S134 Syllabus (Provisional, S25)

Syllabus Revision 2.030225: this syllabus is subject to change

Meeting Schedule: MW 1-4:15 pm, Monday 6/30 to Wednesday 07/30

Cross Listing: MUSI S426

Instructor: Scott Petersen

Scott's Office Hours: Please see below. Extra hours will be added as needed.

Please use the below link to sign up for office hour slots. These are in 15-20 minute increments depending on the day. Please sign up as early as you can. Last-minute sign-ups may not be honored, as I may have devoted the time to another task.

Please book office hours here: https://calendar.app.google/BZ9tf3UJFyju8W967

Office Location: Dunham Lab Room 425

In general, my Monday afternoon OH will be in-person, my Thursday morning OH will be via Zoom. **It's best to confirm beforehand**.

Office Hours Zoom Link (For Zoom-confirmed meetings):

Join from PC, Mac, Linux, iOS or Android: https://yale.zoom.us/j/3669636696
Meeting ID: 366 963 6696
International numbers available:
https://yale.zoom.us/u/acq2i254nm

Prerequisites

There are no course prerequisites for this class, however coding experience in any programming language and knowledge of basic music theory will be extremely helpful. The ability to play an instrument, even by ear, is also valuable.

Course Description

This YSS course is a five-week intensive exploration of topics related to computer music: programming, analysis, composition, machine listening, physical computing, and more. Throughout the semester, you will be challenged to expand your understanding of sound and music and encouraged to experiment with new forms of musical expression. To that end, you will simultaneously learn an audio programming language that will allow you to realize your musical ideas in code and be exposed to many others for future applications.

Module Descriptions and Weekly Schedule of Assignments

Module 1: Sound Synthesis

Reading: Holmes 492-500; Petersen, S*ynthesis Methods in SuperCollider* n-nn; Ruviaro 1-9;

Theory

- Digital representations of sound and music
- Sampling and Sound Synthesis

Praxis

- A tour of Digital Music/Audio Programs
- SuperCollider as programming language

Week 1 — Daily

Monday June 30: Quiz 1

Wednesday July 2:

Friday July 4: Problem set 1 due

Module 2: Digital Signal Processing, Virtual Instruments

Reading: Roads (1996) 397-407; Ruviaro 11-27

Theory

- Digital Signal Processing
- Virtual Instruments

Praxis

- Audio Filters
- Sound spatialization

Week 2 — Daily

Monday July 7: Quiz 2

Wednesday July 9:

Friday July 11: Problem set 2 due

Module 3: Computational Creativity

Reading: Hiller 16-17, 82-93; Ruviaro 28-40

Theory

- Introduction to rule-based composition

Praxis

- Computational Creativity
- Indeterminate musical systems

Week 3 — Daily

Monday July 14: Quiz 3

Wednesday July 16: Problem set 3 due

Sunday July 20: Project 1 due

- SuperCollider as compositional assistant
- Scheduling Events, Use of Patterns

Module 4: Interactive Systems

Reading: Petersen, *Interaction in SuperCollider* n-nn, Wilson 439-461

Theory

- Machine Listening
- Interactive Musical Systems (HCI), live coding and electroc-acoustic music

Praxis

- Real-time audio analysis and processing
- Buses and meta program control structures in SuperCollider
- SuperCollider classes for audio analysis and MIDI interaction

Week 4 — Daily

Monday July 21: Quiz 4

Wednesday July 23: Problem set 4

Sunday July 27: Project 2 due

Module 5: Embedded Audio Applications

Reading: Petersen, *Data Sonification* n-nn;

Theory

- Embedded systems: sensing the real world
- Sonification (physical computing with Arduino and sensors)

Praxis

- Overview of microcontrollers and audio-specific embedded systems
- SuperCollider Classes for data acquisition
- SuperCollider & Faust on the Bela Platform

Week 5 — Daily

Monday July 28:

Wednesday July 30:

Friday Aug 1: Project 3 due

Principle Readings

It is imperative that readings are done *prior* to the start of the module in which they are listed. Given the extremely condensed nature of summer session, class time will be maximized when materials are presented in class that have already been introduced in the reading.

Readings will be assigned and made available through the course Notion, attached to the lecture notes/handouts to which they pertain. This will be explained during the first class meeting. Readings will pertain to the subjects above under *Course Topics.*

The primary text for this class will be lecture notes and examples created by the instructor. These are listed above as “Petersen *subject* n-nn” where n-nn indicate page numbers as applicable.

Additional readings will include, but not be limited to:

Hiller, L. A., and Isaacson, L. M. (1960). *Experimental music* [PDF]. New York, NY: McGraw-hill. https://archive.org/details/experimentalmusi00hill/page/82/mode/2up

Holmes, T. (2015). *Electronic and experimental music: Technology, Music, and Culture*. Routledge.

