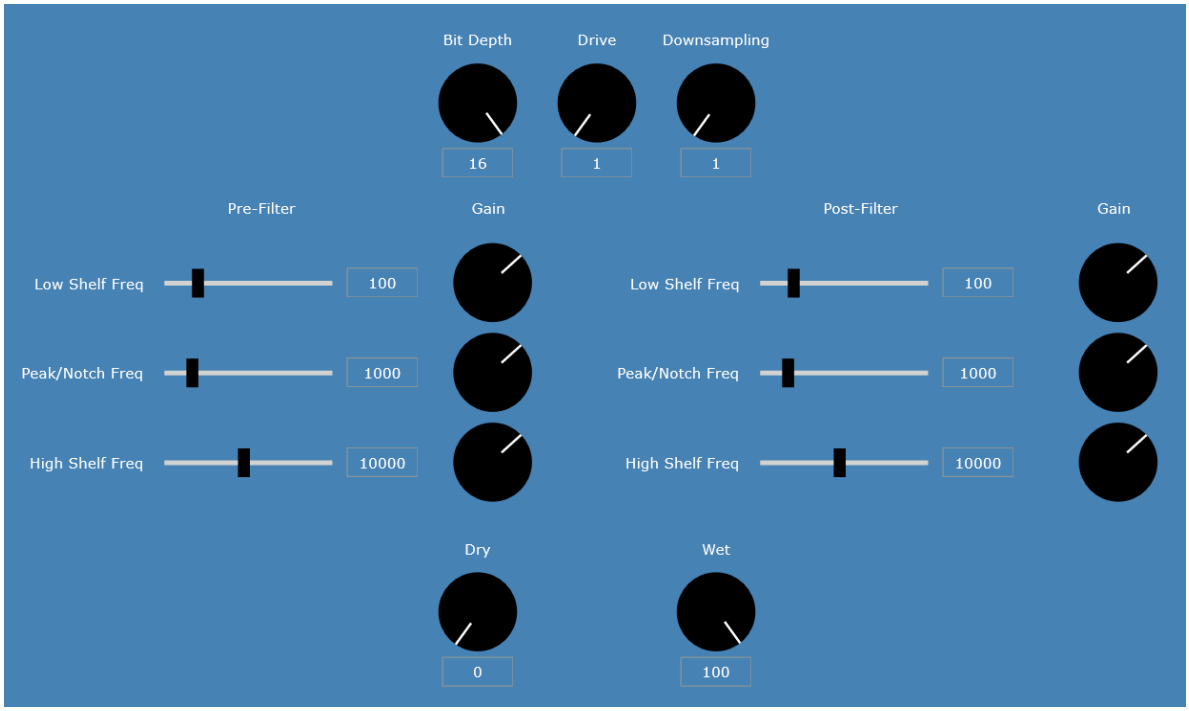
Jaco Asistores

Luke Dahl

MUSI 4545

10 Dec 2024

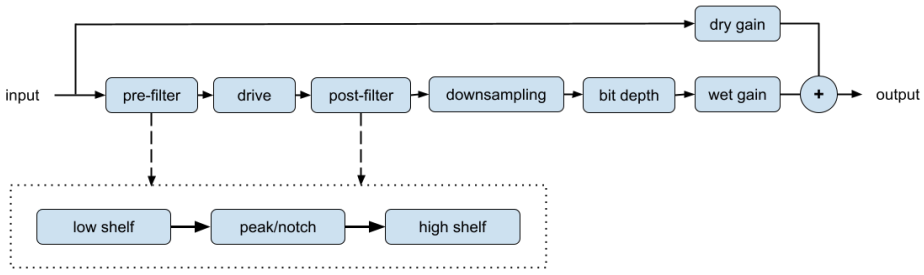
Final Project: Distorticon



For my final project, I decided to create a distortion plugin. I personally enjoy effects in which the original audio input can be completely transformed into something almost unrecognizable, which is why this type of plugin interested me. In addition to the classic drive functionality of my distortion effect, I also wanted to add extra features in order to allow the user to modify the input audio even more, so I added bit depth reduction and downsampling functionalities. I also added filters before and after the distortion in order to give the user finer control over the character of the distorted sound. Collectively, all user parameters are listed as follows:

1. Pre-filter: low shelf, peak/notch, and high shelf filter before the distortion is added
    a. Low shelf transition frequency: 0 Hz – 500 Hz
        i. Gain: -40 dB – +20 dB
    b. Peak/notch center frequency : 200 Hz – 5,000 Hz
        i. Gain: -40 dB – +20 dB
    c. High shelf transition frequency: 1,000 Hz – 20,000 Hz
        i. Gain: -40 dB – +20dB
2. Drive: increases the gain and soft clips it at a certain threshold using the *std::tanh* function
3. Post-filter: filters after the distortion
    a. Same filters and gain control as pre-filters
4. Downsampling: reduces the sample rate resolution by passing through every *n* samples only
5. Bit depth: quantizes input sample to a limited number of levels
6. Dry gain: Gain control for the original, dry audio input. Summed with the wet gain
7. Wet gain: Gain control for the affected, wet audio input. Summed with the dry gain

The order in which the signal is affect by these parameters can be represented by the following signal flow diagram. Note that this diagram represents one of two identical stereo channels, and that both left and right channels follow this same exact signal flow:



I chose to put the post-filter before the bitcrushing effects, because according to my research the convention is to put the filter before the downsampling in order to prevent aliasing. I also discovered that putting the bit depth after the downsampling is more ideal so that the audio that is being bit-reduced already has a low resolution, which exaggerates the bit depth reduction even more.

Originally, I had also planned to add two particular features to this plugin that unfortunately did not make it into the final product. The first feature is a toggle button to switch between soft and hard clipping. Implementing each clipping method individually was not difficult; however, I encountered many problems when trying to add the actual toggle functionality into the plugin, therefore I chose to go with soft clipping only because it is not as harsh to the ear as hard clipping. The second feature I was unable to implement is an attack slider, which would have slowly added the wet signal onto the dry signal. However, I simply did not have enough time to explore adding this functionality to my plugin.

To test this effect, I used a simple drum break as my input audio because the transients of the drums allow for the distortion to sound more noticeable. I used this drum break to create five simple test cases to showcase the features of my plugin:

1. A test case that emphasizes the drive effect
2. A test case that emphasizes the bit depth reduction
3. A test case that emphasizes the downsampling functionality
4. A test case that uses all three effects to an extreme in order to showcase how much my plugin can transform the original audio
5. A test case that uses all three effects, but more subtly and within the context of a larger piece of music

I changed the pre- and post-filter parameters in each test case to accentuate aspects of the specific effect I wanted to emphasize. For test case 1, moderately heavy drive combined with a pre-filter low boost and a post-filter notch gave the audio a very boomy kick drum and noisier high frequencies. For test case 2, heavy bit depth reduction with slight drive to increase volume, as well as lowering the gain on the post-filter high shelf to slightly soften the harsh high frequencies, created a choppy and crunchy sound. For test case 3, a high downsampling factor with a pre-filter low boost and a slight post-filter low cut made for a very gritty, lo-fi effect. For test case 4, I used heavy drive bit depth reduction with a moderate downsampling level, combined with a pre-filter low boost and peak, as well as a post-filter notch created a very distorted and crushed drum break with a very noticeable resonant frequency. For test case 5 (i.e. the use case test), I used similar settings to test case 4, except I turned everything down and kept everything somewhat more subtle.