Fall 9-1-2018

MART 391.01: ST - Programming for Sound

Michael A. Musick

University of Montana, Missoula

Let us know how access to this document benefits you.

Follow this and additional works at: https://scholarworks.umt.edu/syllabi

Recommended Citation

https://scholarworks.umt.edu/syllabi/8855

This Syllabus is brought to you for free and open access by the Course Syllabi at ScholarWorks at University of Montana. It has been accepted for inclusion in Syllabi by an authorized administrator of ScholarWorks at University of Montana. For more information, please contact scholarworks@mso.umt.edu.
# Syllabus | Sonic Programming (MART 391.01 - Special Topics Number)

Date modified: 2018-09-26

## Table of Contents

- Course Overview
  - Basic Info
  - Description
  - Objectives & Student Learning Outcomes
- Professor/Instructor
- Course Websites
- Pre-Requisites
- Books and Supplies
- Policies
  - Course Evaluation
  - Grades
  - Attendance
  - Plagiarism and Cheating Policy
  - Students with Disabilities
  - Changes to the Course
- Intended Weekly Breakdown
  - Week 1
  - Week 2
  - Week 3
  - Week 4
  - Week 5
  - Week 6
  - Week 7
  - Week 8
  - Week 9 - Crit Day #1
  - Week 10
  - Week 11
  - Week 12
  - Week 13
  - Week 14 - Public Presentation
COURSE OVERVIEW

Basic Info

- CRN:
- Credits: 3
- Location: McGill 127 & McGill Sound Studio

Description

In Sonic Programming students will learn how to utilize computer science based techniques to create, process, and distribute original sonic art and sound based work. This course will spend a majority of its time focused on the real-time creation of sonic art, as well as the application of concepts from interactive theory and interactive systems. This course combines students skills and technique in creative coding and sonic arts together.

In a time with an increasing number of job opportunities that require individuals to be capable of not only creating custom content, but of creating custom content that is live with interactive opportunities, it is important that artists become capable of utilizing technologies that allow for this. There are many tools available to artists to create these types of artworks. Fortunately, the principles underlying most of these technologies are transferable. Therefore, it is critical that artists learn at least one of these environments at a deep level. The knowledge and skills developed in one are then transferable to others.

This course will involve engaging classes that involve hands-on work. These will be complimented with in-class lectures on technique, critical listening to ground your work, and discussions of aesthetics as it relates to this type of sonic art. Assessment of students will be conducted through weekly sketches and two larger artworks.

Objectives & Student Learning Outcomes

Through this course, students are expected to demonstrate an ability to:

- Discuss technical and aesthetic concerns within sonic arts that relate to creating custom art via specific programming environments.
- Employ the Unit Generator paradigm in relation to real-time signal based digital environments.
- Explain an understanding of digital audio signal flow as it relates to a non-DAW based sonic art creative process.
- Create organized code and have an ability to asses organization structures in other’s code as it relates to artistic ideas.
- Synthesize creative sonic art works that consider both relevant aesthetics of the medium and the technical demands of working with code.
- Present one significant sonic art work to the public.

Students are expected to bring ideas to explore these techniques and skills. The goals of the projects are not simply about the successful application of the technical processes. Instead, the goal is to integrate these elements to make work that is meaningful.

Professor/Instructor

- Dr. Michael Musick
- E-Mail: michael.musick@umontana.edu

Course Websites

- Course GitHub Repo (https://github.com/Montana-Media-Arts/sonic-programming) (This git repository holds code examples, an issues board, as well as course Wiki.)
Pre-Requisites

Students should have completed MART 120 - Creative Coding I. It is also suggested that students have taken MART 220 Creative Coding II and MART 245 - Introduction to the Language and Practice of Sonic Art and Audio.

Books and Supplies

There are no required texts that you must purchase for this course. All readings, music, or sound examples will be available to you through web links, the library, or the class Moodle.

