**Ramapo College of New Jersey**

School of Contemporary Arts

MUSI 650 - 20 : Creative Musical Coding

Fall 2024

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

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 provides a foundational introduction to music programming languages. Max MSP, SuperCollider, Python, and Processing and Open will be explored through hands-on and creative applications. Students will gain experience with interactive visual art, 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

*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 (paid directly to instructor)

*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 **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)

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 your’e doing.

**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—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, utilizing your audio corpus. The project may include video, audio, drawing, and must somehow utilize a computational process to illuminate your corpus.

**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>

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

*Go to sccode.org. Try out at least 10 sound examples you find there. Select one that you would like to understand better and submit it on moodle*

*OR:*

*Modify an existing short snippet of code from 2 - SC2-examples\_1*

*Also for Next Week:*

*Read:*

*The End of Programming*

[*https://cacm.acm.org/opinion/the-end-of-programming/*](https://cacm.acm.org/opinion/the-end-of-programming/)

*Class 2 Wednesday 9/11*

*Class 3 Wednesday 9/18*

*Class 4 Wednesday 9/25*

*Class 5 Wednesday 10/2*

*Playing with Csound! introduction*

*Class 6 Wednesday 10/9*

*Class 7 Wednesday 10/16*

*Class 8 Wednesday 10/23*

*Hardware Day: utilizing the ESP32 board*

*Class 9 Wednesday 10/30*

*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*

*Class 15 Wednesday 12/18 or TBA*

*Final Exam*