

Progetto CODING – Laboratorio di tecnologie musicali

Composizione musicale algoritmica: implementazione di un processo stocastico markoviano in ambiente Pure Data

Stefano Silvestri

Introduzione

Il progetto rientra a far parte delle attività del laboratorio di informatica musicale e si realizza attraverso varie azioni strettamente legate alla programmazione per la musica e il suono al fine di orientare lo studente verso le logiche e le potenzialità del pensiero computazionale.

- **Musica generativa**

Nell'ambito della composizione musicale alcune correnti di pensiero, da Mozart (1793) a Messiaen (1949), Hiller e Isaacson (1957), Cage (1958), Xenakis (1962) o più di recente Cope 1991, Eno (1993), hanno suggerito nuove modalità di generazione del suono e di costruzione musicale attraverso regole formali e metodi numerici che trovano immediata applicazione con l'uso del computer.

Volendo realizzare un *sistema generativo* che produca automaticamente delle sonorità e delle strutture musicali possono essere applicate numerose strategie come: processi stocastici, grammatiche formali, tecniche di intelligenza artificiale e machine Learning. Il primo caso si riferisce ad alcune formulazioni della teoria delle probabilità dove un certo evento è legato alla sua probabilità di occorrenza. Il secondo caso riguarda un sistema di regole che delineano un insieme teoricamente infinito di sequenze o stringhe finite di simboli appartenenti ad un alfabeto. Modelli semplici di intelligenza artificiale possono essere ricondotti a sistemi basati su grammatiche formali che aggiungono la capacità di produrre a loro volta nuove grammatiche seguendo strategie di Machine Learning o apprendimento automatico: in una prima fase si insegna all'algoritmo come comportarsi (es. training di una rete neurale artificiale) per creare una grammatica che lo condizioni nelle scelte successive.

(Fonte: T. Rosati, Electronic music)

- **Esperienza di *Composizione algoritmica***

Lo schema generale per una composizione algoritmica presenta almeno due fasi che possono essere definite in un ambiente di programmazione musicale:

1. Formalizzazione di un procedimento compositivo → si fissano tutte le operazioni da compiere in sequenza per costruire un brano

2. Esecuzione dell'algoritmo → si lavora sui parametri sonori scelti per essere gestiti entro certi intervalli

E' opportuno notare che, trattandosi di procedimenti formali, tutto quanto definito dagli algoritmi può essere svolto con o senza l'utilizzo di un computer.

L'ambiente Pure Data

L'ambiente di sviluppo indicato per questo progetto è Pure Data ovvero un framework pratico e flessibile che consente di sviluppare algoritmi per l'elaborazione sonora e più in generale strumenti musicali virtuali. Pure Data è un linguaggio di programmazione visuale inizialmente concepito da Miller Puckette (Università di San Diego e IRCAM di Parigi) per la computer music in tempo reale, poi esteso ad opera di una comunità di programmatori ed è attualmente diffuso anche tra artisti visuali, performer e interaction designer. Pure Data è un software open source ed ha uno stretto grado di parentela con MAX/MSP, esso è nato per elaborare e controllare segnale audio: supporta la comunicazione dei dati da un operatore all'altro a due rate diversi ovvero il *sample rate* che per default è di 44100 campioni al secondo ed il *control rate* pari a un sessantaquattresimo del sample rate. Gli operatori che elaborano segnale audio prendono campioni di ingresso in maniera sincrona ad audio rate e allo stesso rate producono segnali di uscita. Tutti gli altri operatori elaborano dati non appena questi sono disponibili (dataflow) con un rate massimo di circa 690 Hz per default (sample rate/64). Tutti i calcoli sono effettuati su numeri floating-point a 32 bit. Questo ambiente di sviluppo, nelle sue differenti release, rappresenta un approccio facilitato al coding incorporando la logica del diagramma di flusso dove ogni operatore è rappresentato da un blocco grafico collegabile a uno o più moduli formando il cosiddetto *patch* ovvero la definizione dell'algoritmo sonoro musicale. La documentazione di Pure Data, con l'insieme delle librerie standard e di terze parti che ne estendono le potenzialità aggiungendo funzioni ad es. per la multimedialità, la grafica, il Physical Computing, etc., è sempre aggiornata e disponibile online.

