PHAUSTO: FAST AND ACCESSIBLE DSP PROGRAMMING IN PHARO

Domenico Cipriani
Pharo Association

Phausto is a library and API for ***Pharo*** that enables sound generation and audio *Digital Signal Processing programming* in ***Pharo***. ***Phausto*** leverages a dynamic library accessed via Foreign Function Interface (FFI) calls within Pharo. This library processes synthesizers and effects defined in Phausto with the help of an embedded ***FAUST*** compiler, which handles real-time audio computation.

Pharo a pure object-oriented programming language and a powerful environment, focused on simplicity and immediate feedback. It is distributed with a non-viral open-source MIT license.

Pharo is a state-of-the-art, modern, cross-platform implementation of the classic **Smalltak-80** programming language and runt time system

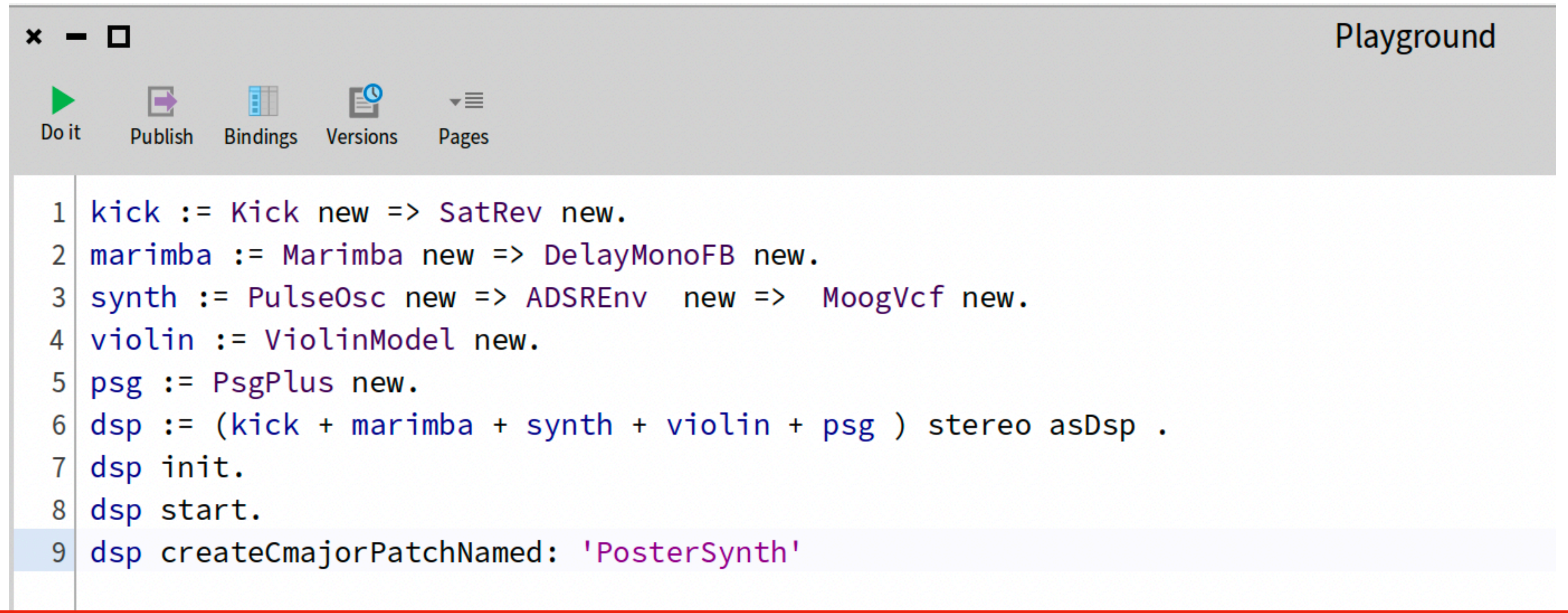
- *Easy to learn!*
- *Everything is an object!*
- *Only 6 reserved keywords!*
- *Platform-independent UI!*
- *Syntax fit on a postcard!*
- *Run-time reflection!*
- *Dynamic inheritance!*
- *Integrated Git support!*

Access a suite of cutting edge functions and generators from ***Faust*** standard library, including:

