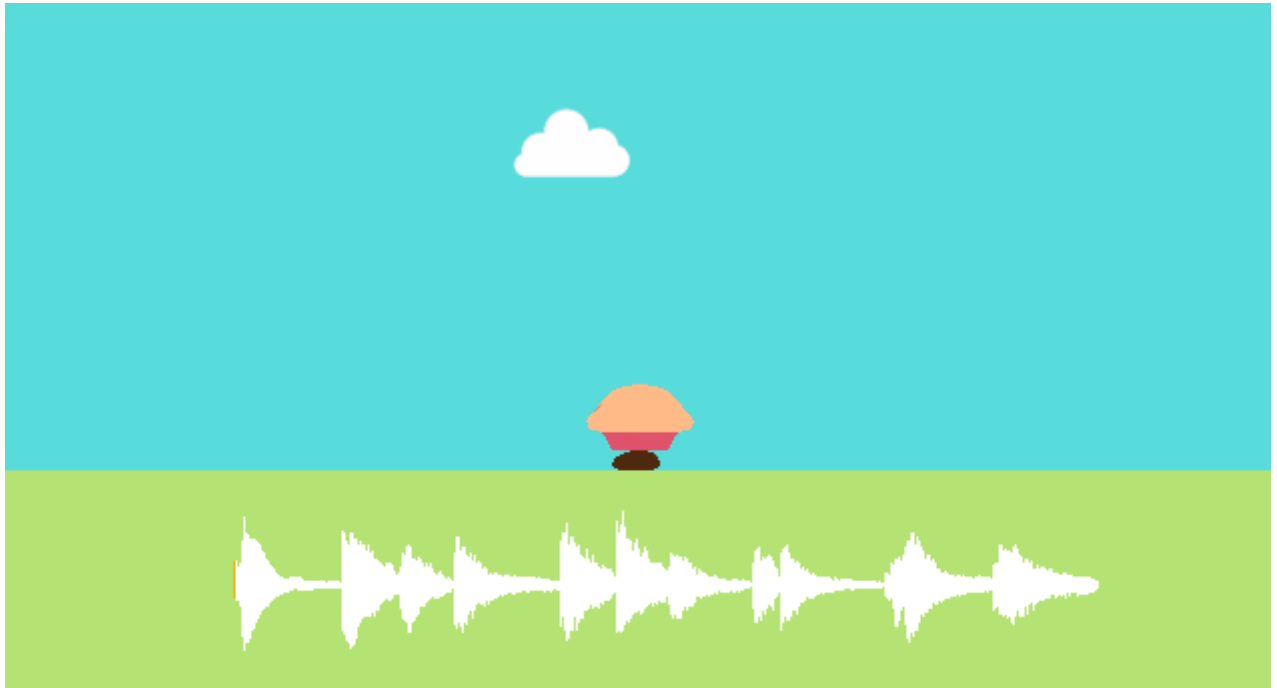
Откроется новое окно с единственной панелью Details. Перейдите в раздел *Sound Wave* и включите *Looping*.



Затем вернитесь в основной редактор и *перетащите* ассет *S_Music* во Viewport.

При этом будет создан актер *AmbientSound* с источником звука *S_Music*. Этот актер будет автоматически воспроизводить *S_Music* при запуске игры.

Нажмите на *Play*, чтобы прослушать музыку. Через 17 секунд (длительность музыки) она заиклится и начнётся сначала.



Теперь нам нужно создать звуковой эффект для шагов маффина. Для этого мы будем использовать *Animation Notify*.

Что такое Animation Notify?

Animation Notify позволяет разработчику запустить выполнение события в определённой точке анимации. Их можно использовать множеством различных способов. Например, можно создать *Notify* для спауна эффекта частиц.

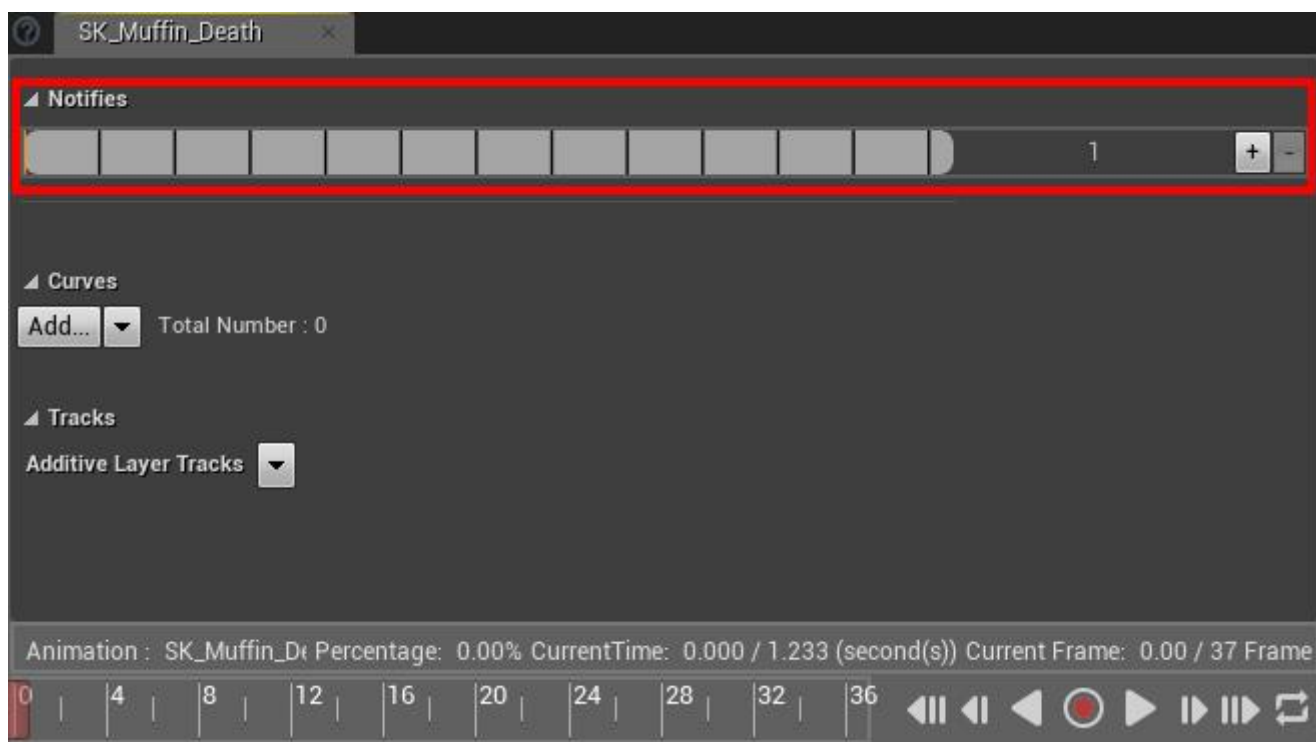
В этой игре при касании маффином земли появляется кнопка перезапуска. Однако с помощью *Notify* можно сделать так, чтобы она появлялась в *конце* анимации смерти.

В этом tutorialе мы будем использовать *Animation Notify* для воспроизведения звука, когда каждая из ног маффина касается земли.

Создание Animation Notify

Перейдите в папку *Characters\Muffin* и откройте *SK_Muffin_Walk*. При этом откроется *Animation editor*.

В панели под Viewport есть область под названием *Notifies*. Светло-серая область — это *Notify Track*. Именно здесь мы будем создавать Notify и управлять ими.



