Syllabus

Tandon School of Engineering of New York University

Technology, Culture and Society

Integrated Digital Media

DM-UY 1133-D Creative Coding FA20

Fall 2020 – Professor: Effie J. Rieper

Tuesday + Thursday 12:00 PM - 1:50 PM

To contact professor: er129@nyu.edu

A note about this syllabus

This is a living document, as are the materials in this course repository. There are certain aspects of our course, such as our calendar or grading criteria, that will not differ from this syllabus unless there are extenuating circumstances. However, the pacing of this course may vary depending on student need, comprehension and technologies that are sometimes rapidly changing in the midst of our learning process. I will do everything in my power to update you on any changes ASAP (in class and via Slack) and to stay fully transparent about what you are expected to learn, produce, and share. Please engage in open communication, as I will do the same.

Creative Coding: An Intro to Creative Coding

This course is an introductory programming class, appropriate for students with no programming experience, who are interested in creating interactivity. Traditionally, introductory programming teaches algorithmic problem-solving, where a sequence of instructions describe the steps necessary to achieve a desired result. In this course, students are trained to go beyond this sequential thinking - to think concurrently and modularly. By its end, students are empowered to write and read code for creating interactivity including: event-driven creative applications, interactive installations, graphical user interfaces, games and user interfaces. Interactivity will be introduced and discussed throughout.

Program Goals

This course reinforces the IDM Program goals by

- Integrating multiple mediums of image, text, sound, physical, space, and interactivity. (Reinforced)
- Adapting to constantly changing technological paradigms by learning how to learn.
- Examining the ways commercial technology has integrated itself into our personal and social lives.
- Combining research and studio practice through inquiry and iteration. (Reinforced)
- Collaborating across disciplines.
- Adopting professional practices and tools. (Reinforced)
- Integrating Cultural Literacy
- Highlighting early communities, movements, and critical work in the digital space.

Course Objectives

Students in this course will:

- Understand the basics of creative coding in Java & Javascript (Processing/p5.js).
- Apply their understanding of digital media to software.
- Learn best practices for designing software within an event-driven, object-oriented, real time frameworks.
- Experiment with different techniques for user input and output, including sensors, sound, external APIs, and additional mediums pursued by student interest.
- Propose, develop, and publish a complete software experience as a final project.
- Leverage a new ability to deconstruct software systems while at the same time deconstructing the systems of culture and commerce that brought them into becoming a part of our day-to-day lives.

Remote Instruction and Asynchronous Learning

This semester we will have students in multiple time-zones with various needs in regards to scheduling and ability to participate in the Zoom sessions. It is my goal to accommodate and coordinate, however possible, a productive experience for you in this class and the way to best facilitate that is through coordination and lots of communication directly with individual student. All class material is available through our GitHub and Google Doc, all Zoom meetings will be recorded and uploaded directly after class, and I am available to assist with any code issues directly via email.

One of the most valuable parts of how this class operates is the chance to receive feedback and participate in group discussion in our Zoom meetings, so, I strongly encourage you to try to attend our Zoom class in real-time if possible, but if that is not possible, we will coordinate to facilitate the most value possible from an asynchronous learning experience.

Participation and “Attendance”

While attendance in the Zoom session will not be mandatory, participating in discussion about your assignments and providing feedback to your peers on their assignments is an extremely valuable part of your education. For those unable to attend class in real-time, your participation grade will be derived from coordinating with me directly and providing feedback to other students via our Google Doc.

Assessment

Assessment will occur by reviewing weekly coding assignments & challenges, project milestones, and final projects. The course is cumulative. Please see the Grading Section for specific grade breakdown, calendar for due dates, and assignments for briefs.

The assignments are your opportunity to put your skills to the test, and pursue precisely what you want to explore. They're designed to test and deepen your knowledge while giving me valuable information about how you're progressing in the class.

Please reach out to me if you find yourself "stuck" at when trying executing an idea- finished is better than perfect. An attempt at them is better than nothing at all.

Students will evaluate their own progress in a course in two ways: 1) A course evaluation form, which will be developed based on the course’s individual objectives. 2) Students will participate in a self reflection assignment around midterms and at the end of the semester, or as otherwise noted. This written evaluation is worth 15 points and to be turned into me as a separate document.

- Critically analyze/evaluate how much time was spent learning syntax & structure, programming concepts vs. actually programming, and how does this reflect on the final quality of your end result.
- Comment on your successes and frustrations with Processing and P5.js.

