**Ramapo College of New Jersey**

School of Contemporary Arts

MUSI 650 - 20 : Creative Musical Coding

Fall 2024

Daniel Fishkin / dfishkin@ramapo.edu / dfiction.com

[**https://github.com/dfict/CreativeMusicalCoding**](https://github.com/dfict/CreativeMusicalCoding)

SYLLABUS IS SUBJECT TO REVISION DURING THE SEMESTER

*Course Description:* This course provides a foundational introduction to music programming languages. SuperCollider, Python, Processing and Open Frameworks will be explored through hands-on and creative applications. Students will gain experience with interactive visual art, game design, audio/music signal processing, and machine learning.

This course does not provide a thorough introduction to any single computing language. Rather, it expects an encourages a humanistic approach translating higher level concepts of art to lower and very low levels of applications specific to your intent.

Course Instructor: Daniel Fishkin

*Lecture Time:* Wednesdays, 6:05 pm — 9:35 pm

*Lecture Location:* Electronic Music Lab, H-Wing 204 or remote.

*Instructor:* Daniel Fishkin

*Office Hours:* **by appointment (IRL or Remote)**

*E-mail:* [daniel.fiction@gmail.com](mailto:daniel.fiction@gmail.com)

dfishkin@ramapo.edu

*Materials Fee:* $50

*Textbooks: Bruno Ruviaro, A Gentle Introduction to Supercollider* [*https://ccrma.stanford.edu/~ruviaro/texts/A\_Gentle\_Introduction\_To\_SuperCollider.pdf*](https://ccrma.stanford.edu/~ruviaro/texts/A_Gentle_Introduction_To_SuperCollider.pdf)

Valle, Andrea. *Introduction to Supercollider. Logos Verlag Berlin. 2016. 3832540172*

Manaris, Bill. *Making Music with Computers: Creative Coding in Python.* Routledge. 2014. 1439867917

Shiffman, Daniel. *The Nature of Code. Creative Commons. 2012. 0985930802*

Software **Max**

<https://cycling74.com/>

Max is an object-oriented programming language that makes easy prototyping of sound environments/instruments accessible without needing to learn to code on levels like JS or C. It is a type of coding language in its own way, but the coding is done primarily with virtual representations of wires. It’s very fun and many of my class demos will get you excited about what’s possible—download it and see what you can get working off the bat!

*Note: you can try Max for free for 30 days, and thereafter you’ll need to purchase it, or sign up for a monthly or yearly student subscription. It is affordable. Conn College has Max on every machine in the computer lab!*

**Max comes with an extensive body of educational materials, tutorials, help files, and project examples that can easily be copy/pasted into your own sketches.** Students will be expected to read and study Max / MSP tutorials on their own and learn the syntax of the software throughout the course of the semester!

**Enter the MAX file browser and search:**

**collection:"Tutorials/Max Tutorials@cycling74"**

**collection:"Tutorials/MSP Tutorials@cycling74"**

**Supercollider**

<https://supercollider.github.io/downloads.html>

Supercollider is a platform for audio synthesis and algorithmic composition, used by musicians, artists and researchers working with sound. It is code-based, completely free, light in size, and a little hard to work with.

**Spear**

<https://www.klingbeil.com/spear/>

Spear performs Fourier Analysis in sound files. This is free software that allows you to turn recordings into individual sine waves and edit/select them.

**Audacity**

<https://www.audacityteam.org/>

Audacity is a basic editing program that is excellent for transforming individual sounds.

**Python**

Extremely powerful code based platform

<https://www.python.org/downloads/>

**Csound**

[*https://github.com/dfict/clawyer*](https://github.com/dfict/clawyer)

**Arduino (Processing)**

Hardware coding 101!

Hardware: <https://www.frommdesign.com/product/ad-1-dsp-dev-board>

AD-1 is a DSP development board based around an ESP32-s3 microcontroller and an es8388 audio codec. Designed for experimenting with digital audio, the AD-1 can generate sound on its own or manipulate incoming stereo audio. This board costs around $50 and will be the basis for many hardware projects

Course Objectives:

● Production Techniques: evaluate and apply advanced techniques in music production, synthesis, recording, programming, editing, and mastering;

● Theoretical Foundations: identify and interpret core theoretical foundations of audio;

● Context: analyze and articulate the importance of context in diverse practices within music technology;

● Musicianship and Production: integrate traditional, non-Western, and experimental approaches to musicianship and composition with music production skills;

● Collaboration: collaborate with others effectively and demonstrate leadership in professional situations; and

● Innovation: recognize and formulate innovative approaches to music technologies and careers.