Ноги маффина касаются земли в *кадре 10* и в *кадре 22*, поэтому нам нужно создать Notify для каждой из этих точек. Для создания Notify *нажмите правой клавишей мыши* на Notify Track и выберите *Add Notify\Play Sound*. При этом создастся Notify с названием *PlaySound*.



Теперь нужно позиционировать Notify, чтобы он происходил в *кадре 10*.

Перемещение Animation Notify

Немного сложно понять, куда двигать Notify, потому что на Notify Track не показано, где находится *кадр 10*. Однако можно отобразить с помощью Timeline маркер.

Во-первых, зайдите в Timeline, расположенную в нижней части панели. *Перетащите* красный указатель воспроизведения и отпустите, когда *Current Frame* будет равен *10*. Сначала необходимо поставить его на паузу!

Теперь на Notify Track появится красная линия, сообщающая, где находится указатель воспроизведения.

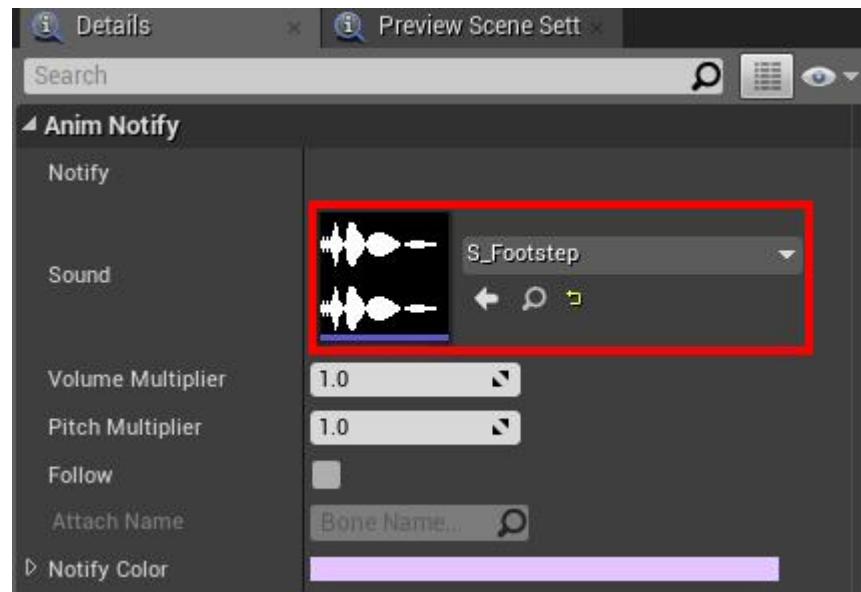


Перетащите *Notify PlaySound* и отпустите его, когда он привяжется к красной линии.

Теперь нам нужно сказать *Notify*, что надо воспроизвести звук шагов.

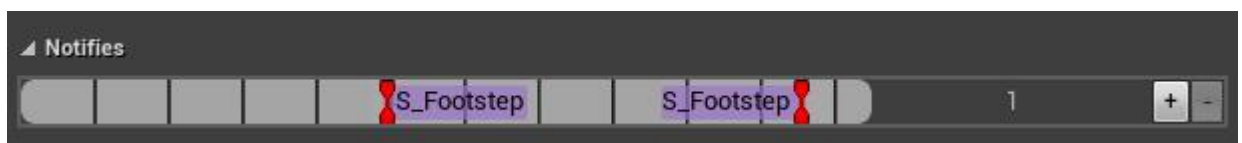
Воспроизведение звука шагов

Нажмите левой клавишей мыши на *PlaySound*, чтобы выбрать его и перейдите в панель *Details*. В разделе *Anim Notify* задайте для *Sound* значение *S_Footstep*.



Затем повторите процесс для следующей ноги. Сделайте следующее:

- Создайте ещё один *Notify Play Sound*
- Переместите *Notify* на *кадр 21*
- Выберите для *Notify* звук *S_Footstep*



Теперь, когда анимация ходьбы будет достигать *кадра 10* и *кадра 21*, *Notify* будет срабатывать и запускать воспроизведение звука *S_Footstep*.

Закройте *SK_Muffin_Walk* и вернитесь в основной редактор. Нажмите на *Play* и начните ходить по земле, чтобы слышать шаги.

Прослушав несколько раз звуки шагов, игроки кое-что заметят. Это каждый раз один и тот же звуковой эффект! Если бы у нас был способ немного варьировать каждый раз звук...

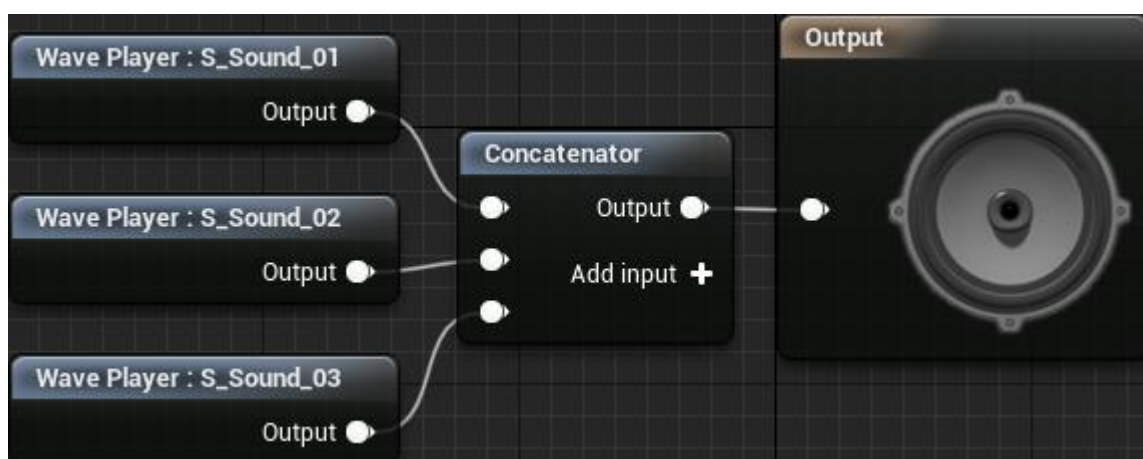
В следующем разделе мы будем воспроизводить звуковой эффект, когда игрок касается

облака. Но в этот раз мы каждый раз будем вносить вариации в звуки с помощью *Sound Cue*.

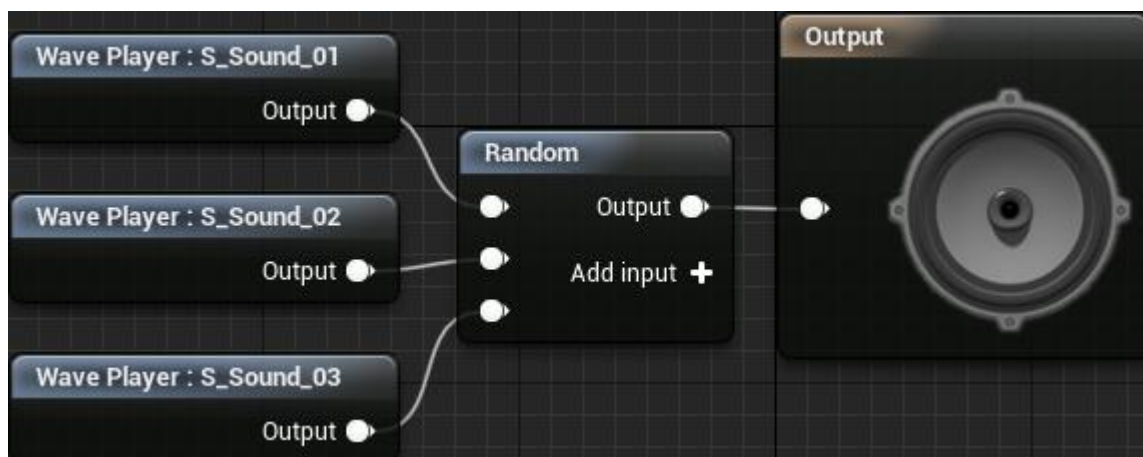
Что такое Sound Cue?