(Fonte: D. Rocchesso, Introduzione alla programmazione visuale con il linguaggio Pure Data, 2008)

Progetto “Sequenze sonore gestite da catene di Markov”

Classe coinvolta

Classe 5ª - Indirizzo Liceo Musicale dell'IS “Bruno Munari” (sede di Acerra).

Premesse

La classe è composta da 20 studenti (11 maschi e 9 femmine) con motivazioni e competenze eterogenee. Gli studenti hanno pregresse conoscenze di:

- Elementi di audio digitale
- Campionamento, quantizzazione, codifica
- Convertitori ADC e DAC della scheda audio
- Teorema di Shannon-Nyquist
- Architettura del computer
- Differenza tra architettura di Von Neumann ed architettura ad Harvard
- Sistema binario e conversioni tra sistemi diversi
- Cenni ai sistemi DSP per la musica
- Fondamenti di informatica musicale
- Linguaggio di programmazione PureData
- Elettronica analogica e digitale riferita alla catena elettroacustica (simboli dei discreti e semplici reti)
- Tecniche di ripresa e registrazione microfonica (Hard Disk recording)
- Come realizzare missaggio audio, sintesi e analisi dei segnali musicali mediante DAW e wave editor
- Conoscenza operativa del teorema di J. B. Fourier, concetto di spettro e spettrogramma
- Comunicazione mediante protocollo MIDI e OSC

Organizzazione e tempi

2 incontri da 2 ore per la parte teorica:

- laboratorio di *PureData*
- *presentazione e descrizione dell'algoritmo*
- impiego di modelli di sintesi e/o riproduzione del suono

5 ore per l'attività pratica di progetto strutturata secondo le seguenti fasi:

Fase 1 (2 ore): programmazione software, progettazione al computer

Fase 2 (2 ore): realizzazione modelli di sintesi del timbro musicale

Fase 3 (1 ore): documentazione dello strumento virtuale →
critica/analisi/esecuzione del risultato sonoro

Per il corretto svolgimento del progetto si prevede l'impiego del laboratorio di informatica musicale del liceo e delle relative postazioni PC.

Attrezzatura e materiali

- Postazioni PC per ogni studente all'interno del laboratorio
- Ambiente di sviluppo Pure Data
- Monitor da studio o cuffie
- Connessione internet
- Accesso ad account Google per l'archiviazione del progetto software

Tabella dei costi

Consulenza esterna → assistenza tecnica durante la parte pratica (5 ore)	400 €
Spese materiali	Nessuna
Totale	400 €

Figure coinvolte

- **Docente** → Tecnologie musicali
- **Soggetti esterni** → Assistente tecnico per le attività di laboratorio

• Descrizione del progetto

Lo studente dovrà familiarizzare con la programmazione di un algoritmo per la composizione automatica di sequenze sonore implementando un processo studiato nell'ambito della composizione algoritmica generativa. Tale esperienza risulta preziosa al fine di una indagine musicale orientata alle tecniche compositive avanzate, largamente impiegate nella scena musicale contemporanea e nelle arti del suono. Viene proposto in tal sede il caso di studio delle scelte stocastiche attraverso l'impiego di catene di Markov semplificate. Questo modello, molto impiegato anche per l'analisi di sequenze musicali, fornirà una possibile soluzione per la composizione automatica di pattern musicali.

Il timbro sonoro verrà realizzato attraverso modelli di sintesi e/o campionamento del suono da interfacciare al processo stocastico implementato.

Si definisce processo stocastico markoviano, un processo aleatorio a probabilità condizionata in cui la probabilità di transizione da uno stato al successivo dipende dallo stato immediatamente precedente e da una certa probabilità di avanzamento o transizione.

Il risultato consisterà in una distribuzione in sequenza di stati realizzati a partire da uno stato di partenza. Ogni probabilità associata a un singolo stato determinerà il passaggio allo stato successivo del sistema in funzione anche dello stato immediatamente precedente (proprietà di Markov relativa a una catena del primo ordine).

