Eito Murakami
November 2021

Pure Data EM Library

Overview:

I created a Pure Data library (temporarily called *EM Library*) that serves as both musical and educational tools for learning the efficient way of patching in Pd. Traditionally, most Pd libraries have aimed to provide objects that would eliminate the need to create complex patches. The *EM Library*, however, encourages users to study and modify the patches as every abstraction in the library is made entirely of the "vanilla" objects.

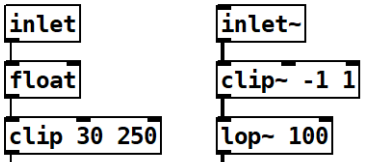
It is my hope that the library and this documentation serve as tutorials and templates for beginners and students while promoting community standards for patching to enhance communication between abstractions.

Patching Standards:

Pure Data offers users the freedom to patch objects in a variety of ways. Such flexibility, however, sometimes makes it difficult to create efficient and legible patches as in the case of other traditional programming languages. In the following section, I introduce patching standards that were used in the library and explain their significance.

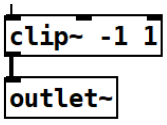
Inlet

1. The [inlet] (control value) object in a subpatch or abstraction is successively connected to the following objects:
    a. The data type object, such as [float] and [bang], to filter data types
    b. The [clip] object to limit the range of the incoming value
2. The [inlet~] (signal) object should be used whenever controlling audio-rate signals, such as a filter cutoff frequency. It is successively connected to the following objects:
    a. The [clip~] object to limit the range of the incoming signal
    b. The [lop~] object if the incoming signal needs to be smoothed out



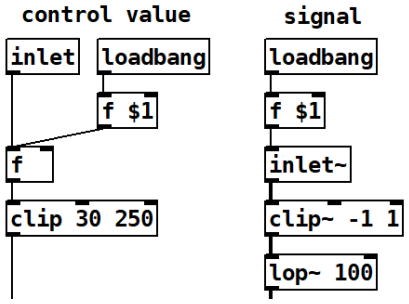
Outlet

1. If a signal is amplified inside a subpatch or abstraction, the [outlet~] object is connected to the [clip~] object to prevent any unwanted distortion.



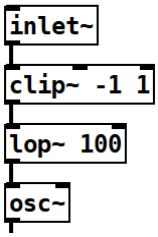
Argument

1. An argument for a control value or a signal that is shared with an inlet can be called by the [loadbang] object connected to the [float] (f) object with "$ + the argument index", such as $1.
    a. For a control value, the output of the [float] object is connected to another [float] object that is shared with the [inlet] object.
    b. For a signal, the output of the [float] object is connected directly to the [inlet~] object. The [inlet~] object will output the specified value until a signal is connected to the inlet.



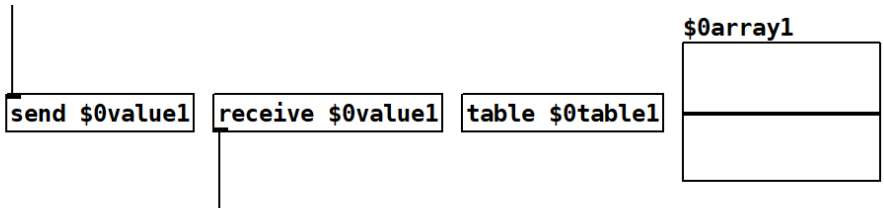
Continuity using [lop~]

1. In order to avoid wave discontinuities, a low-pass filter is applied to a control value or a signal that is controlling an audio-rate parameter. A cutoff frequency between 100 and 1000 Hz is used for the [lop~] object depending on how responsive a signal must be.



$0 for [send], [receive], arrays, and tables

1. In order for multiple instances of an abstraction to be loaded in a single patch, "$0" must be added to the names used in the [send] and [receive] objects as well as arrays and tables. "$0" is a four-digit number unique to an instance of abstraction.

