

MART 385: Coding for Designers & Artists Syllabus

A new course in programming computers for media-rich interactive applications

When	11:10am - 12:25pm Tuesdays & Thursdays Fall Semester, 2017
CRN	95747
Units	3 Units, 42 hours lecture, 32 lab hours
Where	Building 22, Room 113
Who	James Khazar www.khazar.com
Office Hours	Tuesdays & Thursdays 12:30pm - 1pm. Location based on prior arrangement
Email	classes@khazar.com

Introduction

There is a clear need for an alternative approach in teaching the methods and practices of software programming for students whose focus is in the disciplines of visual arts and design. Today's job marketplace calls for individuals who can not only create visuals, such as 3D characters, and designs, like web sites, but also know the intricacies of programming those visuals to behave interactively through programming. While they may not be programming themselves, the chances are high that they will be working with programmers, and having a working knowledge of what can and can not be done with programming and the relative complexity of programming projects is an important job skill. A recent review of the first ten jobs listed for Web Designers on indeed.com showed seven required experience in programming, and they all involved working with a team of one or more programmers.

Visual Design and Art students are predisposed to learning by seeing. While computer programming courses offered through Computer Information Science and Engineering departments cover the relevant skills in writing software applications, they are not targeted for visual learners. As described in the Program Learning Outcomes for the Cañada Community College Computer Science department, they are educating students in how to "apply knowledge of math, science, and computer science to identify, formulate and solve computer science problems," and are specifically targeted with lower division courses for a Bachelor of Science degree.

Art and design students are usually not trying to earn a Science degree. Many find the prospect of taking science courses intimidating. As John Maeda, Design Partner at Kleiner Perkins Caufield & Byers, believes "...hybrids (workers) that can fluidly cross the chasm between technology and the arts are mutations in the academic system."¹ The programming languages covered in the Computer Science department are Assembly Language, C++, and Java, where the industry standard for programming visuals and interactivity on the Web are JavaScript, JQuery, HTML5, and ActionScript.

This course is in the Business, Design, & Workforce division and is designed to address the needs of visual designers and artists and work within their visually-oriented view of the world and their corresponding learning style.

The course is built around the foundational work developed by Casey Reas, professor in the Department of Design Media Arts at UCLA and a graduate of the MIT Media Laboratory, and Ben Fry, a designer, programmer, and author with a doctoral degree from MIT Media Laboratory. Together, they developed a specialized open-source programming language called Processing. It is a simple language that excels at programming drawings, animation, and interactivity. The emphasis is on quickly creating functional projects and using them to learn core programming concepts one step at a time, while providing encouragement and motivation with immediate visual feedback. According to Reas' and Fry's textbook on Processing:

"Processing makes it possible to introduce software concepts in the context of the arts and also to open arts concepts to a more technical audience. Because the Processing syntax is derived from widely used programming languages, it's a good base for continued learning. Skills learned with Processing enable people to learn other programming languages suitable for different contexts including web development, computer graphics, and electronics."²

Our version of the course also strongly references real-world applications of interactivity. We will incorporate JavaScript, JQuery, and Adobe Animate ActionScript, along with Processing, citing examples of interactive code on the Web and on Mobile devices.

Target Students

Students in the Certificate Programs, Associate of Arts Programs, and Transfer Programs for:

- Web Design
- Graphic Design
- Multimedia Art & Technology
- 3D Animation & Videogame Art
- Interdisciplinary Studies with Emphasis in Arts & Humanities

¹ Reas, Casey, and Ben Fry. Processing: A Programming Handbook for Visual Designers and Artists. Cambridge, MA: The MIT Press, 2014. Xiii. Print.

² Ibid. 4-5

Career Opportunities

Many of today's technology sector jobs call for a combined set of skills, which include both visual design and programming. These include, but are not limited to:

- Web Developer/Designer
- Interactive Developer/Designer
- Web Master
- Graphic Designer
- Video Game Artist
- User Experience Designer (UIX)
- App Designer

Strategic Learning Objectives

After taking this course, students should be able to:

- Write basic code that takes to user input and outputs changes in status
- Predict and manage the complexity of coding projects
- Work with coding teams to create valuable user experiences
- Have a working knowledge of the basics of programming, including:
 - Drawing objects
 - Creating and manipulating variables
 - Responding to inputs
 - Working with different media types, like images, fonts, and shapes
 - Animating objects
 - Creating functions
 - Working with classes and objects
 - Creating and manipulating with arrays
- Translating Processing code into other languages, including:
 - JavaScript and JQuery
 - Flash ActionScript
- Understand coding best practices, such as naming, commenting, and style
- Understand data types and their appropriate usage
- Understand sequence of operations in code
- Understand the scope of variables