Sound Cue — это ассет, который позволяет манипулировать несколькими звуками и комбинировать их. Можно считать Sound Cue своим собственным звуком. Там, где можно использовать обычный звук, вы можете использовать вместо него Sound Cue.

Вот пример Sound Cue с использованием *Concatenator* для последовательного воспроизведения трёх разных звуков:



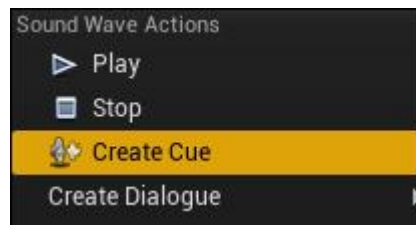
Если использовать вместо него нод *Random*, то вы сможете выбирать случайный звук каждый раз при воспроизведении Sound Cue.



В этом tutorialе мы создадим и будем использовать Sound Cue для изменения тона звука.

Создание Sound Cue

Сначала вернитесь в папку *Audio*. Там есть звуковой эффект *S_Pop*, для которого мы создадим Sound Cue. Чтобы сделать это, нажмите правой клавишей мыши на *S_Pop* и выберите *Create Cue*.



При этом будет создан новый ассет Sound Cue под названием *S_Pop_Cue*. Дважды щёлкните на *S_Pop_Cue*, чтобы открыть его в Sound Cue editor.



Примечание: этот редактор очень похож на редактор материалов, поэтому я не буду подробно его рассматривать. Если вы незнакомы с редактором материалов, то можете узнать о нём в части [Знакомство с движком](#).

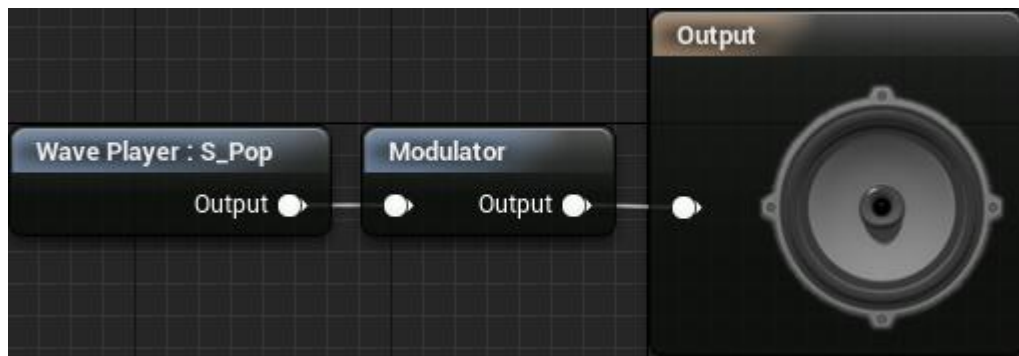
В графе есть два нода: *Wave Player: S_Pop* и *Output*. Sound Cue будет воспроизводить всё, что присоединено к ноду *Output* (в этом случае он воспроизводит звук *S_Pop*). Прослушать Sound Cue можно, перейдя в Toolbar и нажав на *Play Cue*.



Теперь нам нужно узнать, как изменять тон звука.

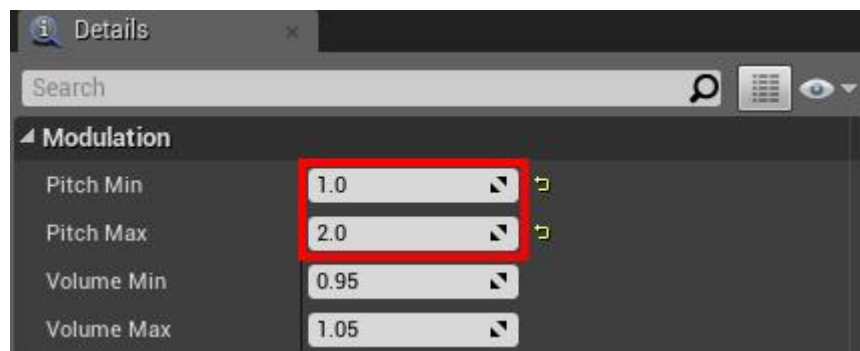
Изменение тона звука

Чтобы изменить тон звука, нам нужен нод *Modulator*. Создайте его и соедините следующим образом:



Теперь необходимо задать величину изменения тона. Выберите нод *Modulator* и перейдите в панель *Details*. Там есть два относящихся к тону поля: *Pitch Min* и *Pitch Max*. Значения меньше 1 показывают, что тон можно уменьшить, и наоборот. Значение 1 показывает, что тон останется прежним.

В этом tutorialе мы будем только повышать тон. Задайте *Pitch Min* значение *1.0*, а *Pitch Max* — значение *2.0*.



Теперь при каждом воспроизведении Sound Cue звук будет иметь тон в интервале от исходного тона до удвоенного исходного тона.

Теперь нам нужно сделать так, чтобы Sound Cue воспроизводился, когда игрок касается облака.

Воспроизведение Sound Cue

Вернитесь к основному редактору и перейдите в папку *Blueprints*. Откройте *BP_Cloud*, а затем функцию *CloudTouched*. Эта функция выполняется тогда, когда игрок касается облака, то есть это идеальное место для воспроизведения Sound Cue.

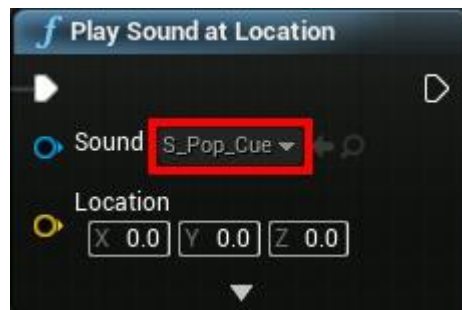
Здесь есть два нода, которые можно использовать для воспроизведения звука:

- *Play Sound 2D*: воспроизводит звук без какого-либо затухания или пространственной локализации (я расскажу о них подробнее ниже). Используйте этот нод для звуков, которые не «существуют» в игровом мире, например, для музыки и звуков UI.
- *Play Sound at Location*: воспроизводит звук в точке 3D-пространства. Используйте этот нод, если хотите, чтобы звук менялся в зависимости от местонахождения и направления игрока.

Поскольку облако существует в игровом мире, звук тоже должен существовать в игровом мире. Добавьте в конец цепочки нодов нод *Play Sound at Location*.



Затем задайте для *Sound* значение *S_Pop_Cue*.



Теперь когда игрок касается облака, будет воспроизводиться *S_Pop_Cue*.

Нажмите на *Compile* и вернитесь в основной редактор. Нажмите на *Play* и начните играть в игру. При каждом касании облака вы должны слышать одинаковый звук с разным тоном.

Тон звука меняется, но непохоже, что он звучит в 3D-пространстве. Для этого необходимо *позиционировать* звук.

Что такое позиционирование?

Позиционирование — это процесс, выполняемый для создания впечатления существования звука в 3D-пространстве. Звуки, приходящие слева, будут слышны левым ухом, и наоборот.

