

New Media / Image Arts
Ryerson University
Winter 2010

MPM16B

Introduction to Computation for Experience Design

Times and rooms:

Wednesday: 9-12am / Section 011 / RCC-359A

Wednesday: 1-4pm / Section 021 / RCC-359A

Instructor: David Bouchard

Email: david.bouchard@ryerson.ca

Office: VIC-823

Office hours: TBA

Class website: <http://www.ryerson.ca/~d2boucha/mpm16>

Description

This foundational studio class will introduce students to fundamental concepts, techniques and strategies for the production of computer mediated art works. Exposure to a broad range of practices will challenge students to move beyond the role of consumers of software into the position of reflective creators of experiences. Integrated with Studio II: Research for Experience Design this hands-on course will allow students to investigate the interplay between creative expression and technological capability. A variety of conceptual and practical approaches to virtual, hybrid and physical media practices will be explored. Emphasis will be placed on computer and coding literacy as well as constructing and evaluating innovative creative forms and media artifacts.

The class will be centred around learning the Processing development environment: *“Processing is an open source programming language and environment for people who want to program images, animation, and interactions. It is used by students, artists, designers, researchers, and hobbyists for learning, prototyping, and production. It is created to teach fundamentals of computer programming within a visual context and to serve as a software sketchbook and professional production tool”* [processing.org]

Communication

Your Ryerson email will be the main method of communication for this class. Class announcements will be made using the BlackBoard system. You will also be required to maintain a simple class webpage where you will post your responses to the assignments. You are expected to have the basic HTML skills required to publish this simple webpage and create links to your assignments.

Textbook

There is a **required** textbook for the class: *Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction*, by Daniel Shiffman. The book is available at the University bookstore, local bookstores (function13.ca) or online through either Amazon or Chapters. *The Processing Handbook* by Ben Fry and Casey Reas is another useful resource although it is **not** required.

Grading & Evaluation

64% - Problem sets (x4)
26% - Final project
10% - Participation

Problem sets will consist of a handful of questions addressing specific programming concepts, each to be answered by a short program, written individually. Programming questions will be graded according to:

- *Correctness*: does the program answer the question and produce the correct output? Are there compile, runtime or logical errors?
- *Style*: is the program clear and well commented? Does it make use of stylistic techniques covered in class? (naming schemes, organization)
- *Creativity*: is the response original and aesthetically appealing? Does it show personal style?

The responses to the problem sets should be available online on the student's class website, *before class* on the day the problem set is due. Late problem sets will be deducted 5% per day, up to a maximum of one week, after which you will get 0.

The **final project** will be an original short program, created either as an answer to a specific design brief (to be announced towards the end of the semester) or as an open project. You will need to submit a short proposal by the end of the semester -- additional guidelines will be handed out closer to the due date. Marking guidelines will be the same as for problem sets. You are encouraged to work in groups of up to three students for this project.

Participation is expected and required. You can demonstrate participation by being on time, voicing your opinions in class and helping others. Failure to sign the attendance sheet will constitute an absence; 3 absences will be an automatic 0 for participation.

University policies & practices

Academic Integrity

Students are expected to follow the Student Code of Academic Conduct which can be found in the calendar or on-line at the Academic Council website: <http://www.ryerson.ca/calendar/2009-2010/pg2030.html>

With respect to writing programs, borrowing source code from various on-line resources is an accepted and wide-spread practice (assuming that the licence allows it). However, make sure that full references are provided (URLs, original author) in your program's documentation for any borrowed code snippet.

Accommodation of Students with Disabilities

If you have a disability that might require accommodation, please make sure you register with the Access Centre and supply any necessary Accommodation Forms. Accommodations can not be applied retroactively. <<http://www.ryerson.ca/accesscentre/>>

Accommodation of Student Religious Observance

At the start of each term, students who have religious observance obligations which will lead to absences from campus or academic activities during the semester should fill out the appropriate forms and have me sign them.

For more information see the Student Guide <<http://www.ryerson.ca/studentguide/AcademicMatters2.html>>

Discrimination Harassment Prevention Policy

Ryerson University is committed to fostering a collegial study and work milieu that is free of discrimination and harassment and one in which all individuals are treated with respect and dignity. <<http://www.ryerson.ca/equity/dhpspolicy/>>

Class Schedule

Dates	Topics	Assignments (due)
Week 1 Jan 8, 9	Introduction Student and instructor introductions The Processing environment	
Week 2 Jan 15, 16	Basics Pixels, Processing and Interaction	<i>Reading I:</i> Processing preface (8 p)
Week 3 Jan 22, 23	Building blocks Variables, Loops and Conditionals	
Week 4 Jan 29, 30	Program organization Functions, Objects	PS 1: Basics
Week 5 Feb 5, 6	Arrays	
Week 6 Feb 12, 13	Putting it together Algorithms, Debugging	<i>Reading II:</i> Golan Levin, Essay for Creative Code, edited by John Maeda
Week 7 Feb 19, 20	<i>Reading week</i>	
Week 8 Feb 26, 27	Mathematics Probability, recursion, noise, trigonometry	PS 2: Structures
Week 9 Mar 2, 3	Advanced drawing Translation, rotation, scaling, 3D drawing Drawing with shapes (Vector)	
Week 10 Mar 12, 13	Typography Creating fonts, drawing text I/O Streams	PS 3: Drawing I
Week 11 Mar 19, 20	Images Loading, displaying, basic processing	
Week 12 Mar 26, 27	External Libraries, Java The outside world	
Week 13 Apr 2, 3	TBA	PS 4: Typography <i>Final project proposal due</i>
Week 14 Apr 9, 10	Final project workshop	
Exam Period	Final project presentations	