Quinn A. Hannah White and Griffin Good
Procedurally Generated Music in Java
6/1/2015

Random Music: Procedural Counterpoint Generator

Introduction

This program automatically generates, based on user input, a counterpoint composition of a chosen key and octave (within the bounds of the Java sound API). As a secondary step, the program also takes this generated composition and creates a MIDI file that plays the composition. The final design of our project turned out to be very similar to the initial design of our project, with a few major and minor changes. First of all, the Line class handles the majority of the logic for creating the composition regardless of the species of counterpoint. In our initial design, this class would only have been a skin for interacting with an ArrayList of type Note. Reflecting this, our Species One through Four classes poses a considerably reduced amount of logic. Apart from this major change, there are a few other minor changes documented in the following list of classes and their applications:

Documented Classes

See the attached JavaDoc index.html for all documented classes and appropriate functions. This should be within the .zip file sent along with this document.

Goals

- Success: Logic to handle the creation of counterpoint
- Success: Handling for created counterpoint to a MIDI file
- Success: Wrote logic for first species of counterpoint
- Failure: Did not write handling for counterpoint to text file
- Failure: Did not create logic for second species of counterpoint
- Failure: Did not create logic for third species of counterpoint
- Failure: Did not create logic for fourth species of counteproint

Sections

- Quinn: Output to MIDI file
- Quinn: Handling of user input
- Quinn: With Griffin: First draft of working organization and logic
- Griffin: Line creation
- Griffin: Note and Pitch handling
- Griffin: Logic for first and fourth species counterpoint
- Griffin: With Quinn: First draft of working organization and logic