Un processo non markoviano, come il singolo lancio di un dado, rappresenta un processo aleatorio a probabilità semplice tale che non valga la proprietà di Markov.

Ogni stato sarà identificato da un simbolo letterale che verrà associato a un evento sonoro ottenuto con un modello virtuale di elaborazione del suono.

Dati di partenza

Si dispone di una *matrice delle probabilità di transizione* (MPT) che riporta le probabilità di transizione in percentuale verso i singoli stati.

In tal caso viene impiegata la seguente matrice quadrata che indica le probabilità di transizione verso un nuovo stato possibile (prima riga). Questa produrrà una sequenza infinita di simboli mentre nel caso di una matrice rettangolare N*M si transiterebbe prima o poi verso uno stato di terminazione della sequenza:

Stati	D	A	B	C
A	50	0	30	20
B	70	10	0	20
C	80	5	15	0
D	10	10	30	50

Tab.1 - MPT

E' possibile osservare alcune proprietà:

- la somma dei termini di ogni riga è 1 ovvero 100%
- è certo che si transiti almeno verso 1 nuovo stato

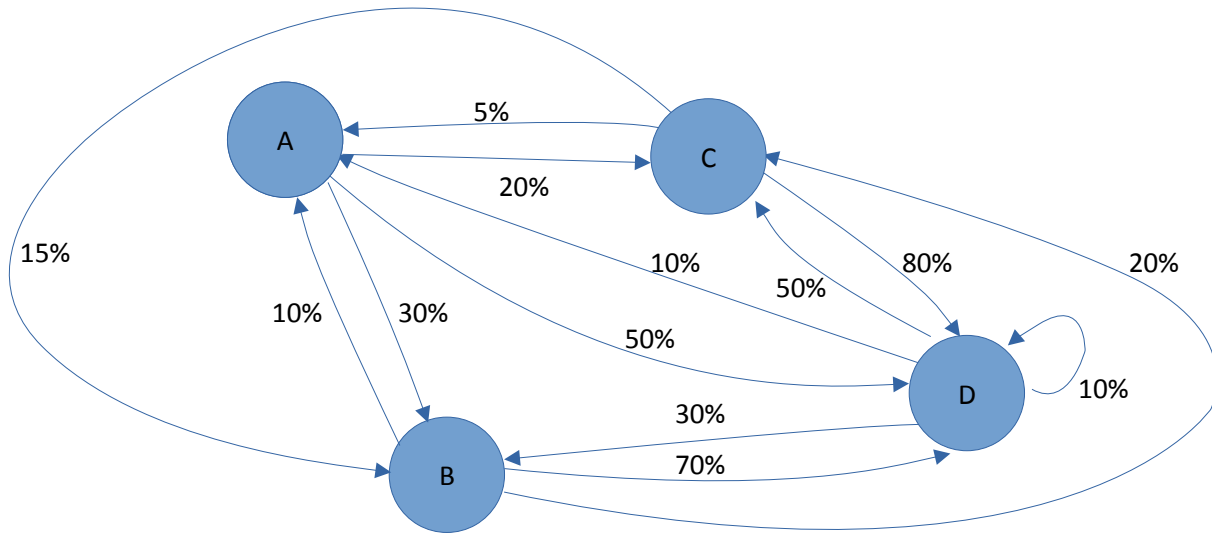
Di seguito viene riportato un esempio di catena di Markov risultante dall'applicazione iterata della precedente matrice a probabilità di transizione:

$$S = \{ B; D; B; D; B; D; B; D; D; D; D; B; D; D; A; B; D; D; D; B; D; B; D \dots \}$$

Se ad ogni stato si associa un certo timbro musicale e se vengono effettuate le opportune scelte di temporizzazione delle transizioni da uno stato al successivo è possibile ottenere una semplice esecuzione musicale automatica.

In relazione alla sequenza prodotta dalla MPT (Tab.1) è da osservare che esiste un unico stato che permette la transizione nuovamente su se stesso. In altre parole nel risultato atteso non esistono ripetizioni consecutive di uno stato tranne che per quello D.

