

Create a synthesiser with a synthesis technique of your choice

1 Introduction

In this lab, you are asked to **create a synthesiser choosing or combining what you have learnt about synthesis techniques** (additive and subtractive methods, complex waveforms, filters and envelopes, and modulation techniques applied extensively).¹

Make sure that **the parameters of your synthesiser are controllable in realtime via GUI objects**.

Since our main interest is in the **sound synthesis quality** of your exercise, you are asked to **record either a short sound-composition (~1-2 minutes), or a brief catalog with three or four sounds that you deem representative of your synthesiser capabilities**, together with a short description of them. In the provided documentation, you should find “audio-rec.pd” objects that you can call inside your patch (abstractions, remember!?), to save your output as a soundfile to disc.

2 Provided documentation

We have added a few documentation patches covering the synthesis techniques reviewed and Pd’s basic objects. You don’t need to look at them, but be aware that they are there to help you!

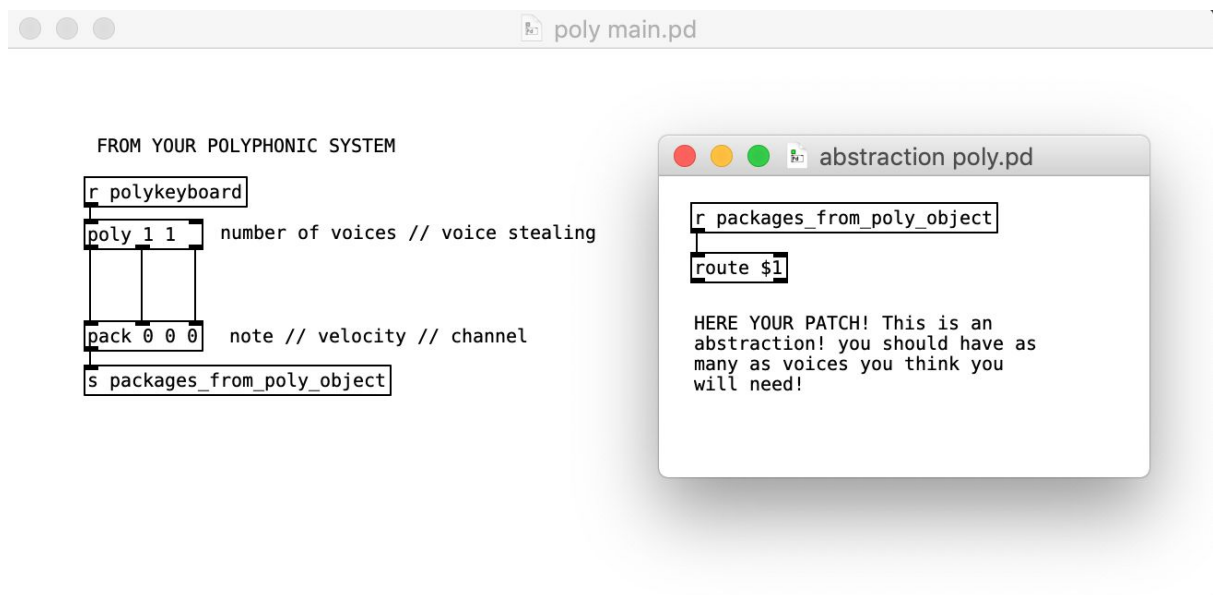
Besides, you can have a look at [Chapter 3 of Johannes Kreidler’s book](#), who addresses the implementation of various synthesis techniques in Pd, or look at [this tutorial](#) on “making your first synth”, introducing some of the core concepts of synthesising and processing audio in Pure Data. Those who are already familiar with audio synthesis should quickly grasp how it works in Pd, while those with no previous knowledge will be introduced to its theory alongside its practical application in Pd.

¹ For the moment, we are excluding any time-based effect (delays, flangers, etc.), live-input processing, wavefiles playback, spectral processing, and granular synthesis. However, feel free to explore these lines of sound synthesis if you feel comfortable enough!

