

# Distance Learning in Computer Graphics

A. Chica, M. Fairén and N. Pelechano

Moving Group, Universitat Politècnica de Catalunya, Barcelona, Spain

---

## Abstract

*Computer graphic courses are pretty established in most computer graphics degrees worldwide. Distance learning has been gaining popularity as fast internet access from home has been reaching wider areas around the world. If we combine this with the fact that the popularity of video games all over the world has made the technology required for rendering real time graphics widely available and affordable, we have the perfect environment for teaching computer graphics through distance learning universities. In this paper, we explain our experience in teaching Computer Graphics basics for the last 10 years at the distance teaching university Universitat Oberta de Catalunya (UOC). By using materials that include interactive applets which allow the student to play with basic Computer Graphics concepts, we observe that students can understand the course despite the teacher not being present in the learning process.*

---

## 1. Introduction

The topic of Computer Graphics is nowadays included in almost all Computer Science degrees worldwide. The concepts in this subject go beyond programming to include complex 3D graphics that requires knowledge of geometry, trigonometry, matrix transformations, coloring and lighting to achieve photorealistic rendering. It is a very visual and interactive subject that is mostly taught in universities where students can interact in a person to person manner with the professor, and “see” the rendering results shown through in-class presentation and blackboard explanations. In this paper we focus on the challenges of teaching computer graphics at a distance learning university, where the only interaction between the professor and the students is through the internet. We have been successfully teaching an introductory course in computer graphics for the last 10 years, and so in this paper we focus on explaining our experience, students results and point out the issues we have dealt with and how we managed to overcome them.

The most important steps towards successfully teaching computer graphics remotely are having fast internet access at home (due to the large amount of data that are required for storing geometry and textures), and having state of the art graphics hardware. These two main issues are nowadays solved, since almost every house has a good ADSL connec-

tion and, given the popularity of video games, good graphics hardware is affordable by the majority of the population.

Once the technical difficulties of teaching computer graphics have been overcome, the next challenge is how to teach the students a visually rich and interactive subject. Our approach, which will be explained in detail in this paper, has focused on providing interactive material to the students. This material is based on an interactive CD book, where the students can navigate all the material with links between related parts of the book, and applets that allow the student to not only passively observe animations, but also actively interact with several parameters to study the impact of the changes in the final visualization. The university also offers dedicated technical support to help the students with all the installation problems that may appear when trying all the different software required for this course.

Furthermore we offer the students the possibility of communicating with their professor not only by email but also through forums, which are open for all students to discuss any subject.

This paper is organized as follows; first we review previous work in teaching computer graphics and distance learning. Then as a reference we detail the syllabus of the course. In section 4 we explain in some detail the materials offered to the students and how the virtual campus works. Finally we provide an evaluation of the results obtained through the

last 10 years and then include a few final projects presented by students, followed by conclusions.

## 2. Related work

In [Sch05] Schweitzer discusses the requirements of distance learning for computer graphics courses. One must ensure that students have access to the necessary software, help is available to answer questions, and feedback on performance is provided. It is also important to keep students motivated through visually engaging course material, thus keeping them involved with the course.

In [BCFH06], an international curriculum for computer graphics is defined. The resulting program respects the Bologna requirements of the European Union, while reflecting the results of computer graphics education workshops. For this reason the summary recommends the introduction of a beginning graphics course into every Bachelors computer science programme, while dividing the rest of the computer graphics curriculum into basic and advanced courses, for which the contents are specified.

Petterson et al [PJS03] introduce a distributed virtual environment (DVE) that allows the execution of a students' OpenGL code while having a shared view of the result. Computer-supported collaborative learning and chat tools complement the previous tool making collaboration easier. Furthermore XML specification of the DVE makes it possible to further customize and extend the system.

In [TRE07], Taras et al describe a multimedia online course on scientific visualization which is based on the Metacoop platform. Metacoop is a Learning Management System with the expected features for such a system as well as a range of authoring tools for the creation of learning material.

It is also important to present computer graphics in the framework of a programming oriented curriculum. In this sense Sobczyk et al [STB08] present the advantages and disadvantages of such an approach. Programming allows students to put their own knowledge to the test, while making them work harder on problems to achieve perfect visual results. As a disadvantage, fixing little errors can take hours of work from the student even if the student already understood the concepts behind the source of the problem.

