

Fundamentals of Procedural Media

USC School Cinematic Arts, CTIN 101, 2 units

1 to 3:50 Thurs

Professor	Student Assistant
Peter Brinson	Christian Bingham
brinson AT usc dot edu	cbbingham AT usc dot edu
Office Hours: Thurs 11-1pm or by Appointment (SCI 201 F)	Office Hours: Mon 3-6pm; Wed 1-4pm (SCI 113)

Course Description:

This course is an introduction to the practice of the creative coder. Today, computation - by way of software - touches nearly every human endeavour, and, in particular, has profoundly altered creative production and dissemination. But what distinguishes interactive media from other expressive forms is that it is not just made with software, it is software. The creative coder explores the computer's unique expressive potential by harnessing its unprecedented ability to execute rules. This will be our practice - writing instructions to make complicated systems out of simple processes. Further, the creative coder writes code that creates meaning and representation, as opposed to the film director, the playwright, and the novelist, who author the representation itself. This course is for the game designer, 3D modeler, interactive writer, digital musician, and installation artist to become procedurally literate.

Therefore, this class involves coupling a structured orientation of the programming language, Processing, with a weekly examination and discussion of relevant computational media and texts. In particular we will investigate what aesthetics and representations lend themselves to, and conversely, influence both the inherent structures of programming and the practices of the programmer. Learning to program has clear value in and of itself, but procedural literacy imbues the student with the ability to carefully read computational media. For example, one will recognize the patterns and behaviors of a given videogame as manifestations of general algorithms found in all sorts of software.

The open source programming language, Processing, was developed to address this need. It is powerful and simple. It provides the advantages of both a scripting and object-oriented language, so that the Processing programmer can focus on fundamentals while learning to create flexible building blocks. By nature of being open source, it is free, works on multiple operating systems, and is well-documented.

During the latter weeks of the semester we will begin again with a new language. With the fundamentals of coding fresh in hand, we will learn the language, C# (within the Unity environment) starting with its core principles.

Finally, through readings, screenings, and discussions we will examine the portrayal of computer programmer as protagonist and as villain. Starting with the earliest "hackers", we will consider the ethos and attitude of the programmer in fiction and culture.

Student evaluation will be based on assignments, projects, and readings.

Evaluation of student performance:

Participation (Discussions of Readings)	15
Weekly Assignments (5)	35
Code.org	10
Indiecade	5
Project 1	15
Project 2	20
Total:	100

Readings:

Learning Processing by Daniel Shiffman
Hackers by Steven Levy
Snow Crash by Neal Stephenson
Expressive Processing by Noah Wardrip-Fruin
Process Intensity and Social Experimentation by Ian Bogost

Course Content

Week 1: Introduction/Overview

definitions of procedurality
pixels, color, shapes
functions
Discussion: Media After Software

Week 2

logical flow
conditionals
- Read for today - Learning Processing Chapter 1, 2, 3, 4, 5
- Due for grade- Draw something that we ought to recognize. Like the Eiffel Tower or Garfield. Use `strokeWeight()`, `fill()`, and such functions.

Week 3

loops
discussion: Expressive Media
- Read for today - Learning Processing Chapter 6
- Read for today - Wardrip-Fruin (pp 1-21)
- Due - think of 1 example of a Wardrip-Fruin effect (based on [reading](#)).
- Due for grade Wednesday at 7pm - Integrate book [example 4-7](#) with a state machine [week2inclassA.pde](#)
- Due for grade - see explanation, below at 'Explanation of Code.org'

Week 4

- Read for today - Snow Crash (pp 1-111)
- Due for grade Wednesday at 7pm - Notice how loops augments [this](#), into [this](#). Make project using loops that is about drawing, or your interpretation of it.
- Test Yourself 1

Week 5

functions

objects

- Read for today: Learning Processing Chapter 7, 8
- Read for today: Levy Chapters 1, 2, 3
- Due for grade Wednesday at 7pm - Write two pages comparing anyone in the Levy book to anyone contemporary - an artist, technologist, or entrepreneur. This can be an essay or a short story.

Week 6

objects and algorithms

- Read for today - Learning Processing Chapter 9, 10 (end on page 177)
- Due for grade Wednesday at 7pm - Make [this](#) using [example 8_2](#).
- Read for today: [Bogost](#)
- Read for today: Levy Chapter 7

Week 7

- Test Yourself 2
- Read for today - Snow Crash (pp 111-227)

Week 8

arrays and objects

Week 9

Indiecade. Attend it.

