



# Course guide

## 804230 - P1VJ - Project I

**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). (Compulsory subject).

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

### LECTURER

**Coordinating lecturer:** Alejandro París  
**Others:** Alejandro París  
Rodrigo de Pedro

### PRIOR SKILLS

Knowledge of programming using C.

### TEACHING METHODOLOGY

Project-based learning: students design, develop and present a video game based on an initial challenge.

During each class, the lecturer will first show the students the theory behind the problem that need solving. Together with the students, the lecturer will explore the different solutions that exist in the present that solve and simplify the complexities of real time applications like videogames.

The lecturer will provide source code for the student to study and complete while integrating it in their own source code for future reference and use. Closing each session, the lecturer will provide with ideas for improving the systems challenging student in order to help and orientate the students in the self learning time.

### LEARNING OBJECTIVES OF THE SUBJECT

- Recognise the concepts and procedures involved in managing video game creation projects.
- Plan projects of casual video games, 2D video games and / or 3D video games, using project management tools as support.
- To make decisions about complex situations based on critical reflection, considering the ethical implications of actions.
- Collaborate effectively and responsibly as a member or leader of a team, in interdisciplinary contexts or not, considering the available resources.

### STUDY LOAD

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



Total learning time: 150 h

## CONTENTS

### 1. Development tools

**Description:**

Distributed work with Git  
Services of github.com  
Tools for communication and teamwork: Trello and Slack  
Development tools: Visual Studio

**Full-or-part-time:** 20h

Theory classes: 8h  
Self study : 12h

### 2. Introduction to raylib programming

**Description:**

Game structure with raylib  
Sprites and transparencies  
Using input devices  
Using audio systems

**Full-or-part-time:** 35h

Theory classes: 14h  
Self study : 21h

### 3. Coding arcade games

**Description:**

Modular code structure  
Rendering and texture management  
The input subsystem  
The audio subsystem  
Sprite animations  
Collision management  
User Interfaces

**Full-or-part-time:** 60h

Theory classes: 24h  
Self study : 36h

### 4. Logic and FSM

**Description:**

Artificial Intelligence  
Graph theory  
Coding FSM  
QA, testing and debugging

**Full-or-part-time:** 35h

Theory classes: 14h  
Self study : 21h



## GRADING SYSTEM

---

- 15% - Assignment 1
- 30% - Assignment 2
- 35% - Assignment 3
- 10% - Final Presentation
- 10% - Attitude and Participation

WARNING: This subject does not feature any content that can be reevaluated.

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.

If the lecturers have indications of the use of AI tools not allowed in the evaluation tests, they may summon the students concerned to an oral test or a meeting to verify the authorship.

## BIBLIOGRAPHY

---

### Basic:

- Robert Nystrom. Game Programming Patterns. Genever Benning, 2014. ISBN 0990582906.

### Complementary:

- Clinton Keith. Agile Game Development: Build, Play, Repeat. Pearson Education Limited, 2020. ISBN 0136527817.

## RESOURCES

---

### Hyperlink:

- <http://www.uml.org/>- <http://www.proyectosagiles.org/>- <https://github.com/raysan5/raylib/wiki>. Raylib Wiki
- <https://www.raylib.com/examples.html>. Raylib examples
- <https://www.raylib.com/cheatsheet/cheatsheet.html>. Raylib cheatsheet