On the Improvement of Teaching Electronic Composition and Computer Music

A project to develop the open-source-software "Csound" for a better application in teaching

Joachim Heintz, Alex Hofmann

Incontri - Elektronisches Studio im Institut für neue Musik, HMT Hannover
www.incontri.hmt-hannover.de; jh@joachimheintz.de; alex@boomclicks.de

Version: February 2010

The electronic studio Incontri of the institute for new music of the HMT Hanover feels cordially attached to the open-source-movement. Lecturers and students alike can profit from the staightforward and legal distribution and update of high-quality open-source-software. The additional possibility to gain insight into the program code allows for a more in-depth analysis of the mode of operation than would be possible in the case of closed-software. The comprehension thus obtained can be used to further develop the functionality of the software or actively participate in the improvement of the software by improving the documentation. Typically, the mailing-list, forum and wiki offer manifold ways of contribution and involvement to students. Without spending a single cent, they can participate in a design that fits their needs: they can be a responsible part of a group of users and developers instead of consumer and customer of a (market-dependent) product. For these reasons the electronic studio as a public institution uses open-source-software whenever it is appropriate from an artistic and pedagogical point of view.

The "Csound"–software is one of the most widely acknowledged and long standing programs in the field of audio-programming. It was developed in the mid-80's at the Massachusetts Institute of Technology (MIT) by Barry Vercoe. Csound is a direct relative of the oldest computer-programs for sound synthesis (MusicN) by Max Mathews. It is distributed under the LPGL licence for free software and is tended and expanded by a core of developers and a wide community.

In 2009, Incontri started a project to make teaching and learning Csound easier. This should on the one hand improve the practical work with the students, while on the other hand facilitate the access to audio programming in the interest of the students. Furthermore the HMT Hanover has the chance to strengthen its profile in this sector internationally by participating in an open-source-project.

Previous work:

1. A benchmark analysis of Csound and other relevant audio-programming environments (SuperCollider, Pd, Max) was conducted. The results can be found in the next paragraph as a list of qualities and shortcomings of Csound.
2. The first step taken was to write a simple introduction. The electronic studio of the HMT Hanover composed a "Getting Started.." tutorial which was integrated into the official distribution of QuteCsound in collaboration with the developer Andres Cabrera. Two of three chapters of the tutorial have already been published in this way.

Result of previous work:

The benchmark analysis of Csound and SuperCollider, Pd and Max/Msp was conducted with respect to:

- the applicability to teaching in the areas "Basics of computermusic and acoustics", "Methods of sound synthesis and signal processing" and "Electronic composition"
- the state of the documentation
- the up-to-dateness of the syntax
- the application to composition and live-elektronics

Qualities:

1. Csound possesses examples reaching back far into the history of computer music, as well as a stringent backwards compatibility. As a result, the comprehensive "Sound Catalog" by Jean-Claude Risset from 1969 can be adapted to Csound with hardly any changes. Any composition in Csound from the last 20 years can today be executed immediately - a unique circumstance within the history of digital technology. This provides students with a near-to limitless choice of palpable examples of composition, sound synthesis and signal processing.

2. Csound's origins in the early times of computer music entails a syntax which is, in respect to signal processing, simple, obvious and easy to learn. Its program code expresses in a direct way that which is visualised in flow charts or graphical programming interfaces of Max/Msp or Pd.

3. Csound exhibits a generally appraised, exemplary sound quality as well as a wide spectrum of functions ("opcodes") in sound synthesis, signal processing and sound spatialization.

4. Csound features an easy and well-proven method to temporally specify events which is derived from the classical score-paradigm. Not only is this convenient for many practical applications, moreover it establishes a connection to composed music by enabling the composer to define the temporal course of musical parameters. Students are encouraged in developing a productive measure of design and realisation as well as reflecting on the temporal architecture of their ideas.

5. Csound separates the audio engine from the graphical user interface (GUI). This allows for alterations in the GUI at any time - in contrast to graphical programming interfaces like Pd. The current authoritative frontend "QuteCsound" was devised thus in 2008/09 on the basis of the up-to-date GUI-Toolkit Qt.

6. Csound is an international and open project. Developers and users are found on all continents. Support comes from academia as well as freelance musicians and coders. By becoming a part of the community, students can establish contacts to relevant academic institutions as well as the world of computer music beyond.

Shortcomings:

1. Generally speaking, the documentation of Csound is good. It lacks, however, some important aides for interested newbies. This is in part typical for open-source-projects: due to the years of continuous growth in the wealth of functions and changes in both the audio engine and GUI, a recent and simply structured beginners' tutorial is not available for QuteCsound. In contrast, the producers of commercial software are dependent on the attractiveness and accessibility to potential customers, which is why the manuals and help functions of this kind of software tend to be better and more up-to-date. Open-source-projects depend on support in order to attain a similar standard of user-friendliness and ease of acquisition.

2. 20 years of program history contain comprehensive study material. Yet classification, weighting and documentation would facilitate the access for beginners significantly.

3. The syntax of Csound is easy and obvious in respect to signal processing. It lacks more modern programming paradigms like object-orientation and interpreters and exhibits limits in the use of arrays and strings.
4. Csound contains functions for the application in live-elektronics (MIDI, OSC, Realtime-Audio, Live-Input), however these are not adequately integrated into neither the GUI nor syntax.
5. The new QuteCsound frontend lacks some commonly used functions like: playing multiple patches at the same time, updating running instruments, generating control devices from within the programming language, and visualisation of buffer contents.

Conclusion:

Alongside the many generally appraised qualities of Csound, which constitute its successful application in teaching and practice, there are some shortcomings: a great deficit was found in the structure of the documentation, which is indispensable in teaching. In open-souce-software, any kind of educational assistance is either the product of individual commitment or requires institutional backing.

Many important and useful functions of Csound have been created by generations of developers, but to find these treasure in the jungle of 1200 opcodes can be a tedious task. Here, the electronic studio can actively participate in improving the software as well as its teaching by developing paradigmatic models and examples.

Major Aims:

Csound should:

1. be easy to learn, by means of a clear introduction to QuteCsound (languages: English, German)
2. offer well documented examples of signal processing and sound synthesis
3. streamlined in terms of syntax
4. be supplemented with a text-book in German language

Task schedule:

1. Completion of the third chapter of the tutorial "Getting started" (March/April 2010)
2. Improvement of the documentation of the linear algebra opcodes and determination of areas of application (March 2010)

3. Establishment of contact to the educational science group of HMT in order to develop a didactically good text book. Discussion of target audience, scope and structure, as well as contact to publishing houses (April 2010)

4. Approach potential collaborators from external institutions (ICST Zurich, FH Hamburg, University of Hanover)

5. Update of Csounds.com Website (collaboration with Prof. Dr. Richard Boulanger, Berklee College of Music, Boston, USA)

6. Setup of a navigation structure in the Csound Manual for easy access to Opcodes

7. Development of examples for sound-editing and -synthesis as menu items in QuteCsound

8. Adaptation of Risset's "Sound Catalog" in cooperation with Jean-Claude Risset

9. Design of the feature "Live Event" in QuteCsound (since December 2009 in collaboration with Andres Cabrera, expected completion in June 2010)

10. Visualisation of buffer contents in Widgets (in cooperation with Andres Cabrera)

11. Optimisation of QuteCsound in terms of CPU efficiency (in cooperation with Andres Cabrera)

12. Parallel audio-engine for multiple tabs; implement live-instruments-updates (in cooperation with Andres Cabrera)

13. Csound Meeting 2011, Invitation of core developers' community to Hanover for concerts and workshops