Кроме того, что позиционирование усиливает ощущения погружения в игру, оно также может помочь геймплею. В соревновательных играх наподобие *Overwatch* и *Counter-Strike*, позиционируемый звук помогает игрокам распознавать положение других игроков.

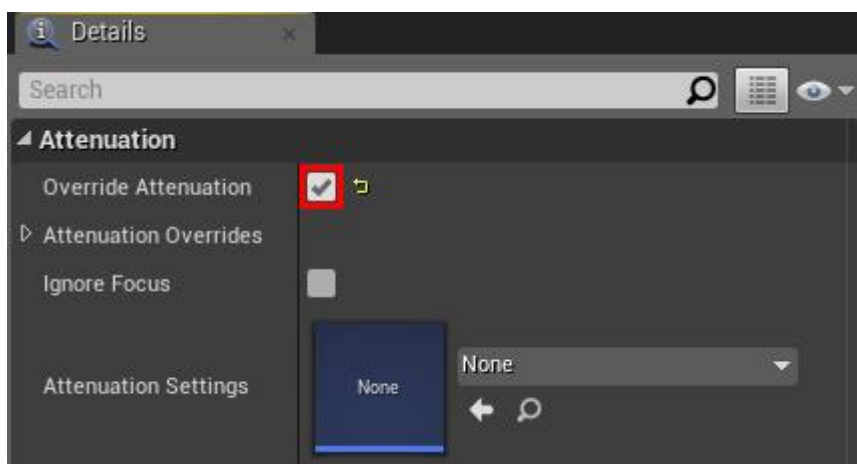
В этом tutorialе мы воспользуемся позиционированием для перемещения звука облака на основании его местоположения.

Создаём позиционирование

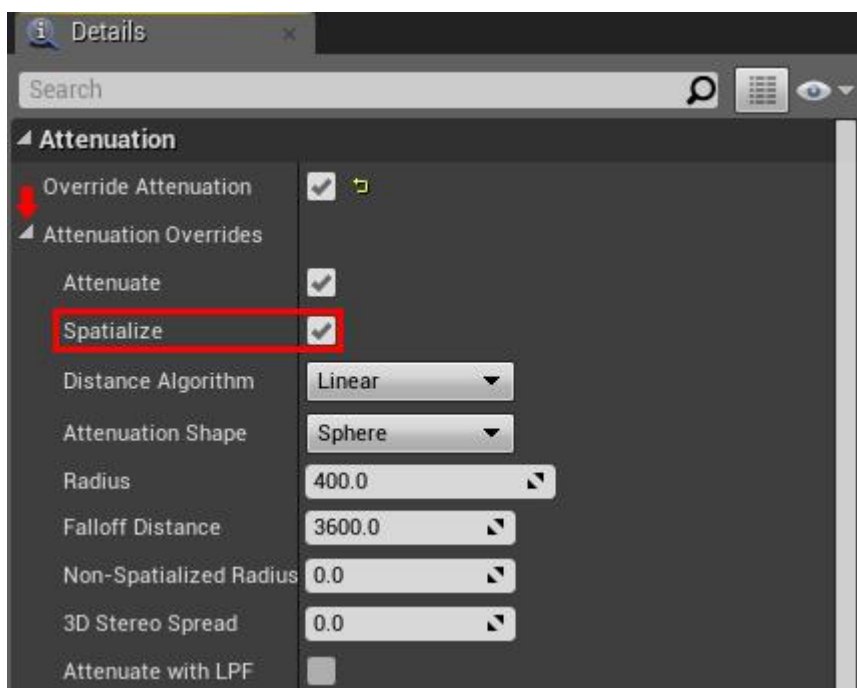
Есть два способа создания позиционирования Sound Cue:

- *Accem Sound Attenuation*: этот ассет содержит настройки, связанные с затуханием и позиционированием. Можно назначать этому ассету различные звуки, чтобы все они имели одинаковые настройки.
- *Override Attenuation*: вместо ассета Sound Attenuation можно задавать настройки внутри Sound Cue. Это позволяет создавать настройки для отдельных Sound Cue.

В этом руководстве мы воспользуемся вторым способом. Откройте *S_Pop_Cue* и перейдите в панель Details. Найдите раздел *Attenuation* и включите *Override Attenuation*. При этом станет активным раздел *Attenuation Overrides*.



Чтобы убедиться в том, что позиционирование включено, нажмите на *стрелку* рядом с *Attenuation Overrides*. Настройка *Spatialize* показывает, что позиционирование звука выполняется.



Мы закончили с настройками, поэтому можно закрыть *S_Pop_Cue*. Теперь нам нужно указать, где находится звук в 3D-пространстве.

Воспроизведение звука в 3D-пространстве

Откройте *BP_Cloud* и создайте нод *GetActorLocation*. Теперь соедините его с контактом *Location* нода *Play Sound at Location*.



Теперь звук будет воспроизводиться в том же месте, где находится облако.

Нажмите на *Compile* и вернитесь в основной редактор. Нажмите на *Play* и начните касаться облаков. Вы должны услышать звуки, которые будут казаться приходящими из разных мест.

Примечание: по умолчанию слушателем звука является камера. Это означает, что вы будете слышать звуки с точки местоположения камеры. Если вы хотите заменить слушателя другим актором, то можно использовать нод [Set Audio Listener Override](#).

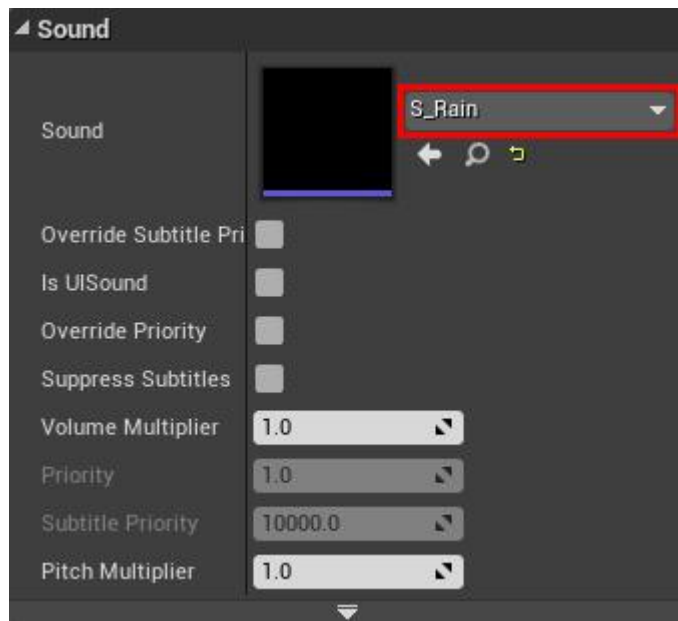
Вы наверно заметили, что из некоторых облаков идёт дождь. Но они не похожи на тучи, потому что звука дождя не слышно! Сейчас мы добавим звук дождя и используем *затухание*, чтобы изменять его громкость в зависимости от расстояния.

Добавление звука дождя

Вместо использования нода для воспроизведения звука дождя можно применить компонент *Audio*. Одно из преимуществ использования компонента заключается в том, что он автоматически воспроизводится в точке местоположения облака.

Откройте *BP_Cloud* и перейдите в панель *Components*. Добавьте новый компонент *Audio* и назовите его *RainAudio*.

