

TUTORIAL XPRESSO E SUONO

Questo è un piccolo tutorial che spiega come usare il nodo "SOUND" di XPRESSO come deformatore. prima cosa apriamo un file nuovo di Cinema 4d.

1500 frame e a 25 frame al secondo(edit->Project settings o ctrl-d pc, mela-d mac).

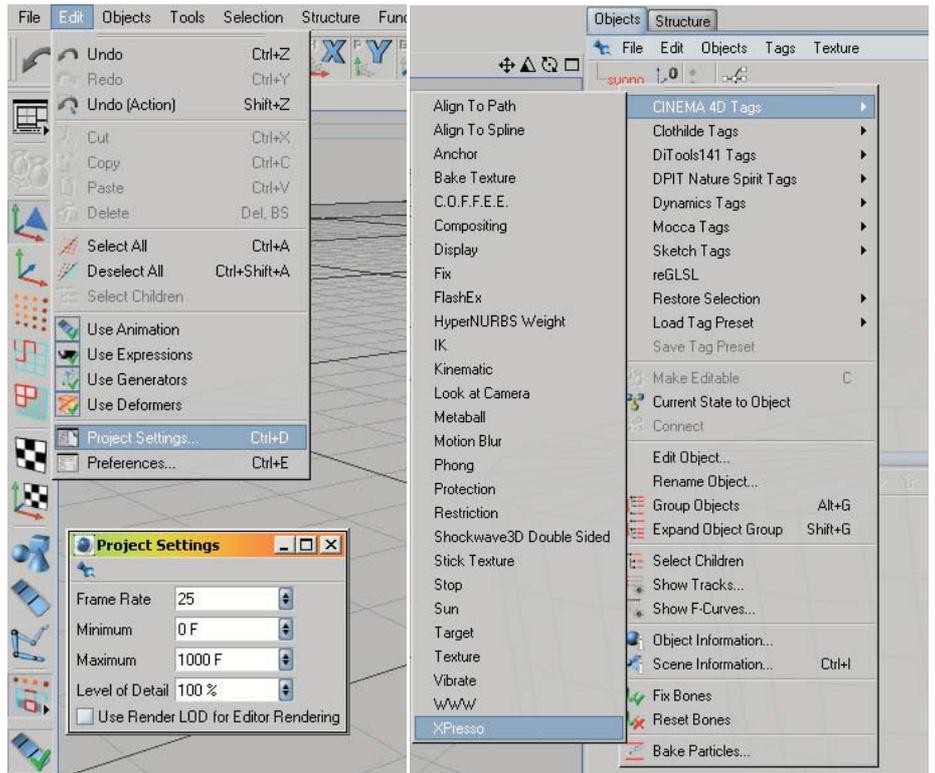
Aggiungiamo un oggetto nullo Object/Null Object.

nell'Object manager (in alto a destra) e lo rinominiamo (doppio click) come Suono.

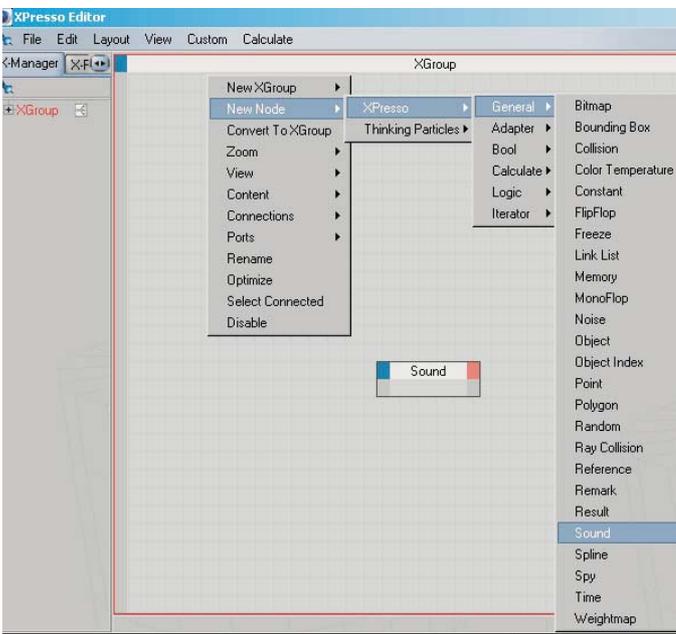
A questo punto tasto destro nell'Object manager sull'oggetto Suono e selezioniamo "Cinema 4d Tags->Xpresso".

