Ogni sezione di questi appunti sarà divisa da una serie di Slash "/"

EXTRA (STOQUA) per sapere dove si è rimasti

0. intro (INTRO1)

1.Spiegazione dei Blueprint. (Short BP1)

2.Uno studio approfondito di Working With Audio (WWA)

3. Wwise + Unreal Engine (WUE)

3.1 Wwise Documentation (WWISEDOC)

4. Appunti libro Richard STEVENS Game Audio Implementation (GAIRS)

5.Ricerche e test dei vari nodi Blueprint (RSBP)

6.Riassunto e ricerche di vari funzioni di C++(fonte CodeAcademy) (CADMY)

//Studio di Verse//

7. C++ Per Unreal Engine (CPPUE)

7.1 C++ varie (CPPVAR)

8. Ricerche e Utilizzo dei Metasound (RUMS)

9.Note finali, con risultati di test, ricerche, approfondimenti di vario genere (NFLY)

LEGENDA

. INIZIO NUOVO CAPITOLO

- PARAGRAFO

--SOTTO PARAGRAFO

1,2,3 NUMERAZIONI

//

BP1

.Implementazione di Interfaccia

Un'interfaccia blueprint è una raccolta di una o più funzioni - solo nome, nessuna implementazione - che possono essere aggiunte a altri progetti.

Qualsiasi progetto a cui è stata aggiunta l'interfaccia è garantito per avere tali funzioni.

Le funzioni dell'interfaccia può essere dotato di funzionalità in ciascuno dei progetti che l'hanno aggiunta.

Questo è essenzialmente come il concetto di un'interfaccia nella programmazione generale,

che consente a più tipi diversi di oggetti di condividere e accedere attraverso un'interfaccia comune.

In parole povere, le interfacce Blueprint consentono a diversi Blueprint di condividere e inviare dati l'uno all'altro.

EXEC

Viene utilizzata per eseguire un comando in un nuovo processo.

```
#include <cstdlib>

#include <iostream>

int main() {

   std::string command = "ls -l";

   std::system(command.c_str());//Parametro

   return 0;

}
```

//

.Classi BP

-Blueprint solo dati

Un blueprint solo dati è una classe blueprint che contiene solo il codice (sotto forma di grafici di nodi), le variabili e i componenti ereditato dal suo genitore.

-Interfaccia Blueprint

Un'interfaccia blueprint è una raccolta di una o più funzioni - solo nome, nessuna implementazione - che possono essere aggiunte a altri progetti.

In parole povere, le interfacce Blueprint consentono a diversi Blueprint di condividere e inviare dati l'uno all'altro.

-Macro

A differenza delle funzioni,

Le macro possono avere più di un pin di esecuzione di output,

quindi è possibile avere un flusso di esecuzione come questo in cui sono attivati diversi pin di esecuzione di output a seconda dei risultati della logica del grafo.

//le macro consentono la sostituzione testuale nel codice prima che venga sottoposto alla compilazione effettiva//

-Le Utilities servon per eseguire azioni

Le trovi cliccando nel grafico con il tasto destro

- Construction script, parte immediatamente dalla creazione dell'oggetto stesso.

- Event Graph, parte solo quando parte il gioco.

- I graph contengono il comportamento in fase di progettazione e di gioco dei tuoi progetti.

-Per avere un riferimento all'oggetto, crea una variabile di tipo con ISTANCE EDITABLE

Riferimento alle variabili, OBJECT TYPE OBJECT REFERENCE, quando scriverai actor nella ricerca di varibili

-Funzioni

Le funzioni possono essere pure o impure. La differenza principale è che le funzioni pure promettono di non modificare lo stato o il membri della classe in qualsiasi modo,

mentre le funzioni impure sono libere di modificare lo stato.

Le funzioni pure sono generalmente utilizzate per getter Funzioni o operatori che restituiscono solo un valore di dati.

Le funzioni sono grafi di nodi appartenenti a un particolare Blueprint che possono essere eseguiti, o chiamati, da un altro all'interno del Blueprint.

A differenza delle macro di Blueprint, è possibile aggiungere input e output di dati solo a Funzioni.

-Function Calls

Le chiamate di funzione sono azioni che possono essere formate all'interno di blueprint che

corrispondono a funzioni appartenenti a un attore o a un oggetto di destinazione.

Nel caso dei Progetti di Livello, l'Attore associato in molti casi è il Livello Blueprint stesso.

Self

Le chiamate di funzione Self sono funzioni che appartengono al Blueprint stesso, tramite essere dichiarate in la classe da cui deriva il progetto o una classe padre.

Other_____

Le altre chiamate di funzione sono funzioni che appartengono ad altri oggetti o attori oltre al blueprint.

Puro

Le chiamate di funzione pura sono azioni speciali che possono essere eseguite e che non influiscono direttamente sul mondo o gli Oggetti in esso contenuti.

Si tratta in genere di funzioni che restituiscono il valore di una proprietà o di funzioni matematiche Operazioni come sommare, sottrarre, moltiplicare, dividere, ecc.

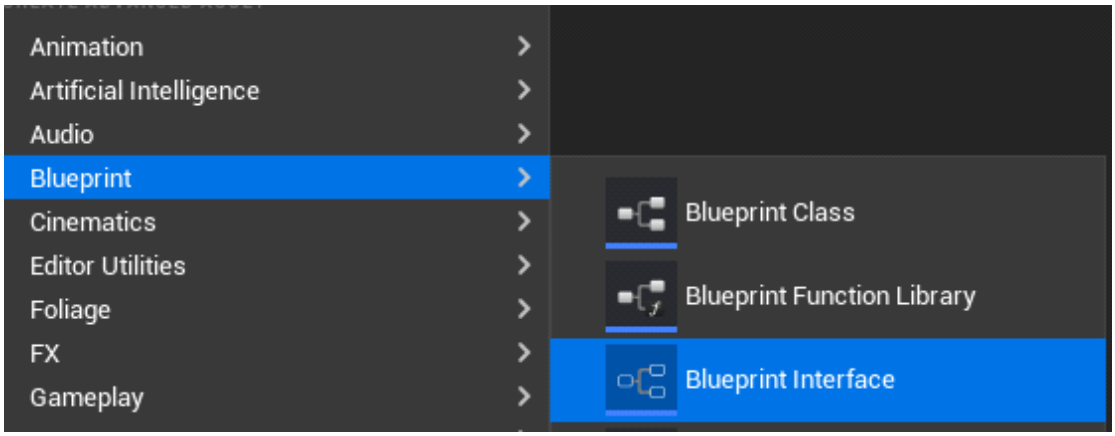
due valori, i cui risultati non hanno alcun impatto su nulla. È l'assegnazione o l'uso del risultato che genera un effetto sul mondo.

Esistono di due tipi STANDARDA e COMPATTE

.Sul grafico ADD ROUTERN NODE

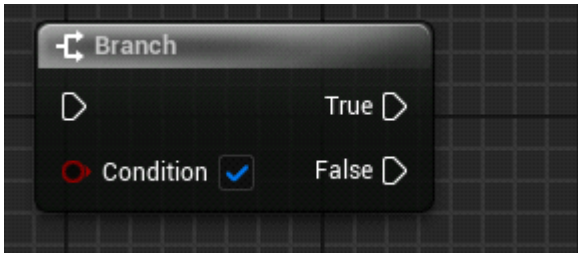
BLUEPRINT SOLO DATI

BLUEPRINT INTERFACE

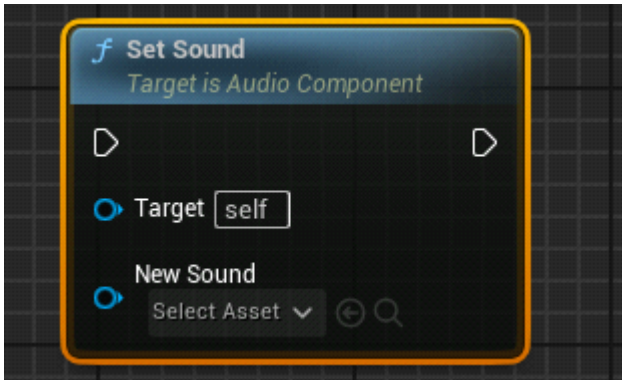


BLUEPRINT MACRO

BLUEPRINT UTILITIES



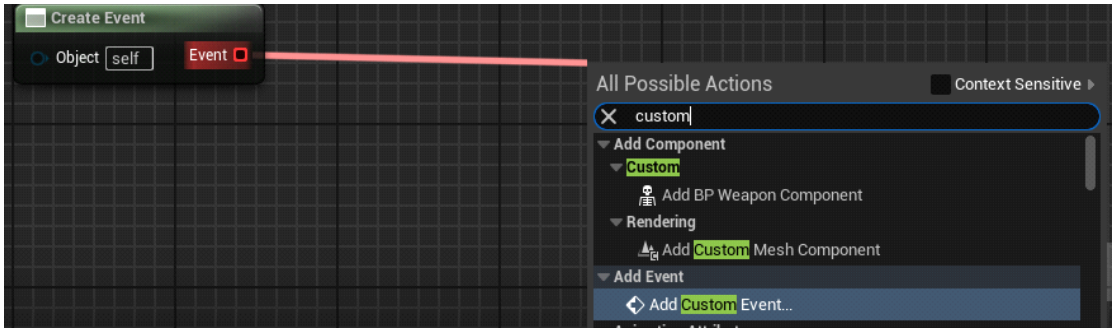
BLUEPRINT FUNCTION



//////////////

Crea eventi e Funzioni

Add event



//

.UTILITIES / FLOW CONTROL

- Branch node, simile a If, viene utilizzato con una condizione di tipo Booleano

- DoN

Il nodo DoN emetterà un impulso di esecuzione N volte. Dopo che il limite è stato raggiunto,

cesserà tutta l'esecuzione in uscita fino a quando non verrà inviato un impulso nel suo ingresso di reset.

Ha bisogno di un Enter e di un Reset, per essere riattivato

Ad esempio, è possibile che un veicolo si avvii venti volte e poi non si avvii fino a quando non viene attivato un evento di rifornimento legato all'ingresso Reset.

- DoOnce

Il nodo DoOnce, come suggerisce il nome, emetterà un impulso di esecuzione una sola volta.

Da quel momento in poi, cesserà tutte le esecuzioni in uscita fino a quando non verrà inviato un impulso nel suo ingresso Reset.

Questo nodo è equivalente a un nodo DoN dove N = 1.

- Flipflop

Il nodo FlipFlop accetta un output di esecuzione e passa da un output di esecuzione all'altro. La prima volta che viene chiamato, viene eseguito l'output A.

La seconda volta, B. Poi A, poi B, e così via. Il nodo ha anche un output booleano che consente di tenere traccia di quando è stato chiamato l'output A.

- Gate

viene utilizzato per aprire e chiudere un flusso di esecuzione.

In poche parole, l'ingresso Enter riceve impulsi di esecuzione e lo stato corrente della porta (aperta o chiusa) determina se tali impulsi escono o meno dall'uscita Exit.

- Sequence

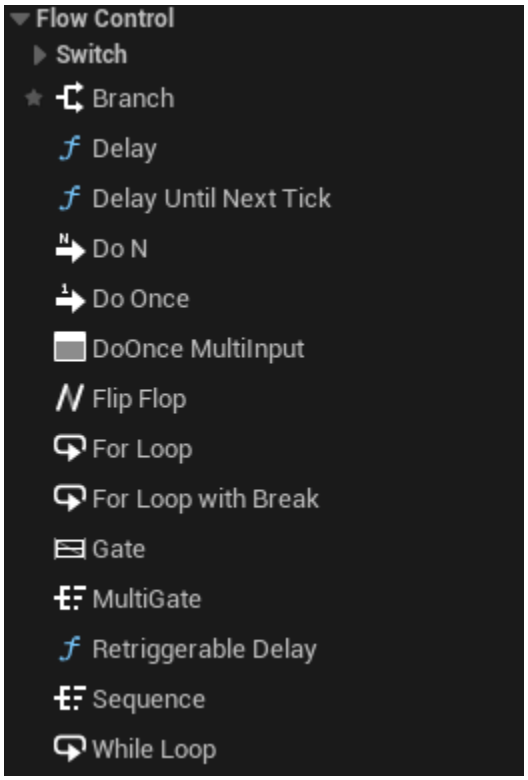
Il nodo Sequenza consente a un singolo impulso di esecuzione di attivare una serie di eventi in ordine.

Il nodo può avere un numero qualsiasi di output, che vengono tutti chiamati non appena il nodo Sequence riceve un input.

Saranno sempre chiamati in ordine, ma senza alcun ritardo.

Per un utente tipico, è probabile che le uscite siano state attivate contemporaneamente.

-While Loop

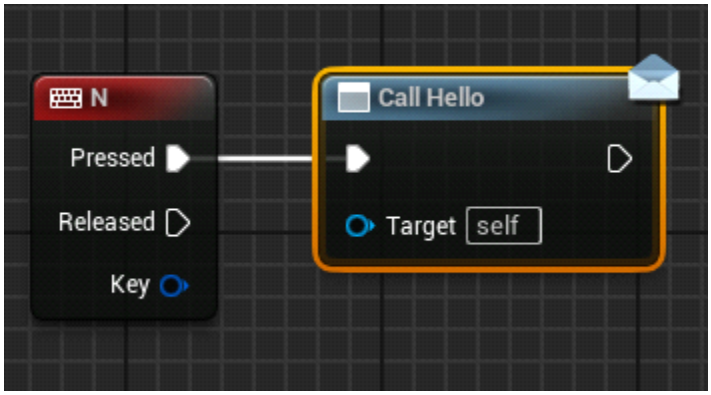


//

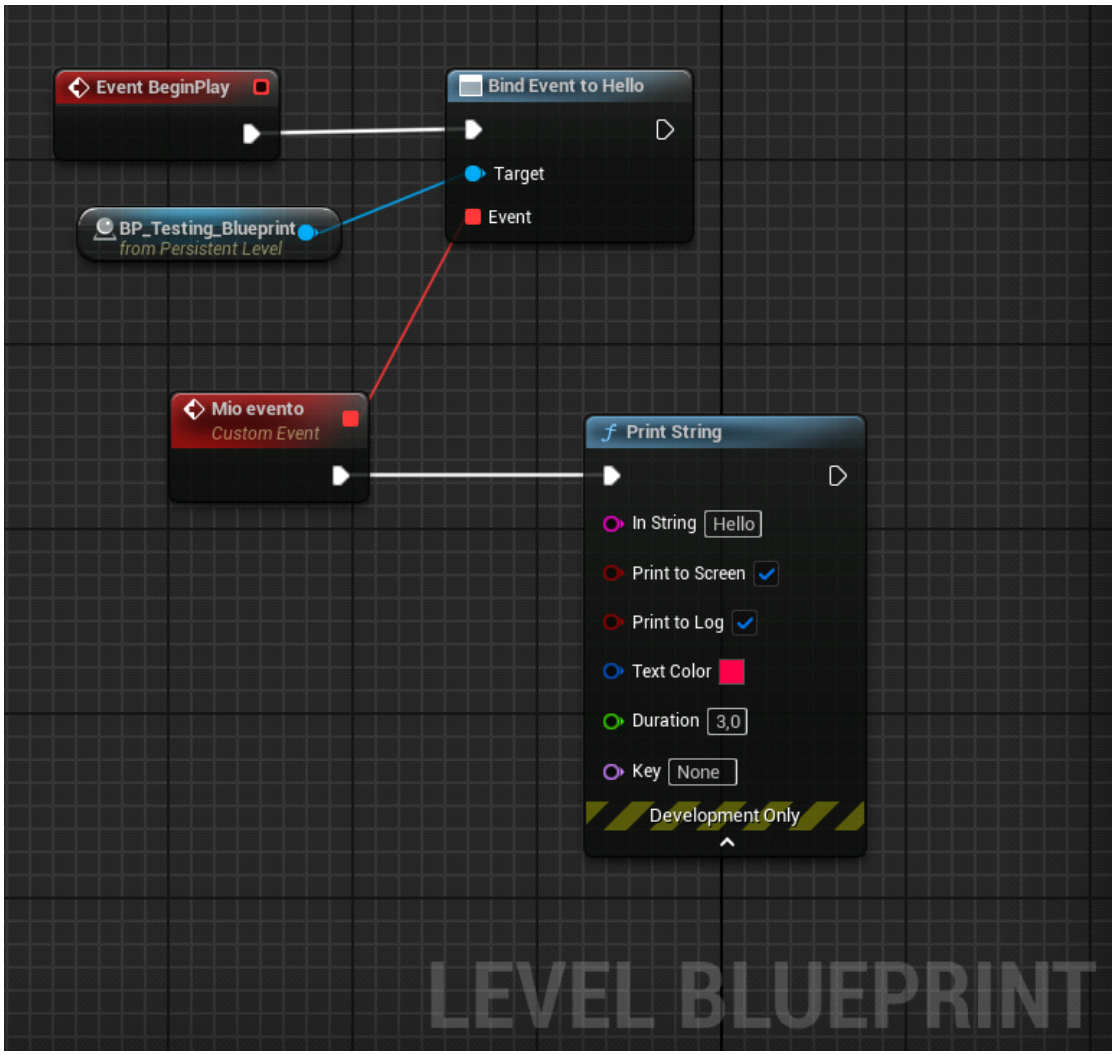
.Dispatcher di eventi

//Un delegato è un tipo che rappresenta riferimenti a metodi con un particolare elenco di parametri e tipo restituito.//

Consente a una classe Blueprint di segnalare il proprio stato al progetto di livello.

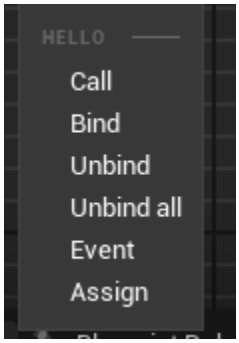


Questi eventi possono essere associati all'interno di una classe Blueprint, ma i Dispatcher di eventi consentono anche di attivare eventi all'interno del Blueprint livello.



Il processo per aggiungere input a Event Dispatcher è simile al flusso di lavoro per l'aggiunta di input e output a funzioni, eventi personalizzati e macro.

I metodi bind, unbinding e assign consentono di aggiungere eventi al dispatcher di eventi elenco eventi, mentre il metodo call attiva tutti gli eventi memorizzati nell'elenco eventi.



- Binding and Unbinding Events

Aggiunta e rimozione di eventi dall'elenco degli eventi di un dispatcher di eventi.

- Chiamata di dispatcher di eventi//Crea un evento personalizzato, vai su BP LEVEL, CALL quell evento al RIFERIMENTO dell'attore associato a quell'evento.

La chiamata a Event Dispatcher esegue tutti gli eventi attualmente associati nell'elenco degli eventi.

- IL DISPATCHER E' UTILE QUANDO

Hai due attori nel tuo livello che vuoi comunicare tra loro.

Interagendo con un interruttore nel tuo livello si apre una porta specifica o si accende una luce specifica (ognuno dei quali sono progetti separati).

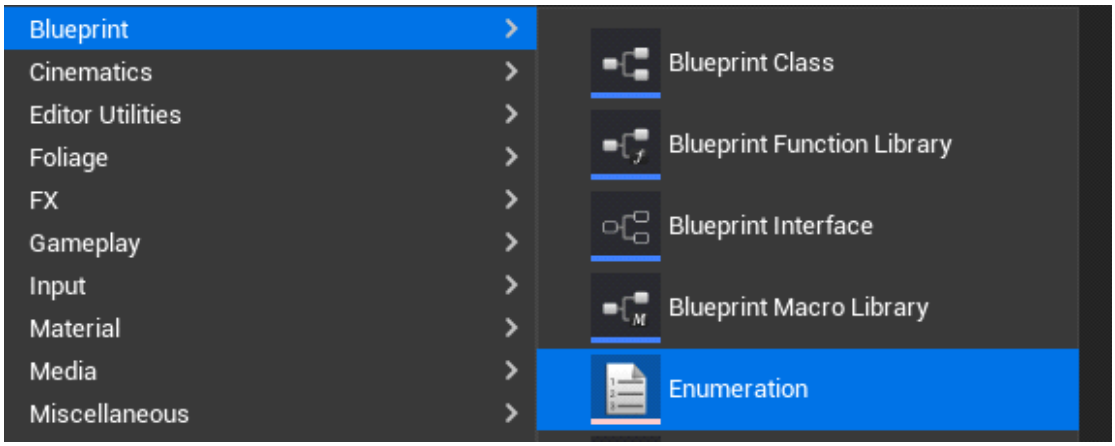
////////////

- Switch(E' LO SWITCH STATEMENT DI C#/C++)

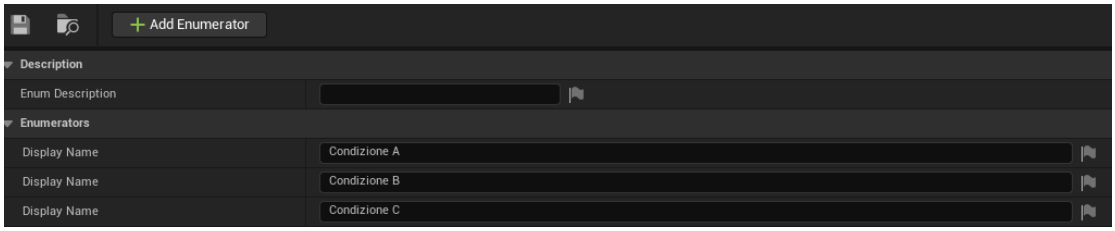
Un nodo switch legge un input di dati e, in base al valore di tale input,

invia il flusso di esecuzione dall'output di esecuzione corrispondente (o predefinito facoltativo).

a) Crea un Enumeratore



b) Aggiungi diversi stati o condizioni all'enumeratore



c) Swith to ILTUOENUMERATORE

//

- I RandomStreams

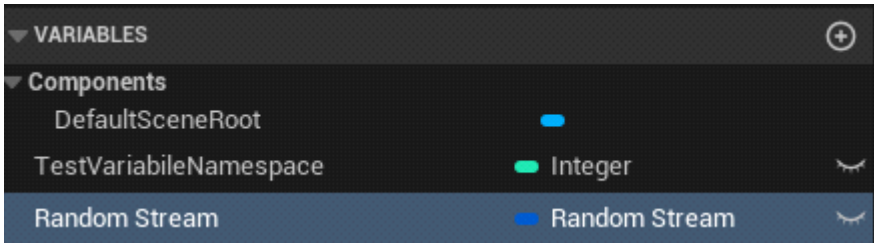
Consentono di generare numeri ripetibili e casuali e di utilizzarli all'interno di Blueprint e Level Blueprint, nonché di AnimBlueprint per le animazioni.

Il nodo Imposta seme flusso casuale può essere utile quando si dispone di un progetto che posiziona molti oggetti, come fili d'erba o pietre, ma si desidera che siano sempre posizionati allo stesso modo per i test o per altri scopi.

Sovrascriverà il seme iniziale impostato casualmente quando il progetto viene posizionato in un livello.

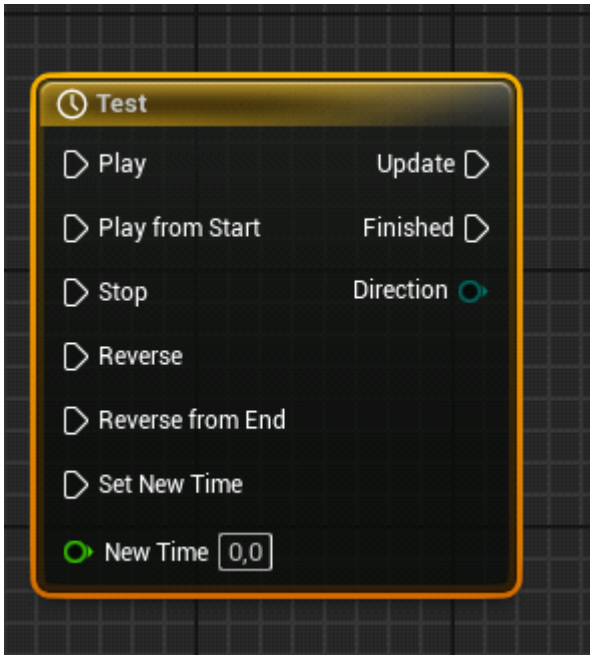
Per utilizzare una variabile RandomStream, esiste un insieme di funzioni che prendono un RandomStream come input e restituiscono un valore casuale di diverso tipo, a seconda della funzione.

Esempio chiamata di funzione *Random Bool From stream*

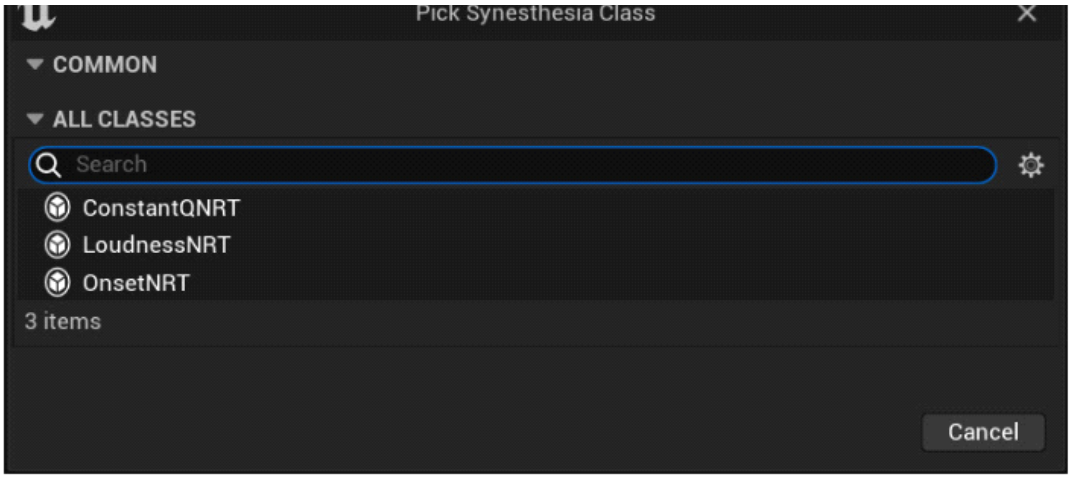


- Timelines

I nodi della timeline sono nodi speciali all'interno dei blueprint che forniscono animazioni basate sul tempo da progettare e riprodurre rapidamente in base a eventi, float, vettori o colori che possono essere attivati in corrispondenza dei fotogrammi chiave lungo la timeline.



////////////



.VolumeNRT

L'analizzatore LoudnessNRT misura il volume percettivo dell'audio.

È simile a un inseguitore di inviluppo o anche a un misuratore di decibel in quanto è correlato all'ampiezza dell'audio e fornisce un numero che cambia nel tempo.

LoudnessNRTSettings viene usato per configurare l'analizzatore LoudnessNRT. I parametri più comunemente sintonizzati in LoudnessNRTSettings sono:

.LoudnessNRT

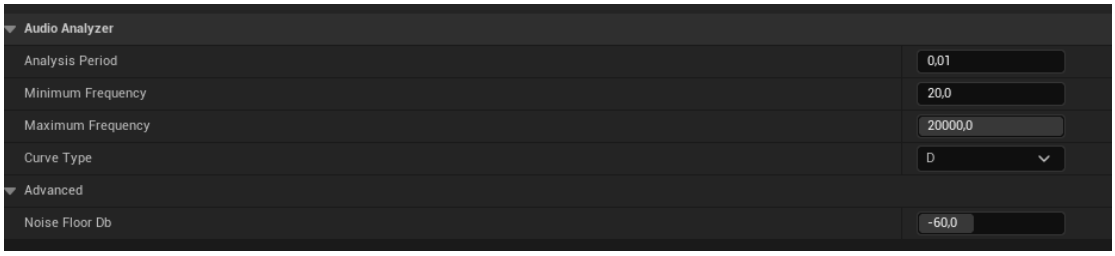
Misura la loudness

AnalysisPeriod controlla la frequenza con cui viene eseguita la misurazione.

MinimumFrequency e MaximumFrequency controllano le frequenze di interesse. Ciò è utile se si desidera ignorare completamente alcuni contenuti nei registri superiore o inferiore o se si desidera contrastare l'attività del volume di più gamme di frequenza.

CurveType controlla in modo specifico quale curva percettiva deve essere utilizzata. Si tratta in genere di un'impostazione avanzata.

NoiseFloorDb indica l'ampiezza che deve essere considerata silenzio.



.ConstantQNRT

L'analizzatore ConstantQNRT misura l'intensità delle singole bande nell'audio.

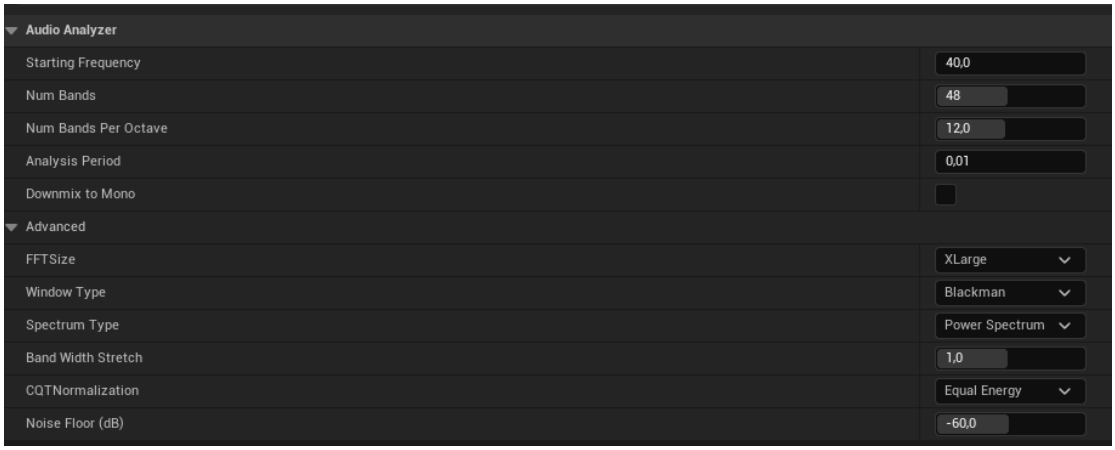
È simile a uno spettrogramma in quanto rappresenta il volume di una banda di frequenza in un dato momento, ma differisce per la disposizione delle bande in modo percettivamente significativo.

ConstantQNRTSettings viene utilizzato per configurare l'analizzatore ConstantQNRT.

AnalysisPeriod controlla la frequenza con cui viene eseguita la misurazione.

StartingFrequency, NumBands e NumBandsPerOctave controllano la spaziatura, la larghezza, la posizione e il conteggio delle bande di frequenza.

DownMixToMono indica se i canali audio devono essere elaborati separatamente o mixati in un singolo canale audio prima dell'elaborazione.



.OnsetNRT

L'analizzatore OnsetNRT rileva gli eventi di insorgenza dell'audio provenienti da una moltitudine di fonti,

tra cui insorgenze di note, colpi percussivi, plosive degli altoparlanti ed esplosioni. L'analizzatore OnsetNRT fornisce agli onset sia i timestamp che la forza.

OnsetNRTSettings viene utilizzato per configurare l'analizzatore OnsetNRT.

GranulartiyInSeconds determina la spaziatura minima tra le insorgenze.

La sensibilità imposta la soglia per il rumore che deve essere un inizio prima che venga rilevato.

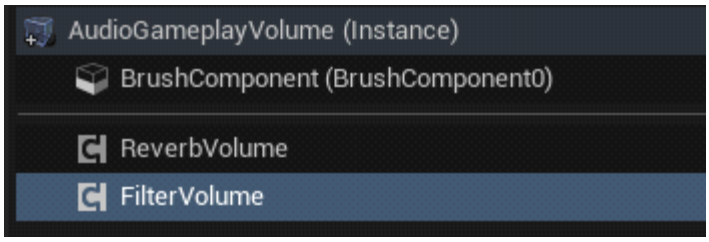
MinimumFrequency e MaximumFrequency controllano l'intervallo di frequenza per cercare le insorgenze.

Ciò è utile se si desidera ignorare completamente alcuni contenuti nei registri superiore o inferiore, o se si desidera contrastare l'attività di insorgenza di più gamme di frequenza.

DownMixToMono indica se i canali audio devono essere elaborati separatamente o mixati su un singolo canale audio prima dell'elaborazione.

Details
Search
Audio Analyzer
Downmix to Mono
Granularity in Seconds
0,01
Sensitivity
0,5
Minimum Frequency
20,0
Maximum Frequency
16000,0

////////////



Ogni componente contiene un comportamento configurabile che è possibile aggiungere a un volume singolarmente o in combinazione con altri, in modo da poter creare funzionalità personalizzate per quando un listener entra o esce dal volume.

I tipi di componenti di comportamento sono:

Attenuazione: interpola il volume corrente dell'audio (loudness) a un volume di destinazione (loudness).

Filtro: applica un filtro passa-basso all'audio.

Riverbero: aggiunge un effetto di riverbero all'audio. Quando l'ascoltatore entra nel volume, questo effetto si applica a tutti i suoni, indipendentemente dal fatto che si trovino all'interno o all'esterno del volume.

Sovrascrittura submix: aggiunge una catena di effetti submix a un submix sonoro. Quando l'ascoltatore entra nel volume, questo effetto si applica a tutti i suoni

Invio submix: invia l'audio a un submix audio. Questo effetto si applica ai suoni all'interno del volume e può essere configurato per essere applicato mentre l'ascoltatore è all'esterno o all'interno. Per funzionare, i volumi di gioco audio richiedono risorse di onde sonore come parte di una classe sonora correttamente impostata.

1. Crea la classe sonora

2. Applica alla sound cue la classe sonora

3. Apri la classe sonora, in Routing < Applica volume di ambiente, setta Default 2DReverb Send Amount

4. Metti in gioco l'audio gameplay volume

5. Aggiungi la componente di audio gameplay volume

6. Gioca

Ogni oggetto di gioco (audio) deve essere settatto se all'interno o all esterno dell audio volume.

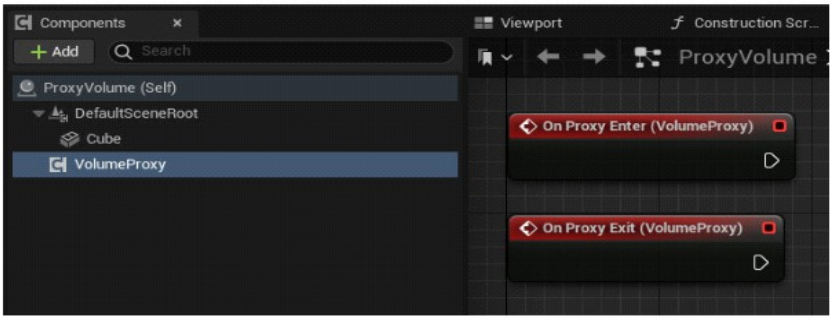
Quindi se all'interno avrà dei settaggi specifici (reverb della stanza ecc)

Se all esterno, attenuzione o filtri

Devi tenere abilitato in routing della classe apply ambient volumes.

//Un proxy primitivo utilizza una mesh statica primitiva sull'attore come base per la collisione per attivare gli eventi di entrata e uscita.//

Proxy Functionality



//

- Audio Mixer

E' il renderer audio multipiattaforma completo di Unreal Engine.

- Funzionalità legacy

Il nucleo delle funzionalità audio legacy di Unreal Engine sono i segnali sonori, le classi di suoni e i mix audio.

- Sound Cues

I segnali sonori sono gli oggetti audio legacy di Unreal Engine.

I segnali sonori sono grafici dei parametri audio da cui i progettisti audio possono riprodurre le sorgenti sonore e manipolare vari parametri, come il volume e l'intonazione.

- Sound Classes and Sound Mixes

Le classi sonore rappresentano un insieme di parametri statici corrispondenti a suoni specifici.

I mix audio offrono ai progettisti audio la possibilità di modulare dinamicamente un set di parametri audio in fase di esecuzione.

Ciò è utile per creare suoni dinamici che reagiscono agli eventi di gioco e ad altre condizioni.

- MetaSounds

MetaSounds ha la stessa funzionalità di Sound Cues e include funzionalità di nuova generazione che consentono ai progettisti di creare esperienze audio coinvolgenti.

- Audio Modulations

I bus dei parametri di modulazione audio offrono ai progettisti audio la possibilità di definire i

propri gruppi di parametri (classi sonore) e di controllarli come desiderano (mix sonori).

- Quarzo

Quartz è un set di funzionalità che porta in Blueprints la temporizzazione accurata degli eventi audio.

- Audio Analysis

L'analisi audio è un insieme di tecnologie che offre analisi audio non in tempo reale e in tempo reale.

//

- Panoramica del motore audio

Il Mixer audio fa essenzialmente la stessa cosa di una console di missaggio hardware: aggiunge insieme le sorgenti sonore dopo averle elaborate attraverso una varietà di parametri e processori di effetti, anche attraverso submixing e processori di effetti master.

Il rendering audio è il processo mediante il quale le sorgenti sonore vengono decodificate e mixate insieme, inviate a un endpoint hardware audio (chiamato convertitore digitale-analogico o DAC) e infine riprodotte su uno o più altoparlanti

The Submix Graph

Dopo la generazione della sorgente, il Mixer audio elabora il grafico del submix. L'output del grafico submix è l'audio ascoltato sull'hardware audio.

Sending To Submixes

Un submix principale viene creato nel motore audio quando il motore audio viene inizializzato.

Il submix di base di una sorgente è il submix che miscela l'uscita della sorgente a tutto volume (post-attenuazione, post-spazializzazione).

È inoltre possibile specificare un numero qualsiasi di submix di mandata, il che consente di mixare insieme le sorgenti prima di inviarle tramite effetti DSP ausiliari. Questi submix di mandata sono analogo ai canali ausiliari per effetti DSP come il riverbero in una workstation audio digitale o in una console di missaggio.

Master Submixes

Oltre al submix master, vengono creati e aggiunti due submix dell'effetto master come figli a questo submix master.

Questi sono il submix del riverbero principale e il submix dell'equalizzatore principale (EQ), sono i meccanismi sottostanti che supportano le varie funzionalità di livello superiore in classi di suoni, mix di suoni, volumi audio e attenuazioni del suono che interagiscono con il riverbero e l'equalizzazione.

Generating Output

Il submix principale è il nodo radice del grafico del submix. L'output finale viene generato recuperando l'audio generato da questo submix principale.

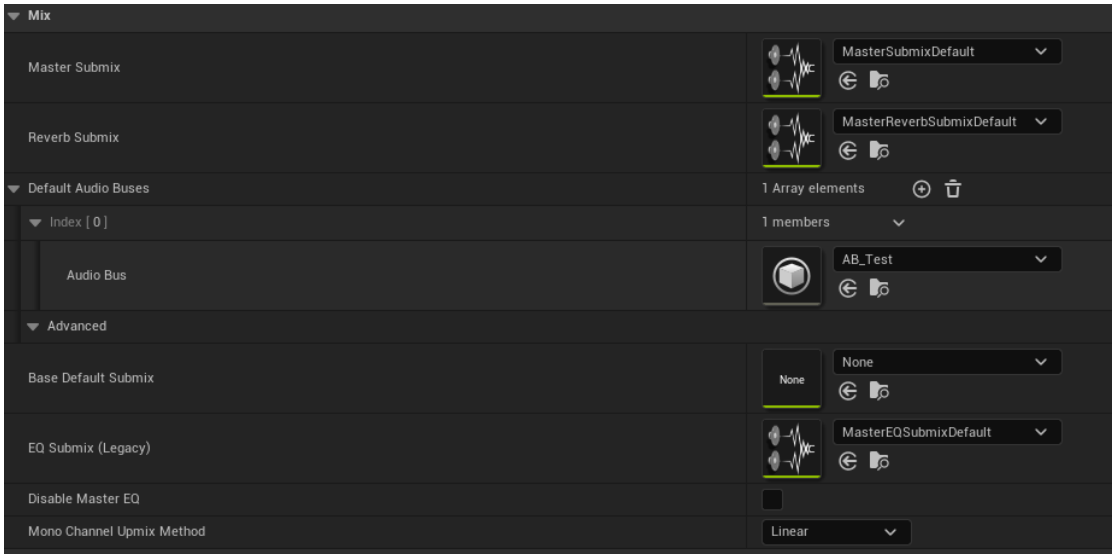
L'output di ciascun oggetto submix viene generato generando prima l'output dei submix secondari del submix, quindi miscelando questo output con qualsiasi sorgente che sia stata registrata (o inviata) ad esso.

Questo audio mixato e intercalato viene quindi alimentato attraverso la catena di effetti DSP del submix per generare un output finale.

Additional Submix Features: Analysis, Recording, and Listening

I submix sono l'oggetto ideale per eseguire analisi audio. Attualmente esiste un meccanismo per registrare i delegati Blueprint, che recupera i valori dell'inviluppo di ampiezza dal submix, per canale. Ciò potrebbe essere utile per una serie di visualizzazioni o sistemi di gioco che desiderano interagire con l'audio. C'è anche un delegato FFT in tempo reale per recuperare i dati spettrali dai submix in Blueprint.

I submix hanno anche una funzione che consente di registrare il loro output su un oggetto USoundWave o su un file PCM () grezzo su disco. .wav



//////////////

Il panning a potenza uguale mantiene la potenza (ampiezza al quadrato) tra i canali di uscita.

Il panning può essere cambiato da Project settings < Engine < Audio < Panning

.Game Volume Mixing

Regolazioni dirette del volume

Le singole risorse, come i segnali sonori e le onde sonore, dispongono di parametri che forniscono il controllo del volume.

I componenti audio consentono inoltre di modificare il volume dell'audio da Blueprint.

.Sound Classes

Oltre al volume, le classi SoundSound possono essere utilizzate per controllare altri parametri sui suoni in un gruppo di classi.

.Sound Mixes

Un mix sonoro è un tipo di risorsa che applica regolatori dinamici di volume e intonazione della classe sonora alle classi sonore.

I mix sonori sono il metodo tradizionale con i giochi per eseguire il controllo del volume basato sulla classe, incluso il ducking.

L'attenuazione del volume si ottiene utilizzando un modificatore di mix passivo, un meccanismo che applica indirettamente un mix quando un suono viene riprodotto su una determinata classe sonora.

In questo modo, un gioco può, ad esempio, causare abbassamenti sull'ambiente di sfondo quando un giocatore parla o quando una pistola spara.

Funzionalità aggiuntive di Submix: analisi, registrazione e ascolto

I submix sono l'oggetto ideale per eseguire analisi audio. Attualmente esiste un meccanismo per registrare i delegati Blueprint, che recupera i valori dell'inviluppo di ampiezza dal submix, per canale. Ciò potrebbe essere utile per una serie di visualizzazioni o sistemi di gioco che desiderano interagire con l'audio. C'è anche un delegato FFT in tempo reale per recuperare i dati spettrali dai submix in Blueprint.

I submix hanno anche una funzione che consente di registrare il loro output su un oggetto USoundWave o su un file PCM (.wav) non elaborato su disco.

.audio Parameter Modulation

I suoni possono ora associare i loro parametri come destinazioni di modulazione di un bus di

parametri. Un bus di parametri

è un oggetto che consente alle sorgenti di modulazione dei parametri arbitrari di scrivere nel bus. Ad esempio,

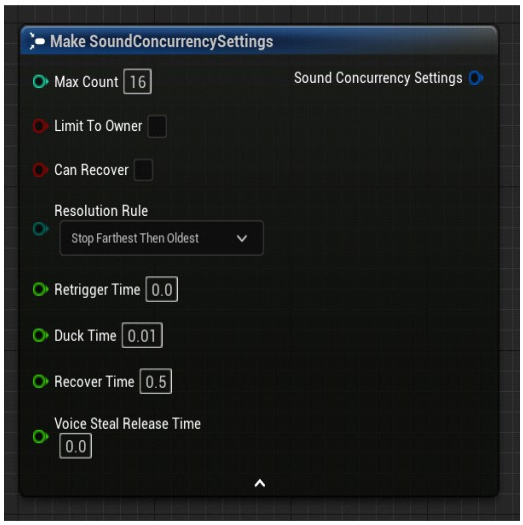
una sorgente di modulazione potrebbe provenire da un Blueprint, derivato da un mix di parametri, da un sistema interattivo o da un oscillatore di modulazione dei parametri come un LFO.

Un metodo di mixaggio più recente per controllare un mix di gioco è il plug-in audio Parameter Modulation.

.Concurrency Management

Un aspetto spesso trascurato del mixaggio dei giochi è il problema della gestione della concorrenza del suono, in sostanza,

quanti suoni di un determinato tipo possono essere riprodotti contemporaneamente. Senza un'attenta gestione, è facile che i giochi generino una marea di suoni di un tipo, come pistole e altre armi nemiche.



.Gestione e prioritizzazione della polifonia globale

Una volta che il motore audio raggiunge il numero di suoni che può emettere, deve decidere quali riprodurre e quali rifiutare o interrompere.

Questo viene fatto attraverso una combinazione del volume totale del suono (il volume finale del suono dopo che tutte le fasi sono state prese in considerazione) e la priorità del suono.

.Effetti DSP e sintesi

Unreal Audio Engine dispone anche di una robusta pipeline di elaborazione degli effetti DSP (Digital Signal Processing).

L'Unreal Audio Engine è abbastanza flessibile da consentire configurazioni canoniche di effetti master come Master EQ,

Master Compression e Master Reverb, supportando anche la sperimentazione e l'esplorazione di grafici DSP e schemi di routing più personalizzati.

Una varietà di effetti sorgente e submix è stata implementata nel plug-in Synthesis and DSP Effects e vengono sempre aggiunti nuovi effetti.

Con la nostra crescente libreria DSP C++ nel nostro Signal Processing Module, è facile aggiungere nuovi effetti tramite l'API DSP e Synthesis C++ di Unreal Audio Engine.

Con altri strumenti e tecnologie affidabili in Unreal Engine tramite la riflessione aut

Per ulteriori informazioni, vedere Priorità.

.Risorsa Cooking

Quando si prepara per una piattaforma di destinazione, le risorse vengono compresse utilizzando un codec specifico sia per la piattaforma di destinazione che per la funzione utilizzata dall'audio.

Questo valore di qualità viene interpretato in modo diverso, in base al codec specifico utilizzato per una piattaforma di destinazione. USoundWave

////////////

AudioLink trasmette ;

Sources : MetaSounds, Sound Cues e Sound Waves.

Submixes : Emette i dati PCM di un particolare submix.

Audio Components : Emette dati PCM da sorgenti pre-attenuate utilizzate da un componente audio.

- Importazione di audio multicanale

1. Fare clic sul pulsante Importa.

2. Individua e seleziona i tuoi file. .wav

3. All'interno della Libreria viene visualizzata una risorsa audio a quattro canali (denominata "Surround")

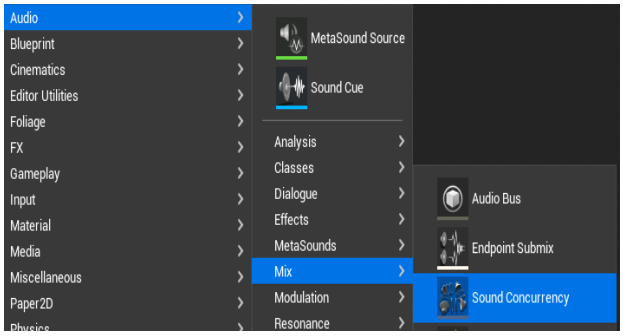
//

.Sound Concurrency Reference Guide

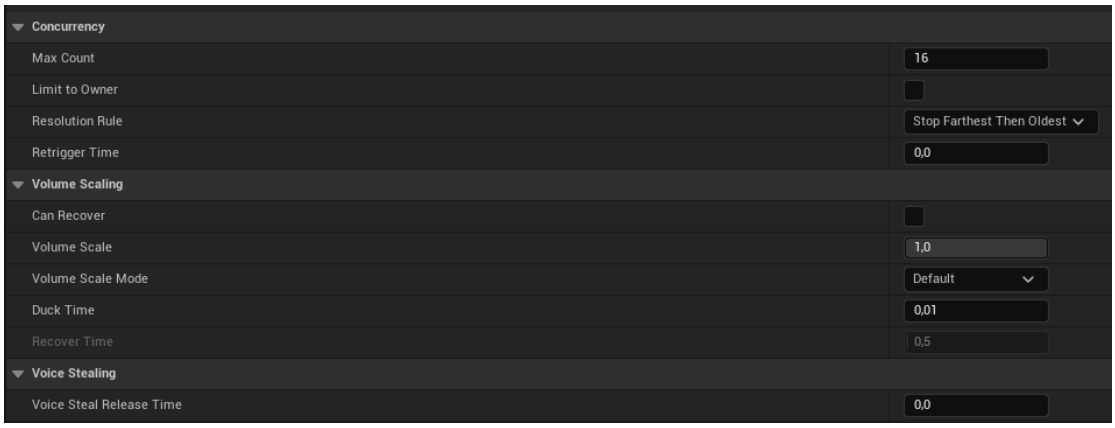
La riproduzione di troppi suoni contemporaneamente può comportare un elevato consumo di risorse e prestazioni scadenti.

Le risorse di concorrenza sonora sono uno degli strumenti principali che è possibile utilizzare per gestire questo problema.

Content Browser and (tasto destro) selecting Sounds > Mix > Sound Concurrency.



In Project Settings > Defaulta Sound Concurrency



È inoltre possibile impostare le risorse di simultaneità audio direttamente sulle risorse sorgente audio, ad esempio Metasuoni, Segnali sonori e Onde sonore.

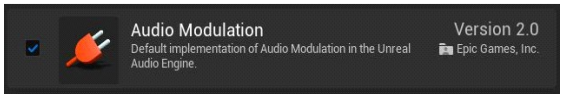
Per assegnare una simultaneità audio a una sorgente audio: Sound cue, metasound

Details > Voice Management > Concurrency > Concurrency Set

Override Concurrency

//

.Audio Modulation



Nella forma più elementare di modulazione, un utente può collegare una sorgente di modulazione a una destinazione di modulazione.

Se non viene fatto riferimento a nessuna sorgente, un valore predefinito, fornito da un parametro di modulazione, viene miscelato con il valore fornito come valore predefinito di destinazione della modulazione (vedere Sorgenti di modulazione e stadi di mixaggio).

-Generatore di modulazione (USoundModulationGenerator)

I generatori di modulazione sono una sottoclasse di modulatori in grado di generare valori proceduralmente nel tempo. I generatori possono essere referenziati da, e quindi pilotare, direttamente i bus di controllo o le destinazioni di modulazione. Un LFO e un Envelope Follower sono implementati all'interno del plug-in Audio Modulation.

- I parametri di modulazione

Forniscono il contesto per il modo in cui un valore associato a un bus, a un mix di bus, a una

destinazione e a una patch viene visualizzato, mixato e trasformato da e verso unità e valori normalizzati.

-Modulation Patches (USoundModulationPatch)

La patch di modulazione è l'analogia di Unreal con una patch bay nella modulazione hardware, ma più potente in quanto fornisce la possibilità di rimappare funzionalmente i valori del bus di ingresso.

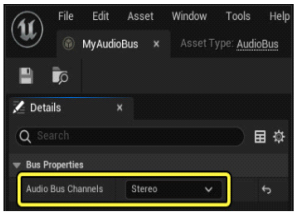
.Mix del bus di controllo (USoundControlBusMix)

Un mix di bus di controllo è un modo per pilotare e raggruppare dinamicamente i valori del bus di controllo in fase di esecuzione.

Più mix possono pilotare un singolo bus e più bus possono essere pilotati contemporaneamente da un singolo mix.

//

.Audio Bus



In Unreal Engine, i bus audio combinano le sorgenti sonore in un unico percorso del segnale.

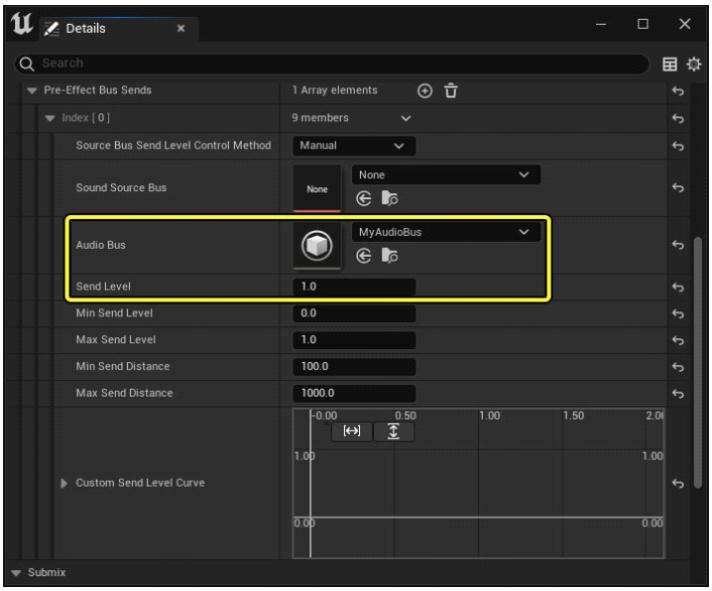
È possibile utilizzare il segnale audio combinato in due modi principali:

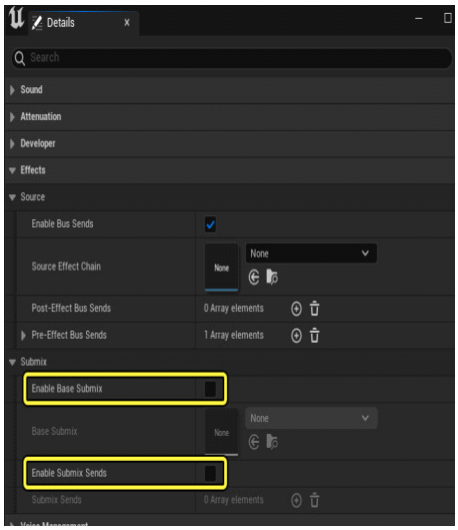
E' possibile utilizzarlo per controllare dinamicamente i parametri dell'effetto DSP (Digital Signal Process) (noto come sidechaining).

Utilizzando l'output renderizzato, è possibile creare altoparlanti o altri oggetti che producono audio all'interno del livello.

È necessario avviare un bus audio prima che possa ricevere l'audio.

Impostare i bus audio predefiniti nelle impostazioni del progetto in modo che i bus audio specificati si avviino automaticamente.



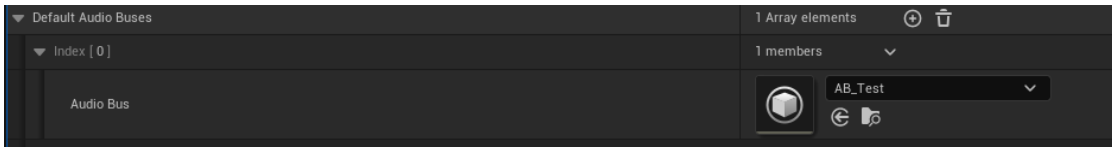


È anche possibile avviare manualmente un bus audio utilizzando il nodo Start Audio Bus.

Analogamente, è possibile arrestare un bus audio utilizzando il nodo Start Audio Bus.

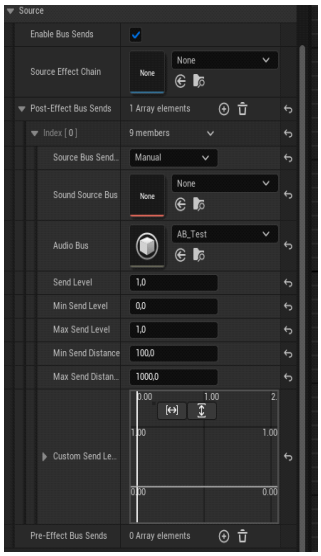
1. Crea un Bus Audio

2. Puoi impostarlo dal Settings, Project Settings > Engine > Audio > Mix > Default Audio Busses > seleziona il Bus

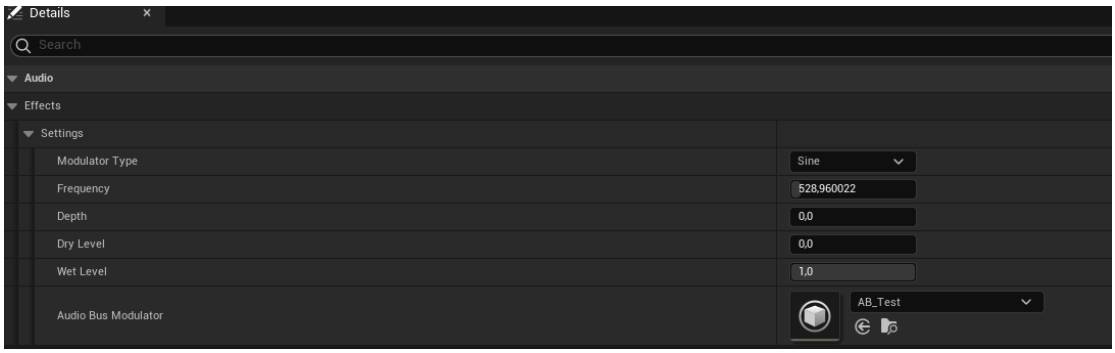


2. Puoi impostarlo dalla Sound Cue, o MetaSounds, Source > Post Effect Bus Sends > Audio Bus > Facoltativo puoi disattivare i Sub Mix in basso

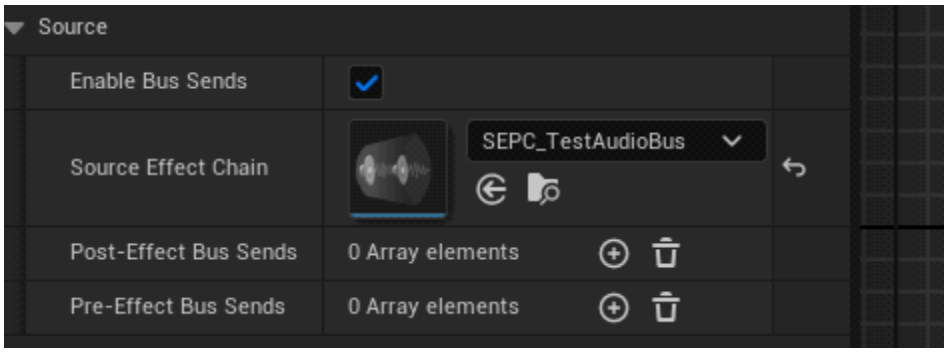
3. Imposta il Bus Level



4. Crea un Effetto (Filtro o Ring Modulation), clicca sull'effetto, assegna il bus



5. Crea un altra Sorgente sonora, assegnagli la catena di effetti con il Bus (Modulatore) che hai creato



5. Fai partire la sorgente sonora

Il Bus racchiude una catena di segnali, e serve per modulare (attraverso l'utlizzo di effetti) un altra sorgente sonora.

In pratica un Segnale audio, viene modulato da una catena di effetti che vengono a loro volta modulati da un Bus che ha assegnato una sorgente sonora)

NODO Start Audio Bus / Stop Audio Bus

.Source Buses

Un bus sorgente esegue il rendering dell'output di un bus audio.

I bus sorgente possono utilizzare le funzionalità della sorgente sonora, ad esempio la spazializzazione, l'attenuazione e la risoluzione della concorrenza.

Se la durata del bus sorgente è impostata su 0, il bus sorgente verrà riprodotto a tempo indeterminato.

Usa qualsiasi valore positivo per creare una sorgente sonora one-shot che si interromperà dopo la durata.

Se la durata del bus sorgente è impostata su 0,

il bus sorgente verrà riprodotto a tempo indeterminato.

Usa qualsiasi valore positivo per creare una sorgente sonora one-shot che si interromperà dopo la durata.

//

.SoundClasses

Una classe audio è un tipo di risorsa per il motore audio all'interno di Unreal Engine che un utente può utilizzare per raggruppare più suoni insieme, quindi modificare i parametri di tutte le onde sonore rilevanti contemporaneamente.

-Routing

Apply Ambient Volume = Se abilitato, i suoni all'interno della classe suono saranno influenzati dalle impostazioni della zona ambientale all'interno dei volumi audio, come le impostazioni del volume interno/esterno e del filtro passa basso.

-Passive Sound Mix Modifiers

Una classe sonora può avere una serie di mix di classi sonore collegati che vengono attivati automaticamente quando viene riprodotto un suono all'interno di quella classe sonora. Questi sono chiamati passivi perché non necessitano di essere attivati manualmente.

A ciascun modificatore del mix audio passivo può essere assegnata una soglia di volume minima e massima per il volume parametrico per controllare quando dovrebbe attivarsi.

-Legacy

Questa scheda contiene le impostazioni create prima che la funzione Mixer audio fosse aggiunta a Unreal Audio Engine.

Output Target = Modifica se i suoni all'interno della classe audio vengono inviati all'altoparlante principale, a un controller esterno o a un controller (con la riproduzione all'altoparlante se non è presente alcun controller).

Radio Filter Volume = Il volume dell'effetto filtro radio che si applica ai suoni all'interno della Classe Suono.

Radio Filter Volume Threshold = Il volume a cui si attiva il filtro radio

Output to Master EQ Submix = Se abilitato, invia i suoni all'interno dell'uscita per questa classe sonora al sistema di riverbero utilizzato prima che il sistema di riverbero del Mixer audio lo sostituisse

Is UISound = Se abilitato, i suoni all'interno della classe sonora sono udibili nei menu dell'interfaccia utente e durante il normale gioco.

Is Music = Se abilitato, il suono è contrassegnato come musica.

-Submix

I suoni all'interno di una classe sonora possono essere inviati a un Submix, che consente di elaborare più tipi di suono contemporaneamente con effetti submix.

Send to Master Reverb Send Amount = Quando abilitato, le impostazioni del riverbero primario vengono applicate ai suoni all'interno della Sound Class.

Default 2D Reverb = La quantità predefinita di effetto riverbero da posizionare sui suoni non spazializzati.

Default Submix = Il submix a cui viene inviato per impostazione predefinita l'output dei suoni all'interno della Sound Class. Se impostato su Nessuno, l'output verrà inviato al Submix predefinito della classe audio specificato in Impostazioni progetto

- Modulation

Se è installato il plug-in Modulazione audio, è possibile aggiungere una serie di impostazioni di modulazione a tutti i suoni all'interno di una classe sonora.

- Loading

Questa impostazione controlla il modo in cui vengono caricati i dati audio compressi.

Inherited:Quando impostata su questa opzione, la classe audio eredita il comportamento di caricamento dalla classe audio principale.

Retain on Load: Il primo pezzo di audio per i suoni all'interno della Sound Class rimane nella cache finché non viene esplicitamente eliminato.

Prime on Load: Dopo essere stati caricati, i suoni all'interno della Sound Class rimangono nella cache, ma rischiano di essere rimossi per fare spazio ad altro audio se inutilizzati per lunghi periodi di tempo

Load on Demand: Nessuna parte dell'audio viene archiviata finché la risorsa non viene riprodotta o preparata.

Force Inline: Forza i suoni all'interno della Sound Class a utilizzare solo percorsi di decodifica non streaming e nessuna parte dell'audio viene archiviata nella cache.

//

.Soundscape

Soundscape genera proceduralmente suoni ambientali che vengono trasmessi in streaming mentre il giocatore si muove in giro per il mondo. Una volta impostato, il plugin gestisce e compone questi sistemi audio in modo autonomo ed elimina la necessità di crearli manualmente.

Il sistema Soundscape è guidato da due diversi tipi di risorse: Palettes e Colors.

Il suono all'interno del tuo colore verrà riprodotto quando la condizione della tua tavolozza è soddisfatta.

Questa condizione sarà soddisfatta quando GameplayTag Soundscape.Active è impostato.

Utilizzare il Level Blueprint e un Trigger Volume per definire una regione del livello in cui riprodurre il Soundscape.

////////////

Canali degli altoparlanti

Mono, Stereo, 4.0, 5.1, 7.1

//

.MetaSounds

Forniscono ai progettisti audio la possibilità di costruire potenti sistemi audio procedurali che offrono temporizzazione e controllo accurati del campione a livello di buffer audio.

//La latenza audio è il tempo che impiega un segnale audio per essere elaborato e convertito da analogico a digitale e viceversa.

Il buffer è una piccola quantità di memoria utilizzata come alloggiamento temporaneo per i dati (i campioni audio digitali) in attesa di essere elaborati dal processore.//

I MetaSound hanno due tipi principali di risorse: MetaSound Source e MetaSound Patch.

Le patch MetaSound sono destinate a essere referenziate da altri MetaSound per incapsulare e riutilizzare le funzionalità.

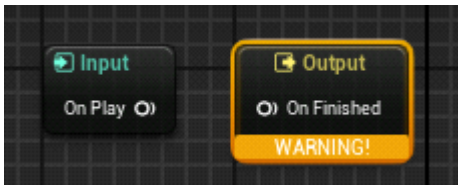
MetaSounds supporta il controllo accurato del campione in diversi modi.

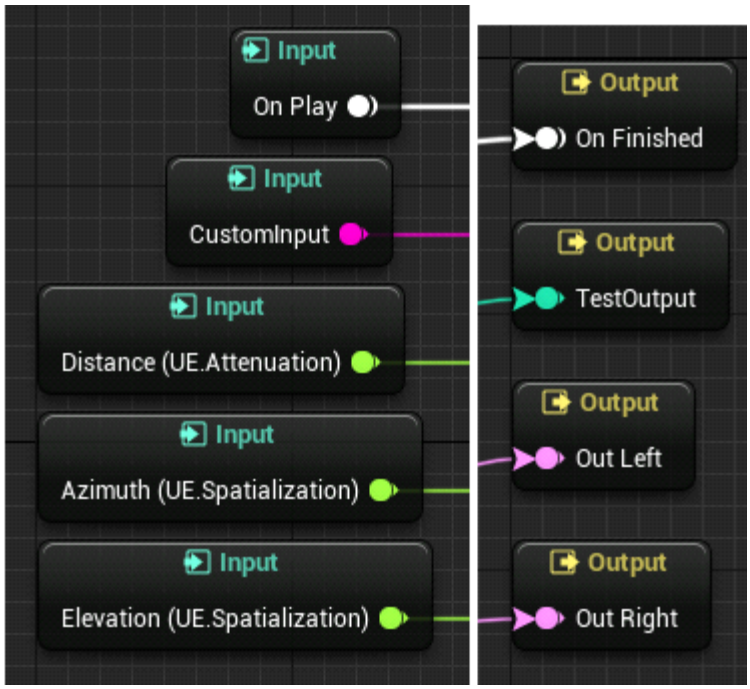
I trigger MetaSound eseguono eventi accurati per il campionamento all'interno del grafico. I trigger possono provenire da eventi di gioco, nodi MetaSound o dal grafico stesso.

Ciò significa che, al termine della riproduzione dell'onda sonora, riprodurrà senza soluzione di continuità l'onda sonora successiva in coda senza intoppi o interruzioni udibili.

.

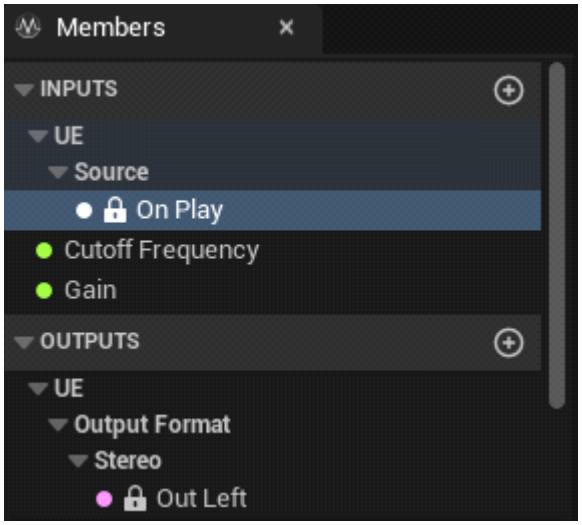
I nodi Input e Output (On Play On finished) forniscono l'accesso ai pin che in genere iniziano o terminano un flusso di dati attraverso un MetaSound.





OnFinished è simile all'event dispacter OnAudioFinished

Il nodo On Play Input ha un'uscita pin Trigger utilizzata per segnalare agli altri nodi l'inizio quando il MetaSound viene riprodotto, mentre il nodo On Finished Output ha un ingresso pin Trigger utilizzato per ricevere un segnale al termine della riproduzione del MetaSound.



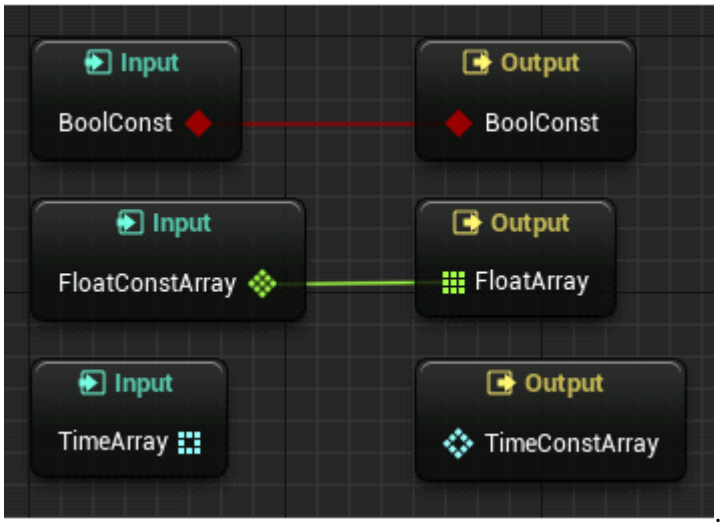
È possibile creare ulteriori ingressi e uscite nel pannello Members dell'editor MetaSound.

Per visualizzare quegli ingressi clicca tasto destro Get Input

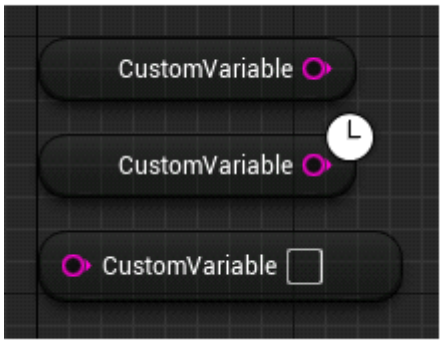
O lo trascini dalla sezione membri

È possibile utilizzare un widget Slider o Knob sui nodi Float e Time Input per modificare facilmente i valori durante la riproduzione di MetaSound per un flusso di lavoro molto più fluido.

I pin del costruttore sono indicati da icone a forma di diamante. Hanno valori di sola lettura che possono migliorare le prestazioni di un MetaSound non aggiornandosi dinamicamente in fase di esecuzione.



Puoi utilizzare i nodi Variabile per accedere e manipolare le variabili che supportano i nodi Funzione nel tuo MetaSound.



Esistono tre tipi di nodo per ogni singola variabile nella risorsa: Get Variable, Get Delayed Variable e Set Variable.

Get Variable e Set Variable forniscono l'accesso immediato in lettura o scrittura alla variabile associata, mentre Get Delayed Variable ritarda la lettura

- Interfacce

Le interfacce consentono a MetaSounds di funzionare con altri sistemi all'interno di Unreal Audio Engine fornendo l'accesso a input e output specializzati.

L'UE. Source.OneShot - Consente di creare suoni one-shot fornendo un On Finished Output che interrompe il MetaSound quando viene attivato.

L'UE. Attenuazione - Fornisce un ingresso di distanza con un valore basato sulla distanza tra il MetaSound e l'ascoltatore.

L'UE. Spazializzazione - Fornisce gli input di azimut ed elevazione con valori basati sulla posizione del MetaSound rispetto all'ascoltatore.

- Graph Node

I nodi del grafo consentono la capacità di composizione in quanto forniscono un punto di accesso ai grafici di altri MetaSuoni nel progetto.

Questi nodi hanno pin di connessione basati sugli ingressi e le uscite di un altro MetaSound,

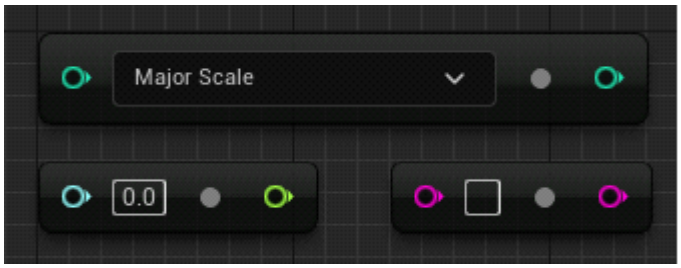
quindi è possibile utilizzarli per influenzare il design del MetaSound che lo contiene.

- Conversion Nodes

I nodi di conversione modificano un tipo di pin in un altro tipo di pin supportato,

in modo simile ai nodi di conversione Blueprint.

Audio	Float
Bool	Float or Int32
Float	Audio, Bool, Int32, or Time
Frequency Multiplier	Semitone
Int32	Bool, Float, Time, or Enum
Semitone	Frequency Multiplier
String	Transmission:Address
Time	Float or Int32
Enum	Int32



- Function Nodes

Disponibili in un'ampia varietà di tipi e forniscono le funzionalità necessarie per riprodurre file audio, mixare suoni, applicare filtri ed effetti e altro ancora.

1. BPM To Seconds

Calcola un tempo di battuta in secondi a partire dai battiti al minuto (BPM), dal Beat Multiplier e dalle divisioni di una nota intera specificati.

Input

BPM : BPM di destinazione dell'input.

Beat Multiplier : Il valore del moltiplicatore da applicare al BPM.

Division of Whole Note : Le divisioni di una nota intera.

Output

Seconds : Il tempo di battuta in uscita in secondi.

2. Envelope Follower

Emette un inviluppo da un segnale audio in ingresso.

Input

In : Il segnale audio in ingresso.

Attack Time : Il tempo di attacco (in secondi).

Release Time : Il tempo di rilascio (in secondi).

Peak Mode

MS: L'inviluppo segue un quadrato medio del segnale audio.

RMS: L'inviluppo segue un Root Meat Squared in esecuzione del segnale audio.

Picco: l'inviluppo segue i picchi del segnale audio.

Output

Envelope : Il valore dell'inviluppo di output (velocità di blocco).

Audio Envelope : Il valore dell'inviluppo di uscita (velocità audio).

3. Flanger

Il nodo Flanger applica un effetto flanger all'audio in ingresso.

Input

In Audio : Il segnale audio in ingresso a cui applicare l'effetto flanger.

Modulation Rate : La frequenza dell'oscillatore a bassa frequenza (LFO) che varia il tempo di ritardo (in Hz).

Questo valore viene bloccato alla velocità di blocco.

Modulation Depth : L'ampiezza (forza) dell'LFO che scala il tempo di ritardo.

Center Delay : La quantità di ritardo centrale (in millisecondi).

Mix Level : L'equilibrio tra segnale originale e ritardato.

Output

Out Audio : Il segnale audio in uscita a cui è stato applicato l'effetto flanger.

4. Get Wave Duration

Il nodo Ottieni durata onda restituisce la durata di una risorsa Onda sonora di input (in secondi).

Input

Wave : L'asset Sound Wave per ottenere la durata di (in secondi).

Output

Duration : La durata dell'asset Onda sonora (in secondi).

4. Get WaveTable From Bank

Recupera (o genera una WaveTable interpolata) da una determinata risorsa del banco WaveTable.

Input
WaveTableBank : L'asset WaveTable Bank da cui recuperare o generare una WaveTable interpolata.
TableIndex : Indice dell'oggetto WaveTable da recuperare.

Output
Out : Oggetto WaveTable recuperato

5. InterpTo//Molto importante

Il nodo InterpTo esegue l'interpolazione tra il valore corrente e un valore di destinazione in un periodo di tempo specificato.
Speficifa in quanto tempo raggiungere valore da noi determinato

Input
Interp Time : Tempo necessario per l'interpolazione dal valore corrente al valore di destinazione.
Target : Valore a cui eseguire l'interpolazione.

Output
Value : Il valore corrente.

6. RingMod

Il nodo RingMod esegue la modulazione ad anello con un segnale portante e un segnale modulatore.

Input

In Carrier : Il segnale audio portante.

In Modulator : Il segnale audio del modulatore.

Output

Out : Il segnale audio modulato.

7.Wave Player

Utilizzato per riprodurre una risorsa Sound Wave.

Esistono più versioni di questo nodo per supportare diverse configurazioni di canale, ad esempio Mono, Stereo, Quad (4.0), 5.1 e 7.1.

Input

Play / Stop : trigger Play e Stop avviano e interrompono la riproduzione di Wave Player in un momento preciso nel tempo.

Wave Asset : L'asset Sound Wave riprodotto da Wave Player durante la riproduzione.

Start Time : Il tempo all'interno della risorsa Sound Wave per iniziare a riprodurre il file audio.

Pitch Shift : Il Pitch Shift da utilizzare per il Wave Player. Questo è definito in unità di semitoni

Loop : Se Wave Player ripeterà il file audio o si fermerà al termine.

Loop Start : L'inizio del loop indica il momento in cui Wave Player ripeterà il file audio.

Loop Duration : La durata del loop rappresenta il tempo totale di riproduzione del loop. te l'intero file audio

Output

On Play : Si attiva quando viene attivato il pin Input Play del giocatore Wave.

On Finished : Si attiva quando Wave Player termina la riproduzione del file audio. Questo pin si attiverà nello stesso punto di campionamento in cui termina la riproduzione del file audio.

On Nearly Finished : Questo è comunemente usato per tornare indietro e scegliere una nuova variante di file audio per Wave Player.

On Looped : Trigger sul campione in cui è stato ripetuto un suono in base alle impostazioni di looping.

On Cue Point : Si attiva quando un cue point è stato analizzato in Wave Player.
I cue point sono spesso utilizzati dai sound designer per attivare eventi o punti di loop nell'audio in punti esatti nel tempo.
I cue point nel file dell'onda audio che sono più vicini tra loro rispetto alla frequenza di blocco del MetaSound si attiveranno solo sull'ultimo cue point in quel blocco.
Cue Point ID : L'ID del cue point analizzato dal file dell'onda audio importata.
Cue Point Label : L'etichetta del cue point analizzato dal file dell'onda audio importata.
Loop Percent : La posizione corrente nel file dell'onda audio all'interno dell'area di loop specificata.
Playback Location : La posizione corrente nel file dell'onda audio come frazione della lunghezza totale del file dell'onda audio.
Out x : L'uscita del canale X dell'asset Sound Wave in ingresso.

8. WaveShaper

Il nodo WaveShaper applica una forma non lineare al segnale audio in ingresso.

Input
In : Il segnale audio in ingresso a cui applicare la modellazione non lineare.
Amount : La quantità di forma d'onda non lineare da applicare.
Bias : L'offset DC da applicare prima della modellazione dell'onda.
OutputGain : La quantità di guadagno da applicare dopo l'elaborazione.
Type : Tipo di algoritmo da utilizzare per elaborare l'audio: Sinusoidale, Tangente inversa, Tangente iperbolica, Polinomio cubico o Clip rigida

Output
Out : Il segnale audio in uscita con la forma non lineare applicata.

9.0. Array da associare a RANDOM GET

Le funzioni array forniscono opzioni per manipolare gli array all'interno di un MetaSound. Ognuna di queste funzioni ha versioni diverse che consentono loro di supportare matrici di

diversi tipi di dati comuni,

tra cui: Bool, Float, Int32, String, Time, Transmission:Address, AudioBusAsset, WaveTableBankAsset e WaveAsset.

9.1 Concatenate

Il nodo Concatenate concatena due matrici su un determinato trigger

Input

Trigger : Trigger su cui concatenare le matrici di input.

Array Left / Right : Le matrici di input

Array : La matrice concatenata.

Output

Array : La matrice concatenata.

9.2 Get

Il nodo Get recupera un elemento da una matrice in corrispondenza dell'indice specificato.

Input

Triger : Trigger in base al quale recuperare l'elemento della matrice specificato.

Array : Matrice da cui recuperare un elemento.

Index : Indice dell'elemento da recuperare

Output

Element : Valore dell'elemento recuperato

9.3 Num

Il nodo Num restituisce il numero di elementi nell'array specificato.

Input

Array : Matrice i cui elementi verranno conteggiati.

Output

Num : Il numero di elementi nell'array specificato.

9.4 Random Get

Il nodo Random Get recupera in modo casuale gli elementi da una matrice di input.
Opzionalmente, è possibile fornire una serie di pesi per regolare la casualità.

Input

Next : Trigger in base al quale ottenere il valore casuale successivo nella matrice.
Reset : Trigger in base al quale reimpostare il valore di inizializzazione della randomizzazione per la matrice.
In Array : Matrice di input da cui recuperare gli elementi in modo casuale.
Weights : (Facoltativo) Matrice di input di pesi utilizzata per definire la probabilità di ogni voce recuperata.
Se non viene fornito, si assume una probabilità pari per tutti gli elementi.
Se questa matrice è più corta della matrice di input, viene ripetuta in modo che corrisponda alle dimensioni.
Seed : Il seed utilizzato per la riproduzione casuale. Il valore predefinito -1 utilizza un valore di inizializzazione casuale.
No Repeats : Il numero di elementi di cui tenere traccia per evitare di ripetersi di seguito. Ad esempio, un valore pari a 2 impedisce a questo nodo di ripetere gli ultimi due elementi selezionati.
Enabled Shared State : Se abilitato, lo stato di questo nodo viene condiviso tra le istanze di questo MetaSound per evitare di riprodurre le stesse identiche variazioni contemporaneamente.

Output

On Next : Si attiva quando viene attivato l'ingresso successivo.

