Course guide
804254 - XJO - Networks and Online Games

Last modified: 18/05/2026

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: 2026 **ECTS Credits:** 6.0 **Languages:** Catalan, English

LECTURER

Coordinating lecturer: Sergi Abadal

Others: Sergi Abadal
Manuel Rello
Laura Gotarra

PRIOR SKILLS

To be able to programme and develop computer applications
To know and be the main Internet tools and services at user level

REQUIREMENTS

Programming.

TEACHING METHODOLOGY

The teaching methodology is divided in four parts:

- Sessions for the contents exposition at classroom.
- Practical working sessions at classroom (training for the use of the tools, presentations and use case debates).
- Practical development of on-line games where the concepts learnt during the course will be applied.
- Autonomous work to study and carry out exercises and activities.

In the content presentation sessions, teachers will introduce the theoretical bases of the subject, concepts, methods and results, illustrating them with convenient examples to facilitate their understanding. This also includes a couple sessions where students will choose a topic related to the contents of the course and will present it to the rest of the class.

In the practical work sessions in the classroom, the teacher will guide the student in the application of the theoretical concepts for problem solving, based at all times on critical reasoning. A large project cut into with multiple deliverables will be proposed that the student solves in the classroom and outside the classroom, in order to foster the contact and use of the basic tools necessary for the resolution of problems.

The student, autonomously, must work on the material provided by the teacher and the result of the work-problem sessions in order to assimilate and fix the concepts. Teachers will provide a study and activity monitoring plan (CAMPUS).

The methodology of the subject is a combination between the face-to-face class and the realization of different complementary activities (resolution of cases, problems, etc.), either individually or in small groups, both inside and outside the classroom. In the contact hours the student receives the basic information to work on the contents of the teaching plan that guarantee the achievement of the established objectives.

In the Virtual Campus all the information relative to the subject is published (rules of evaluation, dates of delivery of the activities, etc.). Attendance at the sessions is important for the learning process, which is based on both individual or team work and classroom participation.

LEARNING OBJECTIVES OF THE SUBJECT

- To identify game servers and protocols for the development of online video games.

- To apply the basic techniques of parallel, concurrent, distributed and real-time programming, in the development of video games for local or online games.

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

- To use network game engines in the development of online games.

STUDY LOAD

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

Total learning time: 150 h

CONTENTS

1. Introduction

Description:
1.1 Presentation of the subject and evaluation criteria
1.2 A brief history of networks and on-line games
1.3 Architectures and metrics
1.4 Graph theory to model networks
1.5 Graph algorithms

Specific objectives:
To obtain basic knowledge about the history of the Internet and on-line games, about the basic types of network architectures and the metrics used to measure their performance. To learn about the fundamentals of graph theory and algorithms that operate over graphs, which are common to networks and many other domains.

Related activities:
Programming with threads

Full-or-part-time: 12h
Theory classes: 4h
Self study : 8h

2. The OSI Protocol Stack

Description:
2.1 Introduction. Application and transport layers
2.2 The lower layers: network, link, and physical
2.3 Other relevant protocols (NAT, HTTP, QUIC, etc.)
2.4 Putting it all together: Wireshark

Specific objectives:
To understand the layered structure used in networks, the process of encapsulation, as well as the usefulness of it all.
To learn about the functionality of each layer of abstraction, as well as to know the classical protocols in each layer, paying special attention in those that are relevant for the programming of on-line games (IP, TCP/UDP, NAT).
To make the connection between the game's data and the network through the data serialization process and the socket.
To learn about the use of packet sniffers, and to employ them to observe the packet structure in real networks

Related activities:
2. Sockets (TCP/UDP)
3. Data serialization
(it also includes practical NAT concepts)

Full-or-part-time: 24h
Theory classes: 8h
Self study : 16h

3. Impact of the network on the programming of on-line games

Description:
3.1 Introduction and general aspects. Object replication
3.2 Latency, jitter, and reliability
3.3 Latency hiding techniques
3.4 Scalability and massive multiplayer

Specific objectives:
To understand how the network impacts on the programming of on-line games through the study of problems related to the loss of information or delays.
To learn about techniques used to hide the network problems, minimizing their impact so that the game performance is not affected.
To learn about widespread techniques in nowaday's games, be them with a few or many simutaneous players.

Related activities:
4. World-State Replication
5. Latency and jitter mitigation techniques
6. Complete online game

Full-or-part-time: 30h
Theory classes: 10h
Self study : 20h

4. Theory presentations

Description:
Students will deepen their knowledge on one of the topics of the course, choosen by them and agreed with the professor, to then perform a brief presentation during class.

Specific objectives:
- Deepen the knowledge on a specific topic related to networks and online games.
- Collect data and study the topic in an autonomous way.
- Present the topic in a clear way.

