Using open source music software to teach live electronics in pre-college music education

Hans Roels

hans.roels@hogent.be
1. Basic components of a live-electronics course

2. Is open source music software suited for such a course?

3. Demonstration of Abunch, a library in Pure Data
Bio

- Piano and Composition studies at the University College of Ghent - Faculty of Music
- Active as a professional composer
- Piano teacher (1995-2007) and teacher of practical harmony and accompaniment courses in pre-college music schools
- Concert producer in the Logos Foundation (2001-2008), a centre for audio arts
- Since 2008: Research (on contemporary polyphony) and teaching (electronic music) at the University College of Ghent - Faculty of Music
Special Project

- Since 2007: special project to teach live electronics in the music school of Deinze (B)
- Focus: performing live electronic music
- For amateur musicians older than 14 years
- A course of 2 hours / week
- In 2009 this special project was officially turned into a course 'experimental music'.
Unique opportunity

- Home computers are powerful enough
- Computers are widely available in households and schools
- Open source music software available for live electronic music

Rethink and redesign our music education!
Outline

1. Basic components of a live-electronics course

2. Is open source music software suited for such a course?

3. Demonstration of Abunch, a library in Pure Data
Is a Digital Musical Instrument unique?

- No acoustical link between user interface and sound production unit
- A multimodal information stream between composers and performers
- An important off-stage component
- Less difference between composer, improviser, performer and instrument-builder

= ?
Content of a live electronic course

- Digital Signal Processing techniques
- Basic audio hardware
- Mapping techniques
- History of electronic music
- Auditory training
- Sound organisation in real time
- Performance training
Mapping techniques

- Basic math
- Basic boolean operators
- Comparison operators
- Assignment operator
- Relay switch
- A module or system to order all this logic and math in time
Methodology

- Performance and action based method with integration of theory
- Auditory based (multimodal information stream, modular instrument)
- Creativity and autonomy (blurring boundaries between composing, performing and instrument-building)
1. Basic components of a live-electronics course

2. Is open source music software suited for such a course?

3. Demonstration of Abunch, a library in Pure Data
Pros and cons

Advantages

- Regular Access
- Mapping techniques are available and flexible
- Transparancy (source code)
- Strong User Community

Disadvantages

- Massive range of possibilities
- Simple (*and applied?*) information is difficult to find
- Too many syntax rules to be creative with instrument design
Outline

1. Basic components of a live-electronics course

2. Is open source music software suited for such a course?

3. Demonstration of Abunch, a library in Pure Data
A tool with an open-ended architecture that enables beginners to learn more about the musical possibilities in real time of a computer
Abunch - content

- performance objects
- demonstration and analysis objects
- information objects and files

\{ abstractions

- record and play sound files
- manipulate and process sound
- generate sounds
- prepare control data (sequencers)
- synchronize control data (clocks)
- analyse sound and control data
- record control data to a score
- receive data from common interfaces
- algorithmically generate control data
### All objects in Abunch (version 048)

#### AUDIO EFFECTS
- phaser
- simple-delay
- long-delay
- feed-delay
- reverb
- simple-chorus
- simple-pitchshift
- envelopes
- simple-filter
- 8band-filter
- lowhigh-filter
- disto-filter
- old-vocoder
- matrix4
- panning
- al-disto

#### SOUND SOURCES
- waves
- waves-add
- play-file
- record-file

#### PLAY SAMPLES
- play-sample
- simple-sampler
- record-sample
- scratch-sample
- crackle
- sound-out
- addy
- ozzy
- tarra
- crynth
- grain-sample
- grain48-sample
- graino-sample
- grain-synth
- grain-live
- wa-synth
- simple-fm
- 2mod-fm
- syna-fm
- cross-fm
- wind-synth

#### CONTROL
- timeline
- sequencer
- sequencer
- multi-seq
- multi-seqn
- keyboard-azerty
- keyboard-qwerty
- clock
- multi-clock
- random-out
- scales
- scales-div
- play-score
- record-score
- midi-ctlin
- metronome
- o-scope
- e-scope
- spectrum
- analyse

#### PRESETS
- presets
Simplified Procedures

- requires Pd Vanilla (any version >= 0.40)
- one folder
- one preset system
- reduced messages
- argument of an object relates to presets
- normalized control connections
- colour indicates audio or control object
- popup windows for common mistakes
Required knowledge

Required knowledge to start playing with Abunch:

- audio on/off
- procedure to create and open a new object
- flow from top to bottom
- audio and control connections and in- and outputs
- names of Abunch objects
- every Abunch object needs an unique number
- only one opened main file at the same time
Future

- Abunch development mostly dictated by short-term educational demands
- Unfinished
- New goal: own research and performances

To do:

- More example files (about the musical application of techniques)
- A more attractive and diverse layout (data structures?)
- A neat and uniform structure within each object
- A style guide for other developers
- An easy-to-use template for algorithms
Conclusions

- Abunch enables to use a performance and sound centered methodology
- Abunch enables newbies to be creative with the instrument design in a very early stage
- Students and pupils use Abunch at home to play

www.hansroels.be/abunch.htm