

# An Introductory Computer Graphics Course in the Context of the European Space of Higher Education: A Curricular Approach

Akemi Gálvez, Andrés Iglesias, and Pedro Corcuera

Department of Applied Mathematics and Computational Sciences,  
University of Cantabria, Avda. de los Castros, s/n, E-39005, Santander, Spain  
{galveza,iglesias,pcorc}@unican.es

**Abstract.** Currently, European countries are in the process of rethinking their higher education systems due to harmonization efforts initiated by the so-called Bologna's declaration. This process of reforms implies not only a re-evaluation of our way of teaching and learning but also a new curricular design in order to achieve the expected goals. In this context, the present paper focuses on the problem of teaching computer graphics, a discipline that is getting increasing importance in the realm of Computational Science. The paper discusses some issues regarding an introductory course on computer graphics taking into account its goals, contents, students' profile and other factors derived from the new scenario given by the European Space of Higher Education.

## 1 Introduction

Bologna's declaration - seen today as the well-known synonym for the whole process of reformation in the area of higher education - was signed in 1999 by 29 European countries with the objective to create *a European space for higher education in order to enhance the employability and mobility of citizens and to increase the international competitiveness of European higher education* [4]. The upmost goal of this process is the commitment freely taken by each signatory country to reform its own higher education systems in order to create overall convergence at European level. Some major objectives of this approach are:

- the adoption of a common framework of readable and comparable degrees,
- the introduction of undergraduate and postgraduate levels in all countries along with ECTS (European Credit Transfer System) credit systems to ensure a smooth transition from one country's system to another one and
- the promotion of free mobility of students, teachers and administrators among the European countries.

The ultimate goal is *to ensure that the European higher education system acquires a worldwide degree of attractiveness equal to [Europe's] extraordinary cultural and scientific traditions* [4].

Unquestionably, Bologna's declaration opened the door to a completely new scenario for higher education in Europe. Nowadays, the European countries are in the midst of the process of restructuring their higher education system in order to attain the objectives of the declaration. At this time, the developments focus especially on academic aspects, such as the definition of the new curricula and grading systems. Although this issue represents a major concern for all areas of knowledge, it is particularly challenging for the field of Computational Science - understood in its most comprehensive meaning - because of its inherent dynamic nature and the continuous emergence of new topics and subfields within. Among them, Computer Graphics (CG) has established itself as an integral part of Computational Science as well as a solid discipline at its own.

This paper concerns the problem of teaching Computer Graphics in the context of the European Space of Higher Education (ESHE). In particular, we present a proposal for a first (introductory) CG course taking into account its goals, contents, students' profile and other factors which will be discussed in this work. The structure of this paper is as follows: in Section 2 we describe the framework of CG for the last recent years (Section 2.1) as well as the profile of our students (Section 2.2), a key question for the adequate design of any course. Section 3.1 discusses some previous work done about introductory CG courses. Then, we discuss our proposal, which consists of the course's goals, syllabus, the teaching methods and some bibliography and complementary material used in this course (Sections 3.2 to 3.5). The main conclusions in Sec. 4 closes the paper.

## 2 Framework

### 2.1 Computer Graphics in Recent Years

In recent years, CG has found remarkable applications in many other fields: Science, Engineering, Medicine, advertising, entertainment, etc. and the list is expanding rapidly. Current CG hardware is orders of magnitude faster and cheaper, and it is much more robust and powerful than earlier technology. Today, we are used to have powerful graphical cards in our desktop computers and laptops. These graphical cards are especially designed for high performance and often incorporate OpenGL or other APIs (Application Programmer Interfaces) in hardware. In addition, we have witnessed extraordinary advances in CG software:

- the appearance and acceptance of standardized APIs, such as OpenGL, Direct3D, Quick-Draw3D, Java3D, etc.
- extensive general-purpose libraries providing high level graphical capabilities and simplified GUIs (Graphical User Interfaces)
- powerful graphical programs (for instance, 3D Studio Max, LightWave or RenderMan) that provide the users with a large and very powerful collection of tools for design, rendering and animation
- the Web and its wealth of free demos, software tools (POVray, rayshade, VRML), data sets and examples, etc. (see, for instance, [19,29] or the nice ACM Siggraph repository at <http://www.siggraph.org>).

