Creative Coding
Module handbook

Computing

Key Information

Credits

Successful completion of Experience Design awards 20 credits.

Summative Assessment deliverables

S1. Creative coder case study (1,500 words) 40%
S2. Creative Coding portfolio, with accompanying 60% (Staged Deadlines Through Module)

1. Module Overview

Creative Coding is a practice that uses code for artistic or creative expression. This can cover a wide variety of media including generative art, animation, data visualisation, games, apps, robotics, music, interactive art, immersive experiences – and many more. Creative Coders are frequently engaged in online community building, the open-source software movement, and campaigns for an open and equitable internet. Communities of creative coders have developed free tools that are used globally by makers including Processing, P5.js, Pure Data and openFrameworks. Throughout your Creative Computing degree, you will engage with creative coding tools and communities making your own contributions through your research and developing creative practice. In this module you explore the context of your own emerging design practice through diving into the field of creative coding. You will research the designers of creative coding tools, the artists and makers using them, and the communities linking creative coders globally. You will gain hands on experience of being a creative coder through learning new languages and techniques from online creative coder communities and your creative coding tutors. Through making a research-informed creative coding piece or experiments collection we hope you will begin to feel at home within creative computing communities and gain a wider understanding of the context in which Creative Computing sits.

Although positioned here as a creative and expressive medium, increasingly creative coding and the thinking that accompanies it drives new forms of engaging digital experiences and is a highly-in-demand skill for creative technologists.

2. Key Advice

Much of the time as a software designer or developer, you are solving defined problems with clear goals. Here you need to think more as an artist or explorer, using code as a medium to realise undefined outcomes. Technical skills still matter and you will develop your code skills from the foundations learned in CodeLab, but you will also use code in unexpected ways, driven by curiosity and iteration rather than finding a particular solution.

You will explore the ideas, works and motivations of key digital artists and their ideas. We will use some of their work as inspiration for learning techniques. You will learn about software movements not driven by commercial interests but by some of the original aims of an open internet.

3. Schedule (tentative)

Creative Coding		
1	11-February-2025	Introduction to Module, p5.js and Basic Drawing
2	18-February-2025	Forms, Shapes, and Variables
3	25-February-2025	Working with Images
4	4-March-2025	Repetition, Decisions, and Randomization
5	11-March-2025	Mouse Interaction
6	18-March-2025	Interactive Art
7	25-March-2025	Assessment 1 starts
8	1-April-2025	Holidays
9	8-April-2025	Audio Visualization
10	15-April-2025	Data Visualization
11	22-April-2025	HTML HTML/ CSS/p5.js Game Part 1
12	29-April-2025	HTML HTML/ CSS/p5.js Game Part 2
13	6-May-2025	3d Geometry and renders
14	13-May-2025	Gaming
15	20-May-2025	**Assessment 2 starts**
16	27-May-2025	**Assessment 2 Deadline**
17	3-June-2025	Assessment 2 Review
18	10-June-2025	Assessment 2 Review
19	17-June-2025	**Summer Vacation Starts**

4. Assessment

Your learning will be measured via a mix of formative and summative assessments that occur across the course of the module. Summative assessments contribute to your module grades whereas formative assessments help you develop your skills. All assessment types are non-combative, and should be seen as an opportunity to identify areas in which you need more practice.

There are two formative assessments for Experience Design, and one summative assessment that contribute to your module mark.

Formative Assessment

Formative Assessment	Deadline
F1. Demonstrate reading and participation in a discussions and group exercises	Weeks 1 - 8
F2. Portfolio Review	Week 6 - 10

Summative Assessment

Summative Assessment	Module Contribution	Deadline (draft, and may change)
S1. Creative coder case study (1,500 words)	40%	TBC
S2. Creative Coding portfolio, with accompanying annotations	60%	TBC

** Please note this is the final deadline for the Creative Coding portfolio, however there are a series of submission points throughout the module. Please refer to the brief for more information*

Creative Coding Portfolio

Your portfolio will be a series of creative code sketches, with links to code, working examples and screenshots, accompanied by notes and reflections. These should show the evolution of sketches from class-taught or self-discovered techniques and styles to your own experimental outcomes.

Submission

Please refer to guidance on assessment briefs.

Deadlines for assessment must be respected. Late work is capped as per university procedure at 40% unless an extension is approved.

Please see assessment briefs for full details of your assignment deliverables.

5. Intended Learning Outcomes (ILOs)

Experience Design has a number of learning outcomes that are measured by both formative and summative assessment. Do familiarise yourself with the outcomes detailed below.