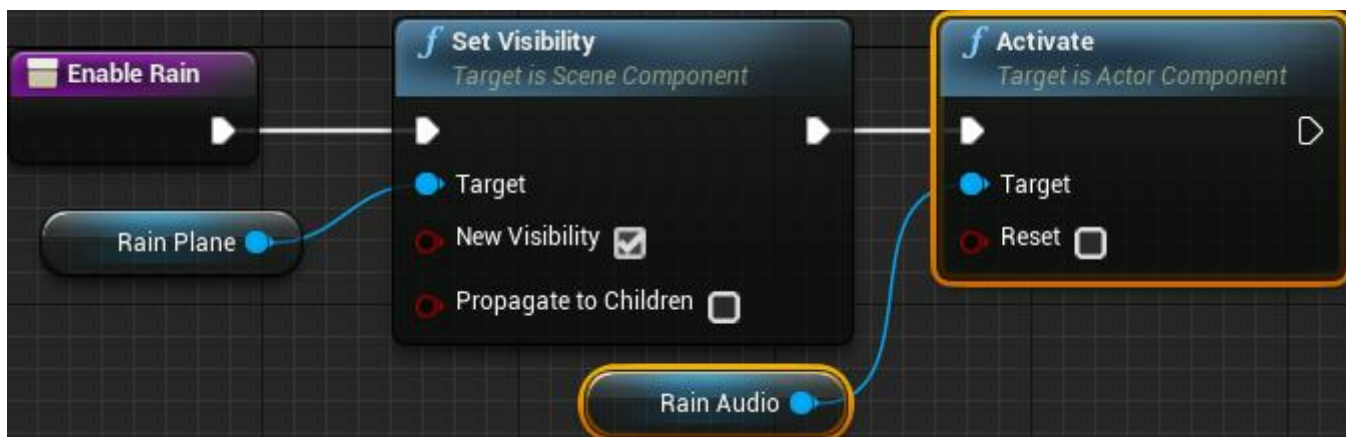
Перейдите в панель *Details* и найдите раздел *Sound*. Измените значение *Sound* на *S_Rain*.



Звук дождя не должен воспроизводиться для обычных облаков. Это значит, что для них *RainAudio* нужно деактивировать. Для этого прокрутите вниз, к разделу *Activation* и отключите *Auto Activate*.



Теперь нам нужно активировать *RainAudio* для дождевых облаков. Удобно это будет делать в функции *EnableRain*. Эта функция выполняется, если облако должно быть дождевым. Откройте функцию *EnableRain* и добавьте выделенные ноды:



Теперь нужно включить затухание и задать настройки затухания.

Настройка затухания

Перейдите в панель *Components* и выберите *RainAudio*. Перейдите в панель *Details* и зайдите в раздел *Attenuation*. Включите настройку *Override Attenuation*.

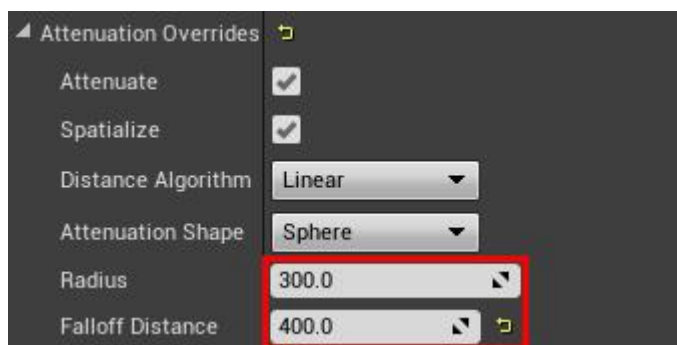
Настройки затухания определяют снижение громкости звука в зависимости от расстояния. В области *Attenuation Overrides* есть две настройки, которыми мы можем воспользоваться:

- *Radius*: максимальное расстояние, на котором может быть игрок, прежде чем громкость начнёт уменьшаться
- *Falloff Distance*: расстояние, на котором может быть игрок, прежде чем звук станет неслышным. Это расстояние прибавляется к радиусу.

Посмотрите на пример:

Когда игрок находится в пределах внутреннего круга (задаваемого настройкой *Radius*), громкость равна 100%. Когда игрок движется из внутреннего круга к краю внешнего круга (задаваемого настройкой *Falloff Distance*), громкость снижается до 0%.

В этом tutorialе мы присвоим настройке *Radius* значение 300, а настройке *Falloff Distance* — значение 400.



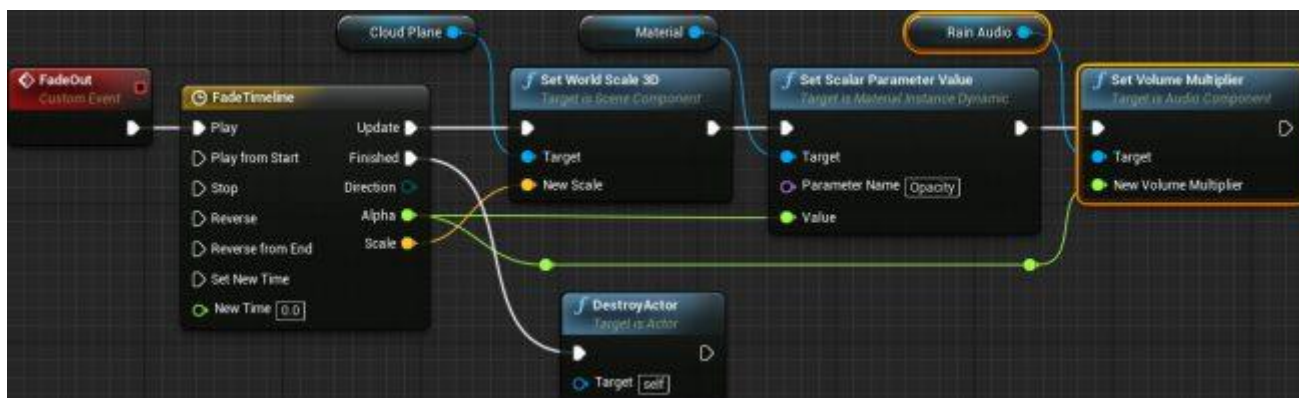
Это значит, что громкость звука будет равна 100%, когда игрок находится в пределах 300 единиц от звука. Когда расстояние будет приближаться к 700 (300 + 400) единицам, громкость начнёт падать до 0%.

Если игрок не выйдет из радиуса затухания при исчезновении облака, то звук резко оборвётся. Чтобы этого не происходило, нам нужно плавно снижать громкость звука.

Плавное снижение громкости звука

Переключитесь в Event Graph и найдите событие *FadeOut*. Это можно сделать, перейдя к панели My Blueprint и зайдя в раздел *Graphs*. Дважды щёлкните на *FadeOut* в списке *EventGraph*.

Добавьте в конец цепочки нодов выделенные ноды:



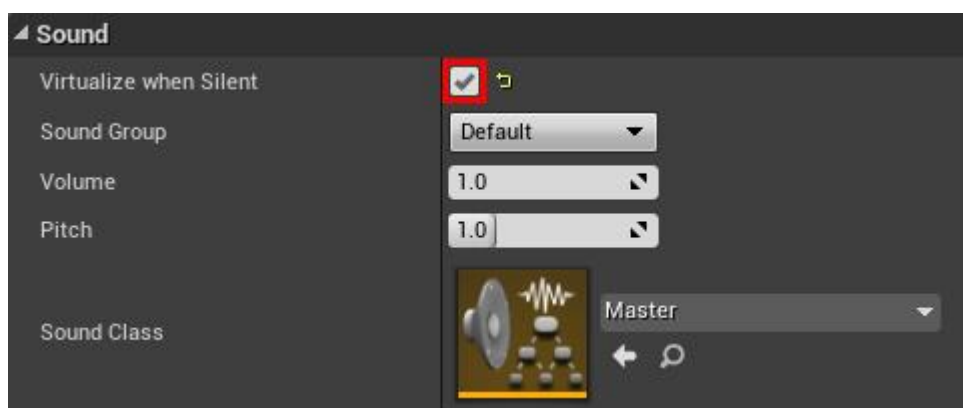
FadeOut выполняется, когда игрок касается облака. Нод *Timeline* (*FadeTimeline*) выводит значение (*Alpha*), которое за указанное время снижается с 1 до 0. Благодаря этому значению громкость *RainAudio* будет постепенно снижаться.