Course Requirements

- Contact me via email about issues or problems
- Consult the github account for the class at twice a week for up to date information and demo code
- Be considerate at all times in your communication and attitude.
- Complete all assignments by due date
- Acquire and keep up with any readings or video viewings.
- Devote at a **MINUMUM** 8 to 12 hours a week OUTSIDE of class, fulfilling homework assignments, readings and studying concepts covered in class, in order to do average.
- Have Desire amounting to Enthusiasm (to Learn and to Explore)
- Have Patience, Persistence, and Discipline
- Be Creative
- Pay attention to Detail & Craft
- Have Self-confidence and Pride in your work
- Take Risks & be Fearless in your assignments
- Think beyond the classroom
- Don’t be afraid to fail! It is a crucial part of learning. Don’t let your failures determine you. Rise again. Keep moving

Required Materials

We will utilize a variety of tools to make this class successful and to create your class projects.

- Text editor - Atom
- Processing
- P5.js Download the P5.js COMPLETE library
- Most importantly: you will basically live inside of Processing Reference and P5.js Reference. Your ability to navigate and use these documentation pages is both a crucial tool to learning, and will be expected of you for completing your assignments. Best practice: always have it open in the background as you work.

Additionally

- A Github account for yourself- https://github.com/ - We will be creating accounts in class if you haven't already. Email me your handle if you already have one (not a link to your login, but simply, your github handle or a link to your account of github).
- Learning Processing, SECOND EDITION by Daniel Shiffman ISBN-13: 978-0123944436 LP This book is available for purchase in the NYU bookstore. You can check the status of the book at the NYU library here. This book in it's initial edition is available as an online book via here.
- Generative Design by Bohnacker, GroB, Laub & Lazzeroni ISBN-13: 978-1616890773 GD This book is

available for purchase in the NYU bookstore. You can check the status of the book at the NYU libraryhere

- Make: Getting Started with P5.js by Lauren McCarthy
- Nature of Code NoC
- A notebook/sketchbook and a pen.
- Slack
- Resources

Highly Recommended Text - Form + Code by Casey Reas and Chandler McWilliams This book is available as an online book here

The above software is free to download.

Course Structure

This is a hybrid course - both lecture and studio, both online and on campus(physically). Students will work on individual projects and group collaborations. Work will be completed both inside and outside of the class. Class will be a mix of lectures, class discussion, Show & Tell, workshops/challenges, studio time and critiques. Guest critics may be present.

The course can be broken down into 3 sections throughout the semester: programming basics, Object Orientated Programming (OOP), and advanced topics (behaviors, using addons - JSON, API's, Computer Vision, Sound, etc).

Important Aspects of this Course

- **P5.js Homework projects will be turned in via OpenProcessing.org.** Use code xxxxx to join our class site there. At times, larger assignments homework projects will be turned in via GitHub. GitHub is a web hosting service that supports Git version control. It is more important than your resume, whether you are an artist, engineer or designer. It's used largely for software, but can be used for anything. It's important to become familar with it, develop good habits. See the assigned reading links for more information.
- **Time** Time management is everything. You will need it to have reasonable accomplishments in this class. The material is cumulative. The readings are essential. Writing code is the only way to make progress and to fully synthesize the material. It becomes evident quite quickly. Develop excellent time management. You will need 8-12 hours outside of class to succeed.
- **Pair programming** is a popular way to learn and to work on projects. Two people share one computer and write code together. For the second half of class, an exercise will be given which will be worked on in pairs. The exercise will be due at the end of the class period and presented to class. On occasions the deadline will be at another point. Yet, the exercise is meant to be worked on together - on one laptop, in one program.

- Projects are not always group projects.

Plagiarism

It's super easy to find code online. As tempting as this is, I am going to ask that you write your own code for the first 2/3 of the semester. This will help you grasp and retain the fundamentals.

It does help to write the code from examples out, line by line.

Code that is not written by you can not be submitted as an assignment.

If you use code from elsewhere (which you will at times), I expect you to cite the work and author, as well as to comment each line, as to what each line does programmatically. Do *not* summarize several lines of code from a high level. 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's bending these concepts to your idea, especially graphically. It's not a copy and paste job.

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.