On Reset : Si attiva quando viene attivato l'input Shuffle o se l'array viene mescolato automaticamente.
Value : Valore selezionato in modo casuale dalla matrice di input.

9.5 Set

Il nodo Set imposta il valore di un indice specificato in una determinata matrice.

Input
Trigger : Trigger su cui impostare il valore nella matrice.
Array : Matrice in cui verrà impostato un valore.
Index : Indice da impostare nella matrice di destinazione.
Value .Valore su cui è impostato l'indice selezionato.

Output
Array : L'array dopo che l'operazione di impostazione è stata completata.

9.6 Shuffle

Il nodo Shuffle restituisce gli elementi da una matrice in modo casuale.

Input
Next : Trigger in base al quale ottenere il valore successivo nella matrice casuale.
Shuffle : Trigger su cui mescolare manualmente l'array.
Reset Seed : Trigger in base al quale reimpostare il flusso di seme casuale.
In Array : Matrice da cui mescolare e restituire gli elementi.
Seed : Il seme utilizzato per la riproduzione casuale. Il valore predefinito -1 utilizza l'ora corrente.
Auto Shuffle : Se abilitato, mescola automaticamente quando l'array è stato letto completamente.
Enabled Shared State : Se abilitato, lo stato viene condiviso tra le istanze di questo MetaSound.

Output

On Next : Si attiva quando viene attivato l'ingresso successivo.

On Shuffle : Si attiva quando viene attivato l'input Shuffle o se l'array viene mescolato automaticamente.

On Reset Seed : Si attiva quando viene attivato l'ingresso Reset Seed.

Value : Il valore dell'elemento attualmente selezionato.

9.7 Subset

Il nodo Subset restituisce un sottoinsieme di una matrice di input.

Input

Trigger : Trigger in base al quale generare un sottoinsieme.

Array : Matrice di input da cui ottenere un sottoinsieme.

Start / End Index : Il primo e l'ultimo indice da includere nel sottoinsieme.

Output

Array : Sottoinsieme della matrice di input.

10 Print Log

Il nodo Log di stampa viene utilizzato per registrare i valori nel log di output in un determinato trigger a scopo di debug.

Input

Trigger : Trigger in base al quale scrivere il valore impostato nel log.

Label: da associare al valore registrato.

Value To Log : Valore da registrare nel log quando viene attivato.

11.0 Delays

11.1 Delay

Il nodo Delay fornisce un ritardo del buffer mono che supporta il livello dry, il livello wet e il feedback.

Per il ritardo del buffer multicanale, utilizzare il nodo Delay stereo.

Input

In : Il segnale audio a cui applicare il ritardo.

Delay Time : La quantità di tempo per ritardare l'audio (in secondi).

Dry Level : Il livello per il segnale non elaborato (dry)

Wet Level : Il livello per il segnale elaborato (bagnato).

Feedback : La quantità di feedback da utilizzare.

Max Delay Time : La quantità di feedback da utilizzare.

Output

Out : Segnale ritardato

11.2 Delay Pitch Shift

L'intonazione si sposterà, ma il suono non accelererà o rallenterà.

Input

In : Buffer audio da elaborare.

Pitch Shift : La quantità di pitch shift (in semitoni) da applicare.

Delay Length : La quantità di ritardo (compresa tra 10 e 100 millisecondi) da applicare.

La modifica di questo valore può ridurre gli artefatti in alcune regioni di pitch shift.

Output

Out : Audio Processato

11.3 Diffuser

Il nodo Diffusore applica la diffusione all'audio in ingresso.

Input

Input Audio : L'audio a cui applicare la diffusione.

Depth : Il numero di filtri da utilizzare per diffondere l'audio (compreso tra 1 e 5). Questo non verrà aggiornato durante l'esecuzione.

Feedback : La quantità di feedback da utilizzare su ciascun diffusore (tra 0 e 1).

Output

Output Audio : l'audio diffuso

11.4 Grain Delay (Complicato da usare, associalo al trigger delay)

Il nodo Grain Delay esegue la granulazione audio ritardata su un determinato buffer audio campionandolo in "granularità" e riproducendoli dopo un ritardo impostato.

Input

In Audio : Il buffer audio deve essere ritardato in termini di granulosità.

Grain Spawn : Il trigger su cui generare un nuovo granulo di audio.

Grain Delay : Il ritardo (in millisecondi, tra 0 e 2000) del successivo grano generato.

Grain Delay Range : Il delta dell'intervallo (in millisecondi) utilizzato per randomizzare il ritardo in relazione al valore centrale di Ritardo granulosità.

Grain Duration : La durata (in millisecondi) del successivo grano generato.

Grain Duration Range : Il delta dell'intervallo (in millisecondi) utilizzato per randomizzare la durata in relazione al valore centrale di Durata granulosità.

Pitch Shift : Il valore dell'altezza (in semitoni) per modificare l'altezza della granulosità di tutte le granulosità di rendering.

Pitch Shift Range : Il delta del pitch shift (in semitoni) utilizzato per randomizzare il pitch shift in relazione al valore centrale del pitch shift.

Grain Envelope ; Il tipo di inviluppo da utilizzare per i grani: Gaussiano, Triangolo, Triangolo verso il basso, Triangolo verso l'alto, Decadimento esponenziale o Attacco esponenziale

Max Grain Count : Il numero massimo di granuli di cui eseguire il rendering alla volta (tra 1 e 100).

Feedback Amount : La quantità di feedback di ogni grano. Se viene applicato il feedback, il ritardo della granulosità reimmetterà l'uscita audio in se stessa.

Output

Out Audio = Buffer audio con ritardo di granulosità

11.5 Stereo Delay

Il nodo Stereo Delay fornisce un ritardo del buffer multicanale.

Input

In Left / Right : I segnali audio in ingresso (canali sinistro/destro) a cui applicare il ritardo.

Delay Mode : Il metodo di ritardo da utilizzare:

- Normale: l'ingresso sinistro si mescola con l'uscita del delay sinistro e si alimenta all'uscita sinistra.

- Cross: l'ingresso sinistro si mescola con l'uscita di ritardo destra e si alimenta all'uscita destra.

- Ping Pong: l'ingresso sinistro si mescola con l'uscita di ritardo sinistra e alimenta l'uscita destra.

Delay Time : La quantità di tempo per ritardare l'audio (in secondi).

Delay Ratio : Il rapporto di ritardo da applicare ai canali sinistro e destro.

Ciò significa che i canali possono avere diversi livelli di ritardo.

Ad esempio, un valore pari a -1 non applica alcun ritardo al canale sinistro mentre ritarda completamente il canale destro, il che può essere utile per la decorrelazione dei canali stereo.

Dry Level : Il livello per il segnale non elaborato (dry)

Wet Level : Il livello per il segnale elaborato

Feedback : La quantità di feedback da utilizzare.

Output

Out Left / Right : Il segnale audio in uscita (canali sinistro/destro)

12.0 Dynamics

12.1 Compressor

Il nodo Compressore riduce la gamma dinamica di un segnale audio in ingresso.

Input

Audio : Il segnale audio da comprimere.
Ratio : Il rapporto di riduzione del guadagno da applicare. Ad esempio, un valore pari a 1 non produce alcuna riduzione del guadagno, mentre un valore maggiore di 1 produce una riduzione del guadagno.
Threshold dB : La soglia di ampiezza (in decibel) al di sopra della quale il guadagno verrà ridotto.
Attack Time ; La quantità di tempo necessaria affinché l'audio al di sopra della soglia (dB) raggiunga il livello di volume compresso.
Release Time : La quantità di tempo necessaria affinché l'audio al di sotto della soglia (dB) ritorni al livello di volume originale.
Tempo di lookahead : La quantità di tempo necessaria per ritardare il segnale compresso dietro il segnale di ingresso analizzato.
Knee : Un valore di decibel che determina l'intensità o la morbidezza della riduzione del guadagno. Un valore di 0 dB non fornisce alcuna fusione.
Sidechain : (Facoltativo) Un segnale audio esterno con cui controllare il compressore. Se non è impostato, viene utilizzato il segnale audio in ingresso.
Envelope Mode: Il metodo di inviluppo che il compressore utilizzerà per il rilevamento del guadagno:

MS: L'inviluppo segue un quadrato medio del segnale audio.

RMS: L'inviluppo segue un Root Meat Squared in esecuzione del segnale audio.

Picco: l'inviluppo segue i picchi del segnale audio.

Analog Mode : Se abilitato, utilizzare la modalità analogica per l'inseguitore dell'inviluppo del compressore.
Upwards Mode : Se abilitato, viene utilizzato un compressore verso l'alto al posto del compressore standard verso il basso. (Expander)
Wet/Dry : Il rapporto tra il segnale elaborato (bagnato) e quello non elaborato (secco).
Ad esempio, un valore pari a 0 è completamente asciutto e un valore pari a 1 è completamente bagnato.

Output
Audio : Il segnale audio in uscita con l'effetto compressore applicato.
Gain Envelope : La quantità di guadagno applicata al segnale.

12.2 Decibels to Linear Gain

Il nodo Decibel to Linear Gain converte un valore di guadagno logaritmico (dB) in un valore di guadagno lineare.

Input
Decibel : Il valore di guadagno logaritmico (dB) in ingresso.

Output
Linear Gain : Il valore del guadagno lineare di uscita.

12.3 Limiter

Il nodo Limiter impedisce a un segnale di superare una determinata soglia.

Input
Audio : Il segnale audio in ingresso deve essere limitato.
Input Gain dB : La quantità di guadagno (in decibel) da applicare all'ingresso prima della limitazione.
Threshold dB : La soglia di ampiezza (in decibel) al di sopra della quale il guadagno verrà ridotto.

Release Time : Il tempo necessario affinché l'audio al di sotto della soglia torni al livello di volume originale.
Knee : La modalità knee determina se le miscele di riduzione del guadagno sono Hard o Soft.

Output
Audio : Il segnale audio limitato.

12.4 Linear Gain to Decibels

Il nodo Guadagno lineare in decibel converte un valore di guadagno lineare in un valore di guadagno logaritmico (dB).

Input
Linear Gain . Il valore del guadagno lineare in ingresso.

Output
Decibel : Il valore di guadagno logaritmico (dB) dell'uscita.

13.0 Envelopes

13.1 AD Envelope

Il nodo Envelope di Active Directory genera un output del valore di inviluppo di decadimento dell'attacco quando viene attivato.

Input
Trigger : Il trigger su cui avviare la fase di attacco del generatore di inviluppo.
Attack Time : Tempo (in secondi) per raggiungere il valore massimo dell'inviluppo (1.0).
Delay Time : Tempo (in secondi) per raggiungere il valore minimo dell'inviluppo (0,0).
Attack Curve : Il fattore di curva esponenziale della fase di attacco.
Ad esempio, un valore pari a 1,0 produce una crescita lineare, un valore inferiore a 1,0 produce una crescita logaritmica e un valore maggiore di 1,0 produce una crescita esponenziale.

Decay Curve : Il fattore della curva esponenziale della fase di decadimento.
Ad esempio, un valore pari a 1,0 produce un decadimento lineare, un valore maggiore di 1,0 produce un decadimento logaritmico e un valore inferiore a 1,0 produce un decadimento esponenziale.
Loop : Se abilitato, l'inviluppo verrà ripetuto

Output
Ontrigger : Si attiva quando viene attivato l'inviluppo.
On Done : Si attiva al termine dell'inviluppo o torna indietro se il ciclo è abilitato.
Out Envelope : Valore di output dell'inviluppo.

13.2 ADSR Envelope //Out Envelope va collegata al mixer

Il nodo Envelope ADSR genera un output del valore di inviluppo attack-decay-sustain-release quando viene attivato.

Input
Trigger Attack : Il trigger su cui avviare la fase di attacco del generatore di inviluppo.
Trigger Release : Il trigger su cui avviare la fase di rilascio del generatore di inviluppo.
Attack Time : Tempo (in secondi) per raggiungere il valore massimo dell'inviluppo (1.0).
Delay Time : Tempo (in secondi) per raggiungere il valore minimo dell'inviluppo (0,0).
Sustain Level : Il livello di sustain dell'inviluppo.
Release Time : Il tempo di rilascio dell inviluppo
Attack Curve : Il fattore di curva esponenziale della fase di attacco.
Ad esempio, un valore pari a 1,0 produce una crescita lineare, un valore inferiore a 1,0 produce una crescita logaritmica e un valore maggiore di 1,0 produce una crescita esponenziale.
Decay Curve : Il fattore della curva esponenziale della fase di decadimento.
Ad esempio, un valore pari a 1,0 produce un decadimento lineare, un valore maggiore di 1,0 produce un decadimento logaritmico e un valore inferiore a 1,0 produce un decadimento esponenziale.
Release Curve : Il fattore di curva esponenziale della fase di rilascio.

Output

On Attack Triggered : Si attiva quando viene attivata la fase di attacco dell'inviluppo.

On Decay Triggered : Si attiva quando viene attivata la fase di decadimento dell'inviluppo.

On Sustain Triggered : Si attiva quando viene attivata la fase di sustain dell'inviluppo.

On Release Triggered Si attiva quando viene attivata la fase di rilascio dell'inviluppo.

On Done : Si attiva al termine dell'inviluppo.

Out Envelope : Valore di output dell'inviluppo.

Ad esempio, un valore pari a 1,0 produce un rilascio lineare

un valore maggiore di 1,0 produce un rilascio logaritmico e un valore inferiore a 1,0 produce un rilascio esponenziale.

13.3 Crossfade

Il nodo Crossfade si fonde tra gli input in modo lineare utilizzando un parametro float block-rate fornito.

Esistono più versioni di questo nodo per supportare diversi conteggi di input (tra 2 e 8).

Input

Crossfade Value : Valore che rappresenta la combinazione corrente tra gli input forniti.

Ad esempio, un valore di 0,5 per i valori di input di 2 e 4 risulterà in un output di 3.

In X : L'ingresso corrispondente alla posizione X.

Output

Out : Il valore prodotto dalla dissolvenza incrociata.

13.4 WaveTable Envelope

Il nodo WaveTable Envelope legge l'oggetto WaveTable specificato per la durata specificata.

Input

WaveTable : La WaveTable da leggere.

Play : Il trigger su cui suonare l'inviluppo.

Stop : Il grilletto su cui fermare l'inviluppo.

Pause : Il trigger su cui mettere in pausa l'inviluppo.

Duration : La durata (in secondi).

Mode : Determina il valore su cui viene completato l'inviluppo e se esegue il ciclo:

-Loop: interpola l'ultimo e il primo valore nella WaveTable e riavvia l'interpolazione dell'inviluppo al completamento.

-Hold: mantiene l'ultimo valore della tabella se la riproduzione trascorre oltre la lunghezza di WaveTable.

-Unità: interpola l'ultimo valore della tabella con 1,0 se la riproduzione trascorre oltre la lunghezza della WaveTable.

-Zero: interpola l'ultimo valore della tabella con 0,0 se la riproduzione supera la lunghezza di WaveTable.

Interpolation : Determina il modo in cui l'inviluppo interpola tra i valori di WaveTable:

-Nessuno (passaggio): nessuna interpolazione tra i valori. Viene utilizzato il valore più basso.

-Lineare: esegue l'interpolazione lineare tra i valori.

-Cubico: esegue l'interpolazione cubica tra i valori.

Output

OnFinished : Si attiva al termine dell'inviluppo.

Out : Il valore di output.

14.0 External IO

14.1 Audio Bus Reader

Il nodo Lettore bus audio emette dati audio da una risorsa bus audio.
Esistono due versioni di questo nodo per supportare diverse quantità di canale (1 o 2).

Input
Audio Bus : Risorsa del bus audio da cui leggere i dati.

Output
Out X : L'uscita audio per il canale X.

14.2 Wave Writer.

Il nodo Wave Writer scrive i segnali audio su disco. Esistono più versioni di questo nodo per supportare diverse quantità di canale (tra 1 e 8).
Il rendering del file viene eseguito a 48.000 Hz e viene salvato nella cartella Saved > AudioCaptures.

Input
Filename Prefix : Il prefisso del nome file utilizzato per il file di output.
Enabled : Se abilitato, questo nodo scriverà i segnali audio su disco.
In X : L'ingresso audio corrispondente al canale X.

15.0 Filters

15.1 Biquad Filter

Il nodo Biquad Filter fornisce un semplice filtro biquad bipolare che supporta una varietà di configurazioni.

Input
In : L'audio da filtrare biquad.
Cutoff Frequency : Il valore della frequenza di taglio.
Bandwidth : Se applicabile, controlla il valore della larghezza di banda del tipo di filtro corrente.

Gain (dB) : Il guadagno (in decibel) applicato alla banda in modalità parametrica.
Type : Il tipo di filtro biquad da utilizzare.

Output
Out : L'audio filtrato biquad.

15.2 Bitcrusher

Il nodo Bitcrusher esegue il downsampling e riduce la profondità di bit di un segnale audio in ingresso.

Input
Audio : Il segnale audio da bitcrush.
Sample Rate : La frequenza di campionamento a cui eseguire il downsampling dell'audio.
Bit Depth : La risoluzione in bit a cui ridurre l'audio.

Ouput
Audio : Il segnale audio bitcrushed .

15.3 Dynamic Filter

Il nodo Filtro dinamico filtra una banda audio in base all'intensità del segnale di ingresso.

Input
Audio : Il segnale audio da filtrare.
Sidechain : (Facoltativo) Un segnale audio esterno con cui controllare il filtro. Se non è impostato, viene utilizzato il segnale audio in ingresso.
FilterType : La forma del filtro da utilizzare: campana, ripiano basso o scaffale alto
Frequency : La frequenza centrale del filtro.
Q : La Q del filtro, o risonanza, che controlla la pendenza del filtro.
Threshold dB : La soglia di ampiezza (dB) oltre la quale il guadagno viene ridotto.
Ratio : L'importo della riduzione del guadagno da applicare.

Un valore pari a 1 non applica alcuna riduzione, mentre un valore più alto fornisce una riduzione maggiore.
Knee : Un valore di decibel che determina l'intensità o la morbidezza della riduzione del guadagno. Un valore di 0 dB non fornisce alcuna fusione.
Range : La riduzione massima del guadagno consentita (in decibel). Un valore negativo applica la compressione e un valore positivo lo trasforma in un espansore.
Gain (dB) : La quantità di guadagno di make-up (in decibel) da applicare.
AttackTime : La quantità di tempo (in secondi) impiegata dall'audio al di sopra della soglia per raggiungere il livello di volume compresso.
ReleaseTime : La quantità di tempo (in secondi) necessaria affinché l'audio al di sotto della soglia torni al livello di volume originale.
EnvelopeMode : Il metodo di inversione che segue l'inviluppo utilizzato dal compressore per il rilevamento del guadagno.
AnalogMode : Se abilitata, la modalità analogica viene utilizzata per l'inseguitore dell'inviluppo del compressore.

Output
Audio : Il segnale audio filtrato.

15.4 Ladder Filter

Il nodo Ladder Filter fornisce un filtro analogico virtuale che ha un piacevole e classico rolloff e risonanza. Rolloff pendenza

Input
In : L'audio che deve essere elaborato dal filtro ladder.
Cutoff Frequency : Il valore della frequenza di taglio.
Resonance : Il valore della risonanza del filtro.

Output
Out : La scala filtrava l'audio.

15.5 Mono Band Splitter

Il nodo Mono Band Splitter suddivide l'audio in ingresso in bande di frequenza separate.
Esistono più versioni di questo nodo per supportare diversi conteggi di input e output (tra 2 e 5).

Input
In : Il canale di ingresso audio di base.
Filter Order : La ripidità dei filtri crossover: due poli, quattro poli, sei poli o otto poli
Phase Compensate : Se abilitato, ogni banda è compensata in fase in modo che possano essere sommate correttamente.
Crossover X : La frequenza dei filtri di crossover dalla banda X alla successiva.

Output
Band X Out : L'uscita audio corrispondente al canale X.

15.6 One-Pole High Pass Filter

Il nodo One-Pole High Pass Filter è un filtro computazionalmente economico utile per una serie di applicazioni semplici, come la simulazione dell'occlusione.

Input
In : Il segnale audio da filtrare.
CutOff Frequency : Il valore della frequenza di taglio.

Output
Out : Il segnale audio filtrato.

15.7 One-Pole Low Pass Filter

Il nodo Filtro passa-basso unipolare è un filtro computazionalmente economico utile per una serie di applicazioni semplici, come la simulazione dell'assorbimento d'aria

Input

In : Il segnale audio da filtrare.

CutOff Frequency : Il valore della frequenza di taglio.

Output

Out : Il segnale audio filtrato.

15.8 Sample And Hold //Utilizzato per triggerare

Il nodo Sample And Hold emette un singolo valore del segnale audio in ingresso quando viene attivato.

Input

Sample And Hold : Il trigger su cui campionare e tenere premuto l'audio in ingresso.

In : Il segnale audio da campionare.

Output

On Sample And Hold : Si attiva quando viene attivato l'ingresso Sample and Hold.

Out : Il segnale audio campionato.

15.8 State Variable Filter //IL MIGLIORE TRA I FILTRI

Il nodo State Variable Filter fornisce un filtro analogico virtuale utilizzato in molte applicazioni di sintesi.

Input

In : L'audio che deve essere elaborato dal filtro.

Cutoff : Il valore della frequenza di taglio.

Resonance : Il valore della risonanza del filtro.

Band Stop Control : Il valore di controllo applicato all'uscita di arresto della banda.

Output

Low Pass Filter : L'uscita del filtro passa-basso.

High Pass Filter : L'uscita del filtro passa-alto.

Band Pass : L'uscita del filtro passa banda.

Band Stop : L'uscita del filtro di arresto della banda.

15.9 Stereo Band Splitter

l nodo Stereo Band Splitter suddivide l'audio in ingresso in bande di frequenza separate. Esistono più versioni di questo nodo per supportare diversi conteggi di input e output (tra 2 e 5).

Input

In L / R : I canali di ingresso audio di base.

Filter Order : La ripidità dei filtri crossover: Two Pole, Four Pole, Six Pole, Eight Pole

Phase Compensate : Se abilitata, ogni banda viene compensata in fase in modo che possano essere sommate correttamente.

Crossover X : La frequenza dei filtri di crossover dalla banda X alla successiva.

Output

Band X L / R : L'uscita audio corrispondente al canale X (nel canale sinistro/destro).

16.0 Generators

MetaSounds dispone di diversi generatori di velocità audio che forniscono opzioni per la modulazione di frequenza.

16.1 Additive Synth

Il nodo Additive Synth sintetizza l'audio sommando le sinusoidi date.

Input

Base Frequency : La frequenza sinusoidale su cui si basano le armoniche. Il valore è bloccato da 0,0 alla frequenza di Nyquist.

HarmonicMultipliers : Una serie di moltiplicatori armonici applicati alla frequenza di base.
Il numero di sinusoidi utilizzati dipende dalla dimensione di questo array.
I valori sono bloccati da 0,0 a un massimo in modo che la frequenza risultante non superi la frequenza di Nyquist.
Amplitudes : L'insieme delle ampiezze sinusoidali. I valori sono bloccati in un intervallo compreso tra 0,0 e 1,0
Phases : L'array delle fasi sinusoidali (in gradi). I valori sono bloccati in un intervallo compreso tra 0,0 e 360.
Pan Amounts : L'array di quantità di pan (usando la legge di uguale potenza).
Ad esempio, un valore pari a -1,0 è completamente a sinistra e un valore pari a 1,0 è completamente a destra.

Output
Out Left / Right Audio : L'uscita audio sintetizzata (canali sinistro/destro).

16.2 (LFO) Utilizzalo nel gain del mixer

Il nodo LFO fornisce un oscillatore a bassa frequenza che è possibile utilizzare per creare una varietà di effetti audio,
come phasing, vibrato e tremolo.

Input
Frequency : La frequenza (Hz) dell'LFO (bloccata alla velocità di blocco).
Shape : La forma d'onda dell'LFO: sinusoidale, sega, triangolare o quadrata
Min / Max Value : I valori di uscita minimo/massimo.
Sync : Ripristina la fase del generatore.
È possibile utilizzarlo in combinazione con altri nodi per ottenere generatori sincronizzati in fase di velocità audio.
Phase Offset : L'offset di fase (in gradi, compreso tra 0 e 360).
Pulse Width : L'ampiezza dell'impulso (tra 0 e 1).

Output

Out : Il valore di uscita dell'LFO (bloccato alla velocità di blocco).

16.3 Noise

Il nodo Noise genera rumore rosa o bianco.

Input

Seed : Il seme per il generatore di numeri casuali. Il valore predefinito -1 utilizza l'ora corrente.
Type : Il tipo di rumore da generare: Rumore rosa o Rumore bianco

Output
Audio : L'output del rumore generato.

16.4 Genaratore di onde

Input
Enabled : Se abilitato, l'oscillatore produce un segnale.
Bi Polar : Se abilitata, l'uscita è bipolare (-1, 1). In caso contrario, l'uscita è unipolare (0, 1).
Frequency : La frequenza di base dell'oscillatore (in Hz).
Modulation : Un ingresso di velocità audio per modulare la frequenza di base.
Sync : Ripristina la fase del generatore. È possibile utilizzarlo in combinazione con altri nodi per ottenere generatori sincronizzati in fase di velocità audio.
Phase Offset : L'offset di fase (in gradi, compreso tra 0 e 360).
Glide : La quantità di glide (interpolazione uniforme nel tempo) utilizzata quando si cambiano le frequenze.
Ad esempio, un valore di 0.0 non produrrà alcuna planata e un valore di 1.0 produrrà molta planata.
Type : tipi di onda

Output
Audio : Il segnale audio sinusoidale.

16.5 WaveTable Oscillator

Il nodo WaveTable Oscillator legge la WaveTable specificata alla frequenza fornita.

Input

Play : Il trigger su cui suonare l'oscillatore (block rate).

Stop : Il trigger su cui arrestare l'oscillatore (block rate).

WaveTable : La WaveTable da leggere.

Freq : Numero di volte al secondo per campionare un periodo di WaveTable della WaveTable. Questa frequenza deve essere impostata tra -20000 Hz e 20000 Hz.

Sync : Trigger su cui riavviare la riproduzione della WaveTable sul limite del trigger (frequenza di campionamento).

Phase Modulator : La sorgente audio da utilizzare per la fase di oscillazione modulante della WaveTable fornita.

Un valore pari a 0 non produce alcuna modulazione di fase e un valore pari a 1 produce una lunghezza completa della tabella (360 gradi) di sfasamento.

Output

Out : Il buffer audio di output.

17.0 Math

17.1 Abs

Il nodo Abs restituisce il valore assoluto di un dato input. Ad esempio, un valore di input pari a -2.0 restituirà 2.0.

17.2 Clamp

Il nodo Clamp restituisce il valore bloccato dell'input all'interno dell'intervallo di valori specificato.

Input

In : Il valore di ingresso da bloccare.

Min / Max : Il valore min / max a cui bloccare il valore di ingresso.

Output

Value : Il valore bloccato.

17.3 Divide

Il nodo Dividi esegue operazioni di divisione sugli input forniti.

17.4 Filter Q To Bandwidth

Il nodo Filtro Q a larghezza di banda converte un determinato parametro Q (fattore di qualità) per il controllo del filtro in un valore di larghezza di banda.

17.5 Linear To Log Frequency

Il nodo Linear To Log Frequency converte un valore di input dello spazio lineare in un output di frequenza logaritmica.

Input

Value : Il valore di input lineare da mappare a un'uscita di frequenza logaritmica.

Min / Max Domain : Il dominio min / max per il valore di input.

I valori di input e output sono bloccati su questo dominio.

Min / Max Range : L'intervallo positivo min / max per il valore della frequenza di uscita (Hz).

I valori di ingresso e di uscita sono bloccati in questo intervallo.

Output

Frequency : La frequenza di uscita (Hz) che è la frequenza logaritmica del valore di ingresso.

17.6 Log

Il nodo Log calcola il logaritmo float-base di un altro float.

17.7 Map Range

Il nodo Mappa intervallo esegue il mapping di un valore di input in un determinato intervallo di input a un determinato intervallo di output.

Input
In : Valore di input da mappare.
In Range A / B : L'intervallo di valori di input minimo e massimo.
Out Range A / B : L'intervallo di valori di uscita minimo e massimo.
Clamped : Se abilitato, l'ingresso verrà bloccato all'intervallo di ingresso specificato.

Output
Ouput value : Valore di output mappato.

17.8 Max

Il nodo Max restituisce il valore più alto (max) di A e B.

17.9 Min

Il nodo Min restituisce il valore inferiore (min) di A e B.

17.9.1 Modulo

Il nodo Modulo restituisce il resto di un'operazione di divisione di due valori Int32 specificati.

17.9.2 Multiply

Il nodo Moltiplica esegue operazioni di moltiplicazione sugli input forniti.
Questo nodo può essere utilizzato per fornire effetti di tipo modulazione ad anello e modulazione dell'ampiezza della velocità audio.

17.9.3 Power

Il nodo Power eleva un determinato Float alla potenza di un altro Float.

17.9.4 Subtract

Il nodo Sottrai esegue operazioni di sottrazione sugli input forniti.

18.0 Mix

MetaSounds fornisce due tipi di nodo per creare mix audio, il nodo Mixer Mono e il nodo Mixer stereo.
Questi nodi hanno diverse versioni che supportano da 2 a 8 buffer audio di ingresso,
che vengono combinati in un singolo buffer utilizzando i valori di guadagno corrispondenti del canale di ingresso e sommandoli insieme.

19.0 Music

19.1 MIDI To Frequency

Il nodo Frequency To MIDI converte un valore di frequenza (in Hz) in un valore di nota di scala MIDI standard (dove il Do centrale è 60).

Input
Frequency In : Il valore della frequenza di ingresso (in Hz).

Ouput
MIDI : Il valore della nota MIDI in uscita.

19.2 MIDI Note Quantizer

Il nodo Quantizzatore note MIDI quantizza una nota MIDI fino alla nota più vicina che corrisponde ai criteri forniti.

Input

Note In : La nota MIDI da quantizzare.

Root Note : La nota MIDI da trattare come la radice.

Ad esempio, un valore pari a 0 è uguale a C, un valore pari a 1 è uguale a C#/DB e così via.

L'ottava non ha importanza. Inoltre, tutti i valori inferiori a 0 vengono bloccati su 0.

Scale Degrees : Matrice contenente un insieme di note (in ordine crescente) che rappresentano i semitoni.

L'array deve iniziare con 0.0 per rappresentare la nota fondamentale e il valore più alto nell'array deve rappresentare la nota fondamentale in un'ottava superiore,

ad esempio 12.0 per un intervallo di una singola ottava o 24.0 per un intervallo di due ottave.

Output

Note Out : La nota quantizzata

19.3 MIDI To Frequency

Il contrario di 19.1 MIDI To Frequency

19.4 Scale to Note Array

Il nodo Scala per serie di note restituisce una matrice di numeri a virgola mobile che rappresentano le note della scala selezionata.

Input

Scale Degrees : La scala preimpostata per ottenere le note.

Chord Tones Only : Se questa opzione è attivata, viene restituito un sottoinsieme della scala che

rappresenta i toni degli accordi.
Ad esempio, ridimensionare i gradi di 1, 3, 5 e 7.

Output
Scale Array Out : Rappresentazione in forma di matrice della scala come semitoni sopra la radice.
Il set è comprensivo a entrambe le estremità (a partire da 0.0f e terminando a 12.0f).

20.0 Random

MetaSounds fornisce diversi nodi casuali classificati in base al tipo di valore di output, tra cui Bool, Float, Int e Time.
Questi nodi restituiscono un valore casuale dal tipo di input e da un valore di inizializzazione.

Input
Next : Trigger su cui generare il valore casuale successivo.
Reset : Il trigger su cui reimpostare la sequenza casuale con il seme fornito.
Seed : Valore di inizializzazione da utilizzare per la randomizzazione.
Il valore predefinito -1 utilizza un valore di inizializzazione casuale.
Min / Max : Intervallo inclusivo del valore casuale.

Output
On Next : Si attiva quando viene attivato l'ingresso successivo.
On Reset : Si attiva quando viene attivato l'ingresso Reset.
Value : Il valore generato in modo casuale.

21.0 Spatialization

21.1 ITD Panner

Il nodo ITD Panner esegue il panning di un segnale audio in ingresso utilizzando un metodo di ritardo interaurale.

La differenza di tempo interaurale (o ITD) quando riguardante esseri umani o animali, è la differenza di tempo di arrivo di un suono tra due orecchie.

Input

In : L'audio di input da spazializzare.

Angle : L'angolo della sorgente sonora (in gradi).

Un valore di 90 gradi è davanti, 0 gradi è destra, 270 gradi è dietro e 180 gradi è sinistra.

Distance Factor : Il fattore di distanza normalizzato (da 0,0 a 1,0) da utilizzare per i calcoli della differenza di livello interaurale (ILD).

Un valore pari a 0,0 è vicino e un valore pari a 1,0 è lontano.

Più lontano è l'audio in ingresso, minore è la differenza di livelli (guadagno) tra le orecchie.

Head Width : La larghezza (in centimetri) della testa dell'ascoltatore.

Output

Out Left / Right : Uscita audio

21.2 Mid-Side Decode / Encode (Il contrario con INPUT IN LEFT e IN RIGHT)

Il nodo Mid-Side Decode converte un segnale stereo con canali mid e side in canali sinistro e destro.

Input

In Mid / Side : I canali audio da convertire.

Spread Amount : La quantità di diffusione stereo.

Un valore di 0,0 non produce alcuno spread, un valore di 0,5 è il segnale originale e un valore di 1,0 è completamente ampio.

Equal Power : Se abilitato, verrà mantenuta una relazione di potenza uguale tra i canali audio in ingresso.

Output

Out Left / Right : Uscita audio

21.3 Stereo Panner

Il nodo Panner stereo esegue il panning di un segnale audio in ingresso sulle uscite sinistra e destra.

Input

In : Il segnale audio in ingresso da eseguire il panning.

Pan Amount : La quantità di panning del segnale audio (-1.0 è completamente a sinistra, 1.0 è completamente a destra).

Panning Law :

-Uguale potenza: la potenza del segnale audio è costante durante il panning.

-Lineare: l'ampiezza del segnale audio è costante durante il panning.

Output

Out Left / Right : Uscita audio

22.0 Triggers

22.1 Trigger Accumulate

Il nodo Trigger Accumulate si attiva quando tutti i trigger di input collegati, sono stati attivati almeno una volta, in fila uno dopo
l'altro.

Input

In X : Gli ingressi di attivazione.

Auto Reset : Trigger in base al quale viene reimpostata l'accumulazione su questo nodo.

Ouput

Out : Si attiva quando tutti i trigger di input sono stati accumulati.

22.2 Trigger Any

Il nodo Trigger Any si attiva quando uno qualsiasi dei trigger di input collegati viene attivato.

Input

In X : Gli ingressi di attivazione.

Output

Out : Si attiva quando viene attivato un trigger di input.

22.3 Trigger Compare//Molto utile per settare stati

Il nodo Trigger Compare attiva true o false in base al confronto degli input collegati.

Input

Compare : Il trigger su cui confrontare A e B.

A / B : I valori da confrontare.

Type : Il tipo di confronto da effettuare: Uguale a, Non uguale a, Minore di, Maggiore di, Minore o Uguale o Maggiore di o Uguale a

Ouput

True/False : Si attiva con il trigger risultante dopo aver effettuato il confronto.

22.4 Trigger Control

Il nodo Controllo trigger fornisce controlli per consentire o bloccare il passaggio dei segnali di trigger all'uscita.

Input

Trigger In : Il trigger di input da controllare.

Open : Trigger su cui consentire il passaggio del trigger di input.

Close : Trigger su cui bloccare il passaggio del trigger di input.

Toggle : Trigger su cui attivare o disattivare lo stato aperto/chiuso di questo nodo.

Start Closed : Se abilitato, il nodo verrà avviato in uno stato chiuso.

Ouput

Trigger Ouput : Trigger di output che passa se il nodo è aperto.

22.5 Trigger Counter (tiene il conteggio dei trigger attivati)

Il nodo Trigger Counter conta le attivazioni dei trigger di input collegate.
Questo nodo è utile per sequenziare gli input di matrice e un gran numero di altri casi d'uso procedurali.

Input

In : Trigger di input per il quale contare le attivazioni.
Reset : Il trigger su cui azzerare il contatore e riportare il valore al valore iniziale.
Start Value : Il valore con cui iniziare all'inizializzazione e al momento del ripristino.
Step Size : Valore da aggiungere al valore corrente per ogni trigger di input. Questo può essere negativo.
Reset Count : Il numero di trigger di input da contare prima del ripristino automatico.

Ouput

On Trigger : Si attiva quando viene attivato il trigger di input e il conteggio viene aggiornato.
On Reset : Si attiva quando viene attivato il reset dell'ingresso o se il contatore si azzera automaticamente.
Count : Il numero di trigger corrente.
Value : Il valore corrente.
Se il valore è impostato su 0, il nodo non verrà reimpostato automaticamente.

22.6 Trigger Delay

Il nodo Ritardo trigger esegue un trigger dopo un determinato tempo di ritardo dall'esecuzione più recente del trigger di input.

Input

In : L'input trigger per ritardare.

Reset : Trigger su cui reimpostare il ritardo e cancellare l'attività di esecuzione, se in sospeso.

Delay Time : La quantità di tempo per ritardare l'attivazione (in secondi).

Output

Out : L'attivazione dell'uscita ritardata.

22.7 Trigger Filter /O uno o l'altro

Il nodo Filtro trigger risponde a un trigger attivando in modo casuale uno dei due trigger di output.

Input

Trigger : Il trigger su cui attivare in modo casuale un trigger di uscita.

Reset : Trigger su cui reimpostare la sequenza casuale con il seme fornito.

Seed : Valore di inizializzazione da utilizzare per la randomizzazione. Un valore pari a -1 utilizzerà un seme casuale.

Probability : La probabilità che un particolare trigger di uscita venga attivato.

Ad esempio, un valore di 0,0 è sempre testa, un valore di 1,0 è sempre croce e 0,5 è uguale a probabilità. Impostalo su 0.5

Ouput

Heads / Tails : I possibili trigger di output.

22.8 Trigger On Threshold

Si attiva quando l'audio in ingresso supera un determinato valore di soglia mentre si sposta nella direzione specificata.

Input

In : Il segnale audio in ingresso.

Threshold : Soglia in corrispondenza della quale attivare l'output.

Type : Il tipo di soglia di attivazione: Fronte di salita, Fronte di discesa o Soglia Abs

Ouput

Out : Trigger della soglia di uscita.

22.9 Trigger On Value Change//si attiva quando viene modificato un valore

Viene attivato quando viene modificato un determinato valore

Input

Value : Valore di input da controllare.

Ouput

Trigger : Il trigger di output.

22.9.0 Trigger Once

Il nodo Trigger Once si attiva la prima volta che viene attivato un trigger di input e ignora tutte le altre occorrenze, a meno che non venga reimpostato.

Input

Trigger In : Il trigger di input.

Reset : Trigger su cui aprire il nodo e consentire il passaggio di un altro trigger.

Start Closed : Se abilitato, il nodo viene chiuso all'inizio della riproduzione.

Ouput

Trigger Out : Il trigger di output.

22.9.1 Trigger

Il nodo Trigger Pipe ritarda l'esecuzione di tutti i segnali di trigger in ingresso del ritardo

specificato.

Questo nodo è simile al nodo Ritardo trigger, ma non reimposta il timer alla ricezione di trigger aggiuntivi.

22.9.2 Trigger Repeat

Il nodo Trigger Repeat emette periodicamente un trigger a una determinata frequenza di campionamento accurata e arbitrariamente precisa.

Input
Start / Stop : Trigger su cui avviare o interrompere l'esecuzione di trigger di output periodici.
Period : Il periodo da attivare (in secondi).

Ouput
RepeatOut : Trigger di output generato periodicamente.

22.9.3 Trigger Route (AUDIO)

Il nodo Trigger Route indirizza valori di input diversi a un singolo valore di output.

Input
Set X : I trigger di input che inizializzano il routing. L'opzione predefinita è 0.
Value X : Valore di input da instradare all'output quando viene attivato dal trigger corrispondente

Ouput
On set : Si attiva quando viene impostato uno qualsiasi dei trigger di input.
Value : Il valore di uscita impostato da un trigger di ingresso attivato.

22.9.4 Trigger Select

Input

In : Il trigger da attraversare.

Index : Indice di output da attivare. Se il valore fornito non è compreso nell'intervallo, verrà ignorato.

Ouput

Out x : L'output si attiva in corrispondenza dell'indice X.

22.9.5 Trigger Sequence

Il nodo Sequenza trigger risponde ai trigger di input inviando il trigger di output successivo in una sequenza.

Inputs

In : Il trigger di input.

Reset : Trigger su cui reimpostare la sequenza su 0.

Loop : Se abilitata, la sequenza torna automaticamente a 0 una volta che tutti i trigger sono stati attivati.

Ouputs

Out X : Il trigger emette in sequenza.

22.9.6 Trigger Toggle

Il nodo Attiva/disattiva attiva/disattiva un valore Bool.

Input

On / Off : Il trigger su cui attivare o disattivare l'uscita Bool.

Init : Gli stati iniziali di Bool.

Ouput

Out : Si attiva quando il Bool viene attivato/disattivato.

Value : Il valore corrente di Bool.

23 Value

Il nodo Valore imposta un valore variabile su un trigger.

Input

Set : Trigger su cui scrivere il valore impostato nell'output.

Reset : Trigger su cui reimpostare il valore al valore iniziale.

Init Value : Valore su cui inizializzare il valore di output.

Target Value : Il valore su cui impostare l'output quando viene attivato.

Ouput

On set : Si attiva quando il valore è impostato.

On Reset : Si attiva quando il valore viene reimpostato.

Output Value : Il valore di uscita corrente.

////////////

Aggiungere lo stesso nome al parametro In nome. Aggiungi una Wave al nodo.

2. Attenuation Node = Il nodo Attenuazione controlla il volume di un suono in base alla distanza della sorgente sonora dall'ascoltatore.

3. Branch Node = Il nodo Branch seleziona un nodo di input in base al valore di un parametro booleano denominato True, False o Parameter Unset.
Parametro Unset è l'output che viene riprodotto quando il valore di un parametro denominato non è stato impostato.
Ad esempio, se il parametro viene salvato come nel nodo Branch, ma l'utente non ha chiamato, Sound Cue utilizzerà il pin di input Parameter Unset, non il pin True o False.

4. Concatenator Node = Il nodo Concatenatore viene utilizzato per riprodurre i suoni in sequenza

5. Continuous Modulator Node = Il nodo Modulatore continuo fornisce un modo per controllare i parametri denominati per la modulazione del volume e dell'intonazione in tempo reale.
Default è il Valore predefinito utilizzato se non viene trovato alcun parametro.
Modalità Param
La modalità con cui viene applicato il parametro:
MIN/MAX INPUT PRE GAIN OFFSET
MIN/MAX OUTPUT POST GAIN OFFSET

Normale: Blocca il valore di input all'intervallo (MinInput, MaxInput), quindi esegue il mapping all'intervallo (MinOutput, MaxOutput).

Assoluto: Uguale a Normale, ad eccezione del fatto che il valore di input viene considerato come un valore assoluto.

Diretto: Utilizza direttamente il valore di input, senza ridimensionamento o riferimento ai valori di input o output Min o Max.

6. Crossfade by Distance Node/Param = consente di eseguire la dissolvenza tra due o più suoni in ingresso in base alla distanza tra l'origine del segnale sonoro e l'ascoltatore.
Ad esempio, uno sparo udito a distanza suonerà diverso da uno sparo da vicino.
Questo nodo può calcolare la distanza dell'ascoltatore dall'inquadratura e riprodurre il suono in base a una distanza breve o lunga, o a un mix dei due.
Questo nodo differisce dal nodo Attenuazione in quanto il suono verrà riprodotto solo se rientra nelle impostazioni di distanza minima e massima.
Fade in Distance Start = Lasciala a 0, La distanza dalla posizione dell'ascoltatore
alla quale il suono inizia ad essere udibile.
Fade in Distance End = La distanza dalla posizione dell'ascoltatore alla quale il suono raggiunge il massimo volume.
Fade Out Distance Start = La distanza dalla posizione dell'ascoltatore alla quale il suono inizia a svanire.
Fade Out Distance End = La distanza dalla posizione dell'ascoltatore alla quale il suono non è più udibile.
Volume = Il volume massimo del suono in ingresso.

Il nodo Dissolvenza incrociata per param non supporta valori negativi.

7. Delay Node = Il nodo Delay inserisce un ritardo nella catena Sound Node per causare una pausa prima che il suono in ingresso venga passato all'output.
L'importo del ritardo è un valore casuale compreso tra i valori di Ritardo minimo e Ritardo massimo specificati nel pannello Dettagli.

8. Dialogue Player Node = Il nodo Dialogue Player è un modo per impostare una risorsa Dialogue Wave da riprodurre in condizioni specifiche.

9. Doppler Node = Il nodo Doppler simula un effetto Doppler, in cui l'altezza(Pitch) di un suono cambia in base al movimento della sorgente rispetto all'ascoltatore.
Doppler Intensity = Valore moltiplicatore per l'intensità dell'effetto Doppler. Aumentate il valore per rendere l'effetto più pronunciato.
Use Smoothing = Quando questa opzione è abilitata, il nodo esegue un'interpolazione uniforme sull'effetto Doppler.
Smoothing Interp Speed = La velocità con cui viene interpolata la scala dell'altezza.

10. Enveloper Node = Crea un inviluppo in cui il volume e l'altezza di un suono vengono manipolati in un periodo di tempo specificato tramite curve di distribuzione.
Modulation
Pitch Min = L'altezza minima per il suono in Input.
Pitch Max = L'altezza massima per il suono in Input.
Volume Min = Il volume minimo per il suono in Input.
Volume Max = Il volume minimo per il suono in Input.

11. Group Control Node = Controlla quali suoni vengono riprodotti in base al numero di suoni attivi, dando priorità al primo gruppo disponibile.
Group size = Il numero massimo di sorgenti di onde sonore consentite per ciascun gruppo, meno 1 (il primo nodo di onde sonore è il nodo 0).
Ogni volta che verrà riprodotto un suono, verrà scelto quello che ha lo slot attivabile viene scelto.

12. Looping Node = Il nodo Looping viene utilizzato per riprodurre in loop il suono all'infinito o per un determinato numero di volte.

13. Mature Node = Utilizzato per controllato i contenuti Maturi.

14. Mixer Node = Il nodo Mixer definisce il modo in cui i suoni simultanei in un Sound Cue vengono mixati insieme fornendo il controllo del volume per ogni singolo ingresso.

15. Modulator Node = Il nodo Modulatore viene utilizzato per randomizzare la modulazione del volume e dell'intonazione.
È possibile impostare i valori massimo e minimo per determinare l'intervallo di randomizzazione. Quando il segnale sonoro viene attivato, seleziona in modo casuale un valore all'interno di tale intervallo.

16. Oscillator Node = Viene utilizzato per introdurre l'oscillazione continua del passo e del

volume.

17. Quality Level Node = Per impostazione predefinita, non ci sono livelli di qualità; il progettista deve aggiungerli alle impostazioni del progetto per popolare gli input del nodo.
Una volta compilato, il nodo può essere utilizzato per passare da un livello all'altro di complessità del grafico SoundCue.
Ciò fornisce ai progettisti un modo per utilizzare grafici SoundCue più semplici per piattaforme a bassa potenza..
Andare in Project Settings < Engine < Audio < Quality, scegli (o scrivi o aggiungi) qualità in base al Display Name(es Low) e max Channels es 16 etc.

18. Random Node = Il nodo Casuale viene utilizzato per attivare in modo casuale un nodo dall'elenco dei nodi di input connessi.
I valori di Weight controllano la probabilità che un nodo venga attivato rispetto agli altri nodi connessi al nodo casuale.
Preselect at Level Load = Se maggiore di 0, ad ogni caricamento di livello il numero specificato di ingressi
verrà selezionato in modo casuale e il resto verrà rimosso. Questo può essere usato per ridurre l'utilizzo della memoria di segnali casuali di grandi dimensioni.
Randomize Without Replacement = Quando abilitato, l'intero elenco dei nodi sonori si esaurirà prima di ripetere un suono.

19. SoundClass Node = Il nodo SoundClass consente di rimappare l'asset Sound Class utilizzato con il nodo di input.
BP è Set Wave Parameter

20. Switch Node = Il nodo Switch seleziona un nodo di input in base al valore di un parametro integer di runtime.
Il parametro specificato deve essere dichiarato nel codice o nei blueprint.
Set Int Parameter
Molto utile per i FOOTSTEPS

//

.Spatialization
Nota anche come localizzazione del suono, si ottiene utilizzando
Orientamento = L'orientamento relativo tra l'ascoltatore e la sorgente sonora.
Attenuazione = La distanza tra l'ascoltatore e la sorgente sonora.
Propagazione = I percorsi che un suono percorre dalla sua sorgente all'ascoltatore.
Ostruzione e occlusione = Un suono può entrare in collisione con altri oggetti mentre viaggia verso l'ascoltatore.
Questi oggetti agiscono come ostruzioni al suono.
Riverbero = Un suono può prendere diversi percorsi alternativi per raggiungere l'ascoltatore, creando un effetto eco percepito.

I tre metodi principali di spazializzazione sono il
1. Panning = Può essere ottenuto regolando il guadagno relativo tra i diversi canali audio

Questa esperienza psico-acustica è indicata come differenza di livello interaurale, dove "livello" è un altro termine spesso usato per il volume.
Il panning lineare è la tecnica di panning più semplice in cui il guadagno relativo del suono tra i canali accoppiati viene interpolato linearmente e il guadagno totale viene mantenuto costante.
Il panning lineare è semplice da calcolare,
ma ha uno svantaggio fondamentale: il volume (loudness), che è il termine usato per descrivere l'esperienza percettiva dell'ampiezza del suono,
non è determinato dall'ampiezza effettiva di un segnale audio ma dalla potenza del segnale audio.
Questo cambiamento di potenza durante il panning influenzerà l'esperienza audio complessiva del gioco, poiché il suono sembrerà muoversi attraverso il campo degli altoparlanti in modo non lineare.
Il metodo di panning a potenza uguale contrasta i cali di intensità presenti nel panning lineare aumentando perfettamente il volume per mantenere una potenza costante.
Questa tecnica mantiene costante il volume mantenendo il vincolo di potenza durante il pan, invece di mantenere costante l'ampiezza.
2. Soundfield spatialization
Ambisonics"campi sonori", ma sono anche ampiamente conosciuti come "ambisonici".
Le armoniche sferiche possono essere pensate come una serie di funzioni in grado di definire con precisione un campo tridimensionale.
3. Binaural audio spatialization

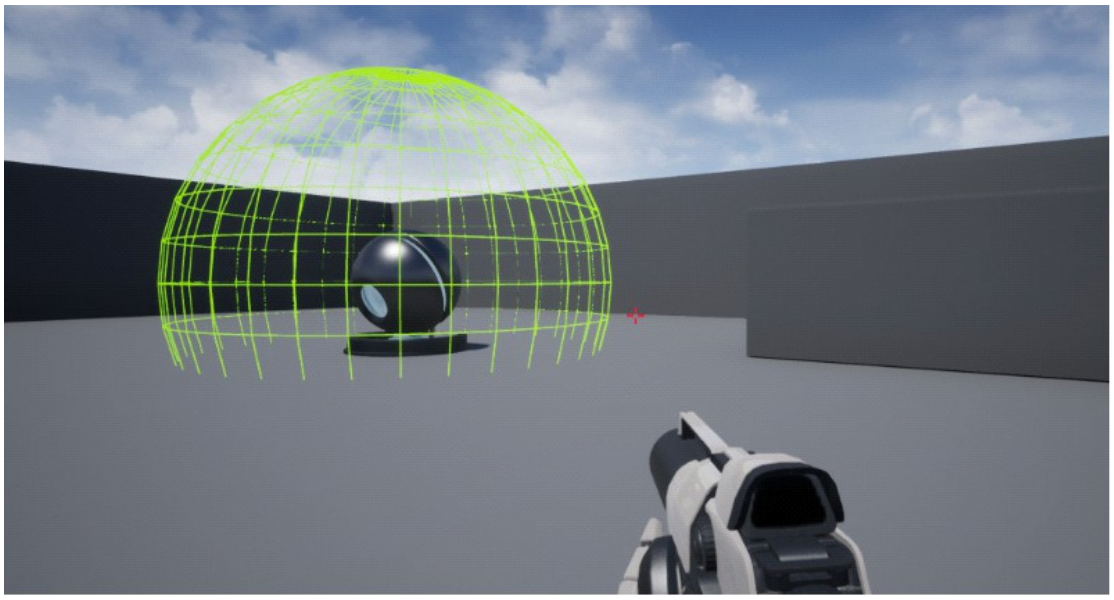
.Sound Attenuation

1. Attenuation (Volume)
Il modo in cui il volume del suono si ridurrà (attenuerà) man mano che l'ascoltatore si allontana da esso.
- Attenuation Function = definisce la funzione che determina la velocità di attenuazione sulla distanza.
.Linear = il volume si attenua linearmente e quindi le variazioni di volume sono costanti man mano che l'ascoltatore si avvicina e si allontana dalla sorgente.
Utile per la dissolvenza incrociata tra suoni di sottofondo di grandi dimensioni che non richiedono impostazioni di decadimento spaziale 3D strettamente focalizzate.
.Logarithmic = Il volume si attenua in modo tale che le variazioni di volume siano maggiori a distanze ravvicinate e minori a distanze lontane.
Questa funzione è utile per i suoni spot che richiedono un buon posizionamento 3D, pur essendo udibili a distanze ragionevoli.
.Inverse = Simili a quelle della curva logaritmica, ma sono più esagerate.
Questa funzione è utile per i suoni che devono essere udibili solo a grandi distanze, ma che diventano significativamente più forti man mano che l'ascoltatore si avvicina alla sorgente.
.Log Reverse = Le variazioni di volume sono minori a distanze ravvicinate, con differenze più marcate a distanze lontane.
Questa funzione è utile per i suoni che devono essere forti a distanze maggiori.
.Natural Sound = Comportamento di decadenza più "naturalistico" per produrre comportamenti che corrispondano più strettamente alla realtà.
.Custom = funzione personalizzata
- Attenuation Shape = Definisce la forma utilizzata per definire i punti di attenuazione minima e massima del suono

- Inner Shape Areas = Queste proprietà definiscono l'area interna della forma di attenuazione, ovvero l'area in cui il suono sarà al massimo volume (in altre parole, completamente non attenuato)
- Inner Radius = Definisce l'area interna della forma di attenuazione della sfera
- Falloff Distance = Definisce la dimensione dell'area esterna della forma di attenuazione.
- Capsule Half Height = Questa proprietà definisce l'altezza dell'area interna della forma di attenuazione della capsula
(definita "mezza altezza" in quanto viene misurata dall'origine del suono che si trova nel punto medio della capsula).
- Capsule Radius = Questa proprietà definisce il raggio dell'area interna della forma di attenuazione della capsula.
- Extents = Definisce le dimensioni della forma di attenuazione del riquadro (misurate dall'origine del suono). Ci sono 3 valori per questa proprietà: x, y, z.
- Cone Radius = Definisce la lunghezza dell'area interna della forma di attenuazione del cono (misurata dall'origine del suono).
- Cone Angle = Definisce la larghezza (in gradi) dell'area interna della forma di attenuazione del cono
- Cone Falloff Angle = Definisce la larghezza (in gradi) dell'area esterna della forma di attenuazione del cono
- Cone Offset = Definisce la distanza direttamente dietro l'origine del suono da dove si desidera che inizi il cono
- Enable Volume Attenuation = Il suono si attenuerà con la distanza in base alle impostazioni definite sopra

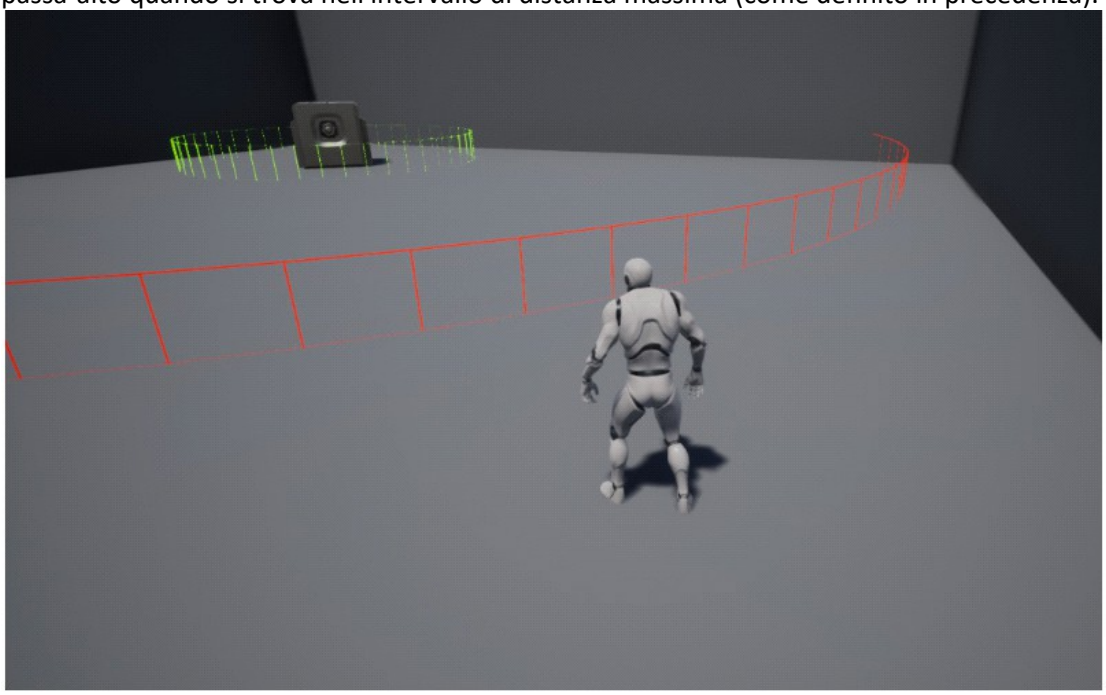
2. Attenuation (Spatialization)
Questa sezione definisce il modo in cui il suono verrà spazializzato nel mondo di gioco.
//Se true, il suono verrà spazializzato da una posizione specifica all'interno del mondo di gioco e si sposterà di conseguenza man mano che l'ascoltatore si muove.
Usalo se vuoi che il suono sembri provenire da un luogo particolare nel mondo di gioco (ad esempio, una sorgente sonora spot).
Se false, il suono non sarà spazializzato e diventerà effettivamente un suono "piatto" 2D e quindi non verrà applicato alcun panning,
indipendentemente dalla posizione della sorgente del suono rispetto all'ascoltatore.
La natura esatta di questo processo dipenderà dalla configurazione degli altoparlanti e dal formato dei suoni. Ad esempio, se si utilizza la riproduzione stereo,
un suono mono verrà mixato in stereo, mentre i suoni quad, 5.1 e 7.1 verranno remixati in stereo.
È possibile impostare il metodo utilizzato per l'upmix dei suoni mono in Impostazioni progetto > Motore > Qualità audio > > Metodo di upmix del canale mono.

Usalo se non vuoi che il suono provenga da un luogo particolare nel mondo di gioco (come un loop di area o il tono della stanza).//
- Spatialization Method
1. Panning Questa è l'impostazione predefinita e utilizza il metodo di panoramica standard di Unreal Engine per calcolare il posizionamento spaziale.
Project Settings > Engine > Audio > Quality > Panning Method
E' possibile definire come impostazione di progetto globale se utilizzare il panning di potenza lineare o uguale.
2. Binaural = Questo utilizza qualsiasi plug-in binaurale che hai abilitato per gestire il posizionamento spaziale del tuo suono.
Project Settings > Platforms > [YourPlatform] > Audio > Spatialization Plugin
3. Non-Spatialized Radius = La proprietà definisce la soglia di distanza al di sotto della quale il suono inizierà a passare dall'essere spazializzato a non spazializzato (diventando un suono 2D).
Dentro la sfera non lo è, fuori lo è
- 3D Stereo Spread = Questa proprietà definisce la distanza tra i canali sinistro e destro di un suono stereo quando viene spazializzato nel mondo di gioco.
Ciò è utile per creare un maggiore senso di ampiezza e dimensione quando si utilizzano suoni stereo
- Normalize 3D Stereo Sounds = Viene utilizzata per abilitare o disabilitare una riduzione del guadagno di 6 dB applicata ai suoni stereo che vengono spazializzati nel mondo.
Ciò è utile se il ritaglio dell'audio è dovuto alla somma dei canali quando l'estensione stereo si riduce a 0,0,
come risultato della proprietà Diffusione stereo 3D di cui sopra, o tramite il panning relativo alla distanza.



3. Attenuation (Air Absorption)
Frequenze più alte decadono più velocemente delle frequenze più basse rispetto alla distanza.
Oltre a un filtro passa-basso per modellare l'assorbimento d'aria,
è disponibile anche un filtro passa-alto che può essere utilizzato per ridurre la dimensione percepita del suono rispetto alla distanza.

- Enable Air Absorption = Abilita o disabilita il modello di assorbimento d'aria.
Se true, il suono verrà elaborato utilizzando un filtro passa-basso il cui punto di taglio varia in relazione alla distanza dell'ascoltatore dalla sorgente del suono
- Min Distance Range = La distanza minima dalla sorgente sonora alla quale deve essere applicato il filtro.
- Min Distance Range = Definisce la distanza massima dalla sorgente sonora alla quale deve essere applicato il filtro.
- Low Pass Cutoff Frequency Min = Definisce la frequenza di taglio del filtro passa-basso quando si trova nell'intervallo di distanza minima (come definito in precedenza)
- Low Pass Cutoff Frequency Max = Definisce la frequenza di taglio del filtro passa-basso quando si trova nell'intervallo di distanza massima (come definito in precedenza).
- Enable Log Frequency Scaling = Abilita la scala logaritmica
- Absorption Method = Questa proprietà viene utilizzata per determinare se si desidera utilizzare la funzione di ridimensionamento incorporata (logaritmica o lineare dall'alto)
. Linear =
. Custom =
- High Pass Cutoff Frequency Min = Questa proprietà definisce la frequenza di taglio del filtro passa-alto quando si trova nell'intervallo di distanza minima (come definito in precedenza).
- High Pass Cutoff Frequency Max = Questa proprietà definisce la frequenza di taglio del filtro passa-alto quando si trova nell'intervallo di distanza massima (come definito in precedenza).



4. Attenuation (Focus)
"Comportamento di messa a fuoco" dei suoni.
- Enable Listener Focus = Questa proprietà abilita o disabilita il sistema di messa a fuoco del listener.
- Focus Azimuth = Questa proprietà determina l'angolo di azimut (rotazione orizzontale) entro il quale un suono deve trovarsi per essere classificato come a fuoco, ed è espressa in gradi. Un angolo di 0 gradi si troverebbe direttamente davanti al vettore in avanti dell'ascoltatore.

- Non-Focus Azimuth = Determina l'angolo di azimut (rotazione orizzontale) che un suono deve superare per essere classificato come fuori fuoco, ed è espressa in gradi.
- Focus Distance Scale = Definisce il valore di scala che verrà applicato alla distanza calcolata del suono dall'ascoltatore e quindi il volume attenuato del suono.
Più vicino
- Non-Focus Distance Scale = Questa proprietà definisce il valore di scala che verrà applicato alla distanza calcolata del suono dall'ascoltatore e quindi il volume attenuato del suono.
Più lontano
- Focus Priority Scale = Questa proprietà definisce il valore di scala che verrà applicato all'impostazione di priorità del suono.
- Non-Focus Priority Scale = Questa proprietà definisce il valore di scala che verrà applicato all'impostazione di priorità del suono
- Focus Volume Attenuation = Questa proprietà definisce il valore di scala che verrà applicato al volume del suono quando viene classificato come attivo.
- Non-Focus Volume Attenuation = Questa proprietà definisce il valore di scala che verrà applicato al volume del suono quando viene classificato come sfocato.
- Enable Focus Interpolation = Consente di abilitare o disabilitare il sistema di interpolazione della messa a fuoco:
può essere utilizzato per passare in modo fluido da un'impostazione all'altra quando i suoni passano da un suono all'altro o da uno sfocato.
- Focus Attack Interp Speed = Definisce la velocità di interpolazione utilizzata quando si passa da un passaggio da fuori fuoco a fuoco. Valori più piccoli si traducono in tempi di transizione più rapidi.
- Focus Release Interp Speed = Questa proprietà definisce la velocità di interpolazione utilizzata quando si passa dalla messa a fuoco alla messa a fuoco.

4. Attenuation (Reverb)
Questa sezione definisce la relazione tra la distanza del suono dall'ascoltatore e la quantità di riverbero applicata al suono.
Ciò presuppone che tu abbia definito default reverb setting in World Settings o che il suono si trovi all'interno di un volume audio con un effetto di riverbero applicato e
che le impostazioni della classe sonora del suono consentano l'applicazione del riverbero.
È possibile controllare il metodo utilizzato per determinare il livello di mandata del riverbero, la quantità di suono da inviare all'effetto di riverbero (Reverb Min/Max Send Level)
e il modo in cui questo potrebbe variare in base alla distanza del suono dall'ascoltatore (Reverb Min/Max Send Distance).
- Enable Reverb Send = Questa proprietà abilita o disabilita il sistema di invio del riverbero.
- Reverb Send Method = Questa proprietà definisce il metodo utilizzato per determinare la quantità di suono inviata all'effetto di riverbero.(Linear, Custom, Manual)
. Manual
La quantità di suono inviata al riverbero sarà definita da un singolo valore costante: questo non cambia rispetto alla distanza,
e quindi è probabilmente più utile per i suoni "in stile ambiente" non spazializzati piuttosto che per i suoni 3D con sorgente puntiforme.
- Reverb Min Send Level = Disponibile solo quando si utilizza il metodo di invio lineare e determina la quantità di riverbero da applicare al suono quando si trova la distanza definita nella proprietà Distanza minima.
In genere si utilizzano valori più bassi in modo che sembri esserci un suono più diretto rispetto a

quello riverberante quando si è vicini alla sorgente del suono.
- Reverb Max Send Level = Questa proprietà è disponibile solo quando si utilizza il metodo di invio lineare e determina la quantità di riverbero da applicare al suono quando si trova alla distanza definita nella proprietà Distanza massima.
In genere si usano valori più alti in questo caso in modo che sembri esserci più riverbero rispetto al suono diretto quando si è lontani dalla sorgente del suono.
- Reverb Min Distance = Questa proprietà definisce la distanza dalla sorgente del suono che corrisponderà al livello di mandata minimo (quando si utilizza il metodo di invio lineare) o al valore più basso della curva personalizzata.
- Reverb Max Distance = Questa proprietà definisce la distanza dalla sorgente del suono che corrisponderà al livello di mandata massimo (quando si utilizza il metodo di invio lineare) o al valore più alto della curva personalizzata.

5. Attenuation (Occlusion)
Questa sezione definisce come si comporterà il sistema di occlusione incorporato al fine di simulare l'effetto del suono che
viene ridotto nel contenuto e/o nell'ampiezza delle alte frequenze come risultato della presenza di un oggetto tra la sorgente e l'ascoltatore.
- Enable Occlusion = Questa proprietà abilita o disabilita il sistema di occlusione incorporato.
- Occlusion Trace Channel = Questa proprietà determina quale canale di traccia verrà utilizzato per determinare se esiste un'ostruzione tra l'origine e il listener
l'impostazione predefinita qui è Visibilità, che probabilmente funzionerebbe bene per la maggior parte degli scopi - essenzialmente, se l'oggetto ha la collisione impostata nelle sue proprietà.
- Occlusion Low Pass Filter Frequency = Definisce la frequenza di taglio del filtro passa-basso che verrà applicato se il sistema rileva un'ostruzione tra la sorgente sonora e l'ascoltatore.
- Occlusion Volume Attenuation = Questa proprietà definisce il modo in cui il valore di scala si applica al volume del suono se il sistema rileva un'ostruzione tra la sorgente sonora e l'ascoltatore.
- Occlusion Interpolation Time = Questa proprietà definisce il tempo di transizione per l'effetto di occlusione quando un suono si alterna tra l'essere ostruito e non ostruito.
Più bassi sono i valori, più il sistema è in grado di essere reattivo.
- Use Complex Collision for Occlusion = Questa proprietà consente di abilitare e disabilitare l'utilizzo di modelli di collisione complessi per il sistema di ostruzione.

6. Attenuation (Priority)
Questa sezione determina se l'audio può essere riprodotto o rimanere attivo se viene raggiunto il limite di canali.
È possibile scegliere il metodo di attenuazione della priorità e definire la distanza minima e massima per attenuare la priorità.

7. Attenuation (Submix)
Questa sezione consente di inviare sorgenti sonore a un submix in modo conveniente.
- Enable Submix Send
- Submix Send Settings = Questa proprietà definisce un insieme di impostazioni di invio submix da utilizzare quando si invia l'audio ai submix in funzione della distanza.
- Submix = Submix a cui inviare l'audio in base alla distanza.
- Submix Send Method = Questa proprietà definisce il metodo utilizzato per gli invii di submix.
- Submix Min Send Level = La quantità di suono inviata al submix quando il suono si trova a una

distanza uguale al valore specificato in Submix Min Send Distance.
- Submix Max Send Level = La quantità di suono inviata al Submix quando il suono si trova a una distanza uguale al valore specificato in Distanza massima di invio del submix.
- Submix Min Send Distance = Questa proprietà definisce la distanza minima da inviare al Submix.
- Submix Max Send Distance = Questa proprietà definisce la distanza massima da inviare al Submix.
- Manual Submix Send Level = Questa proprietà definisce il livello di invio manuale del Submix. Non cambia in funzione della distanza.
- Custom Submix Send Curve = Questa proprietà consente di definire una curva personalizzata da utilizzare per il livello di mandata basato sulla distanza.

8. Attenuation (Source Data Override)

9. Attenuation Plugin Settings
uesta sezione consente di definire tutte le impostazioni rese disponibili dagli sviluppatori di qualsiasi plug-in di terze parti che si sceglie di utilizzare per
la spazializzazione, l'occlusione, il riverbero o l'override dei dati di origine.

//

.Open Sound Control (OSC)
E' un protocollo aperto comunemente utilizzato in molte aree dell'industria audio.
Viene utilizzato principalmente per collegare in rete dati audio generici tra client, sebbene possa essere utilizzato anche per dati non audio.
Il plug-in OSC deve essere abilitato per utilizzare questa funzione.
Vai su Edit > Plugins > Input, quindi seleziona l'opzione OSC (Open Sound Control) per abilitare.

//

.Native Soundfield Ambisonics Rendering
Soundfield, è una rappresentazione audio di un'area spaziale, reale o virtuale.
Il downmixing è il processo di prendere una sorgente audio e renderizzarla come una serie di uscite di altoparlanti.
L'ambisonico è una rappresentazione specifica del campo sonoro che proietta l'audio posizionale su una sfera utilizzando armoniche sferiche.
Per importare un file ambisonico nella Libreria, è necessario terminare il nome della risorsa con: .wav

_fuma.wav Se il file ambisonics utilizza l'ordinamento dei canali FuMa.

_ambix.wav Se il file ambisonics utilizza l'ordinamento dei canali ACN.
Dopo che il file ambisonico è stato importato, è possibile trattarlo come qualsiasi altra onda sonora nel motore, con l'importante eccezione che la rotazione è importante.
Quando si riproduce un'onda sonora ambisonica tramite un componente audio nel mondo, il motore ruoterà il letto ambisonico in base alla rotazione tra la rotazione corrente del componente audio in riproduzione e la rotazione corrente del giocatore nel mondo di gioco.
Soundfield Submixes, possono essere utilizzati e collegati ad altri submix nel grafico dei submix,

ma sono in grado di utilizzare qualsiasi rappresentazione di campo sonoro attualmente implementata nel motore o implementata da un plug-in attualmente abilitato.

Se apri il tuo submix soundfield appena creato e guardi il pannello di ispezione, vedrai un menu a discesa per Soundfield Encoding Formatc he include Unreal Ambisonics come opzione.

Dopo aver selezionato Unreal Ambisonics, è possibile inviare qualsiasi sorgente audio a questo submix, indipendentemente dal fatto che si tratti o meno di una sorgente ambisonica,
e verrà codificata o mixata in un singolo flusso ambisonico.
È inoltre possibile inviare sorgenti audio ambisoniche e non ambisoniche al Submix Resonance che verranno renderizzate su un'uscita stereo binaurale.

//

.Submixes Overview

Un submix è un grafico DSP (elaborazione del segnale digitale) che è sempre in esecuzione, anche quando non viene inviato alcun audio.
Componente fondamentale del renderer audio in Unreal Engine, i submix hanno un duplice scopo:

1.Per mixare l'audio generato da singole sorgenti in un singolo buffer di uscita e
2.Ottimizzare l'applicazione degli effetti di elaborazione del segnale digitale (DSP) a più sorgenti sonore contemporaneamente.

//Pensate ai vostri submix come a fiumi che scorrono: quando un suono viene riprodotto, si riversa nel submix.
Quando un submix si connette a un altro, si riversa a cascata nel successivo.
Tutti i suoni che si riversano insieme si mescolano in un flusso di suono e fluiscono verso il basso.
E come un fiume, il flusso è in una sola direzione.//

Ogni endpoint di submix (output hardware) viene alimentato da un singolo grafico.

Un Sound Cue può essere inviato a un submix come qualsiasi altro file . USoundBase

//Nel motore l'EndpointSubmix è descritto come:
"Classe Sound Submix pensata per l'invio di audio a un endpoint esterno, ad esempio il feedback aptico del controller o un dispositivo audio aggiuntivo".//

- Master Reverb Submix:
Le impostazioni del submix Master Reverb si trovano nella Sound Class:

1.Send to Master Reverb Submix: Invia i suoni al Master Reverb.

2.Default 2DReverb Send Amount: Imposta il livello di mandata per i suoni 2D quando vengono inviati al riverbero master.

- Equalizzatore Master Submix

Le impostazioni per il submix Master EQ si trovano in Sound Class, Legacy, in Advanced. L'impostazione Output to Master EQ Submix deve essere abilitata affinché le impostazioni EQ sui mix di classe sonora funzionino.

- Invio di audio a un submix
Esistono diversi modi per inviare sorgenti sonore a un submix:

1.Configura l'invio manualmente in una risorsa sound cue.

2.Utilizzare le impostazioni di Attenuazione.

3.Impostato dinamicamente in un Blueprint.

4.Impostalo tramite Audio Volume.

-Per riprodurre una sorgente audio su un submix o per inviare l'audio generato a un submix, assegnate la nuova risorsa submix alla proprietà Submix sulla sorgente sonora.

Proprietà in Sound Cue:

1.Submix Sends: L'array Submix Sends su una risorsa audio è un array di ulteriori submix a cui le sorgenti sonore possono inviare una parte del loro audio.
Le spedizioni possono essere effettuate manualmente o a distanza;
Ad esempio, più lontano è un suono, minore è l'audio che potrebbe inviare a un submix.
2.Send Level Control Method: come la sorgente deve inviare il suo audio al submix indicato:
2.1.Manual:Il suono invia l' audio direttamente utilizzando il livello si send level.
2.2.Linear:Il suono invia il suo audio al submix utilizzando una mappatura lineare tra il livello di invio minimo e massimo e la distanza di invio minima e massima.
2.3.Custom Curve:Utilizza la curva del livello di mandata personalizzata per inviare l'audio al submix, invece di utilizzare la mappatura lineare.
3.Send Stage:Determina se l'invio della sorgente deve essere eseguito prima o dopo l'applicazione dell'attenuazione della distanza.
4.Sound Submix:A quale submix la sorgente sonora sta inviando il suo audio utilizzando questa voce di invio submix.
5.Send Level:Il livello di invio da utilizzare se il Metodo di controllo del livello di invio è impostato su Manuale
6.Disable Manual Send Clamp:Abilita o disabilita il limite 0–1 quando si utilizza il metodo di controllo del livello di invio manuale.
7.Min Send Level:Il livello di invio minimo da utilizzare quando si utilizza il metodo di controllo del livello di invio non manuale.
8.Max Send Level:Il livello di mandata massimo da utilizzare quando si applica un metodo di controllo lineare.
9.Min Send Distance:La distanza minima da utilizzare quando si applica un metodo di controllo Lineare.
10.Max Send Distance:La distanza massima da utilizzare quando si applica un metodo di

controllo Lineare.
11.Custom Send Level Curve:La curva da utilizzare per mappare il livello di mandata e la distanza minimo e massimo.

Invio ai submix utilizzando l'attenuazione del suono:

Le mandate submix possono anche essere impostate utilizzando le impostazioni di Attenuazione. Si tratta di un modo pratico per impostare gli invii di sottomix su un gran numero di risorse da un'unica posizione.
Dal momento che le mandate submix possono inviare l'audio all'ascoltatore in base alla distanza, ha senso usarlo quando è definita anche l'attenuazione della distanza.

Invia dinamicamente ai submix in Blueprint:

Per impostare un Submix Send avrai bisogno di:

1.Target: il componente audio da utilizzare per inviare l'audio al submix.

2.Submix: il submix a cui verrà inviato l'audio.

3.Send Level: la quantità di audio da inviare, come % del volume totale.

Gli audio volume supporta l'invio di audio ai submix in modo simile a come vengono impostate le mandate direttamente sulle sorgenti sonore.
La differenza principale è che la mandata submix si basa sulla sua posizione rispetto alla geometria del volume audio.

Le sorgenti sonore invieranno il loro audio a un determinato array di invio submix in base al fatto che lo stato della posizione del listener mostri l'ascoltatore all'interno o all'esterno del volume.

Audio Volume supporta anche l'override submix, a seconda dello stato della posizione dell'ascoltatore.
Il submix viene sovrascritto da una catena di effetti specificata, a seconda che l'ascoltatore si trovi all'interno o all'esterno del volume.

Proprietà del submix (MASTER)
Mute when Backgrounded: Consente a un submix di disattivarsi automaticamente applicando uno scaler di volume di 0,0 al suo output quando l'applicazione è in background.
Questa funzionalità consente a un gioco di continuare a riprodurre l'audio in background per impostazione predefinita per alcuni audio ma non per tutti.
Submix Effect Chain: Si tratta di una serie di effetti submix attraverso i quali verrà alimentato l'audio submix misto
Ambisonics Plugin Settings: Questa proprietà consente ai plugin di consentire facoltativamente a un submix di codificare le sorgenti audio inviate come campo sonoro ambisonico.
Queste impostazioni definiscono le proprietà di codifica del campo sonoro.
Parent Submix:Il genitore del submix selezionato. Il submix genitore riceve l'output renderizzato come input.

Child Submixes:Una serie di submix secondari che alimentano il proprio audio come input nel submix selezionato.
Envelope Follower:Un algoritmo DSP che emette l'ampiezza di un segnale audio che è stato attenuato nel tempo.
I valori del tempo di attacco definiscono la rapidità con cui l'algoritmo reagisce all'aumento dell'ampiezza (attacco) e alla diminuzione dell'ampiezza (rilascio).
Submix Level:Controlla i livelli di volume generali per il submix. I valori in questa categoria possono essere facoltativamente
impostati per visualizzare il guadagno di volume lineare (ad esempio da 0,0 a 1,0) o decibel (da -120 dB a 0 dB).
Output Volume:Controlla il volume di uscita complessivo per l'intero submix sia nei canali dry che in quelli wet.
Wet Level:Il volume di uscita dell'audio inviato attraverso la catena degli effetti submix.
Dry Level:Questo è il volume di uscita dell'audio che non è stato inviato attraverso la catena degli effetti submix.
Audio Link Settings:Un oggetto facoltativo per le impostazioni del collegamento audio
Auto Disable:Ciò abiliterà o disabiliterà automaticamente un submix sia che sia silenzioso o udibile.
Auto Disable Time:Questo è il tempo minimo di attesa prima di disabilitare un submix.

Sono disponibili numerosi effetti submix utilizzando il plug-in Synthesis e DSP Effects.

Blueprint APIs and Submixes

I submix possono essere molto utili nei Blueprint, è possibile fare riferimento a un submix in un blueprint creando un riferimento
Sound Submix Variable.

Submix override settings:
1.Sound Submix: Il submix da sovrascrivere con la catena di preimpostazioni dell'effetto submix.

2.Submix Effect Preset Chain: Una serie di preset di effetti submix.

3.Fade Time Sec: Il tempo di dissolvenza incrociata della catena di effetti submix.
Questo svanirà dalla catena di effetti corrente alla nuova sostituzione della catena di effetti.

//

.Audio Volume Actors

1. Ambient Zones(Audio Volume), sono una soluzione audio ambientale interna/esterna economica. I vantaggi delle zone ambientali includono:

Rende buona la transizione dalle aree interne a quelle esterne.
Riduce la manodopera per i progettisti audio, poiché sarà necessario posizionare e iterare meno attori sonori su spazi interni/esterni.
Riduce il sovraccarico dei sistemi di gioco, grazie alla riduzione degli attori sonori necessari per definire interni ed esterni.