Примечание: можно дважды щёлкнуть на нод *Timeline*, чтобы открыть его и посмотреть, как он работает. Если вы хотите больше узнать о *Timelines*, то изучите [официальную документацию](#).

Прежде чем услышать это в действии, необходимо изменить одну настройку в *S_Rain*. Когда громкость звука равна 0%, его воспроизведение останавливается. Поскольку когда звук начинается, когда мы находимся за пределами слышимости, то громкость *S_Rain* будет равна 0%. Когда мы переместимся в пределы слышимости, то ничего не услышим.

Это можно исправить с помощью настройки *Virtualize when Silent*. Эта настройка будет всегда воспроизводить звук, вне зависимости от его громкости.

Нажмите на *Compile* и закройте *BP_Cloud*. Перейдите в папку *Audio* и откройте *S_Rain*. Перейдите в раздел *Sound* и включите *Virtualize when Silent*.

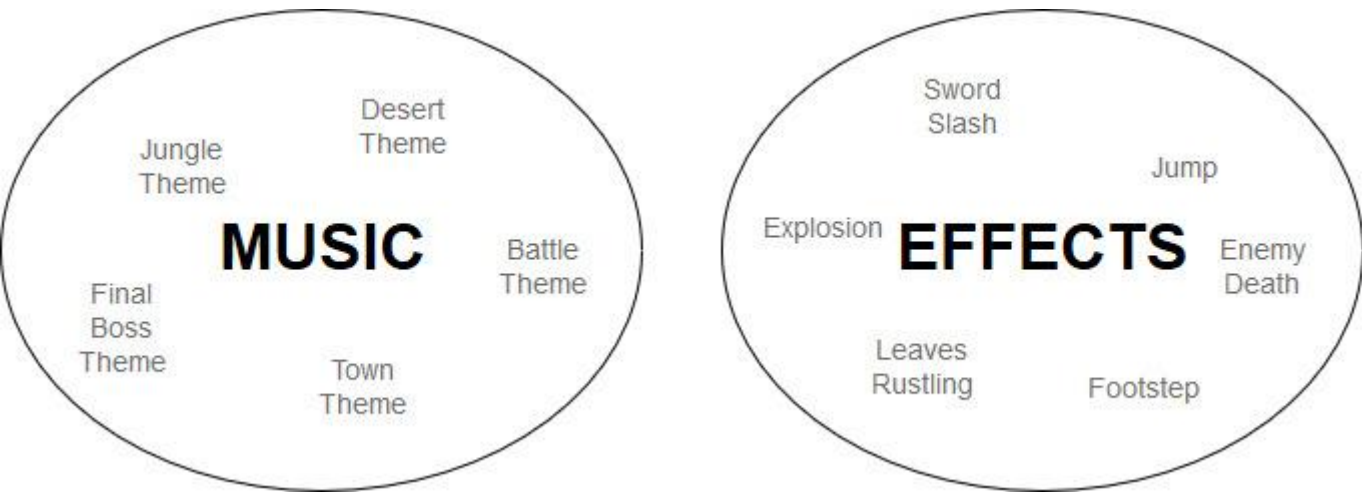


Теперь *S_Rain* будет воспроизводиться даже тогда, когда он не слышен. Закройте *S_Rain* и вернитесь в основной редактор. Нажмите на *Play* и переместитесь в пределы слышимости дождевых облаков, чтобы услышать звук дождя.

В последнем разделе мы будем контролировать громкость звуков с помощью *Sound Class* и *Sound Mix*.

Sound Class и Sound Mix

Sound Class — это простой способ группировки нескольких звуков. Например, можно сгруппировать всю музыку в один класс, а звуковые эффекты — в другой.



Чтобы изменять свойства Sound Class (громкость, тон и т.д.) во время игрового процесса, нам нужно использовать Sound Mix. Sound — это таблица, каждая запись которой является Sound Class. Каждая запись содержит настройки, которые должны быть у Sound Class.

Вот пример того, что может содержаться в Sound Mix:

Sound Class	Volume	Pitch
Music	0.5	1.0
Effects	1.0	2.0

При использовании показанного выше Sound Mix все звуки в классе *Music* будут проигрываться с половинной громкостью. У всех звуков в классе *Effects* будет удвоен тон.

Во-первых, мы создадим классы звука (Sound Class).

Создание классов звука

В этом tutorialе мы будем отдельно регулировать громкость музыки и эффектов. Это значит, что нам понадобятся два Sound Class. В Content Browser нажмите на *Add New* и выберите *Sounds\Sound Class*. Переименуйте Sound Class в *S_Music_Class*.

Создайте ещё один Sound Class и назовите его *S_Effects_Class*.



Теперь нужно привязать каждый звук к Sound Class. Для начала мы сделаем это с музыкой. Откройте *S_Music* и найдите раздел *Sound*. Измените *Sound Class* на *S_Music_Class*.

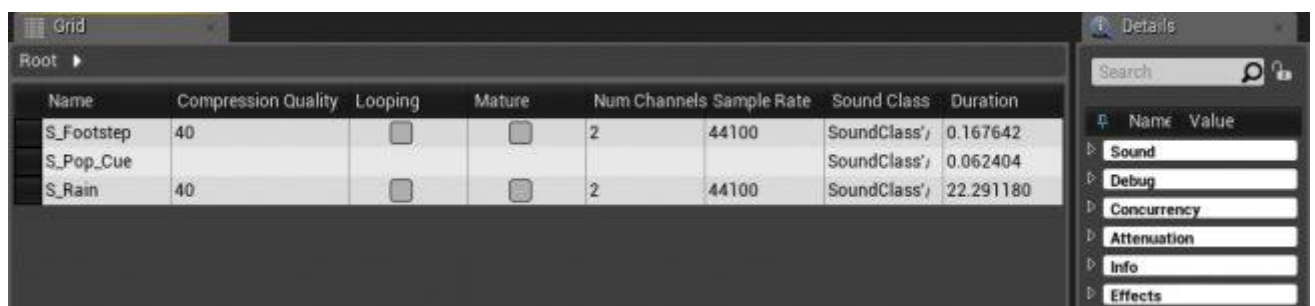


Сделав это, закройте *S_Music*.

