William Kluge

autoDEC Milestone

Abstract:

autoDEC is a program that takes a WAVE or MIDI as input file and turns it into a script for the popular voice synthesizer DECtalk. DECtalk's big claim to fame is being used by Stephen Hawking, and is often known for the signature vibrato that it adds to words.

Introduction:

autoDEC was created because translating a song into a DECtalk script takes a lot of time and work. The remainder of this paper will describe how the many systems of autoDEC interact with each other, the capabilities of autoDEC, and other work that has been done with DECtalk.

System Description:

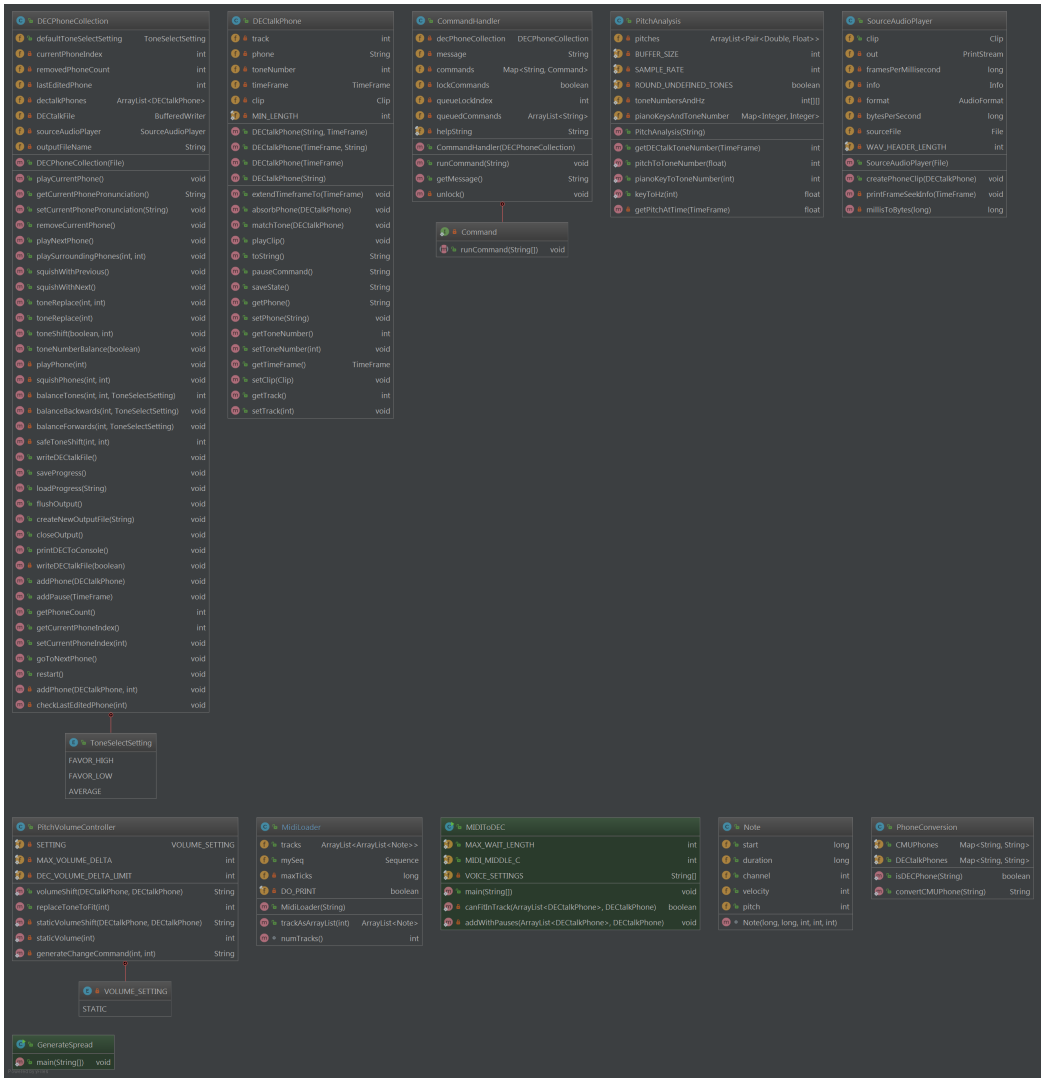
As of now, autoDEC is interacted with solely through a command line interface. All user interaction is currently located in src/test. Only production code is located in src/main; since the user interface has not been developed yet no code relating to interaction is available in that directory. Due to the fact that nothing quite like autoDEC already exists, the many ways of getting the necessary information had to be tested.

The first solution (and the second best) was to use a library called Sphinx for a process called phone recognition. Phone recognition is the process of taking an audio source (in this case a WAVE file) and breaking it down into pieces that are recognized as phones of a specified language. Phones are the sounds that make up words. For example the word “Hello” is made up of of phones (in DECtalk syntax) “hx”, “eh”, “l”, and “ow”. The exact structure of DECtalk syntax is not part of the scope of this document, but the resources section contains plenty of information about DECtalk for those who are curious.

The second solution, which will be how autoDEC actually works in production, is to take a MIDI file, look at all the notes, and break them into separate WAVE files using a special version of

DECtalk that can be run from the command line. This process works extremely well. In fact, at this point songs are recognizable without any manual tweaking by the user.

Many other options were explored and their code remains under the testing directory. These solutions do not work very well, but the results are still interesting to see. The following images are the UML for autoDEC excluding any code I have not worked on (namely the code in the WaveFileHandling package) and code irrelevant to the final version of autoDEC (some of the classes under test/. Classes relevant to PhoneToDECTest is still included as this process may be implemented in the final version if I have time to get the MIDI process working well enough.



Requirements:

autoDEC makes making songs in DECtalk easy. To recreate a song using DECtalk a person would have to find sheet music, manually find the phones related to every word, break up everything based on the time it takes, and arrange everything perfectly. autoDEC simplifies this process by only requiring a MIDI file or a WAVE file containing only lyrics.

Literature Survey:

While there are other projects that seek to accomplish similar things to autoDEC, they're all unfinished.

User Manual:

All the user needs to do is run autoDEC with their file as a parameter and DECtalk syntax will be generated. In some cases the program will need feedback from the user, which is why a GUI is being developed. This is only necessary when using a WAVE file due to the accuracy of Sphinx's phone recognition, MIDI files are handled completely automatically. Currently there is only a command line interface, but this will be completely replaced with a GUI to make the process simpler. The MIDI process requires being run in a Windows environment due to its dependency on the DECtalk distributed.

Conclusion:

Overall, autoDEC is looking extremely promising and is working better than I had thought it would. The implantation of a GUI paired with improvements to code will make autoDEC a program that everyone can use and enjoy.

Resources:

DECtalk Commands
How To DECtallk
DECtalk Phonemic Symbols
Sphinx