Progetto del corso di Programmazione per la Musica (Baratè)

Nell'elenco seguente sono elencate, a titolo esemplificativo, alcune proposte per il progetto del corso di Programmazione per la Musica; è altresì possibile proporre progetti personali.

In ogni caso, sia per i progetti qui presenti che per altre ipotesi, occorre mettersi in contatto con il docente per definire più precisamente le specifiche.

A seguito della presentazione del progetto dovrà essere sostenuta una prova orale.

È anche possibile collaborare (al massimo in due persone) su un progetto comune: resta inteso che lo sforzo implementativo dovrà essere calibrato di conseguenza e che ogni componente del team dovrà sostenere la prova orale individualmente.

TRACCE

1. Creare una classe che implementi le codifiche viste a lezione per le altezze e le durate (PC, NC, CBR, PCI, RDC…) con funzioni di calcolo, conversione, visualizzazione, ecc.

2. Riferendosi al progetto music21 (http://web.mit.edu/music21/): creare un'applicazione che processi un file in input (CBR, IEEE 1599, altro) e generi un'analisi o un grafico a scelta tra quelli implementati da music21. Esempi di soluzioni implementabili: la funzione romanNumeralFromChord, uno dei grafici di music21.graph.plot, uno degli algoritmi presenti nella sezione Post-Tonal tools, ecc.

3. Riferendosi al progetto "Black and Byte", presente alla pagina http://www.lim.di.unimi.it/demo_music_research_ita.php: creare un'applicazione che, a partire da una codifica testuale di una partitura nello stile di Black & White N. 2 di Franco Donatoni, crei una corrispondente realizzazione in un formato a scelta (CBR, IEEE 1599, altro), senza considerare i vincoli introdotti nel prototipo e nel contributo e quindi semplificando l'algoritmo.

4. Generatore IEEE 1599 random: creare un'applicazione che, sulla base di una serie di parametri definibili dall'utente, generi un file IEEE 1599 scegliendo casualmente gli elementi presenti. Esempi di parametri definibili: lunghezza del brano, solo note/note e pause, numero di strumenti, altezza minima e altezza massima, durata minima e durata massima delle figure ritmiche, ecc. (si veda, per ispirazione, https://intermorphic.com/nme/3/#the-rules).

5. Step sequencer: creare un'applicazione Swing che implementi uno step sequencer personalizzabile: mentre resta fisso l'asse temporale (in cui però il numero di passi e l'intervallo temporale di ogni passo sono impostabili) sull'altro asse è possibile scegliere numero di tracce e associazione di ogni traccia, usando una libreria per la generazione audio per la riproduzione/sintesi.

6. Piano roll microtonale: creare un'applicazione Swing che visualizzi un'interfaccia in stile piano roll in cui però le note non corrispondono alla scala cromatica del sistema temperato equabile, ma possa essere scelta frequenza iniziale, frequenza finale e numero di suddivisioni dell'asse verticale.

7. Sonorizzazione/musificazione: prendendo spunto dal progetto presentato dal LIM alla manifestazione MeetMeToonight 2016 (http://mmt16.lim.di.unimi.it/index.php), creare

un’applicazione di sonorizzazione/musificazione che elabori un set di dati scelti appositamente per creare una corrispondenza tra essi e parametri musicali/sonori dell’output audio, consentendo all’utente un’impostazione delle associazioni con i parametri considerati. Ad esempio, ad un dato si può associare il pitch delle note, e l’utente dovrebbe poter stabilire pitch minimo/massivo, scala lineare/logaritmica, ecc.