Apparirà un simbolino tipo diagramma a fianco dell'oggetto Suono. click sul simbolino per aprire Xpresso editor.

Una volta aperto l'editor per aggiungere i nodi basta premere tasto destro nella parte grigio chiaro quadrettata e ci apparirà la lista dei nodi possibili.



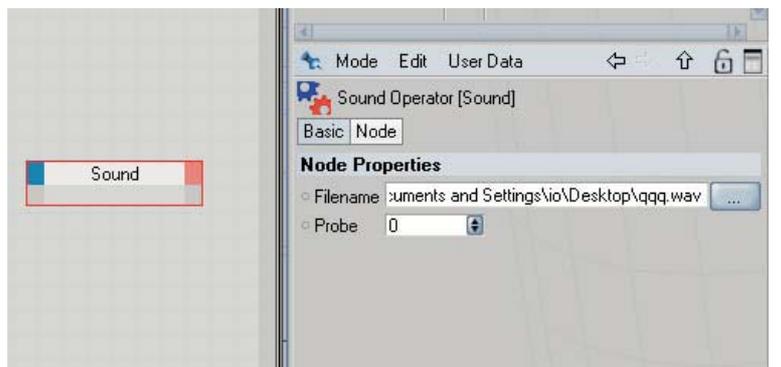
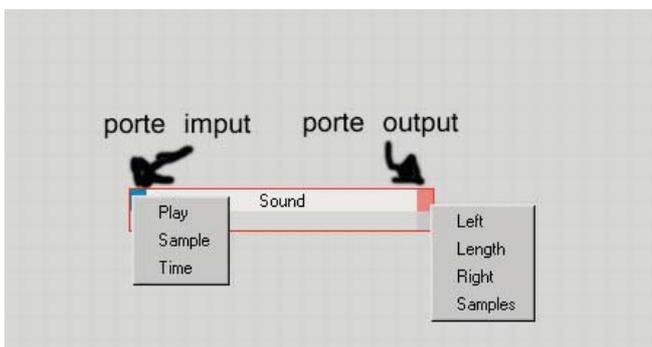
Aggiungiamo un nodo New Node->Xpresso->General->Sound. Verra aggiunto un rettangolino chiamato Sound con due quadratini uno blu in alto a sinistra e uno rosso in altro a destra.



Premendo con il mouse sul quadrato blu aggiungiamo delle porte di tipo input (delle porte che ricevono i dati e i valori da un'altro nodo o dei valori predefiniti impostati da noi).

Premendo con il mouse sul quadrato rosso aggiungiamo delle porte di tipo output (delle porte che ridanno i dati e i valori del nostro nodo per restituirli ad altro) Ridimensioniamo Xpresso Editor in modo da poter avere visibile l'Attribute Manager nell'interfaccia di base di Cinema 4d (in basso a destra).

Selezioniamo il nodo Sound e nell'Attribute manager In Node Properties c'è un attributo che si chiama File Name, premiamo sui tre puntini e svogliamo nel nostro computer per aggiungere il nostro file audio. Se non sbaglio Cinema 4d legge solo file audio non compressi (.wav e .aiff). Quindi se avete un mp3 dovete convertirlo prima convertirlo in Wav o Aiff. lo personalmente uso il pc e mi trovo bene con Gold Wave chè un programmino che si può scaricare da internet come evulation copy tipo Winzip e lo trovate qui <http://www.goldwave.com/> (il settaggio migliore per salvare i wav è quello che lui ha di default: PCM signed



16bit stereo) e se siete dei Mac user sono sicuro che QuickTim e iTunes possono convertire in Aiff altrimenti guardate qui <http://www.pure-mac.com/audio.htm>.

