270019 - IDI - Interaction and Interface Design

Coordinating unit: 270 - FIB - Barcelona School of Informatics
Teaching unit: 723 - CS - Department of Computer Science
Academic year: 2015
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 Beacco Porres (abeacco@cs.upc.edu)
- Alejandro Ríos Jerez (arios@cs.upc.edu)
- Alvaro Vinacua Pla (alvar@cs.upc.edu)
- Antonio Chica Calaf (achica@cs.upc.edu)
- Carlos Andujar Gran (andujar@cs.upc.edu)
- Isabel Navazo Alvaro (isabel@cs.upc.edu)
- Nuria Pelechano Gomez (npelechano@cs.upc.edu)
- Oscar Argudo Medrano (oargudo@cs.upc.edu)
- Pere Pau Vázquez Alcocer (ppau@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 design the programs¿ architecture using techniques of object orientation, modularization and specification.
270019 - IDI - Interaction and Interface Design

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 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 program 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></th>
<th>Hours</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total learning time:</strong> 150h</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Theory classes:</td>
<td>30h</td>
<td>20.00%</td>
</tr>
<tr>
<td>Practical classes:</td>
<td>0h</td>
<td>0.00%</td>
</tr>
<tr>
<td>Laboratory classes:</td>
<td>30h</td>
<td>20.00%</td>
</tr>
<tr>
<td>Guided activities:</td>
<td>6h</td>
<td>4.00%</td>
</tr>
<tr>
<td>Self study:</td>
<td>84h</td>
<td>56.00%</td>
</tr>
</tbody>
</table>
# Content

## 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.
### 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>Module</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Geometry Processing.</strong></td>
<td>10h</td>
</tr>
<tr>
<td>Description:</td>
<td></td>
</tr>
<tr>
<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></td>
</tr>
<tr>
<td>Specific objectives:</td>
<td>2</td>
</tr>
</tbody>
</table>

| **3D visualization of data.**                | 20h    |
| 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   |

| **Programming graphical applications.**      | 29h    |
| 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   |
### 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

### Developing an application with portable devices.

**Hours:** 17h  
- Theory classes: 0h  
- Practical classes: 0h  
- Laboratory classes: 2h  
- Guided activities: 0h  
- Self study: 15h

**Description:**  
Design, evaluate and correct a simple application using portable devices (phones, tablets ...).  
**Specific objectives:**  
1, 4, 5, 6, 7, 9

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

**Hours:** 6h  
- Guided activities: 2h  
- Self study: 4h

**Description:**  
Written exercise that assesses knowledge of graphics interaction and interface design.  
**Specific objectives:**  
2, 3, 5, 9

### Test GL: Laboratory Practice Deployment graphics applications using OpenGL.

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

**Description:**  
The second lab is deploying an application that uses OpenGL.  
**Specific objectives:**  
1, 2, 3
# Introduction to graphical interfaces.

<table>
<thead>
<tr>
<th>Hours</th>
<th>12h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory classes:</td>
<td>0h</td>
</tr>
<tr>
<td>Practical classes:</td>
<td>0h</td>
</tr>
<tr>
<td>Laboratory classes:</td>
<td>6h</td>
</tr>
<tr>
<td>Guided activities:</td>
<td>0h</td>
</tr>
<tr>
<td>Self study:</td>
<td>6h</td>
</tr>
</tbody>
</table>

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

**Specific objectives:**
1

---

# Design user interfaces.

<table>
<thead>
<tr>
<th>Hours</th>
<th>19h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory classes:</td>
<td>11h</td>
</tr>
<tr>
<td>Practical classes:</td>
<td>0h</td>
</tr>
<tr>
<td>Laboratory classes:</td>
<td>0h</td>
</tr>
<tr>
<td>Guided activities:</td>
<td>0h</td>
</tr>
<tr>
<td>Self study:</td>
<td>8h</td>
</tr>
</tbody>
</table>

**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

---

# Introduction to AR, VR and UC

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

**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

---

# Advanced Programming Interface using OpenGL and Qt or glut.

<table>
<thead>
<tr>
<th>Hours</th>
<th>14h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory classes:</td>
<td>0h</td>
</tr>
<tr>
<td>Practical classes:</td>
<td>0h</td>
</tr>
<tr>
<td>Laboratory classes:</td>
<td>6h</td>
</tr>
<tr>
<td>Guided activities:</td>
<td>0h</td>
</tr>
<tr>
<td>Self study:</td>
<td>8h</td>
</tr>
</tbody>
</table>
270019 - IDI - Interaction and Interface Design

Description:
Learn advanced programming interfaces. Using Qt signals and slots or glut callbacks. Integrating complex widgets advanced interfaces.

Specific objectives:
1, 2

<table>
<thead>
<tr>
<th>Test Qt: Laboratory Practice programming interfaces with Qt.</th>
<th>Hours: 0h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>Guided activities: 0h</td>
</tr>
<tr>
<td>The first lab will consist of deploying a simple application using Qt.</td>
<td>Self study: 0h</td>
</tr>
<tr>
<td>Specific objectives:</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test LabProject: Deploying an application with a non-traditional interface such as a mobile device or a tablet.</th>
<th>Hours: 0h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>Guided activities: 0h</td>
</tr>
<tr>
<td>Implementation of a complete application with the design, implementation and evaluation of the interface.</td>
<td>Self study: 0h</td>
</tr>
<tr>
<td>Specific objectives:</td>
<td>4, 5, 6, 7, 9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prova Teo2: Final Exam.</th>
<th>Hours: 14h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>Guided activities: 3h</td>
</tr>
<tr>
<td>Review that will assess the knowledge acquired during the course. It will be a traditional written exam.</td>
<td>Self study: 11h</td>
</tr>
<tr>
<td>Specific objectives:</td>
<td>1, 2, 3, 4, 5, 6, 7, 9</td>
</tr>
</tbody>
</table>
Qualification system

The mark will consist of 5 parts:
- Two lab notes: ProvaQt and ProvaGL in which students are asked to solve small problems. The first will be more emphasis on interaction and second graphic design 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.

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

- Prova ProjecteLab. Relates to the work that the student will perform independently and must be submitted via El Racó. It will mainly consist on a development based on mobile devices.

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

\[ \text{final grade} = 0.2 \times \text{ProvaGL} + 0.05 \times \text{ProvaQt} + 0.2 \times \text{ProjecteLab} + 0.15 \times \text{Prova Teo1} + 0.4 \times \text{Prova Teo2} \]

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 submit work that should be in terms of lack of information. Improvement. Students must work practices in stages in which to demonstrate continuous improvement, eventually guided on the previous phases of the practice. Quality. We assessed the quality of delivered code and documentation taking into account mainly: the fulfillment of requirements statements, code clarity and organization of the document. Ability to adapt to technological or organizational changes. Ability to adapt in situations of lack of information. Restrictions. Students must solve some exercises and deliver a pre-determined time and lack of information.
Bibliography

Basic:


Complementary:


Others resources:

Hyperlink

http://www.opengl.org

http://www.nngroup.com/


http://qt.digia.com

http://www.smashingmagazine.com/