Getting Started with Purr Data

Albert Gräf <aggraef@gmail.com>

June 2018

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Agenda

- **auxiliary materials:** https://github.com/agraef/lac18.git
  (if you want to follow along; also make sure that you have GNU make, gcc, git, ALSA, Jack and Lua 5.3 installed)
- talk a bit about **Pd** and **Purr Data**, how the latter came about and where it's heading
- **install** a "light" version of Purr Data from source
- get **set up** and **start using** Purr Data
- **special highlights:** pd-lua and svg data structures

Miller Puckette's Pd a.k.a. Pure Data

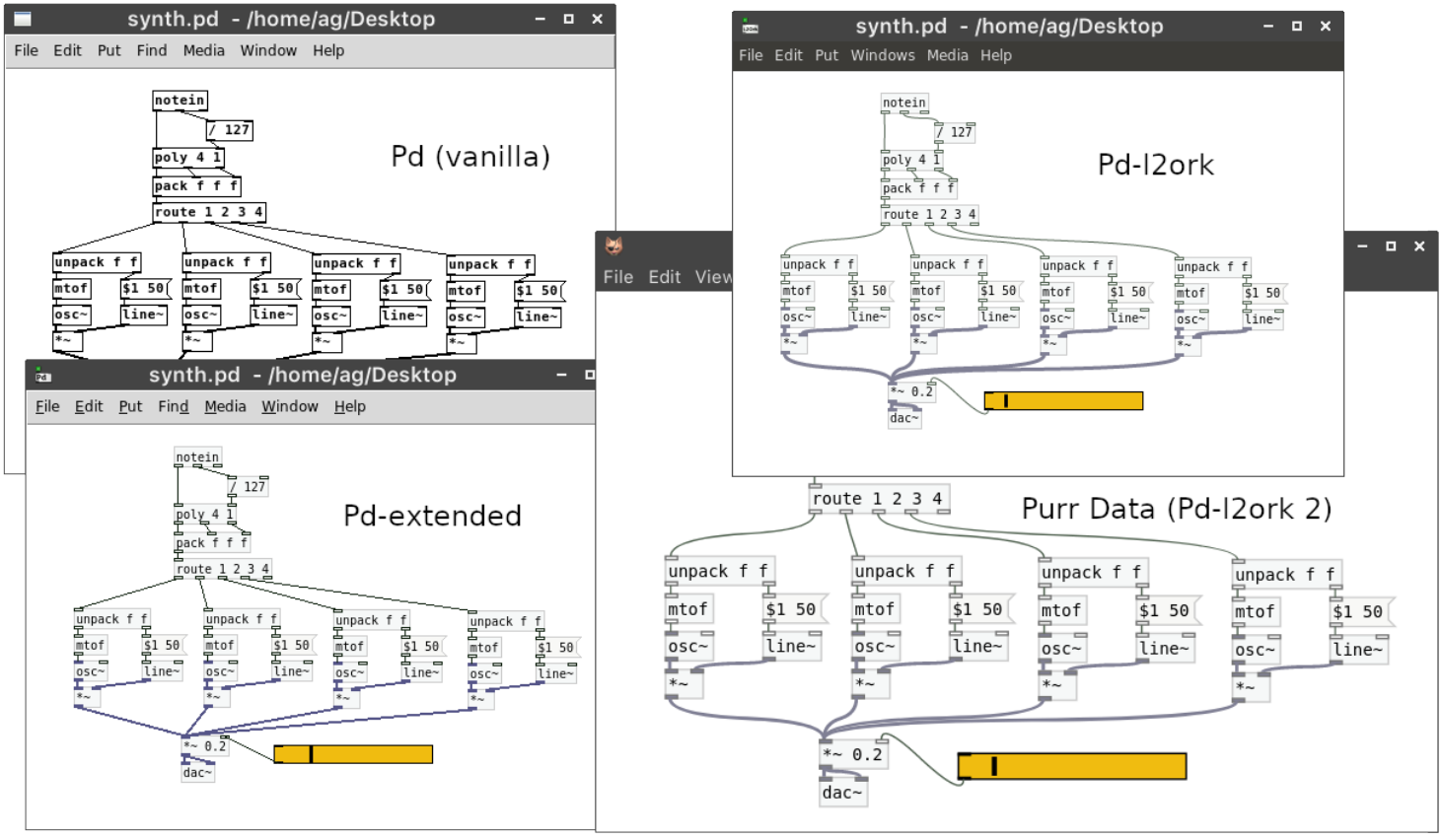
- *open-source* alternative to *Max*, the mother of graphical real-time computer music systems, descended from the MUSIC family of languages
- rock-solid *deterministic* real-time audio engine
- interfaces to *MIDI* and *OSC* and various other hardware
- video and 3D graphics supported through the *Gem* library
- runs on *Linux*, *Mac* and *Windows*
- community *flavors* and *distributions* (Pd-extended, Pd-l2ork)

Purr Data a.k.a. Pd-l2ork 2

- by Jonathan Wilkes (2015), cross-platform version of **Pd-l2ork** by Ico Bukvic (2010 – 2017), which in turn started out as a fork of **Pd-extended** by Hans-Christoph Steiner (2002 – 2013)
- **batteries included**, large bundled collection of popular externals, PDDP help patches
- **improved editing** features (infinite undo, visual editing, …)
- **better GUI**, now written in **JavaScript**
- new **help browser**, multi-level **zooming**, print to **pdf**, improved **SVG graphics** and data structures
- generally **compatible** (but *not* always bug-compatible) with **vanilla**

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

The Different Flavors



Why Purr Data?

