

Assignment 4: Arpeggiator

Overview

In Lecture, we introduced a GM (General Midi) Synthesizer called *FluidSynth* to give us a larger palette of instrumental sounds than sine waves. We also laid the foundation for generating rhythms by creating `CLock`, `SimpleTempoMap`, and `Scheduler` classes. We tested the results with a `Metronome` and `Sequencer` class. In this assignment, you will create a more interesting real-time rhythmic player: an Arpeggiator.

Part 1: Arpeggiator [15pts]

One common technique for automated rhythmic/melodic generation is the Arpeggiator. Create a new class `Arpeggiator` (in the same style as `Metronome` and `NoteSequencer`) that does arpeggiation of a given set of notes. Create real-time controls to:

- Start and stop the arpeggiator's note production
- Change the notes being played
- Change the rhythmic value and duration of the notes
- Change the direction of the notes to be either up, down, or up/down.

For example, if notes are: [60, 64, 67, 72], pulse is 1/8th note, and direction is down, your arpeggiator would produce a steady stream of 1/8th notes: 72, 67, 64, 60, 72, 67, 64, 60, etc...

Start with the template class provided. You will need to add member variables and create the appropriate helper functions.

Test your class by using `MainWidget1`. I have written the test hookups for you. Test all aspects of the class interface and make sure that everything works and sounds right when you change values in real time. You should also keep the `Metronome` running as you test the `Arpeggiator`. This is to make sure that your `Arpeggiator` plays in time to the beat appropriately. When you change to a different rhythmic pulse, make sure that the arpeggiator notes are aligned to that rhythmic grid.

Part 2: Arpeggiator Interactive Control [10pts]

Write a system to control the `Arpeggiator` using the mouse/trackpad and create a simple interactive melody player:

- Mouse down starts the flow of notes
- Mouse up stops the flow of notes
- Mouse x-axis controls the pitch range of the Arpeggiator's notes (low notes to high notes)
- Mouse y-axis controls the rhythmic grid of the Arpeggiator notes (slow notes to fast notes)

The details of how these controls should work is up to you. Remember that you can set any group of notes on the `Arpeggiator` (they don't have to be chord tones). But you should make the resulting notes sound musically coherent.

Try to make the interface as smooth and reactive as possible. As you slowly move the mouse left and right, the pitch-range of the generated notes should responsively follow the mouse movement.

Part 3: Make an Interactive Song [15pts]

Create a multi-part song (for example, a short looping sequence with a bass line, a percussion line, and a lead line). Remember that to make percussion sounds, you can use the FluidSynth percussion bank 128. Some of the parts can be fixed (non-interactive). At least one of the parts should be controlled interactively by the mouse, as in Part 2.

You can use any combination of NoteSequencers and Arpeggiators to create your song. Remember that each part should occupy a different MIDI channel and have a different patch.

Try to make the interactive Arpeggiator select notes based on the changing harmonies of your song. For example, you may pick a I, IV, V bass line and then have the Arpeggiator's note selecting scheme adapt.

Pay attention to the volume balance between your different parts. You can make an overall channel louder or softer by using control-change 7, or you can adjust the velocity parameter of `noteon()`.

Part 4: Add Graphics [10pts]

Generate graphics for your interactive system. When your arpeggiator plays a note, use a callback function to cause graphical animations to appear on screen. Be artistic! Create a video of you performing on your system.

Document how to control your system in a README file.

Finally...

For this assignment, submit a zip file that has all the necessary files to run your pset, but please **do not upload** the 147MB Soundfont bank since I have a copy. This will make uploading/downloading faster.

MIT OpenCourseWare
<https://ocw.mit.edu>

21M.385 Interactive Music Systems
Fall 2016

For information about citing these materials or our Terms of Use, visit: <https://ocw.mit.edu/terms>.