Learning Outcome	Assessment
By successful completion of the module, you will be able to demonstrate:	
1 Demonstrate knowledge of key concepts, practitioners, and communities in the field of creative coding	S1, F1
2 Demonstrate knowledge of contemporary debates in creative computing	S1, F1
3 Demonstrate critical understanding of key artists and communities within creative computing and ability to critically evaluate the quality of online sources	S1, F1
4 Apply knowledge of underlying coding concepts to different environments.	S2, F2
5 Use established coding techniques and critically evaluate the appropriateness of different approaches.	S2, F2

6. Learning and Teaching Methods

Learning will take two simultaneous paths, each with a variety of methods

1. Discovery of key movements and artists in the digital arts
    a. Student reading and accessing other resources
    b. Lecture content (recorded and in-class)
    c. Student discussion of key issues in class
2. Learning and creatively experimenting with specific code techniques and outcomes
    a. Students accessing books and online resources on code techniques
    b. In-class demonstration of techniques and variations
    c. In-class student experiments with techniques
    d. In-class and post session challenges for students
    e. Student reflections on creative code techniques and possibilities

Assessments and Feedback

Your learning is supported by formative and summative assessments (described in section 4). Feedback on your work is provided in a number of ways:

- Written feedback on summative assessments via Turnitin
- Verbal feedback from tutors in class
- Peer feedback on formative assessment in class

Please ensure that you are aware of your assessment deadlines and that you know how to find your grades and feedback via Turnitin.

Individual Study

As a university student, it is expected that you fortify and extend your knowledge outside of class hours, as well as attending regularly. It is unlikely that you could perform well in this module without attending, participating and undertaking additional study. Tutors provide guidance on where you personally should be concentrating your efforts.

7. Learning Resources

Ultra VLE

Module resources, assessment briefs and module announcements will be accessible via Ultra (Minerva). It is crucial that you become familiar with navigating Ultra and visit it regularly.

Core Reading (Full Resource list in the Talis list for this module)

- Zhang, Y.F. (2021) Coding Art : The Four Steps to Creative Programming with the Processing Language. 1st ed. 2021. Berkeley, CA: Apress L. P [Note: 2023 edition due anytime soon]

- Gross, B. et al. (2018) Generative design: visualize, program, and create with JavaScript in p5.js. Revised and Updated edition. New York: Princeton Architectural Press.
- The nature of code. [United States]: [Daniel Shiffman].
- McCarthy, L., Reas, C. and Fry, B. (2016) Getting started with p5.js: making interactive graphics in JavaScript and processing. Sebastopol: Maker Media.
- Reas, C.F. (2014) Processing: A Programming Handbookfor Visual Designers and Artists (2nd Edition). 2nd edn. Cambridge, USA: MIT Press.
- Levin, G. and Brain, T. (2021) Code as creative medium: a handbook for computational art and design. Cambridge: The MIT Press.

Online Resources and Help

- https://thecodingtrain.com/
- https://p5js.org/
- https://processing.org/
- https://openframeworks.cc/

Required Technology

- A GitHub account and a public repository for this module
- Microsoft VS Code editor
- A number of online or installed development environments, libraries and API's which will be introduced through the module.

Everyday Equipment

You should ensure that you have the following items available at each teaching session:

- Your student ID card
- Login details for your university account
- Pens and paper for notes and idea development
- Online note taking app or document for keeping useful links, names of artists and resources

You are encouraged to bring laptop devices into sessions to work on. Do however ensure that you have the relevant software for the module installed in advance (ask your tutor for details) and a working power supply.

8. Rules and Requirements

All essays and reports must conform to university styling and submission guidelines. They must:

- Be word-processed using a conventional font and size (e.g. Times New Roman, 11 or 12) and 1.5 or double line spaced on single-sided paper
- Contain appropriate in-text citation that supplies an accurate list of references
- Employ the Harvard system of referencing. Please refer to http://www.citethemrightonline.com/ as a useful resource for referencing
- Be precise in spelling and paragraphing

Marking criteria

Marking criteria are the guidelines by which we grade your work. You should try to familiarize yourself with these guidelines as they give a clear indication of what we are looking for when marking.

Marking criteria for each of the assessments listed above is provided within their respective assignment briefs. These can be found on Aula.

Academic Misconduct

It is essential that you do not draw upon or duplicate previously submitted materials or employ materials derived in whole or in part from undisclosed sources, such as the web or generative AI applications. Proven cases of plagiarism (passing off material produced by others as your own work or the whole or part use of material not sourced) could lead to disqualification.

If in doubt, ask.

Unacceptable academic practice, particularly in assessment, is known as *Academic Misconduct*. Academic misconduct may take a variety of forms, which cannot all be covered in detail here, but the most common are cheating in formal examinations and the plagiarism of coursework. Others include collusion with other students for the production of written work, impersonation in examinations, or submission of fraudulent

mitigating circumstances evidence. The penalties for academic misconduct are severe and if students are in any doubt about what constitutes acceptable academic practice they must consult their tutors for advice.