



INFORMATICA MUSICALE

UNIVERSITA' DEGLI STUDI DI CATANIA
DIPARTIMENTO DI MATEMATICA E INFORMATICA
LAUREA TRIENNALE IN INFORMATICA
A.A. 2019/20
Prof. Filippo L.M. Milotta

ID PROGETTO: 01

TITOLO PROGETTO: Processing Band

AUTORE 1: Barbagallo Simone

AUTORE 2: Basile Simone

AUTORE 3: Marino Francesco Pio

Sommario

1. Obiettivi del progetto	2
Introduzione	2
Il linguaggio di programmazione Processing	2
La libreria Sound	2
WaveForm	2
SoundFile	2
Noise	3
Effects	3
Problemi riscontrati in fase di sviluppo e soluzioni applicate	3
2. Riferimenti Bibliografici	4
Approfondimenti sulla libreria Sound e le sue classi	4
Approfondimenti riguardanti gli argomenti teorici trattati	4
3. Argomenti Teorici Trattati	5
Principio di sovrapposizione delle onde	5
Filtri audio	6
Rumori colorati	6

1. Obiettivi del progetto

Introduzione

Il progetto da noi realizzato si pone l'obiettivo di analizzare il grafico delle onde prodotte da vari strumenti musicali e di studiare il comportamento dell'onda risultante dal sovrapporsi di più tracce audio, il tutto con la possibilità di effettuare un'analisi con l'aggiunta volontaria di vari tipi di rumori e/o filtri che sono stati trattati a lezione.

Per la realizzazione del software ci siamo appoggiati sul linguaggio di programmazione Processing, linguaggio che abbiamo già utilizzato precedentemente in Interazione e Multimedia dal lato image processing e che pertanto abbiamo voluto approfondire anche dal lato audio processing.



Il linguaggio di programmazione Processing

Processing è un linguaggio di programmazione che eredita tutta la sintassi, i comandi e il paradigma della OOP dal linguaggio di programmazione *Java*. Consente lo sviluppo di un'ampia gamma di applicazioni tra cui giochi, animazioni e, come nel nostro caso, contenuti interattivi.

Per lo sviluppo di un'applicazione in processing bisogna tener conto di due metodi chiave: *setup()* e *draw()*. Il primo metodo serve per la dichiarazione e l'inizializzazione delle variabili, nonostante ciò è possibile anche dichiarare globalmente le variabili; il secondo metodo serve per tutte quelle operazioni

che devono essere ripetute ogni frame.

Le motivazioni che ci hanno spinto ad utilizzare in particolar modo questo linguaggio sono state due: la presenza di librerie native e di terze parti che ci hanno facilitato lo sviluppo del progetto e la semplicità di utilizzo per la realizzazione di elementi grafici.

La libreria Sound

La libreria su cui ci siamo appoggiati è stata la libreria nativa Sound, in quanto ci ha agevolato notevolmente il lavoro in fase di caricamento delle tracce audio e nella fase di audio processing, in particolar modo per l'introduzione volontaria dei rumori e per l'applicazione dei filtri. Inoltre, in Sound sono presenti una serie di classi che sono state molto importanti per il raggiungimento del nostro obiettivo.

Analizziamo dunque le classi con i relativi metodi che sono stati utili nello sviluppo del progetto.

WaveForm

Questa classe ci ha permesso di poter ricavare le informazioni relative al grafico dell'onda generato da ogni singolo strumento musicale da noi preso in analisi. Per l'utilizzo di questa classe, il parametro fondamentale richiesto dal costruttore consiste in un intero numerico che simboleggia il numero di campioni da analizzare. A tal fine è stato necessario analizzare ogni qual volta un piccolo numero di campioni per poter ricostruire il più fedelmente possibile la forma d'onda.

SoundFile

È stata la classe che ci ha permesso di poter caricare ogni singolo file audio. Possiede nativamente dei metodi che ci hanno aiutato nella riproduzione delle tracce, in particolar modo:

- *play()* – per iniziare la riproduzione della traccia
- *stop()* – per concludere la riproduzione della traccia

- *isPlaying()* – per controllare se una traccia fosse in riproduzione
- *loop()* – per riprodurre in loop una singola traccia

Noise

È stata utile per l'inserimento volontario di alcuni rumori che sono stati trattati a lezione. In particolare modo le classi native riguardanti i rumori da noi utilizzati sono stati WhiteNoise, BrownNoise e PinkNoise. Per facilitare lo sviluppo si è scelto di implementare una classe Noise contenente un oggetto rappresentante ciascuno dei rumori precedentemente elencati.

