

- Making Instruments
 - Oscillators
 - There are 7 oscillators in the top middle all in a row. Each oscillator has a waveform, volume, and frequency.
 - The waveform selection is at the top, where abbreviated waveforms can be selected, such as Sin, Squ, Saw, Tri, Cu1, Cu2, and etc.
 - The volume control is in the middle and is linear from 0.0 to 1.0.
 - The frequency is a knob at the bottom. It is a ratio of the base frequency of the note played and is from 0.0625 and 16.
 - ADSR
 - This is not included when saving instruments.
 - Attack: In seconds, the duration it takes for the volume to reach max volume linearly from the start of the note pressed.
 - Decay: In seconds, the duration it takes for the volume to reach the sustain volume linearly from the peak before.
 - Sustain: As a ratio, the volume sustained while the note is held down.
 - Release: In seconds, the duration it takes for the volume to reach zero linearly from whatever the volume was when it was released.
 - Custom wave files (cu1.txt, cu2.txt, etc)
 - These files need to be in a folder in the working directory like `./custom_waves/cu1.txt` , where the `custom_waves` folder is in the same folder as the executable for the microtonal program.
 - The format for custom waves txt files is a list of floats/decimals separated by new lines. There can be any number of floats/decimals. The program will interpolate between each point to make the waveform with equal intervals. Make sure the first and last numbers are the same and also the values are mostly in between 1.0 and -1.0. They can be higher or lower.
- Unused code
 - In the `synth.cpp` there are some unused parameters that give each oscillator their own LFO and ADSR. If kept to their default values, it should hopefully not interfere with saving, loading, and other instrument uses. Also since it adds more computation, it may be the source of popping noises and lag when multiple notes are pressed. The `getOsc` function is used only for the LFO, but is very similar to the `getSamples` function, which each oscillator uses instead of the JUCE oscillator function because the JUCE implementation cannot use lambdas that use global variables or pass variables in other ways.