Headphones

You will need to bring a pair of headphones for work in this class. I would suggest you invest in a good pair, as part of your grade will be based on the technical quality of your work. If you cannot hear issues because you have poor quality headphones, then you are risking poor grades.

You are looking for reference quality headphones. That means headphones that present your audio to you as honest and neutral as possible. A bad mix should sound like a bad mix. Hyped headphones that accentuate bass or highs in flattering ways can hide detail and mix issues.

A discussion of headphones will occur on the first day of the course.

Suggested Headphones

Headphones, like all professional media gear can be cheap or incredibly expensive. With that in mind, you can get good quality headphones that don't break the bank. Here are a few of my suggestions.

- **Samson SR850 Semi-Open-Back Studio Reference Headphones** - $30 - These are the cheapest that will still “get the job done”.
- **AKG K240STUDIO Semi-Open Studio Headphones** - $57 - These are great headphones. However, they are “semi-open” which will both improve sound quality for mixing and make anyone sitting within 10 feet of you potentially stare in annoyance.
- **Sennheiser HD 202 II Professional Headphones (Black)** - $60
- **Sony MDR7506 Professional Large Diaphragm Headphone** - $80 - (An industry standard for recording and working with headphones)
- **Sennheiser HD280PRO Headphone** - $100 - (I have had a pair of these since I was an undergrad. They are still my main pair of headphones for work when I need them. They are built to last and sound great.)
- **Audio-Technica ATH-M50x Professional Studio Monitor Headphones** - $150
- **beyerdynamic DT 770 PRO 32 Ohm Studio Headphone** - $170

For more info, the following guide provides good information on the two major types of headphones and suggestions.

- Best Studio Headphones: The Ultimate Musician’s Guide

Computer

You will need access to a computer capable of running modern audio synthesis and signal processing environments, as well as digital audio workstations (DAWs) and editing software. This computer should also be capable of real-time signal processing through analog inputs and outputs (i.e., using an audio interface or web cam). Please leverage departmental resources whenever necessary.

Software

For this semester, we will primarily be utilizing SuperCollider (SC). This tool is chosen because it is open-source, cross-platform compatible, widely used, and incredibly powerful.

SuperCollider is a platform for audio synthesis and algorithmic composition, used by musicians, artists, and researchers working with sound. It is free and open source software available for Windows, macOS, and Linux.

SuperCollider features three major components:
**scsynth**, a real-time audio server, forms the core of the platform. It features 400+ unit generators ("UGens") for analysis, synthesis, and processing. Its granularity allows the fluid combination of many known and unknown audio techniques, moving between additive and subtractive synthesis, FM, granular synthesis, FFT, and physical modeling. You can write your own UGens in C++, and users have already contributed several hundred more to the sc3-plugins repository.

**sclang**, an interpreted programming language. It is focused on sound, but not limited to any specific domain. sclang controls scsynth via Open Sound Control. You can use it for algorithmic composition and sequencing, finding new sound synthesis methods, connecting your app to external hardware including MIDI controllers, network music, writing GUIs and visual displays, or for your daily programming experiments. It has a stock of user-contributed extensions called Quarks.

**scide** is an editor for sclang with an integrated help system.
Policies

Course Evaluation

Student work and progress will be assessed through:

- In-class discussions and critiques
  - You are expected to engage with class discussions about technical and artistic issues.
  - You are expected to engage with class discussions about outside readings and listenings.
  - You are expected to engage in class critique days of the two projects.
- Sketches
  - You will be assigned sketches/compositions/exercises to practice the techniques being discussed in class.
  - These are critical to your development.
- Projects
  - There will be two larger assigned projects throughout the semester that relate to the current topics being discussed. These are the students opportunity to synthesis creative concerns and technical concerns from multiple weeks of content into singular sonic artworks.
- Project Documentation
  - Each project will require project documentation, including; a demo video, academic report describing aesthetic and technical success in relation to the larger field, and a written description.

