Information about the course

Degree: Degree in Design and Narration in Animation and Video games

Faculty: Faculty of Legal, Economic and Social Sciences

Code: 2050435 **Name:** 3D video game programming

Credits: 6,00 ECTS **Year:** 4 **Semester:** 1

Module: PROGRAMACIÓN DE VIDEOJUEGOS

Subject Matter: PROGRAMACIÓN **Type:** Obligatoria

Branch of knowledge:

Department: Multimedia and Digital Arts

Type of learning: Classroom-based learning

Language/-s in which it is given: Spanish

Teachers:

2054A Jose Luis Soler Domínguez **(Profesor responsable)** joseluis.soler@ucv.es

Module organization

PROGRAMACIÓN DE VIDEOJUEGOS

Subject Matter	ECTS	Subject	ECTS	Year/semester
FUNDAMENTOS DE LA PROGRAMACIÓN	12	Arithmetic foundations of video game programming	6	3/1
		Programming foundations	6	3/1
PROGRAMACIÓN	30	2D video game programming	6	3/2
		3D video game programming	6	4/1
		Artificial Intelligence for Video Games	6	4/1
		Online game programming	6	4/1
		Virtual reality	6	4/2

Other types of requirements

Para cursar las asignaturas: Programación de videojuegos 3D, Inteligencia artificial para videojuegos, Programación de juegos en red y Realidad virtual, se recomienda haber superado las asignaturas de: Fundamentos de programación y Fundamentos aritméticos para programación de videojuegos.

Learning outcomes

At the end of the course, the student must demonstrate having acquired the following learning outcomes:

R12 - Establish the rules of a three-dimensional video game, taking into account the sincere search for the whole truth and the integration of all dimensions of the human being in the face of life's big questions, applying the principles derived from the concept of integral ecology and respecting and putting into practice the ethical principles and proposals for action derived from the sustainable development goals. RA12.91 / RA6.29 / RA7.29 / RA8.36

Learning outcomes of the specified title

Type of AR: Habilidades o Destrezas

- Illustrate and generate specific animation and video game projects using traditional procedures and digital techniques

Type of AR: Competencias

- Apply the principles derived from the concept of integral ecology to their proposals or actions, regardless of their scope, area of knowledge, and the contexts in which they are presented.
- Develop theoretical and practical responses based on the sincere search for the full truth and the integration of all dimensions of the human being in response to life's major questions.
- Respect and implement the ethical principles and action proposals derived from the Sustainable Development Goals, applying them to all academic and professional activities.

R13 - Create levels and scenarios for three-dimensional video games, correctly applying the principles of structural design. RA12.92

Learning outcomes of the specified title

Type of AR: Habilidades o Destrezas

- Illustrate and generate specific animation and video game projects using traditional procedures and digital techniques

Assessment system

In-person modality

Assessed learning outcomes	Granted percentage	Assessment tool
R10, R11, R12, R13	20,00%	SE1 – Written exams.
R10, R11, R12, R13	30,00%	SE6 – Practical exams.
R10, R11, R12, R13	50,00%	SE8 – Project development.

Observations

SINGLE ASSESSMENT:
In accordance with Article 9 of the General Regulations for Assessment and Grading of Official Courses and UCV Qualifications, single assessment is linked to the inability of students enrolled in a face-to-face course to attend. It is, therefore, an extraordinary and exceptional assessment system available to students who, for justified and accredited reasons, are unable to undergo the continuous assessment system and who request it from the professor responsible for the subject, who will expressly decide on the admission of the student's request for single assessment and will notify them of its acceptance or rejection.
As far as the subject of 3D Video Game Programming is concerned, the minimum attendance requirement is 75%, which is therefore the limit to be taken into consideration for any potential request for a single assessment. If granted, the alternative assessment will be based on:
Presentation of the 3 official game jams (individually)
Presentation of an integrative 3D video game project, to be defined together with the professor responsible for the subject.
Written report entitled ‘3D video game engines, evolution, present and trends’.
USE OF AI:
In the 3D Video Game Programming course, and in accordance with the University's ‘Fundamental