Show and Tell

Occasionally, you will be asked to present about a piece of digital art, design, or technology you find fascinating. Please research the assigned artist/designers/firms/pieces/collectives in advance. Each presentation will be about 5-7 minutes long. Be able to address the following, along with *3-4 visual examples of each individual or team's work*:

- What is original/unique about how this artist uses technology?

- What kind of code did the artist use? (JavaScript? Flash? Procesing? openFrameworks? VVVV?) How are they using the code? What is the code achieving?

- Why is it necessary that the artist use code? (ie: particularly if it’s non-interactive, could it have been achieved with traditional or linear media?)

- Your presentation must be prepared and professional. DO NOT USE TEXT ON YOUR SLIDES, other than the artist's/designer's/firm's names. Show. the. work. Discuss. the. work.

Tips for your presentation

- Do not present to me. Look at your audience, your classmates. You are presenting to them.
- THOUGHTS ON PUBLIC SPEAKING BY ZACH HOLMAN Click the links!!!
- Create a 5-7 minutes presentation (pdf, powerpoint, keynote, quicktime, etc.)
- Decide whether or not you want to mirror your presentation or not (if you're using presentation tools - ie, notes) Get familiar with your display preferences in advance!! I'm not kidding either. Borrow a cable from the student aids outside Tammy + Denise's office by the kitchen area, and try it out on your own BEFORE class. Seriously. It makes all the difference in the quality of your presentation and presentation nerves.
- If you have sound during your presentation, set your external sound preferences in system preferences to Crestron. Test this out on the projector before your present. See above.
- PRACTICE PRACTICE PRACTICE

Grading

- **20%** – Participation: Show & Tell, Self-Assessments, In-class Challenges
- **30%** – Sketches
- **15%** – Midterm
- **5%** – RoundTable Presentation
- **5%** – Final Project Milestone 5% (either early or technical)
- **25%** – Final Project: Choice (critiqued)

For each project your grade will be assessed upon the following: Principles, Creativity/Thoughtfulness, Craft. Below gives you a sense of the spectrum, giving examples for high, middle and low grades.

Please note: A letter grade of "C" is an average grade. Average is not bad. It means average. Most of us are average, in some way. An average level of work is being done. It does not mean perfection. It means that the work still has room to develop at it's current state. It also means that the work has noticible achievements and evidence of some acquisition of knowledge. I look at this on an individual basis as well as across the whole class.

Please read examples of each assessment point below to get an idea of my expectations. Note that a "B" is between Satisificatory and Superior.

Principles:

- Superior - A - Your work shows evidence and understanding of programming concepts discussed in readings, lectures, and exercises, as you bend them to your will.
- Satisifactory - C - Your work shows evidence of concepts and is still developing an understanding of course material. You understand some aspects, but aren't utilizing them fully. Usage is cursory. Keep pushing your work and review the material to revisit how you can integrate it to your work.
- Unexceptable - D/F - Your work shows some evidence of concepts discussed, but lacks key understanding, confidence, robustness and authority. Aspects are lacking. Reviewing course material is required. Ask questions in class. Manage your time better.

Creativity, Thoughtfulness:

- Superior - A - Your work demonstrates your personality and a great depth of engagement with the material. It's extremely evident that you are thinking, exploring, playing and taking risks. You are creating wonderful experiences.
- Satisifactory - C - You are executing your ideas, but more time is needed to consider more deeply about what is conceptually and physically happening. They are barely getting off the ground. Or have large bits that are broken. Do more research and exploring. Play.
- Unexceptable - D/F - Your work is so straightforward that it's flat. It's barely coded (or badly broken). It's copied from elsewhere and not expanded upon. Question and Iterate your work to push your it further. Read. Play. Get off the lame track and get inspired.

Craft:

- Superior - A - Your work shows delicate care and consideration to presentation and professionalism. You code is neat, clean, commented and structured. Your friends consider you "Type A." Your style is evident.
- Satisifactory - C - Your work shows the birth of your ideas, but further time and iteration can really push your work to excel. Your code is there, but messy. It can be simplified and made cleaner. What you put into it, is what you get out of it. Practice makes perfect!
- Unexceptable - D/F - Your work is rushed and looks like it was done on your train-ride in or the night before. Make your work something you are proud of. You are here to build your portfolio after all, aren't you?