Projects

We will have two projects throughout the course that are relevant to concepts, techniques, and creative ideas from recent content. These are the students opportunity to synthesis creative concerns and technical concerns from multiple weeks of content into singular sonic artworks.

Project grades will be based primarily on the student’s ability to:

1. Demonstrate an understanding of the specific characteristics and integrative capabilities of the assigned topic through artistic output, written documentation, and technical achievement.
2. Articulate a clear and concise perspective within the artwork.
3. Present an organized artistic piece, as well as; technical report, file/program structure, and demo video.
4. Demonstrate engagement with an iterative process of ideation → creation → presentation → assessment.
5. Demonstrate creativity beyond the expected technical requirements.
6. Properly and punctually deliver all assignment files.

Please Note: All Projects are required for this course. Failure to complete and submit a project will result in a failing grade for the course. There are no exceptions to this rule. Completion of this course requires all projects be completed.

Participation

This class will be participatory, and you are expected to participate in discussions and give feedback to other students.

Grades

Final Grades

Grades will be determined according to the following breakdown:

- In-class discussions and critiques: 10%
- Sketches: 50%
- Project 1: 10%
- Project 2: 30%

Letters are assigned according to the following final course percentages:
<table>
<thead>
<tr>
<th>Grade</th>
<th>% Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>[93–100]</td>
</tr>
<tr>
<td>A-</td>
<td>[90–93)</td>
</tr>
<tr>
<td>B+</td>
<td>[87–90)</td>
</tr>
<tr>
<td>B</td>
<td>[83–87)</td>
</tr>
<tr>
<td>B-</td>
<td>[80–83)</td>
</tr>
<tr>
<td>C+</td>
<td>[77–80)</td>
</tr>
<tr>
<td>C</td>
<td>[73–77)</td>
</tr>
<tr>
<td>C-</td>
<td>[70–73)</td>
</tr>
<tr>
<td>D</td>
<td>[60–70)</td>
</tr>
<tr>
<td>F</td>
<td>[0–60)</td>
</tr>
</tbody>
</table>

**Late Work**

IMPORTANT: Since projects are experienced as a group in class or through a public presentation, late projects are not permissible and will not be accepted.

Documentation assignments or weekly sketches handed in after the due date and time will have points deducted for lateness. This will be in addition to any points deducted for content. Those that are uploaded late but within one day of the due date will lose 5% for lateness. For those uploaded after that, the number of deducted points will be at the discretion of the professor.

**Attendance**

- Attendance will be taken at the beginning of every class
- Critique days are mandatory. No exceptions. No tardiness.
- Contact me in advance if you will not be in class. (email is preferred)
- Unexcused absences will affect your grade.
  - One absence is allowed; after that, your final overall grade for the course will drop by 2.5% for each additional absence.
  - You are expected to work with colleagues to catch-up on what you missed.

**Plagiarism and Cheating Policy**

Students are expected to adhere to academic conduct policies of the University of Montana as explained in Section V of your University of Montana Student Conduct Code: “Academic misconduct is subject to an academic penalty by the course instructor and/or a disciplinary sanction by the University. Academic misconduct is defined as all forms of academic dishonesty, including but not limited to: (1) plagiarism, (2) misconduct during an examination or academic exercise, (3) unauthorized possession of examination or other course materials, (4) tampering with course materials, (5) submitting false information, (6) submitting work previously presented in another course, (7) improperly influencing conduct, (8) substituting, or arranging substitution, for another student during an examination or other academic exercise, (9) facilitating academic dishonesty, and (10) Altering transcripts, grades, examinations, or other academically related documents.”

Dishonesty will not be tolerated in this course. This includes, but is not limited to, cheating on tests, cheating on assignments, fabricating information or citations, having unauthorized possession of examinations, submitting work of another person or work previously used, or tampering with the academic work of other students.