Effects

È una raccolta di classi per l'introduzione di alcuni filtri da noi trattati a lezione specialmente Low-Pass, Band-Pass e High-Pass. Tutte e tre le classi elencate possiedono nativamente due metodi che ci sono stati utili nello sviluppo:

- *process()* – per iniziare l'applicazione del filtro
- *stop()* – per concludere l'applicazione del filtro

Problemi riscontrati in fase di sviluppo e soluzioni applicate

Abbiamo notato che Processing non supporta nativamente il formato GIF, pertanto non è stato possibile inserire direttamente ogni singolo strumento musicale con la rispettiva animazione semplicemente con il metodo *image()*, ma è stato necessario estrarre ogni singolo frame, utilizzare un array di immagini e ogni qualvolta utilizzare il modulo del frame corrente per stampare correttamente l'immagine a video.

2. Riferimenti Bibliografici

Nella scrittura della relazione, ci siamo serviti di una serie di fonti, tra le quali:

Approfondimenti sulla libreria Sound e le sue classi

[IT] Processing – <https://it.wikipedia.org/wiki/Processing>

[EN] *Processing: playing and using sound files* – <https://www.youtube.com/watch?v=DJJCci3kXe0>

[EN] *Panoramica sulle classi di Sound* – <https://processing.org/reference/libraries/sound/>

[EN] *Classe di Sound: SoundFile* – <https://processing.org/reference/libraries/sound/SoundFile.html>

[EN] *Classe di Sound: LowPass* – <https://processing.org/reference/libraries/sound/LowPass.html>

[EN] *Classe di Sound: HighPass* – <https://processing.org/reference/libraries/sound/HighPass.html>

[EN] *Classe di Sound: BandPass* – <https://processing.org/reference/libraries/sound/BandPass.html>

[EN] *Classe di Sound: WhiteNoise* – <https://processing.org/reference/libraries/sound/WhiteNoise.html>

[EN] *Classe di Sound: PinkNoise* – <https://processing.org/reference/libraries/sound/PinkNoise.html>

[EN] *Classe di Sound: BrownNoise* – <https://processing.org/reference/libraries/sound/BrownNoise.html>

Approfondimenti riguardanti gli argomenti teorici trattati

[IT] *Diversi tipi di rumore* – [http://fisicaondemusica.unimore.it/Diversi tipi di rumore.html](http://fisicaondemusica.unimore.it/Diversi_tipi_di_rumore.html)

[IT] *Equalizzatori e filtri* – [http://www.audiosonica.com/it/corso/post/77/Equalizzatori e Filtri-Filtri](http://www.audiosonica.com/it/corso/post/77/Equalizzatori_e_Filtri-Filtri)

[IT] *Filtri audio* – <https://blog.musilosophy.it/filtri-audio/>

[IT] *Principio di sovrapposizione delle onde* – <https://www.chimica-online.it/fisica/principio-di-sovrapposizione.htm>

[IT] *Fase (segnali)* – [https://it.wikipedia.org/wiki/Fase_\(segnali\)](https://it.wikipedia.org/wiki/Fase_(segnali))

[IT] *Principio di sovrapposizione* –

[http://moby.mib.infn.it/~labdida/lib/exe/fetch.php?id=esperimentazioni di fisica&cache=cache&media=lezioneeonde.pdf](http://moby.mib.infn.it/~labdida/lib/exe/fetch.php?id=esperimentazioni_di_fisica&cache=cache&media=lezioneeonde.pdf)

3. Argomenti Teorici Trattati

Il progetto da noi realizzato fa riferimento ad alcuni argomenti teorici trattati a lezione che abbiamo opportunamente approfondito per uno sviluppo del progetto più completo. In particolar modo gli argomenti teorici principali che abbiamo analizzato sono stati: il principio di sovrapposizione delle onde, i filtri audio e i vari tipi di rumore colorato.

Principio di sovrapposizione delle onde

Enunciato del principio di sovrapposizione: *“se due o più onde della stessa natura che si propagano nello stesso mezzo si sovrappongono in un certo punto dello spazio, la perturbazione generata è pari alla somma algebrica delle oscillazioni di ciascuna onda presa singolarmente”*.

Grazie al principio di sovrapposizione delle onde riusciamo a comprendere meglio com'è possibile che durante un concerto due o più strumenti che suonano simultaneamente non vengono sentiti come due suoni distinti ma come un unico suono.

Interessante si rivela una proprietà di questo principio: l'**interferenza**. L'interferenza si genera dal momento in cui due onde che si sovrappongono in un certo punto dello spazio si rafforzano o si indeboliscono a vicenda. Nel concetto di interferenza entra in gioco un parametro fondamentale, ovvero la **fase**. Prima di addentrarci sul fenomeno dell'interferenza, risulta pertanto necessario introdurre il concetto di fase e come essa possa influenzare il comportamento di tale fenomeno.