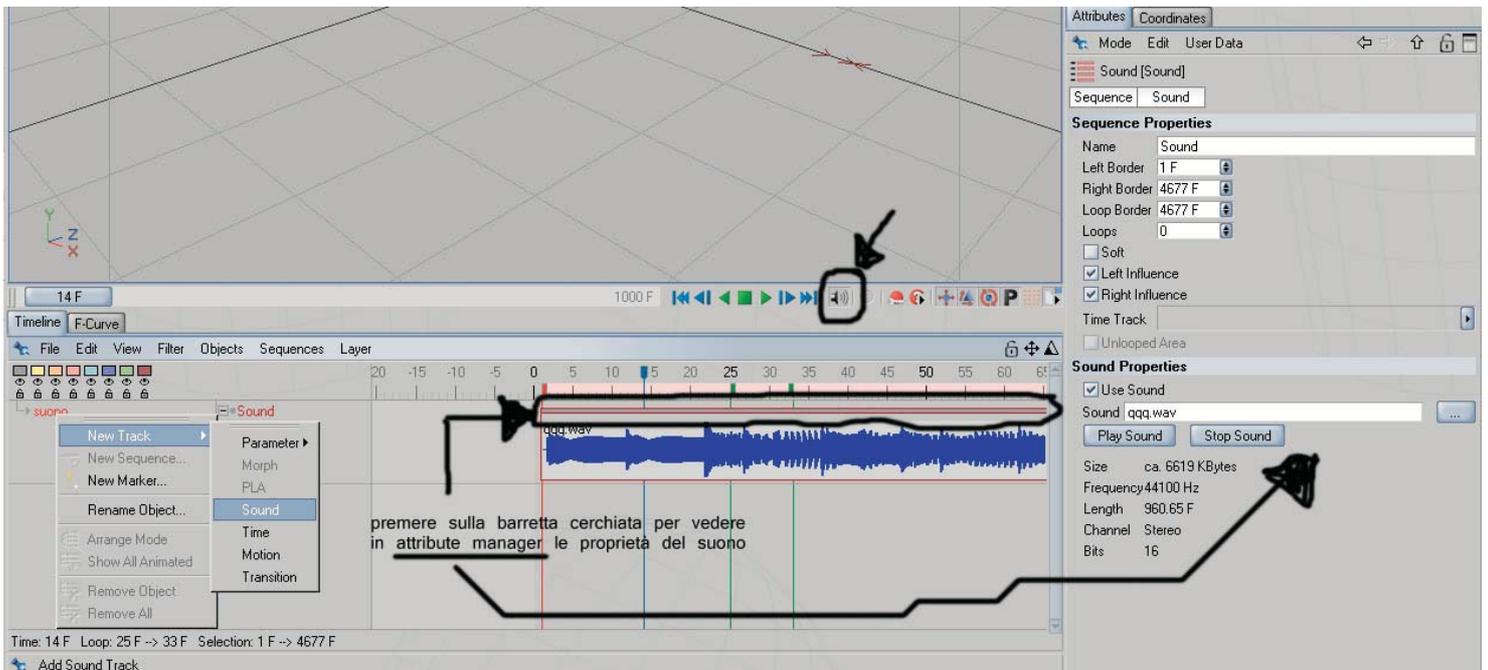
Cambiamo la interfaccia momentaneamente in modalità Animation. In alto a sinistra c'è un tasto sotto la freccia di annulla dove si può selezionare il tipo di interfaccia.

In basso troviamo la Timeline con dentro il nostro oggetto nullo Suono. Tasto destro in timeline sull'oggetto suono, New Track->Sound.

Apparirà un oggetto chiamato Sound e una linea grigia sotto il numero dei frame. Selezioniamo la linea grigia.

Andiamo nell'Attribute Manager ed In Sound Properties sfogliamo e riaggiungiamo il file audio aggiunto prima.

Andiamo sul Playback Controller e premiamo sul simbolino del suono per metterlo Enable. Aggiungiamo il suono perchè vogliamo anche sentirlo per vedere come la nostra animazione reagisce. Riapriamo Xpresso Editor.



Vediamo un attimo le porte del nodo Sound.