Proprietà
Exterior Volume = Il volume finale dei suoni esterni quando il giocatore si trova all'interno di questa zona ambientale.
Exterior Time = Il tempo da sfumare nel nuovo volume esterno in pochi secondi.
Exterior LPF = Moltiplicatore del filtro passa basso applicato ai suoni esterni quando si è all'interno
Exterior LPFTime = Tempo per passare al nuovo livello del filtro passa basso in pochi secondi.
Interior Volume = Il volume finale dei suoni interni quando si è fuori da questa zona ambientale.
Interior Time = Il tempo da sfumare nel nuovo volume interno in pochi secondi.
Interior LPF = Moltiplicatore del filtro passa basso applicato ai suoni interni quando si è all'esterno (0,1 per applicare il LPF massimo).
Interior LPFTime = Tempo per passare al nuovo livello del filtro passa basso in pochi secondi.

Nelle SOUND CLASS, applica in Routing > Apply Ambient Volumes, per creare una classe di AMBIENT ZONES.
Assegnate la Sound class sia ai suoni interni che a quelli esterni dal pannello Dettagli di ciascun suono.

-Settings degli Audio Volumes
Con Audio Volume Actor, è possibile definire le aree all'interno di una mappa Blueprint che possono essere utilizzate per elaborare i suoni,
quindi utilizzare queste impostazioni per applicare effetti di riverbero,
impostare i volumi, definire le zone interessate, emulare l'occlusione su un suono e definire la forma del volume del suono.
1.Transform
2.Audio Volume = Le impostazioni del volume audio definiscono quali impostazioni del volume vengono utilizzate quando sono presenti volumi sovrapposti.
-Priority:Determina la priorità del volume audio, utilizzata per garantire che in caso di volumi sovrapposti o nidificati venga applicato il volume corretto.
Maggiore è il numero utilizzato, maggiore è la priorità. Quando due volumi sovrapposti hanno la stessa priorità, non viene utilizzata alcuna priorità.
-Enabled:Attiva o disattiva il volume dell'audio.
//Se l'ascoltatore si trova nello spazio più piccolo, si trova all'interno di entrambi i volumi audio, quindi è necessario definire come il motore audio sceglie quale volume audio utilizzare.
Quando i volumi audio si sovrappongono, il motore audio utilizza quello con la priorità più alta.//
3.Reverb = È possibile utilizzare le impostazioni del riverbero per attivare o disattivare il volume dell'audio,
selezionare un preset di riverbero da utilizzare quando attivato e definire il volume del segnale di riverbero e il tempo di dissolvenza tra le impostazioni del riverbero.
-Apply Reverb = Abilita o disabilita l'effetto di riverbero assegnato di seguito.
-Reverb Effect = Determina la risorsa effetto riverbero che verrà utilizzata
-Reverb Plugin Effect = Determina l'effetto Reverb Plugin che verrà utilizzato.
-Volume = Imposta il livello del volume per l'effetto riverbero. L'impostazione predefinita è 0,5.
-Fade Time = Il tempo (in secondi) necessario per eseguire l'interpolazione dalle impostazioni attuali del riverbero a quelle nuove quando si entrano ed escono i volumi audio.
4.Brush Settings (definiscono la forma complessiva del volume audio)

-Brush Shapes = Fare clic sul menu a discesa Forma pennello per visualizzare e selezionare da un elenco di forme integrate:

1.Box Brush
1.1 X, Y, Z = Determina la dimensione iniziale del volume audio
1.2 Wall Thickness = Attivo solo quando è abilitata l'opzione Hollow e determina lo spessore dei muri se il volume dell'audio è impostato su Hollow.
1.3 Hollow = Determina se il volume audio è vuoto.
1.4 Tessellated = Determina se vengono generate facce interne aggiuntive per ciascuna faccia del cubo
1.5 Display Shaded Volume = Visualizza il pennello con un volume ombreggiato quando selezionato
1.6 Shaded Volume Opacity Value = Un valore compreso tra 0 e 1 per impostare l'opacità per il volume ombreggiato

2.Cone (ConeBuilder)
2.1 Z = Determina l'altezza iniziale del cono, dalla base alla punta.
2.2 Cap Z = Attivo solo quando è abilitata l'opzione Hollow. Determina l'altezza del cono interno.
2.3 Outer Radius = Il raggio iniziale del volume audio a forma di cono
2.4 Raggio interno = Attivo solo quando è abilitata l'opzione Hollow. Determina il raggio del cono interno
2.5 Sides = Determina il numero di lati del cono
2.6 Hollow = Determina se il volume audio è vuoto.
2.7 Align to Side = Abilita o disabilita l'allineamento del pennello a una faccia.
2.8 Display Shaded Volume = Visualizza il pennello con un volume ombreggiato quando selezionato
2.9 Shaded Volume Opacity Value = Un valore compreso tra 0 e 1 per impostare l'opacità per il volume ombreggiato

3.Curved Stair (CurvedStairBuilder)
3.1 Inner Radius = Determina il raggio interno iniziale del volume audio della scala curva.
3.2 Step Height = Imposta l'altezza iniziale dei gradini all'interno del volume audio della scala curva.
3.3 Step Width = Determina la larghezza dei gradini all'interno del volume audio della scala curva.
3.4 Angle of Curve = L'angolo della curva totale del volume audio della scala curva.
3.5 Num Steps = Il numero di gradini all'interno del volume audio della scala curva.
3.6 Add to First Step = L'importo in cui il volume dovrebbe scendere al di sotto del livello del primo passaggio.
3.7 Counter Clockwise = Determina se la scala curva in senso orario o antiorario.
3.8 Display Shaded Volume = Visualizza il pennello con un volume ombreggiato quando selezionato.
3.9 Shaded Volume Opacity Value = Un valore compreso tra 0 e 1 per impostare l'opacità per il volume ombreggiato.

4.Cylinder (CylinderBuilder)
4.1 Z = Determina l'altezza iniziale del cilindro.
4.2 Outer Radius = Il raggio iniziale del volume audio a forma di cilindro,

che viene quindi ridimensionato in base ai valori X e Y della proprietà Scala nelle impostazioni di Trasformazione.
4.3 Inner Radius = Attivo solo quando è abilitata l'opzione Hollow. Determina il raggio del cilindro interno.
4.4 Sides = Determina il numero di lati del cono, che a sua volta influisce sull'avvicinamento della base a un cerchio.
4.5 Hollow = Determina se il volume audio è vuoto.
4.6 Align to Side = Abilita o disabilita l'allineamento del pennello a una faccia.
4.7 Display Shaded Volume = Visualizza il pennello con un volume ombreggiato quando selezionato.
4.8 Shaded Volume Opacity Value = Un valore compreso tra 0 e 1 per impostare l'opacità per il volume ombreggiato.

5.Linear Stair
5.1 Step Length = Determina la lunghezza iniziale (profondità) dei gradini all'interno del volume audio della scala lineare.
5.2 Step Height = Determina l'altezza iniziale dei gradini all'interno del volume audio della scala lineare.
5.3 Step Width = La larghezza dei gradini all'interno del volume audio della scala curva.
5.4 Num Steps = Il numero di gradini all'interno del volume audio della scala curva.
5.5 Add to First Step = L'importo in cui il volume dovrebbe scendere al di sotto del livello del primo passaggio.
5.6 Display Shaded Volume = Visualizza il pennello con un volume ombreggiato quando selezionato.
5.7 Shaded Volume Opacity Value = Un valore compreso tra 0 e 1 per impostare l'opacità per il volume ombreggiato.

6.Sphere
6.1 Radius = Determina il raggio iniziale della sfera.
6.2 Tessellation = Imposta il numero di iterazioni che il volume audio utilizzerà per calcolare la forma della sfera.
6.3 Display Shaded Volume = Visualizza il pennello con un volume ombreggiato quando selezionato.
6.4 Shaded Volume Opacity Value = Un valore compreso tra 0 e 1 per impostare l'opacità per il volume ombreggiato.

Gli Audio Volume differiscono daglia Audio Gameplay Volume, perchè negli Audio Volume puoi piazzarci i submix.
Nei Audio Gameplay Volume, puoi utilizzarli in primis nei BP come PROXY VOLUME, e piazzarci a loro volta degli effetti.
Nei Audio Gameplay Volume non puoi inserire i Submix.

2. Ambient Sound Actor

Quando una risorsa Onda sonora o Segnale sonoro viene posizionata in un livello, viene creato un attore sonoro ambientale con tale risorsa sonora.

Se l'attore sonoro ambientale è impostato su Auto Activate inizierà immediatamente a suonare

una volta creato.
L'asset sonoro a cui punta l'Ambient Sound Actor verrà riprodotto una sola volta per trigger.

Un Ambient Sound Actor non ha alcuna funzione senza una risorsa sonora associata. È possibile assegnare una risorsa audio dal pannello Details.

Proprietà di un Ambient Sound Actor
1. Sound = Punta a una risorsa Sound Cue o Wave
2. Volume Multiplier = Impostare il volume generale del suono.
3. Pitch Multiplier = Impostare il Pitch complessivo del suono.
4. Is UI Sound = Se la risorsa audio viene riprodotta o meno quando il gioco è in pausa.
5. Low Pass Filter Frequency = Abilita per impostare la frequenza (in Hz) del filtro passa basso.
6. Priority = Abilita per sovrascrivere la priorità del suono selezionato con il valore fornito.
7. Play Multiple Instances = Se vero, il componente audio riproduce più istanze audio contemporaneamente.
8. Source Effect Chain = Determina una catena di effetti sorgente da applicare ai suoni riprodotti sul componente audio.
9. Attenuation
9.1 Allow Spatialization = Abilita o disabilita la spazializzazione del componente audio.
9.2 Override Attenuation = Utilizza le impostazioni di Attenuazione della risorsa o le impostazioni di Sostituzione dell'attenuazione.
9.3 Attenuation Settings = Punta e utilizza le impostazioni assegnate all'interno di una risorsa SoundAttenuation.
10. Modulation Properties = Le impostazioni di Modulazione definiscono quali impostazioni utilizzare per la modulazione di Volume.
10.1 Disable = Disabilita il routing della modulazione.
10.2 Inherit = Il componente audio eredita le impostazioni di modulazione dal suono principale.
10.3 Override = Ignora le impostazioni ereditate e consente di impostare il valore di base, abilitare o disabilitare la modulazione e impostare la sorgente di modulazione per ciascun parametro.

.Convolution Reverb
Il riverbero a convoluzione è una tecnologia entusiasmante che i sound designer possono utilizzare per catturare le risposte all'impulso (IR) di qualsiasi luogo fisico
(come grotte, chiese, uffici o corridoi) ed elaborare l'audio in tempo reale come se venisse riprodotto in quella posizione.

Come usare il riverbero a convoluzione

Abilita il plug-in Sintesi ed effetti DSP Edit > Plugins > Audio > Synthesis and DSP Effects
--Con il plug-in abilitato, fare clic con il pulsante destro del mouse su qualsiasi onda sonora importata nella Libreria e selezionare Create Impulse Response.

Questo crea una risposta all'impulso dall'onda sonora selezionata.
In questo modo, qualsiasi onda sonora potrebbe diventare l'IR per l'algoritmo del riverbero a convoluzione (puoi convolvere con qualsiasi suono).

Per utilizzare l'effetto di riverbero a convoluzione, create un predefinito per effetti submix

facendo clic con il pulsante destro del mouse nella Libreria e selezionando
Audio > Effects > Submix Effect Preset.
--Seleziona SubmixEffectConvolutionReverbPreset

Bypass = Se vero, l'audio in ingresso viene indirizzato direttamente all'audio in uscita con l'applicazione di qualsiasi effetto.
Mix Input Channel Format to Impulse Response Format = Se vero, l'audio in ingresso del submix viene sottoposto a downmix per corrispondere al formato del canale audio della risorsa di risposta all'impulso
Mix Reverb Output to Output Channel Format = Se vero, l'audio riverberato viene sottoposto a upmix o downmix per corrispondere al formato audio di output del submix
Surround Rear Channel Bleed Db = Quanti decibel dell'uscita dell'effetto di convoluzione inviare ai canali posteriori in un ambiente audio surround.
Invert Rear Channel Bleed Phase = Indica se invertire o meno la fase dell'audio inviato ai canali posteriori in ambienti audio surround.
Surround Rear Channel Flip = Indica se invertire o meno i canali sinistro e destro dell'effetto prima di inviarlo ai canali posteriori.

Advaced
Block Size = Imposta la dimensione del blocco di elaborazione interna, che potrebbe avere un effetto sulle prestazioni di runtime
Enable Hardware Acceleration = Abilita l'accelerazione hardware se esiste sulla piattaforma di destinazione.

Per utilizzarlo su un effetto submix, aggiungetelo a qualsiasi submix come effetto submix in una catena di effetti submix.

////////////

Nodi Quartz Clock

1.1 Create New Clock = Crea un nuovo orologio (o restituisce la Handle se l'orologio esiste già) L'obiettivo è il quarzo

1.2 Delete Clock by Handle = elimina un Clock esistente in base alla sua Handle. L'obiettivo è al quarzo

1.3 Delete Clock by Name = elimina un Clock esistente in base al suo nome. L'obiettivo è al quarzo

1.4 Does Clock Exist = restituisce vero se l'orologio esiste. L'obiettivo è al quarz

1.5 Get Current Clock Timestamp = Recupera un timestamp per l'orologio Target is Quartz

1.6 Get Current Timestamp = Recupera un timestamp per l'orologio. L'obiettivo è la maniglia dell'orologio al quarzo

1.7 Get Duration Of Quantization Type in Seconds = Restituisce la durata in secondi del tipo di quantizzazione specificato. Il target è Quarzo

1.8 Get Estimated Clock Run Time = Restituisce il tempo, in secondi, durante il quale l'orologio è stato in funzione.

Attenzione: a causa della latenza, questo non sarà perfettamente accurato. Target is Quartz

1.9 Get Handle for Clock = Get Handle per l'orologio esistente L'obiettivo è il quarzo

1.9.1 Is Clock Running = restituisce vero se l'orologio è in funzione. L'obiettivo è al quarzo

1.9.2 Is Quartz Enabled = Is Quartz Enabled Target is Quartz

2. Quartz Metronome = Il metronomo è l'oggetto del thread di rendering audio che tiene traccia del passare del tempo e decide quando devono essere eseguiti i comandi successivi
(da informazioni fornite dall'utente come BPM (battiti al minuto) e un'indicazione del tempo.

Nodi Quantization

2.1 Get Beat Progress Percent = Restituisce l'avanzamento corrente fino alla successiva occorrenza della durata musicale fornita come valore float da 0 (battuta precedente) a 1 (battuta successiva).

2.2 Get Beats Per Minute = Ottieni battiti al minuto L'obiettivo è Quartz Clock Handle

2.3 Get Milliseconds Per Tick = L'obiettivo dei getter del metronomo è la Quartz Clock Handle

2.4 Get Seconds Per Tick = Ottieni secondi per tick L'obiettivo è la maniglia dell'orologio al quarzo

2.6 Get Thirty Second Notes Per Minute = Ottieni trenta secondi di note al minuto. L'obiettivo è la maniglia dell'orologio al quarzo
2.7 Get Ticks Per Second = L'obiettivo Ottieni tick al secondo è la maniglia dell'orologio al quarzo
2.9 Set Beats Per Minute = L'impostazione del target dei battiti al minuto è la maniglia dell'orologio al quarzo
2.9.1 Set Milliseconds Per Tick
2.9.2 Set Seconds Per Tick
2.9.3 Set Thirty Second Notes Per Minute
2.9.4 Set Ticks Per Second

3. Quartz Clock Handle = La Clock Handle è il gameplay side proxy di un orologio attivo. //Il proxy è un intermediario//
Questo viene acquisito tramite il Quartz subsystem e utilizzato per controllare il Clock in esecuzione nel Mixer audio.

Nodi Quartz Clock Handle
3.1 Get Duration Of Quantization Type in Seconds = Restituisce la durata in secondi del tipo di Quantization target è la Quartz Clock Handle
3.2 Is Clock Running = L'obiettivo corrente dell'orologio è la Quartz Clock Handle//Importante
3.3 Reset Transport Quantized = Ripristina il target quantizzato di trasporto è Quartz Clock Handle
3.4 Resume Clock = Riprendi l'obiettivo dell'orologio è la Quartz Clock Handle
3.5 Start Clock = Manipolazione dell'orologio L'obiettivo è la Quartz Clock Handle//Importante
3.6 Start Other Clock = Manipolazione dell'altro orologio L'obiettivo è la Quartz Clock Handle
3.7 Subscribe to All Quantization Events = Iscriviti a tutti gli eventi di quantizzazione L'obiettivo è la Quartz Clock Handle//Importante
3.8 Subscribe to Quantization Event = Target è metronomo è la Quartz Clock Handle//Importante
3.9 Unsubscribe from All Time Divisions = Annulla l'iscrizione a tutte le divisioni temporali L'obiettivo è la Quartz Clock Handle
3.9.1 Unsubscribe from Time Division

4. Quartz Subsystem = Mentre la Quartz Clock Handle fornisce l'accesso alle funzionalità su un Quartz Clock, il Quartz Subsystem fornisce l'accesso a funzionalità che non si riferiscono a un Clock specifico.

Nodi Quartz Subsystem

4.1 Get Audio Render Thread to Game Thread Average Latency = dati di latenza (thread di rendering audio -> thread di gioco) L'obiettivo è Quartz

4.2 Get Audio Render Thread to Game Thread Max Latency = Ottieni il thread di rendering audio nel thread di gioco. L'obiettivo di latenza massima è Quartz

4.3 Get Audio Render Thread to Game Thread Min Latency

4.4 Get Game Thread to Audio Render Thread Average Latency = dati di latenza (thread di gioco -> thread di rendering audio) L'obiettivo è Quartz

4.5 Get Game Thread to Audio Render Thread Max Latency

4.6 Get Game Thread to Audio Render Thread Min Latency

4.7 Get Round Trip Average Latency = dati di latenza (andata e ritorno) L'obiettivo è Quartz

4.8 Get Round Trip Max Latency

4.9 Get Round Trip Min Latency

4.9.1 Pause Clock = L'obiettivo dell'orologio in pausa è Quartz Clock Handle

4.9.2 Stop Clock = Arresta l'obiettivo dell'orologio è Quartz Clock Handle

5. AudioComponent: PlayQuantized()

Si tratta di una nuova funzione aggiunta a Componenti audio che fornisce un modo per riprodurre un suono su un clock specifico, con un limite di quantizzazione specificato.

- Fasi di creazione di Quartz subsystem

Inizia con il nodo Create New Clock su the Quartz Subsystem.

Nel nodo Create New Clock, trascina il settings Time Signature > Utilities > Make QuartzClockSettings.

In Make QuartzClockSettings trascina il pin di Time signature, > Utilities > Make

QuartzTimeSignature

- Play Sounds on a Quantization Boundary

Per riprodurre un suono su un limite di quantizzazione, chiamare un componente audio e fornire un handle di clock e un limite di quantizzazione.

(Play Quantized)

Proprietà

- Quartz Quantization Boundary è il modo in cui dici esattamente al Mixer audio quando vuoi che inizi il suono

- Quantization è il valore temporale da utilizzare per la pianificazione

1. Bar = Determinato automaticamente dall'indicazione del tempo.
2. Beat = L'impostazione predefinita è al denominatore dell'indicazione del tempo, ma può essere sovrascritta.
3. Tick = On Tick è il valore più piccolo, uguale alla nota di trentaduesimo.

- Multiplier = è il numero di valori di tempo di quantizzazione fino a quando il suono non deve essere riprodotto.

- Counting Reference Point = è il punto di riferimento utilizzato per la pianificazione dell'evento.
1. Bar Relative = Dall'inizio della barra corrente.
2. Transport Relative = Dal momento in cui è stato avviato l'orologio o dall'ultimo ripristino del trasporto.
3. Current Time Relative = Dall'ora attuale.

- Play Quantized evento delegato in Switch (Event Type)

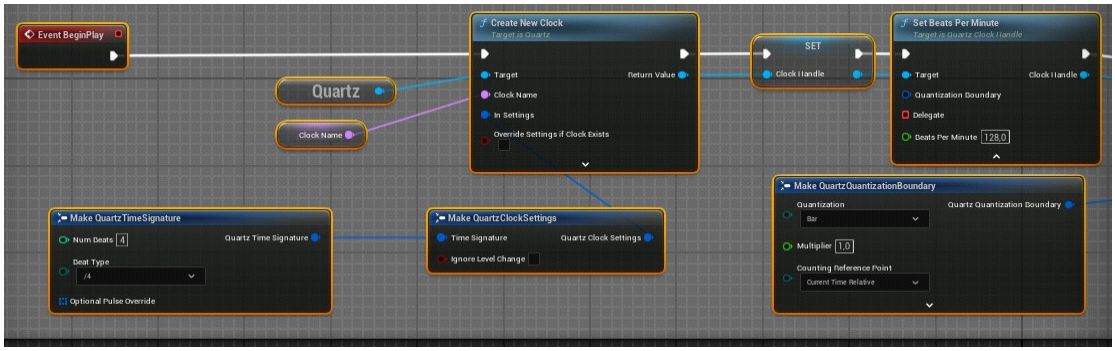
1. Failed To Queue = Il comando non è arrivato al thread di rendering audio (probabilmente la riproduzione quantizzata non è riuscita contemporaneamente).

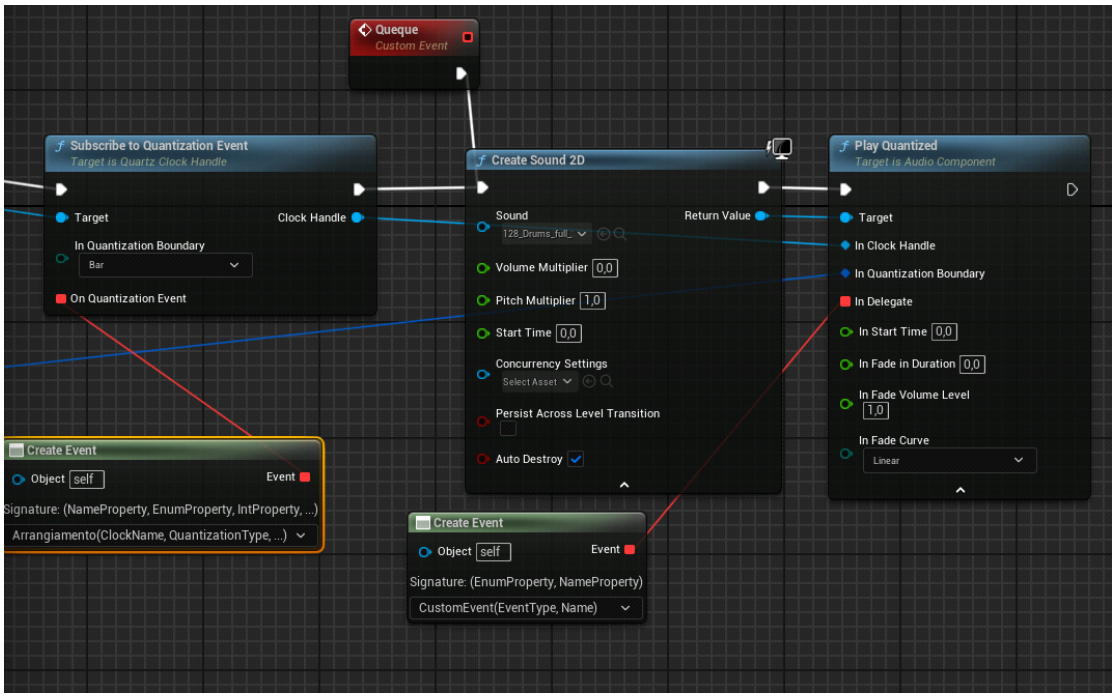
2. Queued = Il comando è stato ricevuto dal thread di rendering audio ed è in attesa di esecuzione (concorrenza superata)

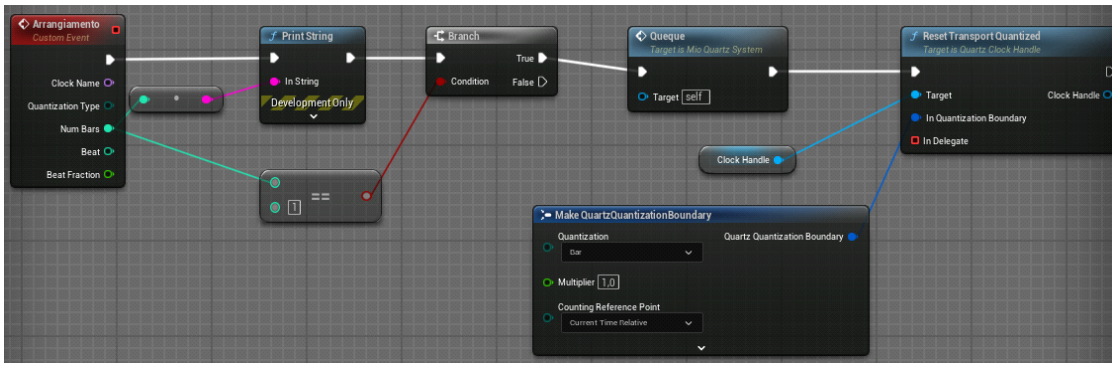
3. Canceled = Il comando è stato annullato nel thread di rendering audio prima di essere eseguito (probabilmente è stato chiamato Stop sul componente audio prima che iniziasse la riproduzione).

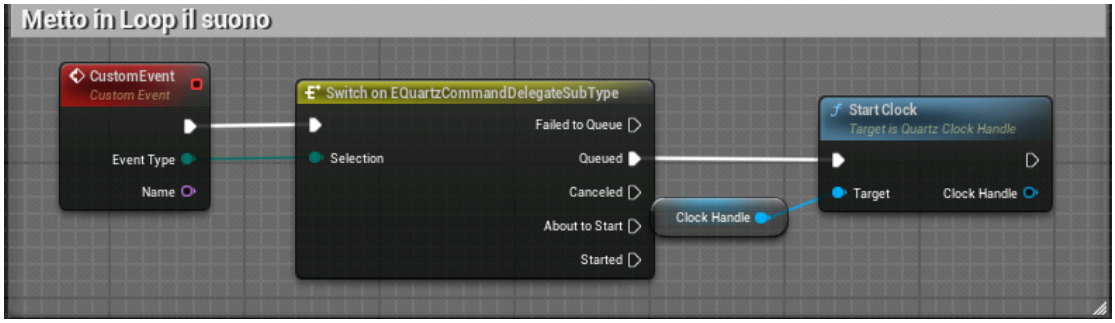
4. About to Start = Il comando sta per essere eseguito

5. Started = Il comando è stato eseguito sul thread di rendering audio.






Arrangiamento
Custom Event
Clock Name
Quantization Type
Num Bars
Beat
Beat Fraction
Print String
In String
Development Only
Branch
True
Condition
False
Queque
Target is Mio Quartz System
Target
self
Reset Transport Quantized
Target is Quartz Clock Handle
Target
Clock Handle
In Quantization Boundary
In Delegate
Clock Handle
Make QuartzQuantizationBoundary
Quantization
Bar
Quartz Quantization Boundary
Multiplier
1.0
Counting Reference Point
Current Time Relative


Metto in Loop il suono
CustomEvent
Custom Event
Event Type
Name
Switch on EQuartzCommandDelegateSubType
Selection
Failed to Queue
Queued
Canceled
About to Start
Started
Clock Handle
Start Clock
Target is Quartz Clock Handle
Target
Clock Handle

////////////

- Unreal audio routing - Audio Wwise e Unreal: utilizza contemporaneamente il sistema audio Unreal e il Wwise SoundEngine.

////////Eventi//////////

1. Pubblicazione di eventi Wwise utilizzando l'attore AkAmbientSound

AkAmbientSound è un attore che contiene un componente Ak Component.
In all, piazza l'evento in AK EVENT

2. Nodi

Pubblicazione di eventi Wwise utilizzando il nodo Avvia suono ambientale
Se non vogliamo pubblicare automaticamente un evento deselezionando l'impostazione Auto Post in Ak Ambient Sound (la componente),
- Possiamo avviare il suono da soli usando il nodo Start Ambient Sound.
AUTO POST, avvia automaticamente il suono in Begin Play.
In modo simile possiamo fermare il suono ambientale utilizzando il nodo Stop Ambient Sound.

Se guardi attentamente la scheda dei dettagli dell'attore, noterai che Ak Ambient Sound contiene solo un AkComponent: AkEvent

- Post Ak Event e Post Associated Ak Event e connettere il riferimento AkComponent al valore di destinazione. (In pratica prende la componente del Game Object)

3. Pubblicazione di eventi Wwise con AkComponent
Innanzitutto, crea un AkComponent selezionando un attore e facendo clic su Aggiungi componente: (Digita semplicemente Ak)
Nella scheda dei dettagli del componente, selezionare l'evento audio Ak desiderato.

Usando il nodo Post Associated Ak Event, pubblicheremo l'evento selezionato in precedenza

4. Pubblicazione di eventi Wwise con il nodo Post Event
Il nodo Post Event è un nodo generale che pubblicherà un evento Wwise e lo collegherà al componente radice dell'attore specificato

5. Post Event at Location Node

Il nodo Pubblica evento nella posizione crea un oggetto di gioco Wwise temporaneo e pubblica un evento Wwise in una posizione specificata.

6. Preparazione per giocare a Wwise Events con C++

7. Ak Component spatial audio Inner radio raggio con spread (interna al raggio) al 100% si sente di più, outer radius è intorno al 50%

Footsteps

Crea cartella switch

In project settings, engine, surface type, crea nuovi surface in base agli switch creati
Crea altrettanti Material e Physics Material (In classe PHYSICS), assegna alle mesh sia i Material che i Physics Material.
In ogni PHYSIC MATERIAL creato assegna la propria SURFACE TYPE create in project settings.
Apri l'animazione di corsa o camminata, in alto a destra clicCa SK_MANNEQUIN. In questa schermata in basso a destra aggiungi una notifica FOOTSTEPS (in ANIMATION).
Nel sequencer aggiungi la notifica.
Crea nodo

https://www.audiokinetic.com/en/library/edge/?source=UE4&id=gs_footstep_sounds.html

---
- Funzioni interessanti

Set Listeners, setta un game object ascoltatore

- AkLateReverbComponent

È possibile aggiungere questo componente a qualsiasi componente per creare una zona di riverbero da quel componente primitivo.

- Post Event At Location

Pubblica un evento Wwise in una posizione specificata. Si tratta di un suono fire-and-forget, creato su un oggetto di gioco Wwise temporaneo senza AkComponent corrispondente.

- AkComponent

AkComponent è derivato da e rappresenta un evento Wwise attivo.USceneComponent

- Spawn Ak Component at Location

Crea un nuovo AkComponent nella posizione specificata. Per impostazione predefinita, il componente viene distrutto automaticamente al termine della riproduzione dell'evento.

Auto Post: controlla se pubblicare l'evento immediatamente dopo la creazione del componente. Il valore predefinito è .false
Auto Destroy: controlla se eliminare il componente quando viene eseguito il primo evento pubblicato su questo componente. Il valore predefinito è .true

---

AkLateReverbComponent

È possibile aggiungere questo componente a qualsiasi componente per creare una zona di riverbero da quel compon

Ak component, aggiungi in spatial audio, reflect, un send reverb che hai creato.
In Ak ambient sound, AUTO POST, per farlo partire automaticamente senza codice

---
Riverbero

Per creare un riverbero SOLO in una stanza, associare a quell attore la componente AK late Reverb, assegnare il riverbero ausiliario

https://www.audiokinetic.com/en/library/edge/?source=UE4&id=features_blueprint.html LINK AGLI SCRIPT PIU IMPORTANTI +

////////////

--Main Output Settings
8. Audio Device ShareSet = Il nome di un dispositivo audio personalizzato da utilizzare.
I dispositivi audio personalizzati vengono definiti nella sezione Audio Device ShareSet di un progetto Wwise.
Lascialo vuoto per l'output normalmente tramite il dispositivo audio predefinito.
9. Device ID = Identificatore specifico del dispositivo quando si utilizzano più dispositivi dello stesso tipo.
Lasciare l'impostazione su 0 (predefinita) se si utilizza un solo dispositivo.
10. Panning Rule = Regola per il panning 3D dei segnali indirizzati a un bus stereo.
Nella modalità "Altoparlanti" viene utilizzata l'angolazione degli altoparlanti anteriori. Nella modalità "Cuffie",
gli angoli degli altoparlanti vengono sostituiti dalla potenza costante tra due microfoni virtuali distanziati di 180 gradi l'uno dall'altro.
11. Channel Config Type = Un codice che completa l'identificazione dei canali da parte di uChannelMask.
12. Channel Mask = Un campo bit, i cui identificatori di canale dipendono da (fino a 20).
13. Number of Channels = Il numero di canali, identificati (dedotto dalla maschera del canale) o anonimi (impostati direttamente).
-Window Spatial Audio Settings
14. Max Sound Propagation Depth = Numero massimo di portali attraverso i quali il suono può propagarsi.
15. Movement Threshold = Quantità che un emettitore o un ascoltatore deve spostare per attivare un ricalcolo delle riflessioni e della diffrazione.
Valori più grandi possono ridurre il carico della CPU a scapito della precisione ridotta.
16. Number Of Primary Rays = Il numero di raggi primari (trasmessi da ciascun ascoltatore) utilizzati nella trasmissione dei raggi stocastici.
17. Reflection Order = Ordine di riflessione massimo compreso tra 1 e 4.
18. Diffraction Order = Ordine di diffrazione massimo: il numero di "curve" in un percorso di diffrazione.
Un ordine di diffrazione elevato si adatta a geometrie più complesse a scapito di un maggiore utilizzo della CPU.
19. Max Emitter Room Aux Sends = Il numero massimo di invii ausiliari definiti dal gioco che possono avere origine da un singolo emettitore.
20. Diffraction on Reflections Order = Numero massimo possibile di punti di diffrazione a ciascuna estremità di un percorso di riflessione.
21. Maximum Path Length = Lunghezza massima di un percorso di riflessione/diffrazione.
22. CPU Limit Percentage = Controlla la percentuale massima di un fotogramma audio che il motore di raytracing può utilizzare.
23. Load Balancing Spread = Il calcolo dei percorsi audio spaziali viene distribuito sui frame LoadBalancingSpread.
24. Enable Geometric Diffraction and Transmission = Abilita il calcolo della diffrazione geometrica e dei percorsi di trasmissione

per tutte le sorgenti che hanno la casella Abilita diffrazione e trasmissione selezionata nella scheda Posizionamento dell'editor delle proprietà Wwise.
25. Calc Emitter Virtual Position = Se impostato su true, Wwise Spatial Audio calcola la posizione apparente o
virtuale degli emettitori che diffrangono attraverso i portali o attorno alla geometria, quindi invia la posizione al motore sonoro.
-Window
--Communication Settings
26. Initialize System Comms = indica se inizializzare o meno il sistema di comunicazione.
27. Pool Size = Size of the communication pool.
28. Discovery Broadcast Port = La porta su cui l'applicazione Wwise Authoring trasmette le richieste "Game Discovery" per rilevare i giochi in esecuzione sulla rete.
29. Command Port = La porta del canale "comando". Impostato su 0 per richiedere una porta dinamica/effimera.s
30. Network Name = Il nome utilizzato per identificare questo gioco all'interno dell'applicazione Wwise Authorings
-Window Advanced Settings
31. Use Head Mounted Display Audio Device = Abilita quando si lavora su progetti per display montati sulla testa con funzionalità audio.
32. Max System Audio Objects = Numero massimo di oggetti audio di sistema da prenotare. Altri processi non saranno in grado di utilizzarli. Il valore predefinito è 128.
33. Enable Multi Core Rendering = Se impostato su true, distribuisce le attività di elaborazione SoundEngine nel Task Graph di Unreal Engine.
Se modifichi questa impostazione, devi riavviare l'editor.
34. Max Num Job Workers = Il numero massimo di lavoratori che Sound Engine può richiedere in qualsiasi momento.s
35. Job Worker Max Execution Time Usec = Tempo massimo assegnato per le attività di elaborazione del Sound Engine in microsecondi.
36. IO Memory Size = Dimensione del pool di memoria per I/O (per flussi automatici).
37. IO Granularity = Granularità della richiesta I/O (byte/richiesta tipici).s
38. Target Auto Stream Buffer Length = Lunghezza del buffer di flusso automatico mirata in millisecondi.
39. Use Stream Cache = Se vero, il dispositivo tenta di riutilizzare i buffer I/O di cui è già stato eseguito lo streaming dal disco.
40. Maximum Pinned Bytes in Cache = Numero massimo di byte che possono essere "bloccati" utilizzando o .AK::SoundEngine::PinEventInStreamCache()AK::IAkStreamMgr::PinFileInCache()
41. Enable Game Sync Preparation = Impostare su true per abilitare l'utilizzo.AK::SoundEngine::PrepareGameSync
42. Continuous Playback Look Ahead = Il numero di quanti successivi in cui i contenitori continui istanziano una nuova voce prima che inizino a suonare i suoni successivi.
43. Monitor Queue Pool Size = Dimensione del pool di code di monitoraggio.
44. Maximum Hardware Timeoput Ms = Tempo (in millisecondi) di attesa affinché i dispositivi

hardware attivino un'interruzione audio.
45. Debug Out Of Range Check Enabled = Impostazione di debug: abilita i controlli per i float fuori intervallo (e NaN) nel codice di elaborazione.
46. Debug Out Of Range Limit = Impostazione di debug: utilizzata solo quando Debug Out Of Range Check Enabled è vero.

-Integration Settings
--Reverb
47. Max Simultaneous Reverb Volumes = Volumi massimi di riverbero simultane = Il numero massimo di volumi di riverbero Ak che possono influenzare un suono.
Impostandolo su zero si disabilitano tutti i volumi di riverbero Ak nel gioco.
Installation
48. Wwise Project Path = La posizione del progetto Wwise per il gioco UE. L'integrazione Wwise richiede questo percorso
per creare le risorse Wwise richieste per il gioco utilizzando la funzione Genera SoundBanks all'interno di Unreal Content Browser (o nel menu Costruisci).
49. Installation = Il percorso della cartella che contiene i metadati del progetto Wwise, in particolare il file.
- Cooking
50. Wwise Staging Directory = La directory in cui verranno copiati i file .bnk e .wem durante la preparazione dei file durante la cottura.
- Obstruction Occlusion
51. Default Collision Channel = Il canale di collisione predefinito utilizzato per l'ostruzione audio e i calcoli dell'occlusione
- Fit to Geometry
52. Default Fit to Geometry Collision Channel = Il valore predefinito per Canale di collisione quando si adattano i portali acustici Ak e
i volumi audio spaziali Ak alla geometria circostante.
- Reverb Assignment
53. Global Decay Absorption = Il valore di assorbimento superficiale globale da utilizzare quando si stima il valore di decadimento ambientale.
54. Default Reverb Aux Bus = Il bus ausiliario predefinito da scegliere per l'assegnazione automatica del riverbero.
55. Reverb Assignment Table = Una tabella che associa i bus ausiliari ai valori di decadimento del riverbero.
--Row Name = Un nome univoco per la riga.
--Decay = Il numero di secondi necessari affinché il riverbero del suono in un ambiente decada di 60 dB.
--Aux Bus = Il bus ausiliario in Wwise Authoring che ha un effetto di riverbero da associare al valore di Decay della stessa riga.
- RTPCs
56. HFDamping Name = Il nome dell'RTPC utilizzato per impostare lo smorzamento delle alte

frequenze prodotto dall'ambiente. Sostituito dal valore HFDamping RTPC, se presente.
57. Decay Estimate Name = Il nome RTPC utilizzato per stimare il tempo necessario affinché il livello di pressione sonora si riduca di 60 dB.
58. Time to First Reflection Name = Il nome RTPC utilizzato per impostare il tempo impiegato dalla prima riflessione per raggiungere l'ascoltatore.s
59. HFDamping RTPC = L'RTPC viene utilizzato per impostare lo smorzamento delle alte frequenze prodotto dall'ambiente.
60. Decay Estimate RTPC = L'RTPC è stato utilizzato per stimare il tempo necessario affinché il livello di pressione sonora si riduca di 60 dB.
61. Time to First Reflection RTPC = L'RTPC imposta il tempo impiegato dalla prima riflessione per raggiungere l'ascoltatore.
-Initialization
62. Audio Input Event = Evento di input associato all'ingresso audio Wwise.
63. Init Bank = L'unico Init Bank per il progetto Wwise, che contiene le informazioni di base necessarie per impostare correttamente il Sound Engine
64. Unreal Audio Routing = Opzioni di instradamento audio che determinano se utilizzare esclusivamente l'audio Unreal o Wwise, i due insieme,
o instradare l'audio Unreal tramite AkAudioMixer o AudioLink.
65. Wwise Sound Engine Enabled = Indica se Wwise Sound Engine è abilitato, a seconda dell'opzione di routing audio selezionata
66. Wwise Audio LInk Enabled = Indica se Wwise AudioLink è abilitato, a seconda dell'opzione di routing audio selezionata.s
67. Ak Audio Mixer Enabled = Indica se Wwise AkAudioMixer è abilitato, a seconda dell'opzione di routing audio selezionata.
--Localization
68. Unreal Culture to Wwise Culture = Mappatura del codice di Unreal Culture nel linguaggio Wwise.
--Asset Creation
69. Default Asset Creation Path = Il percorso in cui creare le risorse trascinate dal Wwise Browser.
--Geometry Surface Properties
70. Verify and Update = Verificare ogni riga della tabella delle proprietà della superficie geometrica di seguito e rimuovere le righe con un materiale fisico non validos
71. Default Acoustic Texture = La texture acustica predefinita viene impostata su una superficie di un attore di volume audio spaziale quando viene utilizzato
Adatta alla geometria e non viene colpita alcuna geometria.
72. Default Transmission Loss = Il valore di perdita di trasmissione predefinito impostato su una superficie di un attore di volume audio spaziale quando viene utilizzato
73. Geometry Surface Properties Table = La tabella che associa le proprietà della superficie geometrica (struttura acustica e perdita di trasmissione) ai materiali fisici.

-User Settings

--Installation
74. Wwise Windows Installation Path = La posizione di Wwise Authoring sul tuo computer di sviluppo Windows.s
75. Wwise Mac Installation Path = La posizione di Wwise Authoring sul tuo computer di sviluppo Mac.
76. Root Output Path User Override = La posizione della cartella che contiene i metadati del progetto Wwise,
come determinato dalla sostituzione dell'utente nelle Impostazioni SoundBank dell'utente Wwise.
--WAAPI
77. WAAPI IP Address = Indirizzo IP utilizzato per connettersi all'API Wwise Authoring (WAAPI).
78. WAAPI Port = Porta utilizzata per connettersi a WAAPI.
79. Auto Connect to WAAPI = Abilita la connessione automatica dell'Unreal Editor a Wwise tramite WAAPI.
80. Auto Sync Selection = Se sincronizzare la selezione tra Wwise Browser e Wwise Project Explorer.
--Error Message Translator
81. XML Translator Timeout = Tempo massimo assegnato per la ricerca degli ID (millisecondi) per il lettore di file XML.
Impostare su 0 per disabilitare.
82. Waapi Translator Timeout = Tempo massimo concesso per la ricerca degli ID (millisecondi) tramite la connessione WAAPI. Impostare su 0 per disabilitare.
--Assets Reload
83. Ask for Wwise Assets Reload = Apre una notifica che l'utente deve accettare prima di ricaricare Wwise Asset
--Viewports
84. Visualize Rooms and Portals = Visualizza stanze e portali nel viewport. L'impostazione "tempo reale" deve essere abilitata nel viewport.
85. Show Reverb Info = Quando abilitato, le informazioni su vengono visualizzate nelle finestre, sopra il genitore UPrimitiveComponent del componente.

-Wwise Simple External Source Manager Settings
86. Media Info Table = Tabella di tutte le informazioni necessarie per caricare correttamente tutti i supporti di origine esterni nel progetto.
87. External Source Default Media = Tabella facoltativa che definisce una voce multimediale predefinita in MediaInfoTable da caricare quando viene caricata un'origine esterna.
88. External Source Staging Directory = Posizione di staging per i media di origine esterna durante la preparazione del progetto.

.Wwise Authoring API (WAAPI)
L'API Wwise Authoring comunica con l'applicazione Wwise Authoring.

.Unreal Engine C++ Projects

-Linking Modules

La Wwise Unreal Integration è suddivisa in più moduli. Per utilizzare la funzionalità di Wwise all'interno di un progetto C++, devi collegare questi moduli all'interno del file di build del tuo progetto (.Build.cs). Puoi disabilitare un modulo rimuovendolo dal file di build.

Nell'esempio seguente, il progetto include i seguenti moduli come pubblici: Core, CoreUOBject, Engine, InputCore, AkAudio e WwiseSoundEngine. AkAudio e WwiseSoundEngine sono moduli Wwise.

```
public class MyModule : ModuleRules
{
   public MyModule(ReadOnlyTargetRules Target) : base(Target)
   {
      PublicDependencyModuleNames.AddRange(new string[] { "Core", "CoreUObject", "Engine",
"InputCore", "AkAudio", "WwiseSoundEngine" });

      // Other settings
   }
}
```

In questo caso sono disponibili le funzionalità del modulo WwiseSoundEngine e del modulo AkAudio. I moduli possono trovarsi in PublicDependencyModulesNames o PrivateDependencyModulesNames.

-Module Overview

AkAudio: Il modulo root per la maggior parte delle funzionalità rivolte all'utente dell'integrazione.

AkAudioMixer: Consente a un progetto di convogliare l'output audio, nonché i submix audio, dai componenti audio Unreal al mixer di un progetto Wwise.

AudiokineticTools: Implementa strumenti e utilità specifici dell'editor come Wwise Browser.

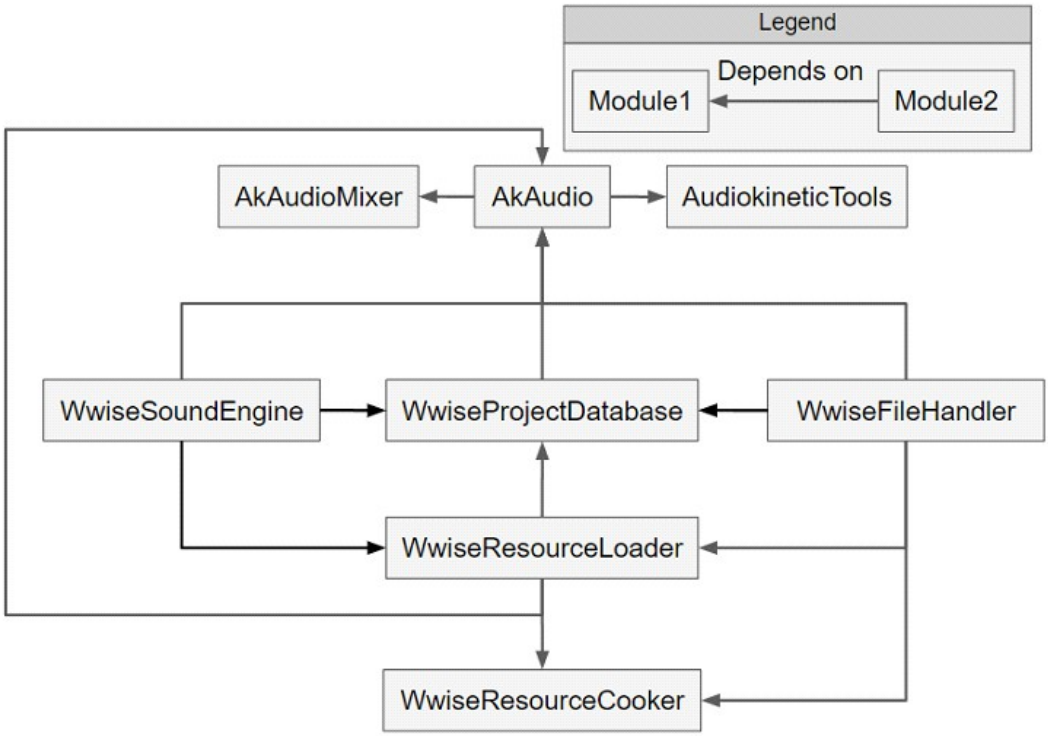
WwiseFileHandler: Fornisce a Wwise Sound Engine i media, i SoundBank e le fonti esterne di cui ha bisogno, inclusa la gestione delle richieste da Wwise Location Resolver e I/O Hook.

WwiseProjectDatabase: Fornisce una visualizzazione del database basata sulla memoria dello stato corrente dei SoundBank generati dal progetto Wwise.

WwiseResourceCooker: Converte le strutture WwiseObjectInfo in strutture cotte e copia le risorse richieste nella directory di gestione temporanea.

WwiseResourceLoader: Gestisce il caricamento e lo scaricamento di tutte le risorse necessarie per tutti i tipi di risorse Wwise.

WwiseSoundEngine: Fornisce il collegamento tra l'API Wwise Sound Engine e il progetto Unreal.



-Overriding the Wwise Unreal Integration
I moduli dell'Integrazione, a parte WwiseSoundEngine, AudiokineticTools e AkAudio, sono progettati per essere estensibili tramite ereditarietà. La maggior parte delle funzioni di questi moduli sono virtuali per consentire agli sviluppatori di sovrascriverle e modificare il loro comportamento per adattarlo al meglio al loro progetto.
Per sovrascrivere un modulo, crea un nuovo modulo e assicurati che erediti il modulo che desideri sovrascrivere. Quindi, sovrascrivere le funzioni desiderate.
Per utilizzare il tuo nuovo modulo, assicurati che sia incluso nel Build.cs del tuo gioco e aggiunto a DefaultEngine.ini in questo modo
[Audio]
 WwiseFileHandlerModuleName=WwiseSimpleExternalSource
-Creating AkComponents Using C++
Quando crei un nuovo AkComponent utilizzando C++, devi collegarlo a un componente principale.
-Using the Wwise API in your Unreal Projects
È possibile utilizzare l'API Wwise tramite Wwise Unreal Integration. Puoi chiamare la maggior parte delle funzioni tramite il modulo WwiseSoundEngine o il modulo AkAudio.
-WwiseSoundEngine Module
Il modulo WwiseSoundEngine fornisce accesso di basso livello alla maggior parte delle operazioni API SoundEngine.

```
#include "AkAudioEvent.h"
#include "AkComponent.h"
#include "Wwise/LowLevel/WwiseLowLevelSoundEngine.h"

void PostEventSoundEngine(UAkAudioEvent* Event, AActor* GameObject)
{

    if (auto* SoundEngine = IWwiseSoundEngineAPI::Get())
    {
        if(Event && GameObject)
        {
            TArray<UAkComponent*> Array;
            GameObject->GetComponents<UAkComponent>(Array);
            //The Actor doesn't have an AkComponent attached. This component is needed if we want sound to be emitted from this actor.
            if (Array.Num() == 0)
            {
                //PostEvent doesn't load the event. The Event needs to be loaded beforehand.
                if (!Event->IsLoaded())
                {
                    Event->LoadData();
                }

                auto* AkAudioComponent = NewObject<UAkComponent>(GameObject);
                AkAudioComponent->SetupAttachment(GameObject->GetDefaultAttachComponent());
                Array.Add(AkAudioComponent);
                AkAudioComponent->RegisterComponent();
            }

            SoundEngine->PostEvent(Event->GetWwiseShortID(), Array[0]->GetAkGameObjectID());
        }
    }
}
```

Questo codice pubblica un evento su un determinato attore. È necessario aggiungere UAkComponent sull'attore per registrarlo con Wwise. Anche l'evento deve essere caricato poiché il caricamento di un evento non viene eseguito automaticamente nel modulo WwiseSoundEngine.

-AkAudio Module

Il modulo AkAudio è il modulo radice per la maggior parte delle funzionalità rivolte all'utente dell'integrazione.
Può realizzare la maggior parte di ciò che il modulo WwiseSoundEngine può eseguire utilizzando Unreal Components. Contiene classi per i tipi di asset e i componenti utilizzati dall'integrazione.

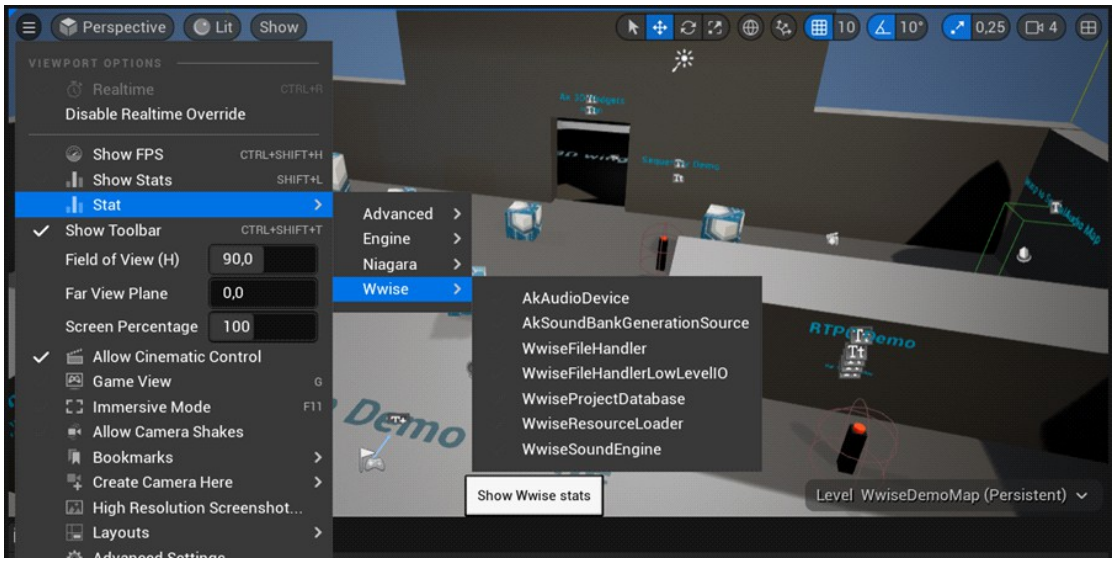
```
#include "AkAudio.h"

void PostEvent(UAkAudioEvent* Event, AActor* GameObject)
{
    if (FAkAudioDevice::Get())
    {
        FAkAudioDevice::Get()->PostAkAudioEventOnActor(Event, GameObject);
    }
}
```

-Debug Features
--Logging
Se si verifica un errore, puoi trovare quale modulo ha prodotto l'errore e quindi puoi impostare la verbosità dei log di quel modulo su Verbose o VeryVerbose per comprendere meglio il motivo per cui si è verificato l'errore.
//In C++, la verbosità si riferisce al livello di dettaglio o alla quantità di informazioni visualizzate durante l'esecuzione del programma.
Ricorda che la verbosità trova un equilibrio tra fornire informazioni utili ed evitare risultati travolgenti.//
È particolarmente rilevante quando si ha a che fare con la registrazione, il debug o il

monitoraggio dell'avanzamento di determinate operazioni.
È possibile modificare il livello di dettaglio di ciascun modulo nella sezione Core.Log del file DefaultEngine.ini

Sono possibili i seguenti livelli di verbosità:
Fatal
Error
Warning
Display
Log
Verbose
VeryVerbose
-Stats
AkAudioDevice = Registra le informazioni sulle chiamate asincrone post evento
AkSoundBankGenerationSource = Registra le chiamate WAAPI relative alla generazione di SoundBank.
WwiseConcurrency = Registra le informazioni sulla memoria utilizzata per gestire il multithreading.
//Che cos'è il Multithreading?
Per aumentare la velocità del core del processore senza modificare la frequenza di clock, si fa in modo che la CPU possa eseguire più operazioni simultaneamente.
Un thread può essere considerato come una porzione di un processo//
WwiseFileHandler = Registra le informazioni sui contenuti multimediali caricati, SoundBank e fonti esterne.
WwiseFileHandlerLowLevelIO = Registra le informazioni sui suoni in streaming.
WwiseMemory = Registra informazioni sull'utilizzo della memoria.
WwiseObstructionOcclusion = Registra informazioni sui calcoli di ostruzione e occlusione
WwiseProjectDatabase = Registra le informazioni sulla memoria utilizzata dal modulo WwiseProjectDatabase.
WwiseResourceLoader = Registra le informazioni sull'AkType di riferimento
WwiseSoundEngine = Registra le informazioni sulle chiamate API



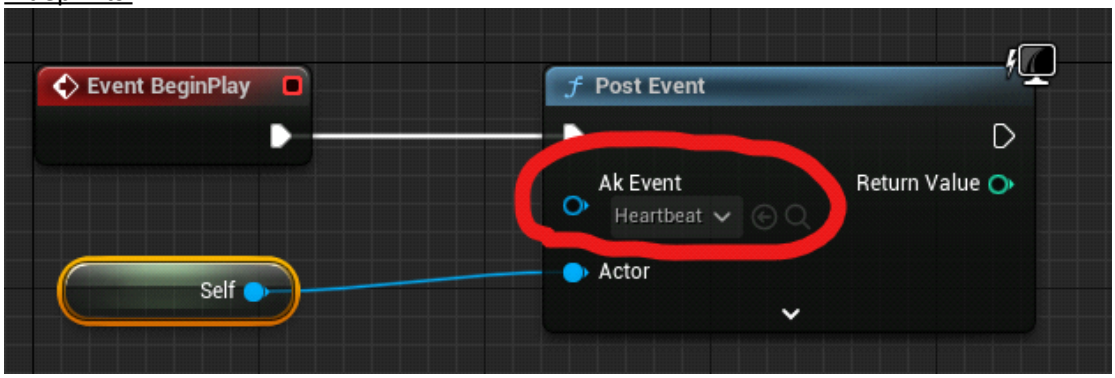
DA COMPLETARE
////-Codin' Time

--Events

Esistono vari modi per pubblicare eventi, tramite blueprint o C++.

1) Post Event, Questo nodo pubblica l'evento che scegli dal menu a discesa sull'attore selezionato.
Post Evento è una funzione che restituisce (valore di ritorno) un valore intero (un ID specifico chiamato ID di gioco) per quell'evento. Utilizzando questo PlayingID, possiamo raggiungere successivamente quell'istanza specifica dell'evento

Blueprints



C++
In generale, quando creiamo una classe C++, genera un file header(.h) e un file C++(.cpp).
Crea una classe C++ (puoi scegliere Actor come classe base) e apri il file di intestazione nel tuo IDE preferito.
La logica di questo è un po' simile al precedente metodo Blueprints; sappiamo che abbiamo bisogno di un evento Ak e di una funzione per pubblicarlo.
Quindi andiamo uno per uno. Evento Ak: dove si trova? È dove vive l'onnipotente AkGameplayStatics.

Per includere quel file nella tua classe, puoi semplicemente aggiungere questa riga di codice qui sotto nelle prime righe del file di intestazione:

#include "Plugins/Wwise/Source/AkAudio/Classes/AkGameplayStatics.h"///////

.Using Wwise Unreal Objects

-Wwise Unreal Assets
L'Unreal Integration richiede agli utenti di creare risorse per rappresentare gli oggetti corrispondenti nel progetto Wwise.
È possibile creare queste risorse utilizzando la funzione di trascinamento del trascinamento dal browser Wwise o dal menu di scelta rapida nel browser dei contenuti.
Gli asset creati tramite drag-and-drop contengono automaticamente le informazioni (GUID, ShortID, Nome) necessarie per collegarli agli oggetti Wwise che rappresentano.
Quando crei risorse tramite il browser dei contenuti, devi aggiungere queste informazioni manualmente.
/Il GUID (Globally Unique Identifier, identificatore unico globale) è un numero pseudo-casuale utilizzato nella programmazione software per distinguere vari oggetti.//
Alcune classi di asset utilizzano una sottoclasse con campi aggiuntivi che influiscono sui comportamenti, ad esempio il modo in cui i dati vengono preparati per il caricamento.
Questa struttura viene serializzata nell'asset quando viene salvata, ma eliminata quando viene cotta per una compilazione in pacchetto.

-WwiseObjectInfo
Quando le risorse nell'editor vengono caricate e cotte, questo struct viene usato per eseguire una query su WwiseDatabase per ottenere
la struttura CookedData utilizzata per caricare tutte le risorse necessarie dell'asset (file con estensione bnk e wem) in memoria.
Proprietà :
1. WwiseGuid: GUID che identifica questa risorsa nel progetto Wwise.
2. WwiseShortId: numero intero a 32 bit che rappresenta la risorsa in Wwise SoundEngine.
3. WwiseName: nome stringa della risorsa nel progetto Wwise.
4. HardCodedSoundBankShortId: numero intero a 32 bit che identifica il SoundBank che l'asset deve caricare come dipendenza.
Ad esempio, se un evento è incluso in più banchi di suoni, è possibile specificare quale banco di suoni creare un pacchetto e caricare con tale risorsa.
Se il campo non è impostato, tutti i banchi di suoni che contengono l'oggetto Wwise vengono caricati con esso.

-WwiseEventInfo
Sottoclasse di FWwiseObjectInfo per le risorse UAkAudioEvent.
FWwiseEventInfo contiene opzioni aggiuntive che consentono agli utenti di personalizzare il comportamento dell'asset evento quando viene caricato ed eliminato definitivamente.
Proprietà aggiuntive:
1. SwitchContainerLoading: "AlwaysLoad" per caricare tutti i supporti commutati con l'evento che li utilizza o "LoadOnReference"
er caricare i supporti solo quando i valori dell'opzione o del gruppo sono presenti anche in memoria.
2. DestroyOptions: "StopEventOnDestroy" per interrompere la riproduzione dell'evento quando viene eliminato o "WaitForEventEnd" per attendere l'interruzione della riproduzione dell'evento.

-WwiseGroupValueInfo
Sottoclasse di FWwiseObjectInfo per gli asset Switch e State Value. FWwiseGroupValueInfo contiene anche lo stato o l'oggetto ShortId del gruppo di switch.
Proprietà aggiuntive:
1. GroupShortId: numero intero a 32 bit che identifica il gruppo che include la risorsa.

-Wwise Asset Cooked e Caricati i dati
Le risorse Wwise contengono una struttura CookedData che viene utilizzata per caricare le risorse richieste (file .bnk e .wem) e per monitorare
le loro dipendenze da altri oggetti Wwise (ad esempio, quali opzioni vengono utilizzate dagli eventi che riproducono i contenitori di interruttori).

////////////

Sottoclasse di bus ausiliari UAkAudioType.
AuxBusInfo = FWwiseObjectInfo che compila la proprietà AuxBusCookedData quando si è nell'editor.

5. UAkEffectShareSet
Sottoclasse di UAkAudioType per gli ShareSet di effetti.
- ShareSetInfo = FWwiseObjectInfo che compila la proprietà AuxBusCookedData quando si è nell'editor.
- ShareSetCookedData = transitorio FWwiseLocalizedShareSetCookedData che contiene la mappatura dalle lingue a FWwiseShareSetCookedData specifici della lingua.

6. UAkGroupValue
Sottoclasse di UAkAudioType e classe base per UAkStateValue e UAkSwitchValue.
- GroupValueInfo = FWwiseGroupValueInfo che compila la proprietà GroupValueCookedData quando si trova nell'editor.
- GroupValueCookedData = FWwiseGroupValueCookedData transitorio.

7. UAkStateValue
Sottoclasse di UAkGroupValue per i valori di stato.

8. UAkSwitchValue
Sottoclasse di UAkGroupValue per i valori Switch.

9. UAkRtpc
Sottoclasse di UAkAudioType per RTPC/Parametri di gioco.

RtpcInfo = FWwiseObjectInfo che compila la proprietà GameParameterCookedData nell'editor.
GameParameterCookedData = transitorio FWwiseGameParameterCookedData.

10. UAkTrigger
Sottoclasse di UAkAudioType per trigger musicali.
TriggerInf = FWwiseObjectInfo che compila la proprietà TriggerCookedData nell'editor.
TriggerCookedData = transitorio FWwiseTriggerCookedData.

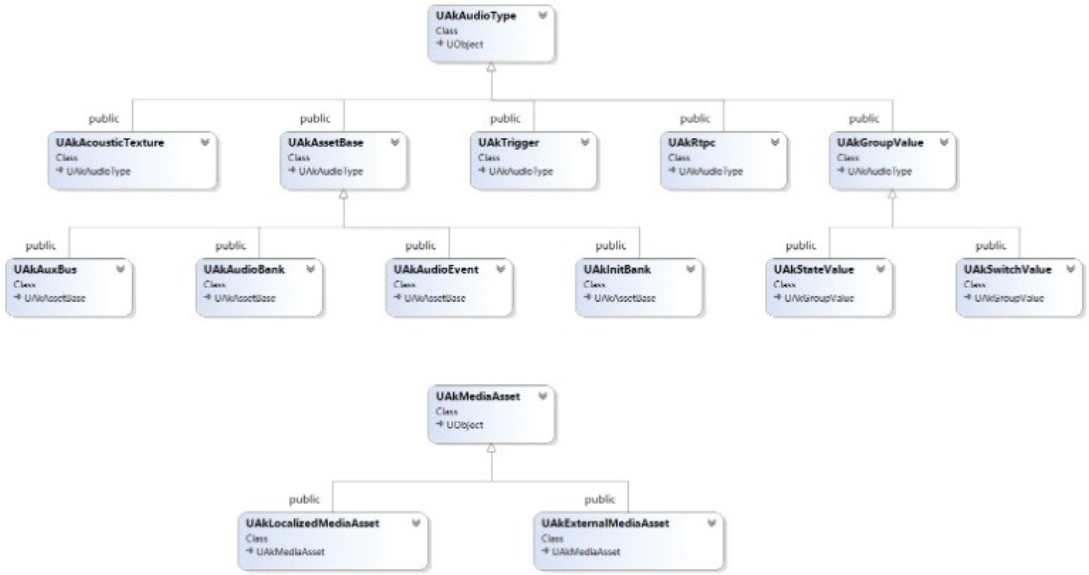
11. UAkAcousticTexture
Sottoclasse di UAkAudioType per trame acustiche.
TriggerInfo = FWwiseObjectInfo che compila la proprietà AcousticTextureCookedData quando si trova nell'editor.
AcousticTextureCookedData = transitorio FWwiseAcousticTextureCookedData.

-Deprecated Asset and Data Types
Le seguenti classi e strutture sono deprecate e vengono utilizzate solo per la migrazione da

progetti precedenti:

UAkAssetData
UAkAssetPlatformData
UAkAudioBank
UAkExternalMediaAsset
UAkFolder
UAkLocalizedMediaAsset
UAkMediaAsset
UAkMediaAssetData

- Viewing Asset Links in Unreal
È possibile utilizzare Unreal Reference Viewer per visualizzare i collegamenti tra le risorse, come mostrato nell'immagine seguente.



//

.Wwise Unreal Actors

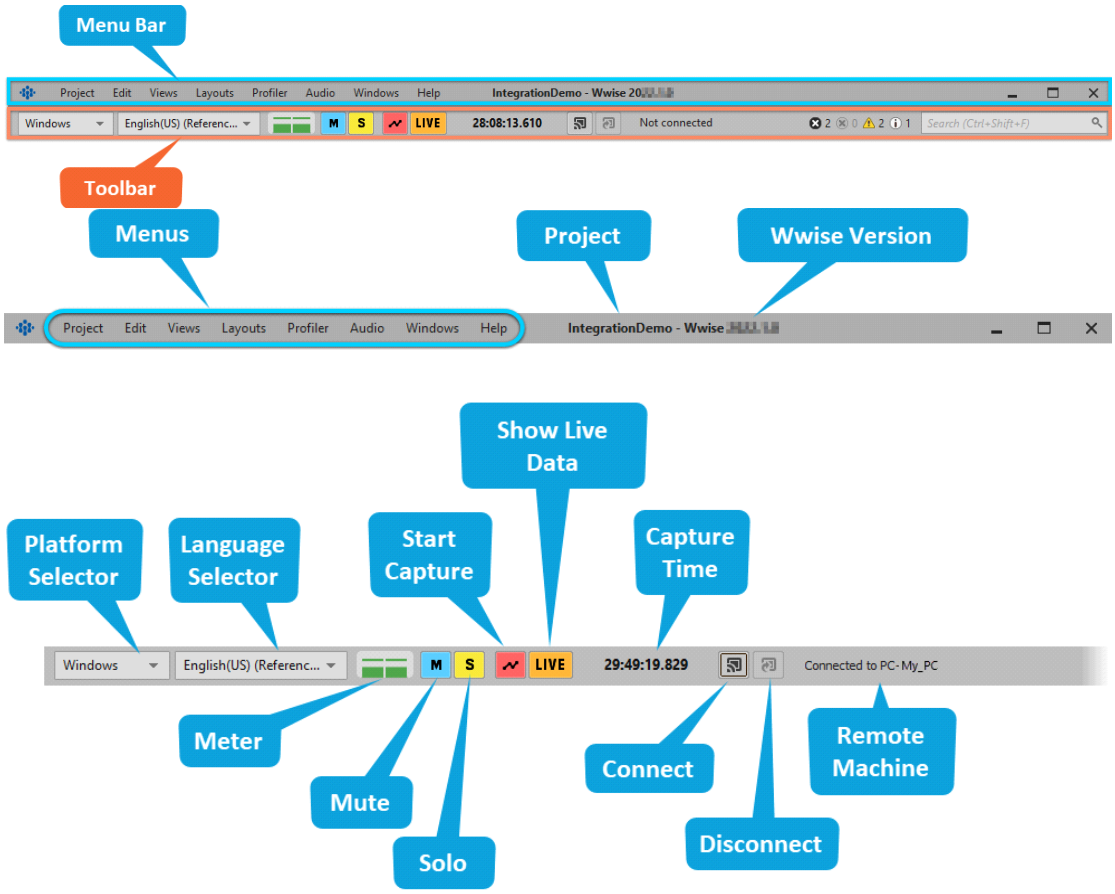
-AkAmbientSound
AkAmbientSound è una classe AActor utilizzata allo stesso modo dell'oggetto AAmbientSound incluso nel sistema Unreal Audio.
Esistono due modi per controllarne la riproduzione :
1. Tramite le funzioni Blueprint
2. Start All Ambient Sounds e Stop All Ambient Sounds funzioni di aiuto globali

Blueprint Functions:
Start All Ambient Sound = Avvia la riproduzione di tutti i suoni ambientali.
Start Ambient Sound = Avvia la riproduzione del suono ambientale selezionato.
Stop All Ambient Sound = Interrompe la riproduzione di tutti i suoni ambientali.
Stop Ambient Sound = Interrompe la riproduzione del suono ambientale selezionato.

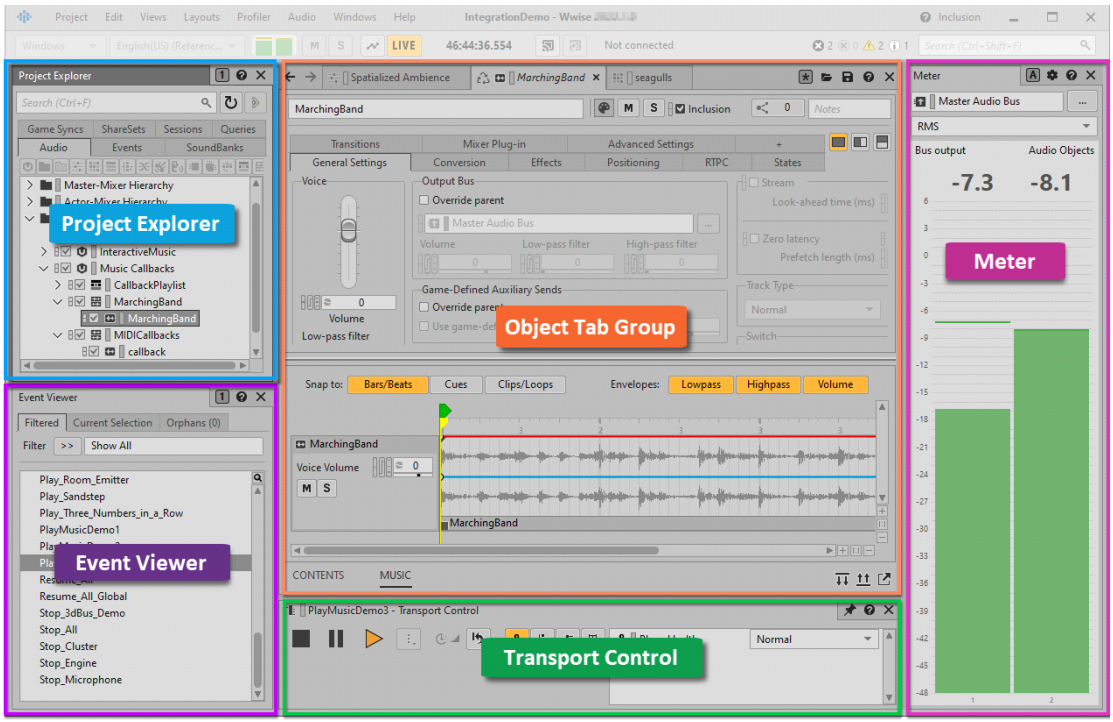
-AkReverbVolume
AkReverbVolume è una classe AVolume utilizzata in modo simile all'oggetto AReverbVolume incluso nel sistema Unreal Audio.
Qualsiasi pennello nell'editor può generarlo. L'effetto riverbero è ottenuto con un AkLateReverbComponent.

////////////

La barra degli strumenti fornisce un accesso rapido a determinati strumenti, come i selettori della piattaforma o della lingua, gli strumenti di acquisizione, il connettore della piattaforma remota e lo strumento di ricerca.



Al primo avvio di Wwise, viene visualizzato il layout Designer.



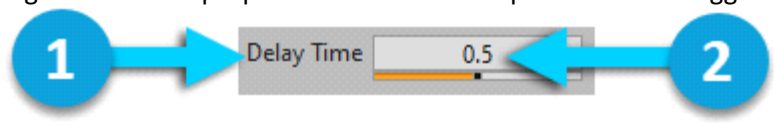
- **Project Explorer :** L'area principale dove gestisci e organizzi i vari elementi del tuo progetto Wwise.
- **Audio** : Una visualizzazione ad albero gerarchica, proprio come Windows Explorer e Mac Finder, in cui puoi organizzare le risorse nel tuo progetto.
- **Events :** Visualizza gli eventi, sia azioni che dialoghi, nel tuo progetto.
- **SoundBanks :** Visualizza tutti i SoundBank nel tuo progetto.
- **Game Syncs :** Visualizza tutti gli interruttori, gli stati, i parametri di gioco e i trigger nel tuo progetto.
- **ShareSets**: Visualizza tutti gli ShareSet di effetti e attenuazione nel progetto.
- **Sessions :** Visualizza tutte le sessioni Soundcaster nel tuo progetto
- **Queries:** Visualizza tutte le query nel tuo progetto

- **Event Viewer :** Visualizza i diversi eventi che sono stati creati per il progetto corrente. Il Visualizzatore eventi ha tre diverse schede: Filtrati, Selezione corrente e Orfani, ognuna delle quali filtra gli eventi in modo diverso**.**
- **Object Tab Group** : Contiene zero o più schede oggetto a seconda delle selezioni effettuate. Qualsiasi oggetto in Wwise può essere visualizzato in una Scheda Oggetto, che si adatta automaticamente alle caratteristiche dell'oggetto.
- **Transport Control:** Riproduce i tuoi oggetti.
- **Meter view:** Visualizza i valori codificati a colori per canale di tre diversi tipi di livelli, vale a dire Picco, Picco reale e RMS. Per i bus ambisonici, la vista Misuratore può anche visualizzare il misuratore 3D, che mostra i dati direzionali.

-Using Text Boxes and Sliders

La maggior parte delle visualizzazioni in Wwise contengono caselle di testo in cui è possibile

digitare valori di proprietà o informazioni specifiche su un oggetto.



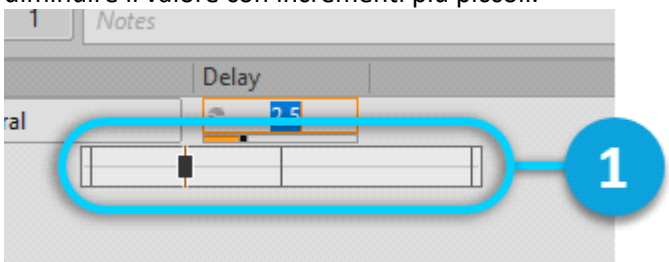
1 Nome della casella di testo.

2 Valore

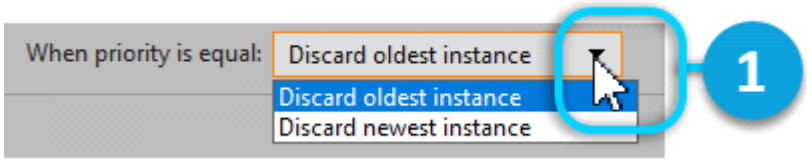
È possibile copiare il testo da una casella di testo facendo clic con il pulsante destro del mouse sul testo e selezionando Copia. Quando il testo viene visualizzato in una casella di testo ma non può essere modificato, non è disponibile.

Se desideri ripristinare l'impostazione predefinita del valore di una proprietà, fai clic tenendo premuto Ctrl nella casella di testo.

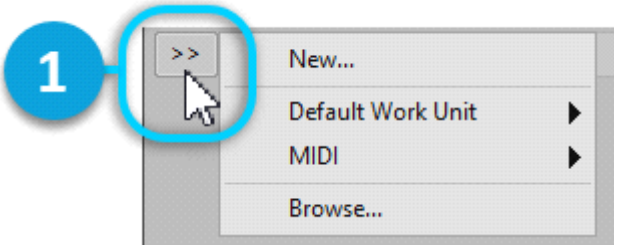
Quando si fa clic e si tiene premuto un punto qualsiasi della casella di testo, viene visualizzato il super cursore, che lo rende Più facile per l'utente ottimizzare il valore della proprietà. Quando si rilascia il pulsante del mouse, più grande è il cursore scompare. Se è necessario definire il valore in modo più preciso, è possibile tenere premuto Maiusc+trascinare il pulsante per aumentare o diminuire il valore con incrementi più piccoli.



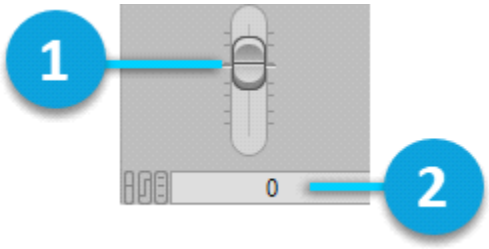
-Using Lists
Ci sono due tipi di liste in Wwise: una lista a discesa e una lista di scorciatoie (a volte chiamato elenco contestuale o contestuale). L'elenco a discesa, denominato semplicemente "elenco" in Questa documentazione della Guida è un campo che contiene una serie di opzioni predefinite. Per visualizzare nell'elenco, fare clic sulla freccia a destra del campo.



Il pulsante di selezione (>>) visualizza una serie di opzioni o azioni. Il menu di scelta rapida Può avere o meno un campo associato per visualizzare l'opzione selezionata. Fare clic sull'icona per visualizzare le opzioni del menu.



-Using Faders



Alcune proprietà, come Volume, utilizzano cursori o fader verticali invece di cursori orizzontali per modificare i propri valori.

-Using Tables

Una tabella è una disposizione di dati che utilizza una serie di righe e colonne. Le informazioni contenute in La prima colonna si riferisce specificamente alle informazioni contenute nelle altre colonne. In Wwise, Le tabelle visualizzano informazioni su elementi come eventi e predefiniti.

Per configurare le colonne in una tabella:

- Posizionare il mouse sulla banda dell'intestazione della colonna.
- Fare clic con il pulsante destro del mouse sulla banda di intestazione.
- Viene visualizzato il menu di scelta rapida con l'unica opzione: Configura Colonne....
- Selezionare Configura colonne....

L'opzione Configura colonne... viene visualizzata la finestra di dialogo in cui è possibile specificare le colonne display e il loro ordine.

No.		Actions	Objects	Scope	Notes
1	>>	Play	rlaunch	Game object	
2	>>	Play	Thrust	Game object	
3	>>	Set State	PlayerInWater : No	Global	Rockets do not fire underwater.

Per filtrare gli elementi in una tabella:

Fai clic sull'icona Cerca nell'angolo in alto a destra. Scorciatoia predefinita: Ctrl+F3.

Viene visualizzato un campo di ricerca alfanumerico standard.

-Comprendere gli elementi visivi in Wwise

Cartelle

Physical Folder, Una cartella che rappresenta una directory effettiva sul disco.

Virtual Folder, Una cartella che può essere aggiunta in tutto l'Esploratore progetto per raggruppare gli elementi del progetto.

Master Mixer Icons

Audio Bus, Un raggruppamento di oggetti sonori che ti consente di lavorare con diverse strutture all'interno di un gioco.

Auxiliary Bus, Un sottogruppo di oggetti audio in qualsiasi punto del progetto per la regolazione del volume, della configurazione del bus, del posizionamento e dell'RTPC, nonché per l'applicazione di effetti o stati prima del reindirizzamento a un bus audio principale. Le regolazioni di ducking, HDR e mix vocale non sono ottimali

Mixing, (Audio) (Auxiliary) Uno stato di elaborazione che Wwise può applicare sia ai bus audio che ai bus ausiliari. Le proprietà di un bus di missaggio vengono passate agli oggetti instradati al bus e ai bus figlio.
Tutti gli effetti definiti sul bus vengono elaborati e il risultato viene emesso in un formato basato sul canale.

