

Progetto Sound in Interaction

Di Salvatore Santangelo

Il progetto presentato consiste nella realizzazione di un ambiente 3D in **Unity** arricchito da una componente sonora procedurale sviluppata con **PlugData**, compilata attraverso **Heavy Compiler** e successivamente integrata in **Audiokinetic Wwise** come plug-in audio.

L'obiettivo principale è stato quello di creare un'esperienza interattiva e immersiva in cui il suono non fosse un semplice playback di campioni audio preregistrati, ma una generazione dinamica e adattiva di contenuti sonori basati sul comportamento del giocatore e sull'ambiente virtuale.

In particolare, sono stati sviluppati:

- Effetti procedurali legati al vento (attraverso ostacoli, tunnel, cavi e fogliame).
- Effetti sonori legati al movimento del personaggio (diverse tipologie di passi: pietrisco, neve, terra).
- Una spazializzazione realistica dei suoni in base alla posizione nel mondo 3D.
- Riverbero rendere più realistica la percezione degli spazi.
- Parametri dinamici per garantire una resa coerente e adattiva.

Strumenti utilizzati

Unity

Unity è stato utilizzato come motore di gioco 3D per la costruzione della scena. In particolare, è stata utilizzata la versione di Unity 6000.0.46.fl in quanto è la versione di Unity più recente che supporta l'integrazione con la versione di Wwise scelta per il progetto. In esso sono stati posizionati gli oggetti e implementati gli script per collegare le interazioni del giocatore con gli eventi audio gestiti da Wwise. Per la realizzazione della scena sono stati utilizzati gli asset gratuiti dello Unity Store, questi comprendevano il Player Controller e i GameObject che componevano la scena.

PlugData + Heavy Compiler

PlugData è stato lo strumento di creazione sonora, sfruttando il paradigma della programmazione visuale audio derivata da Pure Data. Le patch create sono state poi esportate tramite Heavy Compiler, uno strumento che consente la trasformazione delle patch in plug-in compatibili con Wwise. Questo approccio ha permesso di generare suoni sintetici procedurali, evitando l'uso di campioni statici ed evitando di utilizzare il sintetizzatore audio di Wwise, avendo più libertà nella creazione dei suoni procedurali.

Audiokinetic Wwise

Wwise ha svolto il ruolo di **middleware audio** per la gestione del sound design interattivo, in particolare:

- Importazione dei plug-in generati con Heavy Compiler.
- Creazione di eventi e containers per gestire le risorse sonore.
- Implementazione di attenuazione e spazializzazione 3D.
- Riverbero utilizzando gli Audio Bus e i Game State.
- Uso di Switch Container per la gestione contestuale delle tipologie di passi.

Il progetto mostra diverse potenzialità di Wwise come audio middleware. Wwise, infatti, permette di importare plug-in esterni, in questo caso quelli compilati attraverso Heavy Compiler, ampliando le possibilità creative oltre ad utilizzare dei campioni audio; inoltre, Wwise offre la possibilità di una spazializzazione avanzata attraverso parametri di posizione dell'oggetto nella scena e anche di attenuazione. Sfruttando gli Switch Container e i Game Parameter di Wwise è possibile inoltre rendere la scena interattiva, rendendo il comparto sonoro dinamico in base al comportamento del giocatore.

Anche FMOD offre la possibilità di integrazione con Unity e con plug-in esterni, ma nonostante sia percepito come un software più immediato nel suo utilizzo, offre minore dettaglio per quanto riguarda la componente di spazializzazione e interattività con il giocatore. Infatti, FMOD punta sulla semplicità con Eventi facilmente collegabili a parametri di gioco, mentre Wwise dispone di una gestione molto dettagliata degli Switch Containers, States e RTPC (Real-Time Parameter Controls) rendendolo quindi lo standard per giochi AAA.