Roads, Curtis. (2015). *Composing electronic music: A new aesthetic*. United States: Oxford University Press.

Roads, Curtis. (1996). *The computer music tutorial* (2nd ed.). Cambridge, MA: MIT Press.

Ruviaro, B. (2024). *A Gentle Introduction to SuperCollider* (Rev. 2024) [PDF]. Lulu. https://ccrma.stanford.edu/~ruviaro/texts/A_Gentle_Introduction_To_SuperCollider.pdf

Wilson, Scott., Collins, N., & Cottle, D. (2011). *The SuperCollider book*. Cambridge, MA: MIT Press.

Course Structure and Workload

Classes will be divided into interactive lectures, group activities, pair programming and presentations. Quizzes will be given every Monday. Problem sets are due weekly. Projects, which start week 3, build on the problem sets and previous projects.

- Reading (~25-30 pages per week)
- Problem sets (4) (5% each, total 20%)
- Quizzes (5) (3% each, total 15%)

- Pair Programming assignments (8) (completed in-class. 2.5% each, 20% total)
- Projects (3) (45%)

Description of Graded Work

Problem Sets

Problem sets foster mastery of the domain specific language used in the course, SuperCollider. Psets will build on example code provided and discussed in class and are designed to accomplish some musical or audio task integral to the project for each module. Thus, psets are not just assessment of technical mastery, they facilitate successful completion of the projects.

Quizzes

Quizzes are five-question multiple choice given during the first five minutes of Monday's class. They are designed to foster engagement with the reading and theoretical topics introduced in lecture.

In-class Pair Programming Exercises

In each class, students will pair together to solve different musical problems using an audio programming language. After class, that work will be turned in to Canvas. Work can be continued by the pair and turned in same-day if not completed by the end of class.

Projects

Projects are designed to be complete, creative applications of course topics. Starting in week three, students will choose a project option for that week to be completed outside of class and in addition to any other graded work (see above.) These short projects are designed to allow students to choose and realize musical applications in code for the theoretical topics learned in lecture.

Late Work Policy

Late work will be accepted at my discretion. All late submissions must be pre-approved. Late work will be assessed a 10% penalty of the possible grade. Work to be submitted more than two days passed the due date will incur additional late penalties.

Quizzes and pair-programming exercises missed due to absence cannot be made up.

Participation and Attendance

Each class works best when every student is invested and engaged because you can learn from each other as well as from me. Peer advice, evaluation, and encouragement is key to the success of every student. Thus, *students are expected to attend every class* and to participate fully in the learning experience by *contributing to the discussion, providing feedback during presentations, and being fully prepared by completing reading assignments and graded work on-time*. Consistent lateness, unexcused absences, and lack of class participation will negatively affect both your learning and your final grade.

Course Guidelines

All students in the course ***must*** read the course guidelines, which can be found in section 2 of the CS Music Handbook. The guidelines exist to ensure that your in-class interactions and academic work meet course expectations. They cover expectations for presentations, submitted projects, and the **course policy on electronic devices**.

Class Culture:

This class, like all classes at Yale (and all work environments in which you will find yourself after your time at Yale) is rich with a diversity of opinion, expertise and life experiences. While your specific expertise may make some work in this class easier, your peers may have knowledge that facilitates understanding differently. By drawing on our strengths to contribute to discussion and problem-solving, we help each other understand in a more rounded, complete way. This is only possible by actively embracing our differences as strengths through a foundation of respect and a practice of positive inclusion in everything we do.

Accessibility:

Your success in this class and beyond is very important to me. If any aspect of the physical course (format or availability of course materials, course website, etc) is presenting a barrier to your learning, please let me know as soon as possible. Together we can develop strategies that will enable you to succeed in the course. Further, I encourage you to visit Student Accessibility Services to determine how you can improve your learning as well. If you need official accommodations, you have a right to have these met. There is also a range of resources on campus, including the Writing Center, Residential College Tutors, and Academic Strategies. If you have questions or concerns about any of the support options available to you, please speak with me.

Statement on Academic Integrity:

Students will be held to Yale University's standards for ethical conduct and are expected to complete their assignments and exams individually unless explicitly directed to work in groups. Graded coursework involves a variety of materials: written answers, source code, and music/audio project files. I will sometimes encourage you to seek out models or examples, especially of audio programming code. In these cases, I will provide instructions regarding how you are to cite your models. If no explicit indication to use models is given, work must be original and completed by the student alone. Failure to properly attribute non-original work will be considered a violation of Yale's standards for ethical conduct. *Collaboration with ChatGPT or other AI composition software is not permitted in this course.*

Disclaimer:

This syllabus, the nature and number of projects, readings, and topics are subject to change either by necessity or design. Any changes will be reflected in a new (or updated) syllabus and announced in class or via Canvas.