Math Art Code . DATT 2040

Course Director: Dan Tapper <dantap@yorku.ca>

Teaching Assistant: TBC

Course Github repository:
https://github.com/atarilover123/DATT-2040-Math-Art-Code

Course calendar description

Math Art Code explores relationships at the intersection of math, art and creative coding by providing a survey of mathematical concepts and techniques that support creation processes in contemporary code-based art and performance practices. Through a series of weekly studies reviewing mathematical concepts, their programmatic applications and investigations into the artists who employ them, we will build our own creative coding toolset. This toolset can be applied to the creation of computational artworks, such as video games, mixed reality, digital fabrication, and image manipulation and will provide a pathway from simple computational sketches to more advanced projects.

This course will be delivered through a combination of:

Weekly lectures – exploring mathematical concepts and how artists use math to create a wide range of computational and multimedia artworks.

Weekly labs – Putting mathematical concepts into practice, creating our own generative art works and computational media projects.

Required tools

The practical element of this course will be taught using a modular suite of tools which have been selected to showcase the versatility of creative coding practices.

We will be using Processing/p5.js as our primary development tool and exploring TouchDesigner and MaxMSP to expand knowledge of platforms and engage with audio processing, shaders and 3d rendering environments.

Processing – https://processing.org/

TouchDesigner – https://derivative.ca/

MaxMSP – https://cycling74.com/

Submissions and class resources

Students will create a Github account to upload weekly sketches and notes. Tests and quizzes will be conducted via Moodle.

Github – https://github.com/

Moodle – https://moodle.yorku.ca/

Topics and concepts

Topics include modular arithmetic, number systems, boolean algebra, combinatorics, vectors, trigonometry, coordinate systems, linear algebra and their application in the creation of generative artworks, the use of math concepts in the creation of computational artworks, such as video games, mixed reality, digital fabrication, and image manipulation.

Learning outcomes with examples

Students will be able to:

- Think critically about digital artworks and have an understanding of the underlying algorithms used to create them.
- Have an understanding of mathematical concepts and be able to apply these concepts to the creative process.
- Think modularly about problems and tools, utilizing knowledge of fundamental math and code concepts to work in multiple environments.

- Create their own computational artworks using math and code as a creative medium.

Graded assessment

- Sketches: 30%
- Quizzes: 20%
- Midterm exam: 15%
- Final exam: 15%
- Final creative project & evaluative essay: 20%

Sketches are regular creative assignments that apply the mathematical and artistic concepts from class to the creation of a computational arts-based study.

Quizzes will focus on mathematical and coding problems and the theoretical concepts associated with them.

Midterm exam will be a comprehensive exam that includes topics from the first half of the course.

Final exam will be a comprehensive exam that includes topics from throughout the course and will include both a mathematical portion with problem solving, and a reflection component on the concepts and art covered in the course.

Final creative project and evaluative essay is a project where students develop a computational artwork combining 3 or more mathematical/coding concepts from the course. This project will be combined with a short evaluative essay explaining the creation process, concepts used and why they were chosen.

Module breakdown

Weeks 1 - 4, Generative art

Weeks 5 - 6, Natural algorithms and feedback

Weeks 7 - 9, Multimedia processing

Weeks 9 -12, 3D world building and final projects

Additional information

Academic policies/information

Last date to drop a fall term (F) course without receiving a grade: November 11, 2022

Last date to drop a winter term (W) course without receiving a grade: March 17, 2023

Last date to drop a full year (Y) course without receiving a grade: February 10, 2023

Senate Academic Standards, Curriculum and Pedagogy Committee (ASCP) provides information that includes:

Research Ethics: https://www.yorku.ca/research/research-ethics/

Academic Honesty: https://www.yorku.ca/secretariat/policies/policies/academic-honesty-senate-policy-on/