Participation will be based upon: Professionalism, Engagement, and

Improvement

Professionalism:

- Superior - A - Always arrives on time; assignments turned in properly and on time; respectful of others in class and gives feedback appropriately.
- Satisifactory - C - Usually arrives on time; most assignments turned in on time; listens to others.
- Unexceptable - D/F - Often tardy; turns in assignments late; fails to prepare for class; in attentive to instructor or other students

Engagement:

- Superior - A - Always contributes appropriately to class discussion; frequently offers to demonstrate technique; shows leadership in group projects.
- Satisifactory - C - Usually contributes to class discussion; has demonstrated technique; participates actively in group projects.
- Unexceptable - D/F - Does not participate in class discussion; no evidence of technique; fails to contribute adequately to group projects.

Improvement:

- Superior - A - Shows an exceptional and growing understanding of technique; builds on previous lessons; accepts critique and makes proper adjustments.
- Satisifactory - C - Technique is developing; has shown some ability to build on previous lessons; generally able to accept critique.
- Unexceptable - D/F - Little or no development of technique; is unable or unwilling to accept critique; unable to make adjustments.

Late Work

Unless specified, work is due the BEFORE following class period.

Work that is turned in after class but still on the due date get's 1/3 letter grade deduction. Work that is turned in the same week, get's 2/3 grade deduction. Subsequent submissions get one letter grade deduction for ever week that it is late, beyond the first week. The lowest possible grade a late assignment will be given, will be an "F". Deductions are calculated AFTER the initial grade is given.

If work is turned in beyond 4 weeks, the work must still meet the requirements for the assignment, or be an attempt in that direction. Work submitted beyond 4 weeks that does not make an attempt to meet the requirements will not be considered and a zero will be given for that assignment.

Assignments that are not submitted will receive a "0". Please realize that a "0" mark is a fantastic way to tank your grade. Getting an F is far better than getting a zero.

The numerical breakdown for letter grades is as follows:

- A 100-90 Excellent
- B 89-80 Very Good/Good
- C 79-70 Satisfactory
- D 69-63 Poor; Below Average
- F 63 & below Unacceptable

The NYU Tandon School of Engineering Grading Policy can be found in the current Course Catalog.

Academic Integrity

Violations of academic integrity are considered to be acts of academic dishonesty and include (but are not limited to) cheating, plagiarizing, fabrication, denying other access to information or material, and facilitating academic dishonesty, and are subject to the policies and procedures noted in the Student Handbook and within the Course Catalog, including the Student Code of Conduct and the Student Judicial System. Please note that lack of knowledge of citations procedures, for example, is an unacceptable explanation for plagiarism, as is having studied together to produce remarkable similar papers or creative works submitted separately by two students, or recycling work from a previous class.

Please review [NYU's School of Engineering's academic dishonesty policy(https://www.nyu.edu/about/policies-guidelines-compliance/policies-and-guidelines/academic-integrity-for-students-at-nyu.html) in its entirety. Procedures may include, but are not limited to: failing the assignment, failing the course, going in front of an academic judicial council and possible suspension from school. Violations will not be tolerated.

All work for this class must be your own and specific to this semester. Any work recycled from other classes or from another, non-original source will be rejected with serious implications for the student. Plagiarism, knowingly representing the words or ideas of another as one’s own work in any academic exercise, is absolutely unacceptable. Any student who commits plagiarism must re-do the assignment for a grade no higher than a D. In fact, a D is the highest possible course grade for any student who commits plagiarism. Please use the MLA or Chicago Manual style for citing and documenting source material.

This includes copying code for other sources, using code from other sources with only slight modifications and using code from other sources without a reference.

Educational Accessibility Statement

NYU-Tandon is committed to assuring equal educational opportunity and full participation for all students. The mission of the Office for Students with Disabilities is to provide individuals with learning differences (a.ka. disabilities) the same access to programs and activities as other students. We assist students to maximize their potential while helping them develop and maintain independence.

Students who believe they are eligible for course accommodations under the ADA or Section 504 or have had accommodations please contact New York University's Moses Center for Students with Disabilities at 212-998-4980 or mosescsd@nyu.edu. You must be registered with CSD to receive accommodations. Information about the Moses Center can be found at http://www.nyu.edu/csd. The Moses Center is located at 726 Broadway on the 2nd floor.

Faculty can provide course accommodations/modifications only after receipt of an approved accommodations letter from the Moses Center for Students with Disabilities. Accommodation letters can be provided to qualified students at any time during the semester, but grades earned before the faculty receives the letter cannot be changed.