## 2.2 The Students

There is a general consensus that our current students are far different than their previous counterparts in the 1970s, 80s and 90s. From a CG education standpoint, two major features have been pointed out:

- on one hand, today's students exhibit less proficiency in Mathematics, Geometry and other related topics;
- on the other hand, they are much more accustomed to technology.

Regarding the first item, teachers and lecturers have noticed that today's students encounter more problems than former ones in solving questions with mathematical content. In particular, their background (if any) on topics as valuable for CG as Linear Algebra, Analytic Geometry, Euclidean Geometry, Discrete Geometry or Differential Geometry, is nowadays reduced to the minimum in most cases. Students find it difficult to solve logic problems in an axiomatic way, and are less skilled in deduction, mathematical intuition and scientific reasoning. By contrast, they are surrounded by technology since their early days. Technology terms (pixel, navigation, interface) are usual to them at full extent; in fact, they are familiar with all this technological stuff in such a way that frequently cannot imagine the world without it. Today, devices like cell phones, notebook PCs, MP3 players, videogame controllers and fax machines are an essential part of their daily lives. Our Digital Age students have no trouble in using haptic devices such as joysticks or steering wheels and more recently Wii-like game controllers. They can readily identify aliasing artifacts in computer-generated imagery, in the same way only professionals did just a few years ago. They do not need to be trained about texturing, multiresolution or on-line multi-player systems, concepts that can be easily captured from videogames. By the time students access the university they have spent thousands of hours watching TV, playing videogames, using computers and videocameras, surfing at the web, talking by cell phone and sending and receiving e-mails and SMS messages. It is obvious that our approach to teaching CG subjects should take into account this new situation and adapt the course's contents accordingly.

An important factor to be considered is the new role Bologna's declaration assigns to teachers and students. Although current worries focus on "external" topics such as the curricular design, the course's contents and others, the upcoming changes go far beyond these structural changes, as the personal development of students and teachers is also at the roots of this new concept of education. It is clear that the real change should come from the human parties. In what concerns the students, the new philosophy of self-learning and self-teaching demands a new approach to our course's contents.

Another issue is the need to generate a common core of knowledge for each subject so that students can freely move abroad and get their studies recognized. In order to do so, the curriculum must be clearly understood as being "somehow" similar, whatever it means. Of course, it does not mean the course's contents to be the same; on the contrary, Bologna's declaration encourage the diversity as a mean to enrich the global knowledge. The key word of the ESHE is *convergence*,

not *uniformity*. Because of that, the course's contents need not to be similar, but the skills acquired from learning those contents should be. In other words, the subjects must be *equivalent* rather than *equal* and therefore compatibility must be established in these terms.

Other issue is that, as it is their first course on CG, our students are still unaware on the topics and techniques involved. This fact restricts dramatically the course's goals to simply offer a comprehensive overview on the fundamentals of CG as well as a gentle introduction to the main topics and techniques. Consequently, the tools to be used in the course must be carefully chosen in order to prevent students from boredom or discouragement.

### 3 A First Computer Graphics Course

#### 3.1 Previous Work

If courses on all computer science topics have largely evolved in response to the impressive developments in hardware and software, this is particularly true for CG courses. You can, for instance, take a look at the nice paper in [18] to realize how the CG curricula have adapted to changes over the last three decades. Recently, many universities have included elective CG courses in their curricula, in response to the new students' needs and interests. Although this is particularly the case for computer science students, it is still valid for many other scientific studies, such as engineering. To quote just an example close to the authors, the education committees of the some national chapters of Eurographics have contended that CG should be a compulsory subject for computer science studies. This assessment is also shared by many other professional and educational organizations worldwide.

The new regulations for the ESHE have re-opened the discussion about how the CG curriculum should be. This has been a largely vivid debate for years. For instance, the authors participated in a Joint Eurographics/ACM Siggraph panel for CG education in 1999 where the key topic was the definition of a new CG curriculum for the coming years [11]. Subsequent editions of this panel have been held in 2002 [12] and 2004 [13]. We also mention the related event in [14]. The activities of those panels were for good, as new proposals followed up the fruitful discussions [5,7,17,26,30,31].

