Generative Art Online

(aka Creative Coding Online : MADD 23632)

Instructor:	Nick Briz (nbriz@uchicago.edu)
AI-TA:	netnet
When:	Tue : 12:30 PM - 03:20 PM
Where:	Logan Center 802
Office Hours	Tues / Weds (book online) in-person at the MADD Center

Course Description

Though the web was originally conceived as an online space for sharing hyperlinked documents, the modern Web browser has evolved into a creative coding playground capable of producing all manner of networked art and algorithmic compositions. In this course we'll learn JavaScript, the Web's defacto programming language. Throughout the quarter we'll experiment with various different Web APIs for creating generative and interactive Internet art including the DOM (for interactive/dynamic content creation), Canvas (for 2D/3D animations) and Web Audio (for algorithmic sound/music). We'll learn how to produce work that responds to various input sources (trackpad/mouse, touchscreen, keyboard, cameras, microphones) and how to fetch and incorporate data from external APIs elsewhere on the Internet.

Learning Goals

- Foundational understanding of web programming concepts, tools and working knowledge of the JavaScript programming language.
- General understanding of the history and theories behind Generative Art as well as core principles of its practice.

Class Materials

In order to participate in this course you will need to have a decent computer (desktop or laptop with 8-16GB of ram or more) and a modern Web browser like Firefox, Brave, Chrome or others (**do not use** Internet Explorer or Safari, those are subpar browsers). Personally, I will be demoing things in class using Firefox.

You will also need to create a free GitHub account, this is where you'll be uploading your projects (the actual code) before submitting them on Canvas. (*if you are new to GitHub, consider signing up for the GitHub Student Developer Pack*)

You will also need a code editor. For the majority of the quarter I will be demoing concepts and creating examples/sketches using netnet.studio. This is the tool we will use for all of our initial "sketches", however for your final assignment you will be free to use any code editor you'd like. If you're a beginner to code, I would still recommend using netnet.studio (which I designed specifically for beginner creative web coders), however if you have familiarity with a modern code editor like Pulsar, Sublime, or VSCode you can use any of those if you prefer (so long as you know how to get your work published on GitHub using those editors)

Class Discussions

So much of what we're going to cover in class, both in terms of the theory and practice, can be gleaned through your own online research. The most valuable aspect of learning this material in the classroom, rather than on your own, is the chance for real-time interactivity with your professor and peers. I can not stress enough how important it is to take advantage of class discussions. These can be technical discussions (about how the Internet and the Web work, about coding tricks and techniques, and/or any other topic relating to the technology and craft we'll be covering), theoretical discussions (about any of the concepts and ideas introduced in assigned readings and/or addressed by any of the art referenced throughout this course) and/or historical discussions (about any of the various histories we'll be covering this quarter).

If you need to miss class for a legitimate reason (medical or family emergency, professional opportunity, etc), send me an email ahead of time so that I can mark it as an excused absence. Attendance and participation in class is **25% of your final grade.**

Assignments + Evaluations

6 Meditations

Each student will be expected to complete and submit 6 online generative art “meditations” on the dates specified below. These assignments are creative coding experiments, first drafts with potential for further iteration. These will be created on netnet.studio and saved as a “project” (ie. uploaded to your GitHub account as a repository aka “repo”). These should be published online (netnet.studio uses GitHub pages to host your work online and generate its URL) the web address (URL) for which should be submitted to the class canvas.

These projects will be based on the demos conducted in class. Each week (weeks 2 - 7) I will be introducing new techniques and concepts, often building on the previous weeks, and creating code sketches together with the class, often in response to a historical work of generative art. A meditation (the project you submit) can be a variation of something I demonstrated in class or an entirely different implementation/exploration of the techniques and ideas discussed in class that week.

In my evaluations I will be reviewing both craft (how you’ve written your code) and concept (the idea behind the sketch), I will either mark the assignment as “complete” or “incomplete” on canvas, if/when I mark an assignment as “incomplete” I will leave feedback with clear request for changes you will need to make. Once you’ve made those changes (and updated the code on your GitHub) **do not re-submit the assignment,** instead leave a canvas comment in your submission letting me know of the update and that I should re-review the work. I’ll then either mark the assignment as “complete” or leave more feedback.

These “meditation” assignments are worth 50% (8.33% each) of your final grade and must be “complete” in order to receive full credit.

1 Final Project

The final project will be a finished, exhibition ready, work of online generative art. It should extend the work you've created earlier in the quarter, by either combining and/or iterating on one (ore more) of your prior assignments. Unlike your meditations (which are open ended explorations of generative art concepts and techniques) your final project needs to have a stated goal, context and title (Details will be discussed in class). **The final is worth 25% of your grade**, to receive full credit the work will be submitted on canvas as a published URL (just like the prior meditation assignments) and must also be presented in class on either week 9 or 10 (finals week) for an in-person ~15 min critique. *NOTE*: if a final requires alternative hosting options you are welcome to host your project elsewhere (so long as it has a publicly accessible URL), in these instances you should submit both the public URL as well as the URL to your GitHub repo (so I can review the code).