Processing Audio Objects, (Audio) (Auxiliary) Uno stato di elaborazione che Wwise può applicare sia ai bus audio che ai bus ausiliari. Tutte le proprietà di questo bus vengono passate agli oggetti instradati al bus e ai bus figlio. Tutti gli effetti definiti sul bus vengono elaborati e il risultato viene emesso come oggetti audio.

Not Mixing, Uno stato di elaborazione che Wwise può applicare ai bus audio. Le proprietà di un bus Non-Mixing vengono passate agli oggetti instradati al bus e ai bus figli, tuttavia, non viene eseguita alcuna elaborazione e quindi il formato degli oggetti e dei figli non viene modificato. Qualsiasi proprietà creata su un bus Non Mixing viene applicata agli oggetti instradati a livello vocale.

Processing, (Audio) (Auxiliary), Uno stato di elaborazione che Wwise può applicare sia ai bus audio che ai bus ausiliari. Lo stato di elaborazione di questo bus verrà risolto come Mixing o Elaborazione di oggetti audio quando il motore audio Wwise si connette all'endpoint e riceve informazioni relative ai formati supportati.

Actor-Mixer Icons

, Audio source, Un livello di astrazione separato tra il file audio e l'oggetto. È collegato al file audio importato nel tuo progetto ed è dove definisci le Impostazioni di conversione (per le piattaforme di gioco attive).

, Source Plug-in, Una sorgente audio creata da un plug-in sorgente proveniente dall'esterno di Wwise.

, Sound SFX, Un oggetto sonoro contenente effetti sonori.

, Sound Voice, Un oggetto sonoro contenente la voce fuori campo o il dialogo dei personaggi.

, Work Unit, Un file XML distinto che ti consente di dividere il tuo progetto in segmenti separati in modo che diversi membri del tuo team possano lavorare contemporaneamente su parti diverse del progetto. Questi file XML possono essere facilmente gestiti dal sistema di controllo del codice sorgente

, Blend Container, Un gruppo di uno o più oggetti e/o contenitori che vengono riprodotti simultaneamente. I suoni e i contenitori all'interno del contenitore Blend possono essere raggruppati in tracce blend in cui le proprietà del suono sono mappate sui valori dei parametri di gioco utilizzando RTPC. È inoltre possibile applicare dissolvenze incrociate tra gli oggetti all'interno di una traccia di fusione in base al valore di un parametro di gioco.

, Random Container, Un gruppo di uno o più oggetti e/o contenitori riprodotti in ordine casuale.

Sequence Container, Un gruppo di uno o più oggetti e/o contenitori che vengono riprodotti secondo un ordine o una playlist specifica.

, Switch Container, Un gruppo di oggetti e/o contenitori organizzati in una serie di Switch o Stati, in modo che possano essere riprodotti quando lo Switch o lo Stato associato vengono richiamati dal gioco.

<u>Interactive Music Icons</u>

, Music Track, Un oggetto musicale che contiene arrangiamenti di singoli clip musicali visualizzati in forma d'onda in modo da poterli allineare visivamente in un segmento musicale.

, Music Track - Random Step, Una traccia musicale che riproduce le sue sottotracce in ordine casuale ogni volta che viene riprodotto il segmento musicale principale.

, Music Track - Sequence Step, Una traccia musicale che riproduce le sue sottotracce in ordine sequenziale ogni volta che viene riprodotto il segmento musicale principale.

, Music Track - Switch, Un tipo di traccia musicale che riproduce le sue sottotracce in base agli interruttori/stati associati del gruppo di interruttori/stati specificato.

, Music Segment, Un oggetto musicale che contiene tracce musicali che possono essere allineate utilizzando punti di sincronizzazione per arrangiamenti musicali nella musica interattiva.

, Music Playlist Container, Un gruppo di uno o più segmenti organizzati in un modo particolare in modo da poter essere riprodotti in ordine casuale o secondo un ordine specifico

, Music Switch Container, Un gruppo di uno o più segmenti musicali e contenitori organizzati in serie di interruttori o stati, in modo che possano essere riprodotti quando lo switch o lo stato associato viene richiamato dal gioco.

Other Project Element Icons

, Event, Un metodo per attivare l'audio o il movimento nel gioco utilizzando un'azione o una serie di azioni, come Riproduci, Disattiva audio e Pausa, che sono state applicate a uno o più oggetti Wwise.

, Dialogue Event, Un metodo per attivare l'audio, che può essere o meno un dialogo o un movimento nel gioco utilizzando una combinazione di gruppi di stati/interruttori e stati/interruttori disposti in percorsi assegnati a un oggetto nella gerarchia di attori-mixer di Wwise.

, SoundBank, Un gruppo di eventi, oggetti Wwise e contenuti multimediali che verranno caricati nella memoria della piattaforma di gioco in un punto particolare del gioco.

, Switch Group, Una raccolta di Switch correlati che sono stati raggruppati insieme per aiutare a gestire le diverse alternative esistenti per un dato elemento all'interno del gioco.

, Switch, Un'alternativa che esiste per un particolare elemento all'interno del gioco.

, State Group, Una raccolta di stati correlati che sono stati raggruppati insieme per aiutare a gestire i cambiamenti globali che si verificano nell'ambiente di gioco.

, State, Uno spostamento o una regolazione globale delle proprietà audio del gioco che rappresenta i cambiamenti nelle condizioni fisiche e ambientali del gioco.

, Game Parameter (RTPC), Un parametro nel tuo gioco, ad esempio velocità e RPM in un gioco di corse automobilistiche, che può essere mappato sui valori delle proprietà Wwise utilizzando RTPC.

, Trigger, Una particolare azione di gioco che porta alla riproduzione di un breve brano musicale.

, Effect ShareSet, Impostazioni degli effetti audio che possono essere utilizzate per migliorare l'audio del gioco. Queste impostazioni sono state salvate come ShareSet e possono essere condivise tra oggetti.

, Attenuation ShareSet, Impostazioni di attenuazione relative al volume di un suono in base alla sua distanza dall'ascoltatore.
Queste impostazioni sono state salvate come ShareSet e possono essere condivise tra oggetti.

, Conversion Settings ShareSet, Le impostazioni di conversione, che includono frequenza di campionamento, formato audio e numero di canali, aiutano a definire la qualità complessiva dell'output audio. Queste impostazioni sono state salvate come ShareSet e possono essere applicate e condivise tra oggetti.

, Soundcaster Session, Un gruppo di moduli audio, musicali, di movimento ed eventi disposti in un ordine particolare che sono stati salvati insieme alle impostazioni di sincronizzazione del gioco ai fini di una simulazione.

, Mixing Session, Un gruppo di bus e/o oggetti, insieme alle proprietà corrispondenti, che sono stati salvati in un tipo di console di missaggio allo scopo di ottimizzare il mix audio del gioco.

, Query, Un insieme specifico di criteri di ricerca utilizzati per trovare un particolare oggetto o elemento del progetto.

Property Editor and Effect Editor Splitters

, No split, L'editor visualizza una singola scheda di informazioni.

, Column split, L'editor visualizza due schede di informazioni in pannelli verticali.

, Row split, L'editor visualizza due schede di informazioni in pannelli orizzontali.

Property Value Indicators

, Link, il valore della proprietà è collegato ai valori delle altre piattaforme di gioco attive.

, Unlink, Il valore della proprietà non è legato ai valori delle altre piattaforme di gioco attive

, Partial Unlink, Il valore della proprietà per la piattaforma corrente è collegato alle altre piattaforme attive, ma uno o più valori corrispondenti di altre piattaforme sono scollegati.

, Link Mixed, Alcuni degli oggetti selezionati hanno stati di collegamento diversi. Alcuni possono essere collegati, mentre altri sono scollegati o parzialmente scollegati.

, RTPC Disabled, Questo valore di proprietà non è legato al valore di un parametro di gioco.

, RTPC Enabled, Il valore di un parametro di gioco è legato al valore di questa proprietà. Ciò significa, ad esempio, che la velocità di un'auto nel gioco può essere collegata direttamente alla

proprietà del campo in Wwise. Man mano che la velocità dell'auto aumenta nel gioco, il campo in Wwise aumenterà in tempo reale.

, RTPC Mixed, I valori dei parametri di gioco sono legati ai valori delle proprietà per uno o più, ma non tutti, gli oggetti caricati nel Multi Editor. Non vedrai questo indicatore nell'editor delle proprietà o nell'editor dei contenuti.

, Randomizer Enabled, Un valore di proprietà a cui è stato applicato un effetto Randomizer.

, Randomizer Disabled, Un valore di proprietà a cui non è stato applicato alcun effetto Randomizer.

, Randomizer Mixed, A Randomizer effect applies to the property value of one or more, but not all, objects loaded in the Multi Editor. You will not see this indicator in the Property Editor or the Contents Editor.

, State Disabled, Questo valore della proprietà non è legato ad uno Stato.

, State Enabled, Un gruppo di stati è legato a questo valore della proprietà. Ciò significa che la proprietà in questione, come il Volume, può cambiare a seconda dello Stato richiesto.

, State Mixed, Un gruppo di stati è legato ai valori delle proprietà per uno o più, ma non tutti, gli oggetti caricati nel Multi Editor. Non vedrai questo indicatore nell'editor delle proprietà o nell'editor dei contenuti.

Property Shading

-96.3, "Unsupported feature", Se una proprietà è ombreggiata in blu, significa che la funzionalità non è supportata sulla piattaforma corrente

Preset Icons

, Save Preset, Un comando per salvare lo stato corrente di tutti i valori all'interno della vista corrente in una preimpostazione.

, Load Preset, Un comando per caricare un Preset salvato in precedenza.

Selected Controls

0, "Selected Control", Un bordo di attivazione (bianco quando si utilizza il tema Wwise Classic) che appare attorno ai controlli selezionati.

View Icons

, Help, Un comando per visualizzare l'argomento di riferimento correlato a quella particolare

vista, finestra o finestra di dialogo. L'argomento di riferimento può essere online o offline, a seconda delle impostazioni della documentazione. Fare riferimento a Impostazione delle preferenze della documentazione.

, View Settings, Un comando per visualizzare una finestra di dialogo con una serie di impostazioni che possono essere utilizzate per definire la vista.

, Selection Channel, Un comando per cambiare la vista su un particolare canale di selezione (1-4).

, Close View, Un comando per chiudere la vista mobile.

Voice Inspector Icons

//La pipeline di sviluppo del gioco è il processo di creazione di un videogioco dall'ideazione al completamento. Proprio come una linea di produzione, aiuta a organizzare il flusso di lavoro in modo che tutti i soggetti coinvolti sappiano cosa devono consegnare e quando1.//

, Project Value, Una modifica del volume standard che fornisce un contributo alla pipeline vocale quando non è presente una connessione di gioco.

, SoundBank Value, Una modifica del volume standard che determina un contributo alla pipeline vocale quando si è connessi a un gioco.

, Live Value, Un contributo in termini di volume alla pipeline vocale derivante da una modifica del valore della proprietà durante la connessione a un gioco.

, API, Una chiamata API che fornisce un contributo alla pipeline vocale.

, Una finestra HDR che fornisce un contributo alla pipeline vocale.

, Ducking, Un ducking (abbassamento) di un suono che guida un contributo alla pipeline vocale.

, Attenuation Distance, Una distanza dall'emettitore all'ascoltatore, che determina un contributo alla pipeline vocale.

, Blend Track, Una traccia di fusione, che fornisce un contributo alla pipeline vocale.

, Cone Attenuation, Un'attenuazione dell'angolo del cono, che fornisce un contributo alla pipeline vocale.

, Ray, Un raggio di distanza, tra l'emettitore e l'ascoltatore, che guida un contributo alla pipeline vocale.

, Occlusion, Un'occlusione del raggio, che guida un contributo alla pipeline vocale.

, Obstruction, Un'ostruzione del raggio, che determina un contributo alla pipeline vocale.

, Volume, Una modifica alla proprietà/al dispositivo di scorrimento del volume, che determina un contributo alla pipeline vocale.

, Send Volume, Una modifica alla proprietà/al cursore del volume di invio del bus Aux, che determina un contributo alla pipeline vocale.

, Output Bus Volume, Una modifica alla proprietà/al cursore del volume del bus di output, che determina un contributo alla pipeline vocale.

, Bus Volume, Una modifica alla proprietà/al dispositivo di scorrimento del volume del bus, che determina un contributo alla pipeline vocale.

, Make-up Gain, Un aumento di volume applicato alla voce dopo tutte le altre regolazioni del volume, che determina un contributo alla pipeline vocale.

, Volume Mute, Un'azione di disattivazione dell'audio in un evento, che determina un contributo alla pipeline vocale.

, Volume Pause, Un'azione di pausa in un evento, che guida un contributo alla pipeline vocale.

, Low Pass, Un LPF, che fornisce un contributo alla pipeline vocale.

, Send Low Pass, Un LPF, da un bus aux invio, che invia un contributo alla pipeline vocale.

,Output Bus Low Pass Un LPF sul bus di uscita, che fornisce un contributo alla pipeline vocale.

, High Pass, Un HPF, che fornisce un contributo alla pipeline vocale.

, Send High Pass, Un HPF, da un bus aux send, che guida un contributo alla pipeline vocale.

, Output Bus High Pass, Un HPF sul bus di uscita, che fornisce un contributo alla pipeline vocale.

, Multiple Properties, Proprietà multiple, che influiscono su Volume, LPF e/o HPF, che determinano un contributo alla pipeline vocale.

, Send Multiple Properties, Proprietà multiple, da un invio del bus aux, che determinano un contributo alla pipeline vocale.

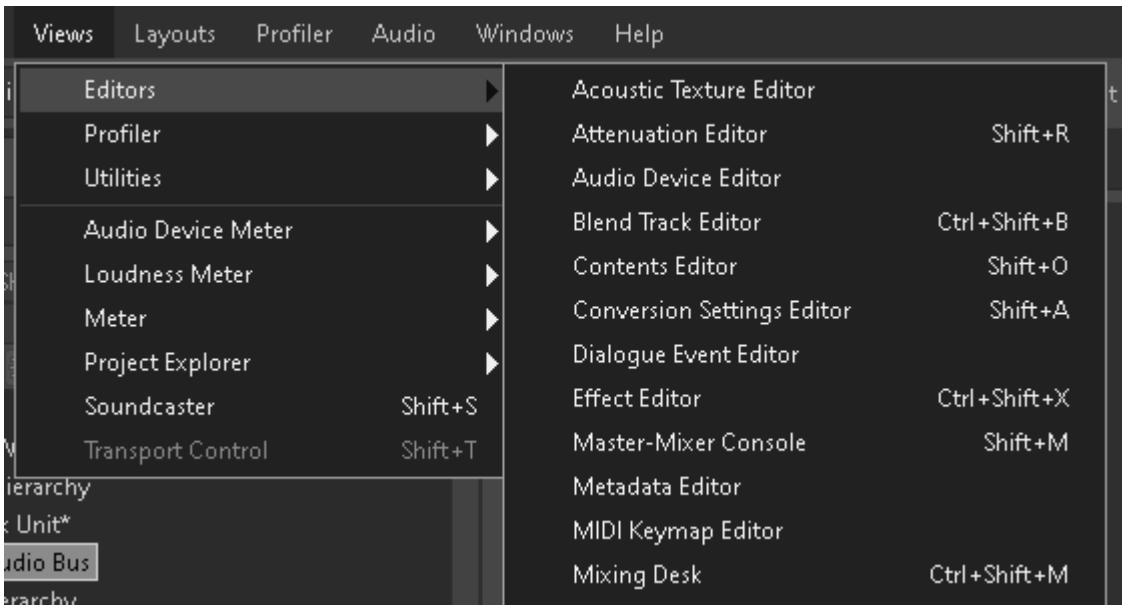
, Output Bus Multiple Properties, Proprietà multiple sul bus di uscita, che determinano un contributo alla pipeline vocale.

.Personalizing Your Workspace
- Working with Views

L'interfaccia di Wwise è composta da una serie di visualizzazioni. Una visualizzazione è una finestra separata nell'interfaccia che contiene un gruppo di informazioni o comandi relativi a un'attività specifica.

Per visualizzare una view;
Dalla barra dei menu, fare clic su **Views > <View Name>.**

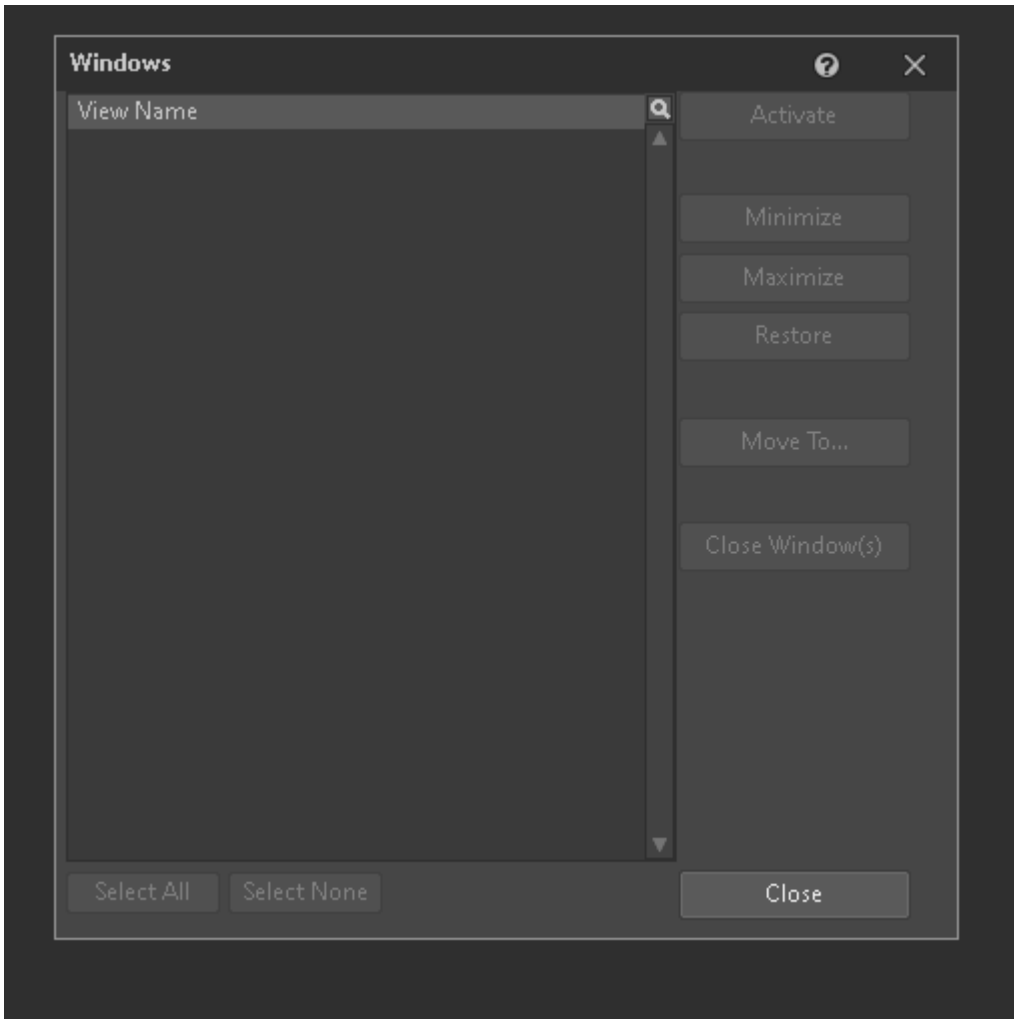
La vista selezionata viene visualizzata come vista mobile.



Per passare da una vista mobile all'altra:

Dalla barra dei menu, fare clic su **Windows** > <**View Name**>.

La vista selezionata diventa attiva nella parte anteriore della pila di finestre mobili.



-Understanding a View's Title Bar Icons

Le viste possono avere una serie di icone sul lato destro della barra del titolo.

Ciascuna icona esegue un comando diverso relativo a quella vista. A seconda della vista, verranno visualizzate diverse combinazioni di queste icone.

, Save Preset, Salva lo stato corrente di tutti i valori all'interno della vista corrente in una preimpostazione.

, Load Preset, Carica un Preset salvato in precedenza..

Sincronizza la vista su un canale particolare (1-4). Se utilizzate in Esplora progetti o Visualizzatore eventi, è possibile sincronizzare più istanze di queste visualizzazioni su più layout.

, Help, Visualizza l'argomento di riferimento relativo a quella particolare vista, finestra o finestra di dialogo.

, View Settings, Visualizza una finestra di dialogo con una serie di impostazioni che definiscono ciò che viene visualizzato nella vista.

, Close View, Chiude la vista mobile.

-Managing Floating Views Within a Layout

Quando si lavora all'interno di un particolare layout, è possibile avere diverse visualizzazioni mobili aperte contemporaneamente. Puoi gestire queste visualizzazioni mobili in modo efficiente utilizzando Windows Manager. All'interno di Gestione Windows è possibile attivare facilmente una particolare vista mobile, massimizza

-Understanding Selection Channels and Meter Instances

Puoi utilizzare i canali di selezione per sincronizzare le selezioni su più visualizzazioni. Con le istanze contatore è possibile sincronizzare i contatori su diversi layout. Ciò significa che il bus o il dispositivo audio selezionato in una vista meter viene automaticamente selezionato in tutte le viste meter della stessa istanza.

È possibile utilizzare i canali di selezione per sincronizzare le visualizzazioni in uno dei seguenti modi:

- Nel Project Explorer e Event Viewer: Se disponi di più istanze di Project Explorer e Event Viewereventi in layout diversi, potresti voler sincronizzare queste visualizzazioni.
- In altre visualizzazioni che includono l'icona del canale di selezione nella barra del titolo. utilizzare questa opzione per sincronizzare la vista con uno specifico canale di selezione

dell'Esploratore progetto e/o del Visualizzatore eventi.

Per sincronizzare più istanze di Esplora progetto o Visualizzatore eventi tra layout:

Pinned: Per disabilitare la sincronizzazione della vista con qualsiasi canale.

Sync with Selection Channel 1, 2, 3, 4 : Per sincronizzare la vista sul canale ...

Per sincronizzare Esploratori progetto e Visualizzatori eventi in altri layout, cambia layout e ripeti i passaggi 1-2

-Using Meter Instances to Sync Meters Across Layouts

Sono disponibili tre diverse visualizzazioni dedicate alla misurazione: Audio Device Meter, Loudness Meter, Meter..
È possibile aprire simultaneamente fino a quattro istanze (A, B, C o D) di ciascuna di queste visualizzazioni all'interno di un layout. Ciascuna vista dello stesso tipo all'interno di un layout deve essere impostata su un'istanza diversa

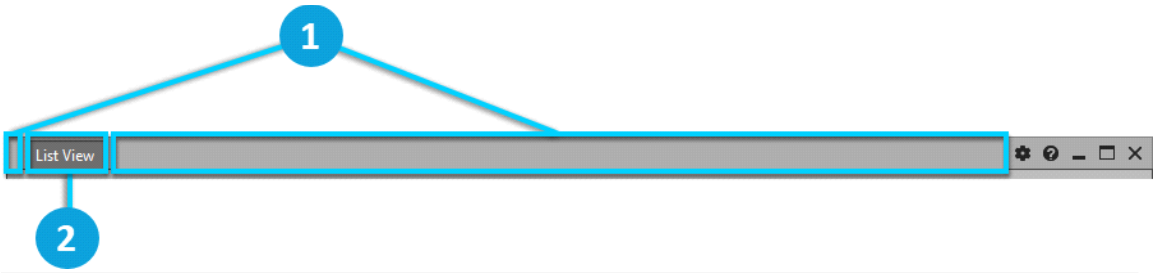
Il bus o Audio Device, selezionato per una particolare istanza viene ripristinato anche la prossima volta che apri lo stesso progetto Wwise.

-Working with Layouts

Puoi utilizzare layout diversi a seconda del lavoro che stai svolgendo.

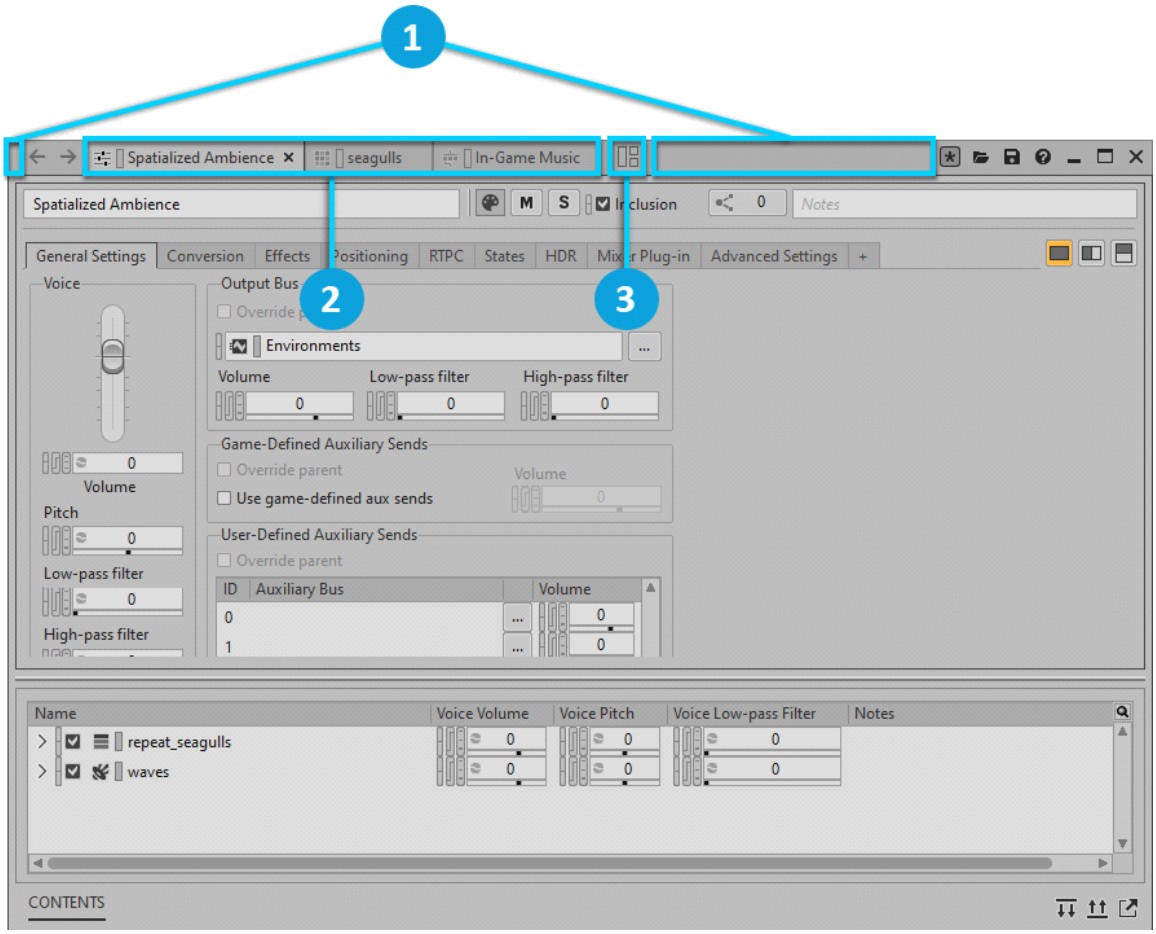
- **Designer:** Utilizzato per gestire, creare e definire le risorse che compongono le parti di effetti sonori, dialoghi e movimento del gioco.
- **Profiler:** Utilizzato per monitorare e analizzare le prestazioni dell'audio del gioco e degli elementi di movimento nel momento in cui si verificano.
- **SoundBank:** Utilizzato per creare, gestire e generare SoundBank per il tuo progetto
- **Mixer:** Utilizzato per creare prototipi e mescolare i vari oggetti in Wwise.
- **Schematic:** Utilizzato per esaminare una panoramica grafica della struttura di un progetto.
- **Audio Object Profiler:** Utilizzato per monitorare e analizzare lo stato degli oggetti audio mentre avanzano attraverso la pipeline del bus.
- **Voice Profiler:** Utilizzato per monitorare e analizzare le singole voci nel gioco.
- **Game Object Profiler:** Utilizzato per monitorare e analizzare le prestazioni audio dal punto di vista dell'oggetto del gioco.

-Docking/Undocking Views and Object Tabs
Sebbene i layout predefiniti siano stati progettati per rendere il flusso di lavoro in Wwise il più efficiente possibile, potrebbero esserci circostanze in cui desideri modificare il contenuto e il posizionamento delle visualizzazioni e dei gruppi di schede oggetto all'interno di un layout.



1	Per spostare una vista senza ancorarla, fare clic su un'area vuota sulla barra del titolo e trascinare.
2	Per avviare l'ancoraggio di una vista, fare clic sulla scheda del titolo della vista e trascinare.

È possibile spostare o avviare l'ancoraggio di un gruppo di schede di oggetti mobile nel modo seguente:



1	Per spostare un gruppo di schede Oggetto senza ancorarlo, fare clic su un'area vuota sulla barra del titolo e trascinare.
2	Per avviare l'ancoraggio di una singola scheda, fare clic su una delle schede Oggetto in un gruppo di schede oggetto e trascinare.
3	Per avviare l'ancoraggio di un gruppo di schede di oggetti nel suo complesso, fare clic sull'icona e trascinare.

-Restoring the Factory Layouts to Their Default Configuration

Dopo aver modificato un layout, potresti decidere di tornare alla configurazione del layout predefinita. Wwise ti consente di ripristinare

Fai **Layouts > Reset Factory Layouts.**

-Setting User Preferences SHIFT + U

Enabling Confirmation Messages
I messaggi di conferma possono fornire informazioni utili quando si eseguono determinate attività in Wwise. Mentre lavori con Wwise, puoi disabilitare questi messaggi uno alla volta selezionando l'opzione "Non chiedere più" in alcune finestre di dialogo di conferma.

Project > User Preferences, **Confirmation Messages, clicca Reset**.

D'ora in poi appariranno messaggi di conferma prima che determinate azioni vengano eseguite in Wwise

Selecting External Audio Editors

In Wwise, puoi modificare i file audio direttamente in qualsiasi editor di file audio di tua scelta. Prima però è necessario aggiungere ciascun programma all'elenco degli editor disponibili.

Project > User Preferences, **External Editors, Add.**

Selezionare l'eseguibile dell'editor audio (EXE) e fare clic su Apri.

Per selezionare l'editor audio esterno predefinito:
Nell'elenco degli editor, seleziona quello che desideri rendere predefinito.
Fare clic su Set As Default. (Questo pulsante non sarà attivo se non ci sono almeno due editor elencati.)

Selecting the Preferred User Interface Font and Scaling

Per impostazione predefinita, il carattere dell'interfaccia utente Wwise è Segoe UI, la dimensione del carattere è 9pt e DPI elevato è abilitato

Per modificare queste impostazioni: **Project > User Preferences**.
User Interface - Font and Scaling

Impostare Abilita DPI alto come preferito.
Enable High DPI

Setting the Output Buffer Latency

Tuttavia, puoi regolare il numero di buffer utilizzati dal motore audio per correggere alcuni problemi di prestazioni che potresti riscontrare durante l'utilizzo di Wwise.

Audio > Authoring Audio Preferences
Nella casella di gruppo Sound Engine, selezionare un numero di buffer di output dall'elenco. Il numero tra parentesi rappresenta la latenza corrispondente.

Setting the Preferred Audio Devices

Per impostazione predefinita, tutto l'audio riprodotto in Wwise viene riprodotto sul dispositivo di riproduzione predefinito del tuo sistema.Puoi modificare il dispositivo di output direttamente in Wwise o modificare quale dispositivo audio ShareSet viene utilizzato per l'ascolto all'interno dell'applicazione di creazione.

Audio > Authoring Audio Preferences. Device Shareset and Hardware Device,

Setting the Audio Channel Configuration

System Default Channel Configuration. **Audio > System Default Channel Configuration**.
Per impostazione predefinita, Wwise utilizza la configurazione dell'impostazione degli altoparlanti dal Pannello di controllo di Windows. Selezionare questa opzione per scegliere il valore selezionato nel Pannello di controllo di Windows.

Stereo Channel Configuration (Speakers),
Stereo Channel Configuration (Headphones),
Channel Configuration (5.1)
Channel Configuration (7.1)

È possibile selezionare la configurazione dei canali 5.1 o 7.1 mentre il Pannello di controllo di Windows è impostato su stereo. Tuttavia, tieni presente che ciò probabilmente costringerà il sistema audio a eseguire il downmix da 5.1 o 7.1 a stereo.

Setting the Music Track Look-ahead Time

La riproduzione audio viene sempre trasmessa in streaming durante la creazione in Wwise. Per aumentare la flessibilità ed evitare problemi di sincronizzazione e di mancanza di voce durante la riproduzione, è possibile regolare il tempo di attesa della traccia musicale come preferenza dell'utente.

Audio > Authoring Audio Preferences.
Digitare la quantità di tempo di previsione in millisecondi nel campo, **Music Track look-ahead time (ms)**

-Speeding Up the Way You Work

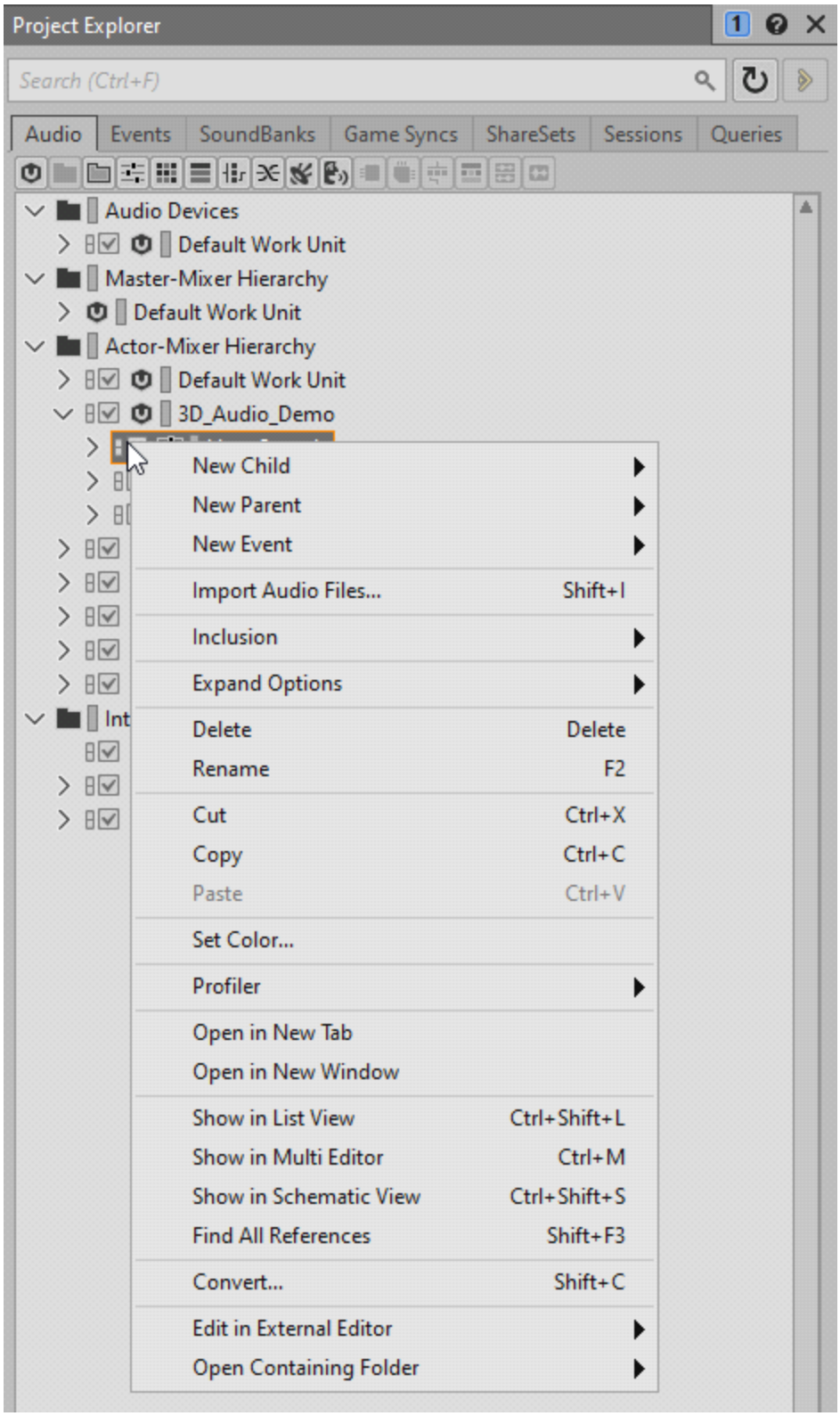
Navigazione con la tastiera

- Tab per passare da un controllo al successivo
- Shift+Tab per tornare al controllo precedente;
- Invio per attivare il controllo selezionato;
- Tasti freccia per selezionare le opzioni dell'elenco e attivare/disattivare i pulsanti di opzione.

Utilizzo delle scorciatoie da tastiera

- Ripristina il valore predefinito Ctrl+click
- Ottimizza il valore utilizzando il super slider Shift+drag
- Compensa i valori nella selezione multipla utilizzando il super slider Alt+drag

Utilizzo dei menu di scelta rapida



- Convert to Sound Voice… / Convert to Sound SFX…: Per cambiare l'oggetto da un oggetto Sound SFX a un oggetto Sound Voice o viceversa.
- Set Color Per aprire il menu colore in cui è possibile selezionare un colore da applicare al

chip colore dell'oggetto, per scopi organizzativi.

- Edit caricare l'oggetto o l'elemento del progetto selezionato nel rispettivo editor.
- Find in Project Explorer Per evidenziare l'oggetto o l'elemento del progetto selezionato nell'Esploratore progetto.
- Show in List View Per visualizzare gli oggetti selezionati nella Visualizzazione elenco.
- Show in Multi Editor Per visualizzare gli oggetti selezionati nel Multi Editor.
- Show in Schematic View Per visualizzare gli oggetti selezionati nella vista schematica.
- Find All References: Per visualizzare un elenco di elementi del progetto che contengono riferimenti diretti all'oggetto corrente.
- Edit in External Editor: Per aprire un elenco di editor esterni
- Edit in Nuendo : Per aprire l'elemento in Nuendo.
- Questa opzione è disponibile solo se hai installato il plug-in Nuendo Game Audio Connect e configurato correttamente le Impostazioni di connessione Nuendo... (aperte dal menu Progetto) e Nuendo è attualmente in esecuzione.
- Create in Nuendo : Per chiamare lo Steinberg Hub in cui è possibile creare un nuovo progetto in Nuendo. Un messaggio in Nuendo ti chiederà se vuoi importare il segmento trasferito in questo progetto.
- Open containing folder: Per richiedere a Esplora risorse (o Finder di Mac) la posizione dell'elemento principale dell'oggetto.

-Using the Batch Rename
La vista Rinomina batch consente agli utenti di rinominare oggetti modificabili o di modificare le note associate in un unico passaggio con un potente meccanismo di sostituzione, rimozione e inserimento simultaneo. Ciò elimina la necessità di attività lunghe, noiose o potenzialmente soggette a errori di rinominazione individuale degli oggetti.

-Opening the Batch Rename View

Per attivare la vista Rinomina batch senza oggetti:
Views > Batch Rename, Ctrl+F2.
La vista Rinomina batch si apre con un pannello Anteprima vuoto.

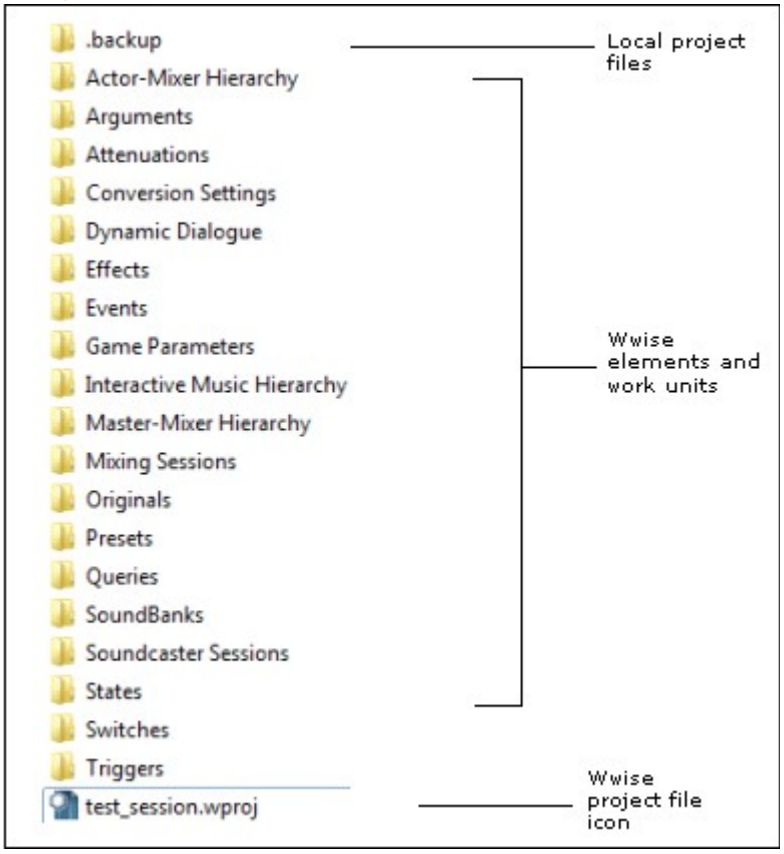
Per richiedere la ridenominazione batch popolata con gli oggetti selezionati:
Seleziona gli oggetti modificabili nell'Esplora progetto (o in un'altra visualizzazione oggetto come la visualizzazione elenco) su cui desideri eseguire le operazioni di rinomina.
Selezionare Rinomina batch dal menu di scelta rapida (clic con il pulsante destro del mouse) o utilizzare la scorciatoia da tastiera.
La vista Rinomina batch si apre con il pannello Anteprima popolato con gli oggetti selezionati.

Per aggiungere oggetti alla vista Rinomina batch:
Con Rinomina batch aperta, seleziona gli oggetti modificabili in Esplora progetto (o un'altra vista oggetti) su cui desideri eseguire le operazioni di rinomina.
Batch Rename o Ctrl +F2

-Setting Up Your Projects

--Working with Projects

---Overview
In Wwise, un progetto contiene tutte le tue risorse, nonché le proprietà e i comportamenti che imposti per le tue risorse per ogni piattaforma e linguaggio che stai sviluppando. Lo stesso progetto contiene anche elementi Wwise come eventi, preset, registri e simulazioni, nonché i SoundBank che genererai.
Devi utilizzare un solo progetto Wwise per gioco. Puoi, tuttavia, suddividere un progetto in Work Units,



La struttura delle cartelle del progetto Wwise è composta da varie cartelle contenenti file XML per i diversi elementi del progetto per facilitare la gestione delle versioni del progetto e di più utenti

Ognuna di queste cartelle avrà una sua **Work units**

-Defining your Project Settings CTRL + K

- General, definire un plug-in di controllo del codice sorgente, la posizione della cartella Originali per le risorse del progetto, le impostazioni di rilevamento automatico della frequenza di campionamento e le impostazioni di creazione del nome dell'evento.

- Conversion, mpostare le impostazioni di conversione ShareSet predefinite per il progetto nonché le impostazioni di rilevamento automatico della frequenza di campionamento.
- SoundBanks, per definire le impostazioni SoundBank per il tuo progetto.
- Logs, per gestire gli avvisi, gli errori e i messaggi visualizzati nei registri Conversione e

SoundBank.

- Obstruction/Occlusion, per definire il volume, le curve LPF e HPF per l'ostruzione e l'occlusione nel progetto.
- External Sources, per specificare i percorsi di ingresso e uscita per le sorgenti audio esterne che verranno utilizzate con il plug-in External Source.
- Network, per specificare i numeri di porta da utilizzare durante la comunicazione tra l'applicazione di creazione e il gioco.
- Custom Properties, per definire le proprietà per gli oggetti Suono e Sorgente audio della gerarchia Attore-Mixer.

Altre impostazioni del progetto si trovano nella finestra di dialogo Platform Manager:

Volume thresholds, Max voice instances

--General Settings

Configuring Source Control Plug-ins

Se utilizzi un sistema di controllo del codice sorgente per gestire le risorse e i file del progetto, puoi selezionare un plug-in di controllo del codice sorgente per il tuo progetto. Se il plug-in è supportato nella versione corrente di Wwise, puoi anche configurarlo per il tuo spazio di lavoro.

- Perforce®
- Subversion (deprecated)

Project, Project Settings, Shift+K., Plug-in, Source Control

Defining Originals Folder Settings

Contiene copie dei file audio originali che importi nel progetto. La sua posizione viene definita al momento della creazione del progetto. Nella scheda Generale è possibile specificare la posizione dei file originali.

Project, Project Settings, Original Audio Files

Defining Cache Folder Settings

La cartella cache del progetto contiene dati intermedi generati da Wwise durante la conversione del file audio e la generazione di SoundBank.

Project, Project Settings, Cached Audio Files

Defining Event Creation Settings

Gli eventi in Wwise possono essere creati in moltissimi modi. Se un Evento viene creato senza un oggetto target, Wwise darà al nuovo Evento un nome generico. Se invece l'Evento viene creato per un oggetto selezionato, puoi specificare come Wwise costruisce il nome del nuovo evento.

Project, Project Settings, Event Creation, Add action name (se abilitato, il nome dell'azione dell'evento viene aggiunto al nome dell'eventor), Modify case (se abilitato, il nome dell'evento viene modificato in maiuscole e minuscole)

--Source Settings

Defining the Conversion Settings for Your Project

Dopo aver creato gli ShareSet delle impostazioni di conversione per il tuo progetto, puoi specificare quale desideri utilizzare come predefinito.

Project, Project Settings, Source Settings, Default Conversion Settings.

Defining the Sample Rate Automatic Detection Settings

Decidere la migliore frequenza di campionamento per i tuoi file multimediali può essere difficile e richiedere molto tempo. Per accelerare il processo, puoi chiedere a Wwise di eseguire un'analisi di ciascun file utilizzando l'algoritmo Fast Fourier Transform (FFT).

Project, Project Settings, Sample Rate Automatic Detection, Volume Thresholds

- Low quality - Un livello di volume di taglio che identifica la frequenza utilizzata per determinare la migliore frequenza di campionamento alla quale convertire i file.
- Medium quality - Un livello di volume di taglio che identifica la frequenza utilizzata per determinare la migliore frequenza di campionamento alla quale convertire i file.
- High quality - Un livello di volume di taglio che identifica la frequenza utilizzata per determinare la migliore frequenza di campionamento alla quale convertire i file.

--Sounbank Settings

-Defining the SoundBank Settings for Your Project

- Defining SoundBank Project Settings, Determina quali informazioni devono essere incluse nei SoundBank generati.
- Specifying a Location for Your Saved SoundBank, Determina la posizione sul tuo disco rigido o sulla rete in cui salvare un SoundBank.
- Defining Steps to be Performed Pre/Post SoundBank Generation, Determina quali attività verranno eseguite immediatamente prima della generazione di SoundBank.
- Defining Steps to be Performed Pre/Post SoundBank Generation, Determina quali attività verranno eseguite immediatamente dopo la generazione di SoundBank.

Defining SoundBank Project Settings

Prima di generare i tuoi SoundBank, devi determinare quali informazioni faranno parte del processo di generazione, come verranno incluse e in quale formato verranno generate.

Project, Project Settings, SHIFT + K, Passa alla scheda SoundBank.

- Allow SoundBanks to exceed maximum size, per generare SoundBank anche se superano la dimensione massima specificata.
- Generate SoundBank content files, per creare file che elencano il contenuto di ciascun

SoundBank. I file di contenuto includono informazioni su eventi, bus, stati e interruttori, nonché un elenco completo di file audio in streaming e in memoria

- Generate header file, per creare un file di intestazione che associa i nomi di eventi, stato, interruttore e parametri di gioco agli ID.
- Max attenuation, per includere le informazioni sull'attenuazione massima nel file SoundbanksInfo.xml per ciascun evento.
- Estimated duration, per includere la durata massima e minima stimata per ciascun Evento, nonché se un suono si ripete all'infinito o è un suono unico, nel file SoundbanksInfo.xml per ciascun Evento.
- Use SoundBank Names, per utilizzare i nomi SoundBank (selezionato) o gli ID (non selezionati) per denominare i file SoundBank generati (BNK) e per fare riferimento a un banco all'interno di un altro banco.

Specifying a Location for Your Saved SoundBank

ProjectName\GeneratedSoundBanks\Platform\

Se questa posizione non è comoda per te, puoi cambiarla in qualsiasi directory sulla tua workstation o rete.

Quando specifichi la posizione del tuo SoundBank salvato, puoi utilizzare un percorso completo o relativo. Quando si utilizza un percorso relativo, utilizzare la cartella del progetto come origine del percorso. Ad esempio, il percorso completo e il percorso relativo seguenti specificano la stessa posizione

C:\Wwise Projects\My Project\GeneratedSoundBanks\Windows

GeneratedSoundBanks\Windows\

Project, Project Settings, SoundBank Paths

Defining Steps to be Performed Pre/Post SoundBank Generation

A seconda del flusso di lavoro, potresti avere un determinato passaggio o attività da eseguire immediatamente prima o immediatamente dopo la generazione dei tuoi SoundBank.

In Wwise, questi tipi di attività vengono definiti creando righe di comando. All'interno di Wwise esiste uno speciale editor della riga di comando che ti consente di creare facilmente tutte le righe di comando di cui hai bisogno.
Le variabili Wwise specifiche disponibili per scrivere righe di comando personalizzate sono le seguenti:

- $(AllowExceedMaximum), Specifica se i banchi di suoni possono essere generati anche se superano la dimensione massima specificata.
- $(GenerateContentFile), Specifica se vengono creati i file che elencano il contenuto di ciascun banco audio.
- $(GenerateContentFile), Specifica se viene generato un file di intestazione che esegue il

mapping dei nomi di eventi, stato, switch e parametri di gioco agli ID.

- $(GenerateMaxAttenuationInfo), Specifica se le informazioni sull'attenuazione massima vengono generate per gli eventi.
- $(GenerateEstimatedDuration), Specifica se la durata massima e minima stimata e le informazioni sul tipo di durata vengono generate per gli eventi.
- $(HeaderFileFullFilePath), Percorso completo del file di intestazione, ovvero: $ (HeaderFilePath)\Wwise_IDs.h
- $(HeaderFilePath), Il percorso o il percorso in cui verrà salvato il file di intestazione.
- $(InfoFilePath), Il nome completo del file Info della piattaforma corrente.
- $(IsRunningFromCmdLine), Specifica se Wwise è stato avviato dalla riga di comando con il flag.-generatesoundbanks
- $(LanguageList), Il nome della piattaforma corrente.
- $(Platform), Il nome della piattaforma corrente.
- $(SoundBankList), L'elenco dei banchi di suoni passati alla riga di comando o ai banchi di suoni selezionati in Gestione banchi di suoni.
- $(SoundBankListAsTextFile), Un file di testo contenente l'elenco dei banchi di suoni passati alla riga di comando o ai banchi di suoni selezionati in Gestione banchi di suoni.
- $(SoundBankPath), Il percorso o la posizione in cui vengono salvati i SoundBank della piattaforma corrente.//
- $(UseSoundBankNames)Specifica se i nomi () o gli ID () dei SoundBank vengono utilizzati per assegnare un nome ai file SoundBank (BNK) generati, nonché all'interno dei SoundBank per fare riferimento ai file multimediali in altri banchi.
- $(WwiseExeDriveLetter), La lettera dell'unità sulla workstation in cui si trova l'eseguibile Wwise (Wwise.exe).
- $(WwiseExePath), Il percorso o la posizione dell'eseguibile Wwise (Wwise.exe).
- $(WwiseExeProcessID), L'identificatore numerico del processo del Eseguibile Wwise (Wwise.exe).
- $(WwiseProjectDriveLetter), La lettera dell'unità sulla workstation in cui si trova il progetto Wwise.
- $(WwiseProjectName), Nome del progetto corrente.
- $(WwiseProjectPath), Il percorso o la posizione del progetto Wwise.
- $(WaapiWampPort), La porta utilizzata da WAAPI per il protocollo WAMP.
- $(WaapiHttpPort), La porta utilizzata da WAAPI per il protocollo HTTP.

Per essere il più flessibile possibile, Wwise consente di definire diverse righe di comando per i seguenti tipi di passaggi:

- Global opening step, Una riga di comando che si applica a tutte le piattaforme e viene eseguita prima di qualsiasi altro passaggio.
- Platform-specific pre-generation step, Una riga di comando che si applica a una piattaforma specifica e viene eseguita prima che vengano generati i SoundBank.
- Platform-specific post-generation step, Una riga di comando che si applica a una piattaforma specifica e viene eseguita dopo la generazione dei SoundBank.
- Global closing step, Una riga di comando che si applica a tutte le piattaforme e viene

eseguita dopo tutti gli altri passaggi.

È possibile, tuttavia, automatizzare qualsiasi tipo di attività eseguendo una riga di comando diversa.
È inoltre possibile salvare le righe di comando create in un file (WCMDLINE) in modo da poterle utilizzare in un secondo momento, all'interno dello stesso progetto, tra progetti o se si desidera condividerle con altri utenti.

Per definire le attività da eseguire prima della generazione del SoundBank:

Project, Project Settings, SoundBanks tab, global, per crear un global Pre-Generation Step
fare clic sul pulsante browse […], per aprire il Global Opening Step Editor.
Pre-Generation Step Editor si aprirà.
Description text box, digitare un nome che descriva chiaramente il passaggio o l'attività che verrà eseguita,
Commands, crea il tuo command line.

Nella casella di gruppo Macro, selezionare una delle seguenti opzioni:

- Built-in Macros, Per visualizzare un elenco di variabili specifiche di Wwise che possono essere utilizzate nelle righe di comando di Wwise
- Environment Variables, Per visualizzare un elenco di variabili di ambiente specifiche di Windows che possono essere utilizzate nelle righe di comando di Wwise.

Per definire le attività da eseguire dopo la SoundBank generation:

Project, Project Settings, Shift+K, Post-Generation Step,
fare clic sul pulsante browse […], per aprire il Global Opening Step Editor.
Pre-Generation Step Editor si aprirà.
Description text box, digitare un nome che descriva chiaramente il passaggio o l'attività che verrà eseguita,
Commands, crea il tuo command line.

Nella casella di gruppo Macro, selezionare una delle seguenti opzioni:

- Built-in Macros, Per visualizzare un elenco di variabili specifiche di Wwise che possono essere utilizzate nelle righe di comando di Wwise
- Environment Variables, Per visualizzare un elenco di variabili di ambiente specifiche di Windows che possono essere utilizzate nelle righe di comando di Wwise.

<u>-Loading Factory/Custom Command Lines</u>

Wwise viene fornito con alcune righe di comando già create, inclusa una che copia i file in streaming nella directory SoundBank e un'altra che raggruppa i file in streaming e SoundBank in un pacchetto.
Queste sono chiamate righe di comando di fabbrica. È possibile caricare queste righe di

comando di fabbrica o righe di comando personalizzate precedentemente salvate su file.

Per caricare una riga di comando predefinita/personalizzata:

Dal Editor dei passaggi di pre/post generazione, clicca Load.

Dallo shortcut, clicca

- **From Factory Folder,** Per aprire Explorer/Finder nella directory in cui si trovano le righe di comando di Wwise Factory.
- **From Last Location,** Per aprire Explorer/Finder nella directory da cui è stata caricata l'ultima volta una riga di comando

Seleziona il command line che desideri caricare e fare clic Open**.**

-Saving Custom Command Lines to a File

Puoi salvare le righe di comando personalizzate che crei in un file in modo da poterle utilizzare successivamente all'interno dello stesso progetto, tra progetti o se desideri condividerle con altri utenti.

Nel editor di Pre Post Generation Step Editor, clicca save as

La riga di comando viene salvata come file WCMDLINE e ora può essere riutilizzata in qualsiasi momento.

--Log Settings

-Managing Messages that Appear in the Logs

Potrebbero esserci situazioni in cui non desideri o non hai più bisogno che avvisi e/o messaggi specifici vengano visualizzati nei registri di SoundBank o di conversione durante il processo di generazione.
Per evitare che questi tipi di avvisi e messaggi vengano visualizzati nei registri di SoundBank e di conversione, è possibile aggiungerli all'elenco Ignora registro.

Project > Project Settings, Logs tab, Deseleziona le caselle di controllo di tutti i messaggi che desideri aggiungere all'elenco Log Ignora.

Per modificare la gravità di un messaggio:

Project > Project Settings, Severity

Per limitare il numero di messaggi che verranno visualizzati nei registri:

Project > Project Settings, Abilita il, Limit number of messages displayed to, opzione e digitare

il numero massimo di messaggi che si desidera visualizzare nel registro.

--Obstruction/Occlusion

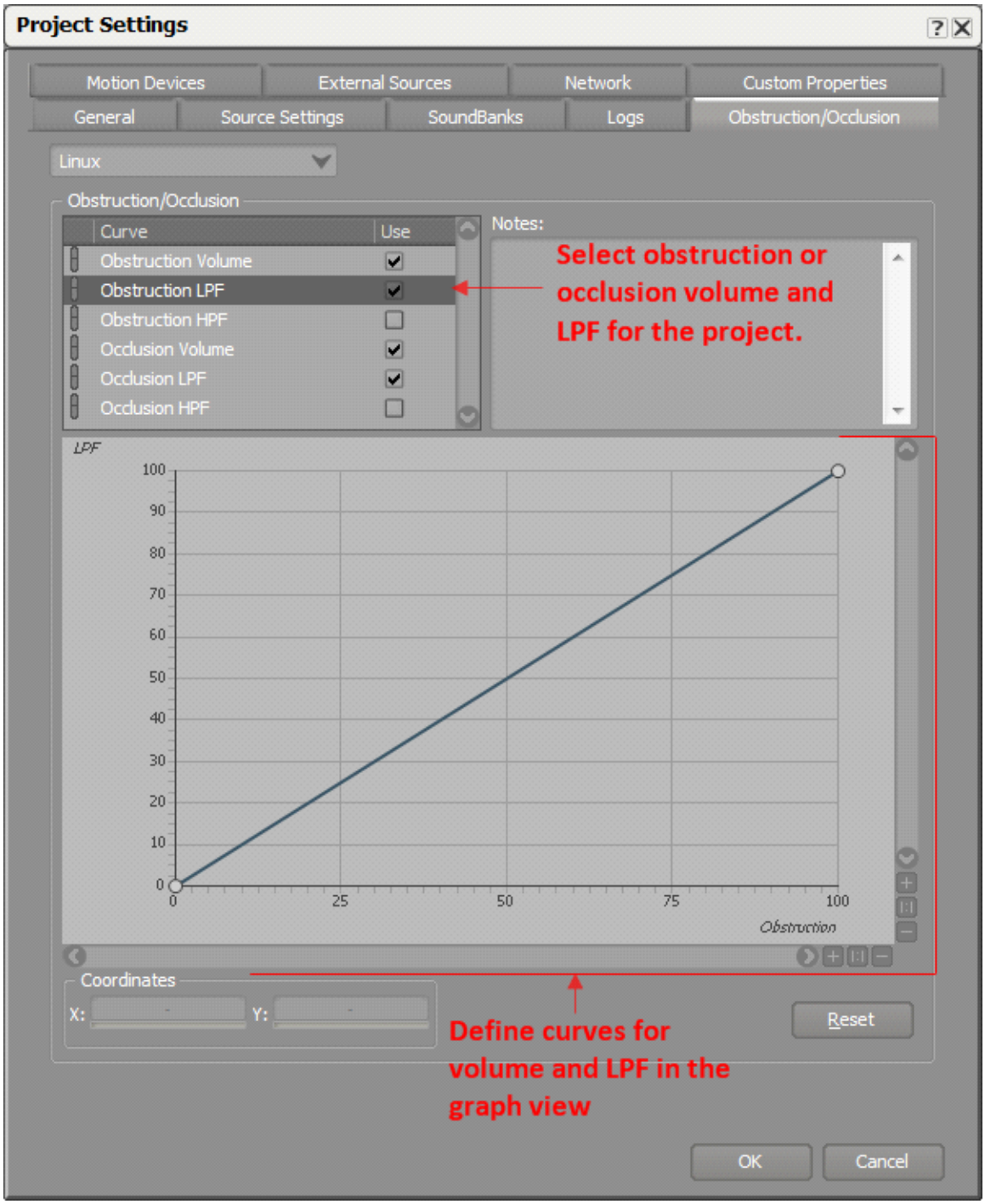
-Defining Obstruction and Occlusion Curves for Your Project

Nella scheda Ostruzione/Occlusione della finestra di dialogo Impostazioni progetto Wwise, puoi definire le impostazioni di ostruzione e occlusione per gli oggetti sonori per ciascuna piattaforma nel tuo progetto. L'ostruzione si verifica quando un oggetto nella geometria del gioco, come un muro o un pilastro, blocca parzialmente lo spazio tra una sorgente sonora e un ascoltatore.

Gli sviluppatori di giochi definiscono a livello di codice la geometria del gioco in cui possono verificarsi le condizioni per l'ostruzione e l'occlusione. Wwise non calcola da solo i livelli di ostruzione e occlusione.

I calcoli fisici devono essere eseguiti dal gioco e i risultati devono essere passati alla seguente funzione: SetObjectObstructionAndOcclusion() .

Wwise ti consente di definire curve di volume, LPF e HPF specifiche della piattaforma che verranno applicate agli oggetti sonori poiché sono influenzati da ostruzione e occlusione durante il gioco.



Per creare curve di ostruzione e occlusione più dettagliate e complesse, è possibile definire la forma di ciascun segmento della curva. Un segmento di curva è qualsiasi parte della curva tra due punti di controllo

Project > Project Settings, Dall'elenco Piattaforma, seleziona la piattaforma per la quale stai definendo le curve per le impostazioni di ostruzione e occlusioni.

Per specificare impostazioni diverse per il volume di ostruzione e/o occlusione, LPF e HPF sulla piattaforma selezionata, fare clic con il pulsante destro del mouse sull'indicatore del collegamento e nel menu di scelta rapida selezionare

Per definire il volume, le curve LPF e HPF per le impostazioni di ostruzione e occlusione per gli oggetti di gioco, procedi come segue:

- Abilitare la casella di controllo Use per evidenziare la curva corrispondente.
- Per creare punti sulla curva, fare doppio clic su un punto sulla curva.
- Per eliminare un punto sulla curva, selezionare il punto e premere Delete.
- Per eliminare tutti i punti definiti finora e ricominciare da capo, fare clic su Reset.

--External Source

Specifying the Input/Output Locations for External Sources

Se intendi utilizzare il plug-in Origine esterna, devi specificare il percorso del file Elenco origini esterne.

Project > Project Settings,

Input Path: specificare una cartella in cui si trova il file dell'elenco delle fonti esterne (WSOURCES) effettuando una delle seguenti operazioni:

- Fare clic su External Sources List e digitare il percorso in cui si trova il file External Sources
- Fai clic sul pulsante browse […] che corrisponde a una delle piattaforme di gioco, vai alla cartella che contiene il file dell'elenco delle fonti esterne per quella piattaforma, quindi fai clic su Apri..

Output Path: specificare una cartella in cui verranno salvate le fonti esterne convertite eseguendo una delle operazioni :

- Fare clic su External Sources Output Folder elenco e digitare il percorso in cui si desidera salvare i file audio.
- Fai clic sul pulsante browse […] che corrisponde a una delle piattaforme di gioco, vai alla cartella in cui desideri salvare i file audio convertiti per quella piattaforma, quindi fai clic su OK

--Network

Specifying Network Ports

Quando colleghi l'applicazione di creazione Wwise a un gioco per la profilazione e la modifica in tempo reale, la comunicazione tra Wwise e il gioco viene gestita da una rete utilizzando le porte di rete aperte per il gioco e l'applicazione di creazione. È possibile personalizzare le porte aperte e utilizzate dalle applicazioni di creazione nella scheda Rete della finestra di dialogo Impostazioni progetto Wwise. Le porte aperte/utilizzate dal gioco possono essere personalizzate durante l'inizializzazione del modulo di comunicazione Wwise da parte del gioco.

È possibile definire un numero di porta specifico (fisso) oppure consentire al sistema operativo di selezionarne uno automaticamente (dinamico).

Game Discovery Broadcast Port

Questa porta viene aperta dal modulo di comunicazione all'interno del gioco per ascoltare i messaggi trasmessi in rete dall'applicazione di creazione quando tenta di scoprire giochi a cui può connettersi.
L'applicazione di creazione trasmette i messaggi Game Discovery a quella porta, quindi è importante utilizzare la stessa porta nel gioco e nel tuo progetto Wwise. Poiché sia il gioco che l'applicazione di creazione devono conoscere questa porta, non può essere dinamica (non può essere impostata su 0).

Game Discovery Response Port

Questa porta viene aperta dall'applicazione di creazione Wwise. Qui è dove il gioco risponde al messaggio Game Discovery.

Customizing Communication Ports in the Game

Come accennato in precedenza, la porta di trasmissione Game Discovery viene aperta nel gioco dal modulo di comunicazione. Ci sono tre porte aggiuntive aperte nel gioco dal modulo di comunicazione e vengono personalizzate durante l'inizializzazione.

Project > Project Settings, Modifica i numeri di porta secondo necessità

--Platform manager

Defining the Volume Settings for Your Project

Nella finestra di dialogo Gestore piattaforma è possibile specificare le soglie di volume e il numero massimo di istanze vocali per ciascuna piattaforma.

La Soglia del volume si riferisce al punto al di sotto del quale una voce sarà gestita dai comportamenti definiti nelle Impostazioni avanzate del Property Editor. Puoi specificare una soglia di volume predefinita nel tuo progetto nella finestra di dialogo Platform Manager. Questo valore verrà utilizzato se la soglia del volume non è impostata specificatamente utilizzando l'API Wwise.

Project, Platform Manager, Shift+Alt+P

-Defining the Default Object Values for your Project

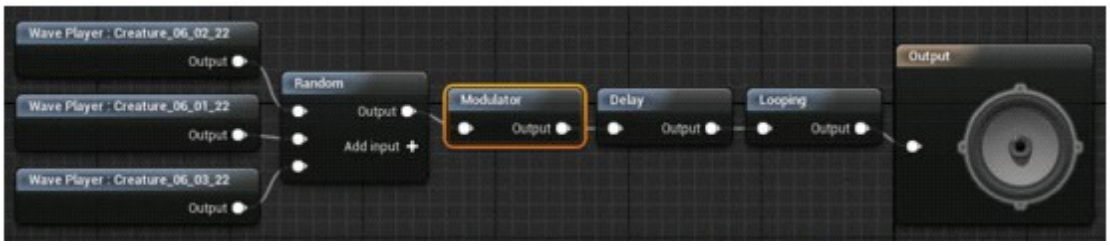
STOQUA

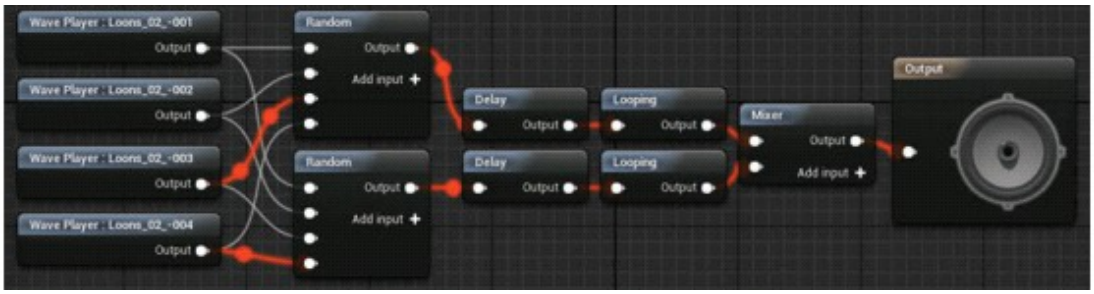
-

////////////

Box: Volume massimo all'interno delle estensioni, per poi scendere a Falloff Distance.
Capsule: Il volume massimo all'interno della capsula è definito da Metà altezza e Raggio, quindi scende a
Distanza di decadimento.
Cone: Volume massimo all'interno del cono interno come definito da Raggio del cono e Angolo del cono, quindi cadente
al cono definito dalla Distanza di Caduta e dall'Angolo di Caduta. Cone Off set ci consente di compensare l'inizio del
cono di ritorno dalla posizione dell'attore del suono ambientale.

Esempi di nodi





2. Procedural Sound Design

Quando interagiamo con gli oggetti nel mondo reale,
i processi fisici che producono il suono non si ripetono mai esattamente nello stesso modo.
//Prima di iniziare a lavorare su un gioco, devi essere pienamente informato sulle capacità della tua piattaforma
e l'approccio alla RAM e allo streaming che i programmatori hanno deciso di adottare (e idealmente voi
dovrebbe essere parte di questo dibattito).//

Il Sound design procedurale per sua natura implica un approccio più modulare alla progettazione del suono,
costruendo un evento sonoro finale a partire da elementi del suono anziché riprodurre un suono completo.
Quindi ricordando il teorema di Nyquist-Shannon (che tu è necessario campionare a una velocità almeno doppia rispetto alla frequenza più alta).

Per migliorare la qualità di compressione della WAVE, clicca la sound Wave, Details >> Quality

-Sequenced Variation
//Sommario Asynchronous loops, random timings, controlling random events for clusters of activity, concatenation for variation //

Riguarda il cambiamento delle relazioni temporali tra i suoni o dell'ordine dei suoni generare varietà.

--Time-based Variation
.Asynchronous Loops
Una cosa che può spezzare tutto questo
Il senso della ripetizione è avere cicli simultanei di lunghezze diverse in modo da creare variazioni
entrano ed escono in fase l'uno dall'altro.
Due suoni di lunghezze diverse in Loop.
.Asynchronous Loops through Pitch Shift
Due suoni di lunghezze diverse, in loop, modulati nel Pitch, con il nodo Modulator
.Random Time:Blocks
Suoni modulati e ritardati.
.Random Time: Fragments
L'uso di singoli elementi one-shot è molto più efficace dei lunghi cicli nel risparmiare memoria e fornendo la variazione per rendere il tuo mondo di gioco reale e vivo.
.Controlling Random Events: Clusters of Activity
Utilizzando il nodo concatenator (Crea una sequenza),ritardando e modulando i suoni.

2.Concatenation
Concatenazione pone i suoni immediatamente contro ciascuno
altro nel tempo, ricreando il senso di un suono continuo

Molti tipi di suono possono essere modificati con successo in singoli pezzi (facendo attenzione a dividerli a zero punti di incrocio).
Quindi questi pezzi possono essere concatenati o ricuciti insieme ma in modo casuale.

-Concatenation for Variation in Loops
Dividere un sample in blocchi e vi abbiamo applicare una varietà di modulazioni e ritardi ogni pezzo per dare l'impressione che il sistema stia collassando.
Il nodo -Concatenatore- lo farà
riproduci ciascuno dei suoi input in sequenza e così ricucirà insieme la nostra frase nel giusto ordine.

-Layered Concatenation
La concatenazione diventa davvero unica quando si sovrappongono un paio di sistemi audio poiché non lo siamo
ottenendo solo l'ordine casuale.
Strati di elementi in loop randomizzati, esemp Primo strato, Secondo strato...

-Concatenation for a Chain of Events
I normali passi a passo d'uomo sono maturi per la concatenazione poiché sono naturali
sono costituiti da due parti: il tallone e la punta.

Modificandoli in elementi separati, possiamo quindi ricombinarli casualmente utilizzando il file -Concatena- nodo

-Concatenate Sample Rates
A volte potresti ritrovarti a progettare l'audio per giochi che hanno budget di memoria estremamente bassi, forse per un giocattolo o un gioco portatile.
Dividere il file, mantenendo un'elevata frequenza di campionamento per un determinato file ma poi scegliendo una frequenza di campionamento più bassa
(e di conseguenza una dimensione del file più piccola) per un altro.
Es
{Cave_Cannon_All}— The original fi le at 44.1kHz = 234kb
{Cave_Cannon_Trans}— The edited crack at 44.1kHz = 44kb
{Cave_Cannon_Tail}— The edited rumble at 16kHz = 70kb

3.Multiple Start Points
Puoi riutilizzare parti di un file più lungo senza la necessità di modificarlo poiché possiamo iniziare a riprodurre il file in qualsiasi punto.
Con il Bp, in Play Audio Component, lo START TIME.

4.Layered Variation
Variare gli strati del suono elementi che si sentono, ad esempio scegliendo casualmente uno dei tre suoni possibili per ciascuno elemento.
Esempio:
Sound 1
Sound 1
Sound 2
Sound 3
Sound 1 + Sound 2
Sound 1 + Sound 3
Sound 2 + Sound 3
Sound 1 + Sound 2 + Sound 3

5.Simple Random Combinations
Combina casualmente uno di ciascuno dei seguenti elementi

Suono1 X 3
Suono2 X 3
Suono3 X 3

6.Random Combinations with Individualized Pitch and Delays
Sound Cue aggiunge ulteriori variazioni agli elementi combinati l'uso di un -Modulator- che randomizza l'intonazione e il volume e attraverso un -Delay.

Puoi vedere come ora stiamo iniziando a pensare ai nostri suoni non come onde sonore uniche ma come
sistemi audio modulari che sovrappongono i singoli componenti dell'evento.

7.Deconstructing a Sound for Procedural Use

Questo approccio modulare richiede che tu inizi davvero a pensare ai singoli elementi che lo compongono sono presenti all'interno di un evento sonoro finale
per decostruirlo nelle sue diverse modalità di produzione del suono componenti.

8.Scalability
L'opzione Preselect at Level Load ci consente di impostare il numero di ingressi su a -Random- dovrebbe essere attivo.
Quando il livello viene caricato, sceglierà casualmente il tuo designato numero di ingressi e, soprattutto, non caricherà quelli che non sono stati selezionati.

9.Modulation/Modifiers for Variation
Utilizza qualsiasi DSP (elaborazione del segnale digitale, pitch, cambio di volume) disponibile per creare variazioni sul
suoni che hai piuttosto che semplicemente mettere più suoni in memoria.

9.1 Pitch
Dalla semplice variazione al cambiamento radicale e agli inviluppi, variare l'altezza di un suono è uno dei modi più efficaci metodi più semplici ed efficaci
per generare variazioni nei tuoi suoni.

9.1.1 Reuse Sounds for Diff erent Sized Sources
Nella produzione fisica del suono, l'altezza è correlata alla dimensione dell'oggetto. Il tono in questo caso è generalmente meglio modulato
al centro (ovvero da 0,8 min a 1,2 max)
Utilizzare lo stesso suono, cambiando il tono intorno in un Range di 0.8 min a 1.2 max

9.2 Pitch Envelopes

9.3 Volume
Emettere un suono che ha diversi envelope diversi, ognuno di questi selezionati randomicamente.

9.4 Looping Volume Envelopes

9.5 Oscillation
Un oscillatore produce una variabile in continuo cambiamento attorno ad un valore centrale.

9.5.1 Volume Oscillation

9.5.2 Pitch Oscillation

9.5.3 Pitch + Volume Oscillation

9.6 Wavetables
Il tono campione originale è stato generato a 220 Hz (altezza LA3). Per ottenere le altre note, le loro Pitch Multipliers saranno scalate di conseguenza

10. Parameterization = Interactivity

Riepilogo: utilizzo delle condizioni e dei parametri di gioco per influenzare i cambiamenti nel suono.

10.1 Swapping out Sounds with -Branch-
Utilizzando il nodo BRANCH, puoi settare il valore Booleano con Set Boolean Parameter

10.2 Parameters to Pitch with <Set Pitch Multiplier>
Utilizzando le Timeline

10.3 Parameters to Volume with <Set Volume Multiplier>

10.4 Parameter-based Crossfades with -Crossfade by Param-

10.5 Using User-defined Curves
La capacità di prendere parametri o variabili dal sistema di gioco e mapparli su quelli definiti dall'utente curve è una tecnica fondamentale

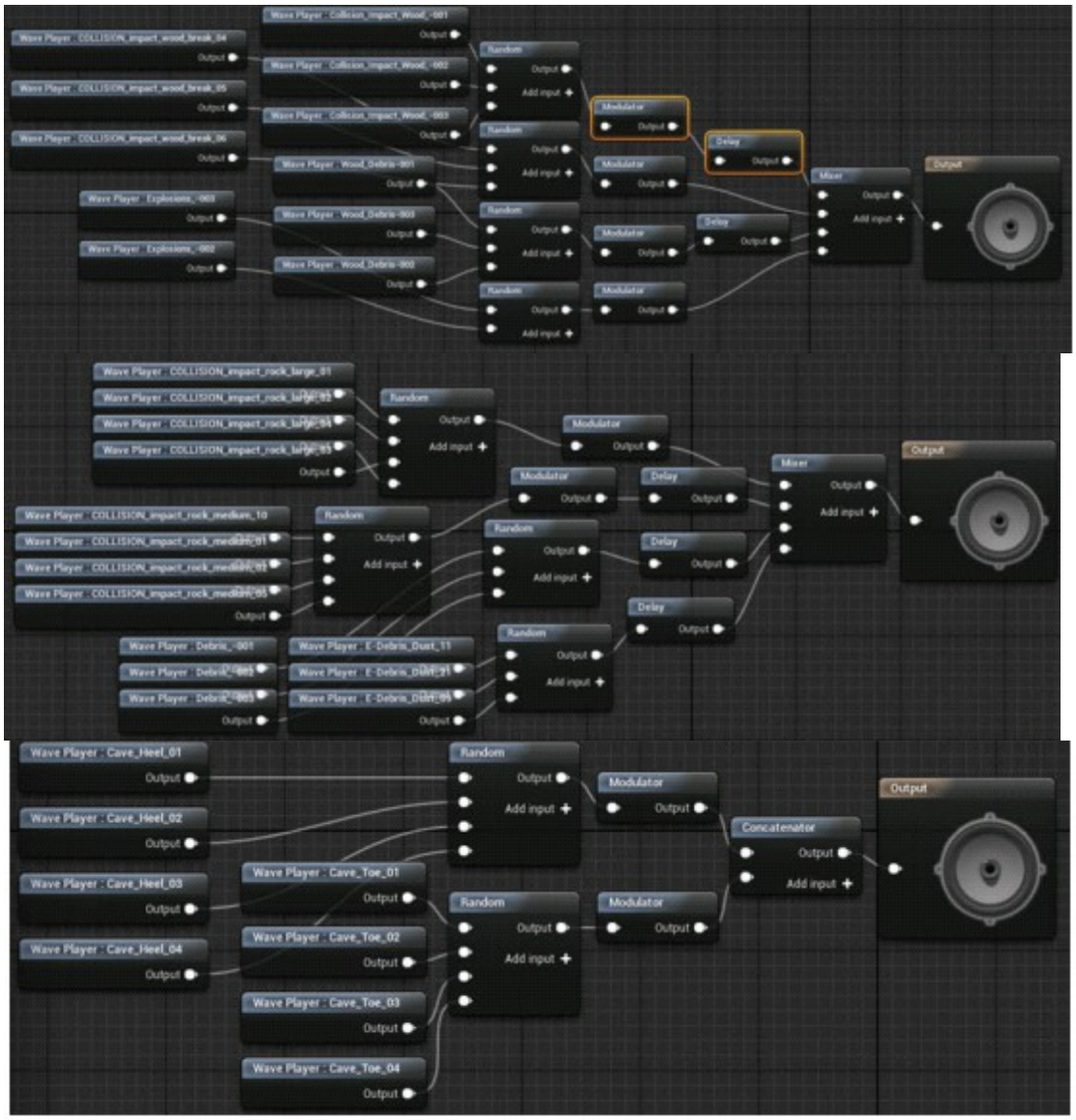
11. Asset Checking
Fare clic con il pulsante destro del mouse sulla cartella audio nel browser dei contenuti e selezionare Size Map per ottenere una rappresentazione grafica delle dimensioni dei file.

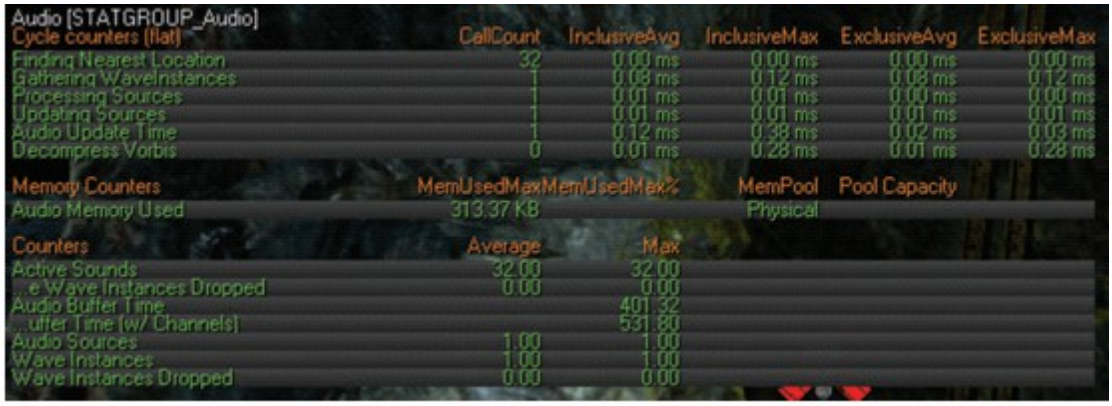
12. Memory Checking
Viewport Options, Stat, Shift + L
Sotto, su STAT >> ADVANCED >> AUDIO

13. Sound Streaming
Ciascuna {onda sonora} dispone di un'opzione aggiuntiva per consentire lo streaming e la priorità quell'onda
Edit > Project Settings > Platforms > Windows > Audio > Enable Use Stream Caching (Experimental).