Следующими будут звуковые эффекты. Вместо того, чтобы открывать каждый звук по отдельности и привязывать его к Sound Class, мы можем сделать это одновременно. Сначала выберите следующие ассеты:

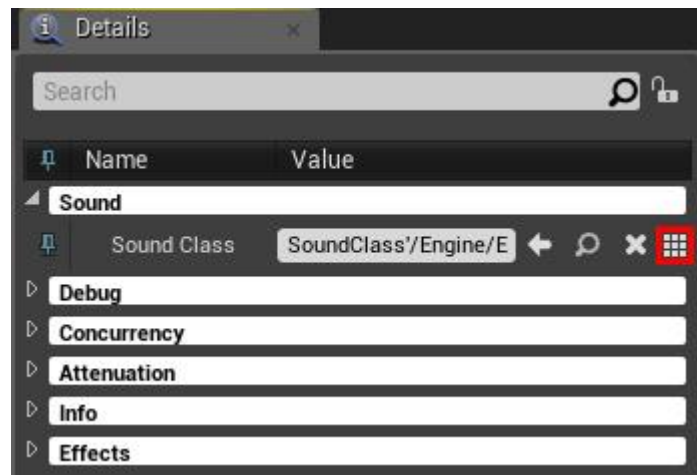
- *S_Footstep*
- *S_Pop_Cue*
- *S_Rain*

Затем *нажмите правой клавишей мыши* на одном из *выбранных ассетов*. Выберите *Asset Actions\Bulk Edit via Property Matrix*. При этом ассеты откроются в редакторе матрицы свойств.



Редактор матрицы свойств позволяет изменять общие свойства за один раз.

Перейдите в панель *Details* и разверните ветку *Sound*. Для выбора Sound Class нажмите на значок *сетки* справа от *Sound Class*.



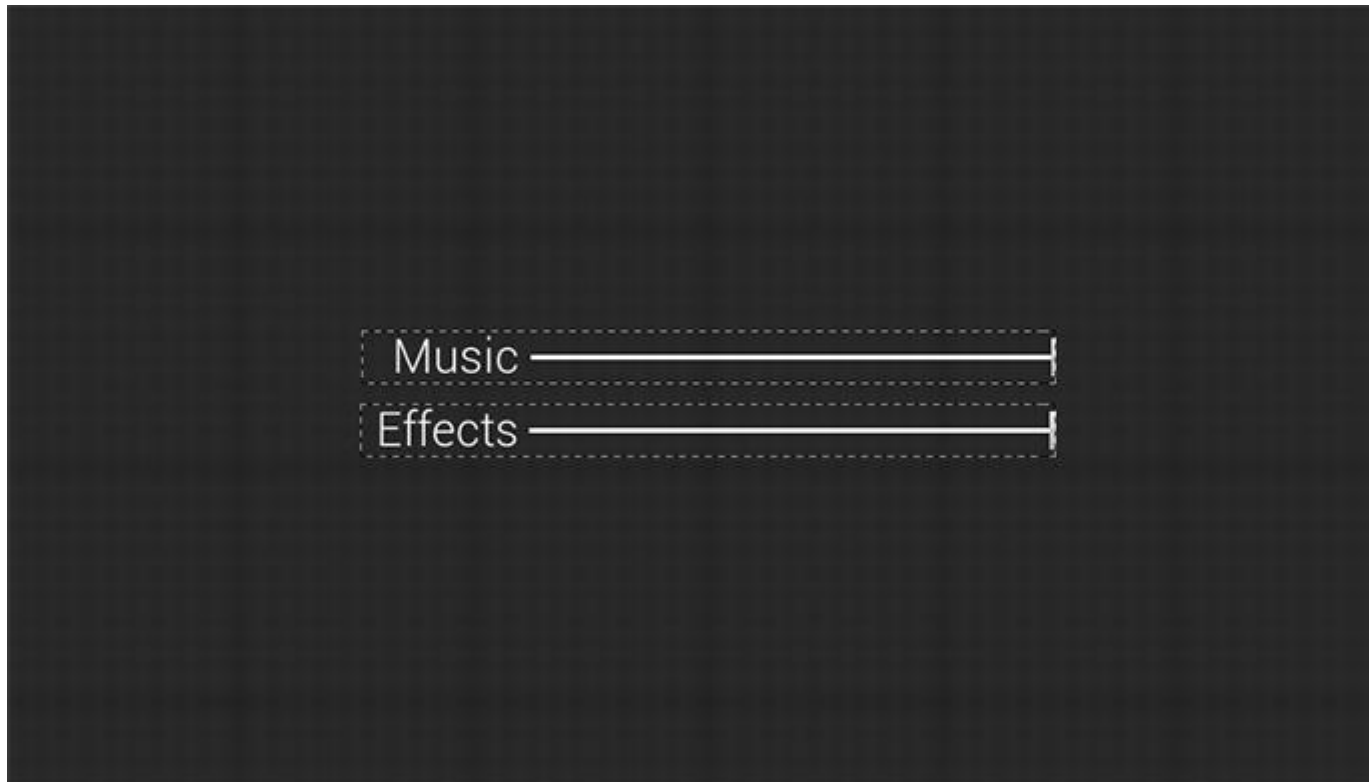
Выберите *S_Effects_Class*, а затем закройте редактор матрицы свойств.

Все звуки теперь находятся в соответствующем Sound Class. Далее мы создадим Sound Mix и настроим его с помощью Blueprints.

Создание и настройка Sound Mix

В Content Browser нажмите на *Add New* и выберите *Sounds\Sound Mix*. Переименуйте Sound Mix в *S_Volume_Mix*.

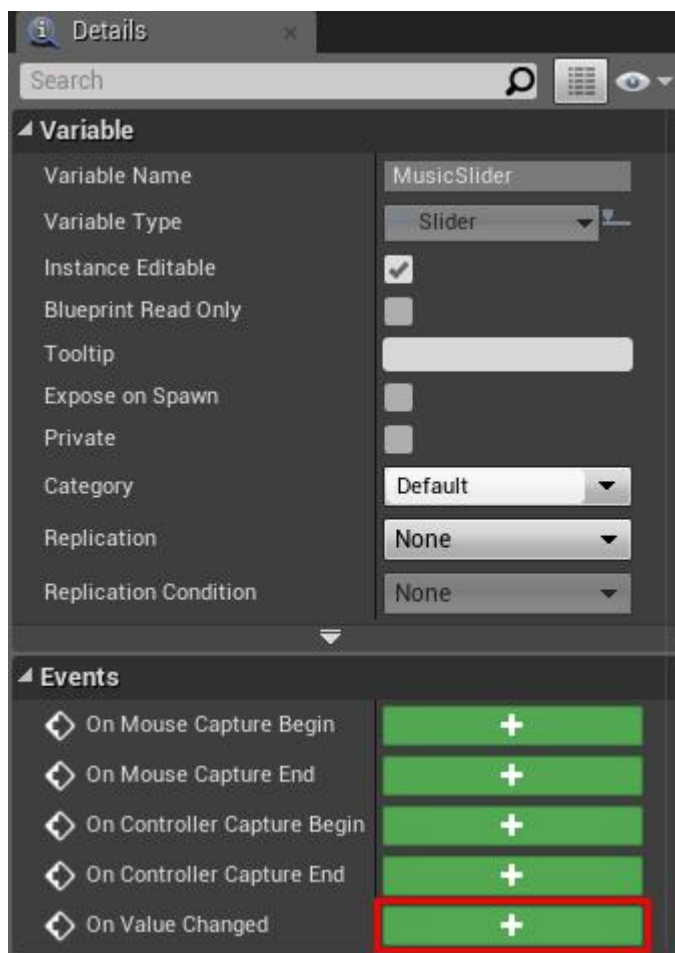
Чтобы управлять громкостью каждого Sound Class, мы будем использовать ползунки. Я уже создал виджет с двумя ползунками, чтобы вы могли его использовать. Перейдите в папку *UI* и откройте *WBP_Options*.



