Computer Music – Languages and Systems

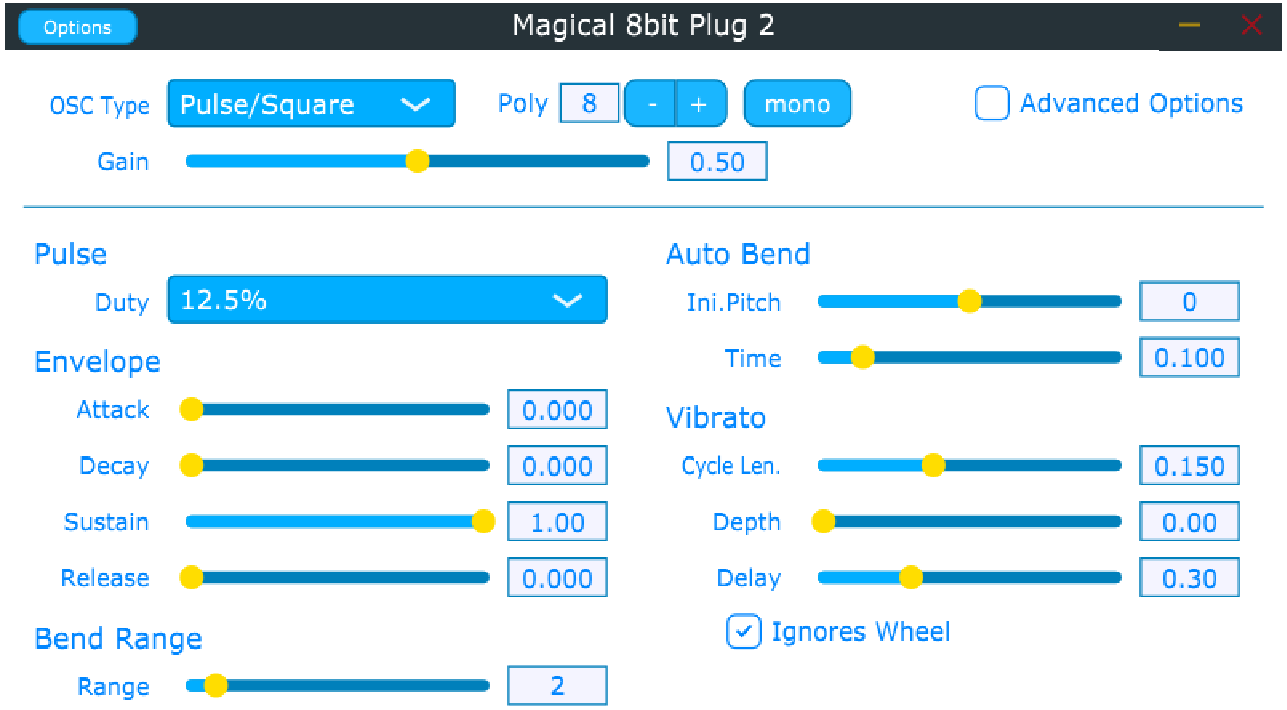
Homework #2
Magical 8bit Plug 2

Group 4: DRUM TEAM
Brusca Alfredo 10936149
Marazzi Alice 10625416
Pomarico Riccardo 10661306

The plugin

Magical 8bit Plug 2 is a software-synthesizer developed by Yokemura YMCK.

It generates primitive electronic sounds like the old 8bit game consoles.