Porte tipo Input:

Play: Porta di tipo booleano (0 o 1) se riceve 0 = disable se riceve 1 = enable. Si può usare per esempio per suonare il file quando detetta una collisione.

Sample: Mi restituisce il numero di Samples di un suono. Le samples non sono altro che i frame del suono (un video Pal ha 25 frame per secondo e Ntsc 30 fps, il suono ne ha un pò di più, normalmente varia tra le 41.000 e le 48.000 samples per secondo dipende dalla qualità e compressione). Questa porta input è piuttosto inutile, non fa altro restituirmi il numero di samples per la lunghezza del suono es. suono di 20 sec con 44.000 samples a secondo restituisce $20 \times 44.000 = 880.000$. Volendo si può variare la qualità del suono a seconda se succede qualcosa. Non ho ben capito se riducendo le samples ne riduce la qualità o la lunghezza.

Time: E' il punto il del playback del suono. Ad esempio voglio sapere che volume ha il canale sinistro a 2 secondi dall'inizio gli imposto come valore 2. Se voglio farlo suonare al doppio della velocità posso collegagli come input un nodo Time con output time moltiplicato x 2 o posso fare cose del tipo screcciare il suono sul valore di un deformatore di un'altro oggetto.

Porte di tipo Output:

Left: Il volume del suono del canale sinistro. Qui ho fatto un pò di prove ed ho visto che ha sia valori positivi che negativi. Ci sono due tipo di sound analisi una basata sullo spettroscopio (spetroscope) e una sull'oscilloscopio (waveform). Lo spettroscopio solitamente ha solo valori positivi (normalmente da 0 a 1) e analizza i volumi delle bande (alti, bassi, medi dipende dalla larghezza della banda e posizione); è la classica analisi che si vede anche sui vecchi stereo che va dai bassi a sinistra agli altri a destra e che si può equalizzare. L'oscilloscopio invece ha sia dei valori positivi che negativi. Quando il suono non c'è è a 0 e poi vibra su e giù (con più alta frequenza di oscillazione e ziglinatura se ci sono suoni tipo noise con + bassa frequenza se ci sono suoni lughì e uniformi tipo bassi booooooooooooouuuuuuuaaaaooo) e con ampiezza

che invece dipende dal volume. A quanto pare quello che analizza lui è una specie di media di tutta la banda dell'oscilloscopio.

Length: Lunghezza del suono in secondi.

Right: Medesimo di left solo che analizza il canale destro.

Samples: Come samples dell'input.

Nel nodo suono c'è anche la voce Probe che a quanto dice il manuale dovrebbe essere il numero di Samples analizzate. In pratica se ho un suono che è a 44.100 samples per secondo e lavoriamo su file Pal a 25 frame al secondo in ogni frame ci sono $44.100/25 = 1764$ samples per frame. Il manuale dice che alzando il numero di Probe il valore del volume del suono si abbassa, non ho ben capito perchè, forse perchè facendo una media tra tante samples il suono diventa + uniforme. Ho provato a cambiare i diversi valori di probe ma ho visto che il funziona meglio lasciandolo a zero.

