TJHSST Computer Systems Lab Senior Research Project
Designing a Music Scripting Language
2008-2009

Casey Mihaloew

October 27, 2008

Abstract

The goal of this project is to create a scripting language that can describe music and be compiled into various forms of displaying music. There are four basic quantities that have to be dealt with in seperate ways: pitch, volume, note length, and tone. There is also the problem of defining these quantities for every note without having miles of repetative text. This problem will be addressed by a unique method that I call mapping that allows groups of commands to have common pieces.

1 Introduction - Elaboration on the problem statement, purpose, and project scope

1.1 Scope of Study

The language will cover basic notes with rhythms and articulations. It will take into account different instruments as well as multiple of the same instrument. It will eventually allow continuous changes of certain factors like pitch, volume, and tempo, known as glissando, crescendo or diminuendo, and accelerando or ritardando. Currently, it can recognize notes of different pitches and modifiers on the pitches, although it doesn't do anything with that.

1.2 Expected results

The result of this project will be to have a program that can parse a description of music into a sound file or a text description of all of the notes. The main differences with this project and other similar projects already copmpleted is that this combines an easy input file with the object-oriented Java language and that the scripting part of the language will use the original concept of mapping commands onto a list of other commands. What I want to learn from this project and the focus of this project is how to describe something as complicated as music. I am also doing this because I want a way to generate sound files consistantly from music that I write.

1.3 Type of research

The type of research is generally use-inspired basic research because it asks how a program could create music.

2 Background and review of current literature and research

The first paper that I read about this topic was Programming Languages for Computer Music Synthesis, Performance, and Composition, by Gareth Loy and Curtis Abbott. The focus of this paper is first to outline the various difficulties in designing a programming language for music, then discuss sound synthesis, and finally review many of the languages already created up to the time of publication (June 1985). There had already been a lot of advances in this area by 1985 as it was one of the earlier hard problems attempted by computer scientists.

As said above, there are many programming languages for music, amny created after this paper. Some are mainly a programming language like CMusic. One (jMusic) uses Java and object oriented design similar to mine. There are also scripting languages such as LilyPond that focus on easy input. My approach is in between jMusic and LilyPond, using Java methods and a scripting language.

3 Procedures and Methodology

This program is currently being written in Java due to that it has object-orientedness and the ease of doing pictures and sound with built in methods. Currently, I have a basic outline of my program that is somewhat functional and the next step of adding sound can work with what I currently have. Over the time period of working on this, other nuances will be implemented like articulation as the need comes. After that, effort can be put into making the sound sound more natural.

The output of the program, as described above, will be a sound file. As for visual representations, there could be a picture of the sound wave, but more descriptive to the naked eye would be rectangles along a time line to demonstrate that the program could keep track of all of the parameters of music. The input to the program will be the music file according to my specifications, so the only way for me to get data is writing it myself.

The testing of the program now consists of makeing sure that the instruments are keeping track of all of the notes and that the notes are keeping track of all that they need to. The analysis is first printing out the parsed input file and then creating a print method for both instrument and note classes to see what they know. The first analysis is succesful, but only for simple input files. The second analysis is succesful.

The requirements that I have for my program in the future will be to keep track of dynamically changing volume, pitch, tempo, and tone for multiple notes on multiple instruments. It should be able to output it to a picture, text, or sound file according to various specifications. It will be structured in an object oriented paradigm, with classes for the note, instrument, and possibly each of the four parameters discussed throughout this paper.

4 Expected Results

Again, I expect to be able to input a piece of music and generate a sound file from it. The result can be shown as a graph of the wave or a pictoral representation of pitch (color), lengths (length), and volume (width) with rectangles. There are other applications that could be used to expand on this language, such as a sheet music generator. This could also be used as the basis of other music based programs such as analysis or random music generation.