OSC Type: Pulse/Square

Poly: 8

□ Advanced Options

Gain: 0.50

Pulse

Duty: 12.5%

Envelope

Attack: 0.000

Decay: 0.000

Sustain: 1.00

Release: 0.000

Bend Range

Range: 2

Auto Bend

Ini.Pitch: 0

Time: 0.100

Vibrato

Cycle Len.: 0.150

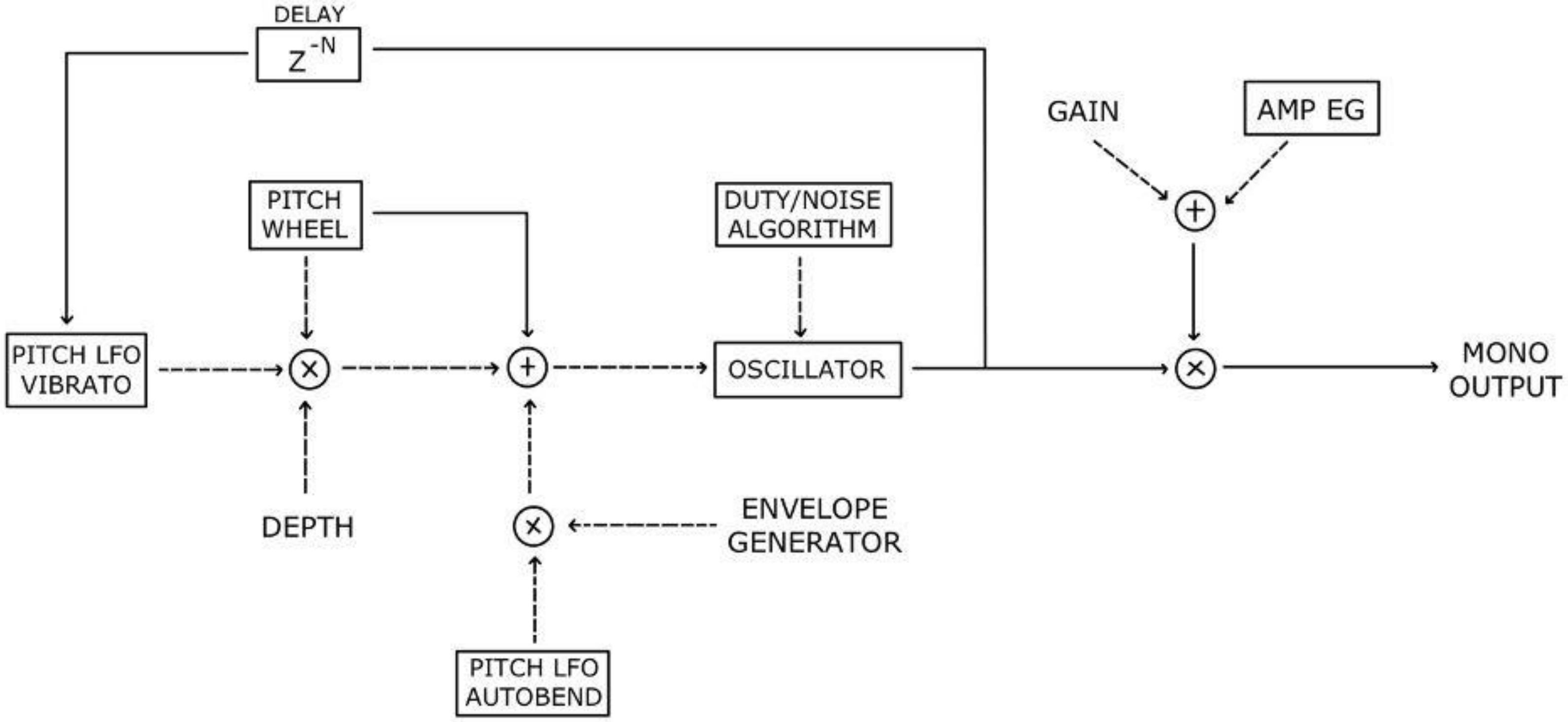
Depth: 0.00

Delay: 0.30

☑ Ignores Wheel

Block Diagram

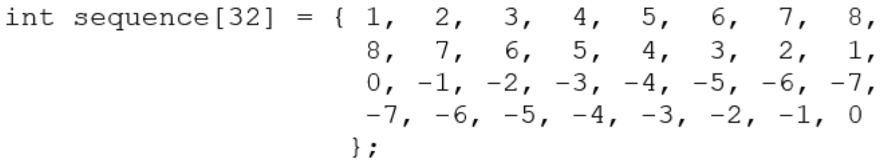
It can produce the 8bit-specific sounds, namely pseudo traingle and low resolution noise, that are hard to reproduce with ordinary synthesizers.
It also implements precise controls for 8bit-style expressions like pseudo polyphony and duty envelope.



