

# Assignment 1

## EECS 4462 3.0 Digital Audio, Fall 2018, Section A

**Due:** Monday, October 15, 2018, 9:00am.

**Format:** In teams of two.

### Creating a more advanced Arpeggiator MIDI plugin

The purpose of this assignment is to give you experience creating MIDI plugins with JUCE. This will help you understand the MIDI format better, as well as how to read, handle, and create MIDI events. It will also help you become familiar with JUCE, which we will use for audio plugins as well.

### To get started

To begin, follow the steps below. All necessary links are posted on the course webpage in the Assignment 1 section.

1. Download and install JUCE if you are working on your own computer
2. Create a Roli account and log in to it when JUCE requests it
3. Run the Projucer, create, and build an empty project to ensure all tools are installed correctly. See the "Getting started with the Projucer" online tutorial for more information
4. Download and compile the Arpeggiator Tutorial plugin
5. Test the Arpeggiator Tutorial plugin on the Audio Plugin Host to ensure that it does arpeggiate incoming MIDI notes. See the "Create a basic Audio/MIDI plugin, Part 1: Setting up" tutorial on how to compile the Audio Plugin host

### What to do

At this point you have a plugin that arpeggiates incoming notes in an ascending order with a user selected speed. For example, if the user pressed down on notes A, B, and C, simultaneously, the arpeggiator will play the following notes in succession: A,B,C,A,B,C,A,B,C,...

For this assignment, you have to complete the following four tasks:

1. Modify the implementation of the arpeggiator plugin, so that the notes can be played also in descending order. The GUI of the plugin must allow the user to choose either ascending or descending order. In the above example, the played sequence would be C,B,A,C,B,A,C,B,A,....
2. Modify the implementation, so that the arpeggio can span multiple octaves. A new parameter must be added to the GUI for the user to choose the number of octaves.

For example, if the input notes were in the second octave, i.e. they were A<sub>2</sub>, B<sub>2</sub>, and C<sub>2</sub>, and the user has selected an ascending pattern of three octaves, the sequence would be: A<sub>2</sub>,B<sub>2</sub>,C<sub>2</sub>,A<sub>3</sub>,B<sub>3</sub>,C<sub>3</sub>,A<sub>4</sub>,B<sub>4</sub>,C<sub>4</sub>,A<sub>2</sub>,B<sub>2</sub>,C<sub>2</sub>,....

Adding 12 to the note number of a MIDI event makes it one octave higher.

3. Modify the implementation so that other note sequences are supported as well, i.e. sequences where the notes are not strictly ascending or descending, or where notes are repeated.  
An example of a not strictly ascending or descending sequence would be A,C,B,A,C,B,A,C,B,...  
An example with repeated notes would be A,B,A,C,A,B,A,C,....
4. Modify the implementation so that notes in the arpeggio can have different durations.  
For example, if the sequence is A,B,C,A,B,C,A,B,C,... the A and B notes combined would last for the same amount of time as the C note by itself.

Please note that submissions that minimally satisfy the above requirements will receive a mark of at most B. For higher marks, submissions must demonstrate user-friendly GUIs that allow customization of the note sequence by the user, as well as some level of originality.

## How to Submit

Before the deadline, send an email to your instructor ([bil@eecs.yorku.ca](mailto:bil@eecs.yorku.ca)) that lists the names, student numbers, and emails of both team members, and provides a link to a .zip file hosted online that contains:

1. The .dll file that is your plugin.
2. All the source files needed to compile your code (if you create new files, keep them all in the same directory)

Your submission will be tested with the Audio Host Plugin executable posted on the course website.

Do not send the .zip file as an attachment as it may be rejected.  
The .zip files will be downloaded immediately after the deadline.