Week 10

- Due for grade Wednesday at 7pm - Assignment given at the end of class on week 8. variables, functions, conditionals, components (in C#/Unity)

Week 11

interactivity, scope and access, loops (in C#/Unity)

- Read for today - Snow Crash (pp 227-354)
- Discussion: The Programmer Protagonist

Week 12

- Due for grade - Represent Time In an Unusual Way
- critique
- arrays and lists (in C#/Unity)

Week 13

Workshop

Week 14

Thanksgiving

Week 15

- Read for today - Snow Crash (pp 355-470)
- Due for grade - A Digital Toy critique

Explanation of Weekly Assignments:

About half of the semester's weeks, you will be given a straightforward and specific coding assignment. Five of them will be graded and each will be explained the week before it is due.

Explanation of Code.org:

Go to: <https://studio.code.org/>. Sign up as a student (button for 'Student Sign Up'). Be sure to put your real name as your 'Display Name' so you are identifiable.

Once logged in, scroll to near the bottom, you'll see 'Add a teacher'. Enter this section CODE: TEIXYN.

Now your progress will be tallied. On the same page, you'll find 'Accelerated Course'

(<https://studio.code.org/s/20-hour>)

You must complete all sections of all coding 'Stages' by Week 3. You can skip all the 'Unplugged' ones.

Explanation of Tests:

The two tests will follow the same format. Students will be given two and a half hours to complete a number of programming tasks that they will implement in the Processing environment. An example task could be, "Write a program that lays 10 spheres on the screen at random locations, except the spheres cannot be touching each other. Using a for loop is recommended." A second example could be, "Add to the project on page 67 of our book so that the lines gradually fade away rather than disappear immediately."

The concepts covered in each test will be based on class lectures from all previous weeks. Students may use any notes, books, or online documentation, and can ask the professor and SA some questions.

These will be graded on effort.

Explanation of Indiecade Assignment:

Go to Indiecade at least one of its 3 days. Take photos and/or video to prove that you went, and that you played the games. Post these images to a single web page of some sort and email to me.

Explanation of Project 1:

"Represent Time in an Unusual Way" asks the student to develop a non-interactive project that presents the current time in an expressive way. Examples might include a clock that represents the time by means other than hands or numbers, or a screensaver that changes based on the time of day.

Explanation of Project 2:

"A Digital Toy" is the most open-ended project assignment. This interactive project is like a digital game without a clear outcome. In other words, a focus is placed on giving the user one or more digital objects to manipulate but there is no winning or losing involved. Students are encouraged to use many of the programming concepts taught during the semester. The projects should be interactive, emergent, and playful.

Missing an Assignment Deadline, Incompletes:

The only acceptable excuses for missing an assignment deadline or taking an incomplete in the course are personal illness or a family emergency. Students must inform the professor before the assignment due date and present verifiable evidence in order for a make-up to be scheduled. Students who wish to take incompletes must also present documentation of the problem to the instructor or teaching assistant before final grades are due. Incompletes are only available after the 12th week withdrawal deadline.

Academic Conduct

Plagiarism – presenting someone else’s ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in *SCampus* in Section 11, *Behavior Violating University Standards*

<https://scampus.usc.edu/1100-behavior-violating-university-standards-and-appropriate-sanctions/>. Other forms of academic dishonesty are equally unacceptable. See additional information in *SCampus* and university policies on scientific misconduct, <http://policy.usc.edu/scientific-misconduct/>.

Discrimination, sexual assault, and harassment are not tolerated by the university. You are encouraged to report any incidents to the *Office of Equity and Diversity* <http://equity.usc.edu/> or to the *Department of Public Safety* <http://capsnet.usc.edu/departement/departement-public-safety/online-forms/contact-us>. This is important for the safety whole USC community. Another member of the university community – such as a friend, classmate, advisor, or faculty member – can help initiate the report, or can initiate the report on behalf of another person. *The Center for Women and Men* <http://www.usc.edu/student-affairs/cwm/> provides 24/7 confidential support, and the sexual assault resource center webpage sarc@usc.edu describes reporting options and other resources.

Support Systems

A number of USC’s schools provide support for students who need help with scholarly writing. Check with your advisor or program staff to find out more. Students whose primary language is not English should check with the American Language Institute <http://dornsife.usc.edu/ali>, which sponsors courses and workshops specifically for international graduate students. The Office of Disability Services and Programs http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html provides certification for students with disabilities and helps arrange the relevant accommodations. If an officially declared emergency makes travel to campus infeasible, USC Emergency Information <http://emergency.usc.edu/> will provide safety and other updates, including ways in which instruction will be continued by means of blackboard, teleconferencing, and other technology.

Disruptive Student Behavior:

Behavior that persistently or grossly interferes with classroom activities is considered disruptive behavior and may be subject to disciplinary action. Such behavior inhibits other students’ ability to learn and an instructor’s ability to teach. A student responsible for disruptive behavior may be required to leave class pending discussion and resolution of the problem and may be reported to the Office of Student Judicial Affairs for disciplinary action.

Syllabus Updates:

This syllabus is liable to change up to the beginning of class and possibly over the semester. Please check the posted syllabus regularly, and note all changes that are shared by the instructor in class.