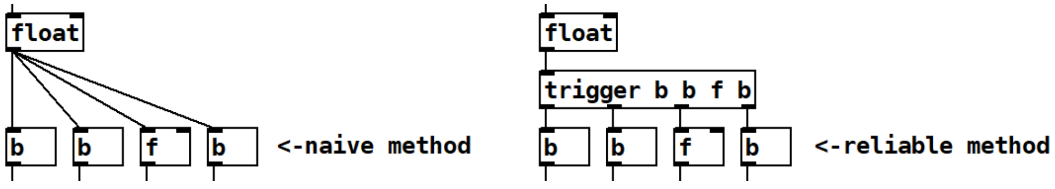


[snapshot~]

1. While the [snapshot~] object is useful for converting a signal to a control value, it should not be used to control an audio-rate signal because it only outputs the sampled value at the end of an audio block, which is 64 samples by default.

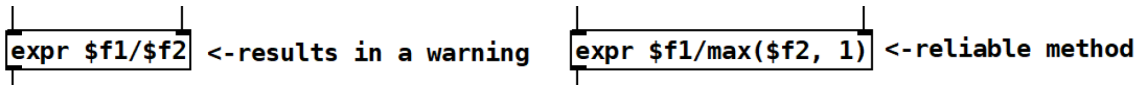
[trigger]

1. In order to maintain the order of operations, the [trigger] (t) object should be used instead of connecting an object directly to multiple objects.



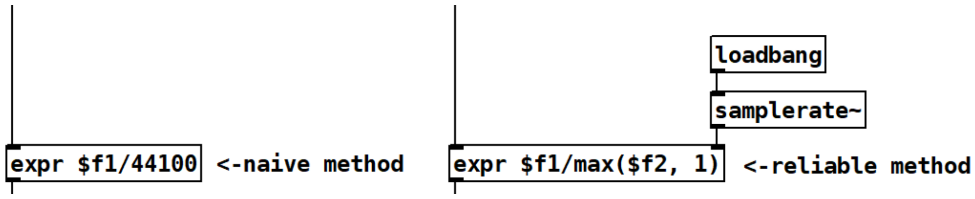
[expr] "expr divide by zero detected"

1. When using the [expr] object, dividing a variable by another variable results in a warning "expr divide by zero detected". While this is not a fatal error, it should be avoided by using the "max()" function such that a variable is always divided by a value other than zero.



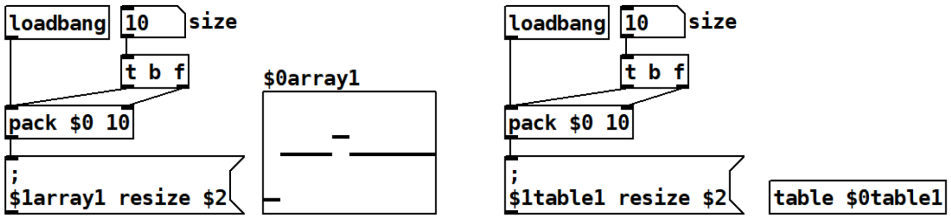
[samplerate~]

1. In order for patches to function properly with any sample rates, the [samplerate~] object is used for calculations involving the sample rate as opposed to hard coding the number, such as 44100.



Tables vs. Arrays

1. While arrays provide intuitive GUI, they take up high memory and CPU usage, especially if one chooses to "save contents". Tables are less expensive alternatives to arrays for saving a series of values without GUI. The size of arrays and tables can be resized using the "resize" message. As previously noted, names of arrays and tables should begin with "$0" so that multiple instances of an abstraction can be loaded in a single patch.



Miscellaneous

1. In order to make patches legible on any OS, objects should be spaced at least 10 pixels apart.
2. Every patch is accompanied by a description of its functionality as well as comments explaining the algorithm.

List of Objects

Audio Objects

- EM_bitcrusher~
- EM_pingpong~
- EM_width~
- EM_fold~
- EM_map~
- EM_switch~
- EM_drywet~
- EM_adsr~
- EM_balance~

Utility Objects

- EM_clock
- EM_map
- EM_loadnum
- EM_div
- EM_makegate
- EM_16seq
- EM_stom

GUI Objects

- EM_ktom
- EM_snapshot
- EM_sampler
- EM_granular