Principles for the Use of Artificial Intelligence', AI is considered a complementary tool to enhance learning, not to replace the student's effort and reasoning.
The use of AI tools is permitted for: Consulting conceptual questions about algorithms, data structures, or principles of 3D programming.
Generating sample code snippets to understand a specific functionality, provided that the student analyses, adapts, and integrates them, demonstrating their understanding.
Obtaining assistance in debugging their own code to identify logical or syntax errors.
Seeking inspiration or generating basic ideas for game mechanics, assets, or level structures in the early stages of prototyping.
Creating 3D assets, textures, or other artistic elements useful for prototypes, but always indicating their origin.
The use of AI tools is not permitted for: Generating all or most of the source code for practical assignments, projects, or any other assessable activity.
Create scripts or any other assessable material and present it as your own work without substantial creative and technical contribution on your part.
Directly solve the problems or exercises proposed in assessment tests.
The intellectual authorship and final responsibility for all work submitted rests exclusively with the student. Submitting material generated mainly by AI without significant personal contribution will be considered a violation of originality and will be treated in accordance with current academic regulations, similar to plagiarism. In accordance with the principle of mandatory transparency, if AI has been used as support in the permitted ways, the student must include a note in the submission specifying which tools were used and how they contributed to the development of their work (e.g. 'ChatGPT was used to generate a basic example of camera movement, which was subsequently modified and expanded by the student').
Honours: The distinction of 'Honours' may be awarded to students who have obtained a grade equal to or higher than 9.0. The number of students receiving this distinction may not exceed five per cent of the students enrolled in a group in the corresponding academic year, unless the number of students enrolled is lower.

MENTION OF DISTINCTION:

The mention of "Honors" may be awarded to students who have obtained a grade equal to or greater than 9.0. Their number may not exceed five percent of the students enrolled in a group in the corresponding academic year, unless the number of students enrolled is lower.

Training activities

The methodologies to be used so that the students reach the expected learning outcomes will be the following:

M2 MD2: Interactive lecture

M4 MD4: Problem-solving exercises

M5 MD5: Case studies

M6 MD6: Project-based learning

IN-CLASS TRAINING ACTIVITIES

ACTVITY	RELATIONSHIP WITH THE COURSE LEARNING OUTCOMES	METHODOLOGY	HOURS	ECTS
AF2 – Active listening, elaboration and formulation of questions, summaries, concept maps and/or notes that organize the information received, and related work.	R10, R11, R12, R13	MD2: Interactive lecture MD4: Problem-solving exercises	12,00	0,48
AF5 – Analysis of exemplary realities — real or simulated — allowing the student to connect theory with practice, learn from real-world models, or reflect on the processes used in the presented cases.	R10, R11, R12, R13	MD4: Problem-solving exercises	12,00	0,48
AF6 – The student, individually or collectively, focuses on producing a tangible final result (product) that incorporates the knowledge and skills necessary for its realization.	R10, R11, R12, R13	MD6: Project-based learning	36,00	1,44
TOTAL			**60,00**	**2,40**

TRAINING ACTIVITIES OF AUTONOMOUS WORK

ACTVITY	RELATIONSHIP WITH THE COURSE LEARNING OUTCOMES	METHODOLOGY	HOURS	ECTS
AF8 – Independent work. Study, memorization, exam preparation, practice of practical skills, preparation of assignments, essays, reflections, metacognitive activities, portfolio development, etc.	R10, R11, R12, R13	MD2: Interactive lecture MD4: Problem-solving exercises	16,00	0,64
AF6 – The student, individually or collectively, focuses on producing a tangible final result (product) that incorporates the knowledge and skills necessary for its realization.	R10, R11, R12, R13	MD6: Project-based learning	18,00	0,72
AF5 – Analysis of exemplary realities — real or simulated — allowing the student to connect theory with practice, learn from real-world models, or reflect on the processes used in the presented cases.	R10, R11, R12, R13	MD4: Problem-solving exercises MD5: Case studies	56,00	2,24
TOTAL			**90,00**	**3,60**

Description of contents

Description of content necessary for the acquisition of learning outcomes.