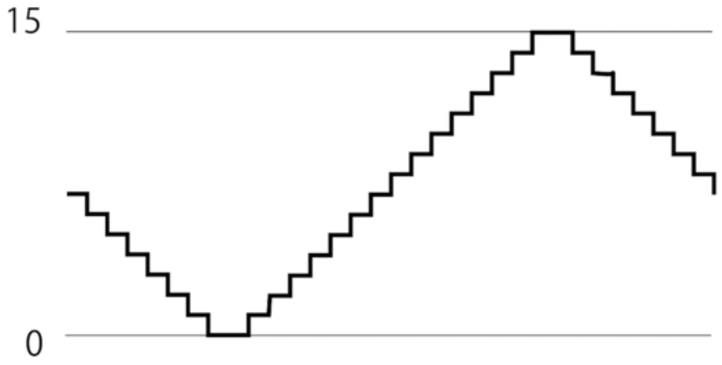
Oscillator

The waveshapes of the oscillator are obtained in the time domain by different approaches.
The main options are triangular wave, rectangular wave, and noise, generated by *TriangleVoice*, *PulseVoice* and *NoiseVoice* respectively.

An interesting wave is the ***triangular wave*** that returns the value of the signal in the time domain from an array containing 32 values.



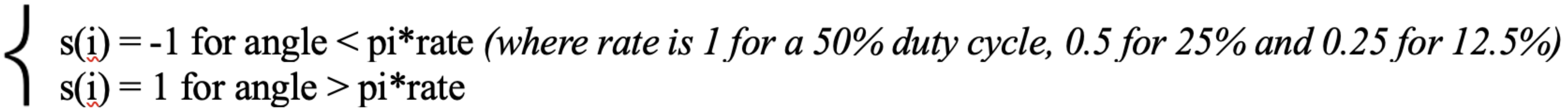
```
int sequence[32] = { 1,  2,  3,  4,  5,  6,  7,  8,
                     8,  7,  6,  5,  4,  3,  2,  1,
                     0, -1, -2, -3, -4, -5, -6, -7,
                    -7, -6, -5, -4, -3, -2, -1, 0
                   };
```



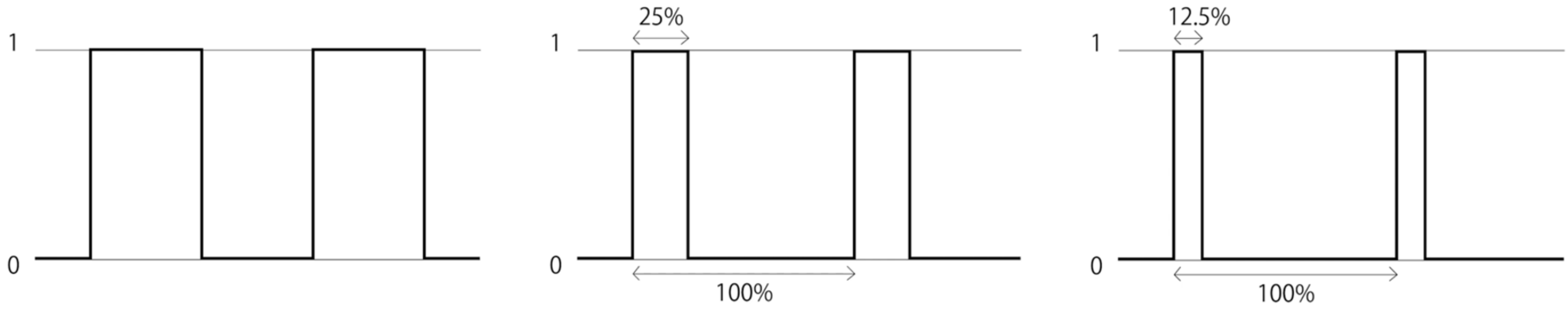
This means the wave is not a triangular one but it is a step-wise motion that recreates the 8-bit sounds.

