

CYCLONE To Do List

Issues with existing objects

1- [average~]

A second right outlet for audio output should be included to make it compatible to current Max behaviour.

2- [triangle~]

Bug: In Pd, if no argument is present, it loads a triangular wave (which means an argument of 0). Although that may be intuitive, given the object name, if no argument is present in Max, it considers it to be "0" and then loads the waveform according to that argument (a saw like waveform).

3- [lookup~]

Help File issues:

- An example of how to use the set message would be nice.
- [lookup~] is particularly convenient for waveshaping, the example could point that out too.

Bug:

- In Max, no maximum value bigger than the table size is permitted, so the values are always truncated.

4 - [sampstoms~] & [mstosamps~]

Bug: Both objects convert number input only to number outlet. It should also convert to signal output!

This is really annoying if you have a number box and would like it to convert to audio signal, cause you have to include a [sig~] object. Signal input only converts to signal output but that agrees with the behaviour in max.

5 - [delay~]

Updates: - In Max 5, [delay~] also accepts signal rate control via the right inlet. Signal-based delay uses interpolation, which introduces a one-sample delay like [vd~], so it behaves like [vd~] and we could use the code there as a basis for the update.

- In Max you can also reset the maximum size in the delay line via a "maxsize" message.

- A "ramp" message can set a time to ramp to a new delay line. This is like using a [line~] object to get to a new value.

6 - [buffir~]

Updates: - Currently, a maximum table size is 4096, not 256

- The message "clear" erases (zeroes) the current input history for the filter.

7 - [teeth~]

Make it into an external

8 - Nettles ([!-], [!/], [==~], [!=~], [<~], [<=~], [>~], [>=~], [!~], [!/~], [%~], [+~])

There's a single nettles.c file containing all the nettles objects (!-, !/, ==~, !=~, <~, <=~, >~, >=~, !~, !/~, %~, +~). You have to load the nettles file before Pd knows the object names. So it takes a lot of work to find these

(like using `[declare -lib nettles]`). One way would be simply to break the nettles file into individual objects.

9 - `[log~]`

`[log~]` now takes audio signal in the right in let, but Pd vanilla has a new `[log~]` object that takes audio inlet too, so this object is really redundant and should just be discarded/deleted.

10 - `[wave~]`

- Help file doesn't properly mention or show how to change the table name with "set" message.

- There are new interpolation modes, from Max 7 reference:

The interpolation modes are (Value - Description):

0 - No interpolation. Wavetable interpolation is disabled using the `interp 0` message.

1 - High-quality linear interpolation (default)

2 - Low-quality linear interpolation. This mode uses the interpolation method found in MSP 1.x versions of the `wave~` object. While this mode is faster than mode 1, it cannot play `buffer~` objects of arbitrary length and produces more interpolation artifacts.

3 - Cosine interpolation

4 - Cubic interpolation

5 - Spline interpolation

6 - Hermite interpolation

Possible values:

0 = 'None' (No interpolation)

1 = 'Linear'

2 = 'Low-Quality'

3 = 'Cosine' (Cosine interpolation)

4 = 'Cubic' (Cubic interpolation)

5 = 'Spline' (Spline interpolation)

6 = 'Hermite' (Hermite interpolation)

for Hermite Interpolation, other attribute messages are:

interp_bias [float]

Hermite interpolation bias

interp_tension [float]

Hermite interpolation tension

11 - [cycle~]

- [cycle~] uses now an internal 64-bit cosine wavetable that's 16K in size.
- The wavetable may be of any size (historically was limited to 512 samples).
- A "setall" message and a table name sets a [buffer~] object to use as the wavetable, using the entire buffer~ length.

From Max 7 reference:

Arguments

value [symbol]

Set a `buffer~` object to use as the wavetable, using the entire `buffer~` length. **This is in contrast to the `setmessage` which only uses the first 512 samples of the `buffer~`.**

- The "set" message:

Arguments

buffer-name [symbol]

sample-offset [int]

Set a `buffer~` object to use as the wavetable, using the first 512 samples. If no name is given, the internal cosine wavetable with 16K samples will be used. Use the `setall` message to use an entire `buffer~` as a wavetable.

- Cycle also takes attribute messages, which we could incorporate as extra arguments, such as

buffer_sizeinsamps Override the size of the buffer used for the wavetable. If using the `set` message the size will be 512, or if using the `setall` message the size will be the whole `buffer~` length. Use this attribute to define a different size (the size should be a power of two). Setting this size to 0 will result in an effective size of 512 and setting it to -1 will result an effective size of the whole `buffer~` length.

Possible values:

'-1'
'0'
'16'
'32'
'64'
'128'
'256'
'512'
'1024'
'2048'
'4096'
'8192'
'16384'
'32768'
'65536'

- Another attribute is phase offset, which would be REALLY COOL

phase [float]

Phase offset