Related activities:
N/A

Full-or-part-time: 18h
Theory classes: 6h
Self study : 12h

ACTIVITIES

1. Threads

Description:
Learning how to use parallel code execution mechanisms (threads) and the necessary tools to maintin a proper synchronization amongst them, and to safely access shared memory regions in a concurrent fashion.

Specific objectives:
- Threads (mechanisms for parallel/concurrent code execution)
- Mutex objects (mechanism for safe concurrent data access)
- Condition variables (mechanism for synchronization among threads)

Material:
Instructions for the exercise and template code.

Delivery:
Deliverable #1 (no weight on the final score)

Full-or-part-time: 6h
Practical classes: 2h
Self study: 4h

2. Sockets (TCP/UDP)

Description:
Learning the main mechanisms to use TCP and UCP sockets to transmit and receive data between two or more machines over the network. This activity will set the foundations of the communication of information between client and server in the context of a final project.

Specific objectives:
- Sockets TCP (connection oriented)
- Sockets UDP (non-connection oriented)
- Non-blocking usage of sockets functions (non-blocking mode, select).

Material:
Instructions for the exercise and code template.

Delivery:
Deliverable #2 (5% final score)

Full-or-part-time: 12h
Practical classes: 4h
Self study: 8h

3. Data serialization

Description:
Here we will see the importance of data serialization to transmit objects in memory that may be potentially complex, as these objects could not be represented contiguously in memory, and can contain pointers to other objects, thus making impossible copying them directly at once. This activity will expand the project adding the tools necessary to serialize, at least, one of the videogame's objects.

Specific objectives:
- Sequential and ordered data access.
- MemoryStream objects.

Material:
Instructions for the exercise and code template.

Delivery:
Deliverable #3 (5% of final score)
Deliverable #4 - Demo I (10% of final score)

Full-or-part-time: 18h
Practical classes: 6h
Self study: 12h

4. World-State Replication

Description:
Learn and apply good practice methods to correctly replicate the world-state of our game. This activity will expand the project to add the methods needed to serialize the data of any object of our world and to guarantee the correct reception at the client side.

Specific objectives:
- Definition of a replication manager
- Object replication types (active/passive)
- Object-oriented programming

Material:
Instructions and definitions.

Delivery:
Deliverable #5 (10% of final score)

Full-or-part-time: 12h
Practical classes: 4h
Self study: 8h

5. Latency and jitter mitigation techniques

Description:
Here we'll see the importance of issues such as latency or jitter. The student will implement techniques to mitigate these and other problems derived from the communication in online games. This activity will be an important part of the final project.

Specific objectives:
- Client-side Prediction
- Server reconciliation
- Entity Interpolation
- Lag Compensation

Material:
Instructions for the lab sessions and auxiliary code

Delivery:
Deliverable #6 (10% of final score)

Full-or-part-time: 12h
Practical classes: 4h
Self study: 8h

6. Complete online game

Description:
We will use the concepts learnt up to this point (threads, sockeets, and data serialization) to develop multiplayer online game over the game's foundation developed over the semester. This deliverable will be evaluated taking into consideration the integration of the previous activities in the context of a complete game.

Specific objectives:
- Client-server architecture.
- Authoritative server.
- Game state replication.
- Messages among hosts.
- Data synchronization mechanisms (commands, syncvars and hooks, client RCPs, etc).

Material:
Lab session instructions.

Delivery:
Deliverable #7 - Demo II (10% of final score)

Full-or-part-time: 18h
Practical classes: 6h
Self study: 12h

GRADING SYSTEM

Lab sessions: 50%
(of which: Sockets 5%; data serialization 15%, which includes a mid-term demo; world-state deliverable 10%; latency and jitter management deliverable 10%; complete game demo 10%)
Theory presentation (participation): 10%
Partial exam (theory): 15%
Final exam: 25%

All activities are evaluated by the professors, with the exception of the mid-term and final demos where the evaluation will combine feedback from professors and students.

In case the subject is not passed through the continuous evaluation, there is the option to perform a re-evaluation exam of the theoretical part, corresponding to 40% of the subject's mark (partial and final exams). In case of passing the course through re-evaluation, 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.

EXAMINATION RULES.

The activities are assessed on an individual basis, although the theory presentation and the lab deliverables are done in groups.
For the practices it is allowed to use all learning material and their submission must be done on the indicated deadline with no option for extension.
Exams will be done with no access to any learning material.

BIBLIOGRAPHY

Basic:
- Tanenbaum, A.S.; Wetherall, D.J. Computer networks. 5th ed. Boston: Pearson, 2011. ISBN 9780132126953.
- Glazer, Josh; Madhav, Sanjay . Multiplayer game programming: Architecting networked games. Addison-Wesley Professional, 2015. ISBN 9780134034331.

RESOURCES

Other resources:
Additional readings that will be published in the Campus Virtual during the course, including:
- The history of Internet
- I shot you first: The Networking of HALO
- 1500 Archers on a 28.8: Network Programming in Age of Empires and Beyond
- The TRIBES Engine Networking Model