Oscillator

The ***pulse wave*** has three different duty cycle parameters that change the duration of the high level with respect to the low level.



$$\begin{cases} s(i) = -1 \text{ for angle} < \text{pi*rate } \textit{(where rate is 1 for a 50\% duty cycle, 0.5 for 25\% and 0.25 for 12.5\%)} \\ s(i) = 1 \text{ for angle} > \text{pi*rate} \end{cases}$$

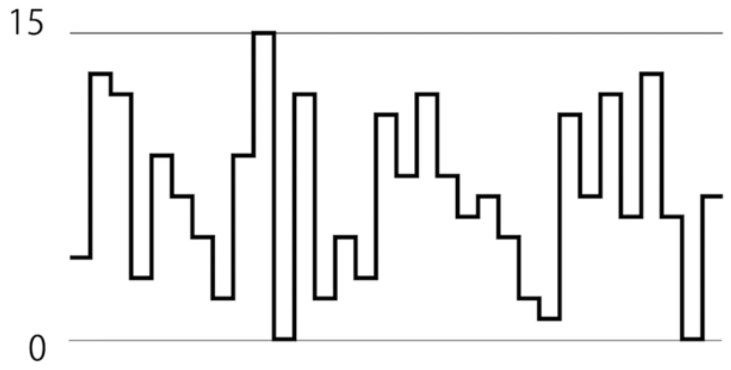


Oscillator

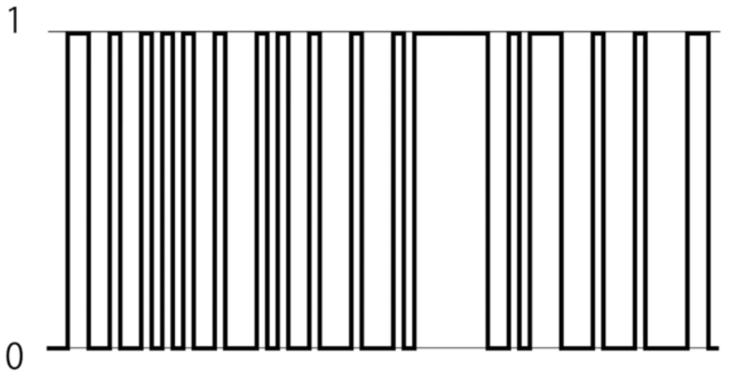
The ***noise*** has three possible configurations:

- a random 4-bit noise,
- a pseudo-random 1 bit short noise,
- a pseudo-random 1 bit long noise.

The *4-bit configuration* returns a random number between 0 and 15 at each update.



The *1-bit configurations* are meant to replicate exactly the pseudo-random nature of the old NES console, and it does so by working with a binary register that is constantly rotated. The difference between the long and short cycle is the length of the register, as the short cycle takes 1 value every 6 of the register.

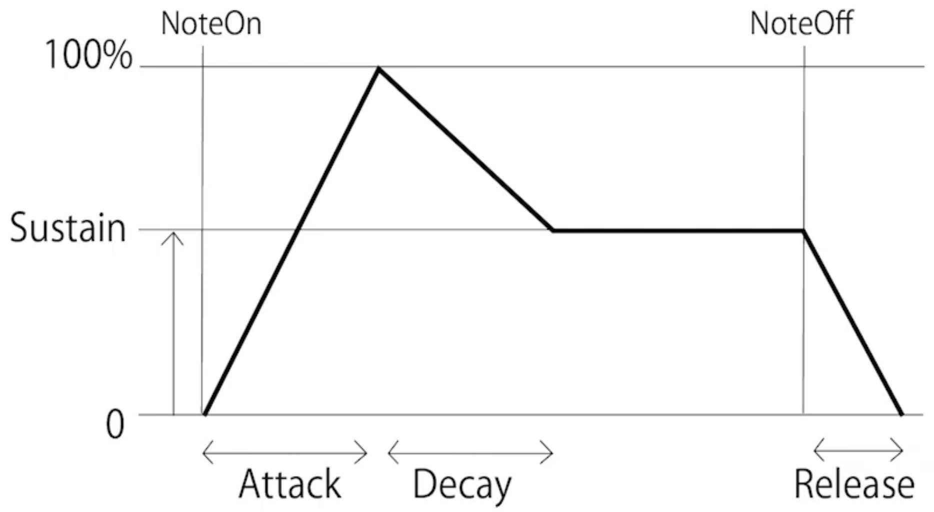


Polyphony

A Synth object is created inside the processor with an arbitrary number of voices limited by the maximum value of polyphony. Every voice has the same parameters. If the mono mode is selected some additional parameters are unlocked, specifically the possibility of playing legato and arpeggio.

