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)
ECTS credits: 6

Teaching languages: Catalan, Spanish, English

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)
ECTS credits: 6

Teaching languages: Catalan, Spanish, English

Degree competences to which the subject contributes

Specific:
3. 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.

General:
1. Interpret the basics of the use and programming of computers, operating systems, databases and in general, computer programs with applications for engineering.
2. Interpret and master the basics of discrete mathematics, logic, algorithmics and computational complexity, and their application to the automatic processing of information using computer systems and their application for solving engineering problems.

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.
• Learn the basis of recursion, searching and fractal generation.
804227 - PROG2VJ - Programming II

- Learn, analyse and be able to implement and adapt the main sorting and search algorithms.

<table>
<thead>
<tr>
<th></th>
<th>Total learning time: 150h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours large group:</td>
<td>24h</td>
</tr>
<tr>
<td>Total learning time:</td>
<td>150h</td>
</tr>
<tr>
<td>Hours medium group:</td>
<td>16h</td>
</tr>
<tr>
<td>Hours small group:</td>
<td>0h</td>
</tr>
<tr>
<td>Guided activities:</td>
<td>20h</td>
</tr>
<tr>
<td>Self study:</td>
<td>90h</td>
</tr>
</tbody>
</table>

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%
| 1. Pointers and memory management | **Learning time:** 8h  
| Theory classes: 4h  
| Guided activities: 4h |
| 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:  
| · Classes and modular design  
| · Overloading of operators  
| · Inheritance  
| · Polymorphism  
| · Friendship and templates |

| 3. Data structures | **Learning time:** 16h  
| Theory classes: 8h  
| Guided activities: 8h |
| Description:  
| · Strings  
| · Stacks and queues  
| · Lists  
| · Dynamic arrays  
| · Trees |
### 4. Recursion and Fractals

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

**Learning time:** 8h
- Theory classes: 4h
- Guided activities: 4h

### 5. Sorting algorithms

**Description:**
- Bubblesort
- Quicksort
- Binary Tree
- Heap Sort
- Merge Sort

**Learning time:** 10h
- Theory classes: 6h
- Guided activities: 4h
### Planning of activities

| Activities about pointers and memory | Hours: 8h  
Practical classes: 4h  
Self study: 4h |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td></td>
</tr>
<tr>
<td>Two sets of questions and small exercises will allow to practice with the concepts of pointers and memory management.</td>
<td></td>
</tr>
<tr>
<td>The first set of exercises 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.</td>
<td></td>
</tr>
<tr>
<td>The second set of exercises includes structs, defining 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.</td>
<td></td>
</tr>
<tr>
<td><strong>Support materials:</strong></td>
<td></td>
</tr>
<tr>
<td>Statement of the exercises and MS Visual C++</td>
<td></td>
</tr>
<tr>
<td><strong>Descriptions of the assignments due and their relation to the assessment:</strong></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td><strong>Specific objectives:</strong></td>
<td></td>
</tr>
<tr>
<td>To practice with the theoretical contents exposed and get practice with the implementation of programmes in C++</td>
<td></td>
</tr>
</tbody>
</table>

| Activities of Foundations of Object Oriented Programming | Hours: 18h  
Practical classes: 8h  
Self study: 10h |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td></td>
</tr>
<tr>
<td>Set of programming exercises 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.</td>
<td></td>
</tr>
<tr>
<td><strong>Support materials:</strong></td>
<td></td>
</tr>
<tr>
<td>Statement of the exercises and MS Visual C++</td>
<td></td>
</tr>
<tr>
<td><strong>Descriptions of the assignments due and their relation to the assessment:</strong></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td><strong>Specific objectives:</strong></td>
<td></td>
</tr>
<tr>
<td>To practice with the theoretical contents exposed and get practice with the implementation of programmes in C++</td>
<td></td>
</tr>
</tbody>
</table>

| Activities of Data Structures | Hours: 18h  
Self study: 8h  
Practical classes: 10h |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Description:
Questionnaire (quiz game) in the classroom with questions to review the concepts explained.  
Set of programming exercises 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 exercises will contribute to reinforce the training in object oriented programming.

## Support materials:
Statement of the exercises and MS Visual C++

## Activities for recursion and fractals
<table>
<thead>
<tr>
<th>Description:</th>
<th>Hours: 8h</th>
</tr>
</thead>
</table>
| To practice and solve some problems and games intrinsically recursive.  
Set of programming exercises for the implementation of solutions to the problems and games used, including numerical series, Hanoi towers and fractals | Practical classes: 4h  
Self study: 4h |

## Support materials:
Statement of the exercises 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 exposed and get practice with the implementation of programmes in C++  
Be able to identify the appropriate data structures to solve different problems and situations

## Activities for sorting algorithms
<table>
<thead>
<tr>
<th>Description:</th>
<th>Hours: 8h</th>
</tr>
</thead>
</table>
| Practice different sorting methods in numerical series  
Set of programming exercises to implement and use i) bubble sort, ii) heapsort y iii) quick sort  
Include the data structures required in each sorting method | Practical classes: 4h  
Self study: 4h |

## Support materials:
Statement of the exercises 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:

- Identify and use the appropriate 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:

\[ M = 0.2 \times ME + 0.4 \times PE + 0.4 \times FE + 0.1 \times \text{PART} \]

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:


Others resources:

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