#### 3.2 Course's Goals

Based on the previous considerations, we think that any proposal for an introductory CG course should restrict its objectives to:

- *offer a comprehensive overview on the fundamentals of CG in terms of basic algorithms and techniques.* Although the definition of “basic algorithms” can be understood in many different ways, it is almost unanimously recognized today that, in spite of their pedagogical value, many fundamental algorithms and procedures are not longer necessary. For instance, Bresenham's algorithms for lines and circles, scan conversion algorithms for polygons, clipping

procedures and similar are currently performed in very low level hardware and, consequently, are not useful to learn graphics software techniques. On the other hand, the original algorithms are very often modified before being implemented in hardware. Lastly, many algorithms designed for optimal performance at software level are today buried in hardware. Because current graphical cards are much more powerful than those from the previous years, some “strategies” for software efficiency are not required anymore.

- *develop the students’ visual sense, which is not usually acquired from most of the traditional courses* [31,32]. At our experience, this visual insight is better acquired when appropriate high-level 3D APIs are applied. In particular, Open GL [22] and Java3D [20] are excellent candidates: they are free and easily available for all students, require little (or none) effort to install, can be used with many standard compilers and tools and support all the fundamental concepts needed for early work.

### 3.3 Syllabus

Table 1 shows a proposal of syllabus for a first (introductory) CG course for undergraduate students. The course is intended to be as general as possible in the sense that not only computer science students but also those from any other technical field (even from arts!) can follow it. Differences lie on how deep and at which extent a specific topic is covered. On the other hand, the course tries to agglutinate as much material as possible (for being an introductory course) so that teachers can select those parts they want to stress more or students are more interested in. Finally, the syllabus has been designed according to Bologna’s declaration principles and recommendations, meaning that the topics are expected to be taught and learned in a participative and cooperative manner and that most of the load falls on the students, who should be able to carry out assignments and a very significant part of the educative process by themselves.

The goal of Chapter 1 is to provide the students with a comprehensive overview about CG. Thus, students are given the basic bibliography so that they can freely study and learn by themselves, the teachers’ role rather being like a tutor and advisor, someone students can go to for questions much like a personalized “Google”. Besides, teachers can focus on tutoring the students with their assignments along with offering them guidance and complementary information that is not easily found in the books and other sources. The chapter is also intended as a motivational chapter; interesting examples of real use of CG in several fields are given. Other sources of information (videotapes, web sites, etc.) are also presented. Finally, the history of CG is revisited by using some educational Siggraph clips and complementary material.

Chapter 2 focuses on the hardware and software used for CG. From the graphical cards and GPUs to haptic devices, virtual reality accessories and the most recent game controllers, students are presented a complete gallery of hardware that accounts for the “machinery” of the field. The second part of the chapter focuses on the software. Special emphasis is placed upon shareware and freeware, with the twofold objective of reducing the institutional budget load required

**Table 1.** Proposed syllabus for an introductory Computer Graphics course

<b>Chapter 1:</b> Introduction to CG. Applications. Information sources. History of CG.
<b>Chapter 2:</b> Hardware and Software for CG and Virtual Reality.
<b>Chapter 3:</b> Basic Algorithms for CG.
<b>Chapter 4:</b> Geometric Modeling: Curves and Surfaces.
<b>Chapter 5:</b> Rendering I: Illumination Models. Ray tracing. Radiosity.
<b>Chapter 6:</b> Rendering II: Texturing and Color.
<b>Chapter 7:</b> Advanced Algorithms for CG.
<b>Chapter 8:</b> Computer Animation (Geometry, Physics, Kinematics, Behavior).
<b>Chapter 9:</b> Virtual characters and avatars. Virtual humans.
<b>Chapter 10:</b> Graphical User Interfaces. APIs. Development Toolkits.
<b>Chapter 11:</b> Multimedia. HD and SW for multimedia.
<b>Chapter 12:</b> Graphical formats for image, video, audio. Industrial formats.
<b>Chapter 13:</b> Fractals and L-systems. Applications.
<b>Chapter 14:</b> Graphics for the Web. Web3D.
<b>Chapter 15:</b> Recent trends. Final project.

for these courses, on one hand, and encouraging students to use free software thus lowering their educational costs. Fortunately, as pointed out by many authors [6,8,18,19,21,28], there is a bulk of very powerful freeware available, including several powerful APIs, such as the universally used and standard *low level* OpenGL (evolved from SGIs proprietary GL and now available on nearly all platforms) or Java3D, graphical and virtual reality markup languages (VRML, X3D, XGL, XDML), visualizers (POVray, rayshade), etc. Other packages can be purchased a very reasonable rates.