Plagiarism is the presentation of the work of another without acknowledgement. As defined by the University of Montana’s Student Conduct Code, plagiarism is “Representing another person’s words, ideas, data, or materials as one’s own.” Students may use information and ideas expressed by others, but this use must be identified by appropriate referencing.
Students who cheat or plagiarize will receive academic sanctions, which may include an “F” grade on the assignment, examination, and/or in the course. Students will also be reported to the Dean of Students for possible further disciplinary action.

**Using Code or Media Found Elsewhere**

It is easy to find code and media (i.e. videos, sounds, images, etc.) online. If you use code or media from elsewhere (which you will at times), I expect you to cite the work and author.

If you use found code, you are expected to comment each line, as to what each line does programmatically. *Do not* summarize several lines of code from a high level (i.e., TV Guide). I expect you to comment each line on a granular level. In addition, in these cases, I am also looking for significant modification of the code, for you to enact your own ideas and to experiment heavily. Significant modification means beyond variable name and value changes. It is bending these concepts to your idea, especially graphically. It is not a copy and paste job. Also, never more than 40% of your code may be supplied from elsewhere. Period. If you use code from online, whether for inspiration, modification or reference, I expect to see a link in your comments from where you got the code and who wrote it. Otherwise it will be considered as plagiarism, and you will fail the assignment. The code must have a reference, along with URL and be commented out LINE BY LINE.

If you use media, *YOU* are responsible to ensure it is used according to fair-use guidelines. The pieces you make in this course are intended to be portfolio-quality works. Therefore, you should not utilize found media with restrictive use guidelines or licenses. You can read more about various licenses at:

- opensource.guide
- choosealicense
- GNU Licenses
- Creative Commons Licenses
- opensource.org

For found media, you are also expected to cite the media in your documentation for the project.

**Students with Disabilities**

The University of Montana assures equal access to instruction through collaboration between students with disabilities, instructors, and Disability Services for Students. Students with disabilities are encouraged to plan ahead and can contact Disability Services for Students (DSS). If you think you may have a disability adversely affecting your academic performance, and you have not already registered with Disability Services, please contact Disability Services in Lommasson Center 154, or call (406)243-2243. I will work with you and Disability Services to provide an appropriate modification.

**Changes to the Course**

I reserve the right to change the intended content of this course throughout the semester. This may be done to adjust for the speed of the class, to better meet educational goals, or to account for changes in technology.
Weekly Breakdown

Week 1

Monday

Welcome and course overview

- Course Overview
- Syllabus
- Discussion of Technologies, Tools, and Supplies
  - SuperCollider
  - Headphones
- Critique Days (Mandatory)
  - Project 1 - Critique in Class; Wednesday, October 24th
  - Project 2 - Critique in Class; Monday, December 3rd
- Extra Meetings (Mandatory)
  - Concert Showing (Project 2) - Monday, December 3rd. From 7pm-9pm
  - Finals Meeting (Wednesday, December 12th; 1:10pm-3:10pm)
- Class Discussion

Wednesday

Listening, Inspiration, and Software Installation

- Headphone Listening and Evaluation Discussion
  - Spotify List of Suggested Tracks
- Discussion and Listening to artists that will inspire our semester.
- Install SuperCollider
- Review git & GitHub.com

Week 2

Monday:

No Class on Monday, Sept. 3rd for Labor Day

Wednesday

Installation Revisited, “Hello World”, and lang vs. engine

- Review install of SC
- Discussion of
  - SCSynth
  - SCLang
  - SCIDE
- Start Your Engines (SCSynth) - Booting the Server
- String-based “Hello World“
  - .postln
- An SC Approach to "Hello World" (aka. your first sine wave)
  - {Sin0Sc.ar()}.play;
- Executing code by lines (shift + return)
- Declaring Code Blocks ( )
- Executing Code Blocks (cmd/cntrl + return)
- Comments
  - line comments ( // )
  - block comments ( /* */ )
Monday

