MacMUSIC, the MUSIC N environment for Macintosh, algorithmic synthesis and composition made easy.

Isidoro Perez.
Labaratorio de informática y electrónica musical, LEM/CDMC.
MNCA Reina Sofia, c/ Santa Isabel 52, Madrid 28002
Tel.+34 1 4682310
isidoro.perez@ediris.cdmcm.es

Abstract

Under the starting rie of developing a graphic user-friendly interface to the well-known C-acoustics compilers MUSICIC and CSOUND, MacMUSIC is being developed like a standard Macintosh application to provide almost the facilities and power of these environments, with extremely easy handling. To design, manipulate and modify instruments and scores as intuitive as drawing them in a paper diagram, in which we can easily observe what the algorithm does.

1. What is MacMUSIC?

Until now the MUSIC N environments are shown as the composer like a programming environment, but MacMUSIC is a simple Macintosh application. So we don’t need a compiler, and we don’t need to know anything about computer languages. A basic experience with the Macintosh interface and of course with electronic music and algorithmic composition is recommended.

1.1 Power

The Macintosh versions of CSOUND and MUSICIC are being fully included in MacMUSIC. So we can generate scores, orchestras and synthesise the sound using both systems in our personal Macintosh. The program can also import/export the score and the orchestra as ASCII text files with the standard format, in order to compute the score in other machines.

MacMUSIC is being developed under THINK C using the object library and it is System 7 compatible. It will include the new system advantages like to be scriptable and to run in background.

1.2 Facilities

The most of facilities of the original environments are available in the program in some way.

You can experiment with the basic unit generators (oscillators, filters, envelope generators, sound files in/out, event generators, function generators, MIDI files in/out, etc...), with some complex synthesis techniques (FGF, phase vocoder, LPC, spectral analysis...), and perform any arithmetic operation and algorithmic control in both score and instrument definition.

In addition we are including new subroutines that generate scores for CSound, based on a algorithmic score definition, with a Csound syntex (the same as Csound generates samples based on a algorithmic-instruments definition).

1.3 Interface

Concepts involved in the MUSIC N are shown in the program windows in a clear way.

For example, an orchestra window can be viewed like the Finder desktop, instruments like folders and basic unit generator are shown like document icons, which we can open, edit and modify their behaviour.

All these icons have inputs and outputs, with an online information about its meaning and function, every output can be wired to any input, and the data types are standard or converted automatically.

MacMUSIC shows some floating windows: a tool palette (for drawing operations: selecting, dragging, writing) and several unit generator palettes organised by class and type.

There are no differences between the score and the orchestra user interfaces, but the concepts are different: orchestra correspond to score, instrument to part, unit generator to event generator, etc.

The same icons will have a different behaviour in each context, icons can be copied from the user to the orchestra and vice versa.

2. A simple example

Using libraries of algorithms the composer can satisfy its needs, then probably he don’t need to use a lot of low level unit generators.

In the following simple example, we show the user interface of an orchestra and score and we can see how easy it is to understand what an algorithm does.

ICMC proceedings 1994

Music Workstations
2.1 Orchestra design.

In figure 1, we have created an orchestra containing a plucked string instrument, a phase vocoder processor and a custom FM instrument. We can edit the last one (these icons can be opened recursively) and observe the designed algorithm. The text window is the .rec file generated.

2.2 Score composition.

There is not a strict distinction between signal level (used in the instruments) and event level (used in the score), the score is sometimes an extension of the instrument.

We can use the score in the traditional form. In figure 1, the score generate random notes (durations, pitches and dynamics) for the FM instrument. The left window show the FM part. Here, the maximum value achieved in the previous random events will limit the range of subsequent random values.

2.3 Sound generation.

Using the Cloudbang compiler we can observe the process of sound computation or compute the sound in background while we continue designing our document in MacMUSIC.

3. About future.

The graphical interface, standard documents and export/import commands will increase the diffusion and use of libraries of instruments and scores created with MacMUSIC.

The new generation of Macintosh based RISK computers, will offer more possibilities of real time work with MacMUSIC.

So we hope that this tool makes introduction and daily work with computer music easier, and contribute to its use and diffusion.

References

1. Music4C a MUSICA4B C-language version by Graeme Gerad from Faculty of Music University of Melbourne.
2. CSOUND by Barry Vercoe from Media Lab, M.I.T. (Macintosh version by Bill Gabbe).