3 Some recommendations and ideas

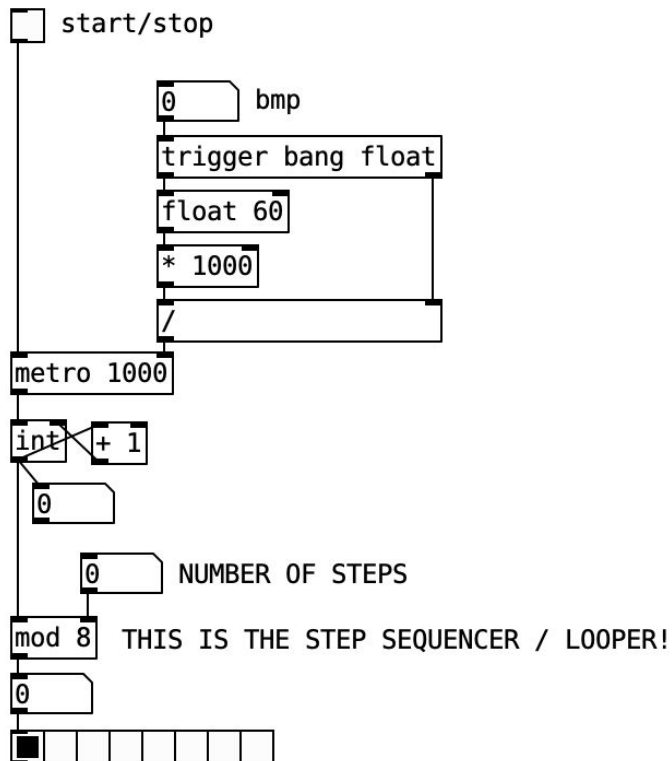
1. Provide an *as clean as possible* GUI without cables (use [send], [receive], [throw~] [catch~]).
2. Create as many subpatches as estimated convenient and reasonable, for making the code the clearest.
3. Generalise what you think will be reusable and encapsulate it in *abstractions*.
4. Add a master volume and an On/Off switch.
5. Create and scale GUI objects to suit your needs.
6. By default, your synth sounds will have no envelope, and sound will go from one pitch to the next without any silence or attack. Adding an “envelope” would allow to create single sounds **events** with attack at every new **trigger** (this could be useful for creating a bassline sequencer, since bass sounds typically have a sharp attack). Envelope objects include [creb/ead~] or [else/adsr~]. But remember that this is only a possibility. Eventless or drone-based approaches are also welcome!
7. If you think your instrument would benefit from “events”, consider adding QWERTY keyboard quantised pitch control.

Depending on the synthesis techniques of your choice, you might want to think about some additional features to add. For example, if you have a piano keyboard in mind, you might want to think about an strategy to make your synth **polyphonic** (it is basically the same idea as having multiple sine-waves synthesising a single tone. If you liked **synthesis by modulation** or **additive synthesis**, this might be your way to go:



Other type of synthesisers, such as subtractive bass-line generators (think of a TB303 as an inspiration), are typically **monophonic**, but they normally incorporate a **step-sequencer** to make pitch sequences and loop through them.

BASIC STEP SEQUENCER



what time of musical time does this represent?

In all cases, beware that **enveloping techniques** such as LFO's or ADSR's might be crucial to give liveliness to your sound, not only connected to your amplitude control, but to any other parameter, such as a filter frequency or bandwidth, the index of a modulation and the like.

Sounds in nature do not normally stay the same for a long time!

4 What to deliver?

A zip file including all your audio files and Pd patches (one or several) with extensive documentation in the patches (comments), and a Pdf describing your audio files.

Your opening patch should be clean and easy to use (no cables, simple GUI, parameters initialising to correct values, etc.). They should also be readable and understandable. This will be definitely taken into account in the evaluation.

Good synthesis!