SuperCollider: a new real time synthesis language

James McCartney
3403 Dalton St.
Austin, TX 78745 USA
james@clyde.as.utexas.edu, james@icsaudio.com

Abstract

SuperCollider is an environment for real time audio synthesis which runs on a Power Macintosh with no additional hardware. SuperCollider features: a built in programming language with real time incremental garbage collection, first class functions/closures, a small object oriented class system, a mini GUI builder for creating a patch control panel, a graphical interface for creating wave tables and breakpoint envelopes, MFI control, a large library of signal processing and synthesis functions, and a large library of functions for list processing of musical data. The user can write both the synthesis and compositional algorithms for their piece in the same high level language. This allows the creation of synthesis instruments with considerably more flexibility than allowed in lower level synthesis languages. Since it is easy to create control panels and graphic displays, SuperCollider is well suited as a tool for teaching various synthesis techniques.

1 Synthesis In a High Level Language

The idea behind SuperCollider was to provide audio synthesis in a high level language with dynamic typing, lists, garbage collection and closures. Normally a language of this type is too slow to do signal processing. By creating a signal data type which represents a buffer of samples over a short time frame, and defining operators for that data type, it is possible to amortize the cost of the interpreter over many samples. It was also necessary to come up with a real time garbage collector that never needed to halt processing. Fortunately, garbage collection techniques have advanced to the point that this is feasible (Wilson,1993). The real time garbage collection simplifies dynamic instantiation of synthesis processes. Thus, no static allocation for maximum numbers of voices is necessary, though a voice stealing algorithm is provided.

Most unit generators are implemented using closures. A closure is a function and its environment, both of which are packaged as a single value [Kogge,1991]. In the case of unit generators, the environment is the state such as oscillator phase, filter delay values, etc. To create a unit generator a function is called which creates a closure that encapsulates the unit generator's state. This closure, each time it is evaluated, will produce the next signal buffer of output for the unit generator. To create a synthesis function, the
user writes an expression which will be evaluated once each control period. A control period represents the time to play a single signal buffer. The user’s function will produce one sample buffer of output each control period.

Another benefit of synthesis is the ability to incorporate algorithmic compositional ideas into synthesis functions. This is especially good for granular synthesis where parameters need to be generated for many grains per second. The built in function library includes noisy functions for creating and manipulating lists of notes, durations or other values which can be used for algorithmic composition.

2 Audio I/O

SuperCollider can read and write audio in real time using the Macintosh sound manager and output ports or by streaming audio from and to a file. Streaming to a file can occur simultaneously with output to the DAC. Audio files are written in Sound Designer II format.

3 Graphic Control Panel Builder

Users can create their own graphic control panels for their program. An editor is provided that lets the user lay out controls in a control panel. The controls provided are sliders, range sliders, push buttons, radio buttons, check boxes, string displays, and graphics panels. These controls can be linked to call specific functions in the user program or can be polled by the program to get the current value.

4 Graphic Wave Table Editing

SuperCollider provides an easy means to graphically edit single cycle wave tables. Wave tables can be drawn, filled interactively with Fourier sine, Chabyshev polynomials, phase modulation waves, guin envelopes, gbez buzz pulses, and can be scaled, windowed, smoothed, etc. Pressing the space bar allows auditioning the table at any time. As editing window for breakpoint envelopes is also provided.

5 MIDI control

The sliders in the control panel can be assigned to MIDI controllers and controlled in real time during execution. Also, functions are provided for MIDI control within a user’s program. A Voice class is provided which implements a voice stealing algorithm that supports the sustain and sustainoff pedals.

6 Unit Generators

Many unit generators are provided, several of which I’ve not seen implemented elsewhere. Some examples of those are: a noise generator that uses parabolic interpolation which produces a smoother sounding noise than linear interpolation; a formant generator that uses a Hannig windowed hard sync sine oscillator; a parabolic envelope generator; wavetable+gap/pulse oscillators; and a function for gradual distortion. The synthesis functions have been multiply special cases to provide maximum performance whether the inputs are audio rate, control rate or even. For example, Aoscolla (an interpolating wavetable oscillator with low frequency modulation) is implemented 12 different ways depending on the inputs and most other functions are similarly optimized. Signals can switch between audio rate and control rate on the fly to maintain minimum possible CPU load.

7 Origin

SuperCollider began as two separate programs that I wrote. The first was a program called Synth-O-Matic which was a non real-time C-like synthesis programming language for the Macintosh written in 1990 and abandoned. The second was a MAX object called Pyrite which contained the interpreter for the language which was extended and used in SuperCollider. Writing SuperCollider involved integrating the language interpreter, garbage collector and function library of Pyrite with the synthesis engine and functions of Synth-O-Matic.

8 Acknowledgement

I’d like to thank Curtis Roads for encouraging me to revive the Synth-O-Matic program which ultimately led to SuperCollider.

9 Availability

SuperCollider is available from the author for a fee. Contact the author via email. A demo version is available via ftp from:

ftp://miror.apple.com/mirors/info-Mac Archive/gui/m supercollider-demo.hqx

References


McCartney

ICMC Proceedings 1996 258