Course guide
804247 - DMOB - Mobile Devices

Last modified: 13/09/2024

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: 2024 **ECTS Credits:** 6.0 **Languages:** Spanish, English

LECTURER

Coordinating lecturer: Cristian Pérez.

Others: Cristian Pérez.
Eric Batllé.

PRIOR SKILLS

High level programming knowledge.

REQUIREMENTS

Programación I, Programación II, Proyecto 1: Casual Games, Desarrollo de Videojuegos y Proyecto 2: Videojuego 2D.

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.
CGFC9VJ. Apply the principles, methodologies and life cycles of software engineering in video game development.

Transversal:
04 COE. EFFICIENT ORAL AND WRITTEN COMMUNICATION. Communicating verbally and in writing about learning outcomes, thought-building and decision-making. Taking part in debates about issues related to the own field of specialization.

TEACHING METHODOLOGY

Sessions with lectures intertwined with programming practice

LEARNING OBJECTIVES OF THE SUBJECT

- Be able to identify and evaluate the technical characteristics of game engines as a technology for creating video games.
- Show understanding and be able to explain how a game engine works.
- Be able to make extensions and modifications to game engines.
- Show mastery and be able to use game engines for creating video games.
- Show knowledge and be able to use libraries for creating user interfaces.

STUDY LOAD

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

Total learning time: 150 h

CONTENTS

1. Unity Intro.

Description:
The “Unity - Intro” block will train students in the essential handling of the Unity Editor, allowing them to create and manage scenes in 2D and 3D spaces. They will learn how to build and transform GameObjects, apply components and scripts, and employ basic physics. The configuration of cameras and lighting systems to achieve realistic effects and specific environments will be explored in depth, along with the implementation of 3D audio. In addition, the block will cover the creation and management of prefabs, and the use of Light and Reflection Probes.

Full-or-part-time: 0h 21m
Theory classes: 0h 04m
Practical classes: 0h 10m
Self study : 0h 07m

2. C# + Architecture

Description:
The "C# and ARCHITECTURE" block will focus on understanding and applying advanced object-oriented programming (OOP) principles with C#. Students will explore key concepts such as classes and instances, superclasses, abstract classes, polymorphism, interfaces and composition, as well as creational, structural and behavioral design patterns. Through theoretical and practical sessions, students will work in groups to apply these patterns to the previously developed game, presenting and discussing their implementation proposals, with a focus on patterns such as Factory and Command.

Full-or-part-time: 0h 38m
Theory classes: 0h 06m
Practical classes: 0h 02m
Self study : 0h 30m

3. Advanced Unity.

Description:
The "UNITY - ADVANCED" block will cover advanced Unity development techniques, such as creating editor tools, using property drawers, and setting up assemblies and tests. Students will learn about Scriptable Objects through a game, and explore asynchronous techniques with Coroutines, Async/Await, and UniTask. They will also cover managing content loading and unloading with addressables for mobile optimization and perform profiling of these to improve game performance.

Full-or-part-time: 0h 38m
Theory classes: 0h 10m
Practical classes: 0h 06m
Self study : 0h 22m

4. Final Project Intro.

Description:
The “FINAL PROJECT - INTRO” block will guide students through the first phase of their final project, starting with group formation and paper design of a minigame, focusing on gameplay. Key requirements will be established such as a 2D single-player game, with a scoring system, and a lobby that allows selecting between multiple games, even if they only implement one. Students will create the project and implement navigation between screens such as Start, Lobby, Meta, and Game, leaving gameplay for later sessions. The block will culminate with the integration of minigames between teams, a roundtable, and a postmortem to reflect on the development process.

Full-or-part-time: 0h 36m
Theory classes: 0h 03m
Practical classes: 0h 11m
Self study : 0h 22m

5. Final Project - Final.

Description:
The "FINAL PROJECT - FINAL" block will be based on the explanation of DI and SOLID principles for development. With this new knowledge, each group will redo their final practice, integrating the improvements learned.

Full-or-part-time: 0h 17m
Theory classes: 0h 03m
Practical classes: 0h 05m
Self study : 0h 09m

ACTIVITIES

Oral Presentation 1.

Full-or-part-time: 1h
Practical classes: 1h

Oral Presentation 2 + Code.

Full-or-part-time: 1h
Practical classes: 1h

Individually made video game.

Full-or-part-time: 1h
Practical classes: 1h

Final team project 1.

Full-or-part-time: 1h
Practical classes: 1h

Final team project 2.

Full-or-part-time: 1h
Practical classes: 1h

GRADING SYSTEM

Mid-term exam: 20%
Deliverables: 70%
- Oral presentation 1. 5%.
- Oral presentation 2 + Code. 5%.
- Video game done individually. 20%.
- Final project by teams 1. 10%.
- Final project by teams 2. 30%.

Participation and disposition: 10%

Students who fail will have the chance to take the reevaluation exam. The mark of this exam will replace the mark of the midterm exam. In case of passing the course, the maximum final mark will be 5.

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 C. Martin. Clean Code.. Anaya Multimedia, 2012.

RESOURCES

Audiovisual material:
- Unity Oficial. https://unity.com/es- Unity Learn. https://learn.unity.com/

Hyperlink:
- Unity - Manual. https://docs.unity3d.com/Manual/- Unity Tutorials.. https://www.youtube.com/@unity- Asset Store. https://assetstore.unity.com/- C# Learn.. https://learn.microsoft.com/en-us/dotnet/csharp/