2010-11 academic year

Computer Graphics (21420)

Degree/study: Bachelor's degree in Computer Sciences (3377)
Year: 2nd
Term: 2nd
Number of ECTS credits: 4 credits
Hours of studi dedication: 100 hours
Teaching language or languages: Spanish and Catalan
Teaching Staff: Marcelo Bertalmío, Narcis Parés, Alun Evans, Pascal Landry

1. Presentation of the subject

Computer Graphics is a subject focused on the application of advanced techniques for the visual information visualization, synthesis and processing.

2. Previous requirements to follow the formative itinerary

It is highly recommended to have a good knowledge on maths, C++, Linux.

3. Competences to be obtained in the subject

- Acknowledges on advanced Computer Graphics techniques.
- Implementation of methods for real applications.
- Review of mathematical tools (geometry, algebra, analysis).
- Practical activities on programming (efficient and clear code, documentation).
- Bibliographic research, understanding scientific papers.
- Teamwork: organization, communication.

4. Evaluation

Evaluation: practical activities (50%), theory (50%), each part must be passed separately. There are three practical activities and they are successively evaluated during the entire term. These activities are carried out in groups of three people and they are evaluated through hand-ins (a report and the mentioned code) and an oral defense (individual questions). If a student doesn't pass the practical activities of the term, he/she will have to do a practical individual work for the September sitting. The theory part is evaluated through a written exam at the end of the term. If a student passes the practical part but fails the theory part, the mark of the practical activities will be kept for the September sitting, but not for the next academic year.

The participation in the seminars will be also evaluated and the final mark of the subject can be increased 1 point if this participation is positive.

5. Contents

1- 3D objects representation and modelling.
2- Light-object interaction and rendering.
3- Ray tracing.
4- Texture mapping.
5- Image-based rendering and 3D model creation by video.
6- Information visualization and physical phenomena simulation.

7- Videogames.
8- Computational photography.
9- Cinema animation and digital effects.

6. Metodology

6.1. Methodology of the lectures

Each lecture session includes one or more expositions on scientific papers, putting emphasis on: -which problem are we trying to solve -which method is proposed to solve it -which are the pros and cons of this method. Most of the papers exposed require basic mathematical tools to understand them, and the presentation of these papers also has the role of refreshing the students mathematical concepts previously acquired. Then, it is an important task and our Syllabus will be very oriented to the practical activities, with few basic mathematical contents. The Computer Sciences studies include the repetition of concepts and mathematical tools in some subjects during the learning process of the students and as a part of the mathematical content (as a result, this content can't be just taught in the strictly mathematical subjects). Referring to the contents of the subject, the teaching process is highly supported by visual tools. The use of slides has been avoided.

6.2. Practical activities

All the practical activities consist on implementing the exposed algorithm in a scientific paper. All the programming material used is open-source and the practical activities must be done in C++ in Linux. This decision has been made because it is thought that students can't be asked to do the practical activities using private or paid programs, because it would mean that they could just work with the university computers (where some licenses for these programs are paid), and this would restrict the possibilities of working conveniently. It would also be an incentive to pirate the programs, and it would mean to invite the students to work at home and buy the programs or obtain illegal copies.

6.3. Practical activities methodology

If we have in mind the objectives of the subject, we have developed the practical activities using the methodology described as follows. There are three practical activities in the subject and they must be done in groups of three people. If the students fail the practical activities, the next sitting will only include one practical activity and it will have to be done individually. In any case, these activities must be defended in a personal interview with the teacher. These sessions, where students defend their hand-in, are the only sessions where the attendance is mandatory. All the practical activities consist of the implementation, from a base point, of a scientific paper. It is only given the bibliographic reference of the paper. The requisites are very general: to hand in a code that works, which from different inputs has different outputs, and also to hand in a report about it. Anything else is asked and it allows the students to have some freedom when designing and checking a program which solves the problem. This type of freedom must logically include the monitoring and supervision by the teacher, as intense as it is asked by each student, but only when the student asks for it. More monitored practical activities would maybe allow implementing more algorithms, or they would mean less work done by the students, but it would mean less quality in the learning process. For the practical sessions, the realization of the activities in groups of three students, instead of an individual work, allows to assign as a task the implementation and the test of an algorism which solves the real problem. Then, the following things are important: -the students feel motivated and rewarded when they can check that its work is working in a real application; -the individual works wouldn't be as ambitious, they would consist only on the implementation of parts of algorithms or only on the solution of specific problems, in general from a code elaborated and given by the teacher: which would mean the decrease of autonomy in the students learning process, and which would also mean a decrease of the motivation as mentioned; -a good organization of the teamwork is essential to complete the practical activities with success: students have to be coordinated when working, avoiding the duplication of tasks, programming modularly and they have to facilitate the intercompatibility of the code.

7. Bibliography and teaching resources

Basic bibliography (on paper or electronic media)

Units (1) to (4):
ALAN WATT, 3D Computer Graphics.

Unit (5):
-"Feature-Based Image Metamorphosis". Thaddeus Beier, Shawn Neely. Proc. ACM SIGGRAPH 1992, pp 35-42.
-"View Morphing", by Steven M. Seitz and Charles R. Dyer, in the Proceedings of SIGGRAPH'96, pages 21-30.
-3D MODELING FROM IMAGES. Marc Pollefeys http://www.cs.unc.edu/~marc/tutorial.pdf

Unit (6):
-Information visualization. Chen, C. Wiley Interdisciplinary Reviews: Computational Statistics, volume={2}, number={4}, pages={387--403}, issn={1939-0068}, year={2010}, publisher={Wiley Online Library}.
-Owens, J. D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A. E. and Purcell, T. J. (2007), A Survey of General-Purpose Computation on Graphics Hardware. Computer Graphics Forum, 26:80-113.

Unit (7):
-Bishop, L.; Eberly, D.; Whitted, T.; Finch, M.; Shantz, M.; , "Designing a PC game engine," *Computer Graphics and Applications, IEEE* , vol.18, no.1, pp.46-53, Jan/Feb 1998
-Xubo Yang; Milo Yip; Xiaoyue Xu; , "Visual Effects in Computer Games," *Computer* , vol.42, no.7, pp.48-56, July 2009
doi: 10.1109/MC.2009.240

Unit (8):
-IEEE Computer, vol. 39:8, 2006: Special issue on computational photography.

Unit (9):
-ALAN WATT, 3D Computer Graphics.
-IEEE Computer, vol.42, no.7: Special issue on visual effects.