Student Conduct: https://oscr.students.yorku.ca/student-conduct

Accessibility: https://accessibility.students.yorku.ca/

Academic Integrity: https://www.yorku.ca/unit/vpacad/academic-integrity/

Religious Observance:
https://rights.info.yorku.ca/accommodating-creed-religion-a-guide-for-students-faculty-and-staff/

Academic Accommodation:
https://accessibility.students.yorku.ca/academic-support- accomodations

Grading Scheme:
https://calendars.students.yorku.ca/2022-2023/grades-and-grading- schemes

• Important University Sessional Dates (you will find classes and exams start/end dates, reading/co-curricular week, add/drop deadlines, holidays, University closings and more.
https://registrar.yorku.ca/enrol/dates/2022-2023/fall-winter

Manage my Academic record
https://myacademicrecord.students.yorku.ca/degree-progress-report

• "20% Rule"
No examinations or tests collectively worth more than 20% of the final grade in a course will be given during the final 14 calendar days of classes in a term. The exceptions to the rule are classes which regularly meet Friday evenings or on Saturday and/or Sunday at any time, and courses offered in the compressed summer terms.

Final course grades may be adjusted to conform to Program or Faculty grades distribution profiles.

Bibliography

Groß, Benedikt. Laug, Julia. Bohnacker, Hartmut. Generative Gestaltung, 2012.

Groß, Benedikt. http://www.generative-gestaltung.de/2/, 2018, accessed August 16, 2022

Tarbell, Jared. http://www.complexification.net/gallery/, accessed August 16, 2022.

Shiffman, Daniel. The nature of code. D. Shiffman, 2012.

Pearson, Matt. Generative Art: A Practical Guide Using Processing, 2011.

Thompson, Jeffrey. Collision Detection. https://www.jeffreythompson.org/collision-detection/, accessed August 20, 2022.

Montfort, Nick. Reas, Casey. Baudoin. Patsy... 10 PRINT CHR$(205.5+RND(1)); : GOTO 10 <https://reas.com/10_print/>, 2012.

Fry, Ben. Reas, Casey. Processing: A Programming Handbook for Visual Designers and Artists. 2007.

Spalter Digital Collection. https://spalterdigital.com/medium/generative/, accessed August 20, 2022.

Shiffman, Daniel. The Coding Train. https://thecodingtrain.com/, accessed August 16, 2022.

Lewitt, Sol. Mass Moca https://massmoca.org/sol-lewitt/, accessed August 20, 2022.

Algorists, http://www.verostko.com/algorist.html, accessed Feb. 12, 2021

Ball, Philip. Patterns in nature: why the natural world looks the way it does. University of Chicago Press, 2016.

Escher, Maurits Cornelis, https://mcescher.com/, accessed Feb. 12, 2021.

Flake, Gary William. The computational beauty of nature: Computer explorations of fractals, chaos, complex systems, and adaptation. MIT press, 1998.

Lomas, Andy. https://andylomas.com/, accessed August 16.

Galanter, Philip, http://philipgalanter.com/, accessed Feb. 12, 2021.

Nervous System, https://n-e-r-v-o-u-s.com/, accessed August. 12, 2021.

Hofstadter, Douglas R. Gödel, Escher, Bach: an eternal golden braid. Vol. 13. New York: Basic books, 1979.

Maeda, John. Design by Numbers. 1999.

Loy, Gareth. Musimathics: the mathematical foundations of music. Vol. 1 & 2. MIT press, 2011.

Molnar, Vera, http://www.veramolnar.com/, accessed Feb. 12, 2021.

Nake, Frieder, https://en.wikipedia.org/wiki/Frieder_Nake, accessed Feb. 12, 2021.

Osinga, Hinke, https://www.math.auckland.ac.nz/~hinke/, accessed Feb. 12, 2021.

Kop, Paloma. From Material Space (Time). <https://palomakop.tv/MFA-Thesis-Book>, 2020

Reas, Casey, https://reas.com/, accessed Feb. 12, 2021.