La matrice può essere più facilmente rappresentata in una forma a grafo (come nel caso di un automa a stati finiti probabilistico) che identifica gli stati possibili con dei nodi etichettati secondo l'alfabeto scelto {A,B,C,...} e le probabilità di transizione con degli archi orientati:



Grafo orientato relativo alla MPT di Fig.1

Con questa modalità di rappresentazione risulta semplificata anche la relativa implementazione in ambiente Pure Data.

Creazione di un nodo

Un singolo nodo sfrutta la probabilità semplice per decretare, una volta raggiunto dal nodo precedente, a quale stato successivo procedere. Segue la serie di istruzioni in pseudocodice:

Es. definizione dello stato A

```

estrai numero casuale compreso tra 0-99
se numero casuale ≤ 50 (P=50%) {
    transita verso il suono allo stato D
}
altrimenti se numero casuale è compreso tra 50 e 80 (P=30%) {
    transita verso il suono allo stato B
}
altrimenti se numero casuale è ≥ 80 (P=20%) {
    transita verso il suono allo stato C
}
non esiste possibilità di restare in A
  
```

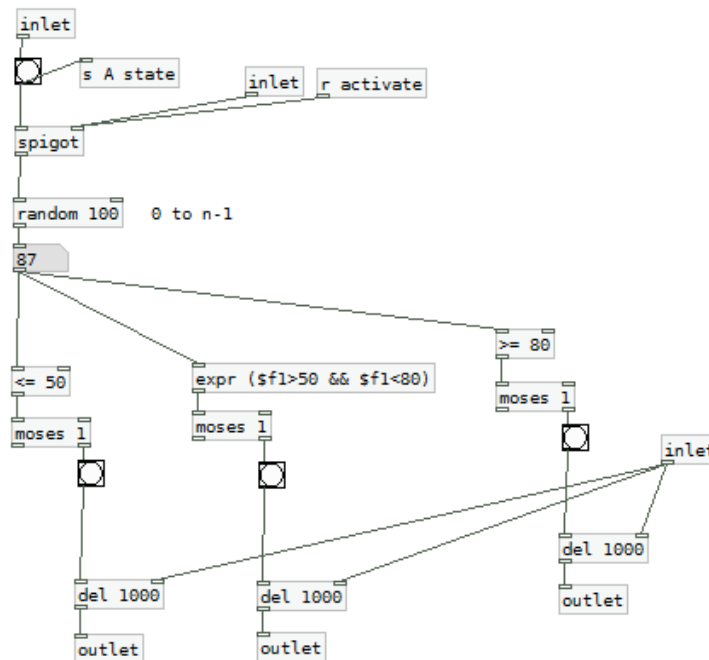
Es. definizione dello stato D

```
estrai numero casuale compreso tra 0-99
se numero casuale  $\leq 10$  (P=10%) {
    ripeti il suono restando nello stato D
}
altrimenti se numero casuale è compreso tra 10 e 20 (P=10%) {
    transita verso il suono allo stato A
}
altrimenti se numero casuale è compreso tra 20 e 50 (P=30%) {
    transita verso il suono allo stato B
}
altrimenti se numero casuale  $\geq 50$  (P=50%) {
    transita verso il suono allo stato C
}
```