Course Outline

Week	Topic
1	Introduction • What is Programming? • What is Interactivity? • Coding languages and their various flavors • The Processing Language Environment
2	Starting to Code • Writing a program to say “Hello World” • Creating a saving code • Posting code
3	Drawing with Code • Basic shapes • Drawing sequence • Shape properties (size, position, color) • Using comments in code
4	Variables • Creating and processing variables • Basic math with variables • Using a <i>for</i> loop
5	Interactivity I: Mouse input • Interacting with the mouse position • Interacting with mouse clicks
6	Interactivity II: Evaluating input • <i>if, then, else</i> statements • <i>while</i> statements
7	Working with different media • Images • Fonts • Shapes
8	Motion/Animation • Controlling speed and direction • Tweening • Using random values • Using timers • Translate, rotate, and scale

9	Coding functions I • Function basics • Creating functions • Returning values from functions
10	Coding functions II • Passing values into functions • Functions talking to functions
11	Objects I • Creating classes and objects
12	Objects II • Classes and objects continued • Dot syntax
13	Arrays • Creating arrays • Repetition and arrays • Arrays of objects
14	Other languages • HTML5 • JavaScript • JQuery • Translating Processing to JavaScript
15	Other Languages • ActionScript • Similarities between JavaScript and ActionScript
16	Mobile and other applications • Processing and the Android SDK • Arduino electronics
17	Review and Final

Administration

Grades

Letter Grade	Grade Value	Min Value to get for final grade
A+	100	Not available through WebSmart
A	95	92.5
A-	90	87,5
B+	85	82.5
B	80	77.5
B-	75	72.5
C+	70	67.5
C	65	62.5
C-	60	57.5
D+	55	Not available through WebSmart
D	50	25
F	0	<25
F		

Grading Components

- Projects & Homework: 99%
- Lab Time: 1% (Binary, you get a Pass or a Fail)

Rules

- Homework is Pass/Fail, but failing any given piece of homework will reduce your final grade by one step, so if you had been getting an A based on your assignments, you'd be knocked down to an A-.
- Homework has proposed due dates, but is not ultimately due until the end of the semester. It is suggested that you keep up with the due dates so you don't fall behind.
- A few minutes will be available at the beginning of class to upload your assignment, but all assignments are due at the beginning of class on the assignment's due date. Don't count on having time to work on the assignment on the day of the crit.
- Late assignments are lowered by one full grade
- Unless otherwise arranged, assignments will not be accepted after the beginning of the next class session after it's due

- Attendance at the final is required
- Plagiarized work will be immediately failed
- Creativity and neatness count!

Lab Hours

- The “32 by arrangement” lab hours required with this class must be done in the studio during open hours. However, you may also work outside the lab on your projects. Download the form for working outside the lab here and turn it in at the end of the semester. When using the lab, don't forget to sign both in and out – failure to sign out will create an attendance report that shows only 15 minutes of lab time.

Misc.

- There are locations on campus for email and surfing the web. Don't do these things during lectures/demonstrations/critiques.
- Please consult your class schedule for any drop class information and CR/NC grading policies as this is the responsibility of the student.
- Sometimes student works will be used by Cañada College for educational purposes and to promote the program or college or in galleries or shows or other forums. It is important to always include your name in your file so proper attribution can be given. Unnamed or improperly named files may be attributed to 'Anonymous'. To 'opt out', submit a request in writing at the any time during the semester.

Important Dates

- August 16th: Classes begin
- August 29th: Last day to register (ADD) and last day to drop and get refund
- September 4th: Last day to drop without a "W"
- November 15th: Last day to drop with a "W"
- November 23rd: Thanksgiving holiday. No class.
- December 10th-16th: Finals

Guidelines For A High Grade (B+ Or Higher)

- Participate in critiques and volunteer constructive criticism to other student's work
- Do your reading and research for your projects and explore more than one design solution
- Successfully incorporate skills and disciplines acquired from earlier projects into future projects
- Neatness counts! Turn in clean, well presented work (well-named layers, etc.)
- Be resourceful, be creative and when in doubt, ask

Materials

- Removable storage (USB Jump Drives: 512MB at least) for storing your images and backing them up. All projects must be kept throughout the semester for the final presentation. Lost files are not an acceptable excuse – back up your work!

