Algorithmic composition with probabilities and markov models

Ángel Faraldo

Introduction

In this lab you are asked to create some algorithmic process with which to partially control a synthesiser. This process should at least control the pitch of the synthesiser by means of “Note On” MIDI messages. Additionally, if you want, you can decide to control other parameters, such as the duration and the amplitude of each note event, or any synthesis parameter of your choice (for which using pre-determined probability distributions might be great!).

The synthesiser to be controlled could be (1) a generic MIDI synthesiser or (2) your own synth from a previous lab.

- If you choose option 1, your algorithms should be able to send NOTE ON and NOTE OFF messages (i.e. messages with velocity 0), since the latter will be needed for stopping notes in a polyphonic synth. Remember that this can be attained by using Pd’s [makenote] object.
- If you choose option 2 and your synth is monophonic, NOTE OFF messages might not be strictly needed, although in that case some option for producing silence (e.g. via amplitude control) could be helpful. In any case explain and briefly justify these and other choices, in the report accompanying the patches.

Some recommendations and ideas

- You should implement at least one Markov process to control pitch. This can be of order-0, order-1 or order-2.
- You are highly encouraged to control any other number of additional parameters, with additional markov processes or other probability distributions.
- You can use tables with [list-wrandom], [coll] or any other pre-compiled distribution (from maxlib, for example).
- Your transition probabilities can be (i) hardcoded, (ii) read from text files (to sonify some data, for example) or (iii) analysed from MIDI files (see the help for the midi-player object in the documentation, and the basshouse example to see how to analyse a whole corpus). You can also try (iv) live input via [notein] messages. In this last case you should document what the user is meant to do and expect.
- Adding some interaction can be a good idea to shape some sythesis parameters (or change their probabilistic behaviour in real time.

What and when to deliver

Please submit a zip file including all your Pd patches (one or several) with EXTENDED documentation and a brief DESCRIPTION of the intended design.

If your patches use MIDI files for probabilities analysis, you should also include at least one MIDI file of your choice. If it uses other resources such as text files also don't forget to include them. Your opening patch should be clean and easy to use (no cables, nice albeit simple GUI, parameters initializing to correct values [loadbang], etc.). They should also be readable and understandable! This will be definitely considered in the evaluation!

Please include some recordings of your patches running (whether an intended composition or catalog of sounds), as you did in the previous lab, although the setup needed for these recordings will be a bit trickier if you decide to use a generic MIDI synth (since the audio will not be produced in Pd). It can be a work in progress.

You can make teams of up to 3 fellow students to accomplish this lab, but please, do not upload duplicates and state very clearly your names in the attached documentation. Additionally, you can explain your approach to the team methodology.