

## MAT 240D - Digital Audio Programming: Sound Synthesis Techniques (Fall, 2009)

The MAT 240 course sequence is a six-part (two-year) practical programming course; it consists of hands-on software development devoted to digital audio and multimedia applications. Students read a selection of papers from the literature, with the emphasis on learning to use and extend the current state-of-the-art programming methods, tools, and programming interfaces. Class assignments involve C/C++/Java programming on Linux, Macintosh, MS-Windows, various plug-in APIs, and other platforms.

In MAT 240D, we will implement a variety of software sound synthesis techniques, starting from traditional additive, subtractive, to non-linear synthesis. Our focus, however, will be on more contemporary techniques such as physical models and granular synthesis. We will explore the internals of several existing synthesis packages and write our own versions of selected techniques in the CSL framework in C++ (<http://create.ucsb.edu/CSL>).

Students are expected to know the basics of digital audio signal representation and processing, and to be proficient in C, C++, or Java (scripting languages, Smalltalk, SuperCollider or LISP are a plus). Grading will be on the basis of in-class participation and solo/group programming projects.

### Course Outline

- Sound synthesis and processing background
- APIs and frameworks for digital audio synthesis
- Additive synthesis and the FFT
- Wavetable oscillators and optimizations
- Subtractive synthesis and dynamic filters
- Nonlinear techniques: FM and wave-shaping
- Sample-based synthesis, processing, granulation
- Physical models: waveguides and simulations
- Chaos and novel techniques
- Building applications and GUIs

### Instructor

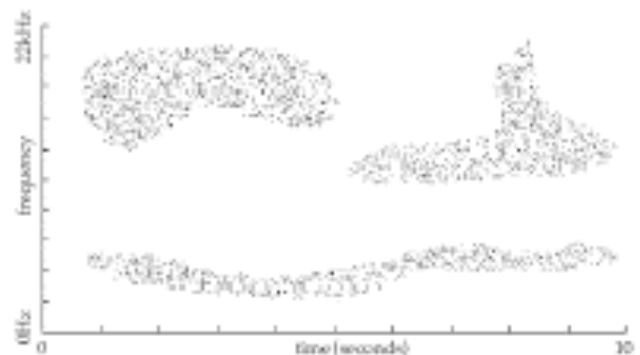
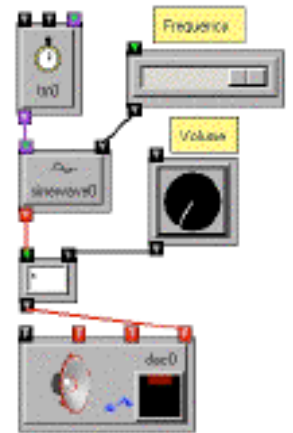
- Stephen T. Pope ([stp@mat.ucsb.edu](mailto:stp@mat.ucsb.edu))

### Meeting time and place

- T/Th 10:00 AM - 12:00 PM
- Music 2215 & South Hall 4340

### Electronic Resources

- Course Web Site & Email List  
See <http://mat.ucsb.edu/240>



# MAT 240D Reader Contents

Signal Processing Aspects of Computer Music: A Survey, J. A. Moorer (DASP book)  
Chapter 7: Physical Modeling and Formant Synthesis. Curtis Roads (CM Tutorial book)  
Granular Synthesis. Curtis Roads (Foundations book)  
Digital Waveguide Modeling of Musical Instruments J. O. Smith (www)  
Synthesis without Lookup Tables, James McCartney (CMJ)

SoftSynth Catalog pages, Full Compass Fall 2007 Catalog

Clear, Efficient Audio Signal Processing in ANSI C. Adrian Freed (www)  
JSyn Programmer's Guide. Phil Burk (www)

The CREATE Signal Library ("Sizzle"): Design, Issues, and Applications. Stephen Travis Pope and Chandrasekhar Ramakrishnan (ICMC 2003)  
Metamodels and Design Patterns in CSL4, Stephen Travis Pope, Xavier Amatriain, Lance Putnam, Jorge Castellanos, and Ryan Avery (ICMC 2006)

Wavetable Synthesis 101: A Fundamental Perspective, Robert Bristow-Johnson  
Effect Design: Part 3: Oscillators, Jon Dattorro (JAES)  
Physically-Based Parameteric Sound Synthesis and Control, Perry Cook  
CSE393: Computer Music & Sound Synthesis, Tom Cortina, SUNYB

## Optional Readings

Granular Synthesis of Musical Signals. Sergio Cavaliere and Aldo Piccialli (MSP book)  
PulsarGenerator Documentation. Curtis Roads and Alberto de Campo (www)  
The Computer Music Tutorial. Curtis Roads.  
The CSound Book. R. Boulanger  
Three Languages for Software Sound synthesis, S. T. Pope (CMJ)

# MAT 240D Code Archive

- 2005\_students ▶
- 2007\_Students ▶
- 2007Code ▶
- amber-1.1.2 ▶
- ASIOSDK2 ▶
- audiofile-0.2.6 ▶
- CAST ▶
- CLAM-0.8.0 ▶
- clm-3 ▶
- CO\_1.0 ▶
- it++3.8.1 ▶
- jmusic\_1.5 ▶
- JSampler-0.1a ▶
- jsyn142\_mac\_osx\_sdk ▶
- libgig-2.0.1 ▶
- linuxsampler-0.3.3 ▶
- loris-1.2.0 ▶
- MouseToOSC-1.1 ▶
- mxv-LATEST-source ▶
- portaudio\_18b ▶
- qsampler-0.1.2 ▶
- RB-GST-07-SEP-2001 ▶
- SamplePlayer ▶
- Samples ▶
- stk-4.3.0 ▶
- String0.2 ▶
- StudentCode ▶
- tao-1.0-beta-10May2006 ▶
- VirtualPiano\_src Folder ▶
- wire10demo\_mac\_osx ▶
- 240D\_Tests.cpp
- csndflow.gif
- js-classic.jar
- JSampler.log
- rhythm\_test.c
- SamplePlayer.zip
- saw\_adsr.c
- sharc.tar.gz
- stk\_granular.cpp
- VST\_Hosts.zip