

1 Introduction

FDWaveShaper32 is an audio processing plugin designed to offer sound shaping and enhancement capabilities. It's partly a mastering tool. It combines multiple signal processing techniques into a single interface, catering to the needs of sound designers, music producers, and audio engineers in general.

Key Features and Functionality

1. High-Quality Signal Processing

- Implements 4x oversampling on the effected signal path, ensuring pristine audio quality and minimizing digital artifacts. Used JUCE's oversampling library. Initially had IIR filtering in this, but it was causing phase differences between the wet and dry signals. Changing this to FIR resolved the problem.

2. Customizable Waveshaping

- Features a unique waveshaping algorithm for harmonic enhancement, allowing users to add richness and character to audio signals. Initially tried to create a waveshaper plugin and was inspired by Paul Frindle's "Sonnox Inflator". Found lots of forum discussions to do with mimicking the waveshape used in the original plugin and tried to approximate that waveshaper.

3. Tonal Balance Control

- Incorporates a Tilt EQ for efficient and musical adjustment of the overall tonal balance.

4. High Frequency Taming after the Tilt EQ

- Uses a variable lowpass filter for smooth high-frequency attenuation, offering precise control over the signal's brightness and presence.

5. Parallel Dynamics Processing

- Includes two fixed parallel compression settings. Chose settings that would be versatile enough to work on a variety of program material. Added a Highpass filter (set at 120Hz) to the side-chain of the "Simple Compressor" processor that was used. This smooths the compression when using it on stereo mixes because it prevents it from reacting to sustained bass tones and helps it to react mainly to the transients of the signal. It is called "Lift" in the GUI, with three settings: 0 (off), 1, and 2.

The buttons themselves, within the Editor, pass the compression parameter settings to the Simple Compressor processor using this function:

```
void FDWaveShaper32AudioProcessorEditor::buttonClicked(juce::Button* button)
{
    if (button == &flatButton)
    {
        audioProcessor.setCompressorParameters(0.0f, 0.0f, 100.0f, 0.0f, 1.0f, 0.0f);
        setting1Button.setToggleState(false, juce::dontSendNotification);
        setting2Button.setToggleState(false, juce::dontSendNotification);
    }
    else if (button == &setting1Button)
    {
        audioProcessor.setCompressorParameters(-40.0f, 20.0f, 40.0f, 0.0f, 1.0f, 4.1f);
        flatButton.setToggleState(false, juce::dontSendNotification);
        setting2Button.setToggleState(false, juce::dontSendNotification);
    }
    else if (button == &setting2Button)
    {
        audioProcessor.setCompressorParameters(-40.0f, 30.0f, 100.0f, 800.0f, 10.0f, 6.2f);
        flatButton.setToggleState(false, juce::dontSendNotification);
        setting1Button.setToggleState(false, juce::dontSendNotification);
    }
    else if (button == &clipperEnabledButton)
    {
        bool isEnabled = clipperEnabledButton.getToggleState();
        audioProcessor.setClippingEnabled(isEnabled);
    }
}
```

6. Clipping

- Features a final-stage clipper for peak limiting and saturation, ensuring optimal output levels and adding analog-style warmth when desired. Not really keen on this but it's an option. Listened to the 5-6 types of clipping/limiting available in the "Peak Eater" code found and settled on the logarithmic. Would like to spend more time finding a more useful implementation of this end stage limiter.

7. User Interface

- Designed with a user-friendly GUI for efficient workflow and real-time parameter adjustments.

Technical Implementation

FDWaveShaper32 ("Nitro N-Rich") is built using the JUCE framework and developed in C++, ensuring cross-platform compatibility and high performance. The plugin integrates several DSP concepts and algorithms, combining them in what is hoped to be a unique architecture to create a powerful and flexible audio processing tool.

Third-Party Contributions

The development of FDWaveShaper32 has benefited from the open-source community. Key components have been adapted or inspired by the following projects:

- "Sonnox Inflator" by Paul Frindle.
- "Peak Eater" Clipper by Vladyslav Voinov.
- Tilt-eq Plugin by Jacob Curtis.
- Lowpass/Highpass Filter implementations by Jan Wilczek.
- SimpleCompressor by Daniel Rudrich.
- "ff meters" Copyright (c) 2017 - 2020 Foleys Finest Audio Ltd. - Daniel Walz

These contributions have been crucial in achieving the high standard of audio processing in FDWaveShaper32.

2 Signal Flow