Course Schedule

week 1: 03.25.2025: **Introduction**
We'll discuss the Internet (and in particular the Web) both as a medium and platform for creating art, we'll also discuss the history of Generative Art and how we'll be engaging with it's history, theories and practice in this course.

week 2: 04.01.2025: **10print**
We'll discuss early generative artists like Vera Molnár and the role "randomness" played in their work. Through a close reading of one of the most iconic works of generative art, 10print, we'll introduce the core concepts of programming as well as a couple of the principle techniques for creating generative systems, namely randomness and recursive animation loops.

week 3: 04.08.2025: **Emoji Ink**
We'll discuss the genre of software art or "artware" and take a close look at the piece emoji.ink by tig.ht (Yung Jake and Vince Mckelvie). We'll learn how to make interactive works by introducing the paradigm of event based programming. We'll also discuss the concept of software libraries and frameworks.

week 4: 04.15.2025: **A Tribute to Heather**
We'll continue to build on the generative art techniques introduced in the weeks prior to create algorithmically generated (2D) visual compositions through a close reading of A Tribute to Heather by Evan Roth.

week 5: 04.22.2025: **the Cut-up technique**
We'll discuss "data driven" online algorithmic works as well as a number of different ways for producing them, including how to incorporate data like date/time, the weather and news feeds using third party APIs and machine learning language models.

week 6: 04.29.2025: **Musikalisches**
We'll discuss how musicians (both pre/post digital) have used generative systems to create music and learn how to use the tone.js library to create our own algorithmic musical compositions.

week 7: 05.06.2025: **the demoscene**
We'll discuss the demoscene, a folk or "outsider art" scene of graphics programmers. We'll learn how to use three.js, one of the libraries to come out of this creative community and one of the most popular for producing 3D visual compositions online.

week 8: 05.13.2025: **final project: studio session**
week 9: 05.20.2025: **final critiques (session I)**
week X: 05.27.2025: **final critiques (session II)**

Plagiarism

Plagiarism of concepts, code, compositions, samples and/or other elements is strongly encouraged, so long as you leave clear attribution within your code via comments. Ensure that anything you copy is in some way transformed, either by creating a variation on the copied elements or combining those elements with other copied elements. NOTE: transformation/combination (however subtle) is not a substitute for attribution, but rather a requirement for all copied elements.

AI Policy

We've entered the era of "machine learning" and generative AI (not to be confused with generative art). These algorithms are having (and will continue to have) drastic effects on every aspect of our society (including art). Today, artificial neural networks trained on troves of data (which are not always ethically sourced) can make "predictions" and create "hallucinations" that would have seemed like impossible sorcery just a few short years ago. In certain high stakes applications this can save lives, but it can also destroy them. In other contexts this biased hallucinatory predictive sorcery can be quite exciting, as is the case with media art. This technology, like many others that came before it (smart phones, the Internet, the computer) will most certainly change everything in our field, exactly how and to what extent is still anyone's guess. In the interest of collectively learning how to leverage its promises and minimize its perils, I encourage anyone interested to experiment with AI so long as you are considerate, deliberate and transparent with your use of these tools.

Incorporating AI models in our work
A couple of different times throughout the quarter we will discuss how to incorporate AI models as one (of many) components in our generative art works. Details about how/why will be discussed in class when those tools/topics come up.

Leveraging AI models to create our work
In this class the most likely use of generative AI will be code generation/evaluation using LLM (large language models). You are encouraged to use these in ways the supplement your learning, rather than impede it. To ensure that is the case I ask that you adhere to the criteria listed below:

- **Ask for help, not for answers.** Do not ask the AI to complete your assignment for you (for reasons which I hope are more than obvious) instead ask it the same sort of questions you would ask a professor or TA. Meaning ask it for help, not for work.

- **Chat with it like a person.** Use this as an opportunity to practice "systems thinking", often times detailing out your question helps you figure out the issue, if not at least better understand your own system. Chat as though you were posting to a coding forum like stackoverflow. Use it as an opportunity to review the technical terminology we've been

learning in class. Don't say things like "not working, fix it" while it's incredibly impressive that LLMs manage to do so much with so little input, short/ambiguous won't lead to the best output from the model nor will it serve you in understanding your own issue.

- **Here is a sample starter prompt:** *Hi, I'm a college student learning to program for the first time in a generative art class. We're creating work for the Internet using JavaScript, Web API's and libraries (name any libraries you're using). I need your help solving a problem I'm having, I don't want you to solve it for me, I want you to explain step by step (with small code snippets for context) how I can address the issue I'm having. If ever I use incorrect terminology, please correct me. If it seems like I'm misunderstanding a core concept based on the way I've presented the issue please point that out as well...*

- **Include documentation,** when working on a project that includes JavaScript libraries, it's best to include documentation for this library in your prompt. Most LLM's allow you to upload additional files with your query which the LLM can then analyze and use in generating it's response. While most LLM's can answer library specific questions (especially when those are popular libraries discussed extensively online and thus in the model's training data) it helps if you mention which version of the library you're using and even more if you include documentation directly. For example, when working with the nn.min.js library include this file in your initial prompt.

- **Never copy+paste, always transcribe.** Once you become fluent in a programming language, then copy and pasting becomes an efficient method of collaborating with an LLM, but when you're learning a new language it's important to manually type out any code the LLM generates/recommends, this will help you: better understand the code, spot potential issues and avoid copying too much code. If the code is written in a style you don't like, take that as an opportunity to rewrite it as you transcribe it into your work. It's ok to make mistakes, this is how we learn to debug (which is a core aspect of programming).

- **Never transcribe any code you don't understand,** if an LLM generates some code that is new to you or otherwise confusing, always ask it to clarify and explain the code until it makes sense, only then should you transcribe it into your project.

- **When you've finished a project, consider sharing your code with an LLM and asking it for feedback**, be specific about what you want feedback on (coding style, clarity, efficiency, creativity, etc).

- **Submit your chats with your homework.** Any conversations related to a project that you have with an LLM should be submitted alongside the assignment on canvas as a "comment" under your submission, this can be done by either generating a share link on ChatGPT or using https://aiarchives.org