14. Conclusion
Considerando i suoni come il prodotto di componenti separati con un approccio modulare, puoi iniziare a considerare la tua
raccolta di onde sonore, o la tua banca sonora, come una tavolozza da cui puoi prendere e riutilizzare elementi in diverse combinazioni per cose diverse.

RECAP
- Scegliere le frequenze di campionamento e le impostazioni di compressione dei file audio appropriate;
- Utilizzare tecniche di variazione sequenziate e stratificate nei Sound Cues;
- Utilizzare inviluppi e oscillatori per la variazione di intonazione e volume;
- Creare sistemi audio parametrizzati utilizzando rami condizionali (variabili booleane True/False);
- Controllare il volume e il tono dei componenti audio con parametri di gioco;
- Tieni traccia dell'utilizzo della memoria audio


Random
Modulator
Delay
Mixer
Output
Concatenator
Wave Player : Cave_Heel_01
Wave Player : Cave_Heel_02
Wave Player : Cave_Heel_03
Wave Player : Cave_Heel_04
Wave Player : Cave_Toe_01
Wave Player : Cave_Toe_02
Wave Player : Cave_Toe_03
Wave Player : Cave_Toe_04


Audio [STATGROUP_Audio]
Cycle counters (flat) CallCount InclusiveAvg InclusiveMax ExclusiveAvg ExclusiveMax
Finding Nearest Location 32 0.00 ms 0.00 ms 0.00 ms 0.00 ms
Gathering WaveInstances 1 0.03 ms 0.12 ms 0.03 ms 0.12 ms
Processing Sources 1 0.01 ms 0.01 ms 0.00 ms 0.00 ms
Updating Sources 1 0.01 ms 0.01 ms 0.01 ms 0.01 ms
Audio Update Time 1 0.12 ms 0.38 ms 0.02 ms 0.03 ms
Decompress Vorbis 0 0.01 ms 0.28 ms 0.01 ms 0.28 ms
Memory Counters MemUsedMax MemUsedMax% MemPool Pool Capacity
Audio Memory Used 313.37 KB Physical
Counters Average Max
Active Sounds 32.00 32.00
...e Wave Instances Dropped 0.00 0.00
Audio Buffer Time 401.32
...uffer Time (w/ Channels) 531.80
Audio Sources 1.00 1.00
Wave Instances 1.00 1.00
Wave Instances Dropped 0.00 0.00

////////////

22. Get Current Audio Output Device Name = Ottiene informazioni sul dispositivo di uscita audio attualmente in uso
23. Get Current Reverb Effect = Restituisce le impostazioni di riverbero con la priorità più alta attualmente attive da qualsiasi sorgente
24. Get Max Audio Channel Count = Recupera il numero massimo di voci attualmente utilizzato dal motore audio.
25. Get Sound Asset Compression Type = Restituisce il tipo di compressione dell'elemento audio. L'obiettivo è l'onda sonora
26. Load Control Bus Mix From Profile = Carica la combinazione del bus di controllo da un profilo nella definizione della miscela UObject
27. Play Dialogue 2D = Riproduce direttamente un dialogo senza attenuazione, perfetto per l'interfaccia utente.
28. Play Dialogue at Location = Riproduce un dialogo nella posizione specificata.
29. Play Sound 2D = Riproduce un suono direttamente senza attenuazione, perfetto per i suoni dell'interfaccia utente. UI
30. Play Sound at Location = Riproduce un suono nella posizione specificata
31. Pop Sound Mix Modifier = Inserire un modificatore di mix audio dal sistema audio (Sound class mix)
32. Prime All Sounds in Sound Class = Prepara/Imposta le onde sonore nell'oggetto USoundClass specificato
33. Prime Sound = Prepara l'audio, memorizzando nella cache il primo blocco dell'audio in streaming.
34. Push Sound Mix Modifier = Inserire un modificatore di mix audio nel sistema audio
35. Save Control Bus Mix to Profile = Salva il mix del bus di controllo in un profilo
36. Set Base Sound Mix = Impostare il mix audio del sistema audio per un equalizzatore speciale
37. Set Control Bus Mix = Imposta un mix del bus di controllo con i dati degli stage forniti, se gli stadi vengono forniti in un proxy di istanza attivo della combinazione.
38. Set Control Bus Mix By Filter = Imposta le fasi filtrate di una determinata classe su un valore di destinazione fornito per l'istanza attiva della combinazione.
39. Set Global Control Bus Mix Value = Imposta un mix di bus di controllo globale
40. Set Global Listener Focus Parameters = Imposta i parametri globali di messa a fuoco dell'ascoltatore
41. Set Global Pitch Modulation = Imposta uno scalare di modulazione dell'intonazione globale che si applicherà a tutti i suoni non dell'interfaccia utente
42. Set Max Audio Channels Scaled = Imposta il numero massimo di voci (note anche come "canali") in modo dinamico in percentuale.
Ad esempio, se si desidera Riduci il numero di voci del 50%, usa 0,50. Successivamente, è possibile tornare al numero massimo di voci originale usando 1.0.
43. Set Sound Asset Compression Type = Impostare proceduralmente il tipo di compressione. L'obiettivo è l'onda sonora
44. Set Sound Class Distance Scale = Imposta la scala di distanza della classe sonora
45. Set Sound Mix Class Override = Imposta il volume di uscita del submix in guadagno lineare.
46. SetSubmixDryLevel (linear gain) = Imposta il volume di uscita del submix in guadagno lineare.
47. Spawn Dialogue 2D = Genera dialoghi 2D
48. Spawn Dialogue at Location = Genera dialoghi sul postoGenera dialoghi sul posto
49. Spawn Dialogue Attached = Dialogo di generazione allegato
50. Spawn Sound 2D = Questa funzione consente agli utenti di creare componenti audio con impostazioni specifiche per componenti non spazializzati

51. Spawn Sound at Location = Genera un suono nella posizione specificata. Questo non viaggia con nessun attore.
52. Spawn Sound Attached = Questa funzione consente agli utenti di creare e riprodurre componenti audio collegati a uno specifico componente scena.
53. Swap Audio Output Device = Scambia dispositivo di uscita audio
54. Un Retain All Sounds in Sound Class = Conserva tutti i suoni nella classe del suono
55. Update Control Bus Mix = Aggionra il Control Bus Mix
56. Update Modulator = Aggiorna il modulator

-Analysis
57. Get Magnitude for Frequencies = Recupera le magnitudini per le frequenze date.
58. Get Phase for Frequencies = Recupera le fasi per le frequenze date.
59. Make Full Spectrum Spectral Analysis Band Settings = Crea una serie di bande distanziate logaritmicamente.
60. Make Musical Spectral Analysis Band Settings = Crea una serie di bande musicalmente distanziate con frequenza crescente.
61. Make Preset Spectral Analysis Band Settings = Creare una matrice di bande che si estendono sull'intervallo di frequenza di un determinato EAudioSpectrumBandPresetType.
62. Start Analyzing Output = Avviare l'analisi dello spettro dell'uscita audio.
63. Start Spectral Analysis = Avviare l'analisi dello spettro dell'uscita audio.
64. Stop Analyzing Output = Interrompere l'analisi dello spettro. Target è Audio Mixer
65. Stop Spectral Analysis = Interrompe l'analisi dello spettro dell'uscita audio.

-Bounce
66. Finish Recording Submix Output = Termina la registrazione dell'audio da questo submix ed esportalo come file wav o USoundWave.
67. Start Recording Submix Output = Avvia la registrazione dell'audio da questo submix.

-Bus
68. Is Audio Bus Active = Interroga se il bus audio specificato è attivo
69. Start Audio Bus = Avvia il bus audio specificato.
70. Stop Audio Bus = Stoppa il bus audio specificato.

-Cache
71. Prime Sound Cue for Playback = Iniziare a caricare nella cache tutti i suoni a cui fa riferimento un segnale sonoro in modo che possano essere riprodotti immediatamente.
72. Prime Sound for Playback = Inizia a caricare un suono nella cache in modo che possa essere riprodotto immediatamente..
73. Trim Audio Cache = Taglia la memoria utilizzata dalla cache audio.

-Components (Per le componenti audio)
AUDIO//ADD AUDIO COMPONENT IN CONSTRUCTION SCRIPT//RIFERIMENTO ALLA COMPONENTE AUDIO//IN CONSTRUCTION SCRIPT LA COMPONENTE AUDIO DOVRA' ESSERE SETTATA IN SET MODE//
74. Adjust Attenuation = Regola attenuazione
75. Adjust Volume = Regola il volume
76. Fade In = Dissolvenza in entrata //Overload synth component
77. Fade Out = Dissolvenza in entrata

78. Get Attenuation Settings To Apply = Ottenere le impostazioni di attenuazione da applicare Restituisce FALSO se non è stata trovata alcuna impostazione.
79. Get Cooked Amplitude Envelope Data = Recupera i dati dell'inviluppo dell'ampiezza cotta (cooked) al tempo di riproduzione corrente.
80. Get Cooked Amplitude Envelope Data For All Playing Sounds = Ottieni i dati dell'inviluppo dell'ampiezza cotta (cooked) per tutti i suoni di riproduzione
81. Get Cooked FFTData = Recupera i dati spettrali cotti (cooked) in tempo corrente dei suoni riprodotti sul componente audio.
82. Get Cooked FFTData for All Playing Sounds = Get Cooked FFTData per tutti i suoni di riproduzione
83. Get Play State = Ottenere lo stato di riproduzione
84. Has Cooked Amplitude Envelope Data = Verifica se il suono dell'istanza del componente audio di destinazione contiene o meno dati di inviluppo di ampiezza
85. Has Cooked FFTData = Interroga se l'onda sonora riprodotta in questo componente audio ha dati FFT cotti, restituisce FALSE se non ne viene trovata nessuna.
86. Is Playing = Restituisce TRUE se è in corso la riproduzione dell'audio del componente audio di destinazione.
87. Is Virtualized = Restituisce se l'audio è virtualizzato.
88. Play = Avvia la riproduzione del suono del componente audio di destinazione all'ora di inizio designata, cercando un suono.
89. Play Quantized = Avviare la riproduzione di un suono su un componente audio su un determinato limite di quantizzazione con Handle clock//
90. Set Audio Bus Send Post Effect = Imposta l'effetto di invio del bus audio
91. Set Audio Bus Send Pre Effect = Imposta la quantità di audio che il suono deve inviare al bus audio specificato (PRE Source Effects)
92. Set Low Pass Filter Enabled = Imposta filtro passa-basso abilitato
93. Set Low Pass Filter Frequency = Impostare la frequenza del filtro passa-basso
94. Set Output to Bus Only = Imposta se emettere o meno l'audio solo sul bus.
95. Set Paused = Mettere in pausa un componente audio che riproduce il segnale sonoro
96. Set Pitch Multiplier = Impostare un nuovo moltiplicatore di intonazione
97. Set Sound = Imposta suono
98. Set Source Bus Send Post Effect = Imposta l'effetto post di invio del bus di origines
99. Set Source Bus Send Pre Effect = Imposta l'effetto pre di invio del bus sorgente
100. Set Submix Send = Imposta la quantità di audio che il suono deve inviare al submix specificato.
101. Set UISound = Impostare se i suoni generati da questo componente audio devono essere considerati suoni dell'interfaccia utente
102. Set Volume Multiplier = Imposta moltiplicatore di volumeImposta moltiplicatore di volume
103. Stop = Stop Delayed
104. Stop Delayed = I segnali richiedono di interrompere il suono dopo il ritardo fornito (in secondi), interrompendosi immediatamente se il ritardo è zero o negativo
-Pawn Noise Emitter
105. Make Noise = Rumori della cache istigati dalla pedina proprietaria per il rilevamento dell'IA

-Effects
106. Add Master Submix Effect = Aggiungi l'effetto Master Submix
107. Add Source Effect to Preset Chain = Aggiungi l'effetto sorgente alla catena preimpostata
108. Add Submix Effect = Aggiungi effetto Submix

109. Clear Master Submix Effects = Cancella effetti Master Submix
110. Clear Submix Effect Chain Override = Cancella tutte le sostituzioni degli effetti submix sul submix specificato e lo riporta alla catena di effetti predefinita.
111. Clear Submix Effects = Cancella tutti gli effetti del submix sul submix dato.
112. Get Number Of Entries in Source Effect Chain = Restituisce il numero di voci della catena di effetti nella catena di effetti sorgente specificata.
113. Register Envelope Follower Listener = Registra un Envelope follower listener con effetto
114. Remove Master Submix Effect = Rimuovi l'effetto Master Submix
115. Remove Source Effect from Preset Chain = Rimuovi l'effetto sorgente dalla catena preimpostata
116. Remove Submix Effect = Rimuovi effetto Submix
117. Remove Submix Effect at Index = Rimuovi l'effetto Submix all'indice
118. Replace Submix Effect = Sostituisci effetto Submix
119. Reset Key = Target is Submix Effect Dynamics Processor Preset
120. Set Audio Bus = Imposta bus audio
121. Set Bypass Source Effect Chain Entry = Impostare se ignorare o meno l'effetto nell'indice della catena di effetti sorgente.
122. Set External Submix = Imposta submix esterno
123. Set Settings = Impostare le impostazioni
124. Set Submix Effect Chain Override = Imposta una sovrascrittura della catena di effetti submix sul submix specificato.
125. Unregister Envelope Follower Listener = Annulla la registrazione
--Bit Crusher
126. Set Bit Modulator = Imposta modulatore di bit
127. Set Bits = Target is Source Effect Bit Crusher Preset//
128. Set Modulation Settings = Impostazione delle impostazioni di modulazione
129. Set Sample Rate = Impostare la frequenza di campionamento//
130. Set Sample Rate Modulator = Impostare il modulatore della frequenza di campionamento
131. Set Settings = Imposta solo i valori di impostazione di base (cioè della portante) senza modificare i riferimenti alla sorgente di modulazione
--Chorus
132. Set Depth = Imposta il depth
133. Set Depth Modulator = Impostare il Depth Modulator
134. Set Dry = Set Dry
135. Set Dry Modulator = Setta il modulator
136. Set Feedback = Setta il feedback
137. Set Feedback Modulator = Setta Feedback Modulator
138. Set Frequency = Setta Frequency
139. Set Frequency Modulator = Setta Frequency Modulator
140. Set Modulation Settings = Setta i settaggi della Modulation
141. Set Settings = Setta Settings
142. Set Spread = Setta lo spread
143. Set Spread Modulator = Setta Spread Modulator
144. Set Wet = Setta Wet
145. Set Wet Modulator = Setta il wet modulator
--Delay
146. Get Max Delay in Milliseconds = Ottieni il ritardo massimo in millisecondi
147. Set Default Settings = Impostare le impostazioni predefinite

148. Set Delay = Imposta la durata effettiva del ritardo, in millisecondi.
149. Set Delay Length = Imposta la durata del ritardo
150. Set Dynamic Settings = Impostazione delle impostazioni dinamiche
151. Set Interpolation Time = Impostare il tempo di interpolazione
152. Set Maximum Delay Length = Imposta la lunghezza massima del ritardo
153. Set Settings = Impostare le impostazioni
--Filter
154. Set Filter Algorithm = Imposta l'algoritmo di filtro
155. Set Filter Cutoff Frequency = Imposta la frequenza di taglio del filtro
156. Set Filter Cutoff Frequency Mod = Imposta la frequenza di taglio del filtro Mod
157. Set Filter Q = Imposta filtro Q
158. Set Filter QMod = Imposta QMod filtro
159. Set Filter Type = Imposta il tipo di filtro
160. Set Settings = Impostare tutte le impostazioni dell'effetto filtro
--Tap Delay
161. Add Tap = Aggiunge un ritardo
162. Get Max Delay in Milliseconds = Ottieni il ritardo massimo in millisecondi
163. Get Tap = Ottieni Tap
164. Get Tap Ids = Recupera una matrice di tutti gli ID TAP per l'effetto submix.
165. Remove Tap = Rimuovi
167. Set Interpolation Time = Impostare il tempo di interpolazione
168. Set Settings = Impostare le impostazioni
169. Set Tap = Imposta il tap

-Envelope Following
170. Add Envelope Follower Delegate = Aggiunge un delegato di inseguimento dell'inviluppo al submix
quando l'opzione di inseguimento dell'inviluppo è abilitata su questo submix.
171. Start Envelope Following = Inviluppo iniziale dopo l'output del submix.
172. Stop Envelope Following = Interrompi inviluppo che segue

-Modulation
--Style
173. Get Control Bus Color = Ottenere il colore del bus di controllo
174. Get Control Bus Mix Color = Ottenere il colore del bus mix di controllo
175. Get Modulation Generator Color = Ottenere il colore del Modulation Generator
176. Get Parameter Color = Ottenere il colore del Parameter
177. Get Patch Color = Ottenere il colore del Patch

-OSC//

-Parameter
178. Boolean Array to Audio Parameter = Da array booleano a parametro audio
179. Boolean to Audio Parameter = Da booleano a parametro audio
180. Execute Trigger Parameter = Esegue un trigger denominato. Non memorizza nella cache il valore di attivazione,
quindi viene eseguito solo se il suono sta già giocando. Se l'intento è che il trigger venga eseguito immediatamente

chiamare 'SetBoolParameter' con il metodo Comportamento iniziale previsto del trigger (true)
181. Execute Trigger Parameter (Message) = Esegue un trigger denominato. Non memorizza nella cache il valore di attivazione,
quindi viene eseguito solo se il suono sta già giocando.
182. Float Array to Audio Parameter = Float Array al parametro audio
183. Float to Audio Parameter = Float al parametro audio
184. Integer Array to Audio Parameter
185. Integer to Audio Parameter
186. Object Array to Audio Parameter
187. Object to Audio Parameter
189. Reset Parameters = Ripristina tutti i parametri ai valori originali.
190. Reset Parameters (Message)
191. Set Boolean Array Parameter
192. Set Boolean Array Parameter (Message)
193. Set Boolean Parameter
194. Set Boolean Parameter (Message)
195. Set Float Array Parameter
196. Set Float Array Parameter (Message)
197. Set Float Parameter
198. Set Float Parameter (Message)
199. Set Integer Array Parameter
200. Set Integer Array Parameter (Message)
201. Set Object Array Parameter
202. Set Object Array Parameter (Message)
203. Set Object Parameter
204. Set Object Parameter (Message)
205. Set Parameters
206. Set Parameters (Message)
207. Set String Array Parameter
208. Set String Array Parameter (Message)
209. Set String Parameter
210. Set String Parameter (Message)
211. Set Wave Parameter
212. String Array to Audio Parameter
213. String to Audio Parameter

--Recording
214. Finish Recording Output = Interrompi la registrazione dell'audio.
Il percorso può essere assoluto o relativo (alla cartella /Saved/BouncedWavFiles). Lasciando vuoto il campo Submix To Record, puoi registrare l'output master del gioco.
215. Pause Recording Output = Metti in pausa la registrazione audio, senza finalizzare la registrazione su disco.
Lasciando vuoto il campo Submix To Record, puoi registrare l'output master del gioco.
216. Resume Recording Output = Riprendi la registrazione dell'audio dopo la pausa.
217. Start Recording Output = Avvia la registrazione dell'audio.

--Spectrum
218. Add Spectral Analysis Delegate = Aggiunge un delegato per l'analisi spettrale per ricevere

notifiche quando per questo sottomix è abilitata l'analisi dello spettro.
219. Remove Spectral Analysis Delegate = Rimuovere un delegato di analisi spettrale. L'obiettivo è il submix sonoro

--Subtitles
220. Are Subtitles Enabled
221. Set Subtitles Enabled

--Voice
Mic
222. Set Mic Threshold = Questa funzione imposta la soglia del microfono per la chat VOIP.
Notification
223. Create Talker for Player = Crea Talker per il giocatore
224. Get Voice Level = Ottieni il livello attuale di quanto forte sta parlando questo giocatore.
225. Register with Player State = Questa funzione configura questo oratore con un lettore specifico.
È necessario usarlo per controllare correttamente la voce di un giocatore specifico e ricevere eventi.
Sender
226. Is Idling = Thread sicuro modo per ottenere se questo componente synth ha ancora pacchetti bufferizzati di audio da riprodurre.
Questa funzione viene utilizzata dal motore vocale per ripulire i suoni vocali quando non vengono riprodotti
per fare spazio ad altri suoni da renderizzare.
L'obiettivo è il componente Voip Listener Synth

--AudioAnalyzer
227. Get Channel Constant QAt Time = Ottieni un canale cqt specifico del suono analizzato in un determinato momento L'obiettivo è QNRT const//Può esser utilizzato con Max of float array
228. Get Channel Loudness at Time = Ottenere un volume di canale specifico del suono analizzato in un determinato momento. L'obiettivo è Loudness NRT
229. Get Channel Onsets Between Times = Restituisce i timestamp che si sono verificati tra i timestamp di inizio e fine. L'obiettivo è la NRT
230. Get Loudness at Time = Ottieni il volume complessivo del suono analizzato in un determinato momento.
231. Get Normalized Channel Constant QAt Time = Ottieni un canale cqt specifico del suono analizzato in un determinato momento.
232. Get Normalized Channel Loudness at Time = Ottieni un volume normalizzato di un canale specifico del suono analizzato in un determinato momento.
Normalizzato Il volume è sempre compreso tra 0,0 e 1,0. 0,0 si riferisce al rumore di fondo, mentre 1,0 si riferisce al Volume massimo del suono.
233. Get Normalized Channel Onsets Between Times = Ottieni un canale cqt specifico del suono analizzato in un determinato momento.
234. Get Normalized Loudness at Time = Ottieni il volume complessivo normalizzato del suono analizzato in un determinato momento.
235. Start Analyzing = Avvia l'analisi dell'audio dal bus audio specificato.
Facoltativamente, eseguire l'override del bus audio che si desidera analizzare.
236. Stop Analyzing = Interrompe l'analisi dell'audio.

--Extra
237. Map Range Clamped = Restituisce il valore mappato da un intervallo a un altro in cui il valore è bloccato all'intervallo di input. +
238. Add Actor Local Offset = Aggiunge un delta alla posizione di questo componente nel suo quadro di riferimento locale.(Lo muove nello spazioe)
239. Enable Input = Inserisce l'attore nello stack di input gestito da un oggetto PlayerController.
240. Get All Actors Of Class = Trova tutti gli attori nel mondo della classe specificata.
Questa è un'operazione lenta, usare con cautela, ad es. non utilizzare tutti i fotogrammi.
241. Aggiungi una Timelines, (SHIFT + CLICK sulla curva per creare punti) per creare una sorta di Inviluppo Temporale,
Add Timelines, setta la NEW TRACK creata nella Timelines al valore di Set Volume Multiplier(NODO),
nel tempo questa estinguerà o aumentera il volume della traccia

--Audio Gameplay Volume
242. On Listener Enter = Chiamato quando un ascoltatore "entra" nel proxy associato L'obiettivo è Audio Gameplay Volume Interazione
243. On Listener Exit = Chiamato quando un listener "esce" dal proxy associato L'obiettivo è Audio Gameplay Volume Interazione
--Audio Gameplay
245. Set Enabled
246. Set Exterior LPF
247. Set Exterior Volume
248. Set Interior LPF
249. Set Interior Volume
250. Set Priority
251. Set Reverb Settings
252. Set Submix Override Settings
253. Set Submix Send Settings
254. Get Owning Actor = Restituisce l'attore proprietario di questo AnimInstance
255. Line Trace By Channel = Esegue una traccia di collisione lungo la linea specificata e restituisce il primo colpo di blocco incontrato.
256. Break Hit Result = Estrae i dati da un oggetto HitResult.
257. Select (UTILITIES) = Restituisce tutti i Physical Surface

--Sound Cue
258. Branch - Set Boolean Parameter
259. Continuous Modulator - Set Float Parameter
260. Crossfade by Param - Set Float Parameter
261. SoundClass - Set Wave Parameter
262. Switch - Set Integer Parameter
263. Wave Param - Set Wave Parameter

-- Extra 2
264. Load Stream Level - Il nodo Carica livello flusso in Unreal Engine 5 (UE5) viene utilizzato per caricare e scaricare dinamicamente i livelli durante il gioco.
265. Random Float In Range - Hai un Range di float, il risultato sarà basato su questo range.

--MIDI Device Manager
266. Create MIDIDevice Controller = Crea un'istanza di un controller di dispositivo MIDI che può essere utilizzato per interagire con un dispositivo MIDI collegato
267. Create MIDIDevice Input Controller = Crea un'istanza di un controller di dispositivo MIDI che può essere utilizzato per interagire con un dispositivo MIDI collegato
268. Create MIDIDevice Output Controller = Crea un'istanza di un controller del dispositivo di output MIDI che può essere utilizzato per interagire con un dispositivo MIDI collegato
269. Find All MIDIDevice Info = Enumera tutti i dispositivi di input e output MIDI e riporta informazioni utili come ID e nomi di tali dispositivi.
270. Find MIDIDevices = Enumera tutti i dispositivi MIDI collegati e riporta gli ID e i nomi di tali dispositivi.
270.1. Break FoundMIDIDevice = Aggiunge un nodo che interrompe (parola chiave Break in for each loop) un 'FoundMIDIDevice' nei relativi campi membro
271. Get Default MIDIInput Device ID = Recupera l'ID del dispositivo di input MIDI predefinito.
Chiamare prima "Find All MIDI Device Info" per enumerare i dispositivi di input disponibili.
272. Get Default MIDIOutput Device ID = Recupera l'ID del dispositivo di output MIDI predefinito.
Chiamare "Find All MIDI Device Info" in anticipo per enumerare i dispositivi di input disponibili.
273. Get MIDIInput Device IDBy Name = Recupera l'ID del dispositivo di input MIDI in base al nome.
Chiamare "Find All MIDI Device Info" in anticipo per enumerare i dispositivi di input disponibili.
274. Get MIDIOutput Device IDBy Name 0 Recupera l'ID del dispositivo di output MIDI in base al nome.
Chiamare "Find All MIDI Device Info" in anticipo per enumerare i dispositivi di output disponibili.

--MIDI Device Controller
275. On MIDIEvent = Registrati con questo per scoprire gli eventi MIDI in arrivo da questo dispositivo

--MIDI Device Input Controller
276. On MIDIAftertouch = Registrati con questo per ricevere gli eventi MIDI Aftertouch in arrivo da questo dispositivo
277. On MIDIChannel Aftertouch = Registrati con questo per ricevere gli eventi MIDI Channel Aftertouch in arrivo da questo dispositivo
278. On MIDIControl Change = Registrarsi con questo per ricevere gli eventi di Control Change MIDI in arrivo da questo dispositivo
279. On MIDINote Off = Registrati con questo per ricevere gli eventi MIDI Note Off in arrivo da questo dispositivo
280. On MIDINote On = Registrarsi con questo per ricevere gli eventi MIDI Note On in arrivo da questo dispositivo
281. On MIDIPitch Bend = Registrati con questo per ricevere gli eventi MIDI Pitch Bend in arrivo da questo dispositivo
282. On MIDIProgram Change = Registrarsi con questo per ricevere gli eventi MIDI Program Change in arrivo da questo dispositivo

--MIDI Device Output Controller
283. Send MIDIChannel Aftertouch = Invia il tipo di evento Aftertouch del canale MIDI La

destinazione è il controller di output MIDIDevice
284. Send MIDIControl Change = Invia il tipo di evento MIDI Control Change La destinazione è MIDIDevice Output Controller
285. Send MIDIEvent = Invia i dati grezzi dell'evento MIDI per un tipo di evento La destinazione è MIDIDevice Output Controller
286. Send MIDINote Aftertouch = Invia il tipo di evento MIDI Note Aftertouch La destinazione è MIDIDevice Output Controller
287. Send MIDINote Off = Invia il tipo di evento MIDI Note Off La destinazione è il controller di uscita MIDIDevice
288. Send MIDINote On = Invia MIDI Nota sul tipo di evento La destinazione è MIDIDevice Output Controller
289. Send MIDIPitch Bend = Invia il tipo di evento MIDI Pitch Bend La destinazione è MIDIDevice Output Controller
290. Send MIDIProgram Change = Invia il tipo di evento MIDI Program Change La destinazione è MIDIDevice Output Controller

--Extra3
291. ToText (integer) = Converte un numero intero passato in testo in base alle opzioni di formattazione
293. Format Text = Compila una stringa formattata utilizzando i valori degli argomenti di formato disponibili. ·
Utilizzare {} per indicare gli argomenti di formato. · I tipi di argomento possono essere Byte, Integer, Float, Text, String, Name, Boolean, Object o ETextGender.
294. Is Valid = Restituisce true se l'oggetto è utilizzabile: non null e non in attesa di uccisione

--NODI EXTRA AUDIO--PT1

--Utilities
Nodi di utilità per la gestione del controllo di flusso, array, stringhe, nomi, testo, ecc.
-Actions
295. Add Force = Aggiungere una forza a un singolo corpo rigido. Questo è come un "propulsore".
Ottimo per aggiungere una raffica in un certo tempo (diverso da zero).
Dovrebbe essere chiamato ogni fotogramma per la durata della forza.
296. Async Load Asset = Carica in modo asincrono un riferimento a un oggetto soft e restituisce un oggetto del tipo corretto se il caricamento ha esito positivo
297. Async Load Class Asset = Asset della classe di carico asincrono
298. Break ARFilter = Spezza uno struct ARFilter nei suoi componenti.
299. Break Soft Class Path = Ottiene la stringa del percorso da un percorso di classe soft.
300. Break Soft Object Path = Ottiene la stringa del percorso da un percorso oggetto soft. La destinazione è la libreria di sistema Kismet//Utilissimo
301. Break Top Level Asset Path = Ottiene la stringa del percorso da un TopLevelAssetPath La destinazione è la libreria di sistema Kismet
302. Cast To Class = Cast da un oggetto a una classe, funzionerà solo se l'oggetto è già una classe.
303. Class Is Child Of = Determine if a class is a child of another class. Booleano
304. Conv Component Reference to Soft Component Reference = Il riferimento del componente

Conv al target di riferimento del componente soft
305. Copy = Restituisce una copia del valore passato al suo interno.
306. Does Implement Interface = Controlla se questo oggetto implementa un'interfaccia specifica.
307. Editor Destroyed = Editor Destroyed Target is VPViewport Tickable Actor Base
308. Get Atmosphere Transmitance On Ground At Planet Top = Ottieni la trasmissione dell'atmosfera a terra sul pianeta
309. Get Class = Restituisce la classe di un oggetto passato, sarà sempre valido se l'oggetto non è NULL Il target è Gameplay Statics
310. Get Class Display Name = Restituisce il nome visualizzato di una classe Target è Kismet System Library
311. Get Class Path Name (Top Level Asset Path) = Restituisce il percorso completo alla classe specificata
come percorso di risorse di primo livello utilizzato dalle utilità delle risorse.
312. Get Command Line = Returns the command line that the process was launched with. Target is Kismet System Library
313. Get Data Table Row = Tenta di recuperare un TableRow da un DataTable tramite il suo RowName
314. Get Display Name = Restituisce il nome visualizzato (o l'etichetta dell'attore), da visualizzare come aiuto per il debug.
315. Get Editor Property = Tenta di recuperare il valore di una proprietà denominata dall'oggetto specificato
316. Get Enum Path Name (Top Level Asset Path) = Restituisce il percorso completo all'enumerazione specificata
come percorso di risorse di primo livello utilizzato dalle utilità delle risorse
317. Get Frame Count = Restituisce il valore di GFrameCounter, un conteggio parziale del numero di fotogrammi che si sono verificati.//Utile
318. Get Object Name = Restituisce il nome effettivo dell'oggetto.
319. Get Object Path String = Restituisce il percorso completo dell'oggetto specificato come stringa
320. Get Outer Object = Restituisce l'oggetto esterno di un oggetto.
321. Get Skip Assigning Gamepad to Player 1 = Ottieni Salta l'assegnazione del gamepad al bersaglio del giocatore 1 nelle impostazioni delle mappe di gioco
322. Get Soft Class Path = Restituisce il percorso completo alla classe specificata come percorso di classe soft (che può essere utilizzato come percorso di oggetto soft)
323. Get Soft Class Top Level Asset Path = Converte un riferimento di classe soft in un percorso di risorse di primo livello utilizzato dalle utilità delle risorse
324. Get Soft Object Path = Restituisce il percorso completo dell'oggetto specificato come percorso oggetto soft.
325. Get Struct Path Name (Top Level Asset Path) = Returns the full path to the specified struct as a Top Level Asset Path used by asset utilities Target is Kismet System Library
326. Has Launch Option = Controlla la riga di comando per vedere se l'opzione desiderata è stata specificata sulla riga di comando
327. Is Unattended = Restituisce vero se eseguito in modalità automatica
328. Is Valid = Restituisce vero se l'oggetto è utilizzabile: non nullo e non in attesa di kill L'obiettivo è la libreria di sistema Kismet
329. Is Valid Class = Restituisce vero se la classe è utilizzabile: non nullo e non in attesa di kill L'obiettivo è la libreria di sistema Kismet

330. Is Valid Soft Class Reference = Restituisce vero se il riferimento alla classe soft non è nullo.
331. Is Valid Soft Object Reference = Restituisce vero se il riferimento all'oggetto software non è nullo.
332. IsA = Restituisce se l'oggetto passato appartiene o meno al tipo di classe (o eredita da).
333. Load Asset Blocking = Risolve o carica immediatamente un riferimento a un oggetto software,
ciò causerà intoppi e, se possibile, è necessario utilizzare Load Asse Blocking //Utile
334. Load Class Asset Blocking = Risolve o carica immediatamente un riferimento di classe soft,
ciò causerà intoppi e, se possibile, è necessario utilizzare l'asset di classe di caricamento asincrono
335. Make ARFilter = Costruisce una struttura ARFilter. Dovresti utilizzare ClassPaths e RecursiveClassPathsExclusionSet,
ClassNames e RecursiveClassesExclusionSet sono deprecati.
336. Make Soft Class Path = Crea una struttura Soft Class Path da una stringa che contiene un percorso completo /cartella/nomepacchetto.class.
337. Make Soft Object Path = Costruisce una struttura Soft Object Path da una stringa che contiene un percorso completo
338. Make Top Level Asset Path = Crea una struttura TopLevelAssetPath da una singola stringa Path o da una stringa PackageName e AssetName.
L'obiettivo è la libreria di sistema Kismet
339. Make UDIMVirtual Texture from Texture 2Ds = Crea una texture virtuale UDIM da un elenco di texture 2D regolari L'obiettivo è la libreria di funzioni UDIMTexture
340. Parse Command Line = Analizza riga di comando
341. Parse Param = Restituisce vero se la stringa contiene -param (non specificare il -)
342. Parse Param Value = Restituisce 'valore' se -opzione=valore è nella stringa
343. Resolve Soft Reference = Risolve o crea un riferimento soft, collega un riferimento soft o hard al pin di ingresso.
344. Select = Restituisce l'opzione all'Indice
345. Select Class = If bPickA is true, A is returned, otherwise B is Target is Kismet Math Library (Compara le classi)
346. Select Object = Se bPickA è vero, viene restituito A, altrimenti B è Target è Kismet Math Library
347. Set Editor Property = Tenta di impostare il valore di una proprietà denominata sull'oggetto specificato
348. Set Skip Assigning Gamepad to Player 1 = Modifica l'opzione "Salta assegnazione del gamepad al giocatore 1" GameMapsSettings L'obiettivo è Impostazioni mappe di gioco
349. To Object (Interface) = Converts an interfance into an object Target is Kismet System Library
350. To Soft Class Path = Converte un riferimento di classe soft in un percorso di classe soft (che può essere utilizzato come un percorso di oggetto soft)
351. To Soft Class Reference = Converte un percorso di classe soft in un riferimento di classe soft di base,
non è garantito che sia risolvibile L'obiettivo è la libreria di sistema Kismet
352. To Soft Object Path = Converte un riferimento a un oggetto soft in un percorso di oggetto soft.
La destinazione è la libreria di sistema Kismet
353. To Soft Object Reference = Converte un percorso di oggetto soft in un riferimento di oggetto soft di base,

non è garantito che sia risolvibile L'obiettivo è la libreria di sistema Kismet
354. To String (SoftClassReference) = Converte un riferimento di classe soft in una stringa di percorso L'obiettivo è la libreria di sistema Kismet
355. To String (SoftObjectReference) = Converte un riferimento a un oggetto software in una stringa di percorso. L'obiettivo è la libreria di sistema Kismet
-Animation-Notifies
356. Get Current Animation Notify State Time = ttiene l'ora corrente in secondi relativa all'inizio dello stato di notifica
357. Get Current Animation Notify State Time Ratio = Ottiene l'ora corrente come rapporto (0 -> 1)
358. Get Current Animation Time = Ottieni l'ora corrente di notifica anim in secondi per quando è stata attivata questa notifica
359. Get Current Animation Time Ratio = Ottieni l'ora corrente di notifica dell'anim come rapporto (0 -> 1)
attraverso l'animazione per quando è stata attivata questa notifica L'obiettivo è la libreria di notifica dell'anim
360. Get Mirror Data Table = Se la notifica è sottoposta a mirroring, restituisce la tabella dati mirror che era attiva.
361. Is Triggered by State = Ottenere se uno stato con il nome specificato in qualsiasi macchina a stati ha attivato
la notifica Target è Anim Notify State Machine Inspection Library = Ottenere se un particolare stato in una macchina a
stati specifica ha attivato la notifica La destinazione è Anim Notify State Machine Inspection Library
362. Is Triggered by State in State Machine = Ottenere se la notifica è stata attivata dalla macchina a stati specificata La destinazione è Anim Notify State Machine Inspection Library
363. Is Triggered by State Machine = //
364. Notify State Reached End = Ottenere se lo stato di notifica ha raggiunto la fine (non è stato annullato) L'obiettivo è Anim Notify Library
-Animation-State Machine
365. State Machine Get Global Weight/Local
366. State Machine Is State Relevant
-Array
367. Add = Aggiungi elemento all'array
368. Add Unique = Aggiungi elemento all'array (univoco)
369. Append Array = Aggiungi un array a un altro array
370. Clear = Array da cancellare
371. Contains Item = Oggetto da trovare
372. Filter Array = L'array da cui filtrare
373. Find Item = Trova l'oggetto
374. Get (a copy) = Dato un array e un indice, restituisce una copia temporanea dell'elemento nell'array
375. Get (a ref) = Dato un array e un indice, restituisce l'elemento nell'array in corrispondenza di quell'indice
376. Identical = Controlla se due array sono identici in termini di membri.
377. Insert = Inserisce un elemento in un dato array e un dato indice
378. Is Empty = Un valore booleano che indica se l'array è vuoto
379. Is Not Empty = Un valore booleano che indica se l'array contiene elementi

380. Is Valid Index = L'Indice, che vogliamo testare per essere valido
381. Last Index = L'ultimo indice valido della matrice
382. Length = La lunghezza della matrice
393. Make Array = Crea un array da una serie di elementi.
396. Random Array Item = Ottiene un elemento casuale dall'array specificato.
397. Random Array Item from Stream = crea un elemento casuale dall'array specificato (utilizzando un flusso casuale) (random stream)
398. Remove Index = Rimuovi indice
399. Remove Item = Rimuovi elemento
400. Resize = La nuova dimensione dell'array
401. Reverse = Invertire gli elementi di un array L'obiettivo è la libreria di array Kismet
402. Set Array Elem = L'elemento da assegnare all'indice dell'array
401. Shuffle = Mescola (randomizza) gli elementi di un array
402. Shuffle from Stream = Mescola (randomizza) gli elementi di un array da un flusso specifico di dati casuali, utile per ottenere il determinismo.
403. Swap Array Elements = L'indice di un elemento da scambiare *
L'indice dell'altro elemento da scambiare
-Casting
-Collision
404. Trace Channel Test Util = Funzione di supporto per tracciare e modificare molte opzioni contemporaneamente.
-Flow Control
405. Branch = Istruzione Branch Se Condition è vera, l'esecuzione va a True, altrimenti va a False
406. Delay = Esegue un'azione latente con un ritardo (specificato in secondi).
407. Delay Until Next Tick = Esegue un'azione latente con un ritardo di un tick. Chiamare nuovamente mentre è in corso il conto alla rovescia verrà ignorato.
408. DoOnce MultiInput = K2Node Esegue una volta ingressi multipli
409. MultiGate = Esegue una serie di pin in ordine
410. Retriggerable Delay = Esegue un'azione latente con un ritardo riattivabile (specificato in secondi).
Chiamando di nuovo mentre è in corso il conto alla rovescia, il conto alla rovescia verrà reimpostato sulla Durata
411. Sequence = Esegue una serie di pin in ordine
412. Switch
-Instanced Struct//
Le structture (chiamate anche struct) sono un modo per raggruppare diverse variabili correlate in un unico posto.
Ogni variabile nella struttura è nota come membro della struttura. A differenza di un array, una struttura può contenere molti tipi di dati diversi (int, string, bool, ecc.).

```
struct {            // Structure declaration
  int myNum;        // Member (int variable)
  string myString;   // Member (string variable)
} myStructure;       // Structure variable//
```

413. Get Instanced Struct Value
414. Is Valid
415. Is Valid (Branch)
416. Make Instanced Struct = Crea una nuova InstancedStruct dal valore di origine specificato.
417. Reset = Reimposta un InstanziadStruct. L'obiettivo è la libreria di funzioni struct Utilis

418. Set Instanced Struct Value = Imposta il valore di InstancedStruct dal valore di origine specificato. L'obiettivo è la libreria di funzioni struct Utilis
- Internationalization
419. Clear Current Asset Group Culture = Cancella la cultura della categoria del gruppo di risorse specificata
420. Get Culture Display Name = Ottieni il nome visualizzato della cultura specificata.
421. Get Current Asset Group Culture = Restituisce la lingua corrente se la categoria del gruppo non ha una sostituzione della cultura.
422. Get Current Language = Ottieni la lingua corrente (per la localizzazione)
423. Get Current Locale = Ottieni le impostazioni internazionali correnti (per l'internazionalizzazione) come tag di lingua IETF
424. Get Default Language = Ottieni la lingua predefinita
425. Get Local Currency Code = Get Local Currency Code
426. Get Local Currency Symbol = Returns the currency symbol associated with the device's locale Target is Kismet System Library
427. Get Localized Cultures = Get the list of cultures that have localization data available for them. Target is Kismet Internationalization Library
428. Get Native Culture = Ottieni la cultura nativa per la categoria di localizzazione specificata.
429. Get Preferred Languages = Restituisce un array delle lingue preferite dell'utente in ordine di preferenz
430. Get Suitable Culture = Dato un elenco di culture disponibili, prova a trovare la cultura più adatta dall'elenco in base alla priorità delle culture. es.)
Se la tua lista fosse [en, fr, de] e la cultura specificata fosse "en-US", questa funzione restituirebbe "en". es.)
431. Set Current Asset Group Culture = Imposta la cultura della categoria del gruppo di risorse specificata da un tag di lingua IETF (ad esempio, "zh-Hans-CN").
432. Set Current Culture = Imposta la cultura corrente
433. Set Current Language = Imposta solo la lingua corrente (per la localizzazione).
436. Set Current Language and Locale = Imposta la lingua corrente (per la localizzazione) e le impostazioni locali (per l'internazionalizzazione).
437. Set Current Locale = Imposta solo le impostazioni internazionali correnti (per l'internazionalizzazione).
-Macro
//In C, una macro è un pezzo di codice in un programma che viene sostituito dal valore della macro. La macro è definita dalla direttiva #define.
/ C program to illustrate macros

```
#include <stdio.h>

// Macro definition
#define DATE 31

// Driver Code
int main()
{
   // Print the message
   printf("Lockdown will be extended"
         " upto %d-MAY-2020",//31-MAY-2020
         DATE);
```

```
   return 0;
}//
```
438. Assign = Assegna valore alla variabile
439. Local Array of Blend Sample Data Structures = Variabile locale temporanea della matrice di strutture dati campione di fusione
440. Local Array of Booleans = Variabile locale temporanea di matrice booleana
441. Local Array of Float (double-precision)s
442. Local Array of Integer64s
443. Local Array of Integers
444. Local Array of Rotators
445. Local Array of Strings
446. Local Array of Texts
447. Local Array of Transforms
448. Local Array of Vectors
449. Local Array of Wildcards
450. Local Blend Sample Data Structure
451. Local Float (double-precision)
452. Local Integer
453. Local Integer64
454. Local Rotator
455. Local String
456. Local Text
457. Local Transform
458. Local Vector
459. Local Wildcard
460. Persistent Local Boolean
461. Persistent Local Integer
-Map/mappa blueprint//
462. Add = Aggiunge una chiave e un valore alla mappa. Se qualcosa utilizza già la chiave fornita, verrà sovrascritta con il nuovo valore
463. Clear = Cancella una mappa da tutte le voci, reimpostandola su una destinazione vuota.
464. Contains = Controlla se la chiave si trova in una destinazione della mappa fornita è la libreria della mappa Blueprint
465. Find = Trova il valore associato alla chiave di destinazione fornita è Blueprint Map Library
466. Is Empty = Controlla se la mappa non ha interi Target è Blueprint Map Library
467. Is Not Empty = Controlla se la mappa contiene voci. L'obiettivo è la libreria mappe blueprint
468. Keys = Restituisce un array di tutte le chiavi presenti nella mappa
469. Length = Determina il numero di voci in una destinazione mappa fornita è la libreria mappa blueprint
470. Make Map = Crea una mappa da una serie di elementi chiave/valore.
471. Remove = Rimuove una chiave e il relativo valore associato dalla mappa. L'obiettivo è la libreria delle mappe dei progetti
472. Values = Restituisce un array di tutti i valori presenti nella mappa. L'obiettivo è la libreria della mappa Blueprint
-Name
473. Enum to Name = Restituisce il nome dell'enumeratore
474. Make Literal Name = Crea un nome letterale La destinazione è Kismet System Library

-Operators
475. Add
476. Divide
477. Equal
478. Greater
479. Greater Equal
480. Less
481. Less Equal
482. Multiply
483. Not Equal
484. Subtract
-Paths
485. Automation Dir = Restituisce la directory per i file di salvataggio dell'automazione.
486. Automation Log Dir = Restituisce la directory per i file di registro di automazione.
487. Automation Transient Dir = Restituisce la directory per i file di salvataggio dell'automazione che devono essere eliminati a ogni esecuzione
488. Bug It Dir = Restituisce la directory utilizzata dal motore per produrre i file BugIt.
489. Change Extension = Cambia l'estensione del nome file specificato (non fa nulla se il file non ha estensione)
490. Cloud Dir = Restituisce la directory per i file locali utilizzati nell'emulazione cloud o nel supporto
491. Collapse Relative Directories = Prende una stringa con percorso completo ed elimina il percorso relativo (es: annienta ".." con la directory adiacente).
492. Combine = Combina due o più percorsi in un unico percorso.
493. Convert from Sandbox Path = Converte un percorso sandbox (in Salvati/Sandbox) in un percorso normale.
494. Convert Relative Path to Full = Converte un nome di percorso relativo in un nome completo relativo al BasePath specificato.
495. Convert to Absolute Path = Converti passati nel nome file per utilizzare un percorso assoluto L'obiettivo è la libreria di sistema Kismet
496. Convert to Relative Path = Converti passati nel nome file per utilizzare un percorso relativo L'obiettivo è la libreria di sistema Kismet
497. Convert to Sandbox Path = Converte un percorso normale in un percorso sandbox (in Salvati/Sandbox). L'obiettivo è la libreria dei percorsi dei progetti
498. Create Temp Filename = Crea un nome file temporaneo con il prefisso specificato. L'obiettivo è la libreria dei percorsi dei progetti
499. Diff Dir = Restituisce la directory per i file temporanei utilizzati per la differenza. La destinazione è la libreria dei percorsi del progetto
500. Directory Exists = Restituisce vero se questa directory è stata trovata, falso altrimenti La destinazione è la libreria dei percorsi del progetto
501. Engine Config Dir = Restituisce la directory in cui si trovano i file di configurazione root. L'obiettivo è la libreria dei percorsi dei progetti
502. Engine Content Dir = Restituisce la directory dei contenuti del motore "core" che può essere condivisa tra più giochi o tra giochi e mod. L'obiettivo è la libreria dei percorsi dei progetti
503. Engine Dir = Restituisce la directory di base del motore "core" che può essere condivisa tra più giochi o tra giochi e mod.
504. Engine Intermediate Dir = Restituisce la directory intermedia del motore. La destinazione è la libreria dei percorsi del progetto

505. Engine Plugins Dir = Restituisce la directory dei plugin del motore.
506. Engine Saved Dir = Restituisce la directory salvata del motore. La destinazione è la libreria dei percorsi del progetto
507. Engine Source Dir = Restituisce la directory in cui vengono conservati i file del codice sorgente del motore.
508. Engine User Dir = Restituisce la directory root per i file del motore specifici dell'utente.
509. Engine Version Agnostic User Dir = Restituisce la directory principale per i file del motore specifici dell'utente che possono essere condivisi tra le versioni.
510. Enterprise Dir = Restituisce la directory aziendale della directory di base.
511. Enterprise Feature Pack Dir = Restituisce la directory FeaturePack aziendale.
512. Enterprise Plugins Dir = Restituisce la directory dei plug-in aziendali. L'obiettivo è la libreria dei percorsi Blueprint
513. Feature Pack Dir = Restituisce la directory in cui sono conservati i feature pack
514. File Exists = Restituisce vero se questo file è stato trovato, falso altrimenti
515. Game Agnostic Saved Dir = Restituisce la directory salvata che non è specifica del gioco.
516. Game Developers Dir = Restituisce la directory che contiene le sottocartelle per il contenuto specifico dello sviluppatore.
517. Game Source Dir = Restituisce la directory in cui vengono conservati i file del codice sorgente del gioco.
518. Game User Developer Dir = Restituisce la directory che contiene contenuto specifico dello sviluppatore per l'utente corrente.
519. Generated Config Dir = Restituisce la directory in cui il motore salva i file di configurazione generati.
520. Get Base Filename = Restituisce la stessa cosa di GetCleanFilename, ma senza l'estensione Target è Blueprint Paths Library
521. Get Clean Filename = Restituisce il nome del file (con estensione), meno le informazioni sul percorso.
522. Get Editor Localization Paths = Restituisce un elenco di percorsi di localizzazione specifici dell'editor
523. Get Engine Localization Paths = Restituisce un elenco di percorsi di localizzazione specifici del motore
524. Get Extension = Ottiene l'estensione per il nome file.
525. Get Game Localization Paths = Restituisce un elenco di percorsi di localizzazione specifici del gioco
526. Get Invalid File System Chars = Restituisce una stringa contenente tutti i caratteri non validi come dettato dal sistema operativo
527. Get Path = Restituisce il percorso davanti al nome del file
528. Get Project Content Directory = Ottenere la directory dei contenuti del progetto corrente
529. Get Project Directory = Ottenere la directory del progetto corrente
530. Get Project File Path = Ottiene il percorso del file di progetto.
531. Get Project Saved Directory = Ottenere la directory salvata del progetto corrente
532. Get Property Name Localization Paths = Restituisce un elenco di percorsi di localizzazione dei nomi delle proprietà
533. Get Relative Path to Root = Ottiene il percorso relativo per ottenere da BaseDir a RootDirectory
534. Get Restricted Folder Names = Restituisce un elenco di nomi di cartelle con restrizioni/interne (senza barre)
che possono essere testati rispetto ai percorsi completi per determinare se un percorso è

limitato o meno.
535. Get System Path = Restituisce il percorso completo del file system a un UObjec
536. Get Tool Tip Localization Paths = Restituisce un elenco di percorsi di localizzazione delle descrizioni comandi
537. Has Project Persistent Download Dir = Restituisce true se è disponibile una directory scrivibile per i dati scaricati che persistono tra le sessioni di riproduzione
538. Is Drive = Restituisce true se questo percorso rappresenta un'unità o un volume radice
539. Is Project File Path Set = Controlla se il percorso del file di progetto, se presente, è impostato.
540. Is Relative = Restituisce true se questo percorso è relativo a un altro percorso
541. Is Restricted Path = Determina se il percorso fornito utilizza una sottodirectory con restrizioni/interno.
542. Is Same Path = Controlla se due percorsi sono uguali.
543. Launch Dir = Restituisce la directory da cui è stata avviata l'applicazione (utile per le utilità della riga di comando)
544. Make Path Relative To = Supponendo che entrambi i percorsi (o nomi di file) siano relativi alla stessa directory di base
545. Make Platform Filename = Prende un nome di percorso "Unreal" e lo converte in un nome di file della piattaforma.
546. Make Standard Filename = Rendi il percorso "Unreal" completamente standard:
Normalizza i separatori di percorso [NormalizeFilename]
Rimuove i separatori estranei [NormalizeDirectoryName, oltre a rimuovere i separatori adiacenti]
Crolla interno..'s
Rende relativo a Engine\Binaries\<Platform> (inizierà SEMPRE con ..\..\..)
547. Make Valid File Name = Restituisce una stringa che è sicura da utilizzare come nome file
548. Normalize Directory Name = Normalizza tutti i / e \ in TEXT("/") e rimuovi qualsiasi TEXT("/") finale se il carattere precedente non è TEXT("/") o due punti.
549. Normalize Filename = Converti tutto / e \ in TEXT("/")
550. Profiling Dir = Restituisce la directory utilizzata dal motore per restituire i file di profilazione.
551. Project Config Dir = Restituisce la directory in cui si trovano i file di configurazione root.
552. Project Content Dir = Restituisce la directory dei contenuti del gioco corrente esaminando FApp::GetProjectName().
553. Project Dir = Restituisce la directory di base del progetto corrente esaminando FApp::GetProjectName().
554. Project Intermediate Dir = Restituisce la directory intermedia del gioco corrente guardando FApp::GetProjectName().
555. Project Log Dir = Restituisce la directory utilizzata dal motore per generare i log.
556. Project Mods Dir = Restituisce la directory mods del progetto corrente esaminando FApp::GetProjectName().
557. Project Persistent Download Dir = Restituisce la directory scrivibile per i dati scaricati che persiste tra le sessioni di gioco.
558. Project Plugins Dir = Restituisce la directory dei plugin del gioco corrente guardando FApp::GetProjectName().
559. Project Saved Dir = Restituisce la directory salvata del gioco corrente guardando FApp::GetProjectName().
560. Project User Dir = Restituisce la directory principale per i file di gioco specifici dell'utente.

561. Remove Duplicate Slashes = Rimuove le barre duplicate nei percorsi.
562. Root Dir = Restituisce la directory principale dell'albero delle directory del motore.
563. Sandboxes Dir = Restituisce la directory in cui il motore memorizza l'output sandbox.
564. Screen Shot Dir = Restituisce la directory utilizzata dal motore per generare i file di screenshot.
565. Set Extension = Imposta l'estensione del nome file specificato (come ChangeExtension, ma applica anche l'estensione se il file non ne ha una)
566. Set Project File Path = Imposta il percorso del file di progetto.
567. Shader Working Dir = Restituisce la destinazione della directory di lavoro dello shader è la libreria dei percorsi del progetto
568. Should Save to User Dir = Le strutture delle directory "salvate" dovrebbero essere rootate nella directory dell'utente o relative al "motore/gioco
569. Source Config Dir = Restituisce la directory utilizzata dal motore per cercare i file ini foglia di origine.
570. Split = Analizza un nome file completo o relativo nei suoi componenti (nome file, percorso, estensione).
571. Validate Path = Verifica che le parti che compongono il percorso non contengano caratteri non validi come indicato dal sistema operativo
572. Video Capture Dir = Restituisce la directory utilizzata dal motore per generare i file di acquisizione video richiesti dall'utente.
-Platform
573. Can Launch URL = La destinazione dell'URL di avvio può essere la libreria di sistema Kismet
574. Collect Garbage = Deletes all unreferenced objects, keeping only referenced objects
575. Control Screensaver = Consente o inibisce il salvaschermo
576. Force Close Ad Banner = Forza la chiusura di qualsiasi annuncio visualizzato.
577. Get Ad ID Count = Recupera il numero totale di ID annuncio che possono essere selezionati tra Target e Kismet System Library
578. Get Device Id = Restituisce l'ID dispositivo univoco specifico della piattaforma.
579. Get Gamepad Button Glyph = Restituisce il glifo assegnato a un pulsante del gamepad (o un ptr nullo se non assegnato)
580. Get Gamepad Controller Name = Restituisce il nome del controller se assegnato a un gamepad
581. Get Platform User Dir = Ottieni la directory dell'utente corrente dal sistema operativo target è la libreria di sistema Kismet
582. Get Platform User Name = Ottieni il nome utente corrente dal sistema operativo target è Kismet System Library
583. Get Volume Buttons Handled by System = Restituisce vero se la gestione predefinita del sistema dei pulsanti volume su e giù è abilitata
584. Hide Ad Banner = Hides the ad banner (iAd on iOS, or AdMob on Android).
585. Is Controller Assigned to Gamepad = Restituisce vero se l'ID controller assegnato a un gamepad (solo Android e iOS)
586. Is Interstitial Ad Available = Restituisce vero se l'annuncio interstitial richiesto è caricato e pronto (solo Android)
587. Is Interstitial Ad Requested = Restituisce vero se l'annuncio interstitial richiesto è stato richiesto con successo (falso se la richiesta di caricamento non riesce) (solo Android)
588. Is Screensaver Enabled = Restituisce vero se lo screen saver è abilitato. L'obiettivo è la libreria di sistema Kismet
589. Launch URL = Apre l'URL specificato nel browser Web scelto dalla piattaforma.

590. Load Interstitial Ad = Caricherà un annuncio AdMob interstitial a schermo intero.
591. Register for Remote Notifications = Richiede l'autorizzazione per inviare notifiche remote al dispositivo dell'utente.
592. Reset Gamepad Assignment to Controller = Reimposta l'assegnazione del gamepad all'ID del controller del giocatore (solo Android e iOS)
593. Reset Gamepad Assignments = Reimposta il gamepad sulle assegnazioni dell'ID controller del giocatore (solo Android e iOS)
594. Set Gamepads Block Device Feedback = Imposta se i gamepad collegati bloccheranno il feedback dal dispositivo stesso (solo dispositivi mobili).
595. Set Volume Buttons Handled by System = Consente o inibisce la gestione predefinita del sistema dei pulsanti di aumento e riduzione del volume (solo Android)
596. Set Window Title = Imposta il titolo della finestra di gioco.
597. Show Ad Banner = Mostrerà un banner pubblicitario (iAd su iOS o AdMob su Android) nella parte superiore o inferiore dello schermo, sopra la visualizzazione GL (non ridimensiona la visualizzazione)
598. Show Interstitial Ad = Mostra l'annuncio interstitial caricato (caricato con LoadInterstitialAd) (solo Android)
599. Show Platform Specific Achievements Screen = Visualizza la GUI degli obiettivi incorporati (solo iOS e Android; questa funzione potrebbe essere rinominata o spostata in una versione futura)
600. Show Platform Specific Leaderboard Screen = Visualizza la GUI della classifica integrata (solo iOS e Android; questa funzione potrebbe essere rinominata o spostata in una versione futura)
601. Unregister for Remote Notifications = Richieste Richiede l'annullamento della registrazione dalla ricezione di notifiche remote sul dispositivo dell'utente.
-Set
I set sono un tipo di contenitore associativo in cui ogni elemento deve essere unico perché il valore dell'elemento lo identifica.
I valori vengono memorizzati in un ordine specifico, ovvero ascendente o discendente.
602. Add = Aggiunge un elemento al set
603. Add Items = Aggiunge tutti gli elementi da una matrice a una destinazione impostata nella libreria di set di blueprint
604. Clear = Cancella un set, rimuove tutto il contenuto.
605. Contains Item = Restituisce vero se l'insieme contiene l'elemento specificato.
606. Difference = Assegna Risultato alla differenza relativa di due insiemi, A e B.
Cioè, Risultato conterrà tutti gli elementi che si trovano nell'insieme A ma non si trovano nell'insieme B.
607. Intersection = Assegna Risultato all'intersezione dell'insieme A e dell'insieme B,
Cioè, il risultato conterrà tutti gli elementi che si trovano sia nel Set A che nel Set B.
608. Is Empty = Controlla se il set è vuoto.
609. Is Not Empty = Controlla se il set contiene elementi.
610. Length = Ottieni il numero di elementi in un set.
611. Make Set = Crea un set da una serie di elemnti.
612. Remove = Rimuovi l'elemento dal set.
613. Remove Items = Rimuove tutti gli elementi di un Array da un set.
614. To Array = Restituisce un Array contenente copie delle voci di un Set.
615. Union = Assegna Risultato all'unione di due insiemi, A e B.
Il risultato conterrà tutti gli elementi presenti nel Set A e inoltre tutti gli elementi nel Set B.

-String
616. Append = Concatena due stringhe insieme per creare una nuova stringa.
617. Build String (Boolean) = Converte un valore booleano->stringa, creando una nuova stringa nel formato AppendTo+Prefix+InBool+Suffix
618. Build String (Float)
619. Build String (Integer)
620. Build String (IntVector)
621. Build String (LinearColor) = Converte un colore->stringa, creando una nuova stringa nella forma AppendTo+Prefix+InColor+Suffix
622. Build String (Name)
623. Build String (Object)
624. Build String (Rotator)
625. Build String (Vector)
626. Build String (Vector2D)
627. Contains = Restituisce se questa stringa contiene la sottostringa specificata.//
628. Cull Array = Accetta un array di stringhe e rimuove tutte le voci di lunghezza zero.
//Stringa di lunghezza zero è un termine utilizzato nella programmazione informatica per riferirsi a una stringa (una sequenza di caratteri) che non contiene alcun carattere.
È essenzialmente una stringa vuota.//
629. Enum to String = Restituisce il nome descrittivo dell'enumeratore
//In C++, un'enumerazione (o enum) è un tipo di dati definito dall'utente che consente di assegnare un insieme di costanti intere denominate.
Queste costanti rappresentano valori specifici e puoi usarle nel tuo codice per migliorare la leggibilità e la manutenibilità.
gli enumeratori vengono utilizzati solo per elencare i valori denominati.
ESEMPIO

```
#include <iostream>
using namespace std;

int main() {
   enum Gender { Male, Female };
   Gender gender = Male;

   switch (gender) {
      case Male:
         cout << "Gender is Male";
         break;
      case Female:
         cout << "Gender is Female";
         break;
      default:
         cout << "Value can be Male or Female";
   }

   return 0;
}
```

Output: “Gender is Male”
630. Equal Exactly (String) = Verifica se le stringhe di input sono uguali (A == B)
631. Equal, Case Insensitive (String) = Uguale, senza distinzione tra maiuscole e minuscole (stringa)
632. Find Substring = Trova l'indice iniziale di una sottostringa nella stringa specificata. La destinazione è la libreria di stringhe Kismet
//string substr (size_t pos, size_t len) const;
pos: Posizione del primo carattere da copiare.
len: Lunghezza della sottostringa.
size_t: È un tipo integrale senza segno

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
   // Take any string
   string s1 = "Geeks";

   // Copy two characters of s1 (starting
   // from position 3)
   string r = s1.substr(3, 2);

   // prints the result
   cout << "String is: " << r;

   return 0;
}
```

633. Get Character Array from String = Restituisce un array che contiene una voce per ogni carattere in SourceString Target è Kismet String Library//Utile
634. Get Character as Number = Ottiene un singolo carattere dalla stringa (come numero intero) L'obiettivo è la libreria di stringhe Kismet
635. Get Substring = Restituisce una sottostringa dalla stringa che inizia nella posizione specificata. L'obiettivo è la libreria di stringhe Kismet
636. Is Empty = Restituisce vero se la stringa è vuota L'obiettivo è la libreria di stringhe Kismet
637. Is Numeric = true se la stringa contiene solo caratteri numerici
638. Join String Array = Concatena una matrice di stringhe in un'unica stringa. L'obiettivo è la libreria di stringhe Kismet
639. Left = Restituisce il numero di caratteri più a sinistra.
640. Left Chop = Restituisce i caratteri più a sinistra della stringa tagliando il numero di caratteri specificato dalla fine.
641. Left Pad = The padded string
642. Len = Restituisce il numero di caratteri nella stringa
643. Log String = Stampa una stringa nel registro Se Stampa su registro è vero, sarà visibile nella finestra Output Log.
644. Make Literal String = Crea una stringa letterale

```
// C++ Proram to illustrate the use of integer literals
#include <iostream>
```

```
using namespace std;

int main()
{

	int age = 35; // An integer literal
	cout << "Age: " << age << endl;
	return 0;
}
```

Output
Age: 35
I valori letterali sono valori che vengono inseriti direttamente nel codice.
I valori letterali sono talvolta chiamati costanti letterali perché il loro significato non può essere ridefinito (5 indica sempre il valore integrale 5).

```
return 5;                    // 5 is an integer literal
bool myNameIsAlex { true };   // true is a boolean literal
double d { 3.4 };            // 3.4 is a double literal
std::cout << "Hello, world!"; // "Hello, world!" is a C-style string literal
```

//
645. Matches Wildcard = Cerca in questa stringa un determinato carattere jolly.
//I wildcard sono caratteri speciali che consentono una corrispondenza di modelli flessibile.
646. Mid = Restituisce la sottostringa dalla posizione iniziale per i caratteri Count, i caratteri di tutta la stringa da 0 a ...
647. Not Equal Exactly (String) = Verifica se la stringa di input non è uguale (A!= B)
648. Not Equal, Case Insenstive (String) = Verifica se la stringa di input non è uguale (A!= B), ignorando le differenze tra maiuscole e minuscole.
649. Parse Into Array = Ottiene una matrice di stringhe da una stringa di origine divisa da un separatore e le stringhe vuote possono facoltativamente essere selezionate.
650. Replace = Sostituisci tutte le occorrenze di una sottostringa in questa stringa.
//Esempio in C++

```
#include <iostream>
#include <string>
#include <boost/algorithm/string.hpp>



int main()
{
   std::string data = "Boost Library is simple C++ Library";

   std::cout<<data<<std::endl;

   // Replace all occurrences of 'LIB' with 'XXX'
   // Case Insensitive Version
   boost::ireplace_all(data, "LIB", "XXX");
```

```
    std::cout<<data<<std::endl;


}
```

Output

Boost XXXrary is simple C++ XXXrary//
651. Replace Inline = Sostituisci tutte le occorrenze di SearchText con SostituzioneText in questa stringa.
652. Reverse = Restituisce una copia di questa stringa, con i caratteri in ordine inverso.
653. Right = Restituisce la stringa a destra della posizione specificata, contando all'indietro da destra (fine della parola).
654. Right Chop = Restituisce la stringa a destra della posizione specificata, contando in avanti da sinistra (dall'inizio della parola)
655. Right Pad
656. Select String = Se bPickA è vero, viene restituito A, altrimenti B
657. Split = Divide questa stringa in una determinata posizione della stringa con distinzione tra maiuscole e minuscole.
658. Starts With = Verifica se questa stringa inizia con la stringa specificata.
659. String To Color = Converti la stringa in colore.
660. String To Float = Converte una stringa in un valore doppio
661. String To Float Vector
662. String To Integer
663. String To Integer64
664. String To Name
665. String To Rotator
666. String To Vector
667. String To Vector2D
668. Time Seconds to String = Converti un numero di secondi in minuti: stringa di formato secondi.millisecondi (compresi gli zeri iniziali)
669. To Lower = Restituisce una stringa convertita in MINUSCOLO
670. To String (Boolean) = Converte un valore booleano in una stringa, 'true' o 'false'
671. To String (Byte)
672. To String (Float Vector)
673. To String (Float)
674. To String (InputActionValue) = Converte un FInputActionValue in una stringa
675. To String (InputDeviceId) = Converte un valore InputDeviceId in una stringa
676. To String (Integer)
677. To String (Integer64)
678. To String (IntPoint)
679. To String (IntVector)
680. To String (LinearColor)
681. To String (Matrix)
682. To String (Name)
683. To String (Object)
684. To String (PlatformUserId)

685. To String (Rotator)
686. To String (Text)
687. To String (Transform)
688. To String (Vector)
689. To String (Vector2d)
690. To Upper = Restituisce una stringa convertita in maiuscolo.
691. ToString (DropTimecode) = Converti una struttura di timecode in una stringa
692. Trim = Rimuove i caratteri di spazio bianco dalla parte anteriore di questa stringa.
693. Trim Trailing = Rimuove gli spazi bianchi finali.
-String Table //In Unreal Engine, le STRING TABLE forniscono un modo conveniente per gestire il testo localizzato.
Ti consentono di centralizzare le tue voci di testo in una o più posizioni note, semplificando il riferimento e il riutilizzo dei contenuti localizzati in tutta l'applicazione.
694. Get Keys from String Table = Restituisce un array di tutte le chiavi all'interno della tabella di stringhe specificata.
695. Get Meta-Data IDs from String Table Entry = Restituisce un array di tutti gli ID di metadati all'interno della voce della tabella di stringhe specificata.
696. Get Registered String Tables = Restituisce un array di tutti gli ID della tabella di stringhe registrati
697. Get String Table Entry Meta-Data = Restituisce i metadati specificati della voce della tabella di stringhe specificata (o una stringa vuota).
698. Get String Table Entry Source String = Restituisce la stringa di origine della voce della tabella di stringhe specificata (o una stringa vuota).
699. Get String Table Namespace = Restituisce lo spazio dei nomi della tabella di stringhe specificata.
//
Namespace è uno strumento fondamentale in C++ per organizzare il codice, prevenire conflitti di nomi e migliorare la gestione dei nomi.

```
namespace mio_spazio {
   // Dichiarazioni di variabili, funzioni o classi
   int mia_variabile;
   void mia_funzione();
   class MiaClasse { /* ... */ };
}
```

Nota che non c’è il punto e virgola (;) alla fine delle dichiarazioni all’interno del namespace.
Utilizzo di un Namespace:
Per utilizzare una variabile, funzione o classe all’interno di un namespace, puoi fare riferimento al nome del namespace seguito da due punti ::. Ad esempio:

```
mio_spazio::mia_variabile = 42;
mio_spazio::mia_funzione();
```

Direttiva using namespace:
Puoi evitare di dover sempre specificare il nome del namespace utilizzando la direttiva using namespace.

```
#include <iostream>
using namespace std;
```

```
namespace primo_spazio {
   void mia_funzione() {
      cout << "All'interno di primo_spazio" << endl;
   }
}
```

670. Is String Table Entry Registered = Restituisce vero se l'ID della tabella fornita corrisponde a una tabella di stringhe registrata e tale tabella lo è.
671. Is String Table Registered = Restituisce vero se l'ID della tabella fornita corrisponde a una tabella di stringhe registrata.
-Struct
//Le strutture (chiamate anche struct) sono un modo per raggruppare diverse variabili correlate in un unico posto. Ogni variabile nella struttura è nota come membro della struttura.