One striking thing is the gap of powerful open-source software for occasional CG users with limited (or not at all) experience in programming. The majority of freeware packages for CG, such as those listed above, force the user to write down some source code in order to get something on the screen. In other words, they are well suited for computer science students, who are assumed to have programming abilities and the know-how on programming. This is not necessarily the case for students from other disciplines. For these reasons, a truly introductory course should include some pointers to commercial software for CG (3D Studio Max, Lightwave, Rhinoceros, Renderman and others), useful for end-users with very limited programming experience. This interplay between freeware and commercial software allows us to approach to CG courses in two different (but partially overlapping) ways: one for students with limited programming abilities and/or a short background in related topics, based on the use of intuitive, user-friendly commercial software; and that for programming-skilled students, mostly relying on advances APIs and their supporting framework for GUI development, such as GLUT (GL Utility Library) for OpenGL. In fact, several initiatives support these approaches, from the Pixar University based on their proprietary software Renderman for the first approach to courses based on OpenGL [2,31] or Java3D [18] for the second one.

Chapter 3 focuses on the basic algorithms for CG, including some background on vectors and matrices, 2D and 3D transformations, perspectives, cameras, projections. The main graphical primitives (points, lines, polygons, triangle fans and strips, etc.) are also studied. Based on our previous comments, Bresenham's line and scan conversion algorithms are only briefly mentioned (or completely omitted). The same applies for clipping techniques. By contrast, the visible line and surface algorithms (painter's algorithm, z-buffer) are analyzed.

Chapter 4 talks about computer design. As such, it is a must-be ingredient of any CG course. Careful attention is put on the free-form parametric curves and surfaces, especially NURBS [23], because of their remarkable applications to computer design and industry. Some brief concepts on solid modeling might also be included in this chapter.

Once the geometry of a graphical scene is generated, illuminating it is the next step. Chapter 5 concerns the illumination models used to tackle this issue. From the physics of light and color to sophisticated shading algorithms (ray tracing, radiosity), students are introduced to the fundamental principles of illumination techniques in a essentially visual way. This chapter is also important from the motivational viewpoint, as it is one students enjoy the most. The continuation is given by Chapter 6, where some texturing techniques (texture mapping, bump mapping, environment mapping) are learned. Even the color itself is seen as a (flat) texture.

Chapter 7 presents some advanced algorithms for CG. They range from special effects for movies to the simulation of natural phenomena (water, fire, smoke, fog, etc.). The chapter also introduces some ideas about computer animation, a topic addressed in Chapter 8. Starting with geometry and moving "up" through (forward and backward) kinematics and physics to behavior, the basic concepts of animation are covered in sequence much like steps in a ladder. The top of the sequence is given by the cognition, leading to the animation of virtual humans and avatars, covered in Chapter 9. This has been one of the most challenging tasks in CG for many years and the subject of great efforts very recently.

Chapters 10 to 15 can be considered as part of a second course if needed, as they introduce supplementary (but still very important) material. Chapter 10 concerns CG software, introducing the students to the design of GUIs and the use of APIs and their associated development toolkits. Here instructors can choose some API of their choice or restrict themselves to some commercial software, according to the students' profile. The accompanying Chapters 11 and 12 are about multimedia and graphical formats. In these chapters students can relate their academic and real life learning in a very smooth way, as the learning process happens naturally: topics will answer questions about their daily use of technology.

Chapter 13 introduces the main topics about fractals and L-systems, two subfields that captured mass media interest during the 80s because of their outstanding applications to the simulation of rough, irregular objects and natural structures. In this sense, the chapter is linked with Chapters 7 and 8 so contents

can be swapped between chapters with little or not influence on the general picture of the course.