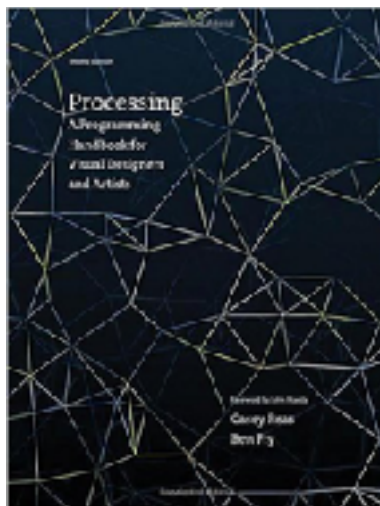
Required Text



Getting Started with Processing

Casey Reas, Ben Fry
Second Edition, 2015
Maker Media
ISBN-13: 978-1457187087
ISBN-10: 1457187086

Recommended Text



Processing: A Programming Handbook for Visual Designers and Artists

Casey Reas, Ben Fry
Second Edition, 2014
MIT Press
ISBN-10: 026202828X
ISBN-13: 978-0262028288

Projects

Each project has two components: A robot, which is created by copying code from the textbook, and an associated variation on the concepts the robot demonstrates.

Project 1: Drawing

Use Processing's drawing functions to create a robot. Includes the use of *fill()*, *stroke()*, *line()*, *ellipse()*, and *rect()*. This robot will be used in other projects. Then use the same functions to recreate a provided image.

Project 2: Variables

Modify the robot from Project 1 and the custom image to incorporate variables. Create multiple variations of the robot and custom image by manipulating the variable's values.

Project 3: Interactivity

Modify the robot from Project 1 and the custom image to interact with mouse clicks and mouse positions.

Project 4: Media

Modify the robot from Project 1 to incorporate images and fonts. Create a new project which is a "collage engine" that generates a new collage each second.

Project 5: Animation

Modify the robot from Project 1 to move. Incorporate motion into the collage project from project 4.

Project 6: Functions

Create four robot variations using a single function. Create 3 functions that can draw a letter of the alphabet.

Project 7: Objects

Create a robot object that controls the position and display of a robot. Convert the 3 functions that draw letters into object-oriented notation with a class for each letter.

Final Project: Interactive Application

Create an interactive application that incorporates the concepts in projects 1 through 7.

Texts

Required Text

Getting Started with Processing
Casey Reas, and Ben Fry.
2010. O'Reilly Media, Sebastopol, CA
ISBN: 978-1-449-3798-3

Suggested Text

Processing: A Programming Handbook for Visual Designers and Artists.
Casey Reas, and Ben Fry.
2014. The MIT Press, Cambridge, MA
ISBN: 978-0-262-02828-8

Other Useful Texts

Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction.
Daniel Shiffman.
2008 (Second edition soon to be released). Morgan Kaufmann Publishing, Burlington, MA
ISBN: 978-0123736024

Processing: Creative Coding and Computational Art 2.
Ira Greenberg.
2013. friendsofED, New York, NY.
ISBN: 978-1430244646

Processing for Visual Artists: How to Create Expressive Images and Interactive Art.
Andrew Glassier.
2010. A K Peters/CRC Press, Boca Raton, FL
ISBN: 978-1568817163

Foundation Actionscript 3.0 Animation: Making Things Move!
Keith Peters.
2008. friendsofED, New York, NY.
ISBN: 978-1590597910

Essential ActionScript 3.0.
Colin Moock.
2007. Adobe Developer Library, San Jose, CA.
ISBN-13: 978-0596526948

Online References

- processing.org
- openprocessing.org
- sketchpad.cc
- processingjs.org
- vimeo.com/tag:processing
- www.facebook.com/page.processing
- www.learningprocessing.com/tutorials/processing-in-eclipse/
- www.w3schools.com/html/html5_intro.asp
- www.w3schools.com/js
- codepen.io/pen
- jquery.com
- jqueryui.com

Academics

Units

3 units

Class Hours

48-54 lecture/32-36 by arrangement lab hours/semester

Recommended Previous Courses

Eligibility for READ 836 and ENGL 836; or ENGL 847 or ESL 400

Prerequisites

None

Course Catalog Description

MART 3NN PROGRAMMING FOR VISUAL DESIGNERS AND ARTISTS

Units (Grade Option) 3; Class Hours: 48-54 lecture/32-36 by arrangement lab hours/semester; Recommended: Eligibility for READ 836 and ENGL 836; or ENGL 847 or ESL 400; Prerequisite(s): None. **Description:** Introduction to writing software programs for non-programmers. Using the Processing development environment, JavaScript, and Adobe Flash ActionScript, students will learn the fundamentals of programming by creating examples of interactive animated graphics. Programming activities include drawing graphics, controlling animation, working with sound, and interacting with user inputs. Programming concepts covered include working with variables and arrays, writing functions, working with loops and logic, creating classes and objects, and translating between different coding languages.