Last update: 11-11-2019

804227 - PROG2VJ - Programming II

Coordinating unit: 804 - CITM - Image Processing and Multimedia Technology Centre

Teaching unit: 804 - CITM - Image Processing and Multimedia Technology Centre

Academic year: 2019

Degree: BACHELOR'S DEGREE IN VIDEO GAME DESIGN AND DEVELOPMENT (Syllabus 2014). (Teaching unit Compulsory)
BACHELOR'S DEGREE IN VIDEO GAME DESIGN AND DEVELOPMENT (Syllabus 2014). (Teaching unit Compulsory)

ECTS credits: 6 Teaching languages: Catalan, Spanish, English

Teaching staff

Coordinator: Díaz García, Jesús

Degree competences to which the subject contributes

Specific:

3. (ENG) Utilizar lenguajes de programación, patrones algorítmicos, estructuras de datos, herramientas visuales de programación, motores de juego y librerías para el desarrollo y prototipado de videojuegos, de cualquier género y para cualquier plataforma y dispositivo móvil.

Generical:

1. (ENG) Interpretar los fundamentos del uso y programación de los computadores, los sistemas operativos, las bases de datos y, en general, los programas informáticos con aplicación en ingeniería.
2. (ENG) Interpretar i dominar els conceptes bàsics de matemàtica discreta, lògica, algorísmica i complexitat computacional, i la seva aplicació per al tractament automàtic de la informació per mitjà de sistemes computacionals i la seva aplicació per a la resolució de problemes propis de l'enginyeria.

Transversal:

4. 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.
5. FOREIGN LANGUAGE: Achieving a level of spoken and written proficiency in a foreign language, preferably English, that meets the needs of the profession and the labour market.
7. EFFECTIVE USE OF INFORMATION RESOURCES: Managing the acquisition, structuring, analysis and display of data and information in the chosen area of specialisation and critically assessing the results obtained.

Teaching methodology

Lectures are divided in 2h sessions. During lectures, the teacher presents the theoretical concepts and explains them by means of examples that are solved in class. Some time is also dedicated to the resolution of exercises by the students with the assistance of the teacher (solve the doubts that may appear).

An extensive use of campus virtual is mandatory, since the material of the course (slides, exercises, exams, etc.) and the communication between the students and the teacher are done through this system.

Learning objectives of the subject

· Consolidate the basic knowledge on pointers and memory management.
· Understand the basics on object oriented programming and be able to apply them adequately.
· Learn to use and implement the basic data structures: arrays, sequential data structures (stacks, queues and lists), trees and graphs.

Last update: 11-11-2019

804227 - PROG2VJ - Programming II

· Learn the basis of recursion, searching and fractal generation.
· Learn, analyse and be able to implement and adapt the main sorting and search algorithms.

Study load

Total learning time: 150h	Hours large group:	24h	16.00%
	Hours medium group:	16h	10.67%
	Hours small group:	0h	0.00%
	Guided activities:	20h	13.33%
	Self study:	90h	60.00%

Last update: 11-11-2019

804227 - PROG2VJ - Programming II

Content

1. Pointers and memory management

Learning time: 8h

Theory classes: 4h
Guided activities: 4h

Description:
Description:
· Pointers
· Pointers and references as parameters
· Management of memory dynamically: allocation and deallocation of resources

2. Object oriented programming

Learning time: 18h

Theory classes: 8h
Guided activities: 10h

Description:
Description:
· Classes and modular design
· Overloading of operators
· Inheritance
· Polymorfism
· Friendship and templates

3. Data structures

Learning time: 16h

Theory classes: 8h
Guided activities: 8h

Description:
Description:
· Strings
· Stacks and queues
· Lists
· Dynamic arrays
· Trees

804227 - PROG2VJ - Programming II

4. Recursion and Fractals	Learning time: 8h Theory classes: 4h Guided activities: 4h

Description:
Structure of recursive algorithms
Recursion in numeric series
The Hanoi Towers
Fractals: The Cantor set and the Sierpinski triangle

5. Sorting algorithms	Learning time: 10h Theory classes: 6h Guided activities: 4h

Description:
Description:
· Bubblesort
· Quicksort
· Binary Tree
· Heap Sort
· Merge Sort

Last update: 11-11-2019

804227 - PROG2VJ - Programming II

Planning of activities

Activities about pointers and memory

Hours: 8h
Practical classes: 4h
Self study: 4h

Description:
Two sets of questions and small exercices will allow to practice with the concepts of pointers and memory management.
The first set of exercices corresponds to the implementation and management of basic variables and pointers to them, modifying both their address and the content of the memory where they point to.
The second set of exercices includes structs, defyning variables that will be either instances of those struct or pointers to them, allowing to start working with the operators for accessing to the members of the struct, that latter on will be used for the objects and their members.

Support materials:
Statement of the exercices and MS Visual C++

Descriptions of the assignments due and their relation to the assessment:
The output of this activity is a set of C++ files to be uploaded to the delivery section in the campus and eventually presented in front of the group in the classroom. These deliveries and presentations will be used to measure the 2% of the mark corresponding to the student's participation.

