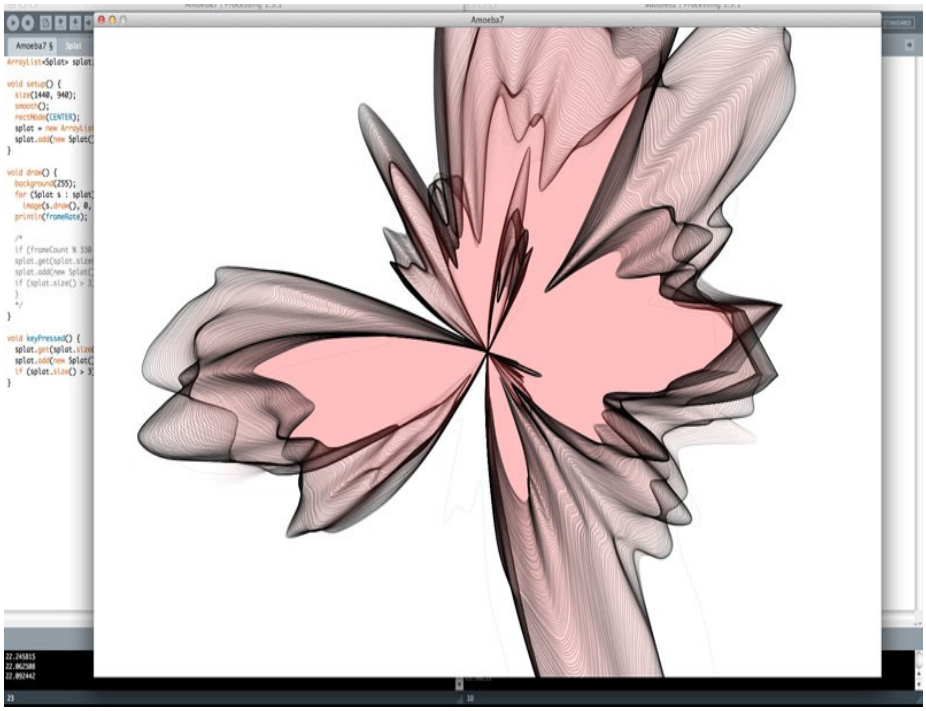
[Workshops, free software, and art]



The free and open source software movement has radically changed long-standing assumptions about the production and accessibility of technology-inflected art. Scantly resembling the last century's cliché of the artist as the brilliant loner closely guarding the technique of his secret genius, the art tech scene of recent years has exploded with large and vibrant communities self-assembled around common principles of sharing, collaboration, riffing on each other's contributions, and mutual aid over online platforms like GitHub, OpenProcessing, StackOverflow, and many other forums. This mode of artistic production confuses traditional notions of authorship, blurring the distinction between toolmakers and tool users, between artists and engineers, and between technology and aesthetics.

Once the esoteric domain of a scarce few scientists and engineers huddled over punch cards and assembly language, code has become an increasingly accessible medium. Artists have responded by collaboratively constructing free and open source platforms like Processing, OpenFrameworks, vvvv, PureData, and SuperCollider. Collectively these communities have an unstated goal of reducing the barrier to entry for aspiring technology artists, leaving aside only the limits of their imaginations. This is underscored by contributed libraries that facilitate the creative re-appropriation of formerly obscure topics like computer vision, artificial intelligence, robotics, and many more.

The goals of technology artists are ambitious and diverse, including opening critical lines of inquiry into emerging technologies, ubiquitous internet, and digital culture, constructing immersive installations, building novel instruments for music and performing arts, and countless others.