Moving beyond static sine waves with modulation

- Controlling Amplitude
  - `mul`
  - BinaryOpUGen
    - `SinOsc.ar() * 0.2`
- Using the mouse to change parameters.
  - `MouseX.kr & MouseY.kr`
  - Let's make a Theremin %
- Introspection with polling %
  - `.poll`
  - `.plot`
- Using `SinOsc.ar` to modulate UGen input arguments %
  - Two ways of approaching Amplitude Modulation (AM Synthesis) %
  - Frequency Modulation Synthesis via oscillators %
- `mul` and `add`
- Error Messages %
- Altering parameters in example code %
- What is a "UGen" %
  - Unit Generators and Synths - SuperCollider help files %
  - Unit generator - Wikipedia %
- Review Basic Data Types %
  - Int %
  - Float %
  - Char %
  - String %
  - Symbol %
  - Array %

Sketch TODO: Use MouseX/Y or Additional SinOsc UGens to play. Make an "interesting" synth that explores the relationships between these UGens. Can you recreate what we have been doing in class on your own? Do you understand what is happening? We will share results at the start of next class.

Wednesday

Audio Rate vs. Control Rate, Signal Flow, Variables, and Arguments

- Audio Rate (.ar) vs. Control Rate (.kr)
- `.plot(duration:1)`
- Variables in SynthDefs
  - `var`
- "Global" Variables
  - `a - z`
  - interpreter variables
  - `~var`
  - environment variables
- Variable scope in SC
- Arguments in SynthDefs
  - `arg`
  - using `.set(\argname, value)` to alter running synths
- Signal Flow
  - Representing Signal Flow in Code
  - Pyramid Structure or Nested Code
  - Using Variables ( `var` )
  - variable reassignment
Week 4

Monday

Tour of Basic Deterministic Generator UGens, and Functions

- Tour of UGens
- Tour of Deterministic Generator UGens
  - UGens → Generators → Deterministic | SuperCollider Help
- Functions
  - {}
  - 04. Functions and Other Functionality | SuperCollider Help
  - 05. Functions and Sound | SuperCollider Help %
  - Functions | SuperCollider Help %
  - Function | SuperCollider Help %
  - Evaluating Functions stored in variables
    - .value()
- Easy SynthDef via function notation ({} .play();)

Wednesday

“In Living Stereo”, Signal Arrays, and Output

- Stereo Expansion
  - SinOsc.ar([400, 440])
  - sig ! 2
  - .dup
  - [sig, sig]
- Stereo UGens
  - Pan2.ar()
- Mixing down signal arrays
  - Mix([])
- The output UGen
  - Out.ar()
- 0-based language (including audio output Buses)

Sketch TODO: Create a simple synth utilizing additive synthesis, subtractive synthesis, amplitude modulation, frequency modulation, or any other deterministic or stochastic UGens that you can play and change via mouse interaction or argument setting. We will share these at the next class.

Week 5

Monday

Envelopes and Randomness

- 1-dimensional envelopes
  - Line.kr() & XLine.kr()
  - Decay.ar()
- Linear Envelope
  - linen.kr()
- ADR, ADSR, Perc, and Custom Envelopes
  - Env.
  - EnvGen.kr()
- Using envelopes beyond amplitude control
- Plotting envelopes w/ .plot
  - doneAction: x
- Looping/Triggering Envelopes
- Random Number generation on the language and server
- Data manipulation
  - .linlin
Wednesday

Server-Side Sequencing

- Clock UGens
  - Impulse
  - Dust
  - LFNoise0/1/2
- Trigger UGens
  - Stepper
  - Select
  - Trig1
  - Latch
- Envelope Gates

Sketch TODO: Create a short algorithmic work that utilizes envelopes and server-side sequencing/triggering.

