Computational & Studio Practice

California College of the Arts
UDIST-3120 Computational & Studio Practice
January 21 – May 7, 2020
Tue/Thu, 7:15-10:15pm
Oakland/Founders, Room 304 (Hybrid Lab)

Instructors: J.D. Zamfirescu-Pereira jzamfirescupereira@cca.edu
Barney Haynes bhaynes@cca.edu
Office hours: by appointment

Goals

The discipline of creating with computational media, sometimes called "computer science" or "software engineering", enables students to create new tools and reprogram others to further new thinking in their own major disciplines. Students in the Computational Practices minor develop a competency in creating with computational media; this course integrates that disciplinary competency with students' own studio practice discipline.

Intended to be a capstone for the minor, this course supports students as they explore the possibilities and consequences of applying their skills in creating new technology to their work as artists and designers.

Understanding and situating computational practice in the broader context of a particular artistic or design discipline is critical for producing work that eloquently blends the technological and the traditional. Case studies will examine how and why particular computational methods are chosen to create particular works and question established orthodoxies. Students will complete a final project that represents the culmination of their work in the minor, demonstrating technical sophistication as well as a deep understanding of the role and context of technology in their discipline.

Assignments include deconstructions of prior work and a "literature review" that collects articles, prior works, etc., with research into what's been done and what are the opportunities and the pitfalls. The final project should take this research into substantial consideration to produce a novel piece.

Expect to have fun, and expect to be challenged. Expect to think in ways you might not be familiar or comfortable with, and expect to find yourself more capable than you thought.

Topics

Topics will span advanced concepts in programming and electronics; individual topics covered will depend on student interest. Tentatively, we plan to cover:

- AI/ML: Use & training
- Mechanisms & fabrication
- Structuring more complex code
- Advanced debugging — web/Arduino/node/python
- Communicating technical content
- Advanced electronics

Class Repository

https://github.com/zamfi/cca-computational-studio-practice-spring-2020

The class repository will be used to assign homework and reading assignments, and will contain useful code, libraries, and links to other resources. Github allows you to receive email updates when a repository changes — take advantage of this feature!

Required Textbook

No textbook is required, but there are many good online references and a few good books out there we can recommend — it might be useful to get a book as a reference.

Prerequisites

- 200 and 300-level (or equivalent) experience with programming & electronics. If you are unsure, you should find this diagnostic test not especially challenging: https://git.io/vinuK
- Algebra. You should be comfortable with simple equations, manipulating numbers, fractions, solving for variables, and the engineering prefixes (micro, milli, kilo, mega, etc.)
- Some understanding of geometry & trigonometry will also be helpful.

Requirements and Expectations

- **Always bring your computer to class**. If you don't have a computer, check one out from the Media Center. You will need a computer to participate, and your grade will be negatively affected if you don't bring one.

- **Participate in class!** Ask questions, guess answers, propose topics, share interesting projects you've found, push the envelope, explore your interests, and teach us! There are no stupid questions; admitting when you don't know something should be a point of pride. Chances are you're not the only one with a question, just the bravest one. *As outlined in the CCA Student Handbook, attendance in class is mandatory and three or more unexcused absences will result in a failing grade.*
- **Plan to spend 3-9 hours a week on your project.** If you don't have a solid foundation in math you may need more time. Plan ahead!
- **Submit your homework on time.** Show your work in homework and exams to receive full credit. Write clearly and legibly. Attend class, and don't be late. These are not requests, these are requirements.
- **You are responsible for checking your email** and the class repository for updates.

Access & Wellness Services

CCA says:

> Students with disabilities, including disabilities that are not clearly evident like chronic diseases or learning disabilities are encouraged to notify their instructor after class or during office hours. CCA will make reasonable accommodations for persons with documented disabilities. Students should contact Suzanne Raffeld, Director of Access and Wellness Services (email: sraffeld@cca.edu; phone: 510.594.3775), to answer any questions or for assistance. For more information, consult CCA's webpage at: http://www.cca.edu/students/resources/disability.

We want you to succeed in this class. Please make use of the Hybrid Lab coaches, the Learning Resource Center, and talk to us if you feel you are

struggling with the material — we can help you do better in this class, but only if you come talk to us!

Tentative Course Outline

Final Project Proposal	February 6
Proof of Concept -- exploration & research finished	March 6
Working Prototype	April 2
Final Critique & Feedback Session	April 23
Final Show	May 9 — **ONLINE**

In addition, throughout the term, we will discuss and assign homework on specific advanced topics.

Grading

We will consider the various components of the class in roughly the following proportions:
70% Final Project
20% Other Assignments
10% Attendance & Participation

Grading Rubric

A: Excellent. You've exhibited exemplary conceptual, technical and perceptual ability implementing projects. You've demonstrated a deft understanding of required readings, and ability to successfully communicate ideas and processes to others. All work is commented and

clearly demonstrates understanding of each lesson.
B: Good. You've completed assignments, and demonstrated a grasp of most of the main aspects of each lesson, but not all. You're able to communicate information, and step by step processes well. In projects, conceptual, perceptual, and technical skills are present.
C: Satisfactory. You've completed the assignments but may lack enthusiasm or drive to push the work into detailed display of comprehension. You have not demonstrated comprehensive knowledge of the application or programming environment.
D: Unsatisfactory. You have not completed the work as assigned. Substantial problems exist in your work.
F: Fail. You did not submit work, or work is below unsatisfactory level.

Words of Advice

Exploration is a key part of this class. You'll get more out of your work if you give yourself extra time and have patience. Sometimes you'll hit a dead end and have to start over — don't despair, it happens to everyone, but give yourself extra time just in case.

This class should be both fun and intense. It's most fun when you enjoy what you're working on, so make a point of taking some time to explore projects you make want to work on in the future!

CCA Learning Outcomes

Students who successfully complete their UDIST requirement will be able to:

- Students can articulate/demonstrate a basic understanding of the methods (learning/working/processing) used in the primary disciplines addressed by the course

- Students can integrate multiple studio practices, combining the knowledge and insights from the disciplines, processes, and media addressed by the course to create original work
- Students can articulate/demonstrate how an interdisciplinary approach is a methodology in itself and thus represents more than simply combining different mediums/disciplines for effect
- Students demonstrate the ability to work productively in groups, recognizing and utilizing the shared knowledge made possible through collaboration

Courses in the Computational Practices minor have additional learning outcomes; this course addresses the following two:

1. Students should have a conceptual and practical knowledge of programming for screen-based and physical computing, and the ability to apply that knowledge to individual art/design practice.
2. Students should be able to communicate and collaborate on technology-reliant projects with other artists and designers, as well as potential peers or professional collaborators with technical backgrounds.