```
struct MyStructure {
   int myNum;       // Member (integer variable)
   string myString;  // Member (string variable)
} myStructure;       // Structure variable
```

672. Break = Utilizza la parola chiave "Break" per utilizzare le strutture
-Text
673. As Currency = Genera un FText che rappresenta il numero passato come valuta nelle impostazioni cultura correnti.
BaseVal is specified in the smallest fractional value of the currency and will be converted for formatting according to the selected culture.
Keep in mind the CurrencyCode is completely independent of the culture it's displayed in (and they do not imply one another). For example:
FText::AsCurrencyBase(650, TEXT("EUR")); would return an FText of "<EUR>6.50" in most English cultures (en_US/en_UK) and "6,50<EUR>" in Spanish (es_ES) (where <EUR> is U+20AC)
Target is Kismet Text Library
674. As Date = Converte una data e un'ora passate in un testo, formattato come una data utilizzando un fuso orario invariante.
675. As Date (from UTC) = Converte la data e l'ora passate in un testo, formattato come una data utilizzando il fuso orario specificato (l'impostazione predefinita è il fuso orario locale)
676. As DateTime = Converte una data e un'ora passate in un testo, formattato come data e ora utilizzando un fuso orario invariante.
677. As DateTime (from UTC) = Converte una data e un'ora passate in un testo, formattato come data e ora utilizzando il fuso orario specificato (l'impostazione predefinita è il fuso orario locale).
678. As Percent = Converte un valore float passato in un testo, formattato come percentuale.
679. As Time = Converte una data e un'ora passate in un testo, formattato come un'ora utilizzando un fuso orario invariante.
680. As Time (from UTC) = Converte la data e l'ora passate in un testo, formattato come ora utilizzando il fuso orario specificato (l'impostazione predefinita è il fuso orario locale).
681. As Timespan = Converte un intervallo di tempo trascorso in un testo, formattato come un intervallo di tempo.
682. AsCurrency (integer) - DEPRECATED (use AsCurrency) = Converts a passed in integer to a text formatted as a currency Target is Kismet Text
683. Equal Exactly (Text) = Restituisce vero se A e B sono linguisticamente uguali (A == B).

684. Equal, Case Insensitive (Text) = Restituisce vero se A e B sono linguisticamente uguali (A == B), ignorando maiuscole e minuscole.
685. Find String Table ID and Key from Text = Tenta di trovare l'ID della tabella di stringhe e la chiave utilizzata dal testo specificato.
686. Find Text in Localization Table = Tenta di trovare il testo esistente utilizzando la rappresentazione trovata nelle tabelle loc per lo spazio dei nomi e la chiave specificati.
687. Format Text = Crea una stringa formattata utilizzando i valori degli argomenti di formato disponibili. · Utilizzare {} per denotare argomenti di formato.
688. Get Empty Text = Restituisce una parte di testo vuota.
689. Is Polyglot Data Valid = Controlla se i dati poliglotti forniti sono validi.
690. Is Text from String Table = Returns true if the given text is referencing a string table.
691. Make Literal Text = Crea un FText target letterale è la libreria di sistema Kismet
692. Make Text from String Table (Advanced) = Tenta di creare un'istanza di testo da una chiave e un ID di tabella di stringhe.
693. Not Equal Exactly (Text) = Returns true if A and B are linguistically not equal (A != B). Target is Kismet Text Library
694. Not Equal, Case Insensitive (Text) = Restituisce vero se A e B non sono linguisticamente uguali (A != B), ignorando maiuscole e minuscole.
695. Polyglot Data to Text = Ottieni l'istanza di testo creata da questi dati poliglotti.
696. Text Is Culture Invariant = Restituisce vero se il testo è invariante della cultura.
697. Text Is Empty = Restituisce vero se il testo è vuoto
698. Text Is Transient = Restituisce vero se il testo è transitorio.
//Oggetti transitori:
Il nome standard per quello che chiami "oggetto transitorio" è in realtà un "oggetto temporaneo".
Nel tuo esempio, la riga contrassegnata con [*] crea un oggetto temporaneo della classe Obj:
Test t = Obj();.//
699. Text to Lower = Trasforma il testo in minuscolo in modo corretto per la cultura.
700. Text to Upper = Trasforma il testo in maiuscolo in modo corretto per la cultura.
701. Text Trim Preceding = Rimuove i caratteri di spazio bianco dalla parte anteriore del testo.
702. Text Trim Preceding and Trailing = Rimuove i caratteri di spazio bianco dalla parte anteriore e finale del testo.
703. Text Trim Trailing = Rimuove i caratteri di spazio bianco finali.
704. To Text (Boolean) = Converte un valore booleano in testo formattato, 'true' o 'false'
705. To Text (Byte)
//In C++ il tipo std::byte è un tipo distinto che rappresenta il concetto di byte come specificato nella definizione del linguaggio C++.
706. To Text (Float)
707. To Text (Integer)
708. To Text (Integer64)
709. To Text (LinearColor)
710. To Text (Name)
711. To Text (Object)
712. To Text (Rotator)
713. To Text (String)
714. To Text (Transform)
715. To Text (Vector)
716. To Text (Vector2d)

-Time
717. Clear and Invalidate Timer by Handle = Cancella un timer impostato.
718. Clear Timer by Function Name = Cancella un timer impostato.
719. Does Timer Exist by Function Name = Restituisce vero se esiste un timer per il delegato specificato, falso altrimenti.
720. Does Timer Exist by Handle = Restituisce vero se esiste un timer per l'handle specificato, falso altrimenti.
721. Get Accurate Real Time = Restituisce il tempo in secondi dall'avvio dell'applicazione. A differenza delle altre funzioni temporali,
questa è precisa all'ora esatta in cui viene chiamata questa funzione invece di essere impostata una volta per fotogramma. L'obiettivo sono le statistiche di gioco
722. Get Audio Time Seconds = Restituisce il tempo in secondi da quando il mondo è stato attivato per il gioco, viene interrotto quando il gioco viene messo in pausa, NON dilatato/bloccato.
723. Get Game Time in Seconds = Ottieni il tempo di gioco attuale, in secondi.
724. Get Global Time Dilation = Ottiene la dilatazione temporale globale corrente.
725. Get Platform Time Seconds = Restituisce l'ora corrente della piattaforma in secondi.
726. Get Real Time Seconds = Restituisce il tempo in secondi da quando il mondo è stato creato per il gioco, NON si ferma quando il gioco viene messo in pausa, NON si dilata/blocca
Il bersaglio è Statica di gioco
727. Get Time Seconds = Restituisce il tempo in secondi da quando il mondo è stato attivato per il gioco, regolato dalla dilatazione del tempo e dall'IS interrotto quando il gioco viene messo in pausa
728. Get Timer Elapsed Time by Function Name = Restituisce il tempo trascorso per il delegato specificato (tempo trascorso dall'inizio dell'iterazione del conto alla rovescia corrente).
729. Get Timer Elapsed Time by Handle = Restituisce il tempo trascorso per l'handle specificato (tempo trascorso dall'inizio dell'iterazione del conto alla rovescia corrente).
730. Get Timer Remaining Time by Function Name = Restituisce il tempo trascorso fino alla successiva esecuzione del delegato da parte del timer. L'obiettivo è la libreria di sistema Kismet
731. Get Timer Remaining Time by Handle = Restituisce il tempo trascorso fino alla successiva esecuzione dell'handle da parte del timer.
732. Get Unpaused Time Seconds = Restituisce il tempo in secondi da quando il mondo è stato attivato per il gioco,
regolato dalla dilatazione del tempo e NON viene interrotto quando il gioco viene messo in pausa
733. Get World Delta Seconds = Restituisce il tempo delta del frame in secondi, regolato dalla dilatazione del tempo.
734. Invalidate Timer Handle = Invalidare il TimerHandle fornito e restituirlo.
735. Is Timer Active by Function Name
736. Is Timer Active by Handle
737. Is Timer Paused by Function Name
738. Is Timer Paused by Handle
739. Is Valid Timer Handle
740. Pause Timer by Function Name = Mette in pausa un timer impostato al tempo trascorso corrente.
741. Pause Timer by Handle = Mette in pausa un timer impostato al tempo trascorso corrente.
742. Set Global Time Dilation = Imposta la dilatazione temporale globale.
743. Set Timer by Event = Imposta un timer per l'esecuzione del delegato.

L'impostazione di un timer esistente ripristinerà il timer con i parametri aggiornati.
L'obiettivo è la libreria di sistema Kismet
744. Set Timer by Function Name = Imposta un timer per eseguire il delegato.
745. Set Timer for Next Tick by Event = Imposta un timer per eseguire il tick successivo del delegato.
746. Set Timer for Next Tick by Function Name
747. Unpause Timer by Function Name = Riprende un timer in pausa dal tempo trascorso corrente.
748. Unpause Timer by Handle = Riprende un timer in pausa dal tempo trascorso corrente.
-Time Management
749. Break Frame Rate = Suddivide un FFrameRate in un numeratore e un denominatore.
750. Break Qualified Frame Time = Suddivide nuovamente un FQualifiedFrameTime nelle sue parti componenti.
751. FrameNumber to Integer = Converte un FrameNumber in un int32 da utilizzare in funzioni che accettano conteggi di frame int32 per comodità.
752. FrameRate To Interval = Converte un FrameRate in un intervallo float che rappresenta il tempo del frame in secondi, ovvero: 1/30 restituisce 0,0333333
753. Get Timecode = Ottieni il timecode corrente del motore.
754. Get Timecode Frame Rate = Ottiene la frequenza fotogrammi del codice temporale corrente.
755. Is Multiple Of = Controlla se questo framerate è un multiplo pari di un altro framerate, ovvero: 60 è un multiplo di 30, ma 59,94 non lo è.
756. Is Valid Frame Rate = Verifica che si tratti di un framerate valido con un denominatore diverso da zero.
757. Make Frame Rate = Crea un FFrameRate da un numeratore e un denominatore.
758. Make Qualified Frame Time = Crea un FQualifiedFrameTime da un numero di frame, una frequenza fotogrammi e un sottoframe bloccato 0-1 opzionale.
759. QualifiedFrameTime To Seconds = Converte un QualifiedFrameTime in secondi.
760. Snap Frame Time = Aggancia l'oggetto SourceTime specificato al fotogramma più vicino nel framerate di destinazione specificato. Utile per determinare il fotogramma più vicino per un'altra risoluzione.
761. Timecode To String = Converte un Timecode in una stringa (hh:mm:ss:ff). Se bForceSignDisplay, il segno numerico verrà sempre anteposto anziché solo quando si esprime un tempo negativo.
762. Transform Frame Time = Converte il tempo specificato da un framerate a un altro framerate.
-World
763. GetWorldSettings = Restituisce l'attore AWorldSettings associato a questo mondo.
-World Partition
764. Get Data Layer Manager = Restituisce l'UDataLayerManager associato a questo mondo.
--Actor
//Un attore è qualsiasi oggetto che può essere posizionato in un livello, come una telecamera, una mesh statica o la posizione di partenza del giocatore. Gli attori supportano trasformazioni 3D come traslazione, rotazione e ridimensionamento//
765. Actor Has Tag = Verifica se l'array Tags di questo attore contiene il tag del nome fornito
Target is Actor
766. Destroy Actor = Distruggi l'attore
767. Find Component by Tag = Cerca l'array di componenti e restituisce il primo componente

incontrato con un determinato tag.
768. Find Nearest Actor = Restituisce l'attore più vicino all'origine dall'array ActorsToCheck.
769. Get Actor Eyes View Point = Restituisce il punto di vista dell'attore. Si noti che questo non significa la telecamera, ma gli "occhi" dell'attore.
770. Get Actor List from Component List = Restituisce una serie di attori univoci rappresentati dall'elenco di componenti fornito.
771. Get Actor Of Class = Trova il primo attore nel mondo della classe specificata
772. Get Actor Time Dilation = Ottieni ActorTimeDilation: può essere utilizzato per il controllo dell'input o il controllo della velocità per slomo.
773. Get All Actors Of Class = Trova tutti gli attori nel mondo della classe specificata.
774. Get All Actors Of Class with Tag = Trova tutti gli attori nel mondo della classe specificata con il tag specificato.
775. Get All Actors with Interface = Trova tutti gli attori del mondo con l'interfaccia specificata.
776. Get All Actors with Tag = Trova tutti gli attori del mondo con il tag specificato.
777. Get All Child Actors = Restituisce un elenco di tutti gli attori generati dai nostri componenti attore figlio
778. Get Attach Parent Actor = Risaliamo la catena degli allegati da RootComponent finché non incontriamo un attore diverso e lo restituiamo.
//RootComponent è un'istanza della classe USceneComponent. Determina il modo in cui un attore è posizionato, orientato e ridimensionato nel mondo di gioco.//
779. Get Attach Parent Socket Name = Risaliamo la catena degli allegati da RootComponent finché non incontriamo un attore diverso e restituiamo il nome del socket nel componente.
780. Get Attached Actors = Trova tutti gli attori collegati direttamente a un componente in questo attore.
781. Get Component by Class = Cerca l'array di componenti e restituisce il primo componente incontrato della classe specificata
//Un componente è una funzionalità che può essere aggiunta a un attore.//
782. Get Components By Class = Ottiene tutti i componenti che ereditano dalla classe specificata.
Attualmente restituisce un array di UActorComponent di cui deve essere eseguito il cast nel tipo corretto.
783. Get Components by Interface = Ottiene tutti i componenti che implementano l'interfaccia specificata.
784. Get Components by Tag = Ottiene tutti i componenti che ereditano dalla classe specificata con un determinato tag.
785. Get Game Time Since Creation = Il numero di secondi (nel tempo di gioco) da quando è stato creato questo attore, relativo a Ottieni tempo di gioco in secondi.
786. Get Life Span = Ottieni la durata di vita rimanente di questo attore.
787. Get Owner = Ottieni il proprietario di questo attore, utilizzato principalmente per la replica di rete.//IMPORTANTE PER IL FOOTSTEP
788. Get Parent Actor = Se questo attore è stato creato da un componente attore figlio, restituisce l'attore che possiede il componente attore figlio, la destinazione è l'attore
789. Get Parent Component = Se questo attore è stato creato da un componente attore figlio, restituisce che il target del componente attore figlio è Atto
790. Is Child Actor = Is Child Actor
791. Set Life Span = Imposta la durata della vita di questo attore. Alla scadenza l'oggetto verrà distrutto
792. Set Owner = Imposta il proprietario di questo attore, utilizzato principalmente per la replica

di rete.
-Tick
//L'Actor Tick è un evento regolare che si verifica durante il gioco. Rappresenta un aggiornamento del frame per un attore (come un personaggio, un oggetto o un nemico) all'interno del mondo di gioco.
Nel mondo di gioco, l'attore aggiorna il suo stato e risponde ai cambiamenti nell'ambiente di gioco.
Quando si verifica il Actor Tick?
Avviene una volta per fotogramma, in genere durante il ciclo principale del gioco.
Durante ogni tick, l'attore può eseguire logica personalizzata, gestire input, aggiornare animazioni e altro ancora.//
793. Add Tick Prerequisite Actor = Fai in modo che questo attore spunti dopo PrerequisiteActor.
Questo si applica solo alla funzione tick di questo attore; se lo si desidera, le dipendenze per i componenti posseduti devono essere impostate separatamente. L'obiettivo è l'attore
794. Add Tick Prerequisite Component = Fai in modo che questo attore spunti dopo PrerequisiteComponent.
Questo si applica solo alla funzione tick di questo attore; se lo si desidera, le dipendenze per i componenti posseduti devono essere impostate separatamente. L'obiettivo è l'attore
795. Get Actor Tick Interval = Restituisce l'intervallo di tick della funzione di tick primaria di questo attore.
796. Get Tickable when Paused = Ottiene se questo attore può spuntare quando è in pausa.
797. Is Actor Tick Enabled = Restituisce se questo attore ha il segno di spunta abilitato o meno.//IMPORTANTE
798. Remove Tick Prerequisite Actor = Rimuovi la dipendenza dal segno di spunta su PrerequisiteActor.
799. Remove Tick Prerequisite Component = Rimuovere la dipendenza dal segno di spunta su PrerequisiteComponent.
800. Set Actor Tick Enabled = Imposta le funzioni di spunta di questo attore su abilitate o disabilitate.
801. Set Actor Tick Interval = Imposta l'intervallo di tick della funzione di tick primaria di questo attore. Non abiliterà la funzione di spunta disabilitata. Ha effetto al tick successivo.
802. Set Tick Group = Imposta il gruppo di spunta per questo attore.
803. Set Tickable when Paused = Imposta se questo attore può tick quando è in pausa.
--Collision
804. Box Overlap Actors = Restituisce una serie di attori che si sovrappongono al riquadro allineato all'asse specificato.
805. Box Overlap Component = Eseguire una sovrapposizione del riquadro rispetto a un singolo componente poiché un target AABB (senza rotazione) è un componente primitivo
806. Box Overlap Components = Restituisce un array di componenti che si sovrappongono al riquadro allineato all'asse specificato
807. Box Trace By Channel = Spazza una casella lungo la linea data e restituisce il primo colpo di blocco incontrato.
Questa traccia trova gli oggetti che RISPONDE al TraceChannel Target specificato è la libreria di sistema Kismet
808. Box Trace By Profile = Spazza una scatola contro il mondo e restituisci il primo colpo di blocco utilizzando un profilo specifico.
809. Box Trace For Objects = Spazza una casella lungo la linea data e restituisce il primo colpo riscontrato.

810. Break Hit Result = Estrae i dati da un HitResult.
811. Can Character Step Up = Restituisce vero se il pedone indicato può salire su questo componente.
812. Capsule Overlap Actors = Restituisce una serie di attori che si sovrappongono alla capsula specificata.
813. Capsule Overlap Components = Restituisce un array di componenti che si sovrappongono alla capsula specificata.
814. Capsule Trace By Channel = Trascina una capsula lungo la linea indicata e restituisce il primo colpo di blocco incontrato.
815. Capsule Trace By Profile
816. Capsule Trace For Objects
817. Clear Move Ignore Actors = Cancella l'elenco degli attori che ignoriamo durante lo spostamento.
818. Clear Move Ignore Components = Cancella l'elenco dei componenti che ignoriamo durante lo spostamento.
819. Component Overlap Actors = Restituisce una serie di attori che si sovrappongono al componente specificato.
820. Component Overlap Components = Restituisce un array di componenti che si sovrappongono al componente specificato
821. Find Collision UV = Prova a trovare l'UV per un impatto di collisione.
822. Get Actor Array Bounds = Vincolare i limiti di una serie di attori
823. Get Actor Bounds = Restituisce il riquadro di delimitazione di tutti i componenti che compongono questo attore (esclusi ChildActorComponents)
824. Get Actor Enable Collision = Ottieni lo stato attuale della collisione per l'intero attore
825. Get Closest Point on Collision = Restituisce la distanza e il punto più vicino alla superficie di collisione.
826. Get Collision Components = Restituisce i componenti di collisione del corpo.
827. Get Collision Enabled = Restituisce la forma della collisione per questo componente.
828. Get Collision Object Type = Ottiene il tipo di oggetto collisione
829. Get Collision Profile Name = Ottieni il nome del profilo di collisione
830. Get Collision Response to Channel = Ottiene il tipo di risposta dato un canale specifico.
831. Get Component Bounds = Ottieni "la scatola di delimitazione"
832. Get Move Ignore Actors = Restituisce l'elenco degli attori che attualmente ignoriamo durante lo spostamento.
833. Get Move Ignore Components = Get Move Ignore Components
834. Get Overlapping Actors = Restituisce l'elenco degli attori a cui questo attore si sovrappone (qualsiasi componente che si sovrappone a qualsiasi componente).
835. Get Overlapping Components = Restituisce un elenco univoco di componenti a cui questo componente si sovrappone.
836. Get Surface Type = Restituisce il tipo EPhysicalSurface dell'Hit specificato. Per modificare il tipo di superficie per il tuo progetto, utilizza la sezione ProjectSettings/Physics/PhysicalSurface section
837. Ignore Actor when Moving = Indica a questo componente se ignorare la collisione con tutti i componenti di un attore specifico quando questo componente viene spostato.
838. Ignore Component when Moving = Indica a questo componente se ignorare la collisione con un altro componente quando questo componente viene spostato.
839. Is Collision Enabled = Utilità per verificare se è presente qualche forma di collisione (query o fisica) abilitata su questo componente.

840. Is Overlapping Actor = Controlla se questo componente si sovrappone a qualsiasi componente del dato attore.
841. Is Overlapping Component = Controlla se questo componente si sovrappone a un altro componente.
842. Is Physics Collision Enabled = Utilità per vedere se c'è qualche collisione fisica abilitata su questo componente.
843. Is Query Collision Enabled = Utilità per verificare se sono presenti conflitti di query abilitati su questo componente.
844. Line Trace By Channel = Esegue una collisione lungo la linea data e restituisce il primo colpo bloccante incontrato
844. Line Trace By Profile = Traccia un raggio contro il mondo utilizzando un profilo specifico e restituisci il primo colpo di blocco.
845. Line Trace Component = Eseguire una traccia della linea rispetto a un singolo componente.
846. Line Trace For Objects = Esegue una collisione lungo la linea data e restituisce il primo colpo riscontrato.
847. Make Hit Result = Crea una struttura HitResult L'obiettivo è la statica del gameplay
848. Multi Box Trace By Channel = Spazza una casella lungo la linea data e restituisce tutti i risultati incontrati.
849. Multi Box Trace By Profile
850. Multi Box Trace For Objects
851. Multi Capsule Trace By Channel
852. Multi Capsule Trace By Profile
853. Multi Capsule Trace For Objects
854. Multi Line Trace By Channel
855. Multi Line Trace By Profile
856. Multi Line Trace For Objects
857. Multi Sphere Trace By Channel
858. Multi Sphere Trace By Profile
859. Multi Sphere Trace For Objects
860. On Actor Begin Overlap = Chiamato quando un altro attore inizia a sovrapporsi a questo attore, ad esempio un giocatore che entra in un trigge
861. On Actor End Overlap = Chiamato quando un altro attore smette di sovrapporsi a questo attore.
862. On Actor Hit = Chiamato quando questo attore colpisce (o viene colpito da) qualcosa di solido. Ciò potrebbe accadere a causa di cose come il movimento del personaggio, l'utilizzo di Imposta posizione con lo "spostamento" abilitato o la simulazione fisica. Per gli eventi in cui gli oggetti si sovrappongono (ad esempio, quando si entra in un trigger), vedere l'evento "Sovrapposizione".
863. On Chaos Physics Collision
864. On Component Begin Overlap = Evento chiamato quando qualcosa inizia a sovrapporsi a questo componente, ad esempio un giocatore che entra in un trigger.
865. On Component End Overlap = Evento chiamato quando qualcosa smette di sovrapporsi a questo componente
866. On Component Hit = Evento chiamato quando un componente colpisce (o viene colpito da) qualcosa di solido.
867. On Component Sleep = Evento chiamato quando gli oggetti fisici sottostanti vengono sospesi
868. On Component Wake = Evento chiamato quando gli oggetti fisici sottostanti vengono

riattivati
869. Set Actor Enable Collision = Permette di abilitare/disabilitare la collisione per l'intero attore Target is ActoR
870. Set Collision Enabled = Controlla il tipo di collisione abilitata per questo corpo.
871. Set Collision Object Type = Modifica il canale di collisione utilizzato da questo oggetto quando si sposta.
872. Set Collision Profile Name = Imposta nome profilo di collisione
873. Set Collision Response to All Channels = Modifica tutto il contenitore ResponseToChannels per questo PrimitiveComponent.
874. Set Collision Response to Channel = Modifica un membro del contenitore ResponseToChannels per questo PrimitiveComponent.
875. Sphere Overlap Actors = Restituisce una serie di attori che si sovrappongono alla sfera data.
876. Sphere Overlap Component = Esegui una sovrapposizione di sfere su un singolo componente
877. Sphere Overlap Components = Restituisce una serie di componenti che si sovrappongono alla sfera data.
878. Sphere Trace By Channel = Trascina una sfera lungo la linea data e restituisce il primo colpo di blocco incontrato.
879. Sphere Trace By Profile
880. Sphere Trace Component
881. Sphere Trace For Objects
--Components
882. Component Has Tag = Verifica se questo componente contiene il tag fornito
883. Destroy Component = Annulla la registrazione e contrassegna l'eliminazione in sospeso di un componente.
Questo non può essere utilizzato per distruggere un componente di proprietà di un attore a meno che l'attore proprietario non chiami la funzione.
884. Get Child Component = Gets the attached child component at the specified location Target is Scene Component
885. Get Children Components = Ottiene tutti i componenti collegati a questo componente, possibilmente in modo ricorsivo.
886. Get Num Children Components = Ottiene il numero di componenti secondari collegati.
887. Get Owner = Segui la catena esterna per ottenere l'AActor che "possiede" questo componente.
888. Get Parent Components = Ottiene tutti i componenti principali dell'allegato fino al root component
889. Is Component Being Destroyed = Restituisce se il componente è in fase di eliminazione.
890. Move Component To = Sposta la componente entro una certa "location"//IMPORTANTE
891. Set Is Replicated = Abilita o disabilita la replica.
-Activation
892. Activate = Attiva lo SceneComponent, dovrebbe essere sovrascritto dalle classi figlie native.
893. Deactivate = Disattiva lo SceneComponent.
894. Is Active = Restituisce se il componente è attivo o meno Target è il componente attore
895. Is in Laptop Mode = Controlla se un laptop convertibile è in modalità laptop.
896. Is in Tablet Mode = Controlla se un laptop convertibile è in modalità laptop.
897. On Component Activated = Su componente attivato: chiamato quando il componente è stato attivato, con il parametro che indica se proveniva da un ripristino
898. On Component Deactivated = Su componente disattivato: chiamato quando il componente

è stato disattivato
899. Set Active = Imposta se il componente è attivo o meno. Il target è il componente attore
900. Set Auto Activate = Imposta se il componente deve essere attivato automaticamente o meno.
901. Supports Convertible Laptops = Controlla se la piattaforma supporta laptop convertibili.
902. Toggle Active = Attiva/disattiva lo stato attivo del componente Target è componente attore
-Animation
903. Get Animation Mode = Ottieni modalità animazione L'obiettivo è il componente mesh scheletrico
904. Get Play Rate = Questi modificano lo stato dell'istanza di animazione
905. Get Position
906. Is Playing
907. Override Animation Data = Questo sovrascrive il parametro AnimationData corrente in SkeletalMeshComponent.
Questo verrà serializzato quando il componente verrà serializzato in modo che possa essere utilizzato durante lo script di costruzione
908. Play
909. Play Animation
910. Set Animation
911. Set Animation Mode
912. Set Play Rate
913. Set Position
914. Stop
-ARFace Mesh
915. Create Face Mesh = Creare la mesh frontale iniziale dai dati mesh grezzi
916. Create Face Mesh from Blend Shapes
917. Get Face Blend Shape Amount
918. Get Last Update Frame Number
919. Get Last Update Timestamp
920. Get Transform
921. Modify auto bind to local face tracking
922. Publish Via Live Link
923. Set the value of a Blend Shape
924. Update Face Mesh from Blend Shapes
925. Update Mesh Section FColor
-Arrow
926. SetArrowColor = Aggiorna il colore della freccia e le dice di aggiornare Target is Arrow Component
-Box//USATO DI PIU'
927. Get Scaled Box Extent = l'estensione del box, scalata dalla scala del componente.
928. Get Unscaled Box Extent
929. Set Box Extent = Modificare la dimensione dell'estensione della casella. Questa è la dimensione non scalata, prima che venga applicata la scala del componente.
L'obiettivo è la collisione della scatola
-Capsule
930. Get Scaled Capsule Half Height = Restituisce la mezza altezza della capsula ridimensionata dalla scala del componente.
931. Get Scaled Capsule Half Height Without Hemisphere = Restituisce la capsula a mezza

altezza meno il raggio (per escludere l'emisfero), ridimensionato in base alla scala del componente.
932. Get Scaled Capsule Radius = Restituisce il raggio della capsula ridimensionato dalla scala del componente.
933. Get Scaled Capsule Size = Restituisce il raggio della capsula e la mezza altezza scalati dalla scala del componente.
934. Get Scaled Capsule Size Without Hemisphere = Restituisce il raggio della capsula e la mezza altezza scalati dalla scala del componente.
935. Get Shape Scale = Get the scale used by this shape.
936. Get Unscaled Capsule Half Height = Restituisce la capsula a mezza altezza, ignorando il ridimensionamento del componente. Ciò include il cappuccio terminale dell'emisfero.
937. Get Unscaled Capsule Half Height Without Hemisphere = Restituisce la capsula a mezza altezza meno il raggio (per escludere l'emisfero), ignorando il ridimensionamento del componente.
938. Get Unscaled Capsule Radius = Restituisce il raggio della capsula, ignorando il ridimensionamento del componente.
939. Get Unscaled Capsule Size = Restituisce il raggio della capsula e la mezza altezza scalati dalla scala del componente.
940. Get Unscaled Capsule Size Without Hemisphere = Restituisce il raggio della capsula e la mezza altezza, ignorando il ridimensionamento del componente.
941. Set Capsule Half Height = Impostare la capsula a metà altezza. Questa è la mezza altezza non scalata, prima che venga applicata la scala del componente.
Se questa capsula si scontra, aggiorna l'array di contatto per l'attore proprietario. L'obiettivo è la collisione della capsula
942. Set Capsule Radius = Imposta il raggio della capsula. Questo è il raggio non scalato, prima che venga applicata la scala del componente.
Se questa capsula si scontra, aggiorna l'array di contatto per l'attore proprietario. L'obiettivo è la collisione della capsula
943. Set Capsule Size = Modificare la dimensione della capsula. Questa è la dimensione non scalata, prima che venga applicata la scala del componente. L'obiettivo è la Capsula Collisio
-Custom Mesh
944. Add Custom Mesh Triangles = Aggiungi alla geometria da utilizzare su questa mesh triangolare. Ciò potrebbe causare un'allocazione.
945. Clear Custom Mesh Triangles = Rimuove tutta la geometria da questa mesh triangolare
946. Set Custom Mesh Triangles = Imposta la geometria da utilizzare su questa mesh triangolare.
-Destructible
947. Apply Damage = Subisci danni Il bersaglio è un componente distruttibile
948. Apply Radius Damage = Take radius damage Target is Destructible Component
949. Get Destructible Mesh = Ottieni target mesh distruttibile è un componente distruttibile
950. On Actor Fracture = Sulla frattura dell'attore
951. On Component Fracture = Sulla frattura del componente: chiamato quando viene toccato un componente
952. Set Destructible Mesh = Imposta target mesh distruttibile è un componente distruttibile
-DMX
//DMX (Digital Multiplex) è uno standard per le reti di comunicazione digitale comunemente utilizzate per controllare l'illuminazione e gli effetti.//
953. On Fixture Patch Received = Alla ricezione della patch del dispositivo: trasmesso quando la patch del dispositivo del componente ha ricevuto DMX

-Flipbook
//In Unreal Engine, i Flipbook 2D di carta ti consentono di creare accattivanti animazioni 2D.//
954. Get Flipbook Framerate = Ottieni il framerate nominale con cui verrà riprodotto il flipbook (ignorando PlayRate),
in fotogrammi al secondo L'obiettivo è il componente Paper Flipbook
955. Get Flipbook Length = Ottieni la lunghezza del flipbook (in secondi) L'obiettivo è il componente Paper Flipbook
956. Get Flipbook Length in Frames = Ottieni la lunghezza del flipbook (in fotogrammi)
957. Get Play Rate = Ottieni la percentuale di riproduzione corrente per questo flipbook
958. Get Playback Position = Ottieni la posizione di riproduzione corrente (in secondi) del flipbook.
959. Get Playback Position in Frames = Ottieni la posizione di riproduzione corrente (in fotogrammi) del flipbook.
960. Is Looping = Scopri se stiamo effettuando un loop o meno.
961. Is Playing = Scopri se questo flipbook viene riprodotto o meno.
962. Is Reversing = Scopri se stiamo invertendo o meno
963. Play = Avvia la riproduzione del flipbook
964. Play from Start = Avvia la riproduzione del flipbook dall'inizio
965. Reverse = Avvia la riproduzione del flipbook al contrario
966. Reverse from End = Avvia la riproduzione del flipbook all'indietro dalla fine.
967. Set Looping = vero significa che dovremmo eseguire il loop, falso significa che non dovremmo.
968. Set New Time = Imposta il nuovo tempo di posizione di riproduzione da utilizzare T
969. Set Play Rate = Imposta la nuova velocità di riproduzione per questo flipbook.
970. Set Playback Position = Passa a una posizione nel flipbook (espresso in secondi).
971. Set Playback Position in Frames = Passa a una posizione nel flipbook (espresso in fotogrammi).
972. Stop = Interrompi la riproduzione del flipbook
-Geometry Cache
//La Geometry Cache è un potente strumento per gestire la geometria animata.//
973. Get Animation Time = Imposta il tempo di animazione corrente per GeometryCache.
974. Get Duration = Ottieni la durata della riproduzione.
975. Get Geometry Cache Component = Restituisce oggetto secondario GeometryCacheComponent * Il target è l'attore Geometry Cache
976. Get Motion Vector Scale = Ottieni la scala del vettore di movimento.
977. Get Number Of Frames = Ottieni il numero di frame L'obiettivo è il componente Geometry Cache
978. Get Override Wireframe Color = Controlla se stiamo sovrascrivendo il colore del wireframe o meno.
979. Get Playback Direction = Imposta il tempo di animazione corrente per GeometryCache.
980. Get Playback Speed = Ottieni la velocità di riproduzione corrente per GeometryCache..
981. Get Start Time Offset = Ottieni l'offset dell'ora di inizio corrente per GeometryCache.
982. Get Wireframe Override Color = Ottieni il colore di sostituzione del wireframe, utilizzato quando è abilitata la sostituzione del colore del wireframe.
983. Is Extrapolating Frames = Scopri se GeometryCache sta estrapolando fotogrammi. L'obiettivo è il componente Geometry Cache
984. Is Looping = Scopri se GeometryCache è in loop o meno.
985. Is Playing = Scopri se GeometryCache è in riproduzione o meno.

986. Is Playing Reversed = Scopri se GeometryCache sta riproducendo al contrario o meno.
987. Pause = Metti in pausa la riproduzione di GeometryCache
988. Play = Avvia la riproduzione di GeometryCache
989. Play from Start = Avvia la riproduzione di GeometryCache dall'inizio
990. Play Reversed = Avvia la riproduzione di GeometryCache al contrario.
991. Play Reversed from End = Avvia la riproduzione di GeometryCache dalla fine e riproduci al contrario.
992. Set Extrapolate Frames = Imposta se GeometryCache sta estrapolando i fotogrammi.
993. Set Geometry Cache = Modifica la cache geometrica utilizzata da questa istanza.
994. Set Looping = Imposta se GeometryCache è in loop o meno.
995. Set Motion Vector Scale = Imposta la nuova scala del vettore di movimento.
996. Set Override Wireframe Color = Sostituire il colore del wireframe?
997. Set Playback Speed = Imposta la nuova velocità di riproduzione per GeometryCache.
998. Set Start Time Offset = Imposta l'offset dell'ora di inizio corrente per GeometryCache.
999. Set Wireframe Override Color = Imposta il colore, utilizzato quando è abilitata la sostituzione del colore del wireframe.
1000. Stop = Interrompi la riproduzione di GeometryCache
1001. Tick at This Time = Selezionare in questo momento
-Instanced Static Mesh
//Un ISM ti consente di creare molte copie della stessa mesh statica con una sola chiamata di disegno.//
1002. Add Instance = Aggiungi un'istanza a questo componente. Target is Instanced Static Mesh Component
1003. Add Instances = Aggiungi più istanze a questo componente.
La trasformazione viene fornita nello spazio locale di questo componente a meno che non sia impostato bWorldSpace.
1004. Batch Update Instances Transform = Aggiorna la trasformazione per un numero di istanze.
1005. Batch Update Instances Transforms = Aggiorna la trasformazione per una serie di istanze.
1006. Clear Instances = Cancella tutte le istanze sottoposte a rendering da questo componente.
1007. Get Instance Count = Ottieni il numero di istanze in questo componente.
1008. Get Instance Transform = Ottieni la trasformazione per l'istanza specificata.
1009. Get Instances Overlapping Box = Restituisce le istanze con i limiti dell'istanza che si sovrappongono alla casella specificata.
1010. Get Instances Overlapping Sphere = Restituisce le istanze con i limiti dell'istanza che si sovrappongono alla sfera specificata.
1011. Is Valid Instance = L'indice fornito corrisponde a un'istanza valida in questo componente
1012. Remove Instance = Rimuovi l'istanza specificata.
1013. Remove Instances = Rimuovi le istanze specificate. Restituisce True in caso di successo.
1014. Set Cull Distances = Imposta le distanze di inizio dissolvenza e fine di selezione per questo componente.
1015. Set Custom Data Value = Aggiorna i dati personalizzati per un'istanza specifica
1016. Update Instance Transform = Aggiorna la trasformazione per l'istanza specificata.
-Lidar Point Cloud
//Le nuvole di punti LiDAR (Light Detection and Ranging) svolgono un ruolo cruciale in vari campi, dall'architettura e pianificazione urbana allo sviluppo di giochi.
1017. Get Point Cloud
1018. Get Point Shape
1019. Set Point Cloud

1020. Set Point Shape
1021. Apply Rendering Parameters
-Movement
//In Unreal Engine, i Movement Components svolgono un ruolo cruciale nel determinare il modo in cui i vari attori si muovono all'interno del mondo di gioco.
Quando si creano Blueprint basati sulla classe Character, il CharacterMovementComponent viene aggiunto automaticamente.//
1022. Get Gravity Z = Restituisce la gravità che influenza questo componente
1023. Get Max Speed = Restituisce la velocità massima del componente nella modalità di movimento corrente
1024. Get Physics Volume = Restituisce il PhysicsVolume utilizzato da questo MovementComponent o il volume fisico predefinito del mondo, se non presente.
1025. Is Exceeding Max Speed = Restituisce vero se la velocità corrente supera la velocità massima specificata (solitamente il risultato di GetMaxSpeed()),
entro una piccola tolleranza di errore.
1026. MoveUpdatedComponent = Sposta il nostro UpdatedComponent in base al Delta specificato e imposta la rotazione su NewRotation. Rispetta il vincolo del piano, se abilitato.
1027. Set Updated Component = Assegna il componente che spostiamo e aggiorniamo.
1028. Stop Movement Immediately = Arresta immediatamente il movimento (azzera la velocità, solitamente azzera l'accelerazione per i componenti con accelerazione).
1029. Stop Movement Keep Pathing = Interrompe immediatamente il movimento (reimposta la velocità) ma continua a seguire il percorso corrente.
- Planar
1030. Constrain Direction to Plane = Vincolare un vettore di direzione al vincolo del piano, se abilitato. Target is Movement Component
1031. Constrain Location to Plane = Vincolare un vettore di posizione al vincolo del piano, se abilitato.
1032. Constrain Normal to Plane = Vincolare un vettore normale (di lunghezza unitaria) al vincolo del piano, se abilitato.
1033. Get Plane Constraint Axis Setting = Ottieni l'impostazione dell'asse del vincolo del piano.
1034. Get Plane Constraint Normal = Restituisce la normale del piano che vincola il movimento, applicata se il vincolo del piano è abilitato.
1035. Get Plane Constraint Origin = Ottieni l'origine del vincolo del piano.
1036. Set Plane Constraint Axis Setting = Configurare l'impostazione dell'asse del vincolo del piano.
1037. Set Plane Constraint Enabled = Imposta se il vincolo del piano è abilitato o meno.
1038. Set Plane Constraint from Vectors = Utilizza i vettori Avanti e Su per calcolare il piano che vincola il movimento, applicato se il vincolo del piano è abilitato.
1039. Set Plane Constraint Normal = Imposta la normale del piano che vincola il movimento, applicata se il vincolo del piano è abilitato.
1040. Set Plane Constraint Origin = Imposta l'origine del piano che vincola il movimento, applicata se il vincolo del piano è abilitato.
1041. Snap Updated Component to Plane = Aggancia il componente aggiornato al vincolo del piano, se abilitato.
-Poseable Mesh
//Poseable Mesh è un componente che consente di manipolare dinamicamente la posa di una mesh scheletrica//
1042. Copy Pose from Skeletal Component = Copia posa dal componente scheletrico

1043. Get Bone Location by Name = Ottieni la posizione dell'osso per nome
1044. Get Bone Rotation by Name = Ottieni rotazione ossea per nome
1045. Get Bone Scale by Name = Ottieni scala ossea per nome
1046. Get Bone Transform by Name = Ottieni trasformazione ossea per nome
1047. Reset Bone Transform by Name = Reimposta trasformazione ossea in base al nome
1048. Set Bone Location by Name = Imposta la posizione dell'osso per nome
1049. Set Bone Rotation by Name = Imposta la rotazione ossea per nome
1050. Set Bone Scale by Name = Imposta scala ossea in base al nome
1051. Set Bone Transform by Name = Imposta trasformazione ossea in base al nome
-Procedural Mesh
//Il Procedural Mesh Component (PMC) è un componente blueprint che consente di generare e modificare la geometria della mesh a livello di codice.//
1052. Add Collision Convex Mesh
1053. Calculate Tangents for Mesh
1054. Clear All Mesh Sections
1055. Clear Collision Convex Meshes
1056. Clear Mesh Section
1057. Convert Quad to Triangles
1058. Copy Procedural Mesh from Static Mesh Component
1059. Create Grid Mesh Split
con UV 0-1 standard su UV0 e UV centrali texel campionati a punti per UV1.
1060. Create Grid Mesh Triangles
1061. Create Grid Mesh Welded
1062. Create Mesh Section
1063. Generate Box Mesh
Per ciascun vertice vengono generate anche normali, UV e tangenti
1064. Get Num Sections
1065. Get Section from Procedural Mesh
1066. Get Section from Static Mesh
1067. Is Mesh Section Visible
1068. Set Mesh Section Visible
1069. Slice Procedural Mesh
1070. Slice Procedural Mesh
-Skeletal Mesh
//Skeletal Mesh è una risorsa fondamentale utilizzata per creare personaggi e altri oggetti animati//
1071. Allow Anim Curve Evaluation
1072. Clear Morph Targets
1073. Get Allow Cloth Actors
1074. Get Allow Rigid Body Anim Node
1075. Get Allowed Anim Curve Evaluate
1076. Get Anim Instance
1077. Get Closest Point On Physics Asset
1078. Get Default Animating Rig
1079. Get Disable Anim Curves
1080. Get LODSync Debug Text
1081. Get Morph Target
1082. Get Post Process Instance

1083. Get Socket Location
1084. Has Valid Animation Instance
1085. Initialize Socket from Location
1086. Reset Allowed Anim Curve Evaluation
1087. Reset Anim Instance Dynamics
1088. Set Allow Anim Curve Evaluation
1089. Set Allow Cloth Actors
1090. Set Allow Rigid Body Anim Node
1091. Set Allowed Anim Curves Evaluation
1092. Set Anim Instance Class
1093.Set Disable Anim Curves
1094. Set Morph Target
1095. Set Update Animation in Editor
1096. Set Update Cloth in Editor
1097. Snapshot Pose
1098. Toggle Disable Post Process Blueprint
-Animation Blueprint Linking
1099. Get Linked Anim Graph Instance by Tag
1100. Get Linked Anim Layer Instance by Class
1101. Get Linked Anim Layer Instance by Group
1102. Link Anim Class Layers
1103. Link Anim Graph by Tag
1104. Unlink Anim Class Layers
-Skinned Mesh
//Una Skinned Mesh è un tipo di mesh utilizzata per personaggi e altri oggetti animati.//
1105. Bone Is Child Of
1106. Clear Skin Weight Override
1107. Clear Skin Weight Profile
1108. Clear Vertex Color Override
1109. Find Closest Bone
1110. Get Bone Index
1111. Get Bone Name
1112. Get Current Skin Weight Profile Name
1113. Get Delta Transform from Ref Pose
1114. Get Forced LOD
1115. Get Num Bones
1116. Get Num LODs
1117. Get Parent Bone
1118. Get Ref Pose Position
1119. Get Ref Pose Transform
1120. Get Socket Bone Name
1121.Get Twist and Swing Angle Of Delta Rotation from Ref Pose
1122. Get Vertex Offset Usage
1123. Hide Bone by Name
1124. Is Bone Hidden by Name
1125. Is Material Section Shown
1126. Is Using Skin Weight Profile
1127. Set Forced LOD

1128. Set Master Pose Component
1129. Set Mesh Deformer
1130.Set Min LOD
1131. Set Physics Asset
1132. Set Post Skinning Offsets
1133. Set Pre Skinning Offsets
1134. Set Render Static
1135. Set Skeletal Mesh
1136. Set Skin Weight Override
1137. Set Skin Weight Profile
1138. Set Vertex Color Override
1139. Set Vertex Offset Usage
1140. Show All Material Sections
1141. Show Material Section
1142. Transform from Bone Space
1143. Transform to Bone Space
1144. Un Hide Bone by Name
1145. Unload Skin Weight Profile
-Sphere
1146. Get Scaled Sphere Radius = Target is Sphere Collision
1147. Get Shape Scale = Ottieni la scala utilizzata da questa forma. Questa è una scala uniforme che è il minimo di qualsiasi scala non uniforme
1148. Get Unscaled Sphere Radius = Target is Sphere Collision
1149. Set Sphere Radius = Modificare il raggio della sfera. Questo è il raggio non scalato, prima che venga applicata la scala del componente.
-Sprite
//Sprite è una grafica 2D che può essere creata e renderizzata nel mondo.//
1150. Add Instance
1151. Clear Instances
1152. Get Instance Count
1153. Get Instance Transform
1154. Remove Instance
1155. Sort Instances Along Axis
1156. Update Instance Color
1157. Update Instance Transform
-Static Mesh
1158. Get Local Bounds = Ottieni il limite locale
1159. Set Static Mesh = Modifica la StaticMesh utilizzata da questa istanza.
-Stereo Layer
//In Unreal Engine, gli Stereo Layers sono una funzionalità utilizzata nello sviluppo della realtà virtuale (VR)
1160. Get Left Texture Target is Stereo Layer
1161. Get Priority Target is Stereo Layer
1162. Get Quad Size Target is Stereo Layer
1163. Get Texture Target is Stereo Layer
1164. Get UVRect Target is Stereo Layer
1165. Mark Texture for Update
1166. Set Anti Alias

1167. Set Auto Curve Max Distance
1168. Set Auto Curve Min Distance
1169. Set Curved
1170. Set Equirect Props
1171. Set Height
1172. Set Left Texture
1173. Set Overlay Arc
1174. Set Priority
1175. Set Quad Size
1176. Set Radius
1177. Set Texture
1178. Set UVRect
-Tick
//In Unreal Engine (UE), la funzione TickComponent fa parte della classe UActorComponent1. Questa funzione si chiama ogni frame ed è dove inseriresti la logica che deve essere aggiornata frequentemente.//
1179. Add Tick Prerequisite Actor = Fai in modo che questo componente spunti dopo PrerequisiteActor
1180. Add Tick Prerequisite Component = Fai in modo che questo componente spunti dopo PrerequisiteComponent.
1182. Get Component Tick Interval = Restituisce l'intervallo di tick per la funzione tick primaria di questo componente,
ovvero la frequenza in secondi con cui verrà eseguita Target is Actor Component
1183. Is Component Tick Enabled
1184. Remove Tick Prerequisite Actor
1185. Remove Tick Prerequisite Component = Rimuovere la dipendenza dal segno di spunta su PrerequisiteComponent.
1186. Set Component Tick Enabled = Imposta le funzioni di spunta di questo componente su abilitate o disabilitate
1187. Set Component Tick Interval = Imposta l'intervallo di tick per la funzione di tick primaria di questo componente.
Non abilita l'intervallo di tick. Ha effetto al tick successivo.
1188. Set Component Tick Interval and Cooldown = Imposta l'intervallo di tick per la funzione di tick primaria di questo componente.
1189. Set Tick Group = Modifica il gruppo di spunta per questo componente
1190. Set Tickable when Paused = Imposta se questo componente può spuntare quando è in pausa.
-Timeline
//In Unreal Engine (UE), un componente Timeline è un nodo speciale all'interno di Blueprints che fornisce animazioni basate sul tempo.
Può essere rapidamente progettato e riprodotto in base a eventi, numeri mobili, vettori o colori che possono essere attivati nei fotogrammi chiave lungo la timeline.
Le sequenze temporali vengono modificate direttamente all'interno dell'editor del BP facendo doppio clic sulla sequenza temporale
Input Pins:

Play: Fa sì che la sequenza temporale venga riprodotta in avanti a partire dall'ora corrente.
Play from Start: Fa sì che la sequenza temporale venga riprodotta in avanti dall'inizio.

Stop: Blocca la riproduzione della Timeline all'ora corrente
Reverse: Riproduce la timeline all'indietro a partire dall'ora corrente.
Reverse from End: Riproduce la sequenza temporale all'indietro a partire dalla fine.
Set New Time: Imposta l'ora corrente sul valore impostato (o immesso)
New Time: Questo pin di dati assume un valore float che rappresenta il tempo in secondi, a cui la timeline può saltare quando viene chiamato l'input Imposta nuova ora.

Output Pins:

Update: Emette un segnale di esecuzione non appena viene richiamata la timeline.
Finished: Emette un segnale di esecuzione al termine della riproduzione. Ciò non viene attivato dalla funzione Stop.
Direction: Emette dati enumerativi che indicano la direzione in cui viene riprodotta la timeline.//
1191. Get Ignore Time Dilation = Scopri se ignorare la dilatazione del tempo.
1192. Get Play Rate = Ottieni la velocità di riproduzione corrente per questa sequenza temporale
1193. Get Playback Position = Ottieni la posizione di riproduzione corrente della destinazione della timeline è il componente della timeline
1194. Get Scaled Timeline Length = Ottieni la lunghezza della sequenza temporale divisa per la velocità di riproduzione.
1195. Get Timeline Length = Ottieni la lunghezza della sequenza temporale
1196. Is Looping
1197. Is Playing
1198. Is Reversing
1199. Play = Avvia la riproduzione della timeline
1200. Play from Start = Avvia la riproduzione della timeline dall'inizio.
1201. Reverse = Avvia la riproduzione della timeline al contrario.
1202. Reverse from End = Avvia la riproduzione della timeline al contrario dalla fine.
1203. Set Float Curve = Aggiorna la curva di una determinata traccia mobile
1204. Set Ignore Time Dilation
1205. Set Linear Color Curve
1206. Set Looping
1207. Set New Time
1208. Set Play Rate
1209. Set Playback Position
1210. Set Timeline Length
1211. Set Timeline Length Mode
1212. Set Vector Curve
1213. Stop

.Audio Capture
1214. Create Audio Capture = Crea destinazione di acquisizione audio è la libreria delle funzioni di Audio CAPTURE
1215. Get Audio Capture Device Info = Restituisce le informazioni sul dispositivo di Audio Capture
1216. Is Capturing Audio = Restituisce vero se l'acquisizione dell'audio
1217. Start Capturing Audio = Inizia a catturare l'audio
1218. Stop Capturing Audio = Interrompe l'acquisizione dell'audio

--Event Dispatcher Audio--
1219. On Audio Envelope Value
1220. On Audio Fade Change Event = Se vincolato,
trasmette all'inizio della dissolvenza (con tempo di dissolvenza) invece di alterare manualmente direttamente il volume principale del dispositivo audio
1221. On Audio Finished = Chiamato al termine della riproduzione dell'audio, perché è stato riprodotto fino al completamento o perché una chiamata Stop()
lo ha disattivato in anticipo
1222. On Audio Multi Envelope Value
1223. On Audio Play State Changed = Questa funzione restituisce lo stato di riproduzione corrente del componente audio di destinazione.
In riproduzione, se il suono è attualmente in riproduzione. Interrotto, se il suono viene interrotto. In pausa, se l'audio è attualmente in riproduzione, ma in pausa.
Fading In, se il suono è in fase di Fading In. Dissolvenza in chiusura, se il suono è in fase di dissolvenza in chiusura.
1224. On Audio Playback Percent = Chiamato mentre un suono viene riprodotto sul componente audio per consentire a BP di eseguire azioni in base alla percentuale di riproduzione.
Calcolato come campioni riprodotti diviso per i campioni totali, tenendo conto dell'altezza.
Attualmente non implementato su tutte le piattaforme.
1225. On Audio Single Envelope Value
1226. On Audio Virtualization Changed = Chiamato quando cambia lo stato della virtualizzazione
//Virtualization In Unreal Engine, il concetto di virtualizzazione si riferisce a varie tecniche e funzionalità che ottimizzano l'utilizzo della memoria,
migliorano le prestazioni e migliorano la scalabilità.

.Variables
1227. Get a reference to self = Ottiene un riferimento a questa istanza del blueprint
1228. Property Access = Accede alle proprietà in base al percorso della proprietà
1229. Set By-Ref Var = Imposta il valore della variabile pass-by-ref//Il più bello dei nodi

.Math
1230. Make Bitmask = Ciò consente di creare e dichiarare una variabile intera come Bitmask
//
Una Bitmask è un modo per organizzare i flag binari in modo compatto.
Pensa alle Bitmask come a una rappresentazione compatta di più flag booleani.
Vengono trattati come numeri interi nei grafici Blueprint, rendendoli versatili per vari scenari.
In informatica, un flag booleano (noto anche come bit di verità o flag di verità) è un valore booleano rappresentato da uno o più bit
Codifica una variabile di stato con due possibili valori: vero o falso.
Un singolo byte può contenere fino a 8 flag booleani separati mappando un flag booleano su ciascun bit.
Questo approccio è noto come bit-packing o rappresentazione compress
La parola deriva dall’inglese “binary digit” e può contenere i due simboli del sistema binario: zero (0) e uno (1).
Il bit è quindi una cifra binaria ed è considerata la più piccola unità di informazione.
I bit sono utilizzati principalmente per rappresentare l’utilizzo dei dati e la velocità di trasmissione di Internet, telefoni e servizi di streaming.

I byte, d’altra parte, esprimono le capacità di memoria
Bit: 1 bit
Byte: 8 bit
Tra i bit e i byte, c’è un’unità intermedia chiamata nibble, che di solito è l’equivalente di 4 bit.
In breve, il bitmask è un concetto correlato, utilizzato per manipolare posizioni multiple di valori.
Gli enumeratori possono essere utilizzati per creare un set limitato di valori che rappresentano uno stato, una scelta o una condizione.//
.Math Boolean
1231. AND Boolean = Restituisce l'AND logico di due valori (A AND B), restituisce 1
False 0, se compari True 1 e False 0, il risultato è 0 = Falso
1232. Make Literal Bool = Crea un valore bool letterale
//Boolean Literals:
true: rappresenta il valore booleano true.
false: rappresenta il valore booleano false.//
1233. NAND Boolean = Restituisce la NAND logica di due valori (A AND B), Mentre AND restituisce vero (1) solo quando entrambi gli input sono veri (1), NAND fa il contrario.
Restituisce 0
1234. NOR Boolean = Restituisce il Not OR logico di due valori (A NOR B),
1235. NOT Boolean = Restituisce il complemento logico del valore booleano (NON A)
1236. OR Boolean = Restituisce l'OR logico di due valori (A OR B) uno dei due deve essere 1 per restiuire True.
1237. XOR Boolean = Restituisce l'OR esclusivo logico di due valori (A XOR B), Restituisce 1 solo se entrambi i bit di input sono 0
.Math Box
1238. Make Box = Crea un FBox da Min e Max e imposta IsValid su true è una Struct
.Math Box 2D
1239. Make Box 2D = Crea un FBox2D da Min e Max e imposta IsValid su true è una Struct
.Box Sphere Bounds
1240. Break Box Sphere Bounds = Suddivide un FBoxSphereBounds in origine, estensione e raggio
1241. Make Box Sphere Bounds = Crea un FBoxSphereBounds con origine, estensione e raggio
.Math Byte
//1 byte = 8 bit = 256 caratteri
Il bit è la più piccola unità nella trasmissione di dati digitali e si compone di due sole condizioni di informazioni: 1 o 0.
Un byte consiste di 8 bit ed è abbreviato con la lettera “B”.
A differenza del bit, che ha solo due possibili condizioni (1 o 0), il byte può rappresentare ben 256 (2^8) condizioni diverse.
Ogni bit all’interno del byte può essere 1 o 0, e ci sono 8 possibilità di posizionare un valore 1 nel byte.//
1242. % (Byte) = Modulo (A % B)
1243. Make Literal Byte = Crea un byte letterale
1244. Max (Byte) = Restituisce il valore massimo di A e B
1245. Max Of Byte Array = Restituisce il massimo di tutte le voci dell'array e l'indice in cui è stato trovato. Restituisce il valore 0 e l'indice -1 se l'array fornito è vuoto.
1246. Min (Byte) = Restituisce il valore minimo di A e B
1247. Min Of Byte Array = Restituisce il minimo di tutte le voci dell'array e l'indice in cui è stato trovato. Restituisce il valore 0 e l'indice -1 se l'array fornito è vuoto.

.Math Color
1248. Break Color
1249. Desaturate (LinearColor) = Restituisce un colore desaturato, dove 0 indica nessuna desaturazione e 1 == desaturazione completa
1250. Distance (LinearColor) = Distanza euclidea tra due punti di colore.
1251. HSV to RGB = Crea un colore dai singoli componenti del colore (spazio HSV; la tonalità è [0..360) mentre la saturazione e il valore sono 0..1)
1251. HSV to RGB (Vector) = Converte un colore lineare HSV (dove H è in R (0..360), S è in G (0..1) e V è in B (0..1)) in RGB
1252. HSV to RGB linear color = Converte un colore lineare HSV (dove H è in R, S è in G e V è in B) in RGB lineare
1253. Lerp (LinearColor) = Interpola linearmente tra A e B in base ad Alpha (100% di A quando Alpha=0 e 100% di B quando Alpha=1)
1254. Lerp Using HSV (LinearColor) = Interpola linearmente tra due colori in base alla quantità Alpha specificata (100% di A quando Alpha=0 e 100% di B quando Alpha=1).
1255. Linear Color Black = Colore lineare nero
1256. Linear Color Blue = Colore lineare blu
1257. Linear Color Gray
1258. Linear Color Green
1259. Linear Color Red
1260. Linear Color Set
1261. Linear Color Set from HSV = Assegna un colore HSV a un colore RGB di uno spazio lineare.
1262. Linear Color Set from Pow 22 = Assegna un FColor proveniente da un output Pow(1/2.2) osservato, in un colore lineare.
1263. Linear Color Set from SRGB = Assegna un FColor proveniente da un output sRGB osservato, in un colore lineare.
1264. Linear Color Set Random Hue = Imposta un colore casuale. Sceglie un colore piuttosto carino basato su una tonalità casuale.
1265. Linear Color Set RGBA = Assegna singoli componenti RGBA lineari.
1266. Linear Color Set Temperature = Converte la temperatura in Kelvin di un radiatore del corpo nero in cromaticità RGB.
1267. Linear Color Transparent = Colore lineare trasparente: nero con 0 opacità/alfa
1268. Linear Color White = Il colore lineare bianco
1269. Linear Color Yellow = Colore lineare giallo
1270. Luminance (LinearColor) = Restituisce la luminosità percepita di un colore su un display tenendo conto dell'impatto sull'occhio umano per canale di colore: verde > rosso > blu.
1271. Make Color = Crea un colore da singoli componenti di colore (spazio RGB)
1272. Max (LinearColor) = Restituisce il valore massimo del canale di colore in questa struttura di colore.
1273. Min (LinearColor) = Restituisce il valore minimo del canale colore in questa struttura colore.
1274. Near Equal (LinearColor) = Restituisce vero se il colore lineare A è uguale al colore lineare B (A == B) entro una tolleranza di errore specificata
1275. New Opacity (LinearColor) = Restituisce una copia di questo colore utilizzando l'opacità/alfa specificata.
1276. Quantize With Rounding To 8-bit (LinearColor) = Quantizza il colore lineare con arrotondamento e restituisce il risultato come colore a 8 bit. Ciò ignora la conversione SRGB.
1277. RGB to HSV = Suddivide un colore in singoli componenti HSV (oltre ad alfa) (la tonalità è

[0..360) mentre la saturazione e il valore sono 0..1)
1278. RGB to HSV (LinearColor) = Converte un colore lineare RGB in HSV (dove H è in R, S è in G e V è in B)
1279. RGB to HSV (Vector) = Converte un colore lineare RGB in HSV (dove H è in R (0..360), S è in G (0..1) e V è in B (0..1))
1280. Select Color = Se bPickA è vero, viene restituito A, altrimenti B
1281. To RGBE (LinearColor) = Converte da RGBE lineare a 8 bit come indicato nell'articolo Real Pixels di Gregory Ward, Graphics Gems II, pagina 80.
1282. ToHex = Converte questo valore di colore in una stringa esadecimale. Il formato della stringa è RRGGBBAA.

MATH .....

////////////

-Access Control Operators
Le classi C++ hanno operatori di controllo dell'accesso che designano l'ambito (scope) dei membri della classe:
1. public, i membri sono accessibili ovunque.
2. private è possibile accedere ai membri solo dalla stessa istanza della classe o delle classi correlate.
Esempio :

```
class City {
  int population;

  public:
    void add_resident() {
      population++;
    }

  private:
    bool is_capital;
};
```

-Overloading
La ridefinizione degli operatori affinché funzionino in modo diverso con i tipi di dati personalizzati è denominata sovraccarico degli operatori in C++.
Ciò consente all'utente di utilizzare questi operatori con oggetti delle proprie classi.
Esempio esegue l'overload degli operatori + e ==:

```
#include <iostream>
using namespace std;

class Point {
public:
  int x;
  int y;

public:
  Point(int x = 0, int y = 0) : x(x), y(y) {}

  Point operator+(const Point& other) const {
    return Point(x + other.x, y + other.y);
  }

  bool operator==(const Point& other) const {
    return (x == other.x) && (y == other.y);
  }

  void display() const {
    std::cout << "Point(" << x << ", " << y << ")\n";
  }
};
```

```
int main() {
  Point point1(3, 4);
  Point point2(1, 2);

  Point sum = point1 + point2;
  sum.display();

  Point point3(5, 2);
  Point point4(3, 6);

  bool isEqual = (point3 == point4);
  std::cout << "Points are equal: " << (isEqual ? "true" : "false") << "\n";
}
```

.Structures, Una struttura in C++ memorizza insieme gli elementi di dati sotto un unico nome. Gli elementi dati, detti anche membri dati, possono essere di diversi tipi di dati.
Una struttura è definita con:
1. La parola chiave struct all'inizio.
2. Parentesi graffe { } per definire il corpo.
3. Un punto e virgola; alla fine.
Esempio :

```
struct name {
  member1_type member1_name;
  member2_type member2_name;
  member3_type member3_name;
};

struct coder {
  long id;
  char name[30];
  char username[15];
};
```

-Declaring and Initializing Structure Variables
I membri di una struttura possono essere inizializzati all'interno della definizione della struttura in C++11 e versioni successive.
Esempio

```
#include <iostream>

struct coordinates {
  int x;
  int y;
};

int main() {
  // Declaring variable c1
  struct coordinates c1;
```

```
  c1.x = 2;
  c1.y = 2;

  // Declaring and initializing variable c2
  struct coordinates c2 = { 0, 1 };

  std::cout << "x1 = " << c1.x << ", y1 = " << c1.y << "\n"; // For variable c1
  std::cout << "x2 = " << c2.x << ", y2 = " << c2.y << "\n"; // For variable c2

  return 0;
}
```

-Array of Structure
Come altri tipi di dati primitivi, è possibile creare un array di strutture.
Esempio :

```
#include <iostream>

struct coordinates {
  int x;
  int y;
};

int main() {
  struct coordinates c[4];
  int i;

  for (i = 0; i < 4; i++)
    std::cin >> c[i].x >> c[i].y;

  for (i = 0; i < 4; i++) {
    std::cout << "x" << i+1 << "=" << c[i].x << "\t";
    std::cout << "y" << i+1 << "=" << c[i].y << "\n";
  }

  return 0;
}
```

-Nested Structure(Strutture annidate)
La variabile di una struttura può anche essere membro di, o annidata in, un'altra struttura.
Esempio :

```
struct coder {
  long id;
  char name[30];
  char username[15];
  struct DOB {
    int month;
    int day;
```

```
    int year;
  } d;
} bjarne;
```

La d è la variabile della struttura per struct DOB e bjarne è la variabile della struttura per struct coder.
È possibile accedere ai membri di una struttura nidificata utilizzando var1.var2.member1.

Inizializzazione di un membro della struttura DOB dall'esempio sopra:
bjarne.d.day = 10;

.Constructors
Un costruttore è un metodo appositamente definito in una classe C++ che viene chiamato automaticamente quando viene creata un'istanza di quella classe.
Viene in genere utilizzato per attività come l'inizializzazione degli attributi di classe in un nuovo oggetto.
Come con le funzioni, un costruttore può accettare argomenti che possono aiutare nell'inizializzazione.
Esempio :

```
class MyClass {
  public:
    // Define a constructor that outputs a message
    MyClass() {
      std::cout << "The constructor was executed!";
    }
};

int main() {
  MyClass myObj;  // This calls the constructor
  return 0;
}
```

-Esempio 2 con argomenti

```
class Circle {
  public:
    int X;
    int Y;
    int radius;

    Circle (int a, int b, int c) {
      X = a;
      Y = b;
      radius = c;
    }
};

int main() {
  // Call the constructor using arguments
  Circle myCircle1(5,5,10);
```

```
  Circle myCircle2(0,0,5);
```

.Macros
Una macro è un'etichetta definita nel codice sorgente che viene sostituita dal suo valore dal preprocessore prima della compilazione.
Le macro vengono inizializzate con il comando #define del preprocessore e possono essere indefinite con il comando #undef.
Esistono due tipi di macro:
1.Object-Like Macros
Queste macro vengono sostituite dal loro valore nel codice sorgente prima della compilazione. Il loro scopo principale è definire le costanti da utilizzare nel codice.
Esempio :

```
#include <iostream>

using namespace std;

#define PI 3.1416//Macro

int main() {
  float radius = 3;
  float area;

  area = PI * radius * radius;

  cout << "Area is " << area;

  return 0;
}
```

2. Function-Like Macros
Queste macro si comportano come funzioni, nel senso che accettano argomenti utilizzati nel codice sostituito.
Esempio :

```
#include <iostream>

using namespace std;

#define PI 3.1416//Prima Macro definita
#define AREA(r) r * r * PI//Seconda Macro definita con utilizzo della prima

int main() {
  float radius = 5;
  float result;

  result = AREA(radius);

  cout << "Area is " << result;

  return 0;
```

}
-Undefining a Macro
Una volta definita, una macro può essere annullata con il comando #undef.
#define TEST 1
#undef TEST

.Templates
Ciò offre la possibilità di definire un insieme di classi o funzioni correlate che possono operare su molti tipi diversi con un'unica dichiarazione.
I modelli sono come le macro in quanto vengono espansi in fase di compilazione. Anche se l'origine ha una sola definizione.
-Syntax
template <class typeidentifier> function;
Dove typeidentifier è un nome di variabile che sostituisce il tipo generico e function è una dichiarazione di funzione,
utilizzando typeidentifier come tipo generico nella sua definizione.
Esempio :

```
#include <iostream>
using namespace std;

template <class T> T myAdd(T x, T y) {
  T result;

  result = x + y;

  return result;
}

int main() {
  cout << myAdd <int> (3, 7) << endl;
  cout << myAdd <double> (3.5, 7.2) << endl;
  cout << myAdd <string> ("Hello", " World!") << endl;

  return 0;
}
```

-Class Templates
I modelli di classe sono classi definite insieme a un tipo generico che può essere applicato in tutta la definizione della classe.
-Syntax
template <class typeidentifier> class;
Esempio :

```
#include <iostream>
using namespace std;

template <class T>
class myPair {
  T x, y;
```

```
public:
  myPair (T first, T second) {//Costruttore del template di classe
    x = first;
    y = second;
  }

public:
  T add ()
  {
    T result;
    result = x + y;
    return result;
  }
};

int main() {
  myPair <int> intPair(3, 7);//Chiamate del template
  myPair <double> doublePair(3.5, 7.2);
  myPair <string> stringPair("Hello", " World!");

  cout << intPair.add() << endl;
  cout << doublePair.add() << endl;
  cout << stringPair.add() << endl;

  return 0;
}
```

.Methods
I metodi della classe C++ sono funzioni definite dall'utente che possono essere utilizzate all'interno di un'istanza della classe.
-Class Methods
Un metodo di classe può essere definito in due modi:

a. All'interno della definizione della classe
b. Al di fuori della definizione di classe

A) Inside the Class

```
class Person {
  string name;

  public:
    // Defines the method
    void get_name() {
      return name;
    }
}

int main() {
```

```
  Person robert;

  // Calls the method
  robert.get_name();

  return 0;
}
```

B) Outside the Class

```
class Person {
  string name;

  public:
    void get_name();
}

// Defines the method
void Person::get_name() {
  return name;
}

int main() {
  Person robert;

  // Calls the method
  robert.get_name();

  return 0;
}
```

I parametri possono anche essere aggiunti ai metodi della classe:

```
class Person {
  string name;

  public:
    // Defines the method
    void set_name(string newName) {
      name = newName;
    }

    void get_name() {
      return name;
    }
}

int main() {
  Person robert;
```

```
  // Sets the name class member
  robert.set_name("Robert");

  // Prints "Robert"
  std::cout << robert.get_name();

  return 0;
}
```
//

Ricerca di CPP UE
CPPUE

.Class Setup

Il file di intestazione, MioAttore.h, contiene la dichiarazione della classe.

```
UCLASS()
   class [PROJECTNAME]_API AMioAttore : public AActor
   {
   GENERATED_BODY()

   };
```

Le dichiarazioni di classe create con la Creazione guidata classe C++ sono automaticamente precedute dalla macro UCLASS().
La macro UCLASS() rende il motore consapevole della classe.

La dichiarazione della classe (file.h) contiene eventuali dichiarazioni di variabili e/o funzioni. Questi possono essere preceduti rispettivamente dalle macro UPROPERTY() e UFUNCTION(), che svolgono funzioni simili alla macro UCLASS(). Anche i componenti vengono impostati con le macro UPROPERTY().

Dichiarare PointLightComponent e SphereComponent. Il PointLightComponent e il SphereComponent vengono resi VisibleAnywhere. Ciò significa che le loro proprietà saranno visibili nella scheda Dettagli di un attore LightSwitchCodeOnly.

```
 public:
   /** point light component */
   UPROPERTY(VisibleAnywhere, Category = "Switch Components")
   class UPointLightComponent* PointLight1;

   /** sphere component */
   UPROPERTY(VisibleAnywhere, Category = "Switch Components")
   class USphereComponent* Sphere1;
```

Dichiara il tuo costruttore, in modo da poter impostare valori predefiniti per i tuoi componenti e variabili:

```
AMioAttore();
```

Dichiara OnOverlapBegin e OnOverlapEnd, le funzioni che verranno chiamate quando un altro attore entra o esce dallo SphereComponent.

```
/** called when something enters the sphere component */
   UFUNCTION()
   void OnOverlapBegin(class UPrimitiveComponent* OverlappedComp, class AActor*
OtherActor, class UPrimitiveComponent* OtherComp, int32 OtherBodyIndex, bool
bFromSweep, const FHitResult& SweepResult);

   /** called when something leaves the sphere component */
   UFUNCTION()
   void OnOverlapEnd(class UPrimitiveComponent* OverlappedComp, class AActor* OtherActor,
class UPrimitiveComponent* OtherComp, int32 OtherBodyIndex);
```

Dichiarare ToggleLight, una funzione che attiva/disattiva la visibilità di PointLightComponent.

```
  /** Toggles the light component's visibility*/
   UFUNCTION()
   void ToggleLight();
```

Dichiara la variabile DesiredIntensity e rendila visibile ovunque con l'identificatore VisibleAnywhere. Verrà visualizzato nella categoria Cambia variabili nella scheda Dettagli di un attore LightSwitchCodeOnly. Per le variabili che non sono oggetti secondari, come questo valore float, l'identificatore VisibleAnywhere fa semplicemente visualizzare la variabile nella scheda Dettagli. È possibile utilizzare anche un identificatore EditAnywhere, ma poiché la variabile DesiredIntensity viene utilizzata solo nel momento in cui l'attore viene aggiunto al livello, non è necessario che la variabile sia modificabile.

```
 /** the desired intensity for the light */
   UPROPERTY(VisibleAnywhere, Category="Switch Variables")
   float DesiredIntensity;
```

<u>RISULTATO FINALE</u>

AMioAttore.h

```
#pragma once

   #include "GameFramework/Actor.h"
   #include "AMioAttore.generated.h"

   /**
    *
    */
   UCLASS()
   class [PROJECTNAME]_API AMioAttore : public AActor
```

```
{
    GENERATED_BODY()
    public:
    /** point light component */
    UPROPERTY(VisibleAnywhere, Category = "Switch Components")
    class UPointLightComponent* PointLight1;

    /** sphere component */
    UPROPERTY(VisibleAnywhere, Category = "Switch Components")
    class USphereComponent* Sphere1;

    AMioAttore();

    /** called when something enters the sphere component */
    UFUNCTION()
    void OnOverlapBegin(class UPrimitiveComponent* OverlappedComp, class AActor*
OtherActor, class UPrimitiveComponent* OtherComp, int32 OtherBodyIndex, bool
bFromSweep, const FHitResult& SweepResult);

    /** called when something leaves the sphere component */
    UFUNCTION()
    void OnOverlapEnd(class UPrimitiveComponent* OverlappedComp, class AActor*
OtherActor, class UPrimitiveComponent* OtherComp, int32 OtherBodyIndex);

    /** Toggles the light component's visibility*/
    UFUNCTION()
    void ToggleLight();

    /** the desired intensity for the light */
    UPROPERTY(VisibleAnywhere, Category = "Switch Variables")
    float DesiredIntensity;

};
```

-La procedura guidata della classe C++ crea anche il file sorgente per la classe, in questo caso AMioAttore.cpp.

Inizierai aggiungendo il costruttore della classe.

```
AMioAttore::AMioAttore()
{

    DesiredIntensity = 3000.0f;
    PointLight1 = CreateDefaultSubobject<UPointLightComponent>(TEXT("PointLight1"));
    PointLight1->Intensity = DesiredIntensity;
    PointLight1->bVisible = true;
    RootComponent = PointLight1;
```

```
        Sphere1 = CreateDefaultSubobject<USphereComponent>(TEXT("Sphere1"));
        Sphere1->InitSphereRadius(250.0f);
        Sphere1->SetupAttachment(RootComponent);
       Sphere1->OnComponentBeginOverlap.AddDynamic(this,
&AMioAttore::OnOverlapBegin);      // set up a notification for when this component overlaps
something
   Sphere1->OnComponentEndOverlap.AddDynamic(this, &AMioAttore::OnOverlapEnd);

   }
```

Risulato finale
ALightSwitchCodeOnly da sostiture con AMioAttor

```
 // Copyright 1998-2018 Epic Games, Inc. All Rights Reserved.

   #include "BasicClasses.h"
   #include "LightSwitchCodeOnly.h"

   ALightSwitchCodeOnly::ALightSwitchCodeOnly()
   {
      DesiredIntensity = 3000.0f;

      PointLight1 = CreateDefaultSubobject<UPointLightComponent>(TEXT("PointLight1"));
      PointLight1->Intensity = DesiredIntensity;
      PointLight1->bVisible = true;
      RootComponent = PointLight1;

      Sphere1 = CreateDefaultSubobject<USphereComponent>(TEXT("Sphere1"));
      Sphere1->InitSphereRadius(250.0f);
      Sphere1->SetupAttachment(RootComponent);

      Sphere1->OnComponentBeginOverlap.AddDynamic(this,
&ALightSwitchCodeOnly::OnOverlapBegin);      // set up a notification for when this component
overlaps something
      Sphere1->OnComponentEndOverlap.AddDynamic(this,
&ALightSwitchCodeOnly::OnOverlapEnd);      // set up a notification for when this component
overlaps something

   }

   void ALightSwitchCodeOnly::OnOverlapBegin(class UPrimitiveComponent* OverlappedComp,
class AActor* OtherActor, class UPrimitiveComponent* OtherComp, int32 OtherBodyIndex, bool
bFromSweep, const FHitResult& SweepResult)
   {
      if (OtherActor && (OtherActor != this) && OtherComp)
      {
         ToggleLight();
      }
```

```
}

void ALightSwitchCodeOnly::OnOverlapEnd(class UPrimitiveComponent* OverlappedComp,
class AActor* OtherActor, class UPrimitiveComponent* OtherComp, int32 OtherBodyIndex)
{
    if (OtherActor && (OtherActor != this) && OtherComp)
    {
        ToggleLight();
    }
}

void ALightSwitchCodeOnly::ToggleLight()
{
    PointLight1->ToggleVisibility();
}
```

//

Property Declaration

Le proprietà vengono dichiarate utilizzando la sintassi delle variabili C++ standard, preceduta dalla macro UPROPERTY che definisce i metadati delle proprietà e gli identificatori delle variabili.

UPROPERTY([specifier, specifier, ...], [meta(key=value, key=value, ...)])
Type VariableName;

-Integers

Variable Type	Description
uint8	8-bit unsigned
uint16	16-bit unsigned
uint32	32-bit unsigned
uint64	64-bit unsigned
int8	8-bit signed
int16	16-bit signed
int32	32-bit signed
int64	64-bit signed

-Come Bitmasks

Le proprietà dei numeri interi ora possono essere esposte all'editor come maschere di bit

```
/*~ BasicBits appears as a list of generic flags in the editor, instead of an integer field. */
UPROPERTY(EditAnywhere, Meta = (Bitmask))
int32 BasicBits;
```

-Floating Point Types
Unreal utilizza i tipi a virgola mobile standard C++, float e double.

-Boolean Types

I tipi booleani possono essere rappresentati con la parola chiave bool del C++ o come campo di bit.

uint32 bIsHungry : 1; bitfield
bool bIsThirsty;

-Strings

A)FString = è un classico tipo di stringa "array dinamico di caratteri".
B)FName = è un riferimento a una stringa immutabile senza distinzione tra maiuscole e minuscole in una tabella di stringhe globale.
Gli FName forniscono un sistema molto leggero per l'utilizzo delle stringhe, in cui una determinata stringa viene archiviata solo una volta in una tabella dati, anche se viene riutilizzata.
I nomi F non fanno distinzione tra maiuscole e minuscole.
C)FText = è una rappresentazione di stringa più robusta progettata per gestire la localizzazione.
In Unreal Engine (UE) il componente principale per la localizzazione del testo è la classe FText.
Tutto il testo rivolto all'utente dovrebbe utilizzare questa classe, poiché supporta la localizzazione del testo fornendo le seguenti funzionalità:
--FText::AsNumber, Converte qualsiasi tipo numerico supportato dall'UE in una rappresentazione testuale intuitiva ("1234,5" diventa "1.234,5").
--FText::AsPercent, Converte un valore float o double in una rappresentazione testuale percentuale ("0,2" diventa 20%).
--FText::AsMemory, Converte un valore (in byte) in una rappresentazione della memoria di facile utilizzo ("1234" diventa "1,2 KiB").
--FText::AsCurrencyBase, Converte un valore nella rappresentazione di base di una valuta in una rappresentazione valutaria di facile utilizzo ("123450" per "USD" diventa "$1.234,50").
--FText::AsDate, Converte un valore FDateTime in una rappresentazione della data intuitiva
--FText::AsTime, Converte un valore FDateTime in una rappresentazione temporale intuitiva.
--FText::AsDateTime, Converte un valore FDateTime in una rappresentazione intuitiva di data e ora.
--FText::AsTimespan, Converte un valore FTimespan in una rappresentazione delta-time intuitiva (in ore, minuti e secondi).
--FText::ToLower
--FText::ToUpper
Puoi creare un FText vuoto utilizzando FText::GetEmpty() o utilizzando solo FText().
D) Printf, La funzione FString, Printf, può creare oggetti FString con gli specificatori degli argomenti di formato come la funzione printf C++. Allo stesso modo, la macro UE_LOG stampa una stringa formattata printf sullo schermo, sull'output di registro e sui file di registro, a seconda del tipo di build UE4 in esecuzione.

In generale, dovresti utilizzare la macro TEXT() quando imposti i valori letterali delle variabili stringa.

-Creating FStrings

FString TestHUDString = FString(TEXT("This is my test FString."));

Conversioni

From FString

From	To	Example
FString	FName	`TestHUDName = FName(*TestHUDString);` **WARNING** FString -> FName is dangerous as the conversion is lossy as FName's are case insensitive.
FString	FText	`TestHUDText = FText::FromString(TestHUDString);` **WARNING** FString -> FText is valid in some cases, but be aware that the FString's content will not benefit from the FText's "auto localization".

To FString

From	To	Example
FName	FString	`TestHUDString = TestHUDName.ToString();`
FText	FString	`TestHUDString = TestHUDText.ToString();` **WARNING** FText -> FString is dangerous as it is a potentially lossy conversion for some languages.

TO FString

Variable Type	Conversion from String	String Format
float	`FString::SanitizeFloat(FloatVariable);`	
int	`FString::FromInt(IntVariable);`	
bool	`InBool ? TEXT("true") : TEXT("false");`	either 'true' or 'false'
FVector	`VectorVariable.ToString();`	'X= Y= Z='
FVector2D	`Vector2DVariable.ToString();`	'X= Y='
FRotator	`RotatorVariable.ToString();`	'P= Y= R='
FLinearColor	`LinearColorVariable.ToString();`	'(R=,G=,B=,A=)'
UObject	`(InObj != NULL) ? InObj->GetName() : FString(TEXT("None"));`	UObject's FName

From FString

Conversions also exist from FString to int and float numeric variables, as well as to boolean variables.

Variable Type	Conversion from String	Notes
bool	`TestHUDString.ToBool();`	
int	`FCString::Atoi(*TestHUDString);`	
float	`FCString::Atof(*TestHUDString);`	

-Searching

Durante la ricerca all'interno di FStrings, sono disponibili due tipi di ricerca.
a)Il primo, FString::Contains(), restituisce true se viene trovata la sottostringa e false altrimenti.

TestHUDString.Contains(TEXT("Test"), ESearchCase::CaseSensitive, ESearchDir::FromEnd);

b)Il secondo, FString::Find(), restituisce l'indice della prima istanza trovata della sottostringa.

TestHUDString.Find(TEXT("test"), ESearchCase::CaseSensitive, ESearchDir::FromEnd, 10);

-Concatenation

Esistono due operatori per concatenare le stringhe:

a) += Aggiunge la stringa fornita all'oggetto FString.
b) + Crea un nuovo oggetto FString e aggiunge la stringa fornita.

- Printf
Le FStrings costruite con FString::Printf possono essere archiviate in FStrings, nonché visualizzate sullo schermo con i messaggi di debug UE_LOG.

