Course guide
804246 - MVJ - Game Engines

Last modified: 10/09/2025

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: 2025 **ECTS Credits:** 6.0 **Languages:** Catalan, English

LECTURER

Coordinating lecturer: Manish Thani

Others: Manish Thani
Jordi Bach

PRIOR SKILLS

Coding in C++. Previous knowledge and experience coding 2D games.

TEACHING METHODOLOGY

During the classes, the teacher will set out the objectives on a theoretical level and the problem to which we are looking for a solution. Together with the students, the teacher will analyse existing solutions that solve the challenges of real-time applications such as video games.

The teacher will provide source code that the students will be able to analyse and must complement and integrate into their own code for future use. After each session, the teacher will pose possible improvements and challenges to the students to help them and to guide them in their autonomous learning hours.

LEARNING OBJECTIVES OF THE SUBJECT

- Describe the operation of a game engine

- To identify graphic libraries for the creation of synthetic images, 2D and 3D graphics and user interfaces.

- To use programming languages, algorithmic patterns, data structures, visual programming tools, game engines, libraries and servers for the development of video games of different genres and for different platforms and devices.

- To analyse the technical characteristics of game engines and tools for the creation of safe and quality video games.

- To develop video games, through the use of programming languages, algorithmic patterns, data structures, visual programming tools, game engines, libraries and servers.

STUDY LOAD

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

Type	Hours	Percentage
Hours large group	18,0	12.00

Total learning time: 150 h

CONTENTS

OpenGL basics

Description:
Initialization
Rendering in Core Profile mode
Vertex Buffers

Full-or-part-time: 15h
Theory classes: 6h
Self study : 9h

Loading 3D models

Description:
Loading of model information: geometry and materials
Rendering of single models

Full-or-part-time: 10h
Theory classes: 4h
Self study : 6h

Camera and scene loading

Description:
Free roaming camera, FPS style and single model
Loading scene information
Execution in threads

Full-or-part-time: 15h
Theory classes: 6h
Self study : 9h

Basic rendering optimizations

Description:
Frustum culling
Level of details
Octree

Full-or-part-time: 10h
Theory classes: 4h
Self study : 6h

Animation systems

Description:
Implementing a Transformation Tree
Structure of an animation system
Loading of animations
Playing and blending of animations

Full-or-part-time: 25h
Theory classes: 10h
Self study : 15h

Component structure and player control

Description:
Component system for entities
Messaging and event system
Physics and player control

Full-or-part-time: 35h
Theory classes: 17h
Self study : 18h

3D Audio

Description:
Loading and playing music
Playing 3D effects

Full-or-part-time: 15h
Theory classes: 9h
Self study : 6h

Graphics effects

Description:
Particle systems
Postprocess effects
Illumination models

Full-or-part-time: 25h
Theory classes: 13h
Self study : 12h

ACTIVITIES

First assignment

Description:
First assignment about scene loading (GameObjects and components) with a weight of 20%.

Full-or-part-time: 12h
Self study: 6h
Theory classes: 6h

Second assignment

Description:
Second assignment about space optimizations, time management, mouse picking and optimized formats with a weight of 20%.

Full-or-part-time: 12h
Self study: 6h
Theory classes: 6h

Third assignment

Description:
Third assignment about a single high level system to choose from: animation, particles, audio, scripting, physics, shaders or UI with a weight of 20%.

Full-or-part-time: 12h
Self study: 6h
Theory classes: 6h

GRADING SYSTEM

Final exam with a weight of 30% with all subject knowledge will be put to test.

First assignment about scene loading (GameObjects and components) with a weight of 20%.

Second assignment about space optimizations, time management, mouse picking and optimized formats with a weight of 20%.

Third assignment about a single high level system to choose from: animation, particles, audio, scripting, physics, shaders or UI with a weight of 20%.

The final exam can be reevaluated for its weight of 30%. In case of passing the course, the maximum final mark will be a 5.

Attitude and class participation will weight 10%

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:
- Gregory, Jason. Game engine architecture. 2nd ed. Boca Raton: CRC Press, Taylor and Francis Group, cop. 2014. ISBN 9781466560017.
- Nystrom, Robert. Game programming patterns. [United States?]: Genever Benning, 2014. ISBN 9780990582908.