



# Course guide

## 804237 - DESVJ - Game Development

**Last modified:** 16/07/2023

**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:** 2023    **ECTS Credits:** 6.0    **Languages:** Spanish, English

### LECTURER

---

**Coordinating lecturer:** Pedro Omedas

**Others:**

### DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

---

**Specific:**

CEVJ 5. Use programming languages, algorithmic patterns, data structures, visual programming tools, game engines and libraries for the development and prototyping of video games, in any genre and for any platform and mobile device.

**Generical:**

CGFC5VJ. Efficiently design and use the most appropriate types and structures of data to solve a problem related to the development of video games.

**Transversal:**

07 AAT N2. SELF-DIRECTED LEARNING - Level 2: Completing set tasks based on the guidelines set by lecturers. Devoting the time needed to complete each task, including personal contributions and expanding on the recommended information sources.

### TEACHING METHODOLOGY

---

Throughout the course, students will collaborate in pairs to work on the development of a 2D platform game using C++ and the SDL library. In each class, the lecturer will demonstrate various approaches and share best practices for implementing functionalities in their projects. To support their learning, the lecturer will provide source code examples and templates for the students to study, complete, and integrate into their own source code for future reference and practical application.

At the end of each session, the lecturer will offer ideas for enhancing the game systems, presenting challenges that encourage students to explore and improve their projects independently. This guidance aims to assist and guide students during their self-learning time, enabling them to develop their skills further and foster a sense of autonomy.

By following this teaching approach, students not only gain hands-on experience in game development, but also receive direct guidance and exposure to industry-standard techniques and practices. The combination of collaborative work, practical examples, and independent learning opportunities ensures a well-rounded and engaging learning experience throughout the course.

## LEARNING OBJECTIVES OF THE SUBJECT

- Develop a comprehensive understanding of the architecture of a 2D game engine, including its main components and their roles in game development.
- Acquire the skills to extend the capabilities of the game engine by adding new features such as entities, GUIs, maps, and physics systems, enabling the creation of engaging gameplay experiences.
- Gain knowledge about the fundamental code components that structure a video game and comprehend their interrelationships, enabling the creation of cohesive and functional game systems.
- Learn programming patterns and techniques that promote efficiency, flexibility, and code quality, ensuring the development of high-quality and stable video games.
- Explore specific approaches for implementing pathfinding algorithms, enabling game entities to navigate the game world intelligently.
- Apply the acquired concepts and techniques in practice by implementing a 2D platform video game, consolidating the learned skills through hands-on experience.

By focusing on these objectives, students will develop a strong foundation in video game programming. They will understand the underlying architecture of game engines, gain the ability to add new features, structure code effectively, employ programming patterns, ensure stability and quality, and implement pathfinding algorithms. The practical application of these concepts through the development of a 2D platform video game will reinforce the learned material and provide a tangible demonstration of the acquired skills.

## STUDY LOAD

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

**Total learning time:** 150 h

## CONTENTS

### Controlling the FPS and timing the logic

**Description:**

How to control the frame rate.  
Ways of manipulating the timing of the logic (pause, bullet time, etc.)

**Full-or-part-time:** 13h

Theory classes: 5h  
Self study : 8h

### Controlling game entities

**Description:**

Theory behind the entity systems for video games.  
Coding a full featured entity system.

**Full-or-part-time:** 19h

Theory classes: 9h  
Self study : 10h



### Serialization

**Description:**

Theory behind the art of loading resources in video games  
Formats (XML, JSON, YAML)  
Serialization of objects using libraries  
Load and save data

**Full-or-part-time:** 16h

Theory classes: 6h  
Self study : 10h

### Loading and rendering Tiled maps

**Description:**

Usage of Tiled to create 2D maps  
Introduction to the TMX file format  
Code to load data from TMX files  
Methodology to render ortogonal maps  
Methodology to render isometric maps

**Full-or-part-time:** 22h

Theory classes: 8h  
Self study : 14h

### Meta information and mask maps

**Description:**

Using Tiled for storing meta information.  
Loading of meta information for navigation.  
Alternative case of using mask maps for navigation.

**Full-or-part-time:** 13h

Theory classes: 5h  
Self study : 8h

### Física

**Description:**

Integrating the Box2D physics engine  
Learn how to use functions from the physics library

**Full-or-part-time:** 8h

Theory classes: 4h  
Self study : 4h



### Pathfinding algorithms

**Description:**

BFS (Breadth First Search)  
Dijkstra  
A\*

**Full-or-part-time:** 30h

Theory classes: 12h  
Self study : 18h

### Graphical User Interface systems

**Description:**

Windows with scroll.  
Buttons with images.  
Textboxes.  
Progress bars.

**Full-or-part-time:** 25h

Theory classes: 10h  
Self study : 15h

### Optimization

**Description:**

Profiling, memory leaks detection

**Full-or-part-time:** 4h

Theory classes: 2h  
Self study : 2h

## ACTIVITIES

### Assignment 1

**Description:**

Create a simple platformer game with the following elements:

- Config file loading
- Data serialization
- Tiled TMX map loading and drawing (orthographic)
- Map collisions detection (platforms)
- Map navigation: player movement and jumping
- Game should be capped to stable 60 frames per second without vsync.

**Full-or-part-time:** 10h

Self study: 10h



## Assignment 2

### Description:

Expanding the platformer from the previous assignment we need to add:

- Walking enemy type that can pathfind to the player. It is not needed that the enemy can jump (although is encouraged) but it should detect that it can reach the player by normal walking and falling down to other platforms.
- Flying enemy type that can pathfind to the player avoiding non-walkable areas.
- Load/Save must consider each enemy state. Enemies normally have a range of perception and not react to the player until they are close by.

**Full-or-part-time:** 10h

Self study: 10h

## Assignment 3

### Description:

Expanding the platformer from the previous assignment we need to add:

- Entity System
- GUI: Title Screen Main Menu
- GUI: Gameplay Screen HUD
- GUI: Gameplay Screen Pause Menu
- Game profiling

**Full-or-part-time:** 10h

Self study: 10h

## GRADING SYSTEM

Three assignments with a weight of 20%, 20% and 20% each of the final grade.

One final examination with a total weight of 30% of the final grade. It will consist of a two-hour practical and theoretical test.

One reevaluation with a total weight of 30% of the final grade (final exam). It will consist of a two-hour practical and theoretical test.

In case of passing the course, the maximum final mark will be 5.

A final 10% grade will be about class participation and attitude.

## BIBLIOGRAPHY

### Complementary:

- Gregory, J. Game engine architecture. 2nd ed. Boca Raton: CRC Press, 2014. ISBN 9781466560017.
- McShaffry, M.; Graham, D. Game coding complete. 4th ed. Boston, Mass: Course Technology, 2012. ISBN 9781133776574.
- Thorn, A. Game engine design and implementation. Sudbury, Mass: Jones & Bartlett Learning, 2011. ISBN 9780763784515.