La fase di una funzione periodica è definita come la frazione di periodo trascorsa rispetto ad un tempo fissato. Prendiamo in considerazione la funzione d'onda delle onde armoniche:

$$y(t) = A \sin(2\pi f t + \varphi_0)$$

Posto $2\pi f = \omega$, avremo

$$y(t) = A \sin(\omega t + \varphi_0)$$

Dove φ_0 rappresenta l'eventuale sfasamento rispetto ad un'onda di riferimento.

Per spiegare al meglio il concetto di interferenza, consideriamo due onde armoniche che viaggiano nello stesso mezzo di propagazione, con la stessa ampiezza e la stessa frequenza, ma sfasate di un certo valore φ .

Le onde avranno le seguenti funzioni periodiche:

$$x(t) = A \sin(\omega t)$$

$$y(t) = A \sin(\omega t + \varphi)$$

In accordo al principio di sovrapposizione delle onde, l'onda risultante sarà uguale alla somma algebrica delle due onde considerate:

$$z(t) = x(t) + y(t) = A \sin(\omega t) + A \sin(\omega t + \varphi) = 2A \cos\left(\frac{\varphi}{2}\right) \sin\left(\omega t + \frac{\varphi}{2}\right)$$

Bisogna quindi studiare come l'onda risultante al variare della fase.

Se la fase è uguale a $2k\pi$, $k \in \mathbb{Z}$ avremo:

$$z(t) = 2A \cos(k\pi) \sin(\omega t + k\pi) = 2A \sin(\omega t + k\pi)$$

L'onda risultante sarà esattamente il doppio dell'onda di partenza. Questo fenomeno prende il nome di **interferenza totalmente costruttiva**.

Se la fase è uguale a $k\pi$, $k \in \mathbb{Z}$ avremo:

$$z(t) = 2A \cos\left(k \frac{\pi}{2}\right) \sin\left(\omega t + k \frac{\pi}{2}\right) = 0$$

L'onda risultante sarà nulla. Questo fenomeno prende il nome di **interferenza totalmente distruttiva**.

Questi due fenomeni rappresentano due casi molto specifici, ma comunque esaustivi per la comprensione del principio di sovrapposizione, il quale ovviamente è implicato nel nostro progetto ma senza raggiungere mai le condizioni che ci permettano di ottenere questi due fenomeni.

Filtri audio

Un filtro è un meccanismo di elaborazione che, preso un segnale in ingresso, produce un segnale trasformato in uscita, contenente un sottoinsieme delle frequenze originali. Il funzionamento dei filtri che prendiamo in considerazione si basa su un parametro: la *Cutoff frequency* (frequenza di taglio).

Distinguiamo 3 tipi differenti di filtri:

- **Filtro passa-basso** – fissata una Cf , questo filtro eliminerà tutte le frequenze al di sopra della frequenza di taglio, facendo passare solo le frequenze comprese nell'intervallo $[0, Cf]$.
- **Filtro passa-alto** – fissata una Cf , questo filtro eliminerà tutte le frequenze al di sotto della frequenza di taglio, facendo passare solo le frequenze comprese nell'intervallo $[Cf, F]$, dove F rappresenta la massima frequenza raggiungibile dal segnale originale.
- **Filtro passa-banda** – fissata una Cf e una frequenza F , questo filtro farà passare tutte e sole le frequenze comprese nell'intervallo tra $[Cf-F, Cf+F]$.

Rumori colorati

Scientificamente il rumore è definito come un segnale di disturbo, indesiderato, che si sovrappone all'informazione o al suono che stiamo analizzando.

Esistono vari tipi di rumore detti colorati caratterizzati dal fatto di avere alcune componenti dello spettro prevalenti sulle altre:

- **Rumore rosa** – viene utilizzato per amplificare maggiormente le frequenze più basse e meno le alte, in particolar modo l'intensità si dimezza quando raddoppia la frequenza. Questo processo corrisponde con un decremento di 3dB per ottava.
- **Rumore marrone** – è il rumore generato dal moto browniano delle particelle, da cui questo rumore prende il nome. Il principio è simile al rumore rosa, ma l'attenuazione dell'intensità è pari a 6dB per ottava.
- **Rumore blu** – è il rumore complementare al rumore rosa, serve dunque per amplificare l'intensità le alte frequenze. Questo processo corrisponde ad un incremento di 3dB per ottava.
- **Rumore viola** – come nel rumore blu serve per amplificare l'intensità delle alte frequenze, ma con un incremento di 6 dB per ottava. Il rumore risulta quindi il complementare del rumore marrone.
- **Rumore grigio** – il rumore grigio ha un andamento molto simile alle *curve isofoniche*, come se l'intensità fosse pari a tutte le frequenze.
- **Rumore bianco** – è caratterizzato da valori di ampiezza del tutto casuali rispetto al tempo e costanti rispetto alle frequenze (i valori seguono la *legge di probabilità uniforme*).