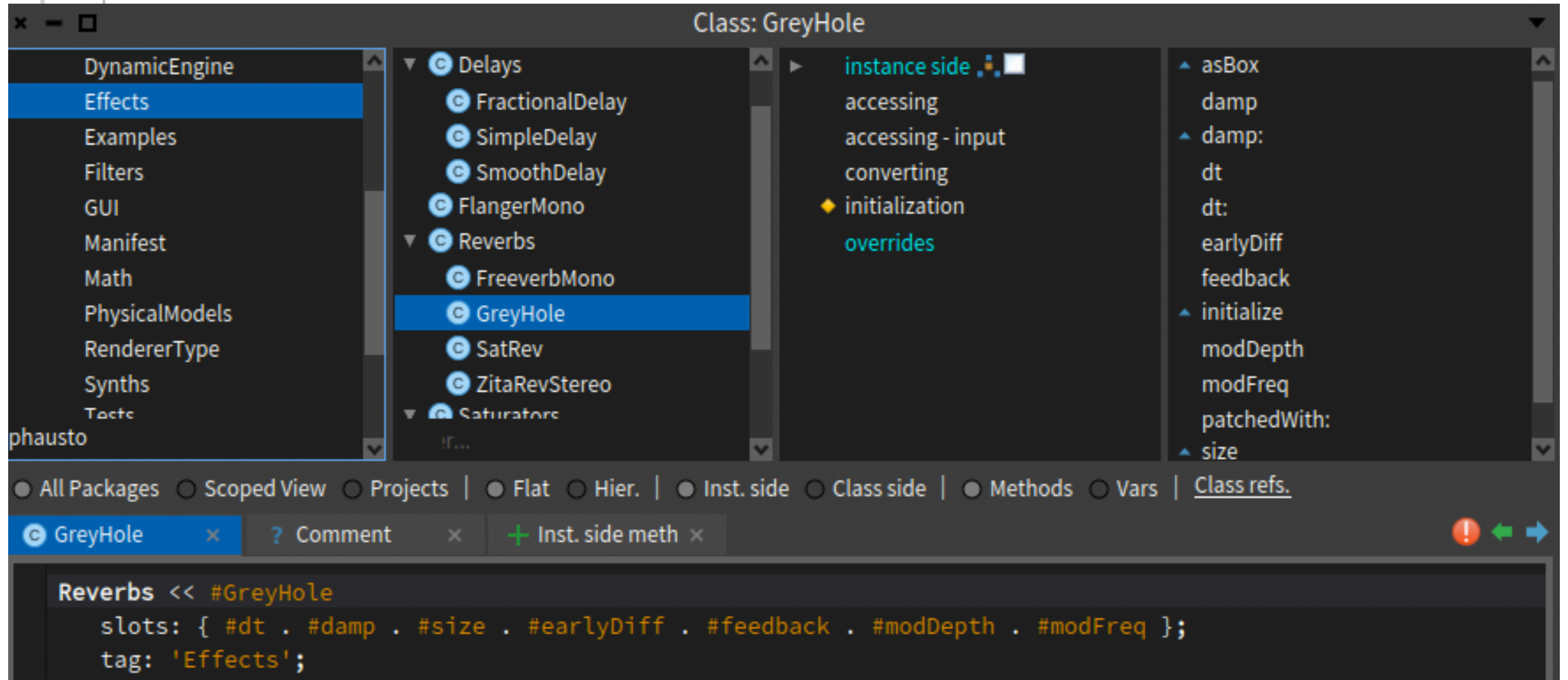
- Oscillators and Filters;
- Envelopes;
- Compressors,
- Delays and Reverbs;
- Phisical Models;
- Saturators;
- Noises;
- Flangers and phasers;
- And much more

Our grandstand audience

- Artists with little or no computer literacy who want to design and develop synthesisers and effects for their music creation fast and easily.
- Students and beginners who want develop their own audio plug-ins, by exporting DSPs developed in Pharo to a ***CMajor*** patch thanks to the *Faust-2CMajor* export.
- Pharo programmers who wants to include sounds or sonic interaction in their Pharo applications, including generators coming from **Faust** extensive physical modelling synthesis library.

