Internet Art II

MAAD 23632 1 (Winter 2022)

> *The Web represents a grand emotional, sensory, and intellectual adventure for anyone willing to explore it actively. [...] For artists, ignoring the imperative to grasp the cultural implications of the Internet means risking irrelevance. [...] As human discourse adapts to its new home, everything we do and think as human beings will be and is being shaped by new values. [...] If it's ever fair to say that anything has "changed everything," it's fair to say so about the Internet.* — Virginia Heffernan

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 HTML5 video, Canvas (2D/3D animations) and Web Audio. 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. This course counts towards the Media Practice and Design requirement for the MAAD program.

where:	Crerar Library 134 ;
when:	Mon, 09:30am - 12:20pm ;
professor:	Nick Briz ;
ta:	netnet ;
email:	nbriz@uchicago.edu ;
office hours:	*by-appointment-only* ;
class website:	http://netart.rocks/uchicago/netart2 ;

Learning Goals

- Foundational understanding of web programming concepts, tools and working knowledge of the JavaScript programming language.
- General understanding of the web's creative potential, by learning how to produce interactive and generative compositions using the browsers native APIs (which may include the DOM, Canvas, WebGL, WebXR, WebAudio among others) as well as through creative libraries (which may include p5.js, three.js, tone.js, aframe among others)

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, Vivaldi or others (**do not use** Internet Explorer or Safari, those are subpar browsers).

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, I will be using Atom in my tutorials, but you're welcome to use any modern code editor like Sublime, or VSCode (if you're using something other than these 3 editors run it by me for approval first). For absolute beginners (those of us that have never once written any code before) I would recommend using netnet.studio instead (which we'll discuss in more detail in class).

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 the online lectures and/or addressed by any of the Internet 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

Each student will be expected to complete and submit 3 assignments on the dates specified below. These assignments are creative code sketches/experiments/projects uploaded to your GitHub account, each as it's own repository (repo). The URL for these repos will then be submitted to the class canvas by the due date for that corresponding assignment. There are a series of different assignments to choose from based on your interest and technical proficiency. Each of these assignments has a corresponding lesson on the class website which you should complete prior to starting on the assignment. These lessons may include a combination of online notes, readings, video lectures, tutorials and interactive code examples. I will be meeting with everyone on the second week of class to formalize an individual plan (which lessons and assignments each student will be working on). Each of these assignments are worth 25% of your final grade.

due dates

- Assignment #1: **Jan 30**
- Assignment #2: **Feb 20**
- Assignment #3: **Mar 06**

assignment options

CSS Art
can apply to assignment #1

The goal of this assignment is to introduce you to code and get you oriented with the basics of creative coding for the web. If you've never written any code before (ie. you are an absolute beginner) consider making this your first assignment this quarter. Make sure to review the material in What's Internet Art? (from net.art to CSS Art) before starting on this assignment.

Code Poetry
can apply to assignment #1

This assignment has 2 goals. The primary goal is to make sure you understand how to fork and modify projects on GitHub. The secondary goal is to learn to see our code in a new light, not merely as instructions to the computer but as an aesthetic space for creative expression itself. If you've never used git or GitHub before, consider making this your first assignment this quarter. Make sure to complete the lesson Versioning (Open Source Code Poetry) before starting on this assignment.

10print
can apply to assignment #1 or #2

The goal of this assignment is to make sure you're comfortable with the core programming concepts in JavaScript, specifically how we work with data (declaring and reassigning variables) and how to control the flow of a program (writing statements and expressions like loops, conditionals and creating functions). If you already know how to work with git/GitHub and you have some prior experience coding (HTML, CSS, C, python, etc) but are new to JavaScript, consider making this your first assignment this quarter. If your first assignment was either the CSS Art or Code Poetry assignments, consider making this your second assignment. Make sure to complete the lesson Randomness (Generative Art in JavaScript) before starting on this assignment.

Form+Code
can apply to assignment #2 or #3

The goal of this assignment is to create an algorithmic (visual) composition where the same code produces slightly different visual outputs each time it's run (ie. each time we refresh the page and/or as we interact with it). We'll do this by learning how to use the web browser's Canvas API (either using the native API directly or via the p5.js library) for creating generative 2D drawings. Make sure to complete the lesson Algorithmic Composition (audio + visual), specifically the first section on Form + Code, before starting on this assignment

algoMuzak
can apply to assignment #2 or #3

The goal of this assignment is to create an algorithmic (musical) composition where the same code produces slightly different musical outputs each time it's run (ie. each time we refresh the page). We'll do this by learning how to use the web browser's Web Audio API. Make sure to complete the lesson Algorithmic Composition (audio + visual), specifically the second section on algorithmic music, before starting on this assignment

artware
can apply to assignment #2 and #3

For more advanced students there is an alternative 3rd assignment which is fairly open. The goal of this assignment is to produce "artware" (or software art) experimenting with any number of browser APIs, 3rd party REST APIs and libraries including the browser's DOM API, the Events API and the Fetch API as well as learning the basics of writing server side code, by creating your own server and REST API (using a framework called express.js) as well as experimenting with the socket API (using a library called socket.io). You can begin watching the tutorials in the the lesson Server Side: artware, surveillance, telepresence, each of which build on the previous one, and each offering different directions you could take this final assignment in.