```
FString AShooterHUD::GetTimeString(float TimeSeconds)
{
// only minutes and seconds are relevant
const int32 TotalSeconds = FMath::Max(0, FMath::TruncToInt(TimeSeconds) % 3600);
const int32 NumMinutes = TotalSeconds / 60;
const int32 NumSeconds = TotalSeconds % 60;

const FString TimeDesc = FString::Printf(TEXT("%02d:%02d"), NumMinutes, NumSeconds);
return TimeDesc;
}
```

Quando si utilizzano parametri %s per includere FStrings, è necessario utilizzare l'operatore * per restituire il TCHAR* richiesto per il parametro %s.

-Manipulating FStrings

A) Debug Messaging

Le FStrings possono essere stampate sul Viewport e sul Log di output per scopi di debug.

Per stampare messaggi di debug sul Viewport, utilizzare :
UEngine::AddOnScreenDebugMessage().

Questa funzione accetta i seguenti quattro parametri (in ordine):

a) Key, int
Una chiave univoca per impedire che lo stesso messaggio venga aggiunto più volte. Utilizza -1 come chiave per rendere transitorio il messaggio di debug.
b) TimeToDisplay, float, Per quanto tempo visualizzare il messaggio, in second
c) DisplayColor, FColor, Il colore in cui visualizzare il testo.
d) DebugMessage, FString, Il messaggio da visualizzare (FString).

GEngine->AddOnScreenDebugMessage(-1, 5.f, FColor::Blue, TestHUDString);

B)Print to Output Log

UE_LOG utilizza il markup printf per la parametrizzazione.

UE_LOG(LogClass, Log, TEXT("This is a testing statement. %s"), *TestHUDString);.

--LogClass è la categoria di registro. È possibile utilizzare una categoria esistente (impostata con la macro DECLARE_LOG_CATEGORY_EXTERN in OutputDevices.h) o definirne una propria utilizzando DEFINE_LOG_CATEGORY_STATIC.
--Log è il livello di dettaglio da utilizzare. La verbosità è definita nell'enumerazione ELogVerbosity. I valori validi sono Irreversibile, Errore, Avviso, Visualizzazione, Log, Verbose o VeryVerbose.
--L'argomento successivo è il testo che desideri visualizzare, incluso il markup per i parametri.

Questo esempio utilizza un parametro %s, quindi l'operatore * viene utilizzato per restituire il TCHAR* richiesto per un parametro %s.

//

Unreal Engine 5 Game Development with C++ Scripting

Zhenyu George Li

.Part 1 – Getting Started with Unreal C++ Scripting

-Qual è la differenza tra C++ e Blueprint?

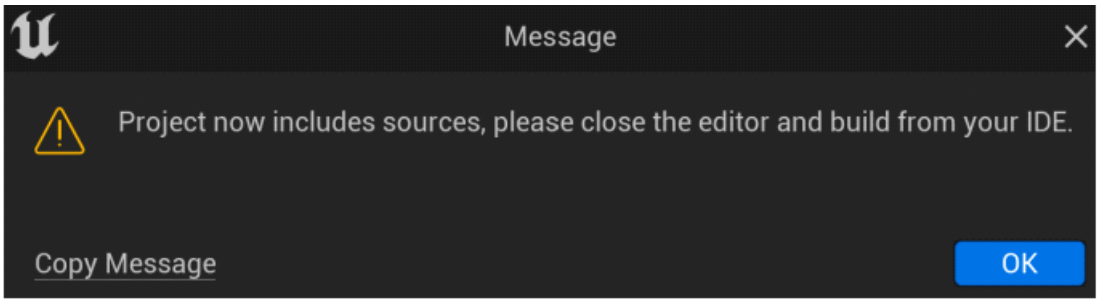
La differenza principale tra loro è che C++ è un linguaggio di programmazione che consente di scrivere codice generico basato su testo, mentre Blueprint è un sistema di scripting visivo per UE.
Se riesci a scrivere script con C++, avrai più possibilità di lavorare in un team di professionisti.
C++ consente di scrivere codice operativo di basso livello. Fornisce inoltre il controllo sul sistema principale che non è accessibile a Blueprint. Inoltre, il codice C++ finale verrà ottimizzato e compilato per essere codice nativo binario compatibile con le macchine.
C++ presenta anche alcune carenze. Un esempio è che può causare errori critici che possono mandare in crash un intero sistema. Questo di solito è causato da errori dello sviluppatore. Poiché Blueprint è un livello protetto, è più sicuro e, quindi, le possibilità che il sistema si arresti in modo anomalo sono minori.

--Quando usi il C++?
"Dove ha senso usare C++ e dove ha senso usare Blueprints?"
Ti consigliamo di basare la tua risposta sui seguenti aspetti e compromessi:
- Prestazioni
- Complessità logica e degli algoritmi
- Accessibilità alle funzioni principali di un sistema
- Esperienza dello sviluppatore

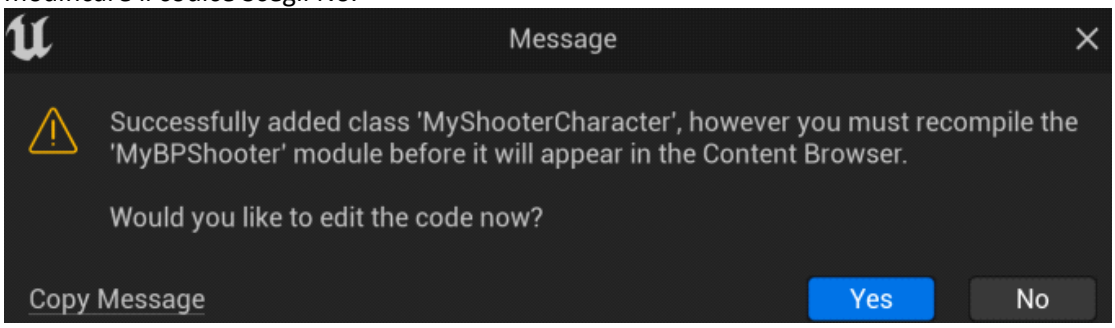
-Qual è la differenza tra la programmazione C++ e lo scripting C++?
1. Programmazione C++ significa utilizzare il linguaggio di programmazione C++ per scrivere codice per qualsiasi scopo; non deve essere solo per i progetti dell'UE.
2. Lo scripting C++, in questo libro, è un dialetto specifico del linguaggio di programmazione C++ supportato dall'UE.
Sfrutta la potenza della sintassi C++ e funziona anche con le API (Application Programming Interface) di UE, che consentono agli sviluppatori di creare ed estendere le funzionalità del motore
/Un'API, acronimo di Application Programming Interface (interfaccia di programmazione delle applicazioni), è un insieme di regole o protocolli che permettono alle applicazioni software di comunicare tra loro per scambiare dati//

-Creazione di un progetto C++ Shooter da un modello
Per selezionare il tipo di VS che si utilizza si andrà su t Edit | Editor Preferences, e General | Source Code.

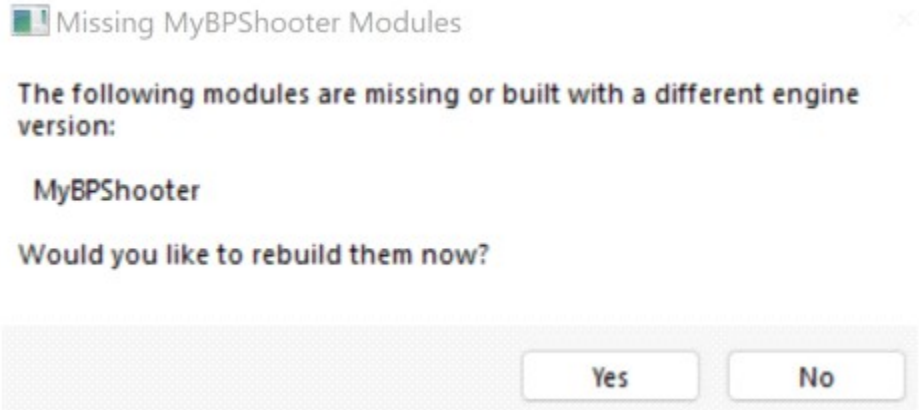
-Conversione di un progetto Blueprint esistente in un progetto C++
Seleziona Tools | New C++ Class, aggiungi classe, scegli Character
Si prega di prestare attenzione al percorso in cui verranno posizionati l'intestazione e i file di origine.
Attendere il messaggio popup che indica che il processo della classe C++ è stato aggiunto



Clicca OK.
A questo punto, verrà visualizzata la finestra di dialogo del messaggio, che chiede se si desidera modificare il codice Scegli No.

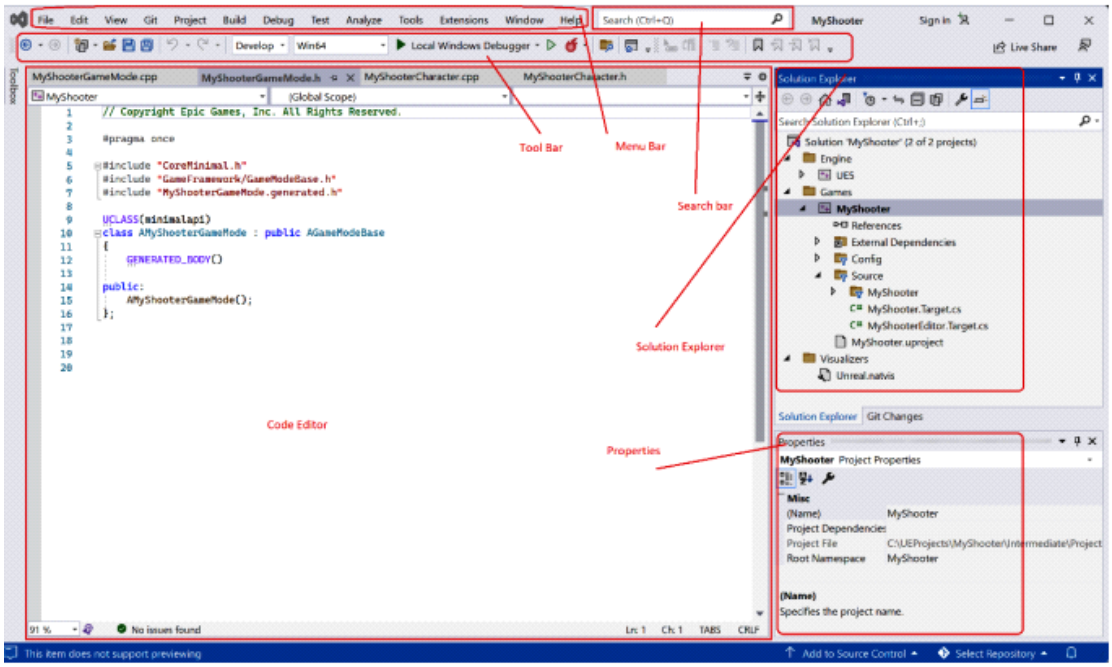


Chiudi l'editor UE e riapri MyBPShooter. Quando viene visualizzata una finestra di dialogo in cui viene chiesto se si desidera ricompilare il progetto, rispondere a Sì.

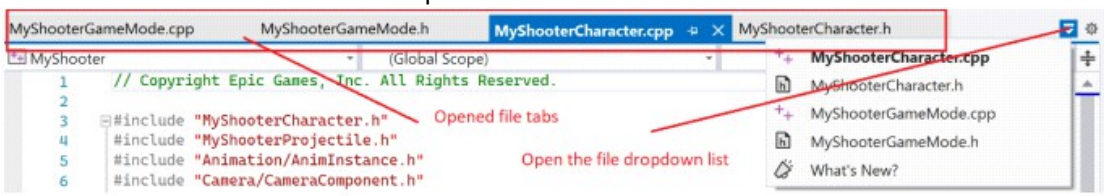


.Editing C++ Code in Visual Studio
VS dispone dei compilatori di linguaggi di programmazione più diffusi integrati con il pacchetto di installazione in modo che il codice sorgente C++ possa essere compilato direttamente come codice macchinario eseguibile. Inoltre, VS supporta in particolare Unreal Engine e funziona bene con l'ambiente di sviluppo del motore.

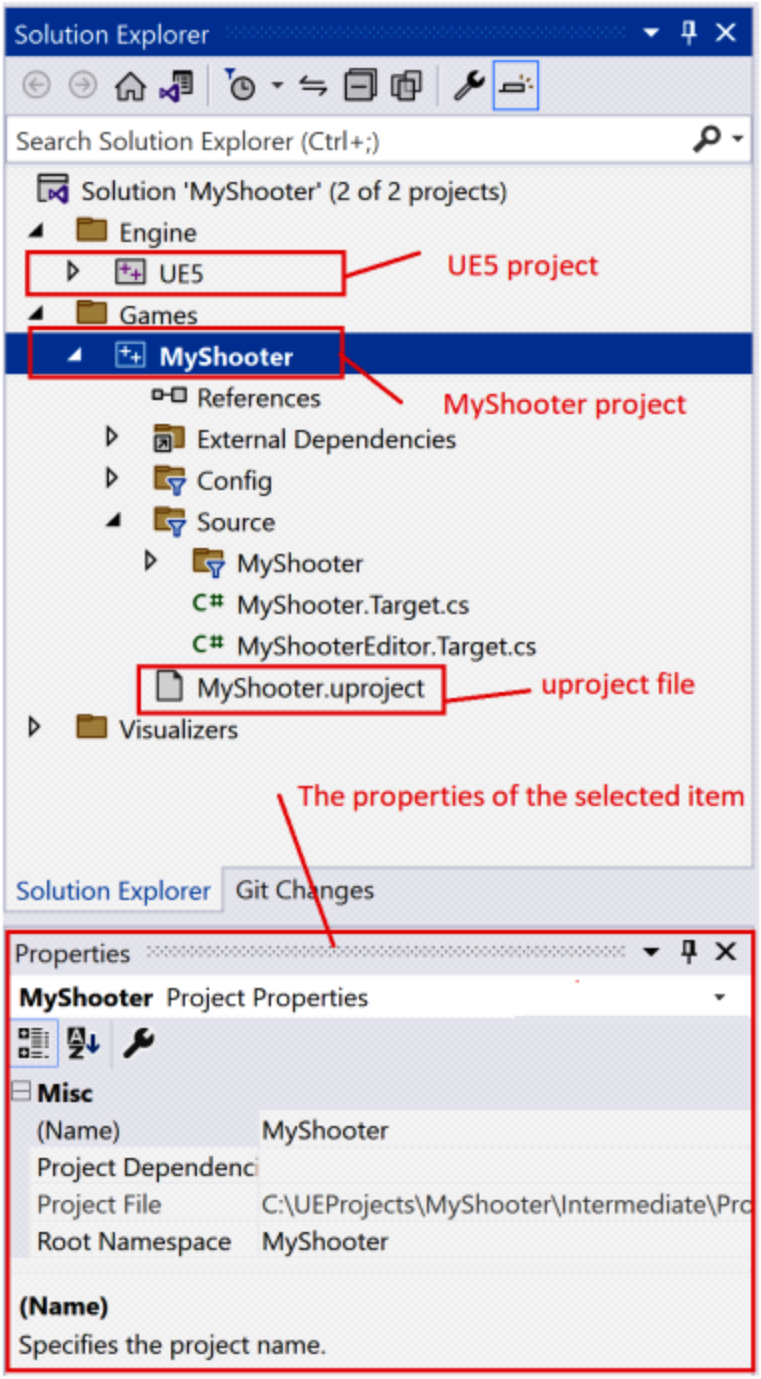
Panoramica di VS



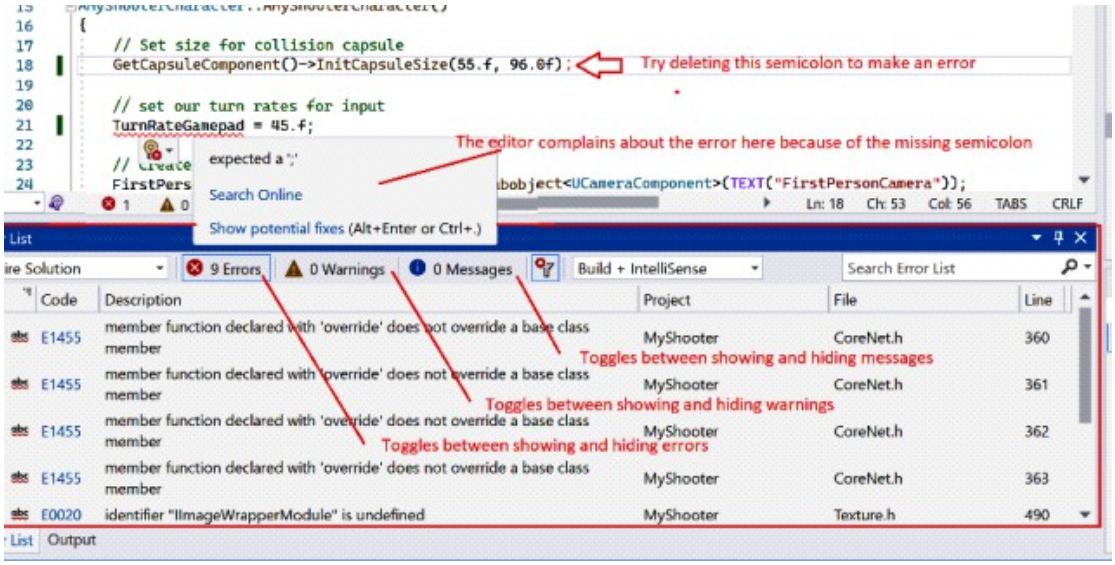
Code editor, L'editor di codice sul lato sinistro è la posizione in cui si scrive il codice C++.
È possibile scegliere di visualizzare qualsiasi file di origine facendo clic sulla rispettiva scheda. Nel caso in cui vengano aperti troppi file e non tutte le schede dei file possano essere visualizzate sulla barra delle schede, è possibile fare clic sul pulsante a discesa nell'angolo in alto a destra per visualizzare un elenco di tutti i file aperti.



Menus, I menu forniscono interfacce per le operazioni, le opzioni e le funzionalità che è possibile utilizzare per il lavoro di codifica.
Search box, Questa funzionalità offre un modo semplice e veloce per eseguire ricerche nelle funzionalità dell'IDE e nel codice. Ti mostrerà un elenco di opzioni rilevanti per il testo inserito.
Toolbar, La barra degli strumenti è una striscia orizzontale sotto la barra dei menu che contiene alcuni pulsanti di scelta rapida associati a comandi, ad esempio Apri file, Salva, Salva tutto, Start, Avvia debug e così via.
Solution Explorer, In Esplora soluzioni vengono visualizzati i progetti, le cartelle e i file di una soluzione in una rappresentazione gerarchica della struttura ad albero. È possibile sfogliare l'albero per selezionare e aprire i file nell'editor. In Esplora soluzioni vengono visualizzati i progetti, le cartelle e i file di una soluzione in una rappresentazione gerarchica della struttura ad albero. È possibile sfogliare l'albero per selezionare e aprire i file nell'editor.
Una soluzione VS può contenere più progetti e ogni progetto ha il proprio progetto, il codice sorgente, la configurazione, le dipendenze e altri file. In questo caso, la soluzione MyShooter contiene due progetti, UE5 e MyShooter:



Output window, vengono visualizzati i messaggi di output del processo di creazione del codice.
Compilare il progetto per visualizzare gli output di compilazione. F7
o RICOMPILA SOLUZIONE CTRL + ALT + F7
Error List window, mostra gli errori di compilazione, gli avvisi e i messaggi relativi allo stato corrente del codice.



I pulsanti di attivazione/disattivazione Errori, Avvisi e Messaggi consentono di abilitare o disabilitare i filtri in modo da poter scegliere di visualizzare le informazioni a cui si è interessati.

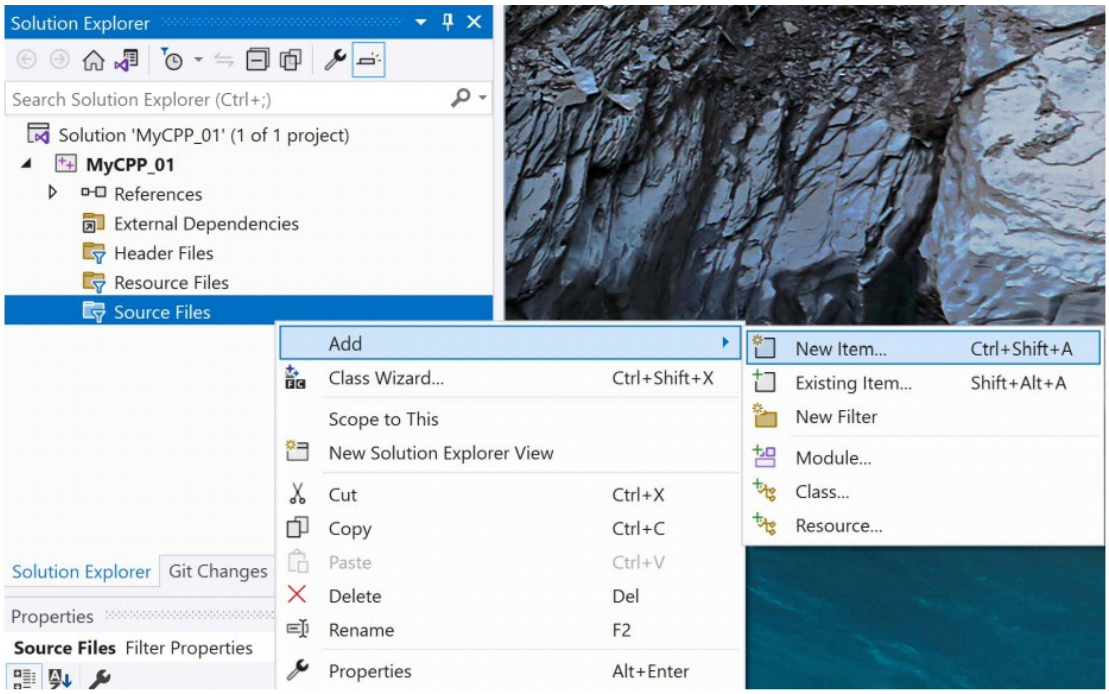
-Editing code in VS

Controlling the caret (input cursor)
Il cursore di inserimento è spesso rappresentato come una linea verticale lampeggiante durante la modifica in Visual Studio
The text editing keys, I tasti di modifica del testo vengono utilizzati per alternare tra le modalità Inserisci e Sovrascrivi, nonché per facilitare l'eliminazione del testo.
Code selection, La selezione del codice consente agli sviluppatori di evidenziare porzioni specifiche di testo per azioni quali copiare, tagliare, eliminare e così via.
Shift + Left arrow key, Shift + Right arrow key: Seleziona o deseleziona caratteri a sinistra o destra.
Shift + Up arrow key, Shift + Down arrow key: Selezionare rispettivamente verso l'alto o verso il basso un'altra riga quando si estende un'area di selezione.
Ctrl + Shift + Left arrow key, Ctrl + Shift + Right arrow key: Selezionare una sezione più significativa, ad esempio una parola, una variabile, un simbolo e così via, rispettivamente sul lato sinistro o destro quando si estende un'area di selezione.
IntelliSense, è una risorsa preziosa quando si scrive codice. È come un agente intelligente che lavora in background e che può mostrare informazioni rilevanti sul nome di un oggetto, sui membri disponibili di un tipo o sui dettagli dei parametri per i diversi overload di un metodo.

Shortcut utili
Ctrl + Shift + F, Globally searches
Ctrl + H, consente di cercare e sostituire una parola chiave con testo diverso all'interno del documento corrente.
CTRL + MAIUSC + H, consente di cercare e sostituire una parola chiave con testo diverso per tutti i file di origine nella soluzione.
Ctrl + K + C, commenta il blocco di codice selezionato
Ctrl + K + U, rimuove il commento dal blocco di codice selezionato
Ctrl + K + F, Dispone automaticamente il blocco di codice selezionato come impostazioni di

formattazione
Ctrl + K + D, Automatically arranges the document code to be the formatting settings.
F12, Ctrl + F12, passa alla definizione o alla dichiarazione della parola chiave selezionata, ad esempio una variabile, un nome di classe, un nome di funzione e così via.
Ctrl + G, consente di passare a una riga in base a un numero di riga.
Ctrl + T, va a tutti. Si tratta di uno strumento utile quando si conosce il nome di un file e si desidera aprire il file, soprattutto quando si lavora su un progetto su larga scala con centinaia o migliaia di file sorgente.
Ctrl + Tab, consente di spostarsi tra le due finestre del documento più recenti.
F9, attiva o disattiva un punto di interruzione su una riga di codice. Quando si esegue in modalità di debug, l'esecuzione del programma verrà sospesa in corrispondenza del punto di interruzione.
F5, Avvia l'esecuzione del programma in modalità Debug
Ctrl + F5, avvia l'esecuzione del programma senza la modalità di debug. I punti di interruzione verranno ignorati.
F10, Quando viene premuto, l'IDE ti mostrerà cosa sta facendo un passo alla volta in modo che tu possa capire dove si trova il problema.
F11, se premuto, l'IDE imposterà lo stato attivo dell'analisi sulla prima riga della funzione in modo da poter approfondire tale funzione per ulteriori dettagli.

-Practicing C++ coding
Crea Nuovo elemento da ***FILE DI ORIGINE***



Scrivi questo codice:

```
#include <iostream>

int main() {
```

```
	std::cout << "Hello, World!" << std::endl;
	return 0;
}
```

Costruisci la soluzione ed esegui il programma facendo clic su Start o su Avvia senza debug pulsante

Abbiamo appena aggiunto la funzione di inserimento main() al modulo main.cpp. Aggiungiamo ora altri due file di origine (Calculator.cpp e Calculator.h) al progetto, che conterranno la definizione e il codice di implementazione della classe Calculator.

-Adding the Calculator class

La classe Calculator è progettata per creare un oggetto calcolatrice in grado di eseguire operazioni di addizione e sottrazione. Per includere la classe Calculator nel progetto, aggiungeremo i file di intestazione e di origine al progetto, come indicato di seguito.

File d'intestazione Calculator

Calculator.h :

```
#pragma once

#include <iostream>

class Calculator
{
public:
	float Add(float a, float b);
	float Subtract(float a, float b);

private:
	void OutputResult(float, std::string, float, float);

};
```

File di Origine

Calculator.cpp:

```
#include "Calculator.h"

float Calculator::Add(float a, float b)
{
	float result = a + b;
	OutputResult(a, " + ", b, result);
```

```
	return result;
}

float Calculator::Subtract(float a, float b)
{
	float result = a - b;
	OutputResult(a, " - ", b, result);
	return result;
}

void Calculator::OutputResult(float a, std::string operation, float b, float
result)
{
	std::cout << a << operation << b << " = " << result << std::endl;
}
```

Main.cpp

```
#include <iostream>
#include "Calculator.h"
int main()
{
 std::cout << "MyCPP_01: Hello world! \n";
 Calculator Calculator;
 Calculator.Add(1.0f, 2.0f);
 Calculator.Subtract(10.0f, 5.0f);
 return 0;
}
```

.Learning C++ and ObjectOriented Programming

-Che cos'è C++?
C++ è un linguaggio di programmazione che supporta OOP, può essere utilizzato per creare applicazioni ad alte prestazioni e offre agli sviluppatori un controllo di basso livello sulle risorse di sistema e sulla memoria.
C++ è progettato per essere un linguaggio compilato, ovvero tradotto in codice in linguaggio macchina che può essere compreso ed eseguito direttamente da un sistema.

Durante la scrittura di script C++ in Unreal Engine, ogni volta che si apportano modifiche, è necessario compilare prima il codice e quindi avviare il gioco.

-Esplorazione della struttura del programma C++

I programmi sorgente C++ seguono in genere la stessa struttura del programma:

- #include all'inizio del programma, che consentono a questo programma di accedere alla libreria di sistema C++ e ad altre funzionalità del programma sorgente C++. #include

istruzioni sono istruzioni speciali che non terminano con un punto e virgola.

- La funzione Main(), che viene eseguita quando il programma viene eseguito.
- Return 0, che si trova alla fine della funzione main(), che notifica il risultato del completamento dell'esecuzione del programma.

-Definizione di funzioni C++

Una funzione è un blocco di codice che viene eseguito solo quando viene chiamato.
Le funzioni sono solitamente definite e utilizzate per eseguire determinate azioni e possono essere chiamate in qualsiasi momento quando necessario; pertanto, sono codice riutilizzabile.

-Definizione di funzioni con o senza parametri
Per definire una funzione, è necessario specificare il nome e il tipo restituito della funzione, seguiti da una coppia di parentesi.

Senza parametro di tipo void che non restituisce nulla:

```
void DisplayLabel()
{
 std::cout << "The result of 1 + 2 is ";
}
```

È possibile passare i dati come parametri in una funzione e i parametri della funzione vengono inseriti tra le parentesi.
Di seguito viene presentata una funzione che ha due parametri int e restituisce l'aggiunta dei due valori di input come risultato.

```
int Add(int a, int b)
{
 return a + b;
}
```

-Calling functions
Per chiamare una funzione, è possibile scrivere il nome della funzione seguito da due parentesi.
Se la funzione richiede un certo numero di parametri, è necessario scrivere i parametri in modo che i valori vengano passati alla funzione.

```
Std:count << DisplayLabel() << Add(1, 2) << std:endl;
```

Il frammento di codice seguente illustra come chiamare le due funzioni definite in precedenza

-Scrivere la funzione main()
Ogni programma C++ deve avere una funzione main. Si tratta di una funzione speciale che indica il punto di avvio dell'esecuzione del programma.
Il tipo restituito della funzione main() può essere int o void,
<u>int main()</u>
<u>void main()</u>
La differenza è che int main() dice al compilatore che il programma restituirà un valore intero al

sistema operativo, mentre void main() dice al compilatore che il programma non restituirà alcun valore.

-Utilizzo di un programma di calcolo di base
Il programma MYCPP_02 dovrebbe avere una funzione main() e una funzione Add().

a)void Main(): Chiama la funzione Add() per calcolare 1 + 2 e 3 + 4 e restituisce i risultati
b) int Add(int a, int b): somma i due valori interi dei parametri, a e b, e restituisce il risultato del calcolo

```
#include <iostream>

int Add(int a, int b)
{
	return a + b;
}

int main()
{
	std::cout << "My calculations" << std::endl;

	int result = Add(3, 4);
	std::cout << "La somma tra 3 e 4 è "
		<< result
		<< std::endl;
	result = Add(5, 6);
	std::cout << "La somma tra 5 e 6 è "
		<< result
		<< std::endl;


	std::cout << "Finished" << std::endl;
	return 0;
}
```

--L'oggetto std:cout rappresenta il dispositivo di output standard, che di solito è la console di output.
--L'operatore << viene applicato a un flusso di output. In questo esempio, trasferisce il valore sul lato destro al dispositivo di output sul lato sinistro.
--std:endl rappresenta un carattere di nuova riga nella sequenza di output. Un altro modo per inserire un carattere di nuova riga in una stringa consiste nell'usare n (in C++, la sequenza di escape n rappresenta un carattere di nuova riga).
<< "Integer addition: 3+4=" << result << "\n";
--L'istruzione int result = Add(1, 2) definisce una variabile di tipo integer per memorizzare il risultato restituito calcolato dalla funzione Add().

-Definizione delle variabili
Una variabile è un contenitore in cui sono archiviati i valori dei dati.
Il modo per dichiarare (creare) una variabile consiste nello specificare un tipo, il nome della

variabile, e assegnargli un valore.

type variableName = value;

-L'inserimento di una parola chiave const prima di una dichiarazione di variabile rende la variabile una costante, ovvero non modificabile e di sola lettura.

-Using C++ arrays
Una matrice è una serie di elementi dello stesso tipo inseriti in un blocco di memoria contiguo a cui può fare riferimento singolarmente l'indice dell'elemento.

Per dichiarare una matrice, definire il tipo di variabile, seguito dal nome della matrice e quindi da una coppia di parentesi quadre. È possibile specificare un numero che specifica la lunghezza della matrice.

string playerNames[4];

È possibile inizializzare gli elementi della matrice elencando i valori stringa.

string playerNames[] = { "George", "Sarah", "Willy", "Mike" };

-Utilizzo degli operatori C++

Operatori aritmetici: eseguono operazioni matematiche

Operator	Operation	Description	Example(s)
+	Addition	Adds together two values	1 + 2, 2.1 + 3.5, and v1 + v2
-	Subtraction	Subtracts a value from another	5 - 1, 1.2 - 3.3, and v1 - v2
*	Multiplication	Multiplies two values	3 * 2, 3.14f * 2, and a * b
/	Division	Divides a value by another	4 / 2, 3.14f / 2, and a / b
%	Modulus	Returns the remainder of the division	11 % 3 (returns 2)
++	Increment	Increases the current value by 1	3++ or ++3 (returns 4)
--	Decrement	Decreases the current value by 1	3-- or --3 (returns 2)

-Operatori di assegnazione: assegnano e memorizzano i valori alle variabili:

Operator	Operation	Description	Example(s)
=	Assign	Assigns a value to the variable	`const float PI = 3.14f;`
+=	Add and assign	Adds a value and assigns the result back	`pi += 3.14f;` (the value of `pi` will be `6.28f`)
-=	Subtract and assign	Subtracts a value and assigns the result back	`pi = PI;` `pi -= 3.14f;` (the value of `pi` will be `0.0f`)
*=	Multiply and assign	Multiplies a value and assigns the result back	`pi = PI;` `pi *= 2.0f;` (the value of `pi` will be `6.28f`)
/=	Divide and assign	Divides a value and assigns the result back	`pi /= 2.0f;` (the value of `pi` will be `1.57f`)
%=	Mod and assign	Mods a value and assigns the result back	`Int n = 13;` `n %= 5;` (the value of `n` will be `3`)
&=	Bitwise And and assign	Applies a bitwise And operation first, and then assigns the result back	`char byte = 0b10010001;` `byte &= 0b11100010;` (the value of `byte` will be `0b01000000`; `0b` is a prefix that indicates a binary value expression)
|=	Bitwise Or and assign	Applies a bitwise Or operation first, and then assigns the result back	`char byte = 0b10010001;` `byte |= 0b11100010;` (the value of `byte` will be `0b11100011`)
^=	Bitwise Exclusive Or and assign	Applies a bitwise exclusive Or operation first, and then assigns the result back	`char byte = 0b10010001;` `byte &= 0b11100010;` (the value of `byte` will be `0b01110011`)
<<=	Left-shift and assign	Shifts the value a certain number of bits to the left, and then assigns the result back	`char byte = 0b10010001;` `byte <<= 2;` (the value of `byte` will be `0b01000100`)
>>=	Right-shift and assign	Shifts the value a certain number of bits to the right, and then assigns the result back	`char byte = 0b10010001;` `byte >>= 2;` (the value of `byte` will be `0b00100100`)

-Operatori logici: determinano il risultato logico finale in base alla combinazione di due istruzioni logiche

Operator	Operation	Description	Example(s) int x = 0, y = 5, z = 0;
&&	AND	Returns `true` if both statements are true	`x > y && y > z` (false) `x < y && y > z` (true)
||	OR	Returns `true` if one statement is true	`x > y && y > z` (true) `x < y && y > z` (true)
!	NOT	Returns `true` if the statement is false, and returns `false` if the statement is true	`!(x > y)` (true) `!(x < y)` (false) `!(x == z)` (false) `!(x > y && y > z)` (true) `!(x < y && y > z)` (false)

-Operatori bit per bit: eseguono operazioni bit per bit. I valori binari vengono solitamente utilizzati come flag o maschere.

Operator	Operation	Description	Example(s) (char b1 = 0b00100101; char b2 = 0b10100011;)
&	Bitwise AND	Compares two values bit by bit. Returns `1` only when both bits are 1; otherwise, it returns `0`.	`b1 & b2` (0b00100001)
|	Bitwise inclusive OR	Compares two values bit by bit. Returns `0` only when both bits are 0; otherwise, it returns `1`.	`b1 | b2` (0b10100111)
^	Bitwise exclusive XOR	Compares two values bit by bit. Returns `1` when the two bits are different; otherwise, it returns `0`.	`b1 ^ b2` (0b10000110)
~	Bitwise NOT	Applies unary complement (bit inversion).	`~b1` (0b11011010) `~b2` (0b01011100)
<<	Shift left	Shifts bits left.	`b1 << 1` (0b01001010) `b1 << 3` (0b00101000) `b2 << 2` (0b10001100)
>>	Shift right	Shifts bits right.	`b1 >> 1` (0b00010010) `b1 >> 3` (0b00000100) `b2 >> 2` (0b00101000)

-Acettazione dell'input dell'utente
CIN legge i dati da una tastiera con l'operatore di estrazione (>>).

```
int x;
cout << "Please input your age: ";
cin >> x;
cout <<"Your age is: "<< x;
```

-Aggiunta di commenti C++
I commenti vengono utilizzati per fornire descrizioni e spiegazioni aggiuntive a una riga o a un blocco di codice.

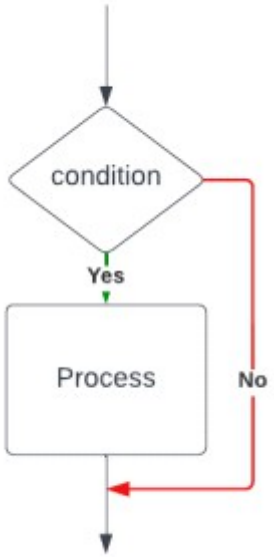
-Controllo del flusso C++
Essendo un linguaggio di programmazione completo e potente, C++ dovrebbe supportare non solo le istruzioni sequenziali, ma anche i rami condizionali e le azioni ripetibili.
Il controllo di flusso si riferisce all'ordine delle chiamate di funzione, delle istruzioni e delle istruzioni che vengono eseguite o valutate durante l'esecuzione di un programma.
Determina quale blocco di codice viene eseguito in una determinata circostanza (condizione).

In C++, le istruzioni if, switch, while loop e for loop vengono utilizzate principalmente per controllare i flussi del programma

-Utilizzo dell'istruzione if
Usiamo l'istruzione if per eseguire un blocco di codice C++ se una condizione è vera.
Un'istruzione if inizia dalla parola chiave if seguita da una condizione, racchiusa tra una coppia di parentesi.
L'istruzione if ha tre forme: if, else e else if.

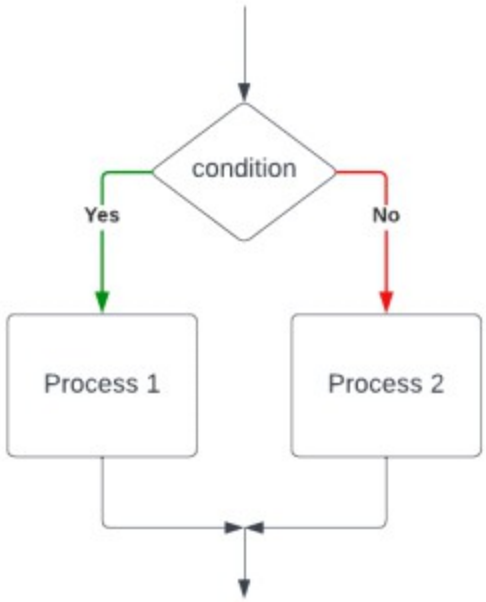
if :

Un'istruzione if controlla prima l'espressione della condizione e il processo viene eseguito solo quando il risultato di una condizione è true:




```
if(condition)
{
 //Process code block
}
```

else :
Un'istruzione else controlla la condizione per determinare se il processo per true (Processo 1) o il processo per false (Processo 2) deve essere eseguito:

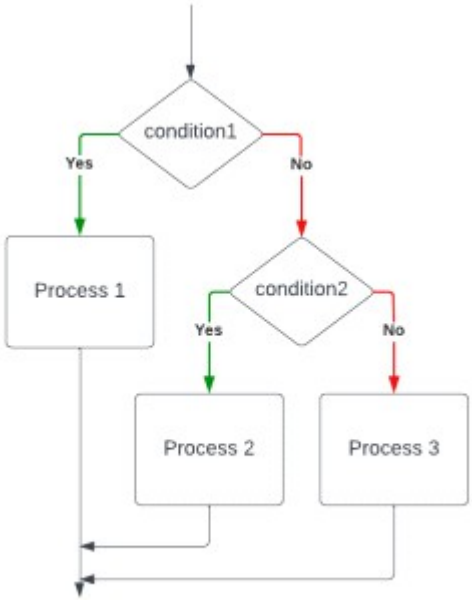



```
if(condition)
{
 //Process 1 code block
```

```
}
else
{
 //Process 2 code block
}
```

else if

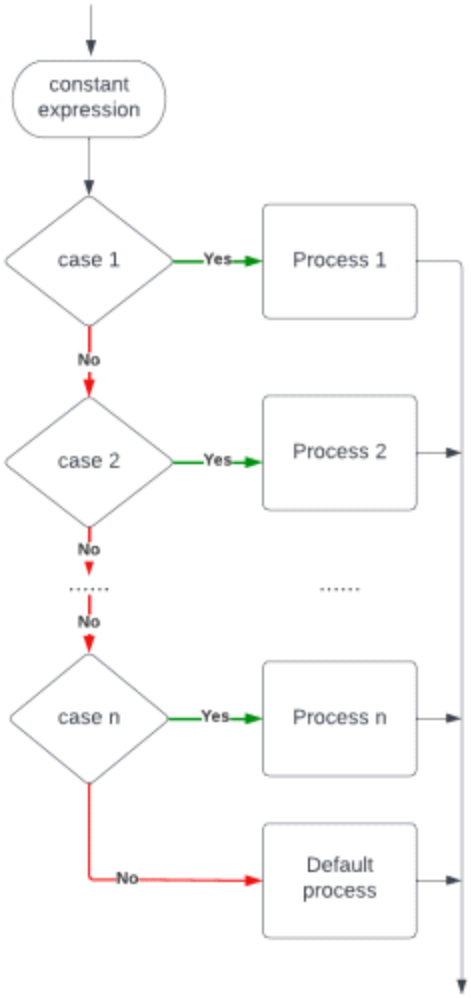
L'istruzione else if consente di valutare in sequenza più rami condizionali; ogni altro se il ramo è controllato solo se le condizioni precedenti sono false.




```
if(condition1)
{
 //Process 1 code block
}
else if(condition2)
{
 //Process 2 code block
}
else
{
 //Process 3 code block
}
```

-Utilizzo dell'istruzione switch

L'istruzione switch viene utilizzata per selezionare uno dei numerosi blocchi di codice da eseguire.




```
switch(expression)
{
case constant1:
 //Process 1 code block
 break;
case constant2:
 //Process 2 code block
 break;
case constantn:
 //Process n code block
 break;
default:
 //Default process code block
}
```

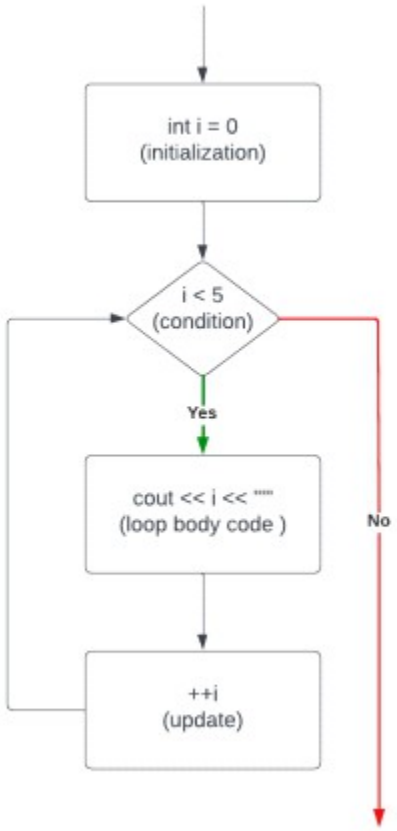
-Utilizzo delle istruzioni di ciclo

Le istruzioni di ciclo vengono utilizzate per ripetere l'esecuzione di un blocco di codice se viene raggiunta una condizione specificata.

Esistono tre tipi di cicli:
• Il ciclo for
• Il ciclo while
• Il ciclo do/while

for loop

Quando si conoscono i tempi di esecuzione di un ciclo.



for(inizializzazione; condizione; aggiornamento)
{ //codice del corpo del ciclo }

- -itruzione di inizializzazione: inizializza una o più variabili. Questa istruzione viene eseguita una sola volta all'inizio del ciclo.
--L'istruzione di condizione: quando questa condizione è falsa, il ciclo termina; in caso contrario, continua. La condizione è in genere correlata al valore delle variabili inizializzate.

L'istruzione update: aggiorna il valore delle variabili inizializzate

```
for(int i = 0; i < 5; ++i)
{
 Cout << i << ",";
```

}

while and do/while loops

I cicli while e do/while sono istruzioni di ciclo simili. Entrambi eseguono un ciclo in un blocco di codice purché una condizione specificata sia vera. La differenza è che

-- primo controlla la condizione prima di eseguire il codice del corpo

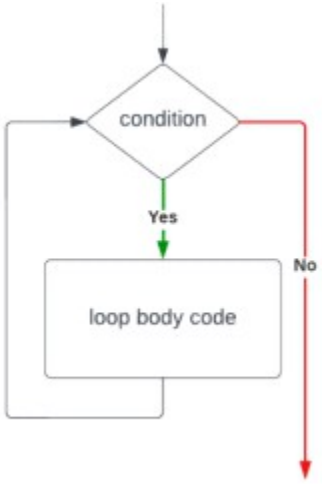
--il secondo esegue il controllo dopo

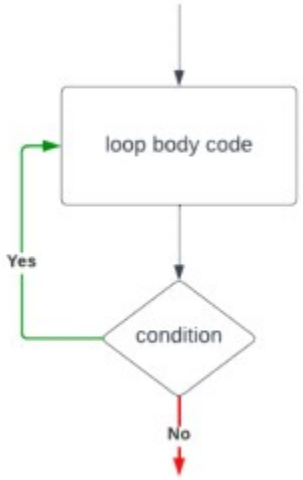
while loop example	do/while loop example

`While(condition)`
`{`
`  //loop body code`
`}`
	`do`
	`{`
	`  //loop body code`
	`} while(condition);`

while loop example	do/while loop example

`bool condition = false;`
`while(conditiion)`
`{`
`  cout << "executed!"`
`}`
	`bool condition = false;`
	`do`
	`{`
	`  cout << "executed!"`
	`} while(condition);`





-Utilizzo delle istruzioni break e continue

L'istruzione break non viene utilizzata solo per uscire da un'istruzione switch, ma può anche essere utilizzata per terminare un ciclo in modo prematuro.

L'istruzione continue interrompe il ciclo corrente e procede all'iterazione successiva, ignorando il codice rimanente all'interno del ciclo.

Esercizio MYCPP_03

Main.cpp

```
#include <iostream>
#include "Calculator.h"

int main()
{

	std::cout << "My calculations !" << "\n";

	float input1, input2{};

	while (true)
	{

		std::cout << "Input the first value (0 to exit): ";
		std::cin >> input1;

		if (input1 == 0)
			break;

		int a = input1;
		int b = input2;
		if (a == input1 && b == input2)
		{
			int result = Add(a, b);
			std::cout << "Integer addition: " << a << " + " << b << "
= " << result << "\n";

		}
		else
		{
			float result = Add(input1, input2);
			std::cout << "Float addition: " << input1 << " + " <<
input2 << " = " << result << "\n";

		}



	}


}
```

Calculator.h

```
#pragma once

int Add(int a, int b);
```

```
float Add(float a, float b);
```

Calculator.cpp

```
#include "Calculator.h"

int Add(int a, int b)
{
	return a + b;
}

float Add(float a, float b)
{
	return a + b;
}
```

-Creazione di riferimenti e puntatori

La copia dei valori delle variabili in più posizioni comporta il rischio di valori delle variabili incoerenti, nonché di prestazioni inferiori e di un maggiore utilizzo della memoria.
L'uso di riferimenti e puntatori in C++ può non solo aiutare a utilizzare meno memoria e migliorare le prestazioni, ma anche fornire la flessibilità necessaria per modificare i valori delle variabili originali.

-References
Una variabile di riferimento è un riferimento a una variabile esistente ed è definita con l'operatore & prima del nome della variabile di riferimento.
Il nome di una variabile di riferimento può essere considerato come sinonimo del nome della variabile originale. Una volta definito, è possibile utilizzare il nome del riferimento o il nome della variabile per indicare la stessa cosa.

```
float Add(float &a, float &b)
{
 return a + b;
}
Void main()
{
 int x = 1, y = 2;
 cout << Add(x, y);
}
```

Quando si chiama la funzione Add, i valori dei parametri non vengono copiati; A e B sono considerati i due alias di X e Y in questo momento. La modifica dei valori delle variabili di riferimento all'interno della funzione Add comporterà modifiche ai valori di x e/o y

```
string myName = "George";     //defines a string variable
string &nameOfMe = myName;      //defines a reference
cout << "My name:" << myName;  //outputs "My name: George"
cout << "My name: " << nameOfme;  //outputs "My name: George"
myName = "Li";
cout << "My name:" << nameOfMe;  //outputs "My name: Li"
```

-Puntatori
Un puntatore è una variabile che memorizza l'indirizzo di memoria di un'altra variabile e viene creata con l'operatore * prima del nome della variabile.

```
float Add(float *a, float *b)
{
 return *a + *b; //add and return the pointed values
}
Void main()
{
 int x = 1, y = 2;
 cout << Add(&x, &y); //pass the address of x and y
}
```

In questo caso, i parametri della funzione sono entrambi puntatori. Quando si chiama la funzione Add, i parametri a e b copiano solo gli indirizzi di memoria dei valori di input; per ottenere i valori puntati all'interno della funzione Add, è necessario aggiungere il prefisso * ai nomi delle variabili puntatore.

```
string myName = "George";   //defines a string variable
string *pMyName = &myName;    //defines a pointer
string *pNameOfMe = pMyName;//defines another pointer
cout << "My name:" << *pMyName;   //outputs "My name: George"
cout << "My name: " << *pNameOfme;//outputs "My name: George"
myName = "Li";
cout << "My name:" << *pMyName;   //outputs "My name: Li"
You just learned cout << "My name:" << *pNameOfMe; //outputs "My name:
Li"
```

-Capire l'OOP
OPP cerca di riflettere i concetti del mondo reale creando oggetti che contengono attributi (campi) e funzioni (metodi)

Ci sono tre pilastri principali su cui si basa OOP:

Incapsulamento: ciò significa che i dati e le funzioni possono essere raggruppati in classi

Ereditarietà: ciò significa che una classe può derivare da un'altra classe base come classe figlio.

Polimorfismo: ciò significa che un metodo di classe può avere più forme.

-Cosa sono le classi e gli oggetti?
Le classi e gli oggetti sono le due caratteristiche principali di OOP.
Una classe è un modello che può essere utilizzato per creare oggetti; D'altra parte, un oggetto è un'istanza di una classe.

In sostanza, la classe definisce le proprietà e i comportamenti comuni, mentre gli oggetti rappresentano singole istanze create in base al progetto di classe.

-Creating classes in C++

Una classe è un tipo di dati definito dall'utente che inizia con la parola chiave class, seguita dal nome della classe. Il corpo della classe è definito all'interno di una coppia di parentesi graffe e termina con un punto e virgola alla fine.

Una classe può avere **attributi**, ovvero variabili che rappresentano le proprietà della classe.
Può anche avere **metodi** (funzioni membro).

```
class Computer
{
 Private:
 bool _ComesWithMonitor = true; //true or false
 Public:
 string CPUType = "Intel"; //"Intel" or "AMD"
 int RAMSize = 4096; //Unit: Giga bytes
 void TurnOn()
 {
 //…
 }
 void Shutdown()
 {
 //…
 }
 void SetComesWithMonitor(bool ComesWithMonitor)
 {
 _ComesWithMonitor = ComesWithMonitor;
 }
 bool GetComesWithMonitor()
 {
 Return _ComesWithMonitor;
 }
 }
```

-Gli attributi della classe
Gli identificatori pubblici e privati definiscono gli ambiti di accesso per attributi e funzioni.
È possibile accedere agli attributi e alle funzioni pubbliche all'esterno della classe.
Gli attributi e le funzioni private sono visibili solo all'interno della classe.

-I metodi della classe
I metodi sono funzioni che appartengono a una classe.

-Creazione di oggetti in C++

Metodo 1 – definizione di una variabile

Di seguito è riportato un esempio di definizione di un'istanza MyComputer dalla classe Computer:

```
Computer MyComputer; //method 1
```

Metodo 2: utilizzo della parola chiave new per creare un'istanza di un oggetto e memorizzarla in un puntatore

Di seguito è riportato un esempio di utilizzo della classe Computer per creare un'istanza di una nuova istanza di MyComputer e archiviare il puntatore della nuova istanza al puntatore pMyComputer:

```
Computer *pMyComputer = new Computer(); //method 2
```

Questo secondo metodo richiede al sistema di allocare un blocco di memoria per la nuova istanza Computer e di restituire il puntatore all'indirizzo di memoria.

-Aggiunta di funzioni costruttore e getter per la classe calculator
Un costruttore di una classe è un metodo speciale che viene chiamato automaticamente quando viene creato un oggetto.
Un costruttore di una classe è un metodo speciale che viene chiamato automaticamente quando viene creato un oggetto. Il nome di un costruttore deve essere uguale al nome della classe e la funzione non deve avere alcun tipo restituito. Gli sviluppatori in genere inseriscono le inizializzazioni degli attributi della classe nei costruttori.

1. Aggiungere un attributo _name privato alla classe Calculator, che memorizzerà il nome della calcolatrice:

```
string _name;
```

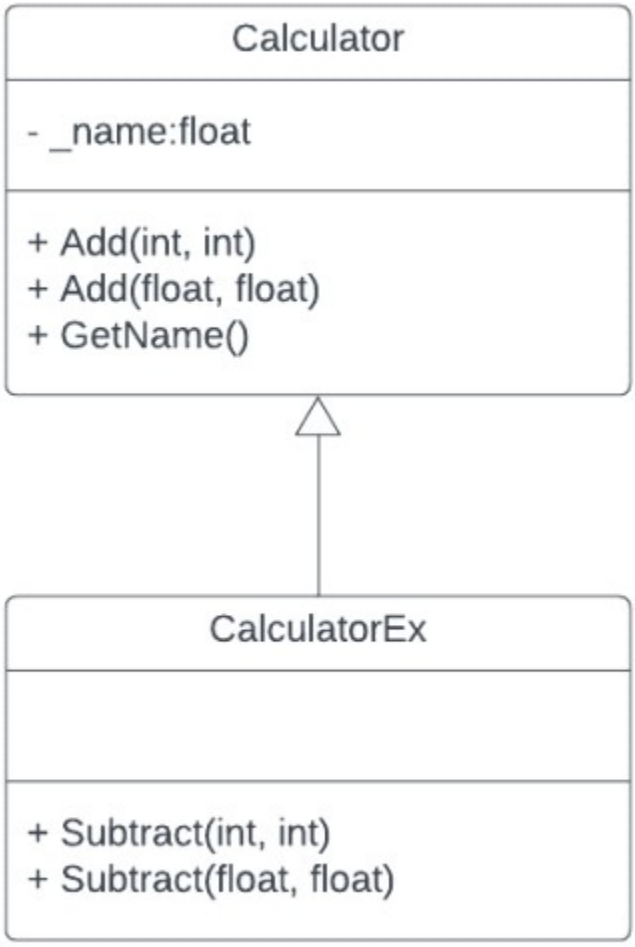
2. Aggiungere il costruttore e le dichiarazioni getter alla classe:

```
Calculator();
String GetName();
```

3. Implementare il costruttore e il getter in Calculator.cpp:

```
Calculator::Calculator()
{
 _name = "Addition Calculator"; //set the name
}
```

```
string Calculator::GetName()
{
 Return _name;
}
```



UML UML è l'acronimo di Unified Modeling Language. Un diagramma classi UML è una rappresentazione grafica per costruire e visualizzare sistemi orientati agli oggetti.

La freccia verso l'alto indica che CalculatorEx eredita da Calculator
La notazione - indica attributi o funzioni privati o protetti
La notazione + indica attributi o funzioni pubbliche

.Investigating the Shooter Game's Generated Project and C++ Code

Gli sviluppatori di giochi scelgono di utilizzare i motori di gioco perché forniscono set di strumenti che aiutano ad accelerare
flussi di lavoro degli sviluppatori. In Unreal Engine, ciò che fa lo scripting C++ è programmare in base a Unreal

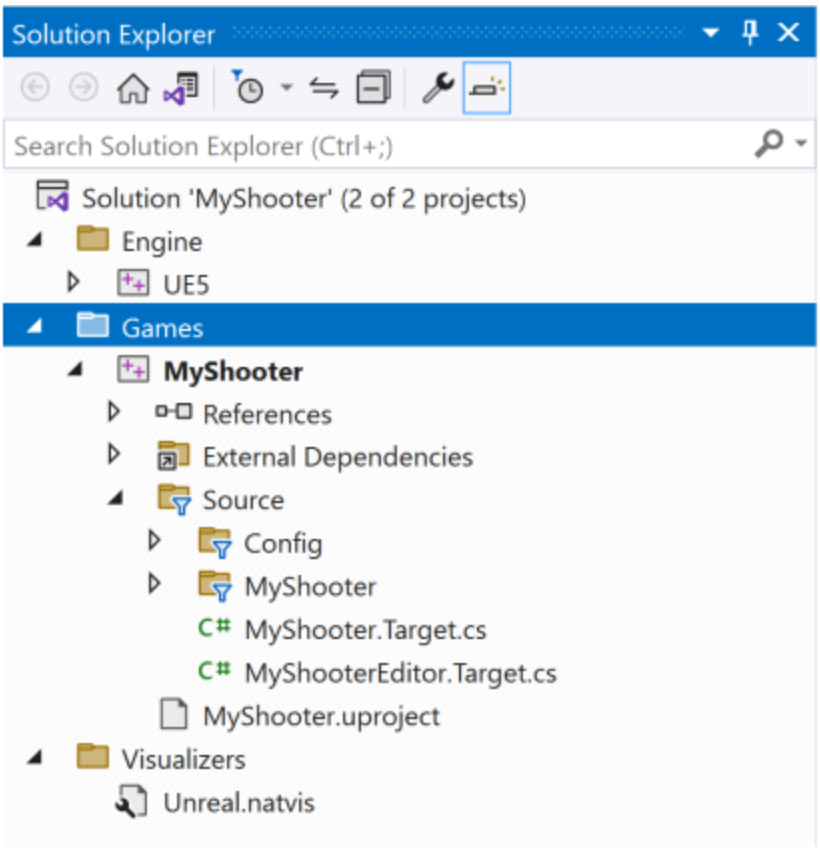
Classi e API predefinite del motore. //Un Api csonsente la comunicazione tra diversi computer, software e componenti di software//

-Understanding the MyShooter C++ project structure

Apri MyShooter in Unreal Editor e seleziona la cartella C++ Classes/MyShooter nella finestra Content Drawer. Qui puoi trovare cinque file di classe C++:

Fare doppio clic su qualsiasi file di classe C++ per aprire il progetto C++ in Visual Studio

Esaminiamo ora Solution Explorer:

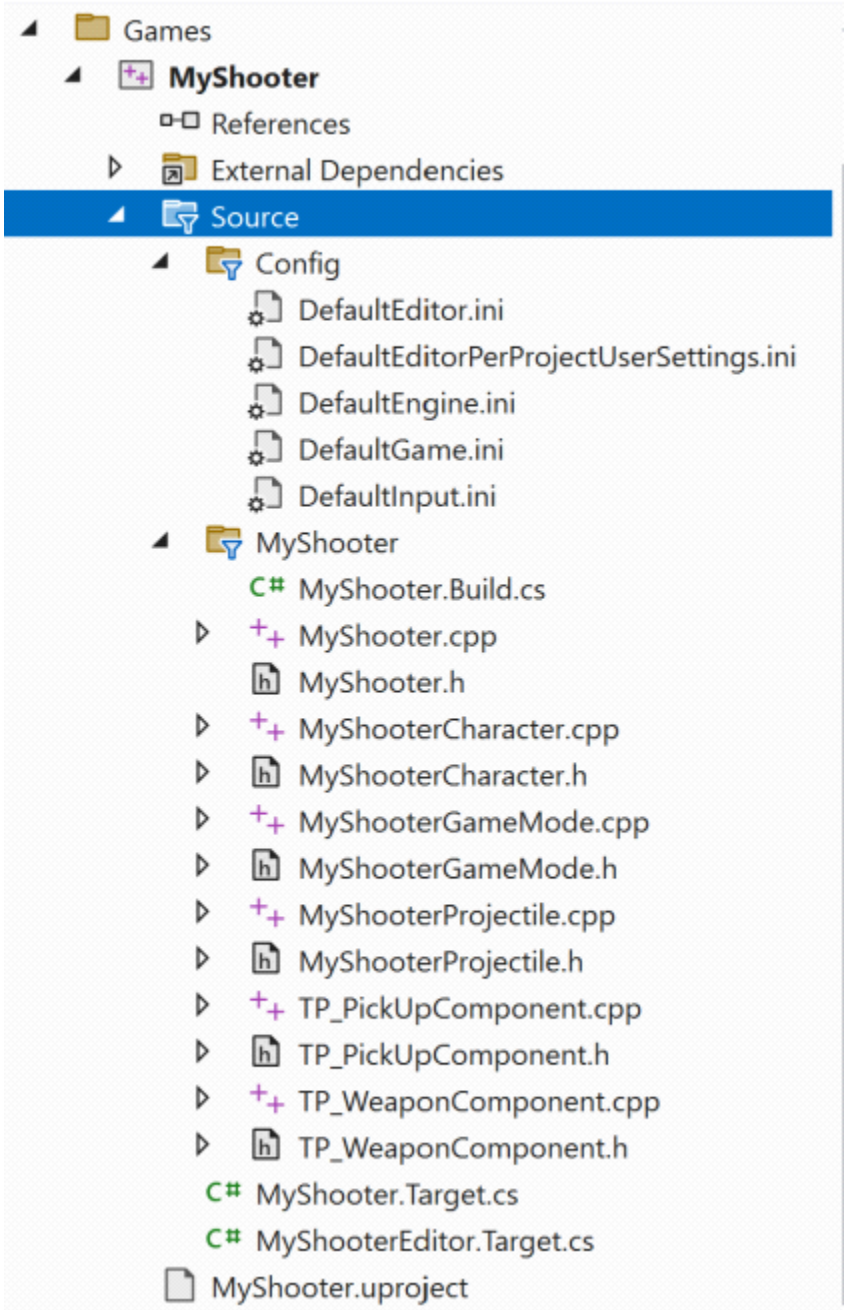


Il primo livello dell'albero ha tre nodi di cartella: Engine, Games, and Visualizers.

- La cartella **Engine** contiene il progetto Unreal Engine e il codice sorgente.
- La cartella **Games** contiene i progetti di gioco. Qui ha un solo progetto: MyShooter.
- La cartella **Visualizers** contiene i file con estensione natvis, che contengono configurazioni della sintassi XML per la visualizzazione di oggetti e altri dati in Visual Studio. Ciò significa che gli utenti possono visualizzare meglio i dati (in base alle proprie preferenze) durante la modifica o il debug del codice.

Per ora, ci interessa solo la struttura di MyShooter.

Questa cartella contiene tutti i file appartenenti al progetto del gioco. Gli sviluppatori in genere aggiungono nuovi file di origine al progetto nella sottocartella **Source**



- La cartella Riferimenti contiene riferimenti ad altre soluzioni o progetti condivisi. Non faremo nulla sui riferimenti in questo libro, quindi tenetelo vuoto

La cartella External Dependencies contiene tutti i file di intestazione delle dipendenze necessari per scrivere il codice del gioco.

- La cartella Source contiene i file di progetto MyShooter.uproject Unreal e due sottocartelle:

1. La sottocartella Config contiene tutti i file di configurazione .ini.
2. La sottocartella MyShooter, che ha un nome identico al progetto Unreal, contiene i file di compilazione e di destinazione (si tratta di file C#.cs). Tutti i file di codice sorgente C++

generati (file .cpp e .h) vengono inseriti nella sottocartella MyShooter.

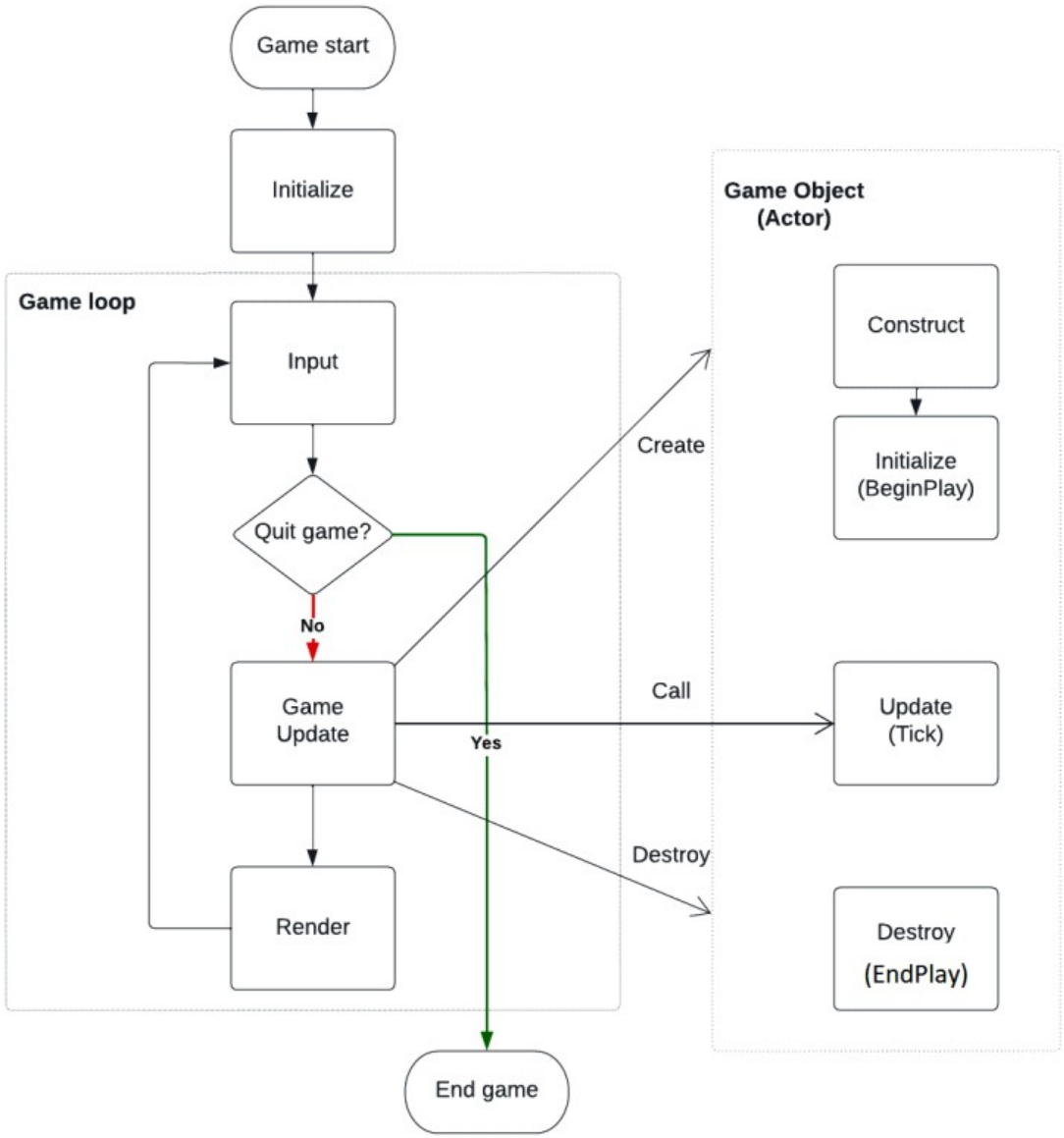
-Understanding the game program structure

Un programma di gioco tipico e semplice di solito ha tre fasi:

3. inizializzazione del gioco,
4. ciclo del gioco e
5. fine del gioco,

e Unreal Engine le gestisce internamente.

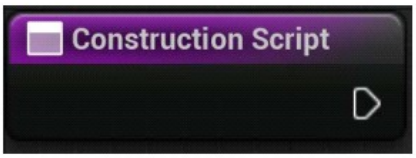
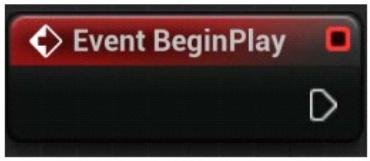
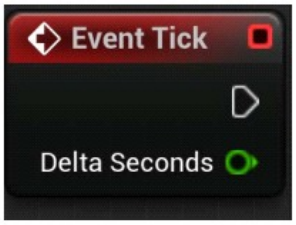
Il diagramma di flusso seguente illustra il flusso di gioco fondamentale e dovrebbe fornire un'idea della scrittura del codice di gioco per i moduli funzionali:



6. **Game initialization**, si verifica all'avvio di un programma di gioco. Tutte le attività di inizializzazione, come l'impostazione della modalità di visualizzazione, il caricamento dei contenuti richiesti, la generazione di oggetti di gioco e così via, vengono eseguite durante questa fase.
7. **Game loop,** è il periodo di gioco principale. Un ciclo di gioco comporta il recupero ripetuto degli input del giocatore, l'aggiornamento del gioco e il disegno della scena di gioco. Un ciclo di gioco completo si riferisce anche al completamento di un aggiornamento del frame.

Durante l'aggiornamento del gioco, è possibile creare oggetti di gioco (attori), ad esempio proiettili che vengono sparati.
Gli attori, OnConstruction() e BeginPlay().
Queste due funzioni vengono chiamate una sola volta, quando l'oggetto viene creato.
La funzione di evento Tick() dell'attore viene chiamata in ogni fotogramma.

- **Game end,** è una fase che chiude il gioco. Si occupa delle operazioni di pulizia, ad esempio l'eliminazione degli attori esistenti, il rilascio di memoria, l'arresto delle connessioni di rete e così via. A questo punto viene chiamata la funzione di evento EndPlay() dell'attore.

Actor Blueprint Event	Actor C++ Event Function	Comment
	`OnConstruction()`	Called when the actor is created
	`BeginPlay()`	Called after `OnConstruction()` when the actor is created
	`Tick()`	Called in every frame
	`EndPlay()`	Called before the actor is destroyed

-Getting familiar with the source code

Prima di tutto, diamo un'occhiata al file di intestazione MyShooterCharacter.h per la definizione di una tipica classe Unreal

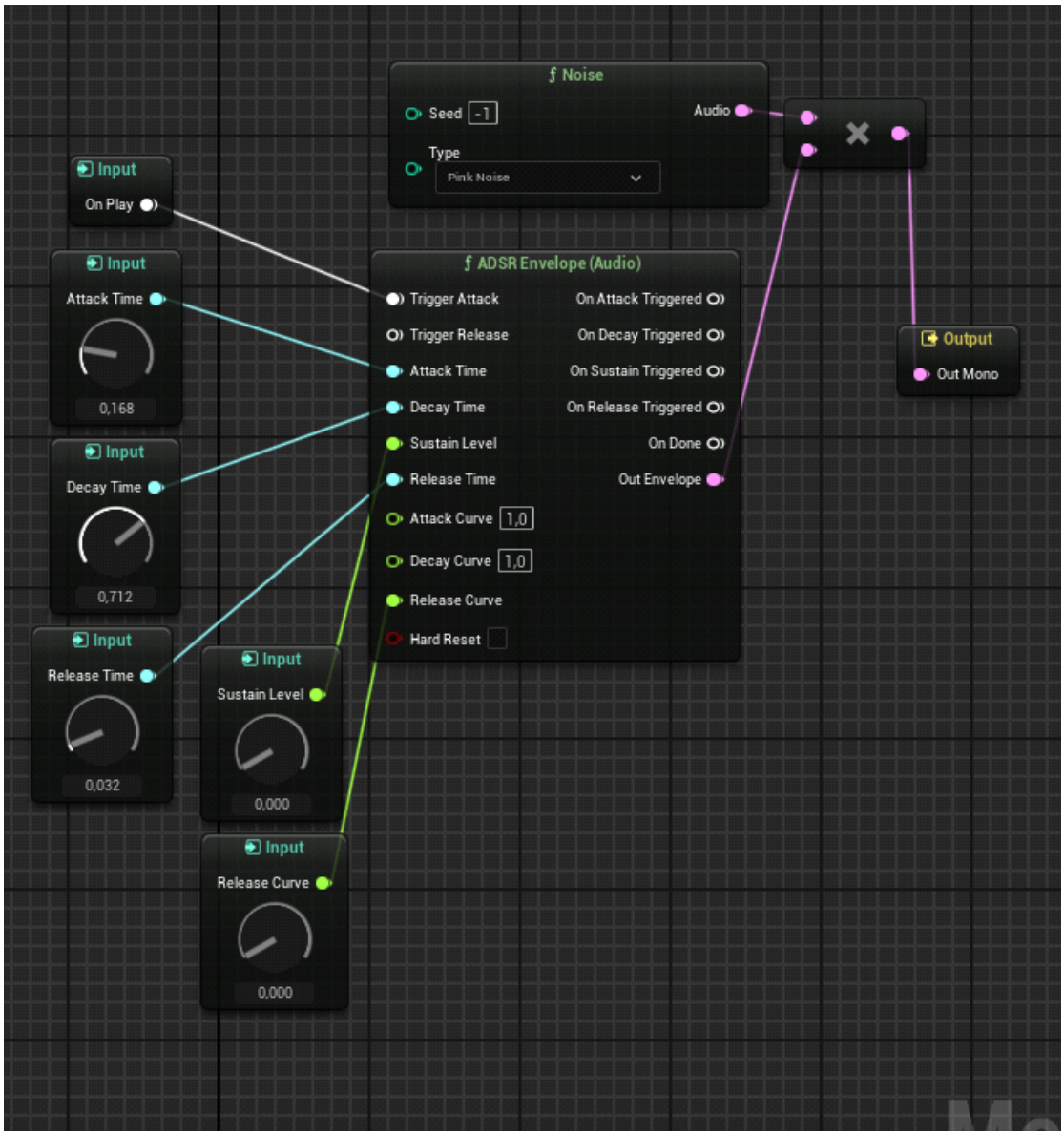
MyShooterCharacter.h
La classe MyShooterCharacter verrà utilizzata per creare i personaggi dei giocatori

STOQUA

////////////

- Std:count; La funzione std::count fa parte della libreria di modelli standard (STL) e viene utilizzata per determinare il numero di occorrenze di un elemento specifico all'interno di un determinato intervallo.
-libreria iostream

////////////


Noise
Seed
-1
Audio
Type
Pink Noise
Input
On Play
Input
Attack Time
0,168
Input
Decay Time
0,712
Input
Release Time
0,032
Input
Sustain Level
0,000
Input
Release Curve
0,000
ADSR Envelope (Audio)
Trigger Attack
Trigger Release
Attack Time
Decay Time
Sustain Level
Release Time
Attack Curve
1,0
Decay Curve
1,0
Release Curve
Hard Reset
On Attack Triggered
On Decay Triggered
On Sustain Triggered
On Release Triggered
On Done
Out Envelope
Output
Out Mono

////////////

Spawn sound attached

Shortcut utili per risalire agli asset che hanno i materiali ALT + SHIFT + R

//Gameplay Audio Volumes
La classe deve avere in Routing, APPLY AMBIENT VOLUMES TRUE
SUBMIX. default 2dreverb send aumentalo-

//Loudness Analyzer, crea un Loudness Nrt, clicca due volte su di esso, aggiungi un suono//

//La formattazione in stringa
La formattazione in programmazione è il processo di conversione di un’istanza di una classe, struttura o valore di enumerazione in una rappresentazione di stringa.//

//Utilizzo MIDI in UE, aggiungere MIDI DEVICE SUPPORT.
In un BP, utilizza un nodo Find MIDIDevices Cercare i MIDI DEVICE ATTIVI.
In base al loro ID, utilizzo Create MIDIDeviceController, con l'ID che voglio utilizzare.
Creo un istanza (Set e Get), e lo associo BIND all EVENT to On MIDI EVENT.
I valori più importanti saranno il Control ID, l'equivalente della nota midi (60 e è il do centrale), la velocity
da 0 a 127 (utile per mappare automatismi vari) e i Channel, quest'ultimi switchabili attraverso la funzione Switch on Int.
Perchè i Channel? Perchè alcuni fader del MIDI sono assegnati al Channel 1 o al Channel 2.

//Un carattere jolly o generico è un concetto di programmazione. In termini più semplici, un carattere jolly è la capacità di una funzione,
un metodo o un'altra chiamata di accettare qualsiasi classe o tipo di oggetto.

//Convertire variabile in variabile per mappa
In variables,Variable, Variable Type, cambia il TIPO DI VARIABILE IN MAP

//Due attori, uno di questi ha un dispatcher eventi, questo dispatcher sarà chiamato Call, insieme ad un evento del suo BP (On component Begin Overlap).
Il secondo attore ha un suo evento personalizzato, ha una variabile che fà riferimento al BP dell'altro attore (Basta scrivere tra le variabile, il nome del BP di quell'attore
e selezionare Object Reference), dopodichè ho associato un evento del Secondo attore all'evento personalizzato del primo tramite il riferimento di quest'ultimo
esempio Boss Died - Target
    Bind event to Boss Died a Event Begin Play
Creo dalla funzione Bind, un evento personalizzato che mi chiami (call) la funzione principale del primo attore

Basta creare un dispatcher, creare un riferimento in un altro BP

I dispatcher di eventi sono un metodo di comunicazione dell'attore in cui un attore invia un evento e gli altri attori che sono in ascolto di tale evento ricevono una notifica.

1. Primo Attore invia evento Boss Died.

2. Secondo Attore, tramite il riferimento al BP del primo attore riceve una notifica e la chiama quell evento, per associarlo ad un suo evento.

//Per quanto riguarda i Submix, è sempre utile avere come base il Master Submix
Gli Audio Volume sono utili per l'aggiunta di Submix che funzionano all'interno o all'esterno del Audio Volume

//Proxy
Aggiungi la componente Proxy. (E' un Audio Gameplay Volume)

In detail setta Audio Gameplay > Proxy to AGV Primitive Proxy.

Per le condizioni

In detail setta Audio Gameplay > Proxy to AGV Primitive Proxy.

Pigia Class Settings, Detail > Add Interfaces > Implemented Interfaces > AudioGameplayCondition

Avrai due Interfacce Condition

1. Condition Met viene utilizzata per impostare qualsiasi condizione che genera un valore booleano.
Con esso, puoi creare i tuoi criteri personalizzati per ciò che definisce l'essere all'interno o all'esterno del volume.

2. Condition Met Position è simile all'interfaccia Condition Met, ma consente di accedere alla posizione dell'ascoltatore per l'utilizzo all'interno della condizione personalizzata.

L'interfaccia Condition Met Position è simile all'interfaccia Condition Met, ma consente di accedere alla posizione dell'ascoltatore per l'utilizzo all'interno della condizione personalizzata.

Nodi
1.Is Shift Down = Restituisce true se uno dei tasti Maiusc era premuto quando si è verificato questo evento
2.Get Modifier Keys State = Restituisce uno snapshot dello stato dei tasti di modifica memorizzati nella cache per l'applicazione.
3. Break Vector = Suddivide un vettore in X, Y, Z

//Priority
Assegnare maggiore Priority in Details nelle sorgenti sonore per sentire di più determinati suoni rispetto che altri.

//Audio Bus, creare un Bus principale, assegnare questo Bus ad un suono (Modulatore), creare effetto Filtro, assegnare all'effetto filtro
il Bus, infine creare un Source Effect Chain Preset, con il Filter che ha il Bus.
Aggiungi al suono da modulare questo Source Effect Chain Preset.
Questa funzione in pratica racchiude una serie di segnali sonori in un Bus, e utilizza questo Bus

per modulare un altra sorgente sonora (grazie ad effetti come il filtro
che hanno associato il Bus).

//AudioLink è un'API che connette Unreal Audio Engine a software esterni, ad esempio altri motori audio
Cosa può trasmettere AudioLink?
- Sources
- Submixes
- Audio Components

//METASOUND per inviluppo
AD Envelope, o ADSR Envelope, entrambi utilizzai audio, l'Out Envelope andrà moltiplicato per output Audio della sorgente sonora (es, Sine).

//Create Event Node, puoi creare un EVENTO IN BASE ALLA FUNZIONE, clicca con il tasto destro e digita Create Event, apri la finestra in basso Create Matching Event, o Create Matching Function

//Il timestamp, o marca temporale, è un numero che esprime una grandezza temporale.

//Quartz Music System

Creo il Clock, assieme al suo Subsystem (Quartz Subsystem) e la sua Handle (Quartz Handle Clock), setto la sua Handle con dei BPM (Nodo Set Beats For Minute).
Quindi creo il suono, e la sua Audio Componenta, verrà suonata con Play Quantized.
Fai partire il Clock (Start Clock)

//Metodo Quartza music system

Crea l'orologio e il metronomo, sottoscrivi gli eventi e crea i suoni (Create Sound 1, Play Quantized, Evento delegato, Create Sound 2, Play Quantized, Evento delegato, ecc)

//Per far funzionare i tasti della tastiera associato agli eventi nei BP, vai nel Class Default del BP, in INPUT > AUTO RECEIVE INPUT > PLAYER 0
Selezione nel GRAPH, l'evento della tastiera, DETAILS > INPUT > CONSUME INPUT IN FALSE > OVERRIDE PARENT BINDING IN FALSE

//Box collision
Node

Set Box Extent = Modificare la dimensione dell'estensione del Box.

//Kismet è un potente sistema di scripting visivo all'interno di Unreal Engine che consente ai non programmatori di creare sequenze di gioco complesse con facilità.

//Per creare un elenco di valori all'interno dell'enumerazione creata, è sufficiente aprire il file e premere il pulsante "Aggiungi enumeratore".
Sentiti libero di aggiungere una descrizione per ogni elemento nell'elenco. Tuttavia, è

facoltativo.
Devi solo inserire il "Nome visualizzato".

//Funzioni K2:
Il prefisso K2 sta per “Kismet 2.0”. In passato, Kismet era il predecessore di Blueprints.

//Metodi Get e Set

Metodo Get (Getter): Questo metodo permette di leggere il valore di una variabile all’interno di un oggetto.
Metodo Set (Setter): Questo metodo permette di assegnare un valore a una variabile all’interno di un oggetto.

//Guardare il BP in funzione mentre il gioco è in PLAY.

//Tips for Slow Running Per diminuire carico CPU
.Disable Realtime Rendering
Dalle Viewport Options, disabilitare le opzioni in tempo reale. Ciò impedirà al motore di tentare di farlo
aggiorna costantemente la visualizzazione del Viewport (Ctrl + R lo attiverà o disattiverà).
.Cambia il numero in Viewport Option Far View Plane
Questo è impostato su 0.0 per impostazione predefinita e significa che c'è una vista infinita aereo in modo che tutti gli oggetti vengano renderizzati, indipendentemente dalla loro distanza.
È possibile impostare un valore diverso su
ridurre il carico di rendering man mano che vengono renderizzati meno oggetti.
.Build Lighting
Garantire che la build dell'illuminazione sia aggiornata può aiutare a ridurre il carico sulla CPU durante la modifica
il tuo livello, poiché non deve generare texture temporanee.
.Engine Scalability Settings
Setta in low
.Cala i frame rate in project settings

//Nelle sorgenti sonore, Voice Managements, visualization mode,
-Restart
-Play when silent

//Console Command
au.Debug.Sounds 0 Disable 1 Enable
au.Debug.SoundReverb 0 Disable 1 Enable

//Shortcut Sound Cue
Fare clic e trascinare i cavi tra i pin per connettere i sistemi, Alt + clic per eliminare le connessioni e
Ctrl + clic per spostare una connessione esistente su un altro pin.

//Blueprint per l'ambiente, posiziona un trigger, On Actor Begin Overlap. Flip Flop A, Fade out all'audio component del suono di fuori, Fade In alla componente del suono interno,

B è viceversa

//APPLY AMBIENT VOLUME, quando utilizzi Audio Volume, ricorda di applicare nella classe Apply Ambient Volume

//Esempio di Macro in c++