Chapter 14 covers the interesting issue of graphics for the web. The recent appearance of powerful Web3D technologies is changing our way to approach the web, as web pages have traditionally been seen as unable to support 3D graphics technology. Note that this chapter can also be subsumed into Chapter 11, as the web itself is now regarded as a multimedia interface.

Finally, Chapter 15 aims at reporting some recent trends of interest for the students. Media news might be analyzed in this chapter thus reinforcing the applied, real-life character of the course. The chapter is also devoted to paving the way for the final course project.

### 3.4 Teaching Methods

The course's syllabus depends strongly on the number of teaching hours and their structure. Our course is designed for 75 lectures (of one hour each), of which 30 are given at the classroom and 45 at the computer labs. Each chapter (typically about 5 lectures, depending on the complexity of the topics involved) consists of a visual introduction to the subject, a theoretical explanation of the concepts and techniques, some computer work at the computer labs and homework. Since our objectives are much easier to achieve if the evaluation is based on practical assignments (one for each chapter of the syllabus) and the number of students is relatively low (no more than 30 per course), we have chosen this kind of evaluation. Assignments consist of a brief exposition on some topic or some implementation. Our students are required to perform these assignments by themselves, either as homework or as computer training at our labs. Of course, this structure demands a significant time and effort to both students and teachers, making this approach feasible only for small groups of highly motivated students. At the end of the course, a final project on some of the topics of the course is also required. To this aim, essential ingredients are a good bibliography and additional documentation and material, analyzed in the next section.

### 3.5 Bibliography and Complementary Material

Regarding the bibliography used in this course, students are suggested to use the books in [9,25,27] for a gentle and general introduction to CG. These books also cover most of the course's topics at a general (but still enough) level. On the computer training counterpart, the books in [2,16] are also recommended. Some topics, however, require additional bibliography. For geometric design we recommend the excellent books in [1,23], while illumination models are also analyzed in [15] and in [10] for ray-tracing techniques. Additional bibliography for specific topics, such as fractals [3], could also be used. Additional material comprises ACM Siggraph videotapes, freeware/shareware programs and libraries available from the Web (see above), journals and additional documentation (manuals, presentations, slides, etc.). Finally, the implementations performed by former students are used to illustrate new students about possible course's projects.

## 4 Conclusions

In this paper an introductory CG course is presented. The paper discusses the main issues involved in its design according to the ESHE regulations, such as the resources, goals, students' profile, course's contents, teaching methods and bibliography and complementary material. We also consider very seriously the suggestions in [24]. Roughly, they establish that creating a single curriculum to meet all needs could be counterproductive, taking into account the different institutional resources, student's background, skills, goals and other factors. Instead, we encourage to carefully analyze the real constraints and goals, and then select accordingly an appropriate small set of concepts and skills we expect our students should acquired.

**Acknowledgments.** The authors would like to thank the financial support from the University of Cantabria and the Spanish Ministry of Education and Science, National Program of Computer Science, Project Ref. #TIN2006-13615.

## References

1. Anand, V.B.: Computer Graphics and Geometric Modeling for Engineers. John Wiley and Sons, New York (1993)
2. Angel, E.: Interactive Computer Graphics, a Top-Down Approach With OpenGL. Addison Wesley, Reading MA (1997)
3. Barnsley, M.F.: Fractals Everywhere, 2nd edn. Academic Press, Boston (1993)
4. The Bologna Declaration on the European space for higher education: an explanation. Association of European Universities & EU Rectors' Conference, p. 4 (1999), (available at: <http://ec.europa.eu/education/policies/educ/bologna/bologna.pdf>)
5. Bouvier, D.J.: From pixels to scene graphs in introductory computer graphics courses. *Computers and Graphics* 26, 603–608 (2002)
6. Cunningham, S.: Re-inventing the introductory computer graphics course: providing tools for a wider audience. In: Joint Eurographics/ACM Siggraph Proceedings of the Graphics and Visualization Education, GVE 1999 (1999)
7. Cunningham, S.: Powers of 10: the case for changing the first course in computer graphics. In: Proceedings of the 31st SIGCSE Technical Symposium on Computer Science Education, Austin, TX, pp. 46–49 (2000)
8. Figueiredo, F.C., Eber, D.E., Jorge, J.A.: A refereed server for educational CG content. In: Proceedings of EUROGRAPHICS 2003 (2003)
9. Foley, J.D., van Dam, A., Feiner, S.K., Hughes, J.F.: Computer Graphics. Principles and Practice, 2nd edn. Addison-Wesley, Reading (1990)
10. Glassner, A.: An Introduction to Ray Tracing. Academic Press, San Diego (1989)
11. Joint Eurographics/ACM Siggraph Symposium on Computer Graphics and Visualization Education, GVE 1999 (Coimbra, Portugal), <http://education.siggraph.org/conferences/eurographics/gve99/reports/papers/gve-fullreport.pdf>
12. Joint Eurographics/ACM Siggraph Symposium on Computer Graphics and Visualization Education, GVE 2002 (Bristol, UK) <http://education.siggraph.org/conferences/eurographics/cge-02/report>