Another possibility is to integrate teaching and the production of teaching tools. The Computer Graphics Virtual Textbook by Czanner et al [CFSN09] follows this path. In it, groups of students from the computer graphics course were exposed to a set of CG topics. Then each of them was asked to develop Java applets that implemented algorithms of their selected topic. This development was subjected to regular discussions and evaluations to assess work progress. The resulting applets were evaluated, the best being used to teach the corresponding topics.

Anderson and Peters [AP10] provide a compilation of middleware, useful to teach game programming and computer graphics. Use of middleware exposes students to the problem of code integration and project development, but with it comes a steeper learning curve.

## 3. Course Contents

The contents of the course were designed to introduce students to the basics of computer graphics and the OpenGL pipeline. Chapters from 1 to 9 are followed in order by the students. Chapter 10, explaining the OpenGL commands, should be followed in parallel in order to be able to start the implementation of the project during the third or fourth week. These contents are in accordance with the guidelines given by the Eurographics Educational Workshop 2006 in Vienna, which are published as a report [BCFH06].

### 3.1. Introduction

This first chapter introduces the problem of visualization that computer graphics tries to tackle. It also presents some of the areas that benefit from computer graphics, as well as explains the structure of the materials to the student.

### 3.2. Graphics output hardware devices

Before entering into the details of how color can be represented, some of the technologies that are used in graphics output hardware are presented. The basic principles of operation of monitors and printers are explained, along with some associated concepts. These concepts include the idea of what a frame buffer is, as well as the concept of video memory, and pixel coordinates inside a framebuffer.

### 3.3. Color

Color representation is presented in this chapter, following with the introduction of the basic concepts needed to understand how the computer graphics pipeline works. We introduce here the fact that light is composed of multiple frequencies and then explain how the vision of colors works for the human eye. This allows us to introduce the additive and subtractive color systems, and in turn explain the RGB and CMY(K) color representations. HSV is also added and the algorithms to convert between these three representations are detailed. Finally, the CIE color space is briefly introduced.

### 3.4. 2D and 3D geometry

This chapter gives the students the necessary geometry knowledge. We present here reference systems, points, vectors, lines and planes, both in 2D and 3D, as well as intersections and distances between a number of basic geometric elements. Geometric transformations are also detailed here (2D and 3D). In addition we explain what a normal vector

is, how to compute it for a general polygon and how to determine if a point is inside a polygon.

### 3.5. Geometric object representation

The students are given some initial code for the project they have to implement during the course. As this code contains some classes that store a boundary representation of a given object, some concepts about how to represent the geometric information of objects have to be introduced. This lecture presents the need for such representations, as well as some of the most common ways to represent objects. For the case of boundary representations, polygonal meshes are extensively detailed, while introducing Euler's equation. We also show how to create objects using the sweeping algorithm, which the students implement for the project. As a result the students gain an understanding of the operations needed to model new objects.

### 3.6. 3D visualization process

With all the previous knowledge already in place we may introduce the graphics pipeline in this chapter and the next. In particular this chapter deals with the inner workings of view volume clipping, and modelview and projection transformations. This includes information on camera models, mainly the orthographic and perspective projection cameras as specified by OpenGL calls `gluPerspective`, `glOrtho`, and `gluLookAt`. The relation between what was explained about geometric transformations (in the 2D and 3D geometry chapter) and the different coordinate systems used by OpenGL are detailed here.

### 3.7. 2D visualization process

This chapter completes the explanation of the graphics pipeline by presenting the viewport transformation, the final clipping of projected primitives and how rasterization works. The rasterization part is explained incrementally, seeing first how to rasterize a single point, then a line using the Bresenham algorithm and finally a polygon.

### 3.8. Visibility and lighting models

Some details about how to paint the scene so that all primitives are rendered properly according to their visibility are still needed. Because of that backface culling and Z-buffer techniques are presented here. Additionally lighting is explained, beginning with the different lighting models, following with a detailed presentation of empirical models (Lambert and Phong), and closing with the way OpenGL does lighting. As a complement, computation of a normal per vertex is included as this allows for smooth lighting transitions between faces.

### 3.9. Graphics input interaction

This chapter introduces the basic principles of operation, as well as their usefulness, of a set of input devices. This includes computer mice, trackballs, graphics tablets, touchscreens, scanners, and haptic devices. The problem of selection is also mentioned without going into too much detail.

### 3.10. OpenGL

This final chapter gives the students information on how to use the OpenGL library.

## 4. Course Resources

As this course was going to be offered online, the materials had to be adapted. A set of interactive materials for the course contents presented in the previous section were assembled. The resulting materials allow the students to progress through the computer graphics syllabus. Some interaction is still needed to clear any doubts that students may have. This problem is aggravated by the fact that this is a programming course and students have to face any difficulties introduced by algorithm implementation and the use of OpenGL. The virtual campus helps in this regard.