[Workshops]

The principal goal of workshops is to introduce coding as a medium for artistic expression. They are structured depending on the timeframe they cover and the makeup of the participants, and are aimed at both beginning and intermediate practitioners, as well as both to artists wishing to learn technology and to engineers wishing to apply their technical skills to creative

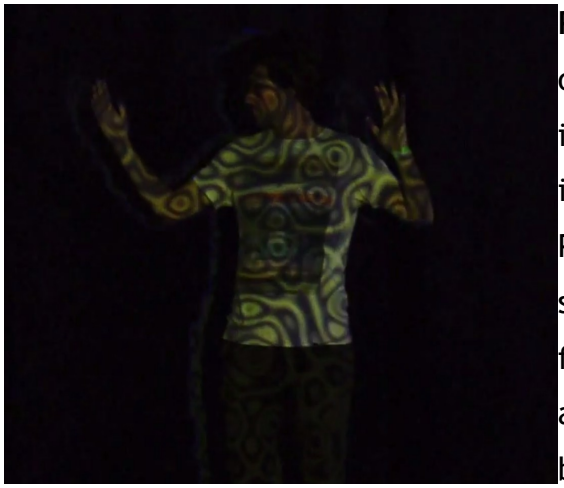
Short-term intensive workshops (1-2 days) present an overview of the field, assessing influential works by various artists and engineers, and giving a survey of the platforms and tools available, emphasizing Processing in particular. Participants will learn very quickly how to prototype simple generative graphics, and will also learn how to investigate deeper into areas which are of interest to them.

Long-term workshops (3-7 days, spaced out over a longer period) go into more technical depth. In addition to the qualitative content offered in the short-term workshop, there is a longer programming component. Students learn hands-on various strategies for generating generative content, integrating libraries and outside media, and creating responsive interfaces and instruments for live performance.

[Topics]

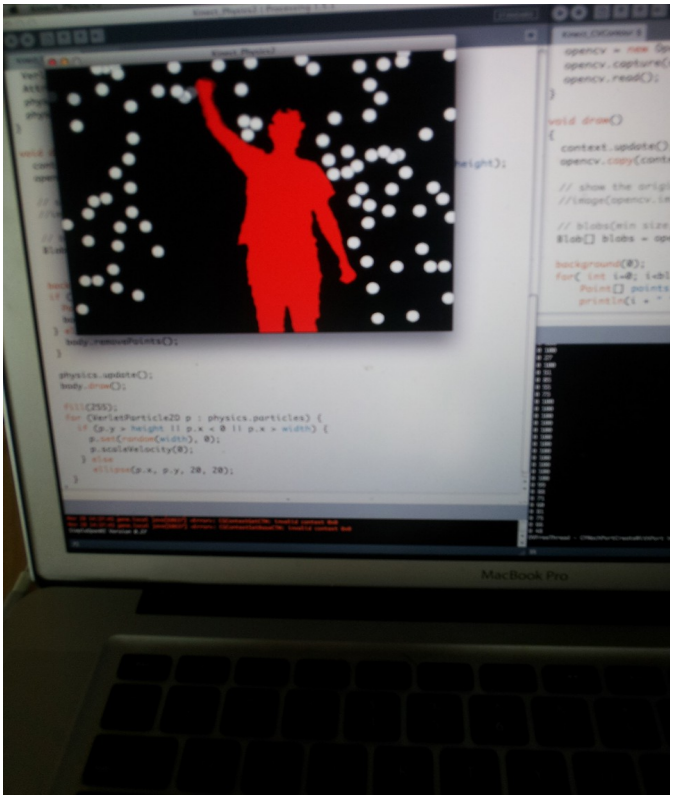
Long-term workshops include inquiries into specific areas of application, which depending on the skill level and interests of the students, can involve any of the following topics:

Computer vision: Techniques learned include tracking people, faces, and hands. The data is used to drive real-time interaction within art and sound installations.



Projection mapping: How to apply projected light onto untraditional untraditional 3-d surfaces, and its uses for installation artists, filmmakers, and interaction designers. Includes a demo of Kinect Projector Calibration Toolkit, an open source software package initiated by the instructor which facilitates “body mapping,” the experimental application of projection mapping onto moving bodies detected using a Kinect depth camera.

Performance instruments: Creating responsive and intuitive instruments for on-the-fly creation of generative graphics to be used in music, dance, theater, and other performance contexts.



Generative art: Automatically rendering graphics and sound by algorithms inspired from nature, mathematics, biology, and computer science.

Experimental geometry: Creating and manipulating 3d meshes of novel geometric forms, and fabricating them using 3d printers.