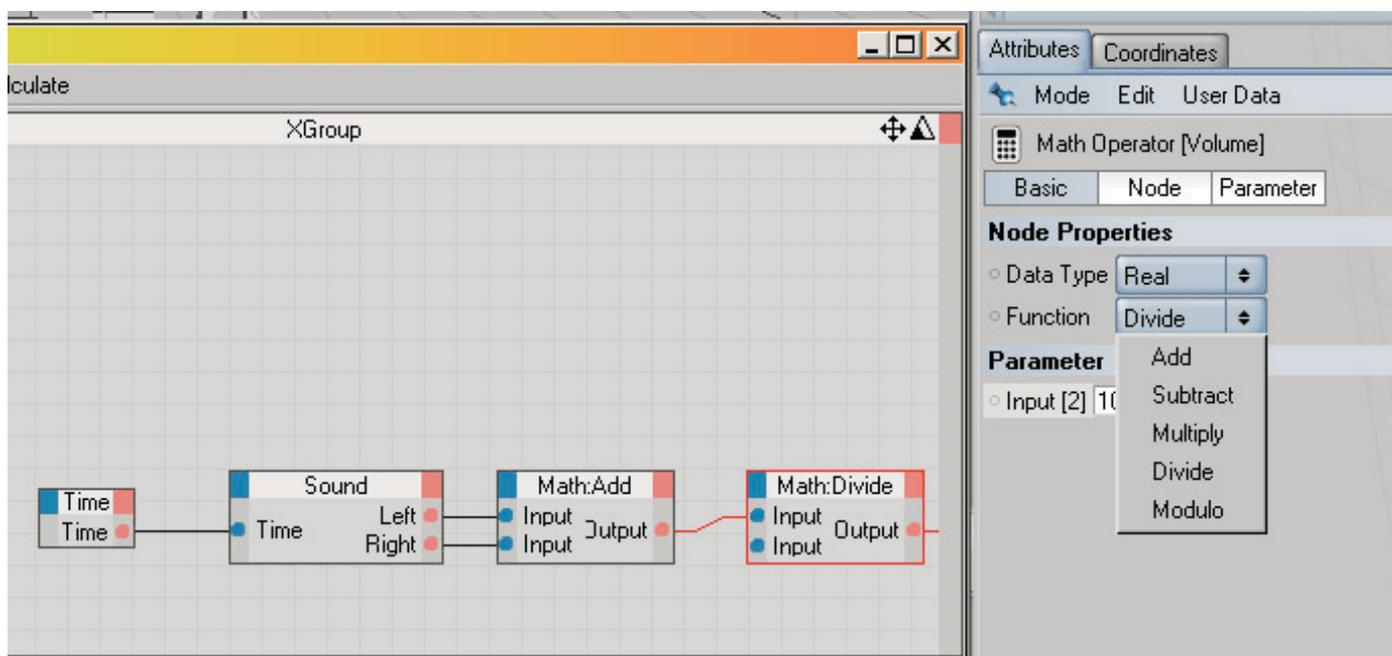
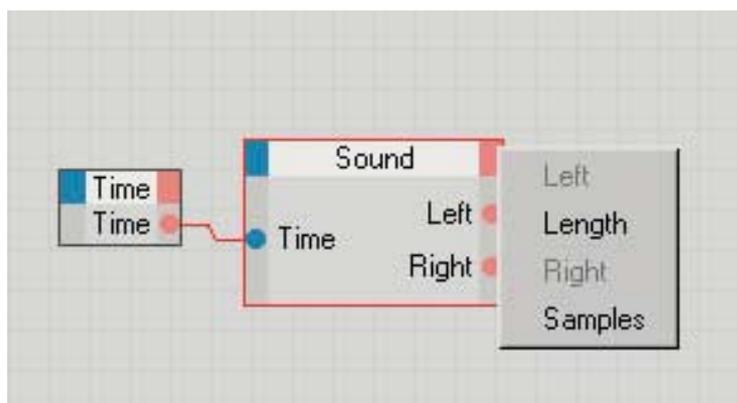
Ora che c'è un pò più chiaro il nodo torniamo nell'editore Xpresso pigiando con il mouse sul quadratino blu del nostro nodo Sound aggiungiamo una porta input "Time" e sul quadratino rosso degli Output aggiungiamo due porte una Right e una Left.

Premendo con il tasto destro fuori dal nodo nella parte grigio chiara quadrettata aggiungiamo un nodo New Node->Xpresso->General->Time.

Il nodo Time ha già di default come output Time. Colleghiamo l'output Time del nodo Time alla porta Input Time del nodo Sound. Facciamo questo perchè l'output Time di Time ci dà il tempo corrente in sec dell'animazione e noi vogliamo che se l'animazione è a 2 secondi analizzi il suono quando anche esso è a 2 secondi se l'animazione è a 3 sec analizzi il suono a 3 sec. Se vogliamo far partire il nostro suono dopo 2 secondi (50 frame in Pal) x esempio basta mettere in mezzo un nodo math:subtract con input2=2. In modo che quando Time dell'animazione è a 2 secondi il time del suono è a $2-2$ quindi a zero e inizia a partire da qui.