**Course Goals**

● Understanding the relationship between computer programming, music, and digital audio

● Developing core proficiencies relevant to algorithmic composition, interface design, computer programming, analysis and signal processing

● Gaining the ability to read, write, and modify computer code related to music.

**Measurable Student Learning Outcomes**

● Comprehend the importance of music programming in improvised and notated contemporary music (Projects)

● Demonstrate the ability to create programs in SuperCollider, Python, Processing and C++ in the production of compositions an performances (Projects)

● Understand the fundamentals of computer programming languages for music and be able to implement them to build original Audio Unit & VSTplugins (Projects)

**Grading Rubric**

Class Participation / Attendance 20%

Weekly Corpus 20%

Etudes / Projects/ 20%

Presentations 20%

Final Project 20%

**Class Participation / Attendance 20%**

Class attendance is mandatory. We work and experiment in class. Discussions, critiques lectures and demonstrations provide the basis for the successful completion of projects, and they are difficult to re-create outside of class. In order to participate, you must be in attendance. You have one unexcused absence permitted for the semester. Your final grade will drop by 4 points for each further unexcused absence. More than four absences will result in a failing grade for the course. You are expected to participate actively in class by asking questions, bringing energy to discussions, and arriving with prepared homework/projects. Independent motivation is expected.

**Weekly Corpus 20%**

Students are expected to keep a weekly audio “journal”. Every week, students will collect/record/curate sound file collections. The collections each week will total up to 5 minutes of sound (or more), featuring at least 5 different recordings. (ie, Week 1, you record five 1 minute recordings. Week 2, twelve 10 second recordings, and one 3 minute recording). It is expected that you carefully label these files and keep good file management, in additional to normalizing audio and trimming the beginning and endings of each file. You can record with a high quality field recorder you borrow from the university, you can record with your phone. You can also “sample” recordings found in the wild (but are discouraged from sampling pop music.)

Some prompts will be provided to help simplify this task. But ultimately you are encouraged to: **make it your own** and may disregard time and file requirements as the semester goes on and you figure out what you’re doing.

Upload and label your corpus with your name as a sample pack weekly on <https://freesound.org/>

PW: soundfriends

**Etudes / Homework 20%**

Short reading assignments or creative prompts will be assigned each week. Students will be expected to complete them and come to class ready to discuss their progress. You cannot “get ahead” of these assignments by doing them all ahead of time—or catch up!—they are meant to be part of a weekly practice that is customized to the flow of the class.

**Artist Presentation 20%**

Do a short research presentation on an artist or piece of art that inspires you. 10 min. The purpose of this assignment is to aid the development of your final project by creating an opportunity to research something that interests you for your own creative purposes and research agenda.

**Final Project 20%**

Make a creative thesis and execute it. The project may include video, audio, and may somehow utilize a computational process to illuminate your corpus. You can use synthesis, sampling. You can also persue something mentioned in class but not explicitly explored. You can write code for the Flying Fish Board, for a Noise Toy, in Supercollider. You are invited not to merely make a demo—Make a piece! I would even accept a project in Max MSP. You can use your recorded corpus or create a new one. Students may work collaboratively. The project must be proposed and accepted by instructor. We may present these projects at the Tristan Perich concert, on 12/11, pending feasibility and student interest.

**AI Policy**

The use of Machine Learning tools such as ChatGPT and Claude are permitted, generally. In some cases we will explore them deliberately. In some instances you will be discouraged from using these tools. AI represents a sea change for humanity. It also represents a paradigm shift for pedagogy in digital literacy.

I expect you will want to use AI (ChatGPT and image generation tools, at a minimum), in this class. In fact, some assignments will require it. Learning to use AI is an emerging skill—be aware of its limits. If you provide minimum effort prompts, you will get low quality results. You will need to refine your prompts in order to get good outcomes. This will take work.

Don’t trust any code that your AI provides you. You will be responsible for any errors or omissions provided by the tool. AI is a tool, but one that you need to acknowledge using. Please include a paragraph or citation on any assignment that uses AI explaining what you used the AI for and what prompts you used to get the results. Failure to do so is in violation of the academic honesty policies.

*SCHEDULE*

*Class 1 Wednesday 9/4*

**Introduction to the SuperCollider 3 environment**

**Server and Language**Reading:

A gentle Introduction to Supercollider

<https://ccrma.stanford.edu/~ruviaro/texts/A_Gentle_Introduction_To_SuperCollider.pdf>

Pages 1-22

SC tutorial: Getting Started With SC: sections 1 – 5

<https://doc.sccode.org/Tutorials/Getting-Started/00-Getting-Started-With-SC.html>

**Etude 1 (due 9/11 next week): Go shopping!**

Try out all examples from 2 - SC2-examples\_1. Go to <http://sccode.org/> Try out at least 10 sound examples. Select one that you would like to understand better and modify it. Submit it on collab.

