## 270019 - IDI - Interaction and Interface Design

### Coordinating unit: 270 - FIB - Barcelona School of Informatics
### Teaching unit: 723 - CS - Department of Computer Science
### Academic year: 2018
### Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Teaching unit Compulsory) BACHELOR'S DEGREE IN ENGINEERING PHYSICS (Syllabus 2011). (Teaching unit Optional)
### ECTS credits: 6
### Teaching languages: Catalan, Spanish

#### Teaching staff

**Coordinator:** - Marta Fairen Gonzalez (mfairen@cs.upc.edu)

**Others:**  
- Alejandro Ríos Jerez (arios@cs.upc.edu)  
- Alvaro Vinacua Pla (alvar@cs.upc.edu)  
- Amalia Duch Brown (duch@cs.upc.edu)  
- Antonio Chica Calaf (achica@cs.upc.edu)  
- Bernardino Casas Fernández (bcasas@cs.upc.edu)  
- Carlos Andujar Gran (andujar@cs.upc.edu)  
- Isabel Navazo Alvaro (isabel@cs.upc.edu)  
- M. Jose Blesa Aguilera (mjblesa@cs.upc.edu)  
- Nuria Pelechano Gomez (npelechano@cs.upc.edu)  
- Pere Pau Vázquez Alcocer (ppau@cs.upc.edu)  
- Ricard Gavaldà Mestre (gavalda@cs.upc.edu)  
- Robert Joan Arinyo (robert@cs.upc.edu)

#### Prior skills

EDA capabilities of the subjects (Data Structures and Algorithms) and M2 (Math 2). In particular, it is expected that the student is able to:  
- Program correctly by using Object-Oriented Programming in C++.  
- Understand and know how to implement basic data structures, lists, stacks, etc..  
- Learn the basics of linear algebra, geometric transformations and changes of base.

#### Requirements

- Prerequisite M1  
- Prerequisite EDA

#### Degree competences to which the subject contributes

**Specific:**

CT1.2A. To interpret, select and value concepts, theories, uses and technological developments related to computer science and its application derived from the needed fundamentals of mathematics, statistics and physics. Capacity to solve the mathematical problems presented in engineering. Talent to apply the knowledge about: algebra, differential and integral calculus and numeric methods; statistics and optimization.

CT2.5. To design and evaluate person-computer interfaces which guarantee the accessibility and usability of computer systems, services and applications.

CT4.1. To identify the most adequate algorithmic solutions to solve medium difficulty problems.

CT5.2. To know, design and use efficiently the most adequate data types and data structures to solve a problem.

CT5.3. To design, write, test, refine, document and maintain code in an high level programming language to solve programming problems applying algorithmic schemas and using data structures.
270019 - IDI - Interaction and Interface Design

CT5.4. To design the programs' architecture using techniques of object orientation, modularization and specification and implementation of abstract data types.  
CT5.5. To use the tools of a software development environment to create and develop applications.

General:  
G8. APPROPRIATE ATTITUDE TOWARDS WORK: to have motivation to be professional and to face new challenges, have a width vision of the possibilities of the career in the field of informatics engineering. To feel motivated for the quality and the continuous improvement, and behave rigorously in the professional development. Capacity to adapt oneself to organizational or technological changes. Capacity to work in situations with information shortage and/or time and/or resources restrictions.

Teaching methodology
In theory sessions the teacher will introduce theoretical concepts and, where appropriate, exercises and problems. The theory is strongly tied to the developments proposed laboratory.

As support for the theory classes, especially for the interaction and design, notes or articles will be provided available via El Racó a few days earlier. These notes are designed for the students to have available during the classes, but at least on paper. During the development of classes, not to disturb the normal operation will not be allowed to use mobile phones or portable devices of any kind.

Students are expected to prepare additional materials that will be provided in the form of notes or other documents in order to prepare for classes, exams and will serve as documentation practices to perform correctly. In the lab, introduced the software to use and pose a number of small practices that students must develop and eventually deliver. Much time will be devoted to laboratory students solve practices raised with the help of the teacher.

Laboratory tests allow the use of code that students have developed independently. Sharing this code between students is considered copying.

Learning objectives of the subject
1. Learn to program interfaces in a high-level programming language and using a specific API. Design and implement interactive applications using APIs to interface design.
2. Learn the basics of Computer Graphics. Implementing simple applications that display data in OpenGL 3D.
3. Knowing the architecture of current GPUs.
4. Be able to design applications with different user interfaces for common as cell phones or tablets.
5. Getting a proactive attitude to quality and continuous improvement.
6. To be able to adapt to technological or organizational changes.
7. Ability to adapt in situations of time constraints and / or resources and / or lack of information.
9. Understand the concept of usability and knowledge to assess whether an interface of an application or a Web page is usable. Be able to assess the usability of a particular interface.
### Study load