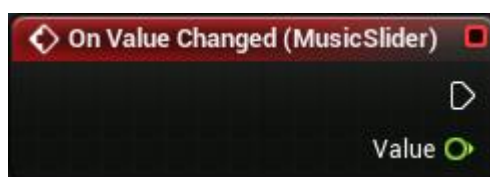
Чтобы настроить громкость, нам нужно использовать значение этих ползунков и

передавать их в Sound Mix. Сначала мы сделаем это для музыки.

Переключитесь в режим Graph и перейдите в панель My Blueprints. В разделе *Variables* выберите *MusicSlider*. Перейдите в панель Details и нажмите на кнопку рядом с *On Value Changed*.



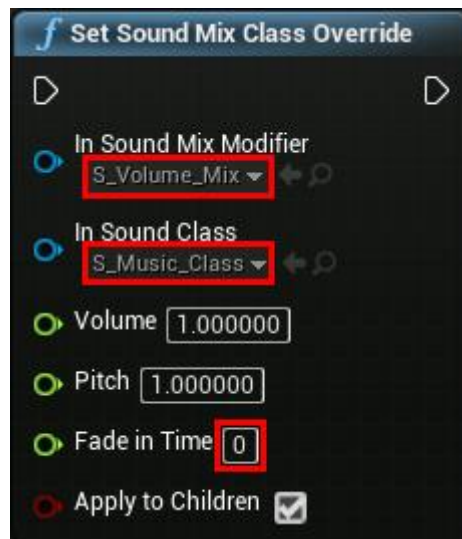
При этом будет создано событие *On Value Changed (MusicSlider)*. Это событие срабатывает, когда мы будем перемещать ползунок.



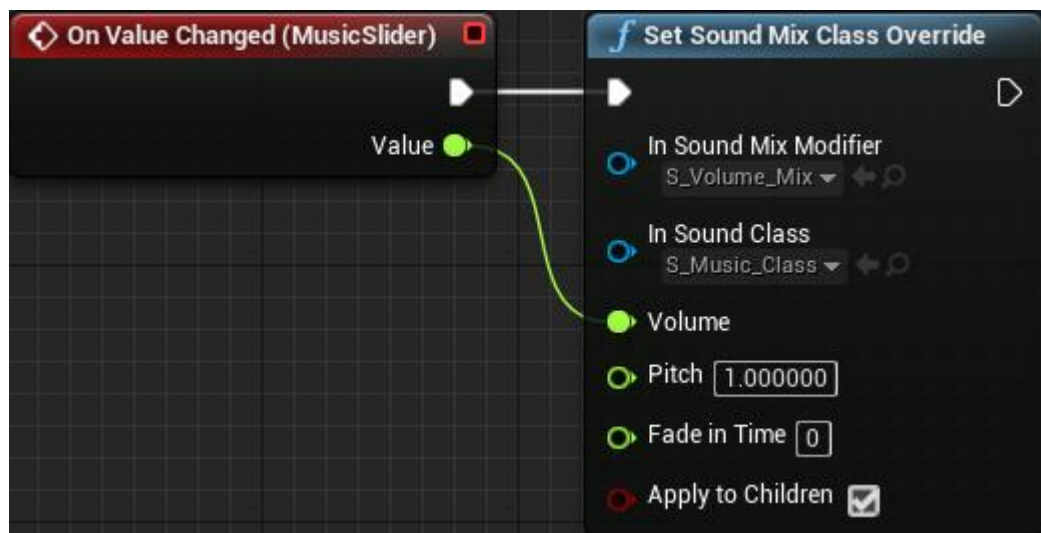
Теперь нам нужно задать громкость *S_Music_Class* в *S_Volume_Mix*. Для этого нам нужно использовать нод *Set Sound Mix Class Override*. Этот нод позволяет указать Sound Mix и Sound Class. Если Sound Class не находится в Sound Mix, то он будет добавлен. Если он уже есть в Sound Mix, то он будет обновлён.

Добавьте нод *Set Sound Mix Class Override* и выберите следующие опции:

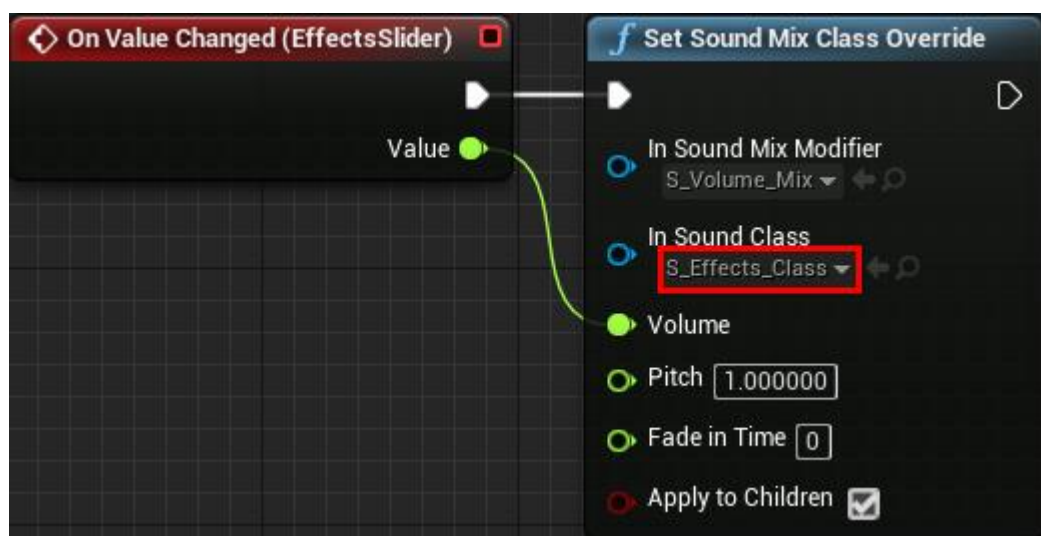
- *In Sound Mix Modifier:* *S_Volume_Mix*
- *In Sound Class:* *S_Music_Class*
- *Fade in Time:* 0 (благодаря этому регулировка громкости будет выполняться мгновенно)



Затем соедините ноды следующим образом:



Повторите эти шаги для *EffectsSlider*. Замените контакт *In Sound Class* на *S_Effects_Class*.



Теперь при изменении значения ползунка *S_Volume_Mix* будет регулировать громкость

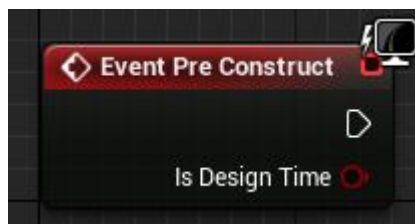
соответствующего Sound Class.

Чтобы это заработало, нам нужно активировать Sound Mix.

Активация Sound Mix

В подобных случаях (когда громкость регулируется через UI) лучше активировать Sound Mix при запуске игры. Таким образом, Sound Class будет автоматически использовать регулировку громкости из Sound Mix. Однако ради простоты мы будем активировать Sound Mix внутри виджета.

Создайте нод *Event Pre Construct*. Он подобен ноду *Event BeginPlay*, используемому в Blueprints.



Чтобы активировать Sound Mix, нам нужно использовать нод *Push Sound Mix Modifier*. Создайте его и соедините с *Event Pre Construct*. Затем задайте *In Sound Mix Modifier* значение *S_Volume_Mix*.



Благодаря этому *S_Volume_Mix* будет активироваться при спауне *WBP_Options*.

Нажмите на *Compile* и закройте *WBP_Options*. Нажмите на *Play* и нажмите клавишу *M*, чтобы открыть ползунки. Двигайте ползунки, чтобы изменять громкость каждого Sound Class