Week 6

Monday

SynthDef's and Synths

- SynthDef's
  - Loading and storing them
- Playing Synths
- Initializing Arguments
- Altering Synths
- Load and Play a SynthDef
- The Synth Tree/Graph
- free-ing synths

Launch Project 1

- Create a significant sonic artwork using the skills you have thus far acquired with SuperCollider and in Sonic Programming.
- This artwork may be presented as a live work that you "perform" or a fixed-work that is played back from a sound file or SC itself.
- If you create a fixed-work, please feel free to utilize a DAW to edit, mix, and master your content.
  - This does not have to be a worked fully realized within SuperCollider.
  - However, all sounds should originate from SC, but may be further edited together and mixed in an external environment.
- These are to be significant pieces that show engagement with the creative process as it relates to working with sonic programming.
  - To that end, you should document your creative process, including:
    - ideas
    - tests
    - personal responses
    - iteration
    - creative goals
    - desired artistic outcomes
  - Documentation should be completed through a combination of recording, git commits, and personal reflection.

Wednesday

Language-Side Scheduling and Sequencing

- Scheduling with `{}.fork & t.wait
  - .do{} loops
  - .stop
- TempoClock()’s
- Task({})
Routine({})

Sketch TODO: Write a short work that encapsulates your synth’s as SynthDefs. Then utilize language-side scheduling to launch and alter synths algorithmically.

Week 7

Monday

Sound Input and Buses

- Processing Audio from the Real World
  - `SoundIn.ar()`
  - Warnings about feedback
- Server Options
  - Specifying number of input or output bus channels
- An in-depth discussion of Buses
- Audio Buses
  - `a = Bus.audio(s,1)`
    - `a.index`
  - `Out.ar()`
  - `In.ar()`
  - `InFeedback.ar()`
  - `LocalIn.ar()`
  - `LocalOut.ar()`
  - Overwriting Buses
  - Block Processing Explained
- Control Buses
  - `c = Bus.audio(s,1)`
  - `c.index`
  - `c.set(9)`
- Using Buses as Synth Arguments Maps
  - `'Synth__00'.map(freq, c.index)`

Wednesday

Nodes, Groups, Effects, and Synth Order
- Nodes and the Synth Node Tree
- Order of operation
- specifying node order
  - `.tail`
  - `.head`
  - `.before`
  - `.after`
- Using Synths as Effects Processors
  - Tour of Delay UGens
  - `CombN/L/C`
  - Delay
  - DelayN
  - `Delay UGens | SuperCollider Help`
- Tour of Filter UGens
  - `LPF`
  - `RLPF`
  - `HPF`
  - `RHPF`
  - `BPF`
  - `BRF`
  - Resonz
  - `Filter UGens | SuperCollider Help`
Week 8

Monday

*Sound Files and Buffers*

- What is a Buffer?
- The various ways the Buffers are used in audio processing environments
- Allocating Buffers
  - `b=Buffer.alloc(s, 10*)`
- Freeing Buffers
  - `b.free`
- Discussion of the importance of freeing buffers
- Loading Audio Files into Buffers
- Playing Buffers
  - Rate
  - Direction
  - Amplitude
- Playing Back LOOOOOONG Audio Files
- Recording SC

Sketch TODO: Create a short sound work that explores the use of sound input or soundfiles along with inter-synth bussing.

Wednesday

*Granular Synthesis*

- Granular Synthesis
  - Discussion and Description
  - Theoretical Underpinnings
- Playing with Grains
- Grain Rate
- Window Functions
- Precise Timing in SC

Week 9 - Crit Day #1

Monday

*Individual Review Day in Preparation for Crit #1*

- This is a chance for you to work on your works and get help from the instructor or your fellow classmates.

Wednesday

*Crit Day #1*

- Present Project 1 artworks in class.
- These will be critiqued in class with the whole class participating.
- Regardless of whether you work is a live or fixed piece, you should submit a fixed version to the instructor. This should be accompanied with a program/gallery note as well as longer artistic description.