- **cross-platform** (Pd is, but Pd-l2ork only ever ran on Linux)
- Tcl/Tk is getting **old**, programming in Tcl is easy but has its limitations
- switch to a better, more **modern GUI** as well as a **better language and toolkit** was in order
- solution: **nw.js** = node-webkit = JavaScript runtime (*NodeJS*) + browser engine (*Chromium*)

Latest Developments (since LAC 2017)

- improvements in the **build system**, toplevel Makefile
- **Pd-Lua** is now included to facilitate external programming
- lots of **bug fixes** and **performance improvements**
- continual **backports** of vanilla features (roughly compatible with Pd 0.48.0 at this point)
- work is underway to add **double precision support** (GSoC project by Pranay Gupta)

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Installation

- **Jonathan's Gitlab**: https://git.purrdata.net/jwilkes/purr-data
- **Github mirror**: https://agraef.github.io/purr-data/
- **Arch:** AUR or JGU repo, see https://l2orkaur.bitbucket.io/
- **Ubuntu:** JGU PPAs on Launchpad, see https://l2orkubuntu.bitbucket.io/
- **Mac**, **Windows**, **Linux** (Debian): https://github.com/agraef/purr-data/releases

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Installation from Source

- building the **full** package from source takes approx. 20 min (mostly Gem compilation), *lots* of dependencies required!
- therefore we only build the **light** (vanilla-like) version here (with pd-lua included)
- **prerequisites:** make sure that you have GNU make, gcc, git, ALSA, Jack and Lua 5.3 installed
- **clone** the repository:
  `git clone https://github.com/agraef/purr-data.git`
- **build:** `make light pdlua_ext`
- **install:** `sudo make install`
- **run:** launch Jack, then: `pd-l2ork -jack -lib pdlua`

Using Purr Data

- preferences
- help browser
- creating and editing patches
- audio and MIDI
- external programming with pd-lua
- SVG drawing with Pd data structures

Pd-Lua

- Lua (Portuguese *Moon*) is an interpreted, functional and object-oriented scripting language
- developed 1993 by **Roberto Ierusalimschy**, Luiz Henrique de Figueiredo and Waldemar Celes (PUC-Rio = Pontifícia Universidade Católica do Rio de Janeiro)
- light-weight, easy to embed, very popular in **game development**
- Pd-Lua 2008 by **Claude Heiland-Allen**, lets you write Pd objects in Lua, included in Purr Data as of version 2.5

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Lua Syntax

- **Numbers:** `3`, `345`, `0xa0`, `3.1416`; **Strings:** `"Hello, world!\n"`; **Variables:** `x`, `foo`, `Bar`; **Constants:** `true`, `false`
- **Expressions:** `5+7/2`, `x>2`, `"Hello " .. "world"`, `x[2]`, `z["bar"]`, `z.bar`, `#x`, `function(x) return 2*x end`
- **Tables:** `{}`, `{1, 2, 3}`, `{[1] = 1, [99] = 2}`, `{foo = 1, bar = 99}`
- **Assignment:** `x = 99`, `local x = 99`; **Multiple assignment:** `x, y = y, x`
- **Statements:** `x = 5+7/2; print("x =", x)`
- **Comments:** `-- this is a comment`