<table>
<thead>
<tr>
<th>Total learning time: 150h</th>
<th>Hours large group:</th>
<th>30h</th>
<th>20.00%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hours medium group:</td>
<td>0h</td>
<td>0.00%</td>
</tr>
<tr>
<td></td>
<td>Hours small group:</td>
<td>30h</td>
<td>20.00%</td>
</tr>
<tr>
<td></td>
<td>Guided activities:</td>
<td>6h</td>
<td>4.00%</td>
</tr>
<tr>
<td></td>
<td>Self study:</td>
<td>84h</td>
<td>56.00%</td>
</tr>
<tr>
<td>Content</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Introduction to interactive systems</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Degree competences to which the content contributes:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description:</strong></td>
<td>Basics of interactive systems. Introduction to Human Computer Interaction.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Ubiquitous computing and augmented reality</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Degree competences to which the content contributes:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description:</strong></td>
<td>Definitions. Systems Virtual Reality and Augmented Reality.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Design user interfaces</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Degree competences to which the content contributes:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description:</strong></td>
<td>Introduction to Usability. Basic principles of designing user interfaces. Design Rules.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Colour models</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Degree competences to which the content contributes:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description:</strong></td>
<td>Representations of color. Translation between models.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Programming Interface</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Degree competences to which the content contributes:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description:</strong></td>
<td>Introduction to Qt. Advanced Programming with Qt interface.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Processing and visualization of 2D and 3D geometry</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Degree competences to which the content contributes:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description:</strong></td>
<td>Introduction to Computer Graphics. Visualization using OpenGL.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Developing user-centered

**Degree competences to which the content contributes:**

**Description:**

### Architecture and programming of graphics cards

**Degree competences to which the content contributes:**

**Description:**
Introduction to GLSL. Vertex shaders and fragment shaders.
### Planning of activities

<table>
<thead>
<tr>
<th>Activity</th>
<th>Hours</th>
<th>Description</th>
<th>Specific objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Geometry Processing.</strong></td>
<td>10h</td>
<td>Studying the different forms of modeling scenes and objects. Learn the elements of the processing of 2D and 3D geometry: triangle meshes. Review data structures. Analysis of different data structures for geometry. Studying the elements of an interactive graphic system. Deploying simple OpenGL and 3D.</td>
<td>2</td>
</tr>
<tr>
<td><strong>3D visualization of data.</strong></td>
<td>22h</td>
<td>Studying the basics of 3D visualization. Learn the camera model. Doing exercises camera model and geometric transformations. To study the programmable visualization pipeline.</td>
<td>2, 3</td>
</tr>
<tr>
<td><strong>Programming graphical applications.</strong></td>
<td>33h</td>
<td>Program data structures that store meshes triangles. Program basically GPU by using GLSL. Understand and use lighting calculations OpenGL. Aprender manage interaction Qt.</td>
<td>2, 3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Theory classes</th>
<th>Practical classes</th>
<th>Laboratory classes</th>
<th>Guided activities</th>
<th>Self study</th>
</tr>
</thead>
<tbody>
<tr>
<td>4h</td>
<td>0h</td>
<td>0h</td>
<td>0h</td>
<td>6h</td>
</tr>
<tr>
<td>12h</td>
<td>0h</td>
<td>0h</td>
<td>0h</td>
<td>10h</td>
</tr>
<tr>
<td>0h</td>
<td>0h</td>
<td>18h</td>
<td>0h</td>
<td>15h</td>
</tr>
</tbody>
</table>
## Introduction to interactive systems.

**Hours:** 7h  
- Theory classes: 2h  
- Practical classes: 0h  
- Laboratory classes: 0h  
- Guided activities: 0h  
- Self study: 5h

**Description:**  
Studying the Human-Computer Interaction notes. Usability study notes. Learn to design graphical interfaces. Learn to perform assessments of usability of applications and websites.

**Specific objectives:**  
9

## Extension or exercises on an interface design theme.

**Hours:** 10h  
- Theory classes: 0h  
- Practical classes: 0h  
- Laboratory classes: 0h  
- Guided activities: 0h  
- Self study: 10h

**Description:**  
Solve exercises or study an interface design theme autonomously.

**Specific objectives:**  
1, 4, 5, 6, 7, 9

## Test Teo1: Control on graphics interaction and interface design.

**Hours:** 7h  
- Guided activities: 3h  
- Self study: 4h

**Description:**  
Written exercise that assesses knowledge of graphics interaction and interface design.

**Specific objectives:**  
2, 3, 5, 9

## Test Lab: Laboratory Practice Deployment graphics applications using OpenGL and designing the interface using Qt.

**Hours:** 0h  
- Guided activities: 0h  
- Self study: 0h

**Description:**  
The lab test is deploying an application that uses OpenGL and Qt.

**Specific objectives:**  
1, 2, 3
# Introduction to graphical interfaces.

**Description:**
Learn to design and implement graphical interfaces. Perform practical Qt.

**Specific objectives:**
1

**Hours:**
- Theory classes: 0h
- Practical classes: 0h
- Laboratory classes: 6h
- Guided activities: 0h
- Self study: 6h

<table>
<thead>
<tr>
<th>Hours</th>
<th>Theory classes</th>
<th>Practical classes</th>
<th>Laboratory classes</th>
<th>Guided activities</th>
<th>Self study</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0h</td>
<td>0h</td>
<td>6h</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0h</td>
<td>0h</td>
<td>0h</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1h</td>
<td>0h</td>
<td>0h</td>
<td></td>
</tr>
</tbody>
</table>

# Design user interfaces.

**Description:**
Study Model View Controller. Study notes on usability of applications and websites. Learn to design applications on mobile devices. Evaluating interfaces of mobile devices.

