

LINGUAGENS DE PROGRAMAÇÃO APLICADAS AO DESIGN

PROGRAMMING LANGUAGES APPLIED TO DESIGN

Área científica Scientific domain	Design de Comunicação Communication Design
Anos letivos Academic years	2021–2023
Grau de ensino Degree	Mestrado Master of Arts
Duração Duration	Semestral Semestral
Horas de contacto Student workload	(TP 21 horas + OT 14 horas) (TP 21 hours + OT 14 hours)
ECTS	(6 ECTS)

PROGRAMA CURRICULAR

COURSE UNIT CONTENT

1 — Objetivos de Aprendizagem

Intended learning outcomes

A disciplina de Linguagens de Programação aplicadas ao Design visa dotar os alunos dos seguintes conhecimentos e aptidões:

- a) contextualizar historicamente o aparecimento e uso do código como matéria prima para a prática criativa.
- b) reconhecer os paradigmas dominantes e as características operativas das principais linguagens e ambientes de programação.
- c) conhecer e pôr em prática as principais técnicas e algoritmos aplicáveis ao desenho gerativo, animação procedural, manipulação audiovisual e desenvolvimento de sistemas interativos.
- d) adquirir autonomia na implementação de sistemas computacionais originais.
- e) desenvolver uma perspectiva crítica informada sobre o papel da programação na prática criativa.

The Programming Languages applied to Design class aims to teach the students the following skills:

- a) to historically contextualize the appearance of code as prime matter for creative practice.
- b) to recognize the dominant paradigms and operative characteristics of the main programming languages and programming environments.
- c) to know and put into practice the main techniques and algorithms that can be applied to generative drawing, procedural animation, audiovisual manipulation, and interactive systems development.
- d) to acquire autonomy in the implementation of novel computational systems.
- e) to develop an informed critical perspective on the role of programming in creative practice.

2 — Conteúdos Programáticos

Syllabus

Introdução à Programação Criativa.

- Evolução do uso do código como material criativo
- Tecnologias emergentes.

II. Fundamentos da Programação Criativa.

- Paradigmas e características das linguagens de programação
- Estrutura de um programa
- Variáveis e expressões
- Controlo de fluxo
- Funções e objetos
- Bibliotecas

III. Desenho Generativo.

- Sistemas de coordenadas e referenciais
- Desenho de primitivas gráficas
- Transformação geométrica e composição

- Cor e estilos visuais
- Aleatoriedade e gramáticas generativas

IV. Animação Procedimental.

- Tempo, movimento e interpolação
- Simulação física
- Sistemas de partículas
- Agentes inteligentes

V. Manipulação Audiovisual.

- Formatos de dados
- Aquisição de sinal (áudio e vídeo)
- Manipulação temporal
- Efeitos e filtros

VI. Sistemas Interativos.

- Estrutura de um sistema
- Input do utilizador
- Desenho de interfaces
- Protocolos de comunicação

I. Introduction to Creative Programming

- Evolution of the use of code as creative material.
- Emergent Technologies.

II. Fundamentals of Creative Programming

- Paradigms and characteristics of programming languages.
- Program structure
- Variables and Expressions
- Flow control.
- Functions and Objects.
- Libraries.

III. Generative Drawing

- Coordinate systems and referential
- Graphics primitives
- Geometric transforms and composition.
- Colour and visual styles.
- Randomness and generative grammars.

IV. Procedural Animation

- Time, movement and interpolation.
- Physics simulation.
- Particle systems.
- Smart agents.

V. Audiovisual Manipulation

- Data formats.
- Signal acquisition (audio and video)
- Time-domain manipulation.
- Effects and Filters

VI. Interactive Systems

- Systems structure.

- User input.
- Interface drawing.
- Communication protocols.

3 — Metodologias de Ensino e Avaliação

Teaching and Evaluation Methodologies

As aulas, de cariz teórico-prático são compostas por momentos de exposição teórica dos tópicos a analisar, seguidas da discussão em grupo dos temas acompanhada de pequenos exemplos ilustrativos. No final de cada módulo os alunos fazem um pequeno exercício teórico (pesquisa) ou prático (aplicado). A avaliação contínua (15%) incide no acompanhamento dos exercícios, aferindo a evolução dos alunos, assiduidade, participação e empenho. As avaliações qualitativas periódicas de cada exercício (65%) permitem aferir criatividade, metodologia, adequação do trabalho e cumprimento de prazos, revertendo para a avaliação quantitativa final (20%). A classificação final contempla a totalidade do trabalho desenvolvido, incluindo avaliação contínua 10% + exercícios teóricos 30 % + exercícios práticos 60%.

The classes combine theory and practice, comprising moments of theoretical exposition of the topics to be analyzed followed by group discussions of the subject matter accompanied by small illustrative examples. At the end of each module students are required to deliver a small theoretical (research) or practical (applied) exercise. Continuous assessment (15%) is based on the monitoring of exercises, attesting the student's evolution, attendance, participation and commitment. Periodic qualitative evaluations of each exercise (65%) enable the assessment of creativity, methodology, adequacy of the work and conformity to deadlines, reverting to the final quantitative assessment (20%). The final grade contemplates the totality of the work developed, including continuous assessment 10% + theoretical exercises 30% + practical exercises 60%.

4 — Bibliografia Recomendada

Recommended Bibliography

- Cramer, F. (2005). *Words Made Flesh: Code, Culture, Imagination*. Piet Zwart Institute.
- Fuller, M. (2008). *Software Studies – A Lexicon*. Cambridge, Mass.: MIT Press.
- Klanten, R., Ehmann, S. & Hanschke, V. (2011). *A Touch of Code. Interactive Installations and Experiences*. Berlin: Die Gestalten Verlag.
- Parent, R. (2012). *Computer Animation: Algorithms and Techniques* (3rd Edition). San Francisco, Ca.: Morgan Kaufmann.
- Reas, C & Fry, B. (2007). *Processing : a programming handbook for visual designers and artists*. Cambridge, Mass.: MIT Press.
- Reas, C., McWilliams, C., & LUST. (2010). *Form+Code in Design, Art, and Architecture*. New York: Princeton Architectural Press.
- Shiffman, D. (2012). *The Nature of Code: Simulating Natural Systems with Processing*. Amazon CreateSpace.