Ci serve fare una media tra i due canali destro e sinistro. Aggiungiamo un nodo New Node->Xpresso->Calculate->Math. Lo selezioniamo e nell'attribute Manager lo lasciamo su Function Add, inseriamo i due Output "Left" e "Right" del nodo sound rispettivamente a Input1 e Input2 in modo di avere la somma dei volumi dei 2 canali. Se colleghiamo un nodo Result al risultato della somma dei due canali e facciamo scorrere la timeline vediamo che abbiamo un risultato che va quasi sempre oltre le migliaia dipende dal volume del file wav o aiff inserito. Le variazioni che ci interessano sono sull'ordine delle centinaia quindi per evitare di scrivere sempre i due 00 delle unità e delle decine ci conviene dividerlo per 1000 o per 10000 a seconda di come ci troviamo meglio.

Aggiungiamo un nuovo nodo xpresso/calculate/Math e lo impostiamo su Divide invece che Add come Input1 colleghiamo il risultato della addizione (l'output del nodo Mat:Add) e come input2 io ho impostato 1000.



Sappiamo che nell'oscilloscopio un volume -1000 corrisponde a un volume 1000 la posizione positiva o negativa dipende solo dall'uniformità e variazione dei tipi suoni (suoni tipo noise vibra con più frequenza su e giù ma il volume è dato dalla distanza da zero) ed è quindi abbiamo bisogno per tracciare il reale volume che anche i valori negativi ritornino positivo.

Esiste una funzione apposta che fa questo (absolute + spesso conosciuta in sigla come abs(valuta)). Aggiungiamo un nodo di tipo Xpresso/Calculate/Absolute e gli inseriamo come input il risultato del nodo Math/Divide. L'output del nostro nodo Absolute darà il nostro volume base.

BEAT DETECTION

Il beat è praticamente quando si sente una percussione. Normalmente in una traccia audio abbiamo una melodia di base e le percussioni (drums) che danno il tempo. La melodia è solitamente stabile ed ogni tot tempo si aggiungono le percussioni quindi quando c'è la percussione il volume di questa si somma al volume della melodia generando così un picco di volume + alto.

Quindi per dettare il beat in modo un pò artigianale basta vedere quando il nostro volume base è più alto del normale. Aggiungiamo un nuovo nodo New Node->Xpresso->Logic->Compare.

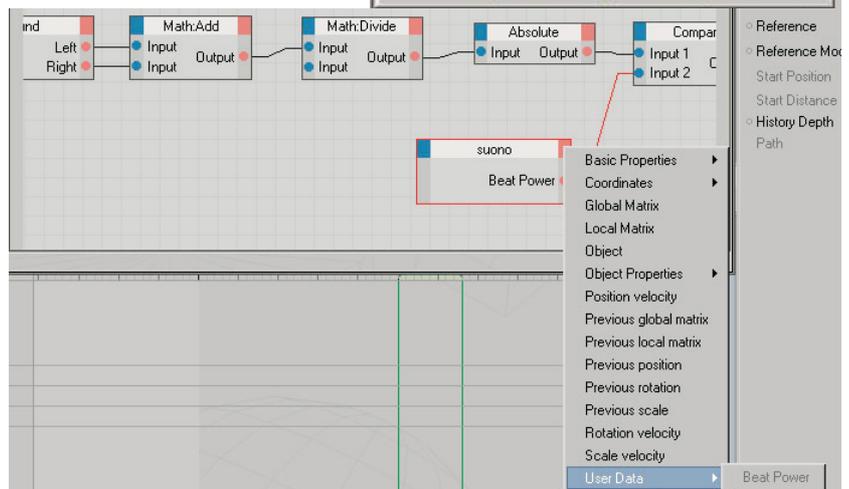
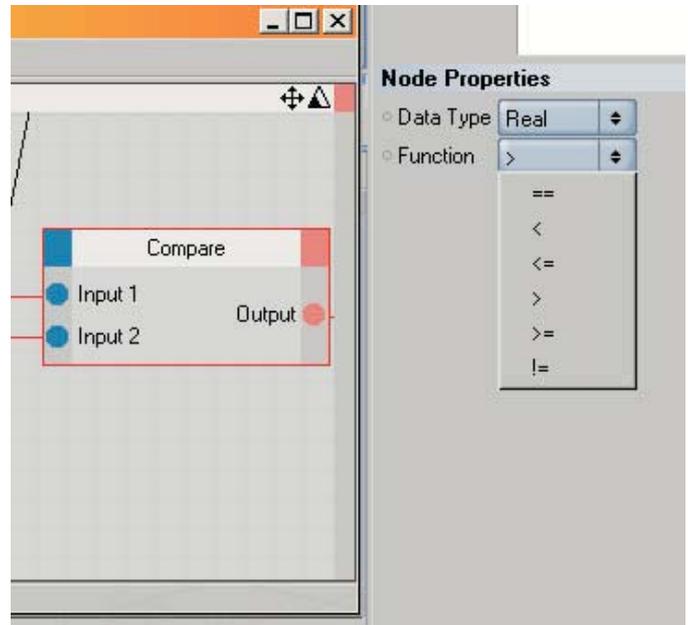
Colleghiamo al nodo Compare come Input1 il risultato l'output del nodo Absolute ovvero il nostro volume base.

