270182 - GEOC - Computational Geometry

Coordinating unit: 270 - FIB - Barcelona School of Informatics
Teaching unit: 749 - MAT - Department of Mathematics
Academic year: 2019
Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Teaching unit Optional)
ECTS credits: 6
Teaching languages: English

Teaching staff

Coordinator: - Vera Sacristan Adinolfi (vera.sacristan@upc.edu)
Others: - Rodrigo Ignacio Silveira (rodrigo.silveira@upc.edu)

Prior skills

- Programming
- Basic knowledge of data structures
- Basic knowledge of algorithmic techniques

This course is recommended for students with knowledge and interest in computation. Students with other specializations of without any specialization are kindly asked to take this into account before enrolling.

Students need to do their presentations in English. This course is not recommend for students with very rudimentary English skills.

Degree competences to which the subject contributes

Generical:
G3. THIRD LANGUAGE: to know the English language in a correct oral and written level, and accordingly to the needs of the graduates in Informatics Engineering. Capacity to work in a multidisciplinary group and in a multi-language environment and to communicate, orally and in a written way, knowledge, procedures, results and ideas related to the technical informatics engineer profession.

Teaching methodology

Theory classes will set out the contents of the course, oriented to the resolution of examples and applications.

Exercise classes will be centered in the resolution of problems by the instructors as well as by the students. Students will be assigned problems and will have enough time to think about them in advance, so that they will be able to propose their solutions during the class. The problems will be mainly algorithmic.

The purpose of the lab classes is to implement the solutions discussed in the theory and exercises classes, the effective solution of problems being one of the goals of the course. The problems to be solved in the lab classes will start being of elementary complexity, and will end with the resolution of a problem, preferably applied and real.

Learning objectives of the subject

1. Learn the several kinds of problems in Computational Geometry, as well as their applications.
2. Learn the capacity of combining geometric tools with the appropriated data structures and algorithmic paradigms.
3. See in action several algorithmic paradigms and data structures useful in geometric problems.
4. Apply geometric results to real problems.
270182 - GEOC - Computational Geometry

5. Ability to solve basic problems that appear in computational geometry.
6. Ability to implement the solutions proposed in the class, as well as those that can be found in the basic references of the course.
7. Ability to recognize the geometric problems behind the applications, and to propose adequate algorithmic tools to solve them.
8. Practice and improve the capability of working in an English speaking professional surrounding

<table>
<thead>
<tr>
<th>Study load</th>
<th>Total learning time: 150h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours large group:</td>
<td>45h</td>
</tr>
<tr>
<td>Hours small group:</td>
<td>15h</td>
</tr>
<tr>
<td>Guided activities:</td>
<td>6h</td>
</tr>
<tr>
<td>Self study:</td>
<td>84h</td>
</tr>
</tbody>
</table>
## Introduction to Computational Geometry

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

**Description:**

## A basic tool

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

**Description:**
Oriented area. Left/right. Intersection of two lines. Intersection of two segments. Oriented turn.

## Sweep line algorithms

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

**Description:**
Bentley-Ottmann algorithm

## Basic geometric problems on polygons

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

**Description:**
Line/polygon intersection, point location in a polygon, supporting lines to a polygon from a point, etc.

## Convex hull

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

**Description:**
Algorithms for the construction of the convex hull of 2D point sets.

## Duality. Intersection of halfplanes.

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

**Description:**
Geometric duality. The parabola duality. Intersection of halfplanes and convex hulls.
## Polygon triangulation

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

**Description:**
Triangulation of monotone polygons, decomposition of a polygon into monotone polygons.

## Proximity

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

**Description:**
Voronoi diagrams and their applications

## Triangulations of point sets

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

**Description:**
Delaunay triangulation

## Line and plane arrangements

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

**Description:**
Description, properties, and construction. Levels. Relationship with Voronoi diagrams.

## Point location in planar subdivisions

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

**Description:**
Variety of strategies. Preprocessing complexity vs query efficiency.

## Shape reconstruction

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

**Description:**
Alpha-shapes, crust, anti-crust and beta-skeletons.
**Students presentations of further subjects**

<table>
<thead>
<tr>
<th>Degree competences to which the content contributes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
</tr>
<tr>
<td>Extensions of the course contents.</td>
</tr>
</tbody>
</table>
### Planning of activities

<table>
<thead>
<tr>
<th>Theory presentations</th>
<th>Theory presentations</th>
<th>Solving problems</th>
<th>Solving problems</th>
<th>Lab</th>
<th>Lab</th>
<th>Exam</th>
<th>Exam</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hours:</strong> 57h 30m</td>
<td><strong>Hours:</strong> 40h</td>
<td><strong>Hours:</strong> 49h 30m</td>
<td><strong>Hours:</strong> 49h 30m</td>
<td><strong>Hours:</strong> 3h</td>
<td><strong>Hours:</strong> 3h</td>
<td><strong>Hours:</strong> 3h</td>
<td><strong>Hours:</strong> 3h</td>
</tr>
<tr>
<td>Theory classes: 37h 30m</td>
<td>Theory classes: 0h</td>
<td>Theory classes: 0h</td>
<td>Theory classes: 0h</td>
<td>Theory classes: 0h</td>
<td>Theory classes: 0h</td>
<td>Theory classes: 0h</td>
<td>Theory classes: 0h</td>
</tr>
<tr>
<td>Practical classes: 0h</td>
<td>Practical classes: 15h</td>
<td>Practical classes: 0h</td>
<td>Practical classes: 0h</td>
<td>Practical classes: 0h</td>
<td>Practical classes: 0h</td>
<td>Practical classes: 0h</td>
<td>Practical classes: 0h</td>
</tr>
<tr>
<td>Laboratory classes: 0h</td>
<td>Laboratory classes: 0h</td>
<td>Laboratory classes: 0h</td>
<td>Laboratory classes: 0h</td>
<td>Laboratory classes: 0h</td>
<td>Laboratory classes: 0h</td>
<td>Laboratory classes: 0h</td>
<td>Laboratory classes: 0h</td>
</tr>
<tr>
<td>Guided activities: 0h</td>
<td>Guided activities: 0h</td>
<td>Guided activities: 0h</td>
<td>Guided activities: 0h</td>
<td>Guided activities: 0h</td>
<td>Guided activities: 0h</td>
<td>Guided activities: 0h</td>
<td>Guided activities: 0h</td>
</tr>
<tr>
<td>Self study: 20h</td>
<td>Self study: 25h</td>
<td>Self study: 45h</td>
<td>Self study: 45h</td>
<td>Self study: 45h</td>
<td>Self study: 45h</td>
<td>Self study: 45h</td>
<td>Self study: 45h</td>
</tr>
</tbody>
</table>

#### Description:

**Theory presentations:**
- Students will be in charge of the final sessions.

**Specific objectives:**
- 1, 2, 3, 8

**Solving problems:**
- Only some of the sessions will be run by the instructor. The remaining ones will consist of presentation and discussion of the solutions of problems, done by the students.

**Specific objectives:**
- 5, 7, 8

**Lab:**
- Implementing geometric algorithms

**Specific objectives:**
- 4, 6, 8

**Exam:**
- The exam will cover the material taught in the course.
The evaluation will be based on the work done by the student along the course. The four components to be considered will be:

- Problems presented in class (P)
- Final presentation of the chosen subject (T)
- Lab exercises (L)
- Exam (E)

The final course grade will be calculated as follows:

Final grade = 0.2*P + 0.2*T + 0.35*L + 0.25*E

Bibliography

Others resources:

Hyperlink

https://dccg.upc.edu/people/vera/teaching/courses/computational-geometry/