Lua Control Structures

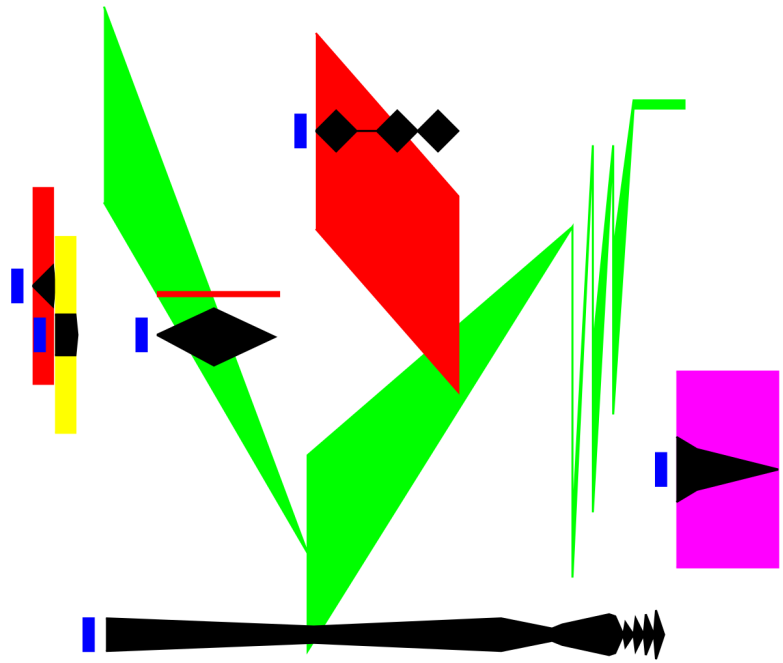
- **Functions:** `function foo(x) return 2*x end`
- **Lambdas:** `function(x) return 2*x end`
- **Conditionals:** `if x<0 then y = x elseif x>10 then y = 3*x else y = 2*x end`
- **Loops:** `while x>0 do y = y*x; x = x-1 end`, `repeat y = y*x; x = x+1 until x>10`, `for x = 1, n do y = y*x end`, `for k,v in pairs(t) do print(k, v) end`
- **Blocks:** `do local x, y, z; ... end`
- **Method calls:** `x:foo(...)` = `x.foo(x, ...)` where `x` is a Lua table (class-less, prototype-based OOP, like in Self and JavaScript)

Pd-Lua Example

```
local hello = pd.Class:new():register("hello")

function hello:initialize(sel, atoms)
   self.inlets = 1; self.outlets = 1; self.n = 0
   return true
end

function hello:in_1_bang()
   self.n = self.n+1
   self:outlet(1, "float", {self.n})
end

function hello:in_1_float(n)
   self.n = n
end
```

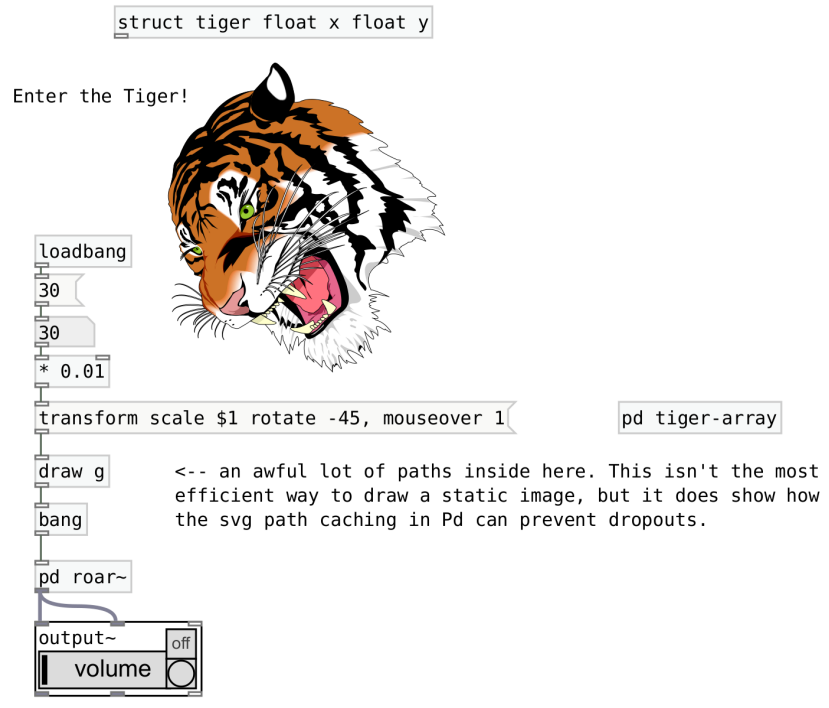
Data Structures

- **visualize data collections** using simple geometric shapes
- data can be *generated* and *edited*, *written* to and *read* from files, and *traversed* using "pointers"
- **typical application:** representing *scores* whose data elements can be *sequenced* in order to *synthesize* sounds



Data Structures in Purr Data

- Purr Data greatly *enhances* Pd's data structure visualization capabilities by introducing **SVG drawing commands** (`draw`)
- data elements can be created by simply **typing the template name**
- graphical attributes, transformations and mouse events are all **configurable** at run-time



JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Links

- **Purr Data website**: https://agraef.github.io/purr-data/
- **author's github page**: https://agraef.github.io/
- **materials** for this workshop: https://github.com/agraef/lac18.git

Other examples:

- **EZ-AG**: https://github.com/agraef/ez-ag (helper patch for the Yamaha's EZ and Jamstik MIDI guitars)
- **pd-jacktime**: https://github.com/agraef/pd-jacktime (pd-lua external to synchronize with Jack transport)
- **xwiimote-lua**: https://github.com/agraef/xwiimote-lua (pd-lua external to interface to the Wii Remote)

JOHANNES GUTENBERG UNIVERSITÄT MAINZ