Realizzazione del progetto

Suoni procedurali del vento

È stata sviluppata una serie di patch in PlugData finalizzate alla creazione del vento in diverse condizioni ambientali:

- Attraverso un ostacolo: simulazione delle turbolenze.
- Attraverso un cavo: generazione di un fischio sottile, tipico del vento che scorre su un oggetto teso.
- Dentro un tunnel: suono più cupo e risonante, con un senso di profondità.
- Attraverso foglie: rumore fruscianti, con componenti ad alta frequenza.

Per tutte e quattro le patch sono state utilizzate quelle mostrate a lezione, modificate eliminando la componente del panning, in quanto la spazializzazione audio si sarebbe poi implementata su Wwise.

Ciascun suono è stato realizzato come plug-in procedurale su Wwise dove a ciascun suono è stata aggiunta la componente di spazializzazione sulla rotazione e di attenuazione facendo in modo che il suono si facesse sempre più flebile allontanandosi dalla sorgente sonora ed è stato poi collegato in Unity tramite eventi Wwise.

Per ciascun suono è stato poi creato un Event_Play in Wwise che permette di riprodurre il suono nella scena su Unity.

Suoni dei passi

Sono stati realizzati tre plug-in procedurali distinti per simulare il suono di passi su:

- Pietrisco.
- Terra/sporco.
- Neve.

Anche in questo caso per realizzare il plug-in di queste tre tipologie di suoni il punto di partenza è stata la patch di PlugData mostrata a lezione. In questo caso si è deciso di separare la superficie su cui è riprodotto il suono in tre patch diverse. Inoltre, il parametro walkspeed che controlla la velocità della camminata, precedentemente rappresentato da uno slider con valori compresi tra 0 e 1, è stato sostituito dal [r walkSpeed @hv_param 0 1]. Questo oggetto di PlugData permette a HeavyCompiler di utilizzare il parametro walkSpeed dentro Wwise come parametro RTCP controllabile. In questo modo è stato possibile collegare questo parametro ad un GameParameter all'interno della scena di Unity e fare in modo che modificasse la velocità della riproduzione del suono della camminata in base alla velocità di movimento del giocatore nella scena 3D.

In Wwise i tre suoni sono stati organizzati in uno Switch Container, e poi creati tre Switch ognuno collegato ad un SFX, controllato in Unity dalla logica di superficie calpestata dal personaggio. Questo ha permesso al sistema di scegliere dinamicamente il tipo di passo da riprodurre in base al contesto ambientale, migliorando notevolmente il realismo dell'esperienza sonora. Infine, è stato creato due eventi dallo Switch Container: un evento di play e un evento di stop.

Integrazione, spazializzazione e riverbero

In Unity, dopo aver importato i plug-in tra gli asset, gli eventi Wwise sono stati richiamati tramite componenti dedicati e associati agli oggetti della scena. Nella scena è stato creato un GameObject vuoto e sono stati aggiunti i componenti AkAmbient che richiamano gli eventi per ogni suono del vento quando viene chiamato il metodo Start. Nell'AkAmbient è stato selezionato come Position Type la Large_mode. In questo modo è possibile inserire la lista di tutte le posizioni degli oggetti in cui è presente la sorgente sonora per ottenere così il suono spazializzato nella scena.

Per la riproduzione dei passi nella scena è stato modificato lo script del PlayerController, nel metodo Move() che gestisce il movimento del giocatore. Quando il giocatore si muove toccando e il personaggio è a terra si riproduce il suono dei passi, cambiando il tipo di suono in base al tipo di terreno calpestato, facendo un Raycast verso il basso e cambiando lo switch in base al tag del collider individuato. Inoltre, in base alla velocità del movimento del personaggio, cambia anche il valore di walkSpeed che modifica la velocità di riproduzione del suono dei passi. Quando invece il personaggio salta o non tocca terra viene chiamato l'evento che ferma la riproduzione del suono rendendo più realistica la scena.

Per arricchire ulteriormente l'esperienza sonora e rendere più realistica la percezione degli spazi, è stato introdotto un sistema di riverbero dinamico sfruttando gli **Audio Bus** di Wwise in

combinazione con i **Game States**. Gli Audio Bus in Wwise sono elementi che servono a instradare, raggruppare e processare i segnali audio.