Selezioniamo il nodo compare e settiamolo nell'Attribute manager su > (maggiore) invece di ==, verifichiamo quando in nostro volume cioè il nodo Input1 del nodo Compare supera un certo volume quindi quando c'è il beat.

Dato che il volume cambia da file audio a file audio e per non dover aprire sempre l'editor di Xpresso x settare a che volume si verifica il beat selezioniamo il nostro oggetto nullo Suono nell'Object Manager e dal menu dell'attribute manager selezioniamo User Data/add User Data. Come Name: mettiamo Beat Power, Data Type Float, Interface: Float, Unit: Real, min:0, max:100; step: 0.1.

Riapriamo Xpresso Editor gli trasciniamo dentro il nostro oggetto Suono. Premiamo sul quadratino rosso degli Output e aggiungiamo una porta UserData->Beat Power. Colleghiamo poi questa porta all'Input2 del nodo Compare in modo da verificare quando l'Input1 (il volume) è maggiore di Input2 (beat power). Output del nostro nodo Compare ci da come risultato 1 quando l'Input1 è maggiore uguale all'Input2 se no 0 (1 quando c'è beat, 0 quando non c'è).

prova, ho attaccato un nodo result al nodo math:divide per vedere i risultati del volume ed ho messo un cubo nell'object manager con la porta enable come input e gli ho collegato la porta output del nodo compare (enable al beat, disable quando non c'è beat) ed ho messo play un paio di volte. Cambiando un pò di valori dell'User Data BeatPower, ho visto che nel mio file audio si verifica + o - il beat quando il Volume è sopra 10 ed ho impostato in beat power 10 (ricordati che comunque beat power è comunque animabile con key frame quindi se hai un file audio lungo che cambia puoi settare dei valori diversi a seconda della posizione).



.La nostra analisi del beat però resta ancora molto artigianale. Nel senso che non è precisissima e se c'è rumore o delay dopo il beat vediamo sfarfallare in maniera fastidiosa il nostro cubo per un frame sì e un frame no. Per evitare questo potremo dire al computer di contare almeno 3 frame dopo un beat prima di verificarne un altro beat. Per fare questo ho usato un nodo Coffee. Non sto a spiegarvi le basi della programmazione e del linguaggio coffee xchè sarebbe cosa troppo lunga (se non capite leggetevi qualcosa a riguardo, per il linguaggio coffee ci sono gli sdk sul sito della maxon <http://www.maxon.net> nella sezione plug-in cafe e un pò di materiale sul nodo coffee lo trovate qui <http://www.peranders.com/tutorials/tutorials.php?index=3>).

Il codice del nostro nodo coffee è:

//definisco due variabili.

```
var contatore;  
var contauno;  
main()  
{
```

//Se le 2 variabili non hanno valore

//impostale uguali a zero

```
if (!contatore) contatore=0;  
if (!contauno) contauno=0;
```

// Input1 è l'output del nostro nodo

// compare (1=beat, 0=no beat)

// Se c'è beat riporta contauno a 0

// altrimenti ogni frame incrementa la

// variabile contauno di 1.