Week 10

Monday

*Patterns & PBind*

- Please Read:
Part II - Patterns" from Bruno Ruviaro's A Gentle Introduction to SuperCollider

This week is a deep dive into the world of algorithmic composition, as allowed through SuperColliders Pattern capabilities.

- Basic Pattern techniques with PBind
- Effecting and controlling everything with patterns
- Patterns Math
- Patterns in Patterns, effecting patterns.
- Creative Approaches to Algorithmic Driven Art

Launch Project 2

- You are to create a significant sonic artwork that will be presented to the public the night of December 3rd.
- This artwork can be a performance-based work or gallery-based work.
- This is a significant work.
- This work is to be made exclusively within SuperCollider %
  - NOTE: You may bring other media into SC, but you may not create works that are finished outside of SC as in Project 1. %
- This is to be significant pieces that show engagement with the creative process as it relates to working with sonic programming. %
  - To that end, you should document your creative process, including:
    - ideas
    - tests
    - personal responses
    - iteration
    - creative goals
    - desired artistic outcomes
  - Documentation should be completed through a combination of recording, git commits, and personal reflection.

Sketch TODO: Create a short artwork utilizing patterns and pattern manipulation

Wednesday

Performance Issues with SC & Inter-App Audio Routing

- Technical considerations for SC when performing or presenting live.
- How to setup SC as a performance environment.
- How to setup SC as an installation engine.
- Inter-Application Audio Routing

Week 11

Monday

Fast Fourier Transform (FFT) Processing

- Overview of the mathematical concepts behind Fourier transforms.
- Signal Analysis in the Frequency-Domain
- Converting signals between time-domain and frequency-domain
- Frequency-Domain based processing

Wednesday

Analysis and Basic Machine Listening

- How can we use audio signals as control data?
- Amplitude following
- Peak following
- Fundamental Frequency Estimation
- Frequency-Domain Analysis Techniques

Week 12

Monday

Open Sound Control
What is the Open Sound Control (OSC) protocol
OSC and your computer
OSC between SCSynth and SCLang
Inter-app communication with OSC
Computer to Computer communication with OSC
Using your phone to control SC
TouchOSC
Lemur

Wednesday

Physical Computing

- Review of Arduinos, Teensy, etc.
- Communicating between Arduino and SC with Serial connection
- OSC Libraries for Arduino and Teensy
- Using an Arduino as a controller
- Controlling Arduino from SC

Week 13

Monday

Work and Feedback on Project 2

- You are encouraged to present or perform your work for the professor.
  - There will be a second room setup to accommodate this.
  - This is an opportunity to practice your performance or present your gallery piece and receive critical feedback before the official crit day.
  - The professor will provide you with information about what was and was not successful in their view.
  - If this is a performance, the professor will work with you on the technical details of performing and the reception of the performance.
  - If this is a gallery work, the professor will work with you on the technical details of presenting a self-running work and the reception of the work.

Wednesday

Work and Feedback on Project 2 Continued

- We will continue informal presentations and feedback with the professor.

Week 14 - Public Presentation

Monday

Public Performance and Presentation

- During class we will setup the presentation areas and performance space for the evening show.
- For those of you presenting gallery-works, you will be responsible for setting up, cleaning, and preparing the gallery spaces.
  - After setup is complete of the spaces themselves, you may install your works and test them.
- For those of you presenting performance/concert works, you will be responsible for setting up, cleaning, and preparing the performance space.
  - After setup is complete, we will briefly sound-check each piece.
- Call for the evening presentation will be 5:00pm. Please plan to be to the space by that time.
  - Gallery spaces will open at 5:30pm.
  - The show will start around 7pm.
  - Gallery spaces will remain open for 30 minutes after the shows conclusion.

Wednesday
Formal Critique and Wrap-up Discussion

- We will have a formal critique of presentations from Monday. Please be prepared with a fixed video/audio-representation of your work.
- We will also discuss the semester, future opportunities, and directions that this work can take you.