```
#include <iostream>
#define INSTAGRAM FOLLOWERS
#define FOLLOWERS 138

int main() {
   std::cout << "Geeks for Geeks have " << INSTAGRAM << "K followers on Instagram" <<
std::endl;
   return 0;
}
```

Output: “Geeks for Geeks have 138K followers on Instagram”

//In Details >> Actor >> Advanced >> Tags è un array e puoi aggiungere tutte le proprietà di testo che desideri.
Fai clic sull'icona più per aggiungere un tag. Gli spazi sono consentiti, anche se non sono sicuro che li userei davvero io stesso.
Essendo un array e tutto il resto, possiamo anche aggiungere più tag.

//Trace by channel Draw Debug Type
For One Frame:La linea di tracciamento verrà visualizzata per un solo fotogramma sullo schermo.
For Duration:La linea verrà visualizzata per la durata impostata in "Tempo di estrazione".
Persistent:Le tracce delle linee rimangono sullo schermo e non scompaiono.

//Spostamento componente tra una serie di attori, utilizzando i set timer by event (delegati)
Ho creato un BP, con una componente audio di un corvo, che si sposta tra 4 attori

//Per i suoni ambientali, gli area loops, utilizzare spazializzazione con NON SPATIALIZED RADIUS alto, per far si che il suono sia allo stesso livello
di volume quando si è dentro il raggio della sfera

//Key and Gamepad Inputs
The Bindings menu (Edit/Project Settings/Input/Bindings)

Variabili
Puoi creare automaticamente <Get> o <Set> tenendo premuto Ctrl o Alt mentre trascini nel file ➡Grafico eventi.)

Attaccare oggetti dalla Outliner
Nella lista Outliner, clicca con il tasto destro, seleziona Attach to >> scegli l'oggetto a cui attaccarlo
Puoi anche collegare gli attori tra loro tramite i progetti, utilizzando <Attach To> , <Attach Actor

To Actor> o <Attach Actor To Component> (quale utilizzerai dipenderà da come
vuoi che l'allegato funzioni, ma molto probabilmente utilizzeresti <Attach Actor To Actor>)

Audio Finished
Questo è un evento che può essere ottenuto da un componente audio all'interno di un Blueprint, oppure trascinandolo
il valore restituito da un nodo <Play Sound attached> o da un riferimento a una variabile del componente audio.
Cerca "Fine" e seleziona Assegna su audio terminato. Questo creerà la funzione <Bind> e il file <Evento personalizzato> che si attiverà al termine della riproduzione dell'audio.

BIND EVENT TO ON AUDIO FINISHED

Shortcut BLUEPRINT

RMB Fare clic, tenere premuto e trascinare: spostare il grafico
Rotellina del mouse: zoom avanti/indietro
LMB Fare clic sul nodo: Selezionare il nodo
LMB Fare clic, tenere premuto e trascinare nello spazio vuoto: selezione del riquadro di selezione
LMB Fare clic sui nodi selezionati, tenere premuto e trascinare: Sposta gli elementi selezionati
Fare clic con il tasto RMB in uno spazio vuoto: aprire il menu Azioni -Grafico
LMB Fare clic sul perno del nodo, tenere premuto e trascinare nello spazio vuoto: aprire il menu Azioni grafico
Fai clic con il pulsante sinistro del mouse sul pin del nodo, tieni premuto e trascina su un altro pin: crea nuove connessioni

Struct
Le variabili di struttura (o struct) contengono una serie di diversi tipi di variabili (un po' come un array, ma con
diversi tipi di variabili). Questi sono modi utili per memorizzare una varietà di informazioni diverse per il riutilizzo
all'interno dei tuoi Blueprint.

Casting
Il casting è una funzionalità estremamente utile all'interno del sistema Blueprint: è il punto a cui fai riferimento
un oggetto o attore e quindi trasmetterlo alla sua classe genitore in modo da poter accedere alle variabili al suo interno (o
get o set) e richiamare funzioni ed eventi al suo interno.

//SHORTCUT UE

.H (su un attore)

Nasconde l'attore selezionato.

.Ctrl + H

Mostra gli attori nascosti

.Ctrl + B

cerca l'oggetto, nell editor del blue print, sul content browser.

.CTRL + SHIFT + F

Apre content browser

.CTRL + E

Apre l'editor della skeletal mesh

.Maiusc + E

Seleziona tutti gli attori corrispondenti nel livello dello stesso tipo dell'attore selezionato.

.CTRL + G

Seleziona Gruppo

.Ctrl + A

seleziona tutti gli attori.

.(ALT + TRASCINARE ROTELLINA MOUSE)

cambio Pivot di un actor, cliccare sul pivot e tenere premuto ALT mentre si scorre la rotellina del mouse.

.Per utilizzare lo snap dei vertici, tenete premuto il tasto V mentre utilizzate il gizmo di traduzione.
Mentre tieni premuto il tasto V, non appena inizi a spostare un oggetto, vedrai i vertici poligonali su cui puoi agganciare.

Icona surface snapping
L'allineamento della superficie consente agli attori di allinearsi al pavimento o a un'altra superficie.

//Tipi di anteprima

Play In Editor (PIE) a cui si accede direttamente tramite il pulsante Play situato sulla barra degli strumenti principale
Simulate In Editor (SIE) a cui si accede tramite il menu a discesa Play (Alt+S)

Quando si esegue questa operazione, il testo cambierà in Maiusc + F1 per il cursore del mouse (questo consente di restituire il controllo del mouse all'editor stesso).
Premere F8 per espellere dal personaggio giocante.

//Come utilizzare la Timeline per creare un INVILUPPO per la componente Audio
Assegna la componente Audio
Event Begin Play, aggiungi all'output exec una timeline
Setta il Time, aggiungi una float curve
Nel Ouput della Timeline, in UPDATE, assegna il set volume multplier dell'audio component
Al volume multiplier assegna il valore della nuova traccia della timeline
Nel tempo, il volume dell'audio component si abbasserà (o alzerà secondo i settaggi della curva)

//Simple Sound, si creano cliccando con il tasto destro sulla wave, o su una serie di Wave, potrai avere delle varianti di riproduzione in cui in base al "weight" verrà riprodotto uno dei suoni.

//Gli Ambient Sound possono essere disattivati alla automatica riproduzione una volta posizionati nel mondo, in Details, disabilita Auto Activate.

//Utilizzare Timeline con i Parametri

//Per registrare suoni con il mic, Aggiungi all'attore una Compononente Audio Capture
Assegna un submix alla componente audio.
Alla componente Audio Capture potrai assegnare Start, IL NODO.
Utilizza start recording output, assegnagli il submix creato
Finisci la registrazione con finish recording output, dicendogli il percorso, il nome e il submix creato
Per ascoltare in tempo reale la tua registrazione, utilizza una Source Bus, assegna a questa Source Bus, il submix creato
Suona la componente audio assegnandogli la Source Bus (Play)

//Playback Percent Event
Quando un componente audio (come UAudioComponent) riproduce un suono, questo avanza attraverso la sua durata.
Nel tuo progetto Unreal Engine, puoi impostare un evento percentuale di riproduzione allegandolo a un componente audio.
Definire le soglie percentuali di riproduzione desiderate (ad esempio, 25%, 50%, 75%).
Associa azioni o progetti personalizzati a queste soglie. Ad esempio:
Al 25% di riproduzione, attiva un effetto visivo.
Alla riproduzione al 50%, regola il volume in modo dinamico.
Al 75% della riproduzione, genera particelle aggiuntive.

//Actor che emette il suono/Pawn noise emitter component//Make noise
Actor che ascolta il suono//On hear noise//Fai qualcosa
Pawn senising component

//Per avere un riferimento a "self" se stesso, tasto destro Get Reference To Self.

//Analisi audio NON in tempo Reale
Quando crei l'audio component, se clicchi con il tasto destro sulla componente audio, ADD EVENT, appariranno tutti gli eventi a lui associati, tra cui Audio Finished

//Debugger Blueprint
.Breakpoints
I punti di interruzione sono indicatori che possono essere posizionati sui nodi del grafico Blueprint.
Quando un nodo con un punto di interruzione sta per essere eseguito durante la modalità PIE o SIE, il gioco verrà messo in pausa e lo sviluppatore verrà indirizzato al nodo nella vista grafico dell'editor del progetto.
.Stack
Lo stack di chiamate disponibile durante le sessioni di debug è concettualmente simile allo stack di chiamate che si trova nella maggior parte degli ambienti di sviluppo C++.
Lo stack di chiamate in C++ è uno stack che contiene tutte le chiamate di funzione durante l'esecuzione del programma.
.Watches
Tengono traccia dei valori dei pin dei nodi Blueprint in modo che possano essere visualizzati durante le sessioni di debug.
Clicca con il tasto destro sul pin del valore corrente del noto, esempio print string, clicca con il tasto destro In String, Watch this Value, mostrerà il valore corrente
.Execution Trace
Mostra un elenco dei nodi eseguiti con il più recente in alto.
.Data Flow Questo strumento è progettato per aiutare gli sviluppatori a tracciare il flusso dei dati all’interno delle loro applicazioni
//
.Get Cooked FFTData function = Consente di recuperare i dati spettrali per tutti i suoni riprodotti su un componente audio
FFT (Fast Fourier Transform) è un algoritmo matematico utilizzato per analizzare e trasformare i segnali dal dominio del tempo al dominio della frequenza.
FFTData rappresenta le informazioni spettrali ottenute tramite l'analisi FFT. Fornisce dettagli sui componenti di frequenza presenti in un segnale audio.
L'output del nodo sarà una serie di strutture di dati spettrali delle onde sonore, ciascuna contenente informazioni sull'ampiezza di tali frequenze.
Magnitude of frequency, la grandezza di una frequenza si riferisce al suo valore assoluto o intensità.
.Get Cooked Amplitude Envelope Data = fornisce l'accesso ai dati dell'inviluppo di ampiezza del tempo corrente di tutti i suoni riprodotti su un componente audio.
L’inviluppo dell’ampiezza rappresenta i cambiamenti dinamici nell’ampiezza (volume) di un suono nel tempo.

//Il nodo For Each Loop with Break consente di scorrere una raccolta (come un array) ed eseguire azioni specifiche su ciascun elemento.
È utile quando desideri interrompere prematuramente il ciclo in base a determinate condizioni.

//Virtualization In Unreal Engine, il concetto di virtualizzazione si riferisce a varie tecniche e funzionalità che ottimizzano l'utilizzo della memoria,
migliorano le prestazioni e migliorano la scalabilità.

//Event dispatcher
Creo un BP, con un suo evento dispatcher, assegno un evento e (chiamo) Call questo event Dispatcher
Nel level Blueprint, Associo (Bind) l'evento della reference del Blueprint precedente, a questo ci allego un altro evento, a cui assegnerò un
altra funzione. Questo evento, rimarrà in ascolto fintanto che, l'evento associato dell'altro BP, sarà chiamato (Call).

//Audio Analysis, puoi associare (Bind) l'evento on audio playback percent all'evento begin play.
Quando l'audio andrà in play, l'evento (Poi creato, con custom event) analizzerà l'audio in base alle varibili che hai creato, LoudnessNRT, ConstantQNRT, OnsetNRT
NRT, sta per NON IN REAL TIME
Per ottenere la Durata in secondi di ogni variabile NRT, usa il Get della Variabile, esempio LoudnessNRT, manda un pin (per creare un nuovo nodo) e cerca Get duration in seconds.
LoudnessNRT, acquisiamo Loudness in un particolare momento
ConstantQNRT, riferito alla intensita di ogni singolo pitch che forma un suono. Scala di Do, dal Do M, al DoM successivo, 8 node, 8 bande d'intensità
OnsetNRT, restituisce quando un particolare suono inizia.

//Gli enumeratori possono essere utilizzati per creare un set limitato di valori che rappresentano uno stato, una scelta o una condizione.

//Bitwise Operators in C
& (bitwise AND) = Il risultato di AND è 1 solo se entrambi i bit sono 1.
| (bitwise OR) = Il risultato di OR è 1 se uno qualsiasi dei due bit è 1.
^ (bitwise XOR) = Il risultato di XOR è 1 se i due bit sono diversi.
<< (left shift)) = in C prende due numeri, quello di sinistra sposta i bit del primo operando e il secondo operando decide il numero di posizioni da spostare.
>> (right shift) = in C prende due numeri, sposta a destra i bit del primo operando e il secondo operando decide il numero di posizioni da spostare.
~ (bitwise NOT) =
Variables, Math, Transformation = in C prende un numero e ne inverte tutti i bit.

//
La variabile Bitmask, rappresente lo stato di una delle mie condizioni
Esempio, Enumeratore Character, diviso in Mago, Paladino, Guerriero.
La mia variabile Bitmask rappresenterà o uno, o tutti gli stati del mio Enumeratore character.
Se uno degli stati che rappresenta è utilizzato restituisce 1, altriment 0
Queste variabili Bitmask sono utilizzabili con i Bitwise Operator.
Ogni "stato", quindi una variabile trasformata in Bitmask, dovrà essere convertita in Bool
La cosa utile di un Enumeratore, è che qualora si utilizzi il nodo switch associato al enum, il nodo eseguirà solo la condizione
che viene impostata dalla variabile di quell'enumeratore.
Esemio, Character, Mago, Paladino, Guerriero.
Nodo SWITCH ON ENUM CHARACTER, scegli quale delle 3 condizioni (o tutte) eseguire in base alla variabile di quell'enumeratore che avrai nell'input.

//Anatomy of a Blueprint
1. Classi Blueprint, risorsa che consente ai creatori di contenuti di aggiungere facilmente funzionalità alle classi di gioco esistenti.
2. Componenti.
3. Graphs
Graphs, contiene il comportamento in fase di progettazione e di gioco dei tuoi progetti. Lo script di costruzione viene eseguito
seguendo l'elenco dei componenti quando viene creata un'istanza di una classe blueprint, consentendoti di regolare dinamicamente l'aspetto del nuovo oggetto o attore.
EventGraph. scontiene un grafico del nodo che utilizza eventi e chiamate di funzioni per eseguire azioni in risposta agli eventi di gioco associati al progetto.
Viene utilizzato per aggiungere funzionalità comuni a tutte le istanze di un progetto.
4. Organizing and Reusing Script, Man mano che crei più script nei tuoi progetti, potresti scoprire che ci sono alcune sezioni di script che riutilizzi frequentemente
Functions
Function Calls
Creating Functions
Macros
Making Macros
Collapsing Graphs

/Il Blueprint ha questi tipi di collapse
1. Collapsing Nodes, Il metodo più semplice per comprimere, illustrato di seguito, è tramite la compressione dei nodi in un grafico collegato.
Questi vengono utilizzati principalmente per l'organizzazione e per mantenere le cose belle e ordinate all'interno del grafico degli eventi.
All'interno del Collapsing Nodes, puoi aggiungere pin ai nodi Input o Output dal pannello Dettagli.
2. Collapsing to Function, Un altro metodo che può essere utilizzato per comprimere i nodi è comprimerli in una funzione
3. Collapsing to Macro, È particolarmente utile per tutto ciò che fai frequentemente, come un'equazione matematica complessa.

//Blueprint Namespaces, ottimizzare il tempo necessario per aprire una risorsa blueprint nell'editor impedendo il caricamento di risorse non necessarie.
Permette di organizzazare e gestire delle risorse all'interno del sistema Blueprint.
Crea una BP, in class Settings, Blueprint Options, Blueprint Namespace, chiama il tuo Namespace.
Puoi Importare questo Namespace in altri BP, per magari passare le proprie funzioni, che puoi creare creando Blueprint Function Library, in class setting, Blueprint Options, Blueprint Namespace
aggiungi la Namespace create.
Ora qualsiasi BP, che avrà quella Namespace, potrà utilizzare le funzioni create.
Inoltre, la BP principale da cui è stata create la Namespace, potra essere usata come oggetto reference nelle variabili.
Basta digitare in ricerca variabili il NOME del BP in cui è stata create la Namespace

//Blueprint Function Libraries, sono una raccolta di funzioni statiche che forniscono funzionalità di utilità non legate a un particolare oggetto di gioco.

//Compilation Process

Clean the Class

Create Class Properties

Create Function List

Bind and Link the Class

Compile Functions*

Finish Compiling Class

Backend Emits Generated Code*

Copy Class Default Object Properties

//I bookmark sono utili per saltare da un nodo ad un altro, basta cliccare la serie di nodi che si vuole raggiungere con il bookmark.
Cliccare poi l'icona Bookmark, sul viewport (ha il simbolo
di un segnalibro, rinomina il bookmark.
Aggiung il Bookmark da Window

//Blueprint Header View, converte le classi di blueprint di Unreal Engine e le strutture di blueprint in codice C++.
Puoi utilizzarlo dal menu contestuale, clicca sul tasto destro sul BP, clicca PREVIEW EQUIVALENT C++ HEADER

//La scheda Class Defaults contiene informazioni sulle impostazioni e proprietà predefinite del Blueprint e su qualsiasi variabile contenuta nel Blueprint.
Default ; Elenca tutte le variabili e i relativi valori
Rendering : Contiene informazioni sulle proprietà relative a come (e se) un attore basato sul progetto viene visualizzato nel gioco.
Replication : Contiene informazioni sulle proprietà che determinano il comportamento del tuo attore in un gioco in rete, come la priorità della rete e la pertinenza per gli altri client
Input : Contiene informazioni su come gli attori basati su questo modello rispondono all'input.
Actor : Contiene informazioni sugli attori basati su questo progetto, ad esempio se possono essere danneggiati.
Tieni presente che per i progetti di livello, solo le proprietà Tag sono rilevanti in questa sezione.
Get Class Defaults, È possibile accedere alle proprietà definite all'interno dei Class Defaults anche in fase di runtime tramite l'utilizzo del nodo Get Class Defaults Blueprint.
Fai clic con il pulsante destro del mouse all'interno della classe Blueprint o del grafico della funzione e inserisci GetClassDefaults dal menu contestuale.
Con il nodo GET CLASS DEFAUT, è più facile accedere alle varibili della classe Blueprint in

runtime.

//Toggle Differences, clicca nel menu contestuale, due BP da analizzare, seleziona DIFF SELECT, scorri per trovare differenze
Rosso
Manca qualcosa nel pannello di sinistra a destra (sottrazione.)
Verde
Manca qualcosa nel pannello di destra a sinistra (aggiunta).
Ciano
Qualcosa è cambiato.
Grigio
Manca qualcosa nel pannello di sinistra a destra (nodi spostati, commenti).

//
.Interface Blueprint è una raccolta di una o più funzioni - solo nome, nessuna implementazione - che possono essere aggiunte a altri progetti.
.Level Blueprint è un tipo specializzato di progetto che funge da grafico di eventi globali a livello di livello.
.Blueprint Macro Library è un contenitore che contiene una raccolta di macro o grafici autonomi che possono essere posizionati come nodi in altri blueprint.
Questi possono far risparmiare tempo in quanto possono memorizzare sequenze di nodi di uso comune complete di input e output sia per l'esecuzione che per il trasferimento dei dati.

//Interfaccia Utente Blueprint
1. Blueprint Editor Compiler Results, fornisce feedback sulla compilazione degli script Blueprint, indicandoti se la compilazione di uno script Blueprint ha avuto esito positivo o se si sono verificati errori o avvisi.
2. Blueprint Editor Debug Panel
3. Blueprint Editor Details Panel (Pannello Details), fornisce l'accesso in modifica alle proprietà degli elementi selezionati all'interno dell'editor del blueprint.
-Search Filter, Digita il nome della proprietà che ti serve e le proprietà sottostanti filtrano verso il basso
-Property Matrix, Digita il nome della proprietà che ti serve e le proprietà sottostanti filtrano verso il basso
-Visibility Filter, Ti consente di mostrare o nascondere proprietà modificate o avanzate, nonché di comprimere o espandere tutte le categorie.
-Collapsible Categories, Queste aree vengono utilizzate per raggruppare proprietà correlate e possono essere espanse e compresse utilizzando il piccolo triangolo bianco a sinistra del nome.
4. Blueprint Editor Find Result Panel, Find Result consente di rintracciare rapidamente una varietà di oggetti in base ai seguenti criteri:
.Node name
.Pin name
.Node Comment
.Property name
.Property value
SHORTUCT CTRL + F
5. Blueprint Editor Graph Editor, È qui che creerai le reti di nodi e fili che definiranno il tuo comportamento programmato

.Ctrl + clic + trascina da un pin all'altro
Sposta i fili dal pin di origine al pin di destinazione.
.Alt + clic su un pin, Rimuovere tutti i cavi collegati al pin selezionato.
6. Blueprint Editor Menu, Menu utilizzato per eseguire più azioni, come salvare, caricare e aprire finestre.
--File
-Save
-Open Asset, Richiama la finestra di selezione delle risorse, CTRL + P, aggiunge un Asset, e la sua componente associata.
-Save All
-Choose Files to Save, Apre una finestra di dialogo con opzioni di salvataggio per contenuto e livel
-Connect to Source Control, Se il controllo del codice sorgente è abilitato, apre una finestra di dialogo con le opzioni di check-in per contenuto e livelli.
-SEZIONE BLUEPRINT
-Compile, Compila il Blueprint
-Refresh All Nodes, Aggiorna tutti i nodi nel grafico per tenere conto delle modifiche esterne.
-Reparent Blueprint, Cambia il genitore del Blueprint aperto.
-Diff, Differenza rispetto alle revisioni precedenti. Richiede che il controllo del codice sorgente sia attivo.
-Developer, Apri il menu Sviluppatore, dove puoi modificare le impostazioni del compilatore e ricompilare moduli come l'editor grafico.
--Edit
-History
-Undo, CTRL + Z
-Redo, CTRL + Y
-Undo History, Visualizza l'intera cronologia degli annullamenti.
-SEZIONE SEARCH
-Search, Trova riferimenti a funzioni, eventi, variabili e pin nel progetto corrente.
-Find in Blueprints, Trova riferimenti a funzioni, eventi, variabili e pin in TUTTI i Blueprints
-Delete Unused Variables, Elimina tutte le variabili che non vengono mai utilizzate.
-Configuration
-Editor Preferences, Apre le impostazioni per l'editor.
-Project Settings, Apre le impostazioni per il progetto corrente.
-Plugins, Apre la scheda Browser plugin.
--Asset
-Find in Content Browser, Richiama il browser dei contenuti e accede a questa risorsa. CTRL + B
-Reference Viewer..., Avvia il visualizzatore dei riferimenti per mostrare a cosa fa riferimento la risorsa corrente e cosa fa riferimento alla risorsa corrente. ALT + SHIFT + R
-Size Map...Visualizza una mappa interattiva che mostra la dimensione approssimativa di questa risorsa e tutto ciò a cui fa riferimento ALT + SHIFT + M
--View
-Pin Visibility
-Show All Pins, Mostra tutti i pin su tutti i nodi.
-Hide Unused Pins, Nasconde tutti i pin senza connessioni e senza valori predefiniti.
-Hide Unconnected Pins, Nasconde tutti i pin senza connessioni. Questa opzione nasconderà i pin di input per gli input che sono stati impostati direttamente sul nodo.
-Zoom

-Zoom to Graph Extents, Adatta la vista corrente all'intero grafico. Tast 6
-Zoom to Selection, Adatta la vista corrente alla selezione Tasto num /
--Debug
-Breakpoints
-Disable All Breakpoints
-Enable All Breakpoints
-Delete All Breakpoints
-Watches
-Delete All Watches
--Window, dispone di una sottosezione specifica per la visualizzazione delle schede specifiche dell'editor dei blueprint
-Toolbar, Mostra la barra degli strumenti se attualmente non è visibile.
-Details, Mostra il riquadro Dettagli se attualmente non è visibile
-Debug, Mostra il riquadro Debug se attualmente non è visibile.
-Palette, Mostra il riquadro della tavolozza dell'editor del blueprint se non è attualmente visibile.
-My Blueprint
-Compiler Results
-Find Results
-Components
-Viewport
-Content Browser
-SEZIONE DEVELOPER TOOLS
-Blueprint Debugger
-Collision Analyzer
-Debug Tools, Mostra il pannello Strumenti di debug, che contiene un assortimento di funzioni per attività di debug comuni
(ricaricamento di texture, visualizzazione di atlanti di texture, correzione gamma, ecc...)
-Message Log, Apre il registro dei messaggi. È qui che andranno a finire gli errori o gli avvisi dell'editor.
-Output Log, Apre il registro di output. Se stai eseguendo il debug di qualcosa utilizzando Stampa, apparirà qu
-Visual Logger
-Class Viewer
-Device Manager
-Device Profiles
-Session Frontend
-Widget Reflector
-Project Launcher, Mostra il Project Launcher che ti consente di eseguire il tuo progetto su qualsiasi dispositivo opportunamente configurato e connesso.
-Plugins
-Reset Layout...Ripristina il layout predefinito per l'intero editor di Unreal Engine 4. Ciò richiede il riavvio dell'editor, ma riaprirà il progetto corrente.
-Save Layout
-Enable Fullscreen
6. Blueprint Editor My Blueprint Panel (My Blueprint)
Si tratta essenzialmente di uno schema del progetto che consente di visualizzare facilmente gli elementi esistenti del progetto, nonché di crearne di nuovi.
7. Blueprint Editor Palette, Il pannello Palette visualizza un elenco di tutte le funzioni e variabili

che possono essere utilizzate nei progetti.
8. Blueprint Editor Toolbar, I pulsanti della barra degli strumenti dell'editor blueprint forniscono un facile accesso ai comandi comuni necessari durante la modifica dei blueprint.
Class settings, Apre le Proprietà del progetto nel riquadro Dettagli.
Class Defaults, Mostra il pannello Impostazioni predefinite classe nella scheda Dettagli
9. Blueprint Editor Viewport

//Play Options,
Start Location, Camera Location, Default Player Start
Per decidere da dove partire quando parte il gameplay PIE o simuli il gameplay SIE

//Variabili in Blueprint
1. Variable Name
2. Variable Type
3. Instance Editable = Se la variabile è modificabile pubblicamente sulle istanze del progetto.
4. Blueprint Read Only = Se questa variabile può essere impostata dai nodi Blueprint o se è di sola lettura.
5. Tooltip = Informazioni aggiuntive su questa variabile, visualizzate quando il cursore si trova sopra la variabile.
6. Show 3D Widget = Se vero, consente all'utente di modificare la variabile vettore utilizzando un widget di trasformazione 3D nel viewport.
7. Expose on Spawn = La variabile dovrebbe essere esposta come pin durante la generazione del progetto.
Ti consente di impostare se la variabile deve essere accessibile al momento della generazione del Blueprint in cui si trova.
8. Private = Se la variabile è privata (i Blueprint derivati non possono modificarla).
9. Expose to Cinematics = La variabile dovrebbe essere esposta affinché Sequencer la modifichi?
10. Config Variable = Consente di impostare la variabile da un file di configurazione.
11. Category = Consente di classificare la variabile in base a un'etichetta specificata. Le variabili verranno ordinate in base alla loro etichetta.
12. Replication = Questa variabile dovrebbe essere replicata sulla rete o no?
Ctrl Get variable
Alt Set Variable
--Advanced Variable Properties
13. Config Variable = Utilizza questa opzione per personalizzare i valori predefiniti e il comportamento delle variabili tra diversi progetti e configurazioni.
14. Transient = Non serializzare e riempire a zero al caricamento.
15. SaveGame = Essere serializzato per i giochi salvati
16. Advanced Display = Essere nascosto nelle finestre dei valori predefiniti della classe per impostazione predefinita.
17. Multi line = Essere in grado di visualizzare più righe. Per aggiungere una nuova riga durante la modifica della variabile, premere Maiusc+Invio.
Nota: questa opzione si applica solo alle variabili String e Text.
18. Deprecated = Tutti i nodi che fanno riferimento alla variabile produrranno un avviso del compilatore che indica che deve essere rimosso o sostituito.
19. Deprecation Message, È possibile specificare un messaggio da includere nell'avviso.

//Struct Variables in Blueprints consentono di archiviare insieme diversi tipi di dati che

contengono informazioni correlate.
Una struttura è una raccolta di diversi tipi di dati correlati e tenuti insieme per un facile accesso.
Come accedere ai valori della struttura??
1. Vai sul Pin della variabile, SPLIT STRUCT PIN
2. Breaking Structs, utilizza un Break Struct node
3. Making Structs, Puoi anche creare una struttura con i dati giusti.
Per creare un nodo Make struct, trascina fuori un pin di input della struttura e seleziona Make [Nome struttura] dal menu contestuale.
4. Setting Members in Structs
A volte, le strutture possono contenere molti dati ed è necessario modificare solo alcuni elementi di quel set.
L'impostazione dei membri in una struttura ti consente di essere molto specifico su quali dati modificare, senza dover collegare tutti i pin di dati che rimangono costanti.
Per modificare quali membri sono disponibili tramite il nodo Set Members in Struct, selezionare il nodo.
Nel pannello Dettagli sono presenti caselle di controllo per ogni possibile membro da esporre come pin sul nodo.
Le variabili membro che non sono esposte non verranno modificate dal nodo Imposta membri nella struttura.
5. Using Custom Structs
Nel menu contestuale da Blueprints, seleziona Structure.

//Wires
Le connessioni tra i pin sono chiamate Wires. I fili possono rappresentare il flow of execution o il flow of data.
-Execution Wires, I fili tra i pin exec rappresentano il flusso di esecuzione.
I cavi exec vengono visualizzati come frecce bianche che si estendono da un pin exec di output a un pin exec di input. La direzione della freccia indica il flusso di esecuzione.
I cavi Exec producono un indicatore visivo durante l'esecuzione, il filo tra i relativi pin di esecuzione evidenzia il fatto che l'esecuzione viene trasferita da un nodo all'altro.
-Data Wires
I cavi dati collegano un pin dati a un altro pin dati dello stesso tipo.
Vengono visualizzati come frecce colorate e vengono utilizzati per visualizzare il trasferimento dei dati,
con la direzione della freccia che rappresenta la direzione in cui viaggiano i dati. Il colore di un filo dati dipende dal tipo di dat

//Tunnels
Il collapsed graph utilizza i nodi del tunnel per comunicare e interagire esternamente con il grafo che lo contiene.

//Creazione di codice in C++ che restituisce la rappresentazione binaria di un numero dato

```
#include <iostream>
#include <vector>

int x, y;//Due variabili intere
```

```
int main() {

  std::cout << "Enter a number: ";
  std::cin >> x;
  y = x;
  std::vector<int> vettore;//Vettore int

  while (x > 1) {//While Loop
    x = x / 2;
    vettore.push_back(x);
  }

  vettore.push_back(y);
  for (int val : vettore) {//For each Loop, ho creato una nuova variabile
    std::cout << val % 2 << std::endl;
  }
}
```

SPIEGAZIONE
La parola chiave for viene utilizzata per avviare un ciclo.
int val dichiara una variabile val di tipo int.
: indica che val eseguirà l'iterazione su ciascun elemento nel contenitore vettore.
vettore è il contenitore su cui viene eseguita l'iterazione.

Quindi, per ogni elemento nel vettore vettore, il ciclo assegna l'elemento alla variabile val e il corpo del ciclo esegue, elaborando o manipolando l'elemento val.

//Puoi utilizzare una macro che hai creato in un altro Blueprint, semplicemente trascinando nell'altro Blueprint.

//.Gameplay Tags
I tag di gioco possono essere utilizzati per identificare, classificare, abbinare e filtrare gli oggetti.
I tag di gioco sono etichette concettuali e gerarchiche con nomi definiti dall'utente. Questi tag possono avere un numero qualsiasi di livelli gerarchici.
"Famiglia.Genere.Specie", dove "Famiglia" è l'identificatore più ampio nella gerarchia.
I tag di gioco devono essere aggiunti al dizionario dei tag centrale affinché il motore ne venga a conoscenza.
1. Aggiungendoli o rimuovendoli manualmente nel file Project Settings menu.
2. Creando e modificando file .ini nella cartella "GameplayTags".
3. Creando risorse Data Table
Tutti e tre i metodi sono impostati in Project Settings, aprendo il Gameplay Tags

-Manually Adding Tags
Import Tags From Config nel Project Settings
-Editing .ini Files Directly
Per aggiungere tag utilizzando i tuoi file .ini, l'opzione Importa tag da configurazione deve essere abilitata.

I tag di gioco verranno quindi caricati da ciascuno dei file .ini trovati nel percorso del tuo progetto, che sono facili da modificare manualmente.

//- Unreal Engine Reflection System, L'Unreal Engine Reflection System incapsula le tue classi con varie macro che forniscono funzionalità del motore e dell'editor.
Quando si programma con Unreal Engine(UE), è possibile avere classi, funzioni e variabili C++ standard.
// l'incapsulamento è implementato utilizzando classi. Una classe combina dati e funzioni in una singola unità.
Questo metodo per nascondere i dettagli di una classe è chiamato astrazione.
Le classi possono contenere membri privati, protetti e pubblici.
Pubblico (public): I membri pubblici di una classe sono accessibili da qualsiasi parte del programma.
Protetto (protected): I membri protetti sono accessibili solo all'interno della classe stessa e dalle sue sottoclassi.
Privato (private): I membri privati sono accessibili solo all'interno della classe stessa.//

1. UObject = La classe base per gli oggetti in Unreal è UObject. Ogni classe definisce un modello per un nuovo attore o oggetto.
2. UCLASS = È possibile utilizzare la macro UCLASS per contrassegnare le classi derivate da UObject in modo che il sistema di gestione UObject ne sia a conoscenza.
3. TSubclassOf = TSubclassOf è una classe modello che fornisce l'indipendenza dai tipi UClass.
4. Structs = Le classi possono contenere strutture. Le strutture sono strutture di dati che aiutano con l'organizzazione e la manipolazione delle relative proprietà dei membri.
5. Unreal Smart Pointer Library = Le strutture possono essere definite da sole utilizzando la macro USTRUCT().
Unreal Smart Pointer Library è un'implementazione personalizzata di puntatori intelligenti C++11 progettata per alleviare il carico di allocazione e tracciamento della memoria
6. Interfaces = Le interfacce forniscono funzioni e comportamenti di gioco aggiuntivi che puoi implementare in classi multiple o diverse.
7. Metadata Specifiers = Gli specificatori di metadati controllano il modo in cui classi, interfacce, strutture, enumerazioni, funzioni o proprietà interagiscono con vari aspetti del motore e dell'editor.
8. UFUNCTION, and UPROPERTY = Le macro UFUNCTION e UPROPERTY rendono l'UE consapevole di nuove classi, funzioni e variabili

1.1.Objects
Unreal ha un sistema robusto per la gestione degli oggetti di gioco. La classe base per gli oggetti in Unreal è UObject.
La macro UCLASS può essere utilizzata per contrassegnare le classi derivate da UObjec
- The UCLASS Macro
La macro UCLASS fornisce all'UObject un riferimento a una UCLASS che descrive il suo tipo basato su Unreal.
Ogni UCLASS mantiene un oggetto chiamato Class Default Object (CDO), un oggetto "modello" predefinito, generato dal costruttore della classe.
Una UCLASS contiene un insieme di proprietà e funzioni che definiscono la classe.
- Proprietà e tipi di funzioni
Gli UObject possono avere variabili membro (note come proprietà) o funzioni di qualsiasi tipo.

- UObject Creation
Gli UObject non supportano gli argomenti del costruttore. Tutti gli UObject C++ vengono inizializzati all'avvio del motore e il motore chiama il costruttore predefinito.
Se non esiste un costruttore predefinito, il tuo UObject non verrà compilato.

1. NewObject<class>, Crea una nuova istanza con parametri facoltativi per tutte le opzioni di creazione disponibili.
Offre un'ampia gamma di flessibilità, inclusi casi d'uso semplici con un nome generato automaticamente.
2. CreateDefaultSubobject<class>, Crea un componente o un oggetto secondario che fornisce un metodo per creare una classe figlio e restituire la classe padre.

-Functionality Provided by UObjects
Non è necessario né opportuno utilizzare questo sistema in tutti i casi, ma ci sono molti vantaggi nel farlo, tra cui:

Garbage collection
Reference updating

-The Unreal Header Tool
Per sfruttare la funzionalità fornita dai tipi derivati da UObject,
è necessario eseguire una fase di preelaborazione sui file di intestazione affinché questi tipi raccolgano le informazioni necessarie

-Header File Format
Mentre l'implementazione di un UObject in un file sorgente (.cpp) è proprio come qualsiasi altra classe C++, la sua definizione in un file di intestazione (.h)
deve aderire a una determinata struttura di base per funzionare correttamente con Unreal Engine.

Un file di intestazione di base per una classe derivata da UObject potrebbe assomigliare a questo,
presupponendo che il derivato di UObject si chiami UMyObject e che il progetto in cui è stato creato si chiami MyProject:

```
#pragma once

   #include 'Object.h'
   #include 'MyObject.generated.h'//File d'intestazione del derivato di UObject

   UCLASS()//La macro UCLASS rende UMyObject visibile a Unreal Engine
   class MYPROJECT_API UMyObject : public UObject
   {
      GENERATED_BODY()//La macro GENERATED_BODY non accetta argomenti, ma imposta la
classe per supportare l'infrastruttura richiesta dal motore.

   };
```
-Updating Objects

Il ticchettio si riferisce al modo in cui gli oggetti vengono aggiornati in Unreal Engine.
Tutti gli attori hanno la possibilità di selezionare ogni fotogramma, fornendoti un modo per eseguire eventuali calcoli di aggiornamento o azioni necessarie.
Gli attori e gli ActorComponents hanno le loro funzioni Tick chiamate automaticamente quando vengono registrati per farlo, tuttavia,
gli UObject non possiedono alcuna capacità di aggiornamento incorporata.
Questo può essere aggiunto ereditando dalla classe FTickableGameObjec, utilizzando lo specificatore di classe inherits
Possono quindi implementare la funzione Tick(), che verrà chiamata ogni frame dal motore.
-Destroying Objects
La distruzione dell'oggetto viene gestita automaticamente dal sistema di raccolta dei rifiuti quando non viene più fatto riferimento a un oggetto.
Quando viene eseguito il Garbage Collector, gli oggetti senza riferimenti trovati vengono rimossi dalla memoria.
-UObject Instance Creation
1. NewObject
NewObject() è il metodo factory UObject più semplice. Accetta un oggetto e una classe esterni opzionali e crea una nuova istanza con un nome generato automaticamente.
Esempio :

```
template< class T >
   T* NewObject
   (
      UObject* Outer=(UObject*)GetTransientPackage(),
      UClass* Class=T::StaticClass()
   )
```

1.Outer, (Facoltativo) Un UObject da impostare come Esterno per l'oggetto da creare.
2.Class, (Facoltativo) Una UClass che specifica la classe dell'oggetto da creare.
-3.Return Value, Puntatore all'istanza generata della classe specificata.

3.1.TSubclassOf è una classe modello che fornisce l'indipendenza dai tipi UClass.

4.1. Una Struct è una struttura dati che consente di organizzare e manipolare le proprietà dei suoi membri.
Il sistema di riflessione di Unreal Engine riconosce le strutture come UStruct, ma non fanno parte dell'ecosistema UObject e non possono essere utilizzate all'interno delle UClass.
Un UStruct è più veloce da creare rispetto a un UObject con lo stesso layout dei dati.
UStruct supporta UProperty, ma non sono gestiti dal sistema Garbage Collection e non possono fornire la funzionalità di una UFunction.
-Implement a UStruct
Per trasformare una struttura in un UStruct, attenersi alla seguente procedura:

1.Apri il file di intestazione (.h) in cui desideri definire la struttura
2.Per definire la struttura C++, inserisci la macro USTRUCT sopra la definizione della struttura.
3.Includere la macro GENERATED_BODY() come prima riga della definizione.
Esempio :

```
USTRUCT([Specifier, Specifier, ...])
   struct FStructName
```

```
{
    GENERATED_BODY()
};
```

5.1. Unreal Smart Pointer Library è un'implementazione personalizzata di puntatori intelligenti C++11 progettata per alleviare il carico di allocazione e tracciamento della memoria.

6.1. Interfaces
Le classi di interfaccia vengono utilizzate per garantire che un insieme di classi potenzialmente non correlate implementi un insieme comune di funzioni.
-Interface Declaration
Dichiarare una classe di interfaccia è simile a dichiarare una normale classe Unreal, ma con due differenze principali:
1.Innanzitutto, una classe di interfaccia utilizza la macro UINTERFACE invece della macro UCLASS ed eredita direttamente da UInterface invece che da UObject.
Esempio :

```
UINTERFACE([specifier, specifier, ...], [meta(key=value, key=value, ...)])
    class UClassName : public UInterface
    {
        GENERATED_BODY()
    };
```

2.In secondo luogo, la classe UINTERFACE non è l'interfaccia vera e propria. È una classe vuota che esiste solo per la visibilità sul sistema di riflessione di Unreal Engine.
L'interfaccia effettiva che verrà ereditata da altre classi deve avere lo stesso nome di classe, ma con la "U" iniziale cambiata in "I"
Esempio :

```
#pragma once

    #include "ReactToTriggerInterface.generated.h"

    UINTERFACE(MinimalAPI, Blueprintable)
    class UReactToTriggerInterface : public UInterface
    {
        GENERATED_BODY()
    };

    class IReactToTriggerInterface
    {
        GENERATED_BODY()

    public:
        /** Add interface function declarations here */
    };
```

7.1.Metadata Specifiers utilizzate durante la dichiarazione di UClasses, UFunctions, UProperties, UEnums e UInterfaces
per specificare come si comportano con vari aspetti di Unreal Engine e dell'editor

8.1. UFUNCTION, and UPROPERTY
-Una UFunction è una funzione C++ riconosciuta dal sistema di riflessione di Unreal Engine.
Esempio :
UFUNCTION([specifier1=setting1, specifier2, ...], [meta(key1="value1", key2, ...)])
   ReturnType FunctionName([Parameter1, Parameter2, ..., ParameterN1=DefaultValueN1, ParameterN2=DefaultValueN2]) [const];
-Le Propertiss vengono dichiarate utilizzando la sintassi delle variabili C++ standard, preceduta dalla macro UPROPERTY che definisce i metadati delle proprietà e gli identificatori delle variabili.
Esempio :
UPROPERTY([specifier, specifier, ...], [meta(key=value, key=value, ...)])
   Type VariableName;
-Integers
uint8 : 8-bit unsigned
uint16: 16-bit unsigned
uint32: 32-bit unsigned
uint64: 64-bit unsigned
int8: 8-bit signed
int16: 16-bit signed
int32: 32-bit signed
int64: 64-bit signed
La convenzione per i tipi di dati integrali è "int" o "uint" seguito dalla dimensione in bit.
As Bitmasks: Le proprietà dei numeri interi ora possono essere esposte all'editor come maschere di bit.
Per contrassegnare una proprietà intera come maschera di bit, basta aggiungere "bitmask" alla sezione meta, come segue:
/*~ BasicBits appears as a list of generic flags in the editor, instead of an integer field. */
   UPROPERTY(EditAnywhere, Meta = (Bitmask))
   int32 BasicBits;
-Floating Point Types
Unreal utilizza i tipi a virgola mobile standard C++, float e double.
-Boolean Types
I tipi booleani possono essere rappresentati con la parola chiave bool del C++
Esempio :
uint32 bIsHungry : 1;
   bool bIsThirsty;
-Strings
Unreal Engine 4 supporta tre tipi principali di stringhe:
1. FString è un classico tipo di stringa "array dinamico di caratteri".
2. FName è un riferimento a una stringa immutabile senza distinzione tra maiuscole e minuscole in una tabella di stringhe globale.
È più piccolo e più efficiente da passare rispetto a un FString, ma più difficile da manipolare.
3. FText è una rappresentazione di stringa più robusta progettata per gestire la localizzazione.
Per la maggior parte degli usi, Unreal si basa sul tipo TCHAR per i caratteri. La macro TEXT() è disponibile per denotare valori letterali TCHAR.
Esempio :
MyDogPtr->DogName = FName(TEXT("Samson Aloysius"));//FName stringa, e macro TEXT()

.Coding Standard

.Containers in Unreal Engine
1.TArray: Arrays in Unreal Engine
La classe contenitore più semplice in Unreal Engine è TArray.
TArray è responsabile della proprietà e dell'organizzazione di una sequenza di altri oggetti (chiamati "elementi") dello stesso tipo.

-Creating and Filling an Array
TArray<int32> IntArray;
Questo crea un array vuoto progettato per contenere una sequenza di numeri interi.
//Gli allocatori in C++ sono componenti fondamentali per la gestione della memoria dinamica//
I TArray possono essere popolati in diversi modi.
1. Un modo è con la funzione Init, che riempirà un array con un numero di copie di un elemento:

```
 IntArray.Init(10, 5);
   // IntArray == [10,10,10,10,10]
```

2. Add and Emplace queste funzioni possono creare nuovi elementi alla fine dell'array:

```
Array<FString> StrArr;
   StrArr.Add    (TEXT("Hello"));
   StrArr.Emplace(TEXT("World"));
   // StrArr == ["Hello","World"]
```

Add (or Push), copierà (o sposterà) un'istanza del tipo di elemento nell'array.
Emplace, utilizzerà gli argomenti forniti per costruire una nuova istanza del tipo di elemento.
3. Append aggiunge più elementi contemporaneamente da un altro TArray o da un puntatore a un normale array C e la dimensione di quell'array:

```
FString Arr[] = { TEXT("of"), TEXT("Tomorrow") };
   StrArr.Append(Arr, ARRAY_COUNT(Arr));
   // StrArr == ["Hello","World","of","Tomorrow"]
```

4. AddUnique
aggiunge un nuovo elemento al contenitore solo se non esiste già un elemento equivalente.
L'equivalenza viene verificata utilizzando l'operatore del tipo di elemento==:

```
StrArr.AddUnique(TEXT("!"));
   // StrArr == ["Hello","World","of","Tomorrow","!"]

   StrArr.AddUnique(TEXT("!"));
   // StrArr is unchanged as "!" is already an element
```

5. Insert
aggiunge un singolo elemento o una copia di un array di elementi in un dato indice:

```
 StrArr.Insert(TEXT("Brave"), 1);
   // StrArr == ["Hello","Brave","World","of","Tomorrow","!"]
```

6. SetNum, la funzione può impostare direttamente il numero di elementi dell'array

```
StrArr.SetNum(8);
   // StrArr == ["Hello","Brave","World","of","Tomorrow","!","","",""]
```

7. Iteration
Esistono diversi modi per eseguire l'iterazione sugli elementi dell'array, ma il modo consigliato è utilizzare la funzionalità ranged-for di C++: FOR EACH

```
FString JoinedStr;
   for (auto& Str : StrArr)
   {
      JoinedStr += Str;
      JoinedStr += TEXT(" ");
   }
   // J
```

Infine, gli array hanno anche il proprio tipo di iteratore per un maggiore controllo sull'iterazione. Esistono due funzioni chiamate CreateIterator e CreateConstIterator che possono essere utilizzate rispettivamente per l'accesso in lettura-scrittura o in sola lettura agli elementi:

```
for (auto It = StrArr.CreateConstIterator(); It; ++It)
   {
      JoinedStr += *It;
      JoinedStr += TEXT(" ");
   }
```

8. Sorting

Gli array possono essere ordinati semplicemente chiamando la funzione Sort:

```
StrArr.Sort();
   // StrArr == ["!","Brave","Hello","of","Tomorrow","World"]
```

9. Num

Possiamo chiedere all'array quanti elementi contiene

```
int32 Count = StrArr.Num();
   // Count == 6
```

10. Per rimuovoere elementi da un array

La funzione Rimuovi rimuove tutti gli elementi considerati uguali all'elemento fornito, in base alla funzione operator== del tipo di elemento. Per esempio:

```
TArray<int32> ValArr;
   int32 Temp[] = { 10, 20, 30, 5, 10, 15, 20, 25, 30 };
   ValArr.Append(Temp, ARRAY_COUNT(Temp));
   // ValArr == [10,20,30,5,10,15,20,25,30]

   ValArr.Remove(20);
   // ValArr == [10,30,5,10,15,25,30]
```

.TSet

I TSet sono una classe contenitore veloce per memorizzare (solitamente) elementi univoci in un contesto in cui l'ordine è irrilevante.

```
TSet<FString> FruitSet;
```

.Gameplay Architecture

Quando si programmano elementi di gioco utilizzando codice C++, ogni modulo può contenere molte classi C++.

Ogni classe definisce un modello per un nuovo attore o oggetto.

.Gameplay Classes

Ogni classe di gioco in Unreal Engine è costituita da un file di intestazione della classe (.h) e da un file sorgente della classe (.cpp)

L'intestazione della classe contiene le dichiarazioni della classe e dei suoi membri,
come variabili e funzioni, mentre il file sorgente della classe è il luogo in cui viene definita la funzionalità della classe implementando le funzioni che appartengono alla classe.

I prefissi per le classi di gioco sono:
1. A
Si estende dalla classe base degli oggetti di gioco generabili. Questi sono attori e possono essere generati direttamente nel mondo.
2. U
endi dalla classe base di tutti gli oggetti di gioco. Questi non possono essere istanziati direttamente nel mondo; devono appartenere ad un Attore.
Questi sono generalmente oggetti come Componenti.

-Class Headers
Le classi di gioco in Unreal Engine generalmente hanno file di intestazione di classe separati e univoci.
#include "ClassName.generated.h"

-Class Declaration
La dichiarazione della classe definisce il nome della classe, da quale classe eredita e, quindi, eventuali funzioni e variabili che eredita.
La dichiarazione della classe definisce anche altri comportamenti specifici del motore e dell'editor

-Class Specifiers
Quando si dichiarano le classi,
è possibile aggiungere identificatori di classe alla dichiarazione per controllare il modo in cui la classe si comporta con vari aspetti del motore e dell'editor.

.Programming Quick Start

- Create a New C++ Class
Nel Unreal Editor, clicca File, dopo selezione New C++ Class...
Scelgi la classe Actor
Rinomina l'attore

//Course
Your First Game In Unreal Engine 5
-Select Mode, è uno strumento specializzato per velocizzare e semplificare la creazione dell'ambiente in Unreal Engine 4.
-Con lo shortuct F10 posso aprire e chiudere il Tab dei Details, World Outliner ecc
-In Window Env ligh mixer per creare luce
-Crea un Blueprint per creare una logica che puoi replicare più volte
-Aggiungi una componente static mesh, abilita physics in Details
-Select Mode, Landscape, schiaccia il tasto Create. Imposta il pennello, disegna.
-Per creare un giocatore, crea tre BLUPRINT
1.GameBaseMode
2.PlayerController

3.Character
Apri il Blueprint di GameBaseMode aggiungi al Player controller class il Blueprint del player controller
al defaul spawn class il blueprint del character
In project settings in Default game mode, aggiungi il BP del game mode.
Nel livello aggiungi il Player start
Aggiungi una componente telecamere al BP del character
Per muovere il personaggio vai su il blueprint del player controller, aggiungi una logica
//I componenti attore (classe UActorComponent)
Actor Components, sono molto utili per comportamenti astratti come il movimento, l'inventario o la gestione degli attributi e altri concetti non fisici.
I componenti attore non hanno una trasformazione, ovvero non hanno alcuna posizione fisica o rotazione nel mondo.
Posso creare una blueprint componente attore, aggingere un event dispacther (Con magari una variabile) creare eventi customizzati, ecc.
Qualora dovessi creare un nuovo Blueprint (solo Actor), potro utilizzare la il componente Actor component create in precedenza e utilizzarlo
come componente di questo nuvo Bp, in questo modo posso anche utilizzare le sue funzioni

//Begin Play Course

Un modulo è una raccolta di classi
Per registrare dei suoni, puoi utilizzare in Metasound, i Wave Writer
Oppure in Window, Cinmetatics, Take recorder, Source, Microphone Audio
I metasound Patch sono come la Blueprint Function Library
Audio stream Caching, ottimizza l'utilizzo della memoria
Utilizza Prime on Load per i suoni critici, Details (della sorgente sonora), Loading
Nodi da guardare, Math, Variables, i rimanenti di audio e Pawn

//I control Bus Mix sono utili per creare un Mix che ti permette di gestire i vari Bus, abbassando il volume ecc
La Parameter Patch ha diverse curve, una di queste è customizzabile

//Wwise
WAAPI è l'acronimo di Wwise Authoring API, ovvero un'API che consente alle applicazioni esterne di comunicare con lo strumento Wwise Authoring.
Gli utenti possono controllare lo strumento di creazione Wwise chiamando queste API, in modo che <u>WAAPI possa anche essere inteso come un ponte tra il progetto UE4 e il progetto Wwise.</u>

//Moduli di Unreal Engine I moduli sono gli elementi costitutivi dell'architettura software di Unreal Engine.
Puoi organizzare il tuo codice in moduli per creare progetti più efficienti e manutenibili.
I moduli sono gli elementi costitutivi di base dell'architettura software di Unreal Engine (UE).
Questi incapsulano strumenti di editor specifici, funzionalità di runtime, librerie o altre funzionalità in unità di codice autonome.

//In Unreal Engine (UE), un componente primitivo è un elemento fondamentale utilizzato per

creare e manipolare la geometria all'interno del mondo di gioco.
Un componente primitivo è un tipo di SceneComponent in UE.
Contiene o genera una qualche forma di geometria, che viene generalmente utilizzata per il rendering o il rilevamento delle collisioni
Un componente primitivo eredita da diverse classi genitore:
UObjectBase
UObjectBaseUtility
UObject
UActorComponent
USceneComponent
UPrimitiveComponent
Sebbene esistano diverse sottoclassi di componenti primitivi, quelle utilizzate più frequentemente includono:
CapsuleComponent: genera la geometria di collisione (utilizzata per il rilevamento dei colpi) ma non viene sottoposto a rendering
StaticMeshComponent: contiene geometria precostruita di cui è stato eseguito il rendering e che può essere utilizzata per il rilevamento delle collisioni.
SkeletalMeshComponent: simile a StaticMeshComponent, ma progettato specificamente per le animazioni scheletriche.

//Il runtime o run-time (tempo di esecuzione) indica il momento in cui un programma per computer viene eseguito, in contrapposizione ad altre fasi del ciclo di vita del software. Tradizionalmente, questa contrapposizione era soprattutto intesa rispetto al tempo di compilazione o compile-time, relativa alla stesura e traduzione del programma sorgente.

//Bilanciamento di blueprint e C++
.Blueprint sono il linguaggio di programmazione basato su nodi di Unreal Engine e offrono un flusso di lavoro diverso rispetto a C++.
Per programmare con Blueprint, fai clic e trascina variabili, funzioni ed eventi insieme per creare la funzionalità all'interno di un attore,
come mostrato nella gif seguente. Con queste funzioni ed eventi, puoi persino comunicare e passare dati tra altri attori Blueprint.
Devi solo sapere che Blueprint è un metodo di programmazione C++ basato su nodi.
Con la modifica e la programmazione basate su nodi, è possibile creare rapidamente prototipi di codice senza dover aprire un IDE,
consentendo di eseguire l'iterazione delle idee più velocemente e quindi di convertire successivamente il progetto in codice C++.
Con Blueprints, puoi scrivere rapidamente codice più piccolo e leggero che va e viene nel tuo progetto senza sapere come programmare.
Tuttavia, i Blueprint non sono limitati alle sole classi Attore, ma possono essere utilizzati con una varietà di classi: modalità di gioco,
stati di gioco, personaggi e pedine, solo per citarne alcune.
Il codice C++ può essere esposto a Blueprint. Ci sono molte caratteristiche e funzioni di base che puoi usare con Blueprint.
Blueprint non è un linguaggio di programmazione chiuso per Unreal Engine,
ma è un modo per consentire ai non programmatori di programmare e per tutti i membri di un team di estendere il codice C++ scritto.
.C++

Con C++, ottieni un accesso più approfondito e più ampio alle funzionalità di base di Unreal Engine.
È possibile cambiare o modificare le caratteristiche e le funzionalità principali del motore e dell'editor stesso, a seconda delle esigenze del progetto.
-I blueprint possono essere un po' più costosi in termini di prestazioni rispetto a C++ perché vengono eseguiti su una macchina virtuale.
Anche se i blueprint supportano enumeratori e struct, alcuni non possono essere usati in C++ se fanno parte di un blueprint.

//Unreal Engine Modules
I moduli sono gli elementi costitutivi dell'architettura software di Unreal Engine.
Puoi organizzare il tuo codice in moduli per creare progetti più efficienti e manutenibili.
I moduli sono gli elementi costitutivi di base dell'architettura software di Unreal Engine (UE).
Questi incapsulano strumenti di editor specifici,
funzionalità di runtime, librerie o altre funzionalità in unità di codice autonome.
Tutti i progetti e i plugin hanno il proprio modulo principale per impostazione predefinita, tuttavia puoi definire altri moduli al di fuori di questi per organizzare il tuo codice.
-Setting Up a Module
1. Crea una directory per il tuo modulo al livello più alto della cartella Source del tuo progetto. Questa directory dovrebbe avere lo stesso nome del tuo modulo.
1.1 Crea un file [ModuleName].Build.cs nella directory root del tuo modulo e usalo per definire le dipendenze con altri moduli.
Ciò consente al sistema di compilazione Unreal di scoprire il tuo modulo
I file CS appartengono principalmente a Visual Studio di Microsoft. Un file con estensione CS è un codice sorgente per i linguaggi di programmazione C# e C# Net.
È essenzialmente utilizzato con Visual C#.NET sviluppato da Microsoft.
Puoi creare un file cs o cpp, dal blocco note, quando salvi il blocco note cambia il formato da .txt a .cs o .cpp .
2. Crea sottocartelle private e pubbliche all'interno della directory principale del tuo modulo.
4. Crea un file [ModuleName]Module.cpp nella sottocartella Private per il tuo modulo.
5. Per controllare come e quando viene caricato il modulo, aggiungi le informazioni di configurazione per il tuo modulo nel file .uproject o .uplugin.
Ciò include il nome, il tipo, le piattaforme compatibili e la fase di caricamento del modulo.
6. Elenca il tuo modulo come dipendenza nel file per qualsiasi modulo che dovrà utilizzarlo.
Ciò potrebbe includere il file per il modulo principale del tuo progetto. Build.csBuild.cs
7. Genera i file della soluzione per il tuo IDE ogni volta che modifichi il file o sposti i file di origine tra cartelle.
A tale scopo è possibile utilizzare uno dei metodi seguenti: [ModuleName].Build.cs
a. Run . GeneraFileProgetto.bat
b. Fai clic con il pulsante destro del mouse sul file del tuo progetto(.uproject), quindi fai clic su Generate Project Files. .uproject
c. Nell'editor Unreal, fare clic su File > Refresh Visual Studio Project.
-Understanding the Structure of a Module
Tutti i moduli devono essere inseriti nella directory Source per un plugin o un progetto.
La cartella principale del modulo dovrebbe avere lo stesso nome del modulo corrispondente.

.[ModuleName]

.Private

. [ModuleName]Module.cpp

. All .cpp files and private headers

.Public

.All public headers

.[ModuleName].Build.cs

-Configuring Dependencies in the Build.cs File

Il sistema di compilazione Unreal crea progetti in base ai file nei tuoi progetti e ai file nei tuoi moduli, non in base ai file della soluzione per il tuo IDE. Target.csBuild.cs

La soluzione IDE viene generata automaticamente durante la modifica del codice, ma Unreal Build Tool (UBT) la ignorerà durante la compilazione dei progetti.

Tutti i moduli richiedono un file inserito nella directory root del modulo affinché il sistema di build Unreal li riconosca. [NomeModulo].Build.cs

All'interno del tuo file, devi definire il tuo modulo come una classe ereditata dalla classe ModuleRules.

//<u>File .uproject e .uplugin:</u>

Questi sono file descrittori che forniscono informazioni di base sul tuo progetto o plugin.

Ad esempio, .uproject specifica le impostazioni a livello di progetto, mentre .uplugin fornisce informazioni sui plugin.

<u>File .target.cs</u>

Questi file definiscono le impostazioni di creazione per vari target (come gioco, editor, client, server o programma).

Ereditano dalla classe base TargetRules.

Quando si compila per una destinazione specifica, lo strumento di compilazione applica le impostazioni del file .target.cs corrispondente.

Ad esempio, puoi avere file .target.cs separati per le destinazioni del gioco e dell'editor.

Se non esiste un file .target.cs specifico per una destinazione, vengono utilizzate le TargetRules di base.

<u>Module Building:</u> La decisione su quale modulo compilare è gestita dal file .build.cs.

Nel file .build.cs, specifichi le dipendenze utilizzando PublicDependencyModuleNames o PrivateDependencyModuleNames

Puoi creare un progetto senza menzionare esplicitamente un modulo in .uproject o .uplugin, ma il compilatore emetterà un avviso.

Di seguito è riportato un esempio di un file semplice: [ModuleName].Build.csBuild.cs

```
using UnrealBuildTool;

   public class ModuleTest: ModuleRules
```

```
{

public ModuleTest(ReadOnlyTargetRules Target) : base(Target)

{

PrivateDependencyModuleNames.AddRange(new string[] {"Core"});

}

}
```

-Private and Public Dependencies
Dovresti usare l'elenco se usi le classi da un modulo pubblicamente, come in un file pubblico. Ciò consentirà ad altri moduli che dipendono dal tuo modulo di includere i tuoi file di intestazione senza problemi. PublicDependencyModuleNames.h
Dovresti inserire il nome di un modulo nell'elenco se vengono utilizzati solo privatamente, come nei file. Ove possibile, si preferiscono le dipendenze private,
poiché possono ridurre i tempi di compilazione del progetto.
PrivateDependencyModuleNames.cpp

-Using the Private and Public Folders
Se il tuo modulo è un normale modulo C++ (ovvero non è impostato su in your o), i suoi file C++ dovrebbero essere posizionati nelle sottocartelle Private e Public all'interno della directory root del tuo modulo.
ModuleTypeExternal.uproject.uplugin

-Implementing the Module in C++
Per esporre un modulo al resto del tuo progetto C++, devi creare una classe che si estende, quindi fornire quella classe alla macro. IModuleInterfaceIMPLEMENT_MODULE
#include "Modules/ModuleManager.h"

```
IMPLEMENT_MODULE(FDefaultModuleImpl, ModuleTest);
```

A)DefaultModuleImpl è una classe vuota che estende . Per un'implementazione più dettagliata, puoi scrivere la tua classe da implementare in questo file. IModuleInterface.cpp
B)IModuleInterface presenta diverse funzioni che si attivano quando il modulo viene caricato e scaricato, simili alle funzioni e nella classe **. AvvioSpegnimentoIstanza di gioco**

-Using Modules in Your Projects
Ogni volta che crei un nuovo progetto o plug-in Unreal Engine, verrà automaticamente configurato un proprio modulo principale, che puoi trovare nella cartella del progetto.
Puoi includere moduli esterni nel tuo progetto aggiungendoli al file per il modulo principale del tuo progetto. SourceBuild.cs

-Creating a Gameplay Module
1. Required Setup
Per seguire questa guida, inizia creando un nuovo progetto denominato MyProject e assicurati che sia un progetto C++.

Se stai seguendo un progetto solo Blueprint, crea un nuovo pezzo di codice C++ in Unreal Editor per convertirlo in un progetto C++.

2. Setting Up Your Directories

Per prima cosa è necessario impostare le directory per contenere il modulo e il suo codice.

a) Passa alla directory principale del tuo Source, quindi apri la cartella Sorgente.

b) All'interno della cartella Source del tuo progetto, crea una nuova cartella denominata ModuleTest. Questa servirà come directory root del tuo modulo.

c) All'interno della directory principale del tuo modulo, crea due nuove cartelle chiamate Private e Public.

La struttura della tua directory dovrebbe assomigliare a questa:

```
.MyProject(NomeProgetto)
    .Source
      .MyProject.Target.cs
      .MyProjectEditor.Target.cs
      .MyProject (Primary game module folder)
            .Private
            .Public
            .MyProject.Build.cs
            .Other C++ classes in your game module.
      .ModuleTest
            . Private
            .Public
```

3. Creating the Build.cs File

Quando Unreal Build Tool (UBT) cerca le dipendenze nelle directory del tuo progetto, ignora il file della soluzione IDE e esamina invece i file Build.cs nella cartella Source.

Unreal Build Tool (UBT) è uno strumento personalizzato che gestisce il processo di compilazione del codice sorgente di Unreal Engine in diverse configurazioni di compilazione

Unreal Engine è suddiviso in molti moduli. Ogni modulo ha un file .build.cs che controlla come viene compilato,

inclusi gli aspetti come le dipendenze tra i moduli, le librerie aggiuntive e i percorsi di inclusione.

Di default, questi moduli vengono compilati in DLL e caricati da un singolo eseguibile.

Ogni modulo deve avere un file Build.cs, altrimenti UBT non lo rileverà.

Per impostare il file Build.cs, attenersi alla seguente procedura:

a. Crea un file chiamato ModuleTest.Build.cs nella directory radice del modulo.

Apri questo file e aggiungi il seguente codice:

```
using UnrealBuildTool;

  public class ModuleTest: ModuleRules
  {
        public ModuleTest(ReadOnlyTargetRules Target) : base(Target)
        {

        }
  }
```

Ciò renderà il tuo modulo visibile al sistema di build Unreal.

b. Modificare il costruttore in modo che sia come segue:

```
 public ModuleTest(ReadOnlyTargetRules Target) : base(Target)//Costruttore
   {

       PrivateDependencyModuleNames.AddRange(new string[] {"Core", "CoreUObject",
"Engine"});

   }
```

Ciò aggiungerà i moduli Core, CoreUObject e Engine come dipendenze private per il tuo modulo. Ciò renderà disponibili diverse classi per questo modulo per i passaggi successivi di questa guida.

4. Implementing Your Module in C++
Sebbene il file Build.cs renda il tuo modulo rilevabile da UBT, il tuo modulo deve anche essere implementato come classe C++ in modo che il motore possa caricarlo e scaricarlo.
Fortunatamente, Unreal Engine include macro che possono semplificare questo processo per le implementazioni più comuni.
Per creare un file di implementazione rapida, attenersi alla seguente procedura:
a.Passa alla directory principale del tuo modulo e apri la cartella Privata. Crea un nuovo file chiamato ModuleTestModule.cpp all'interno di questa cartella.
Non è necessario creare un file .h per questa classe.
b. All'interno di ModuleTestModule.cpp, aggiungi il seguente codice:

```
   #include "Modules/ModuleManager.h"

   IMPLEMENT_MODULE(FDefaultModuleImpl, ModuleTest);
```

Ciò fornisce un'implementazione predefinita per il tuo modulo che lo rende disponibile al codice C++

5. Compiling Your Module
Dopo aver definito e implementato il modulo, il processo per compilarlo è molto semplice.

a. Fare clic con il pulsante destro del mouse su MyProject.uproject e fare clic su Genera file di Visual Studio per rigenerare la soluzione IDE.
Ciò garantirà che il tuo nuovo modulo diventi visibile nel tuo IDE e all'interno di Unreal Editor.

6. Including Your Module In Your Project
Il tuo modulo dovrebbe essere in grado di compilare, ma per utilizzare qualsiasi codice dal tuo modulo nel tuo progetto, devi registrarlo con il tuo file .uproject,
quindi creare una dipendenza per il modulo principale del tuo gioco.

a.Apri la directory root di MyProject, quindi apri MyProject.uproject in un editor di testo e modifica l'elenco "Moduli" come segue:

```
"Modules": [
```

```
    {
      "Name": "MyProject",
      "Type": "Runtime",
      "LoadingPhase": "Default"
    },
    {
      "Name": "ModuleTest",
      "Type": "Runtime"
    }
  ]
```

È possibile utilizzare questa voce dell'elenco per configurare in quale fase di caricamento verrà caricato e il suo tipo.
Questo esempio è un modulo runtime, quindi può essere utilizzato quando il progetto viene eseguito come applicazione autonoma o nell'editor.

b. Passare alla cartella MyProject/Source, quindi aprire il file MyProject.Build.cs. Aggiungi ModuleTest all'elenco PublicDependencyModuleNames. Dovrebbe essere il seguente:

```
PublicDependencyModuleNames.AddRange(new string[] { "Core", "CoreUObject", "Engine", "InputCore", "ModuleTest" });
```

Ora sarà possibile includere il codice del tuo modulo nel modulo di gioco principale.

7. Adding Code to Your Module
Puoi aggiungere manualmente file C++ alle directory pubblica e privata nel tuo modulo oppure puoi aggiungerli in Unreal Editor.
Questi passaggi ti guideranno attraverso l'utilizzo della Creazione guidata nuova classe per aggiungere codice a un modulo:
a) Apri il tuo progetto in Unreal Editor.
b) Aggiungi New C++ Class
c) Scegli Actor
d) Sulla riga di Name, in Runtime, cambia il module e scegli quello creato
f)Assegna un nome alla classe, imposta il tipo di classe su Public, quindi fai clic su Crea classe.
i file .h e .cpp della nostra classe si apriranno automaticamente nel tuo IDE.
Il file .cpp della tua classe verrà aggiunto alla cartella Privata del tuo modulo, mentre il suo file .h verrà aggiunto alla cartella Pubblica.
g) Apri ModuleActorBase.cpp, quindi aggiungi la seguente riga alla funzione AModuleActorBase::BeginPlay:

GEngine->AddOnScreenDebugMessage(0, 5.0f, FColor::Blue, TEXT("Hello, world!"));

8. Extending Code From Your Module

Infine, segui i passaggi seguenti per testare la tua nuova classe Attore e la capacità del tuo modulo di gioco principale di vederla.

a.Apri il tuo progetto in Unreal Editor.
Crea una Create a New Blueprint Class, quindi espandi l'elenco All Classes.
b. Scegli ModuleActorBase(Modulo creato),
Se non vedi ModuleActorBase nell'elenco delle classi, assicurati che la sua intestazione (file.h) sia nella cartella Public, che abbia l'identificatore
BlueprintType UCLASS e che tu abbia aggiunto il modulo ModuleTest alle dipendenze del tuo progetto.

//Per convertire un progetto in BluePrint, vai su Tools New Class c++

//Una volta creato il codice c++, c'è un tasto in basso a destra, "Recompiles and Reload c++ code..."
Una volta la compilazione sarà avvenuta
Crea un BP, utilizza l'attore che hai creato come classe.
Metti il BP nel livello, manda in Play

//- Invio a submix
Un submix master viene creato nel motore audio quando il motore audio viene inizializzato.

//- Importazione di audio multicanale

1. Fare clic sul pulsante Importa.
2. Individua e seleziona i tuoi file. .wav
3. All'interno della Libreria viene visualizzata una risorsa audio a quattro canali (denominata "Surround")

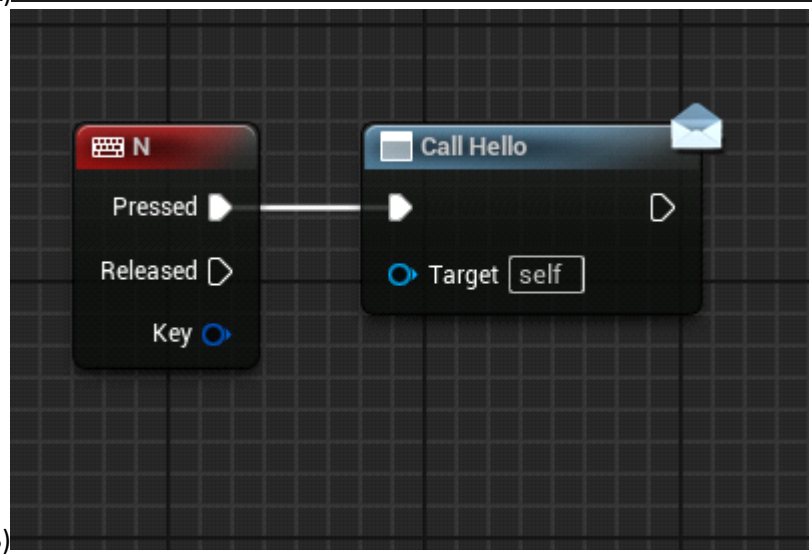
//Quando crei una classe C++ e la associ al BP
Dove viene dichiarato il costruttore nel formato .h, dichiara delle variabili, quelle variabili appariranno come variabili pubbliche di quel attore, e modificabili in tempo reale

//ESEMPIO DI C++ CALCULATOR
Crea un file Main.Cpp
Crea una Classe Calculator, .cpp, .h
.Nel file d'intestazione .h, dichiara gli attributi, metodi, e il costruttore se serve
    tipo NomeMetodo(Parametro 1, Parametro 2)
.Nel file d'origine, cpp, definisci quei metodi e costruttori.
    NomeClasse::NomeMetodo(argomento 1, argomento 2)
Chiama i metodi e costruttori nel file Main.cpp
    NomeClasse.Nomemetodo(argomento 1, argomento 2)

//Event Dispatchers esempio con immagini

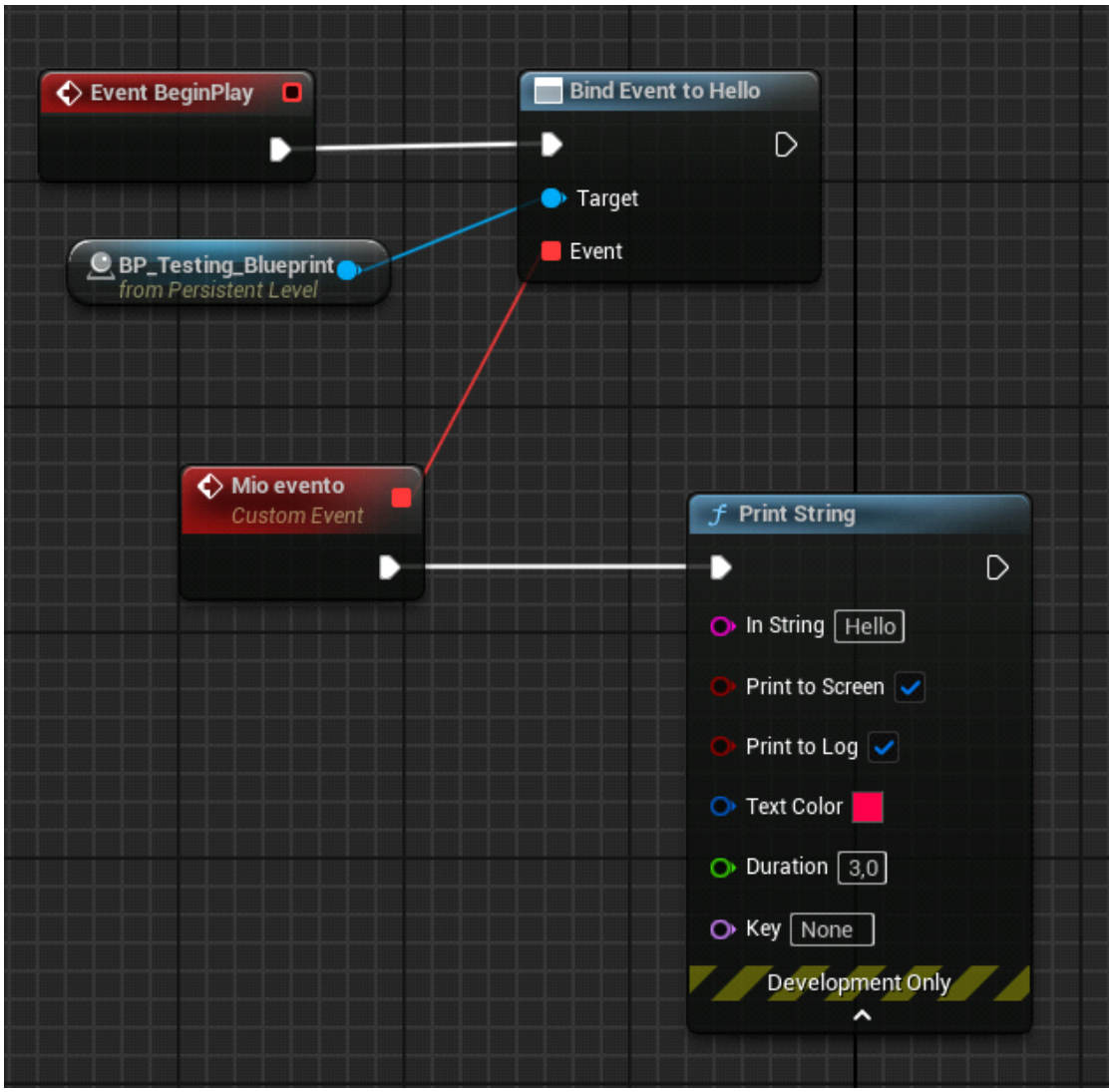
BLUEPRINT Actor

A)

B) 

LEVEL BLUEPRINT

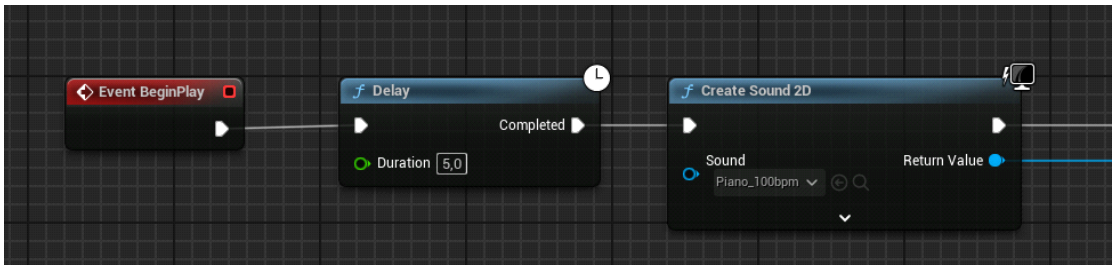
A)



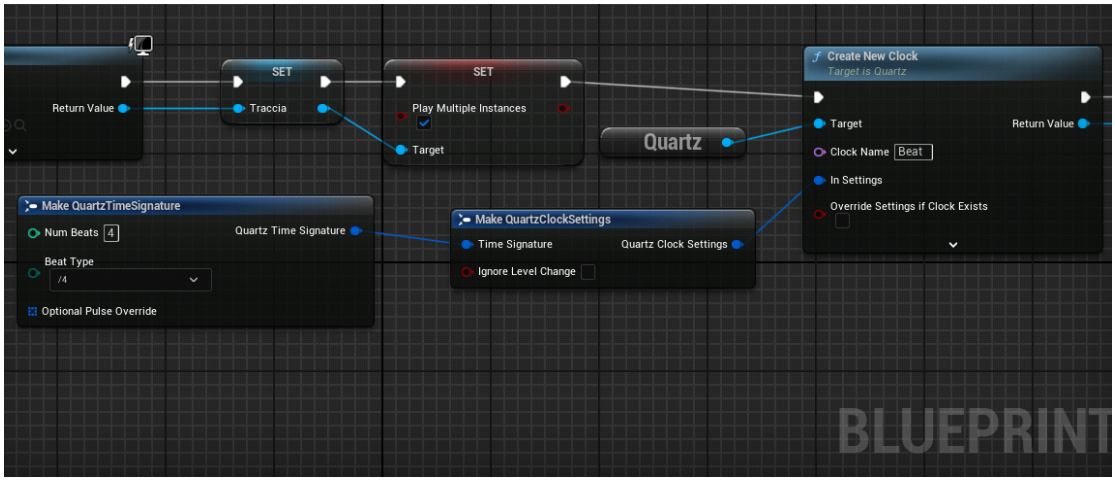
//Root Component

Il Root Component è importante per vari motivi:
Posizione e Rotazione: Determina la posizione e l'orientamento dell'intero oggetto.
Movimento: Quando sposti l'oggetto, il Root Component viene spostato insieme a esso.
Collisioni: Le collisioni tra oggetti si basano sulle forme dei loro Root Component.
Animazioni: Le animazioni spesso si basano sulla trasformazione del Root Component.

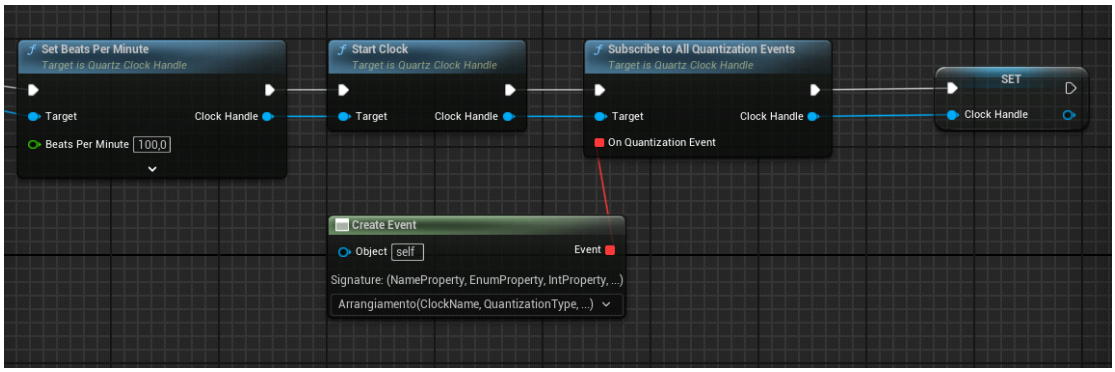
//Ricerca relativa a Quartz
Parto col creare un suono



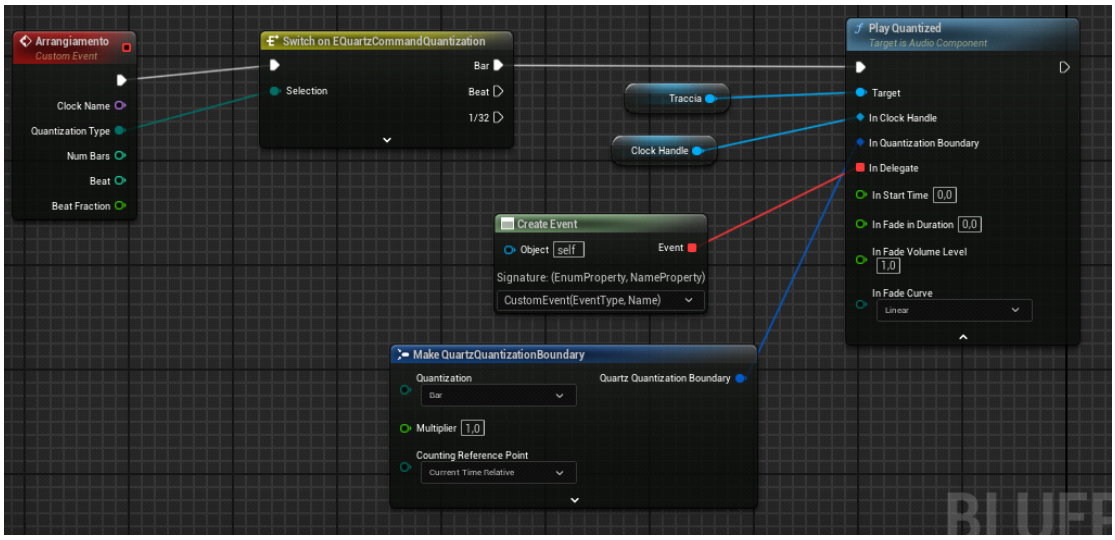
Setta la componente audi o con Play Multiple Instances



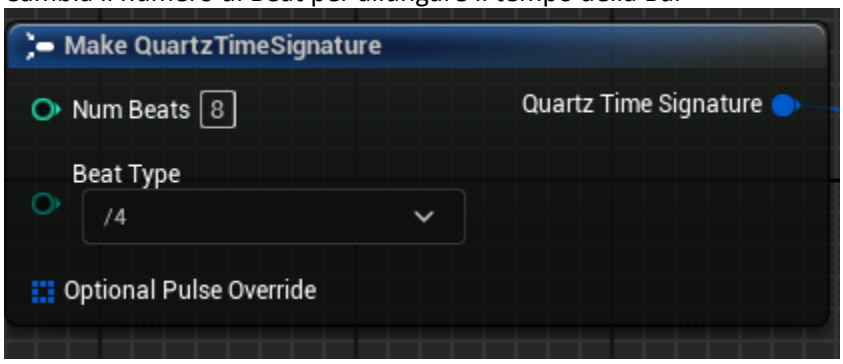
Setta i BPM, fai partire l'orologio, crea l'event dispatcher della quantizzazione



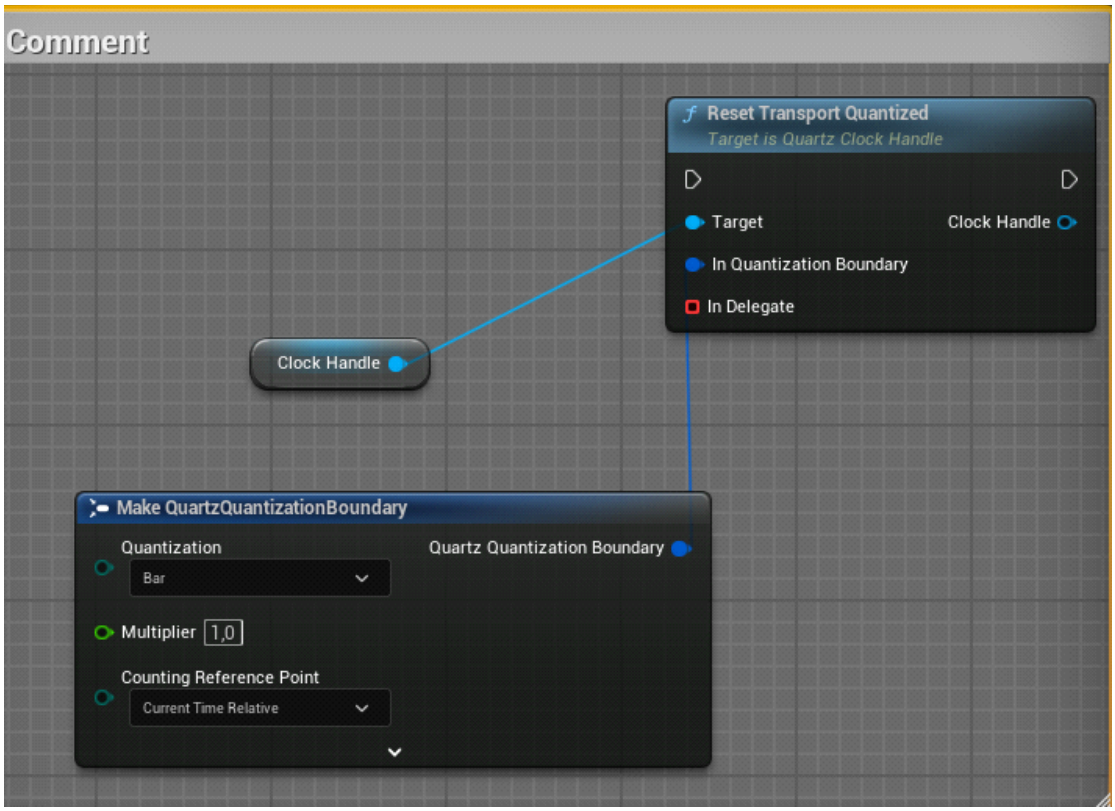
Switch on Quantization type, suona la traccia quantizzata



Cambia il numero di Beat per allungare il tempo della Bar



Puoi anche resettare la quantizzazione dell'orologio


Comment
Reset Transport Quantized
Target is Quartz Clock Handle
Target
Clock Handle
In Quantization Boundary
In Delegate
Clock Handle
Make QuartzQuantizationBoundary
Quantization
Bar
Quartz Quantization Boundary
Multiplier
1,0
Counting Reference Point
Current Time Relative