Ogni evento o suono in Wwise viene inviato a un bus, che può avere effetti applicati, controlli di volume, routing verso altri bus o verso l'uscita principale. Esistono principalmente due tipologie:

- **Audio Bus principali** → organizzano i segnali in categorie, permettendo di controllare separatamente i volumi o applicare effetti comuni a un gruppo di suoni.
- **Auxiliary Bus (Aux Bus)** → pensati per gestire effetti come riverbero o delay. Un evento sonoro può inviare parte del proprio segnale a un Aux Bus (send), mantenendo il segnale diretto pulito ma aggiungendo l'effetto in parallelo.

I Game States sono uno strumento di Wwise che consente di modellare condizioni di gioco e collegarle a modifiche dinamiche del comportamento audio.

Un Game State rappresenta uno "stato globale" del gioco, definito dal programmatore o dal sound designer, che può influenzare:

- Il routing dei segnali verso bus o effetti.
- I parametri di riproduzione degli eventi sonori.
- Il mix complessivo.

I Game States vengono gestiti in Unity (o nell'engine di gioco) tramite chiamate alle API di Wwise, e permettono di sincronizzare l'audio con ciò che accade nel gameplay in modo coerente e immersivo.

In Wwise è stato creato un Audio Bus dedicato al riverbero, all'interno del quale sono stati creati tre Aux Bus: Room1, Room2, Tunnel. A ciascuno dei tre Aux Bus è stato assegnato un diverso effetto di riverbero nella categoria "Effects". Questi effetti di riverbero sono presenti come plug-in in Wwise e consentono di utilizzare diverse configurazioni di riverbero, i cui parametri sono anche modificabili. In questo caso i tre riverberi utilizzati sono: Light_Bright_chamber per Room1, Medium_Room per Room2 e Tunnel1 per Tunnel. Successivamente è stato creato un Audio Bus Footstep assegnato alla sorgente sonora dei suoni passi. All'interno dell'Audio Bus Footstep sono stati selezionati i tre Aux Bus del riverbero per influenzare il suono dei passi.

Questa configurazione avrebbe attivato l'effetto di riverbero ogni volta che il suono dei passi fosse stato riprodotto, per attivare il riverbero solo negli ambienti chiusi sono stati usati i Game States.

Sono stati definiti, tre State Group, uno per ogni tipologia di riverbero. All'interno degli Stati Group sono stati definiti gli stati di On e Off. Ogni Group State è stato quindi assegnato ad un Aux Bus definendo che, quando lo stato fosse stato in Off il riverbero non sarebbe stato attivo e si sarebbe attivato solo con lo stato in On.

Per integrare il riverbero in Unity, è stato aggiunto un Box Collider utilizzato come trigger alle due stanze e al tunnel presenti nella scena; quindi, sono state aggiunte ad ogni oggetto due componenti AkState, in cui è stato definito quale sarebbe stato l'effetto dello Stato, cioè On a Trigger Enter e Off a Trigger Exit. Infine, sono state aggiunte le componenti di AkState relative ad ogni Stato al WwiseGlobal object, in modo da inizializzare di default lo stato di tutti i riverberi ad Off.

Conclusioni

Il progetto ha dimostrato come sia possibile unire creazione sonora procedurale (PlugData + Heavy), gestione interattiva del suono (Wwise) e rendering 3D (Unity) in un unico workflow coerente e professionale.

L'uso di suoni procedurali al posto di campioni statici offre vantaggi significativi:

- Maggiore variabilità e realismo.
- Maggiore interazione con i parametri di gioco (posizione, superfici, ambienti).

In prospettiva, questo approccio può essere esteso ad altri aspetti sonori: generazione dinamica di musica, effetti ambientali complessi, o interazioni avanzate con la fisica del gioco.

In conclusione, il progetto non solo rappresenta un esercizio tecnico, ma mette in luce le enormi potenzialità di Wwise come piattaforma per l'audio interattivo, soprattutto se combinato con strumenti di sintesi e sound design procedurale.