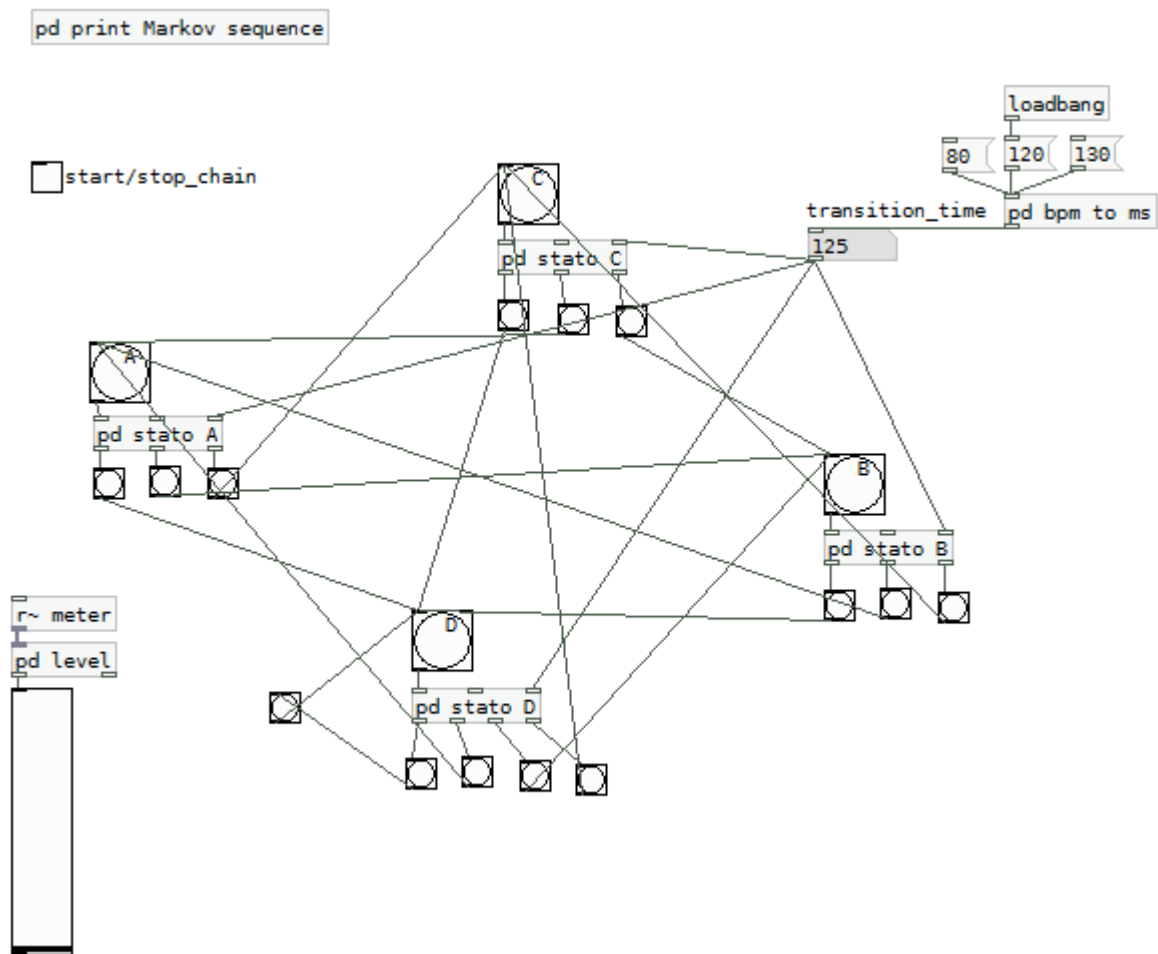
E' possibile aumentare a piacimento il numero di stati definendo nuove probabilità condizionate al fine di aumentare la complessità della macro-struttura sonora risultante.

