

Java Music Systems, Fall 2016
E85_2608, New York University
Nick Didkovsky, nick@didkovsky.com

Websites:

<http://www.algomusic.com> (JMSL – Java Music Specification Language)

<http://www.softsynth.com> (JSyn – Java music synthesis API)

Lecture notes will be posted at <http://www.didkovsky.com/JavaMusicSystems>

Books (both available from Amazon)

Computer Music. Synthesis, Composition, Performance, by Charles Dodge and Thomas Jerse.
Schirmer Books, ISBN 0-02-873100-X

Meta-Hodos and Meta Meta-Hodos : A Phenomenology of 20th Century Musical Materials and an Approach to the Study of Form, by James Tenney, Frog Peak Music, ISBN: 0945996004

*You also need a **book on Java programming**. You may find one of your own or choose one of the following. The best thing to do is go to Borders or Barnes and Noble and browse the Java books. You'll find one that matches your level best that way!*

Thinking in Java, by Bruce Eckel

Prentice Hall, ISBN 0-13-659723-8

Full text, updates, and code available at <http://www.bruceekel.com>

For the intermediate user, or Java beginner with some programming background.

Java 2 for Dummies, or Beginning Programming with Java For Dummies both by Barry Burd

While I cannot personally recommend these books, a former student of mine does. Careful to distinguish this book from Java for Dummies by Aaron Walsh which got miserable reviews!

Java Programming: From the Beginning

by K. N. King

Has received excellent reviews. Assumes no prior programming knowledge. Expensive though!

Grading

• Homework assignments (should be uploaded to student's dropbox unless otherwise notified) 40%

• Final project 60%

IMPORTANT: Final Project must have an **accompanying paper** submitted in **PDF format!** *The due date of the paper is the same as the project presentation date, which is the last day of class.*

Your **final project papers** should follow this format:

- Abstract - 3 or 4 sentences describing your project
- Introduction - an expanded, more narrative restatement of your abstract which also includes your motivations and intentions. If your work is online, include the URL here.
- Detailed description - here you dig into the non-technical perspective on your project. What the user does, how the piece reacts if it is interactive, for example. Include screen dumps of your various GUIs or whatever else is graphically included in your project (Syntona patches for example)
- Technical description - without resorting to a bunch of Java source code, discuss your algorithms and structures. For example, "Every time it repeats, the MusicShape playing the main theme modifies itself by adding new elements, then time compressing all its durations so the overall duration is 8 seconds. I created a Playable to accomplish this task. The Playable was added to the MusicShape's repeat playables. The algorithm chooses a random integer between 1 and 8 and adds that many elements to the MusicShape. Durations are chosen from a set of durations (0.5, 0.25, 0.125), pitches are chosen using a 1/F generator, scaling it to the range of bass clarinet. It uses a two-pass algorithm to time compress. On the first pass, it sums the durations in the MusicShape to get the total. Then it computes a scaler and scales the duration of each element so that the sum remains 8 sec". So you see here that this is a technical description without needed source code. If you need source code, include it.
- Artistic analysis - here's where you write a critique of your work. Did it do what you wanted it to? Why, why not? What directions does it point to for the future? What surprised you? For example, "I was puzzled why pitches tended to cluster around the average pitch of the theme after a while. Upon further inspection I discovered that my melody modifier behaved like a band pass filter centered on the mean. I loved this / I hated this. etc"
- conclusion - wrap it up and point to the future. Include a discussion of the role the Java music technologies played in realizing your idea.
- bibliography and sources

Sep

- 6 Introduction to JMSL and JSyn
Demonstrations, examples, history, and overview. Syntona demo.
Get Syntona at <http://www.softsynth.com/syntona/>
Get Eclipse at <http://www.eclipse.org>
- 13 Introduction to Java
Java Fundamentals, designing GUI's and handling user events
Homework 1: Syntona UnitVoices.
Extra Credit: event handling applet
Extra Credit: note to pitch applet.
- 20 More Java
Designing GUI's and handling user events, hard disk organization
Extra Credit: random drawing
- 27 Introduction to JSyn
Overview of unit generators, circuits, connecting units, getting sound out.
Homework 1 DUE
Homework 2: Design a JSyn UnitVoice in Syntona (or by hand), export as Java Source code, deploy in sound tester applet.
Extra Credit: Design FM pair with hardwired Fc:Fm and ModIndex ports, deploy in applet.

Oct

- 4 More JSyn
Envelopes, using JSyn's event buffer to schedule events
Extra Credit Homework: Polytimbral polyrhythms
- 11 Introduction to JMSL
Hierarchies, and scheduling, using JSyn and JMSL together
Homework 2 DUE
Homework 3: Algorithmic melody using MusicShape and JSyn UnitVoice
- 18 JMSL sonification of musical data
MusicShape, Instruments and Interpreters
Extra Credit Homework: Design a MandelMusic instrument
- 25 JMSL Score, part 1
JMSL's Music Notation package
Designing JSyn UnitVoices that can be imported into JMSL Score
Designing algorithmic transformations of user-selected notes.
How JMSL's plug-in api works and how to use it.
Homework 3 DUE
Homework 4: Compose a piece in JScore either algorithmically or by hand, with custom UnitVoice.

Nov

- 1 JMSL Score, part 2
JMSL's Music Notation package
Notating algorithmically generated music using addNote() and using the transcriber
Notating and routing signal processing synth circuits in a score
- 8 JMSL Real-time performance *Final Project Proposals Due*
Players, Midi Output, Midi Input, making decisions based on real-time performance data.
Extra Credit: Design a SynthNote and control it via MIDI input
Homework 4 DUE
- 15 More JSyn Instrument design
WaveShaping, Chebyshev Polynomials, playing a sound file, waveshaping a sound file.
- 22 MaxScore: music notation in Max/MSP, and topics in network music
Useful algorithms, mapping mathematical processes to musical events, organizing music algorithmically, recursion, using TCP/IP to create networked JMSL/JSyn pieces, intro to Transjam (Phil Burk's general purpose server)

29 Special Topics

Dec

6 Final Project Presentations

13 *Legislative Day – we do not meet. Monday classes run instead of Tuesday classes*