### 4.1. Course Materials

The materials of this course are offered to the students as an html book which they can easily explore in a sequential manner. As new topics are presented, they can also find links to related material in the book, or to issues explained in detail as appendixes. Figures 1 and 2 show two screenshots of the book, where we can see the initial page of the book and the page explaining the polygon rasterization algorithm. In the upper part of the page the students have the possibility of jumping to any subsection of the current chapter, changing to a different chapter, or navigating sequentially backwards or forwards from page to page.

The book not only offers illustrations showing results such as different rendering techniques, but also offers applets such as the one showed in figure 3. This example shows an applet where the students can choose two different color models in order to see the conversion of parameters between them. The user can introduce the parameters of one of the color models by moving a set of sliders, and interactively the applet shows the corresponding values for the other model. At the same time a color tag is shown between the two models.

In figure 4 we can see another applet which allows the students to learn the local lighting models: ambient, Lambert, and Phong. For each of these models, the student can interactively modify the following parameters: object material properties, ambient light, position and color properties of a given light focus, and position of the observer. As they change these values they can immediately observe in

UOC

Portada | Instal·lació | Introducció | Mòdul 1 | Mòdul 2 | Mòdul 3 | Mòdul 4 | Mòdul 5 | Mòdul 6 | Mòdul 7 | Mòdul 8 | Mòdul 9 | Mòdul 10 | Mòdul 11 | Mòdul 12 | Mòdul 13 | Mòdul 14

Informàtica Gràfica I

Objectius de l'assignatura  
Glossari  
Bibliografia

CERCADOR

© Fundació per a la Universitat Oberta de Catalunya  
XW08I05020I00771

Informàtica Gràfica I - XW08I05020I00771

[Instal·lació](#)  
[Introducció](#)

Mòduls

[Mòdul 1. Introducció a la informàtica gràfica. Aplicacions](#)  
[Mòdul 2. Dispositius de sortida gràfica](#)  
[Mòdul 3. Color](#)  
[Mòdul 4. Geometria en dues i tres dimensions](#)  
[Mòdul 5. Representació dels objectes geomètrics](#)  
[Mòdul 6. Procés de visualització 3D](#)  
[Mòdul 7. Procés de visualització 2D](#)  
[Mòdul 8. Anàlisi de la visibilitat i models d'il·luminació](#)  
[Mòdul 9. Interacció gràfica. Entrada](#)  
[Mòdul 10. Visualització més realista](#)  
[Mòdul 11. Animació](#)  
[Mòdul 12. Realitat Virtual](#)  
[Mòdul 13. OpenGL](#)  
[Mòdul 14. Projecte](#)

Figure 1: The initial page of the book.

UOC

Portada | Instal·lació | Introducció | Mòdul 1 | Mòdul 2 | Mòdul 3 | Mòdul 4 | Mòdul 5 | Mòdul 6 | Mòdul 7 | Mòdul 8 | Mòdul 9 | Mòdul 10 | Mòdul 11 | Mòdul 12 | Mòdul 13 | Mòdul 14

Informàtica Gràfica I

Objectius de l'assignatura  
Glossari  
Bibliografia

CERCADOR

© Fundació per a la Universitat Oberta de Catalunya  
XW08I05020I00771

Mòdul 7 > 2 > 2.3 > 2.3.3. Rasteritza polígon

2.1. Transformació aplicació-dispositiu  
2.2. Retalat  
2.3. Rasterització  
2.3.1. Rasteritza punt  
2.3.2. Rasteritza segment  
2.3.3. Rasteritza polígon  
2.4. Mapa sensible. Resum

Mòdul 7 > 2 > 2.3 > 2.3.3. Rasteritza polígon

**Rasterització de polígons**

Donat un polígon definit pels seus  $n$  vèrtexs en coordenades de dispositiu, ens proposem calcular el conjunt de píxels interiors al polígon i pintar-los del color definit per l'atribut corresponent.

Un dels algorismes més utilitzats és l'anomenat de *seguir del contorn*. Aquest algorisme aconsegueix omplir el polígon considerant seqüencialment les línies d'escorbatge del dispositiu. Lleugeres variants de l'algorisme permeten d'omplir el polígon amb ratlles o altres mostres. A continuació teniu l'algorisme de principi i un exemple del resultat en una pantalla de baixa resolució que mostra els píxels que s'omplirien de color.