• Implementazione in linguaggio Pure Data

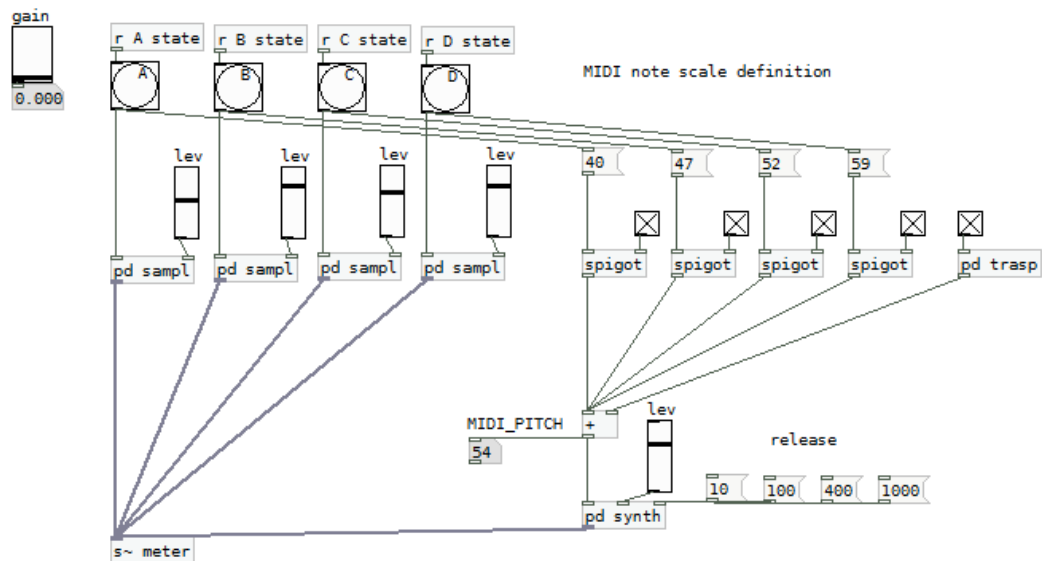
Stato A



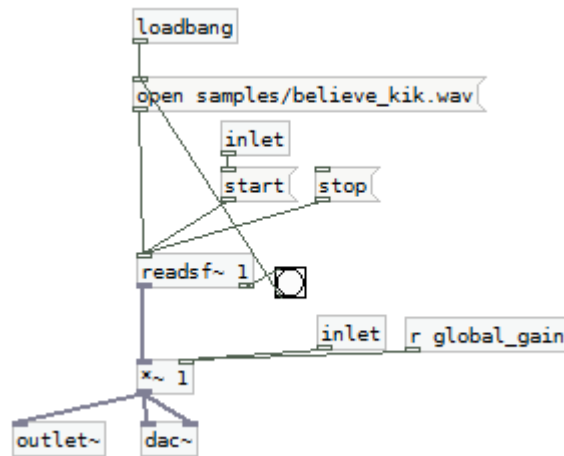
Matrice di transizione completa



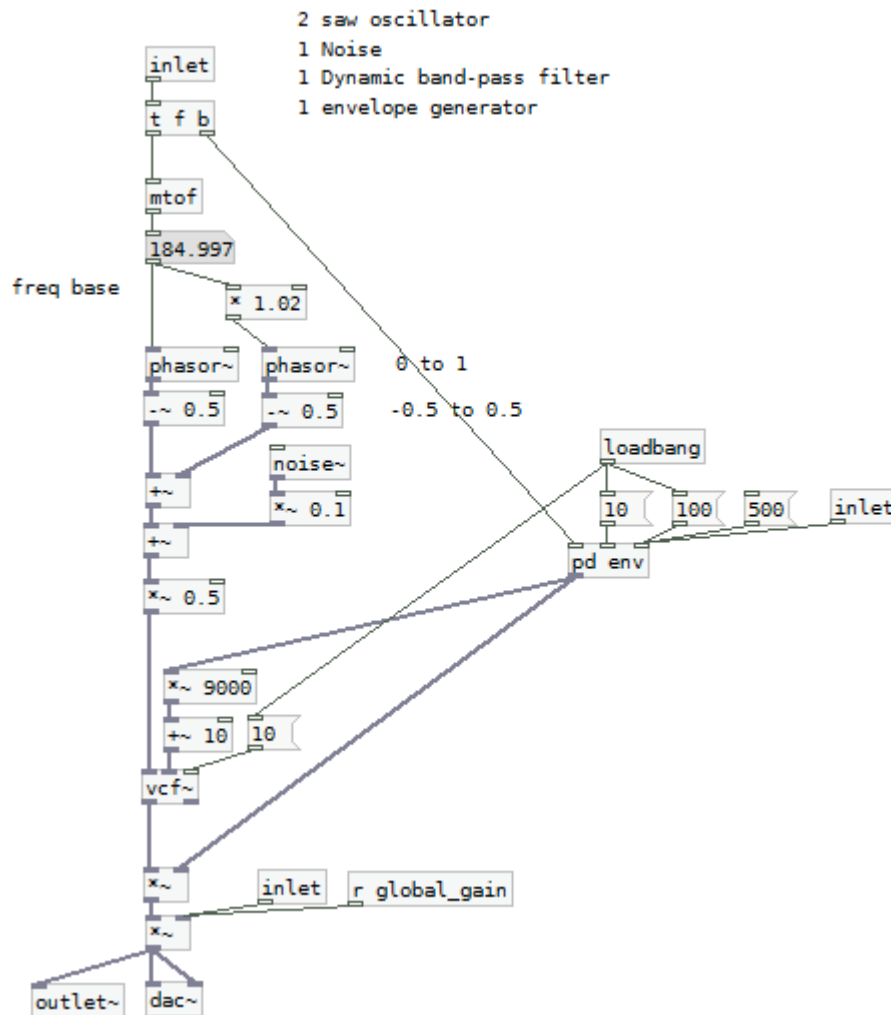
Interfaccia del pannello di regolazione livelli e selezione timbri sonori