// esempio quando c'è beat contauno=0, il frame dopo il beat è = 1; 2 frame dopo è

// = 2; 3 frame dopo è uguale 3 e così via. Quando c'è beat torna a zero

// e ricomincia a contare.

```
if (Input1==1) contauno=0;  
else contauno = contauno+1;
```

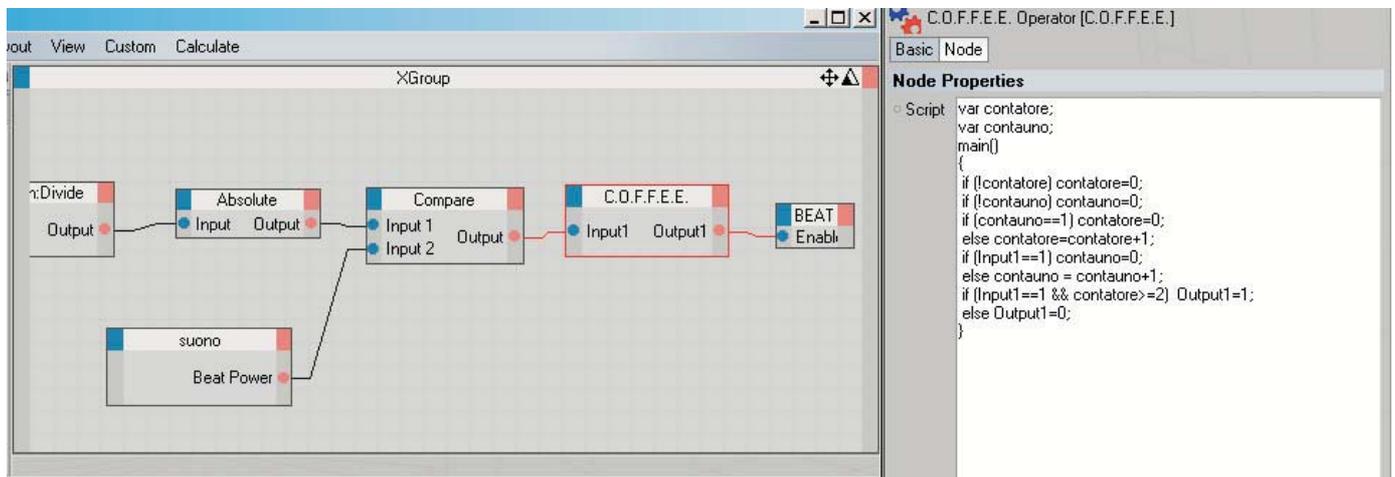
//Uguale a prima solo che torna a zero, quando contauno è uguale a 1 ovvero un frame dopo il beat.

```
if (contauno==1) contatore=0;  
else contatore=contatore+1;
```

// se c'è beat e contatore ha almeno contato fino a 2 detetta il nuovo beat se no non dettare il beat.

```
if (Input1==1 && contatore>=2) Output1=1;  
else Output1=0;
```

```
}
```



Se vedete ho dovuto usare 2 variabili perchè se facevo tornare a zero contatore direttamente quando c'è il beat quando Input1 era uguale a 1 contatore era sempre = 0 e quindi questa condizione (Input1==1 && contatore>=2) era sempre falsa. Usando le due variabili contatore torna a zero il frame dopo il beat in modo da poter verificare la condizione. Il frame dopo il beat contatore è uguale a 0; 2 frame dopo a 1; 3 frame dopo a 2. Quindi conta almeno 3 frame prima che ci possa essere un altro beat evitando così sfarfallamenti un frame sì e uno no.

Ora il nostro motore di analisi sonora è pronto (Output di Absolute = Volume - L'output del nodo coffee = 1 al beat se no = 0).

Ho preparato un file dove ho messo un testo estruso che compare al beat (ho collegato l'output del nodo coffee ad enable), un altro testo che mi da il volume in valore numerico (collegato l'output di absolute al parametro testo) e una barretta che scala sul volume (ho collegato absolute a scale), Vi preparerò esempi più pratici in un altro tutorial.