ECS512 Sound Design
Lab 1: Introduction to Pure Data

TA: Will Wilkinson (w.j.wilkinson@qmul.ac.uk)

The aim of this lab is to introduce the basics of the graphical audio programming language Pure Data and put them to use by building a simple drum sequencer.

Use the following online tutorial as a reference:
`http://www.pd-tutorial.com/english/index.html`
Go through Section 2: "Programming with Pd for the first time" to familiarise yourself with the following concepts:

- Creating objects, message boxes and number boxes
- Inlets and outlets
- Edit mode
- The different types of connections (signal/control data)
- Creating and playing audio (`[osc~]` and `[dac~]`)
- Comments
- Arithmetic
- Hot/Cold Inlets
- Bang/Trigger
- `[moses]`, `[sel]`, `[==]` and `[random]`
- Float and Counters
- List - Pack/Unpack
- Message - `[set]`
- `[metro]` and `[delay]`
- `[line]`
- Send/Receive

When you feel comfortable programming in Pure Data, build a patch for a simple kick drum sound by implementing the following concepts:

- Use the [line~] object to create an envelope which controls the amplitude of an oscillator (set the oscillator to 100Hz, roughly the resonant frequency of a kick drum). The [line~] envelope should immediately jump to 1, and then decay to 0 over 150ms (note: use a *Message* box with a comma separating the instructions).

When a kick drum is struck, the skin deforms such that its resonant frequency during the attack is around 180Hz. As the skin settles, the resonant frequency drops down to around 80Hz.

- Use the output of the same [line~], such that when triggered, the frequency of the oscillator is set to 180Hz and then decreases to 80Hz over 150ms.

Now build a snare drum using similar concepts:

- Create a new envelope (again using [line~]). A snare drum decays faster than a kick, so set the decay time to 90ms.
- A snare drum deforms less than a kick due to the tighter skins, so keep the oscillator frequency fixed.

Since a snare has multiple skins, it contains two resonant frequencies. The complex interaction of the skins' vibrations also give the drum a noisey feel.

- Rather than one oscillator, sum together two oscillators (at 210Hz and 330Hz) and a noise term (using [noise~]).
- Apply a gain of 0.33 to ensure it doesn't get too loud (using [*~]).

Now put your kick and snare into two separate *subpatches*. Each one should contain an [inlet] which receives a [bang], and an [outlet~] which outputs the drum signal.

Build the sequencer shown in the image on the next page which continuously cycles between the numbers 0-7 and triggers the [bang] objects below. Make sure you go through and understand exactly how it works and what each object does.

- Connect the [bang] outputs to your subpatch inlets such that they sequence the drums in your desired way.
- Sum the outlets and apply a gain of 0.25 to reduce volume. Send to a [dac~].

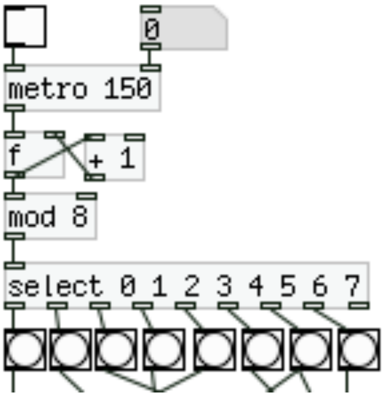

Figure 1: Sequencer