Envelope

The **envelope** is implemented with an ADSR model.



By changing the *Attack*, *Decay*, *Sustain* and *Release* values the user can create different types of sounds. For example, applying them to a noise wave creates sounds similar to those of a drum.

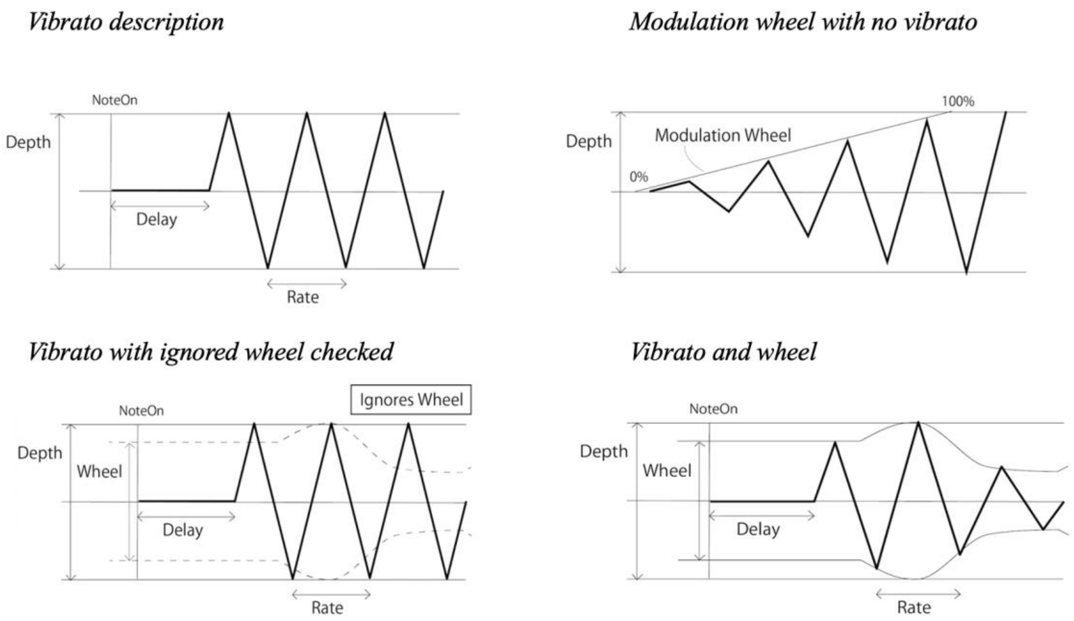
Bend Range and Auto Bend

The **Bend Range** parameter is responsible for the change in pitch of the signal. The bigger the Bend Range is, the greater the change in pitch.
It can also be applied **automatically**, and in this case the parameters the user can set are the *initial pitch* (how far away from the original pitch the pitch starts) and the *time* (how much it takes for the pitch to get to its original).

Vibrato

The parameters of the **Vibrato** are the *rate*, the *depth* (in semitone steps) and the *delay*.

An important parameter is the **Ignores Wheel**. If it is unchecked, the vibrato will respect the value of the modulation wheel. If it is checked, vibrato will be applied without taking into account the modulation wheel.

