# Course guide
# 804260 - PGA - Advanced Visual Programming

**Last modified:** 18/05/2026

**Unit in charge:** Image Processing and Multimedia Technology Centre
**Teaching unit:** 804 - CITM - Image Processing and Multimedia Technology Centre.

**Degree:** BACHELOR'S DEGREE IN VIDEO GAME DESIGN AND DEVELOPMENT (Syllabus 2014). (Optional subject).

**Academic year:** 2026 **ECTS Credits:** 6.0 **Languages:** Catalan, English

## LECTURER

**Coordinating lecturer:** Ríos Jerez, Alejandro

**Others:** Ríos Jerez, Alejandro
Solé Carreras, Ferran

## PRIOR SKILLS

Knowledge of object-oriented programming in C++, knowledge of OpenGL, knowledge of mathematics applied to video games, specifically in 3-dimensional geometry

## TEACHING METHODOLOGY

Lectures are divided in 2h sessions. During part of the sessions, the teacher exposes the theoretical concepts and proposes to the students their implementation. In this way, students practice the concepts introduced in addition to dedicating this time to the resolution of doubts and problems that have been encountered during their realization. The teacher will keep track of the progress that will serve to evaluate the attitude of the students.

## LEARNING OBJECTIVES OF THE SUBJECT

- To describe the advanced theoretical concepts of computer graphics.

- To apply advanced theoretical concepts of computer graphics to video game development projects.

- To obtain the maximum performance of GPUs for the development of complex graphics algorithms in real time.

## STUDY LOAD

| Type | Hours | Percentage |
|---|---|---|
| Hours large group | 18,0 | 12.00 |
| Self study | 90,0 | 60.00 |
| Hours medium group | 30,0 | 20.00 |
| Guided activities | 12,0 | 8.00 |

**Total learning time:** 150 h

## CONTENTS

### Shader programming with Qt and GLSL

**Description:**

1. Introduction to Qt
2. Introduction to GLSL
3. GLSL Basic Tools
4. Introduction to basic shader programming

**Related activities:**
Delivery 1: Implementation of a procedural raycasting shader

**Full-or-part-time:** 50h
Theory classes: 10h
Practical classes: 10h
Self study : 30h

### GPU programming with OpenGL

**Description:**
1. Global view of the render pipeline
2. Normalized device coordinates (NDC)
3. Rendering of triangle meshes
4. Passing "global" variables to shaders
5. State handling (visibility tests, blending, etc)
6. Debug and optimization tools
7. Geometric transformations
8. Render to texture

**Full-or-part-time:** 50h
Practical classes: 10h
Guided activities: 10h
Self study : 30h

### Advanced computer graphics techniques

**Description:**
1. Bump, normal, and relief mapping
2. Full-screen effects
3. Screen Space Ambient Occlusion (SSAO)
4. Environment mapping
5. Physically based rendering
6. Water
7. Bloom

**Related activities:**
Final project: Implementation of a graphics engine

**Full-or-part-time:** 50h
Theory classes: 10h
Practical classes: 10h
Self study : 30h

## GRADING SYSTEM

Any academic fraud, plagiarism or use or possession of unauthorised means in an assessment activity leads to a mark of zero (0) for the affected test or assignment. Furthermore, in accordance with the University's regulations, if disciplinary proceedings are initiated the subject is recorded as "Pending Assessment" until a decision is reached. These incidents are managed in accordance with the **Action Framework for Academic Integrity in Assessment at the UPC**.

Three practical exercises (PE) related to specific contents of the subject will be developed and evaluated during the course , in addition to a final project (PR) that will consist of a viewer that allows to visualize the techniques explained in class. The participation (PART) of the student will also be evaluated based on their interventions in class, their daily work and attendance. The practical exercises will be carried out in class, without internet connection and the project will be carried out throughout the course incrementally. As it is a practical subject, there will be no re-evaluation exam.

Final grade 0.20 * EP1 + 0.20 * EP2 + 0.20 * EP3 + 0.30 * PR + 0.1 * PART

Irregular actions that may lead to a significant variation of the grade of one or more students constitute a fraudulent performance of an evaluation act. This action entails the descriptive grade of failure and a numerical grade of 0 for the ordinary global evaluation of the course, without the right to re-evaluation.

## EXAMINATION RULES.

All the activities will be developed at class with the assistance of the lecturer and will be individual. All of them must be delivered through the Virtual Campus following the instructions provided in the respective statements (names of the correct files, etc.). The projects submitted after the agreed date and time will be considered as non-presented. Any circumstance that does not allow a project to be submitted in time must have a justified cause and will be communicated to the teacher well in advance.

The proposed projects and activities must be able to be executed in the CITM classrooms, therefore, it is the student's responsibility to make sure that they work at home with the same version of the software used in the center and that their project can be executed without errors in the CITM.

## BIBLIOGRAPHY

**Basic:**
- Kessenich, J.M.; Sellers, G.; Shreiner, D. OpenGL programming guide: the official guide to learning OpenGL, version 4.5 with SPIR-V. 9th ed. Upper Saddle River, NJ: Addison-Wesley, cop. 2017. ISBN 9780134495491.
- Rost, Randi J; Licea-Kane, Bill; Ginsburg, Dan. OpenGL shading language. 3rd ed. Upper Saddle River, NJ: Addison Wesley, 2010. ISBN 9780321637635.
- Möller, Tomas; Haines, Eric; Hoffman, Naty. Real-time rendering [on line]. 4th ed. Boca Ratón: Chapman and Hall/CRC, 2018 [Consultation: 14/07/2025]. Available on: https://doi-org.recursos.biblioteca.upc.edu/10.1201/b22086. ISBN 9781351816144.

**Complementary:**
- Nguyen, Hubert. GPU gems 3. Upper Saddle River, NJ: Addison-Wesley, 2007. ISBN 9780321515261.

## RESOURCES

**Other resources:**
ShaderToy: https://www.shadertoy.com />GLSL Sandbox: http://glslsandbox.com />OpenGL: https://www.opengl.org />OpenGL loader: https://glad.dav1d.de />GLM library: https://github.com/g-truc/glm />GLFW library: https://www.glfw.org />ImGui library: https://github.com/ocornut/imgui />STB library: https://github.com/nothings/stb />Assimp library: https://www.assimp.org