**Nice videos to get you started:**

Live Coding in SuperCollider: a Tutorial with Eli Fieldsteel <https://www.youtube.com/watch?v=rlf8XBxLfRM>

<https://www.youtube.com/playlist?list=PLPYzvS8A_rTaNDweXe6PX4CXSGq4iEWYC>

Watch one or two of these each week

**Also for Next Week:**

Read: The End of Programming

<https://cacm.acm.org/opinion/the-end-of-programming/>

**First Corpus Prompt:**

Record 5-7 sounds from your life. At least one sound should be a drone. One should contain a voice. The second file should contain a pitched instrument sound. The third file should contain a percussion sound. All files should not be too short (less than a second) or too long (more than a minute). They should not be boring sounds. Don’t record something like typing on your computer. Don’t waste tape!

*Class 2 Wednesday 9/11*

Going over Joo Won Park’s

Four Hit Combo (for laptop ensemble)

**Second Corpus Prompt:**

Record 10 different samples than can be used in a drum machine / sampler approach.

*Class 3 Wednesday 9/18*

Nodeproxy and gui introductions

**First Etude**

Bring 1-3 synthdefs, nodeProxy, or nDef with Gui machines. Feel free to use the SC examples from week 1 to have something that you know works, to work off from.

Or make something new! ☺

Bring to class these working instruments and prepare by learning how to use them. We will improvise together!

**Corpus Week:**

Bring whatever you want!

*Class 4 Wednesday 9/25*

**Iteration and Buffers**

Going over iteration, for loops, and different classes and methods for filling arrays of data.

**Corpus 6:**

Find a class mate and prepare to become their interview partner. Partners can be chosen the good old fashioned way or using a random number generator.

Conduct five 60 second interviews with them about a topic. Your task is to spend a good 30 minutes getting to know them and figuring out what stories they can tell are really interesting, and you’ll practice getting them to condense their story (with a minimum of prompting) to a short “podcast” news style interesting gem, so they don’t waste time talking about boring things. Students can think, however, about interesting ways to tell their story, and they can use a personal style of delivery. After the rough concept is determined, take turns recording each other’s five stories. The interviewer can direct and prompt the five recordings in any way they choose. However, the voice of the interviewer is never to be heard—only record the interview subject. Each student should have a corpus of five recordings, 60 seconds each, from their interview partner.

*Class 5 Wednesday 10/2*

***Soldering microcontrollers— the Loud Objects Noise Toy!***

**Homework:**

Now that your circuitt is working, you need to install “avrdude.”

<https://github.com/avrdudes/avrdude>

(bring a working personal labtop with a usb-a connection (old style))

HINT: use chatgpt or claude AI as your installation assistant. I used an incompatible version of “crosspack” so I had to uninstall that over the terminal, and then reinstall all the helper files with brew, and finally install avrdude, which took 2 minutes instead of the 45 minutes it took in class! ;P

**To get AVRDUDE for Windows, install the latest version from the** [**Releases**](https://github.com/avrdudes/avrdude/releases) **page.**

## Getting AVRDUDE for Linux

To install AVRDUDE for Linux, install the package avrdude using the software package manager. For example, under Debian/Ubuntu, you can use the following commands:

sudo apt-get install avrdude

## Getting AVRDUDE for macOS

On macOS, AVRDUDE can be installed through MacPorts or **Homebrew** (recommended!)

Alternatively, you may [build AVRDUDE](https://github.com/avrdudes/avrdude/wiki) yourself from source.

ETUDE 3:

Modify the “piano and accordion patch” with your own sounds.

Try changing the durations, too.

*Class 6 Wednesday 10/9*

***Online class***

*Exploring buffers with SC…*

*Reviewing your “piano and accordion” modified patch*

*checking AVRdude…did the install work?*

*Class 7 Wednesday 10/16*

*Programming and hacking with the AVR attiny85*

*Arduino intro*

*Class 8 Wednesday 10/23*

*Programming and hacking with the AVR attiny85*

*Using your arduino programmers!*

*Class 9 Wednesday 10/30*

*Artist Visit:*

***TRISTAN PERICH***

*Class 10 Wednesday 11/6*

***Election Day Special!***

*Creative Musical Responses to the Presidential Election*

*Class 11 Wednesday 11/13*

*Class 12 Wednesday 11/20*

*NO CLASS Wednesday 11/27 Happy thanksgiving~!!!*

*Class 13 Wednesday 12/4*

*Class 14 Wednesday 12/11*

***Loud Objects Concert!***

*Class 15 Wednesday 12/18 or TBA*

*Final Exam*