**Specific objectives:**
4, 9

**Hours:**
- Theory classes: 11h
- Practical classes: 0h
- Laboratory classes: 0h
- Guided activities: 0h
- Self study: 8h

<table>
<thead>
<tr>
<th>Hours</th>
<th>Theory classes</th>
<th>Practical classes</th>
<th>Laboratory classes</th>
<th>Guided activities</th>
<th>Self study</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11h</td>
<td>0h</td>
<td>0h</td>
<td>0h</td>
<td>8h</td>
</tr>
</tbody>
</table>

# Introduction to AR, VR and UC

**Description:**
Studying the basics of 3D interaction, computer graphics, virtual reality and augmented reality and make the corresponding exercises. Studying the notes of the course.

**Specific objectives:**
2

**Hours:**
- Theory classes: 1h
- Practical classes: 0h
- Laboratory classes: 0h
- Guided activities: 0h
- Self study: 0h

<table>
<thead>
<tr>
<th>Hours</th>
<th>Theory classes</th>
<th>Practical classes</th>
<th>Laboratory classes</th>
<th>Guided activities</th>
<th>Self study</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1h</td>
<td>0h</td>
<td>0h</td>
<td>0h</td>
<td>0h</td>
</tr>
</tbody>
</table>

# Advanced Programming Interface using OpenGL and Qt.

**Hours:**
- Theory classes: 0h
- Practical classes: 0h
- Laboratory classes: 6h
- Guided activities: 0h
- Self study: 8h

<table>
<thead>
<tr>
<th>Hours</th>
<th>Theory classes</th>
<th>Practical classes</th>
<th>Laboratory classes</th>
<th>Guided activities</th>
<th>Self study</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0h</td>
<td>0h</td>
<td>6h</td>
<td>0h</td>
<td>8h</td>
</tr>
</tbody>
</table>
Last update: 05-02-2019

270019 - IDI - Interaction and Interface Design

**Description:**
Learn advanced programming interfaces. Using Qt signals and slots. Integrating complex widgets advanced interfaces.

**Specific objectives:**
1, 2

<table>
<thead>
<tr>
<th>Prova Teo2: Final Exam.</th>
<th>Hours: 15h</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Guided activities: 3h</td>
</tr>
<tr>
<td></td>
<td>Self study: 12h</td>
</tr>
</tbody>
</table>

**Description:**
Review that will assess the knowledge acquired during the course. It will be a traditional written exam.

**Specific objectives:**
1, 2, 3, 4, 5, 6, 7, 9

**Qualification system**

The mark will consist of 3 parts:
- One lab note: ProvaLab in which students are asked to solve small problems. It will do more emphasis on graphic design but also including interaction and programming interfaces. Laboratory must be resolved and allow the use of code that students have developed independently. Sharing this initial code between students is considered copying. To get to the exam with confidence, students develop scripts provided by teachers.

During the course, a set of laboratory exercises (n) will be specified that the student will have to complete and deliver. If all of these exercises are given, the student will opt for 1 extra point in the laboratory's note provided that in the test of a laboratory (ProvaLab) a score equal to or greater than 4.5 is achieved. If all of these exercises are not delivered and only x of these n are delivered, the formula to decide the portion of the point that can be reached is (x / n) ^ 2.2. For example, if n = 4, the values that would be obtained depending on the number x of deliveries is: 0 - 0; 1 - 0.05; 2 - 0.22; 3-0.53; 4-1.

- Two theory tests: Prova Teo1 and Prova Teo2. Do not use notes or calculators or calculators or mobile devices.

To calculate the final grade, using the following formula, where all the notes are on 10:

\[ 25\% \text{ ProvaLab} + 25\% \text{ Prova Teo1} + 50\% \text{ Prova Teo2} \]

Rating generic competition: You will have values A, B, C or D (where A corresponds to an excellent standard, B corresponds to a desired level, C corresponds to a sufficient level and D corresponds to a level not exceeded). A good evaluation of this competition the students will have: Motivation to perform professionally. Performing professionally. At a theoretical level, they will begin to introduce ideas of the broad career opportunities in the field of computing. On a practical level, when possible, there will be talks by invited professionals.

Proactive attitude to quality and continuous improvement. Motivation. Students must work in terms of lack of information. Ability to adapt to technological or organizational changes. Ability to adapt in situations of lack of information.
Bibliography

Basic:


Complementary:


Others resources:

Hyperlink

http://www.opengl.org

http://www.nngroup.com/


https://www.qt.io/

http://www.smashingmagazine.com/