



## Course guides

# 804237 - DESVJ - Game Development

**Last modified:** 16/09/2021

**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:** 2021    **ECTS Credits:** 6.0    **Languages:** Catalan, 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

---

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

---

Develop the capabilities to code the main components of a video game.

Gain the knowledge about the basic code components that structure a video game and the relationship to each other in order to create the final result.

Learn to structure the code in the most efficient and flexible way to create results of high quality and stability.



## STUDY LOAD

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

**Total learning time:** 150 h

## CONTENTS

### Loading resources and the XML format

**Description:**

Theory behind the art of loading resources in video games  
The XML format  
The JSON format  
Parsing XML with the help of a library

**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:** 12h

Theory classes: 4h  
Self study : 8h



### Controlling the FPS and timing the logic

**Description:**

How to control de 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

### Pathfinding algorithms

**Description:**

BFS (Breadth First Search)  
Dijkstra  
A\*

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

Theory classes: 12h  
Self study : 18h

### Controlling game entities

**Description:**

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

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

Theory classes: 6h  
Self study : 10h

### 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

### Real time tweaking systems

**Description:**

Cvar system.  
Console to be able to introduce commands in real time.  
Menu system to tweak values in real time.

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

Theory classes: 6h  
Self study : 10h



## ACTIVITIES

### Assignment 1

**Description:**

Create a simple platformer game with the following elements:

- XML config file loading
- Load/Save game state using XML file
- Tiled TMX map loading and drawing (orthographic)
- Map collisions detection (platforms)
- Map navigation: player movement and jumping

**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.
- Game should be capped to stable 60 frames per second without vsync.
- Window title must show: FPS / Avg. FPS / Last-frame MS / Vsync: on/off
- Game should have all it's movement normalized using dt (deltaTime), so in slow/fast machines it would keep the same movement speed.

**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

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

Self study: 10h

## GRADING SYSTEM

Three assignments with a weight of 15%, 15% and 30% 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 revaluation 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.