13. Joint Eurographics/ACM Siggraph Symposium on Computer Graphics and Visualization Education, GVE 2004 (Hangzhou, China): <http://education.siggraph.org/conferences/eurographics/cge-04Rep2004CGEworkshop.pdf>
14. 2006 SIGGRAPH knowledge base: <http://education.siggraph.org/curriculum/knowledge-base/report>
15. Hall, R.: *Illumination and Color in Computer Generated Imagery*. Springer, New York (1989)
16. Hill. *Computer Graphics Using OpenGL*. Prentice-Hall, Englewood Cliffs, NJ (2000)
17. Hitchner, L., Cunningham, S., Grissom, S., Wolfe, R.: Computer graphics: The introductory course grows up. Panel session. In: *Proceedings of the 30th SIGCSE Technical Symposium on Computer Science Education (SIGCSE 1999)*, New Orleans, LA, USA (1999)
18. Hitchner, L., Sowizral, H.: Adapting computer graphics curricula to changes in graphics. *Computers and Graphics* 24(2), 283–288 (2000)
19. Hunkins, D., Levine, D.B.: Additional rich resources for computer graphics educators. *Computers and Graphics* 26, 609–614 (2002)
20. Java3D web site: <http://www.j3d.org>
21. Luengo, F., Contreras, M., Leal, A., Iglesias, A.: Interactive 3D graphics applications embedded in web pages. In: *Proceedings of Computer Graphics, Imaging and Visualization-CGIV 2007*, pp. 434–440. IEEE Computer Society Press, Los Alamitos, California (2007)
22. OpenGL web site: <http://www.opengl.org>
23. Piegl, L., Tiller, W.: *The NURBS Book*, 2nd edn. Springer, Berlin Heidelberg (1997)
24. Roberts, E., LeBlanc, R., Shackelford, R., Denning, P.J.: Curriculum 2001: Interim Report from the ACM/IEEE-CS Task Force. In: *Proceedings of the Thirtieth SIGCSE Technical Symposium on Computer Science Education*, pp. 343–344 (1999)
25. Rogers, D.F.: *Procedural Elements for Computer Graphics*, 2nd edn. Mc Graw-Hill, New York Boston (1998)
26. Taxen, G.: Teaching computer graphics constructively. *Computers and Graphics* 28, 393–399 (2004)
27. Watt, A.: *3D Computer Graphics*. Addison Wesley, Reading (2000)
28. Wolfe, R.J.: OpenGL: Agent of change or sign of the times? In: *Computer Graphics November 1998*, pp. 29–31 (1998)
29. Wolfe, R.J.: 3D Freebies: a guide to high quality 3D software available via the Internet. In: *Computer Graphics May 1998*, pp. 30–33 (1998)
30. Wolfe, R., Bailey, M., Cunningham, S., Hitchner, L.: Going farther in less time: responding to change in the introductory graphics courses. In: *Computer Graphics Annual Conference Series, ACM SIGGRAPH (SIGGRAPH 1999)*, Los Angeles, CA (1999)
31. Wolfe, R.: Bringing the introductory computer graphics course into the 21st Century. *Computers and Graphics* 24(1), 151–155 (2000)
32. Wolfe, R.J.: *3D Graphics. A Visual Approach*. Oxford University Press, New York (2000)