Specific objectives:
To practice with the theoretical contents expossed and get practice with the implementation of programmes in C++

Activities of Foundations of Object Oriented Programming

Hours: 18h
Practical classes: 8h
Self study: 10h

Description:
Set of programming exercices to practice with the concepts of encapsulation, inheritance and polymorphism. Among others they will target the representation and manipulation of bubbles, capsules, vehicles, Pokemons and vectors.

Support materials:
Statement of the exercices and MS Visual C++

Descriptions of the assignments due and their relation to the assessment:
The output of this activity is a set of C++ files to be uploaded to the delivery section in the campus and eventually presented in front of the group in the classroom. These deliveries and presentations will be used to measure the 2% of the mark corresponding to the student's participation.

Specific objectives:
To practice with the theoretical contents expossed and get practice with the implementation of programmes in C++

Activities of Data Structures

Hours: 18h
Self study: 8h
Practical classes: 10h

804227 - PROG2VJ - Programming II

Description:
Questionnaire (quiz game) in the classroom with questions to review the concepts explained.
Set of programming exercices to implement objects linked to basic structures and to practice by using them with some examples, namely: i) Stacks, ii) Queues, iii) Lists, iv) Dynamic Arrays, v) Matrix and vi) Trees.
These exercices will contribute to reinforce the training in object oriented programming.

Support materials:
Statement of the exercices and MS Visual C++

Descriptions of the assignments due and their relation to the assessment:
The output of this activity is a set of C++ files to be uploaded to the delivery section in the campus and eventually presented in front of the group in the classroom. These deliveries and presentations will be used to measure the 2% of the mark corresponding to the student's participation.

Specific objectives:
To practice with the theoretical contents expossed and get practice with the implementation of programmes in C++
Be able to identify the apropiate data structures to solve different problems and situations

Activities for recursion and fractals

Hours: 8h
Practical classes: 4h
Self study: 4h

Description:
To Practice and solve some problems and games intrinsically resursive.
Set of programming exercices for the implementation of solutions to the problems and games used, including numerical series, Hanoi towers and fractals

Support materials:
Statement of the exercices and MS Visual C++

Descriptions of the assignments due and their relation to the assessment:
The output of this activity is a set of C++ files to be uploaded to the delivery section in the campus and eventually presented in front of the group in the classroom. These deliveries and presentations will be used to measure the 2% of the mark corresponding to the student's participation.

Specific objectives:
Acquire the understanding of the recursion programming concept
Be able to implement recursive algorithms

Activities for sorting algorithms

Hours: 8h
Practical classes: 4h
Self study: 4h

Description:
Practice different sorting methods in mumerical series
Set of programming exercices to implement and use i) bubblesort, ii) heapsort y iii) quicksort
Include the data structures required in each sorting method

Support materials:
Statement of the exercices and MS Visual C++

804227 - PROG2VJ - Programming II

Descriptions of the assignments due and their relation to the assessment:
The output of this activity is a set of C++ files to be uploaded to the delivery section in the campus and eventually presented in front of the group in the classroom. These deliveries and presentations will be used to measure the 2% of the mark corresponding to the student's participation.

Specific objectives:
Identify and use the apropiate data structures for each sorting method
To know how to apply and understand the functioning of the recursive algorithms in the sorting methods that use them
Be able to use the sorting algorithms to sort different objects in different scenarios

Qualification system

There are three types of tasks to measure the progress of the student:
- Four test (practical exercises) to evaluate specific topics of the course that are completed in class (PE)
- A midterm exam that is done in the midterms week specified in the academic calendar (ME)
- The final exam (FE)

The mark of the subject (M) is computed as follows:
M = 0.2 * ME + 0.4 * PE + 0.4 * FE + 0.1 * PART

where ME is the mark of the midterm exam, PE are the 4 practical exercises that will be done throughout the course (note that the mark each one corresponds to the 10% of the final mark of the subject) and FE is the mark of the final exam. Participation and learning attitude (PART) is also considered. This part will be evaluated according to the participation of the student in class (solving exercises, proposing solutions/alternatives), the interest shown in learning the different topics, etc.

The students who do not pass the course will have the possibility to do a reevaluation exam (90% corresponding to the exams is evaluated and 5 will be the maximum mark of the subject).

Regulations for carrying out activities

Conducting different tests will be conducted exclusively through the mechanisms established on the dates and times indicated.

Bibliography

Basic:

Stroustrup, B. A tour of C++. Addison-Wesley, 2013. ISBN 9780321958310.

Kernighan, Brian W; Ritchie, Dennis M. The C programming language. 2nd ed. New Jersey: Prentice Hall, 1988. ISBN 0131103628.

Franch Gutiérrez, Xavier [et al.]. Fonaments de programació: problemes resolts en C++ [on line]. Barcelona: Edicions UPC, 2006 [Consultation: 10/12/2018]. Available on: <http://hdl.handle.net/2099.3/36692>. ISBN 9788483018828.

Others resources:

C++ Reference: http://www.cplusplus.com/reference/
C and C++ Programming: http://www.cprogramming.com/