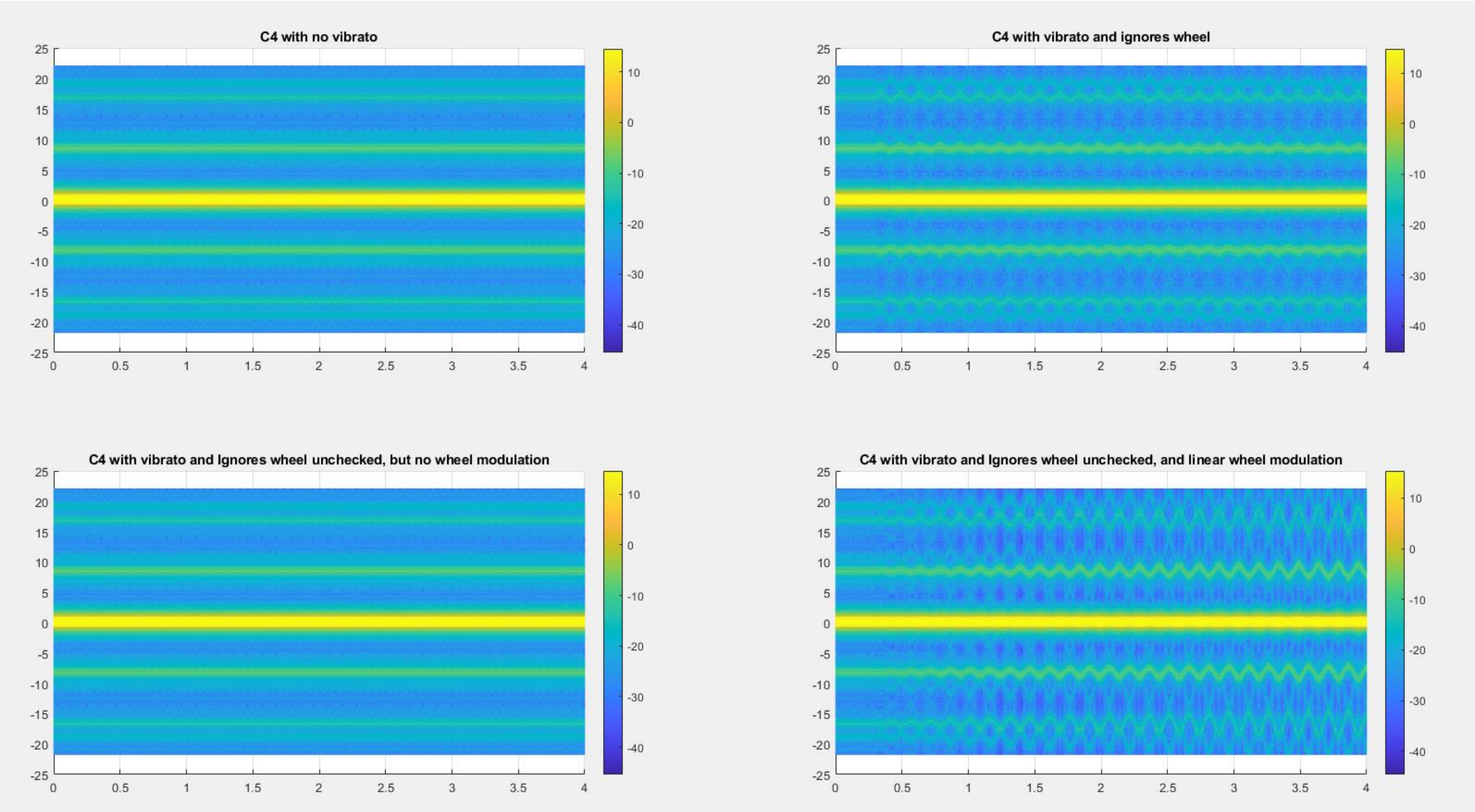

Vibrato description

Modulation wheel with no vibrato

Vibrato with ignored wheel checked

Vibrato and wheel

An example of Vibrato



Advanced options

The user has the possibility to specify custom numbers for volume and pitch envelope and duty.

The **Custom Envelope** disables the default ones and has some added features, as the possibility to keep a note for more than one frame, to release it in a specified interval of time and to repeat one or more notes.

An envelope that customs the **pitch** does not disable the default one but can coexist with the auto bend and vibrato features.

The **resolution** is another option that can be change and can be either *coarse* (in semitone units) or *fine* (in this case, the resolution is finer as 8 corresponds to a semitone).

Lastly, **duty** can have a custom envelope as well. The user can choose between 0, 1 or 2 to have a duty cycle of 12,5%, 25% or 50% respectively.

Demonstration Video