Scelte timbriche 1: strumento sampler (lettura di file da HD)



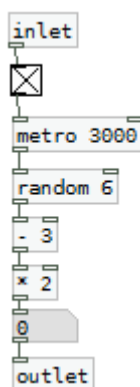
Scelte timbriche 2: strumento synth (modello basic virtual analog)



Trasposizione random

Per la sequenza di note si può optare per un elemento di diversificazione di tipo aleatorio che trasponga ogni dato MIDI pitch.

In tal caso viene impiegata la probabilità semplice estraendo un numero casuale tra -6 e +4 da sommare algebricamente alle note della sequenza markoviana:



Valutazione concordata con il consiglio di classe

Punti	Lavoro di gruppo	Completezza del progetto	Qualità della documentazione	Esposizione del progetto
10	Impegno e attenzione	Progetto portato a termine	Documentazione portata a termine	Lo studente dimostra di padroneggiare con gli argomenti inerenti al Coding e con gli approfondimenti riguardanti la composizione algoritmica e l'elaborazione del suono
8	Il gruppo lavora in modo saltuario ma procede con autonomia	Progetto portato a termine	Documentazione portata a termine ma con qualche imprecisione	Lo studente illustra il progetto generalmente in modo corretto
6	Il gruppo procede facendo affidamento al supporto tecnico del docente	Progetto portato a termine con l'ausilio del docente	Documentazione non completa e superficiale	Lo studente espone il codice con parziale consapevolezza ed uso di termini non del tutto corretti
5	Il gruppo partecipa con scarso profitto	Progetto non portato a termine	Documentazione non portata a termine e non corretta	Lo studente illustra il codice con linguaggio non sufficiente

Osservazioni conclusive

Si sottolinea quanto, nella semplice attività proposta, la disciplina del CODING sia parte integrante del pensiero musicale e come sia presente in più livelli di complessità crescente.

In primo luogo vengono suggerite allo studente nuove soluzioni compositive basate sull'applicazione di espressioni dell'algebra booleana, strutture logiche-formali, procedure di calcolo, tramite operatori matematici e costrutti decisionali che mirano a familiarizzare con il pensiero computazionale. Inoltre le sequenze letterali ottenute sono basate su un modello generativo che rappresenta un interessante approfondimento legato in parte alla statistica e alla teoria delle probabilità.

Il progetto offre la preziosa opportunità di mettere in pratica alcune tecniche compositive sperimentate in differenti repertori della composizione musicale elettroacustica e informatica, come in opere musicali paradigmatiche composte da importanti esponenti pionieri del pensiero tecnologico musicale fin dagli inizi del XX sec.

Le esercitazioni proposte invitano a lavorare sia su algoritmi di gestione parametrica della struttura musicale sia sull'implementazione dei modelli di sintesi ed elaborazione del suono (DSP) affrontati durante l'anno.

Progetto sviluppato nell'ambito del Corso CODING – I.S.I.S. Europa
Docente formatore Avanguardie Educative Prof. Angelo Giordano

Napoli, 03/12/2021 – *Stefano Silvestri*