Theoretical content:

Block of content	Contents
Introduction to the Professional Video Game Development Sector and Game Engines	This thematic block offers a panoramic view of the video game development sector, one of the most dynamic and fastest-growing creative and technological industries worldwide. The professional ecosystem will be explored, identifying the different roles and profiles that collaborate in the creation of a video game, from programmers and artists to designers and producers. In turn, the fundamental concept of a "game engine" will be introduced, analyzing its role as the technological backbone of any interactive project. The characteristics, advantages, and use cases of the most relevant engines on the market, such as Unity and Unreal Engine, will be studied, thus laying the foundation to understand the tools and workflows that define contemporary professional production.
Basic Unity UI + IDE (Visual Studio)	In this fundamental block, we will dive into the main work environment for any Unity developer. We will take a guided tour of the engine's user interface (UI), identifying and explaining the purpose of its essential windows: the Scene view for the visual composition of the game world, the Hierarchy to organize objects, the Inspector to modify their properties and components, and the Project window to manage assets. Simultaneously, we will establish the connection with the quintessential Integrated Development Environment (IDE) for Unity, Visual Studio. We will learn how to create, edit, and navigate the C# scripts that will bring our projects to life, understanding the essential workflow between editing code in Visual Studio and its application and testing within the Unity editor. Mastering this interaction is the first crucial step to bringing any idea into a functional video game.

Basic Scripting in C#	This block delves into the pillars of interactive programming using C#, Unity's primary language. Starting from the basics, students will learn fundamental syntax, including variable declaration, handling essential data types (int, float, bool, string), and implementing logical and arithmetic operators. It will go deeper into flow control structures, such as conditional statements (if-else) and loops (for, while), which are crucial for building any game logic. The focus will be on practical application within the engine, exploring the MonoBehaviour class and its key lifecycle methods, such as Start() for initialization and Update() for logic that executes every frame. Upon completion, the student will be able to write functional scripts to manipulate GameObjects, respond to player input, and create the first game mechanics.
Game Jam I: Platforms with a twist	Individual project
Game Jam II: Idle4health	Group project
Game Jam III: Global Game Jam	Group project
Lighting, Animations and Sounds	This module focuses on the audiovisual pillars that transform a functional prototype into an immersive and polished experience. **Lighting** techniques will be explored to create atmospheres, define the visual tone, and guide the player, working with different types of lights and shadows in Unity. Next, we will delve into the **animation** system to bring characters and objects to life, learning to create and control movement sequences through the Animator. Finally, the integration of **sounds** will be addressed, from audio effects (SFX) that provide instant feedback to background music that sets the mood, thus completing the design of a believable and engaging game world.

Physics: Forces and collisions	This block introduces students to Unity's physics engine, an essential component for creating dynamic and reactive worlds. The Rigidbody component will be explored, which allows objects to be affected by gravity and other external influences. Students will learn to programmatically apply **forces** to simulate pushes, jumps, or explosions, thus controlling the movement of objects realistically. In addition, the **collision** system will be covered in depth, configuring Colliders to define the physical shape of objects and detect interactions. The difference between solid collisions and triggers will be explained, and the corresponding C# events (OnCollisionEnter, OnTriggerEnter) will be used to execute game logic, such as inflicting damage, collecting an item, or activating an event in the scene. Upon completion, the student will be able to implement game mechanics based on credible physical interactions.

Temporary organization of learning:

Block of content	Sessions	Hours
Introduction to the Professional Video Game Development Sector and Game Engines	2	4,00
Basic Unity UI + IDE (Visual Studio)	2	4,00
Basic Scripting in C#	4	8,00
Game Jam I: Platforms with a twist	7	14,00
Game Jam II: Idle4health	5	10,00
Game Jam III: Global Game Jam	6	12,00
Lighting, Animations and Sounds	2	4,00
Physics: Forces and collisions	2	4,00

References

http://www.csharpcourse.com/
https://unity.com/es/campaign/unity-6-resources
Unity 6 Game Development with C# Scripting: Leverage C# scripting in Unity to create immersive games and VR experiences by Lem Apperson (Autor)