Workshop 3

Patterns, Routines & Synths

Format

6 tasks are described that need implementation with scheduling, using patterns and routines.

Students shall work in groups to solve at least some of the tasks during class.

Solving all of them probably takes longer than two hours so students can choose those they nd interesting to solve.

The nal 30 minutes of class should involve a short presentation of each group where they show or discuss what they have done.

Tasks

Tasks, Patterns

1. Implement a sequence of two layers, both using brownian motion for pitch and duration values. One layer should stop before the other.
(hint: see Pbrown for brownian motion).

2. Implement a Process based on two or more layers where the pitches for each layer a created from a series going first up and then down until they repeat. The layers should not repeat at the same time.
(hint: see Ppar for parallel patterns and Pseries for series).

3. Implement a process where pitch values are determined either according to cauchy or exponential distribution. Dynamics should be determined with a geometric rise.
(hint: see Pcauchy, Pexprand and Pgeom).

Task, Synthesis

4. Create two simple SynthDef's that synthesize sound. The first one should use some sort of ***noise*** as its base sound. It is then fed through a ***filter*** and later an ***amplitude envelope***. The second one should use a rich ***oscillator*** (***square*** or ***saw***) that is also fed through a filter and envelope. The filters for both synths should me modulated by a ***sine wave***.

Finally play the synths using a ***pattern*** that plays first some events ***noise*** synth and then some of the ***square*** (or saw) ones.

The pattern should also alter the frequency of the filter modulation for each event.

Tasks, Routines

5. De ne a function f that will play a synth once. It should take as argument the pitch of a note. Then de ne a Routine that will call that function several times. Each time the pitch should be different as well as the time the routine waits between notes. *(hint: see routine class where both functions and routines are defined).*

6. Implement envelopes for duration and frequency values of a synth and a routines that schedules these in time. The amplitude of the synth should be set by an external variable that could be changed after the process is started. Once the envelopes have completed they should be repeated (looped) with slight variations.
(hint: use Env and env[i] as a form of indexing.)