**New York University**

Interactive Telecommunications Program

Creative Code Orchestra: Improvising with Computers

Topics in ITP (11413) (Catalog 2378)

Spring 2025

Daniel Fishkin / df1575@nyu.edu / dfiction.com

SYLLABUS IS SUBJECT TO REVISION DURING THE SEMESTER

*Course Description:* In this class, students use computers to play music together, improvising and composing with code. Using the Max graphical language, students will learn to create their own software for musical improvisation, exploring sample manipulation, digital synthesis, and also principles of psychoacoustics and sound spatialization. Students are encouraged to bring in their own instruments in order to explore sound processing of acoustic instruments. In the beginning of the class, students will solder and assemble their own portable amplifiers to create a spatialized laptop sound system.

As students develop code together, the class forms a laptop ensemble, moving from theory to practice by designing software instruments for live-computer improvisation in the classroom. The class explores large format ensemble pieces, as well as smaller groups such as trios and duets. Students will learn to improvise, play existing repertoire for laptop, and can compose their own pieces. The class also explores hardware hacking topics such as microphone design and pressure transducers, bringing the sounds of the acoustic world inside and outside of our computers. The class will culminate in a group concert that features a range of performances, involving acoustic instruments, some duets, and large ensemble pieces.

Course Instructor: Daniel Fishkin

*Lecture Time:* Mondays, 12:10 pm — 2:40 pm

*Lecture Location:* Room 412 / 370 Jay St

*Instructor:* [Daniel Fishkin](https://dfiction.com/)

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

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

[df1575@nyu.edu](mailto:df1575@nyu.edu)

course website: TBD

*Materials Fee:* $50

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!

**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)**

<https://www.arduino.cc/>

Hardware coding 101! Arduino is a basic way to interface with sensors, LEDs, motors, electronic components, to make physical changes in the real world

**P5JS**

<https://p5js.org/about/>

<https://p5js.org/reference/p5.sound/>

P5js is the modern update of processing, which was a java based language with similarities to C.

“p5.js is a friendly tool for learning to code and make art. It is a free and open-source JavaScript library built by an inclusive, nurturing community. p5.js welcomes artists, designers, beginners, educators, and anyone else!”

P5js runs in the browser and doesn’t need to compile.

P5 sound library extends p5.js with Web Audio functionality including audio input, playback, analysis and synthesis.

**Sublime Text**

<https://www.sublimetext.com/>

My favorite code editor! Simple, elegant, free.

**Github**

All homework assignments, course materials, etc will be on github!

We won’t use instructure except to link your own submissions as webpages

Please make an account and fork the following:

<https://github.com/dfict/Code-Orchestra>

recommended: set up your system with github desktop!

**Then make your own repo for code homework, call it MCI-homework and upload all your assignments there.**

**Grading Rubric**

Class Participation / Attendance 50%

Etudes / Projects/ 25%

Final Project 25%

**Class Participation / Attendance 50%**

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 must always be prepared, technically. You must always come with the cables, adaptors, and equipment you need. You must always bring your computer. You have one unexcused absence permitted for the semester. Your final grade will drop by 6 points for each further unexcused absence. Students arriving more than 5 minutes late will be marked with a half absence. Students who leave early will receive a full absence. More than three 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.

**Etudes / Homework 25%**

Often, 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 “catch up” to these assignments by doing them late. They are meant to be part of a weekly practice that is customized to the flow of the class.

**Final Project 25%**

Make a creative thesis and execute it. Compose for the class. Or, bring a piece for the class composed by someone else and find a way to present, contextualize, spearhead it. The project may include video, audio, and may somehow utilize a computational process. You can use synthesis, sampling. You can also pursue something mentioned in class but not explicitly explored. You may compose a piece for our ensemble.

**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.

**Repetoire***Potential Repertoire is Here:*

[*https://github.com/dfict/Code-Orchestra/tree/main/rep*](https://github.com/dfict/Code-Orchestra/tree/main/rep)

*SCHEDULE*

*Class 1 Monday 1/27*

*Introduction to Class Policies, Syllabus, improvising*

***Student intros***

*Who are you?*

*What year?*

*What experience do you have with music?*

*What music do you like?*

*What instruments do you play?*

*What software do you know?*

*Do you play in bands?*

*Do you have performance history?*

*What do you hope to learn?*

What is a Laptop Orchestra?

Read: *Why a laptop orchestra?* By Dan Trueman

<https://manyarrowsmusic.com/papers/WhyALaptopOrchestra.pdf>

*Find your own example of a “laptop band”* *or* *computer band, can be any genre, bring to class for discussion*

**Possible goals:**

Be able to improvise using Anthony Braxton’s Language Music cues

Read: *Anthony Braxton's Language Music* by Nate Wooley

<http://archive.soundamerican.org/sa_archive/sa16/sa16-language-music.html>

**Homework: Prepare a “rig” for next week**

**Your rig can be electroacoustic, electronic, acoustic.**

*Class 2 Monday 2/3*

**Playing Language Music on Tocante Synthesizers**

And exploring whatever “rig” you may have brought!

**Exploring Opposite Earth (Jeff Snyder)**

**Homework prompt:**

research and bring “heterogeneous instrument” (personal percussion sound) to class next week.

**Joo Won Park’s *Four Hit Combo***

(for laptop ensemble)

**Second Etude Prompt:**

Record 5-7 sounds from your life. At least one sound should be a drone—an unchanging sound with pitch components. 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 (not less than a second) or too long (not more than a minute). They should not be boring sounds. Don’t record something like typing on your computer. Remember, recording processes used to be physically limited by the parameters of recording technology, represented by the cost of ferromagnetic tape. Don’t waste tape!

*Class 3 Monday 2/10*

**Getting deeper into Opposite Earth (Jeff Snyder)**

**Exploring Joo Won Park’s *Four Hit Combo***

**Introducing Bryan Jacobs**

*Controlling acoustic instruments with computers*

**Introducing Tristan Perich**

*Making Music on Microcontrollers*

*HW: please bring your finale sound for the joo won park piece!*

*HW: please install supercollider on your computers*

***NOTE No class presidents day 2/17!***

***Class 4 Tuesday 2/18***

Exploring Supercollider

Revisiting Joo Won Park

And learning a new Park piece!

**HW:**

**Soon we will be coding crazy 1 bit synthesizers.**

**But we need to made sure the Command Line tools you’ll all need for this process are installed on your own machines. This can be tricky with windows machines or LINUX platforms, but it is possible.**

Before working with the ATTINY85, 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.**

Min32 Download:

<https://sourceforge.net/projects/mingw/>

## 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.

*Class 5 Monday 2/24*

Hardware programming!

Exploring Sam Pluta

**Tristan Perich — Loud Objects —making music on the ATTINY85**

**HW:**

*Class 6 Monday 3/3*

**HW: Print Recorder Gate**

*Class* *7 Monday 3/10*

***Class workshop:***

***3d printing in the makerspace,***

*Homework: Print*

*Class 8 7 Monday 3/17*

***Class Visit: Bryan Jacobs***

**3/24—3/28 SpRING BREAK!**

**No class (!**

*Class 9 Monday 3/31*

*Class 10 Monday 4/7*

*Class 11 Monday 4/14*

*Class 12 Monday 4/21*

*Class 13 Monday 4/28*

*Class 14 Monday 5/5*