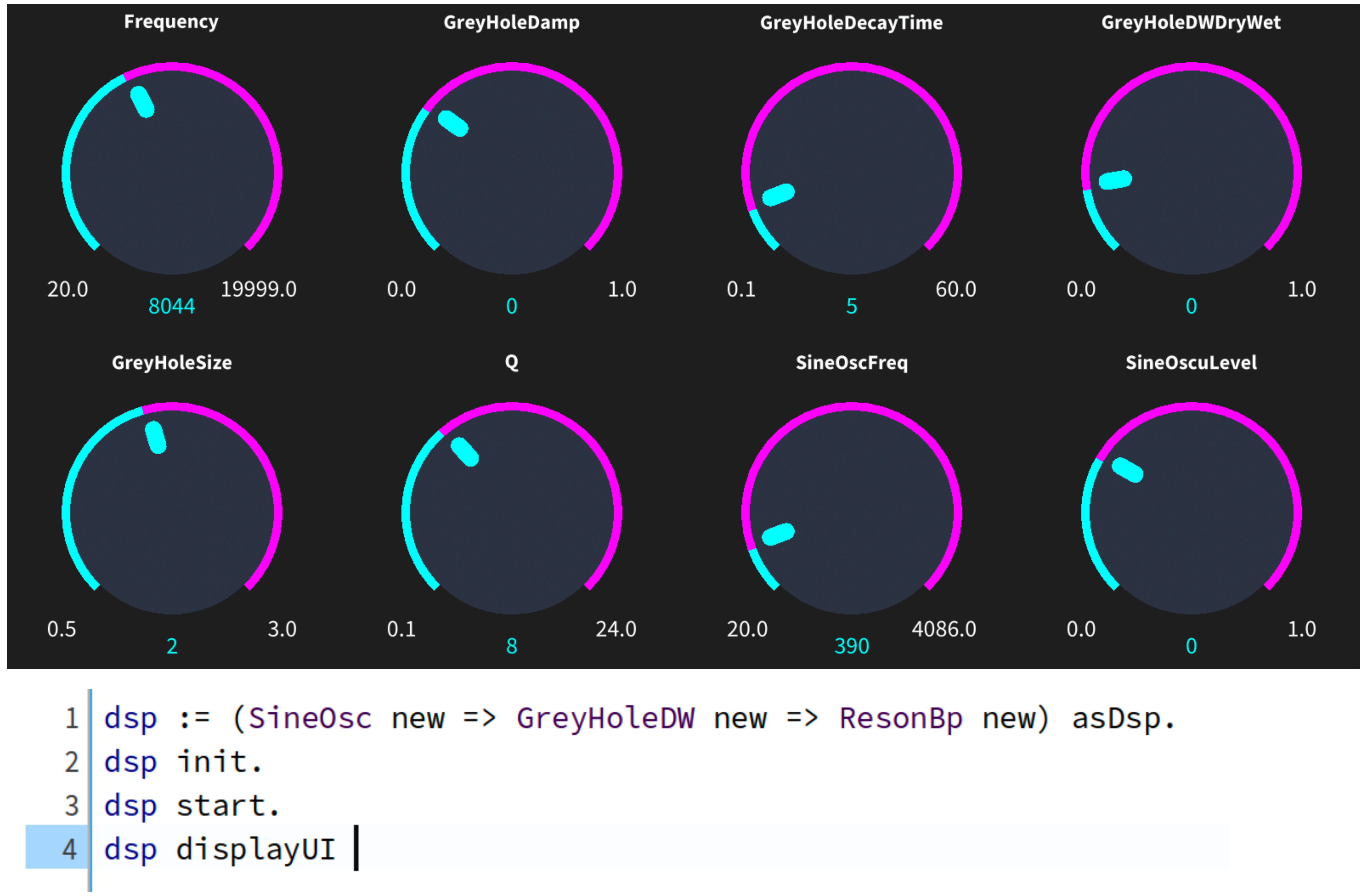


```
kick := Kick new => SatRev new.
marimba := Marimba new => DelayMonoFB new.
synth := PulseOsc new => ADSREnv  new =>  MoogVcf new.
violin := ViolinModel new.
psg := PsgPlus new.
dsp := (kick + marimba + synth + violin + psg ) stereo asDsp .
dsp init.
dsp start.
dsp createCmajorPatchNamed: 'PosterSynth'
```



```
Reverbs << #GreyHole
    slots: { #dt . #damp . #size . #earlyDiff . #feedback . #modDepth . #modFreq };
    tag: 'Effects';
```

Easy to display UI made with ***Bloc,*** a low-level UI infrastructure & framework for ***Pharo***.



```
dsp := (SineOsc new => GreyHoleDW new => ResonBp new) asDsp.
dsp init.
dsp start.
dsp displayUI
```

- Develop custom audio plugins through the **Cmajor** exporter. Once the Cmajor files are generated, use them in a DAW with the **Cmajor** VST plugin wrapper,.
- Export DSPa made in **Phausto** into C++ code suitable to build and run on **Bela**,