270019 - IDI - Interaction and Interface Design

Coordinating unit: 270 - FIB - Barcelona School of Informatics
Teaching unit: 723 - CS - Department of Computer Science
Academic year: 2019
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)
           - Marc Comino Trinidad (mcomino@cs.upc.edu)
           - Nuria Pelechano Gomez (npelechano@cs.upc.edu)
           - Pere Pau Vázquez Alcocer (ppau@cs.upc.edu)
           - Ricard Gavalda 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 EDA
- Prerequisite M1

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.

CT5.4. To design the programs\textsuperscript{2} 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.

Generical:
G8. APPROPRIATE ATTITUDE TOWARDS WORK: to have motivation to be professional and to face new challenges, have a width vision of the possibilies 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.

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: 30h</th>
<th>20.00%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hours medium group:</td>
<td>0.00%</td>
</tr>
<tr>
<td></td>
<td>Hours small group:</td>
<td>20.00%</td>
</tr>
<tr>
<td></td>
<td>Guided activities:</td>
<td>6h</td>
</tr>
<tr>
<td></td>
<td>Self study:</td>
<td>84h</td>
</tr>
</tbody>
</table>
## Introduction to interactive systems

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

## Ubiquitous computing and augmented reality

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

## Design user interfaces

**Degree competences to which the content contributes:**
**Description:**
Introduction to Usability. Basic principles of designing user interfaces. Design Rules.

## Colour models

**Degree competences to which the content contributes:**
**Description:**
Representations of color. Translation between models.

## Programming Interface

**Degree competences to which the content contributes:**
**Description:**
Introduction to Qt. Advanced Programming with Qt interface.

## Processing and visualization of 2D and 3D geometry

**Degree competences to which the content contributes:**
**Description:**
Introduction to Computer Graphics. Visualization using OpenGL.
# 270019 - IDI - Interaction and Interface Design

## 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

#### Geometry Processing.

**Description:**
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.

**Specific objectives:** 2

**Hours:** 10h  
Theory classes: 4h  
Practical classes: 0h  
Laboratory classes: 0h  
Guided activities: 0h  
Self study: 6h

#### 3D visualization of data.

**Description:**
Studying the basics of 3D visualization. Learn the camera model. Doing exercises camera model and geometric transformations. To study the programmable visualization pipeline.

**Specific objectives:** 2, 3

**Hours:** 22h  
Theory classes: 12h  
Practical classes: 0h  
Laboratory classes: 0h  
Guided activities: 0h  
Self study: 10h

#### Programming graphical applications.

**Description:**
Program data structures that store meshes triangles. Program basically GPU by using GLSL. Understand and use lighting calculations OpenGL. Aprender manage interaction Qt.

**Specific objectives:** 2, 3

**Hours:** 33h  
Theory classes: 0h  
Practical classes: 0h  
Laboratory classes: 18h  
Guided activities: 0h  
Self study: 15h
## 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
<table>
<thead>
<tr>
<th>Course Title</th>
<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>Introduction to graphical interfaces.</td>
<td>12h</td>
<td>0h</td>
<td>0h</td>
<td>6h</td>
<td>0h</td>
<td>6h</td>
</tr>
<tr>
<td>Description: Learn to design and implement graphical interfaces. Perform practical Qt.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specific objectives:</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design user interfaces.</td>
<td>19h</td>
<td>11h</td>
<td>0h</td>
<td>0h</td>
<td>0h</td>
<td>8h</td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specific objectives:</td>
<td>4, 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Introduction to AR, VR and UC</td>
<td>1h</td>
<td>1h</td>
<td>0h</td>
<td>0h</td>
<td>0h</td>
<td>0h</td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specific objectives:</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advanced Programming Interface using OpenGL and Qt.</td>
<td>14h</td>
<td>0h</td>
<td>0h</td>
<td>6h</td>
<td>0h</td>
<td>8h</td>
</tr>
<tr>
<td>Description:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specific objectives:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Prova Teo2: Final Exam.

**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

**Hours:** 15h
Guided activities: 3h  
Self study: 12h

**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 mobile devices.

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

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

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/