Ymax

Ymin

Figure 2: A page of the book explaining polygon rasterization.



Figure 3: Applet to convert between different color models.

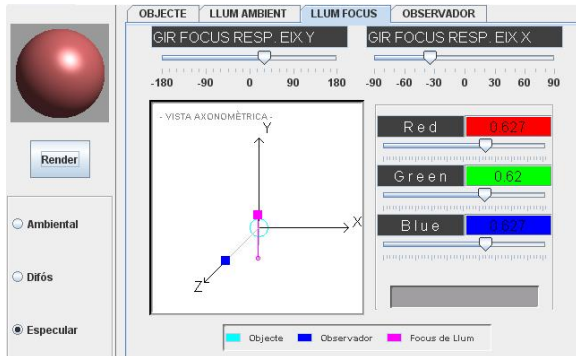


Figure 4: Applet to study the shading of a sphere under different lighting properties.

the center display the positions of observer and light focus. When all the parameters are set, the user can click on the render button to observe the effect of the selected lighting model on a smooth sphere.

The third example shown in figure 5, allows the student to observe how a set of cubes are projected onto the screen depending on the camera model selected (perspective or axonometric) and modelview parameters (VRP, observer position, and camera angles).

These applets allow the students to visualize the graphics concepts taught in the book without having to deal with programming issues. We have observed through the years of teaching graphics, that the students are more interested in learning and understand the concepts faster through the use of applets that interactively and immediately show results, rather than spending hours through trial and error in the loop of programming, compiling, and running to see the results.

#### 4.2. Virtual Campus

Besides the html book offered to the students, the university provides an online campus forum where students can log in to see the professors notifications, download homework assignments, submit their exams and programming exercises, and also communicate with the professor and other students.

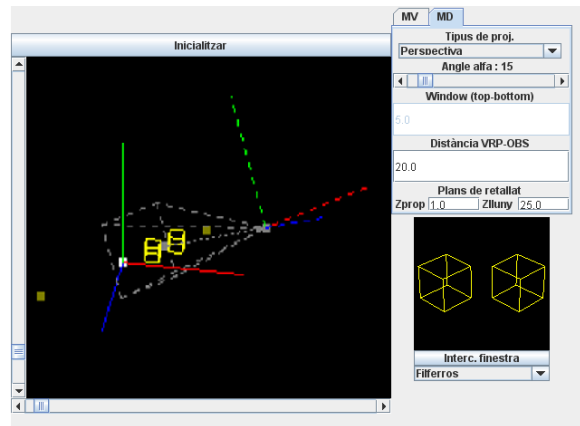


Figure 5: Applet to study the different camera parameters.

Even though email is also a suitable way of contacting the professors, we strongly encourage students to use the forum for general questions regarding their difficulties understanding the subject. We find this tool very useful for several reasons. First of all it allows students to compare themselves with the rest of the students and get an idea of their level of understanding, which is extremely useful in a virtual class where students can not work alongside each other. Secondly it allows teachers to evaluate student progress, understand the general difficulties they are having and provide the same level of help to all the students at once. Lastly, we let the students discuss their ideas and try to help each other before we provide support, since we consider that the best way for full understanding a subject is to be able to explain it to others.

#### 5. Evaluation

The hardest challenge that a professor faces when teaching a class is always to get the students motivated and working on the material throughout the semester rather than leaving all the work for the last week before the exam/deadline. In the traditional presence university, professors have the possibility of interacting face to face with the students, observing their faces during the theory classes, and asking questions to see how many people are following the lecture. All of this provides feedback which allows the professor to change the speed of teaching or level of detail depending on the degree of understanding of the students. Furthermore, during practical lessons, the professor can supervise how the students are working on their programming homework, make sure they are not getting lost or stuck on a problem and also motivate them to ask questions. Many times in class we notice that some students may be shy to ask questions but, if prompted they feel more comfortable sharing their difficulties with the professor.

This face to face interaction is obviously missing in distance learning, which means there is an even bigger chal-

lenge in interacting with the students and tracking their progress is more difficult. For this reason we need to make sure that we find homework assignments and programming projects that are appealing for the student, and also spread multiple deadlines during the semester to get them working during the whole length of the semester. The way we evaluate this course is thus oriented towards achieving this goal.

There are two quantitative measures of student performance, one to assess the knowledge of the theoretical concepts explained in the course, and the other that evaluates the required programming skills.

The theoretical component is evaluated through continuous assessment. Four tests to be completed at home, each of which consists of a set of questions covering a proportional part of the given materials, are given to the students to solve. After the students have submitted their solutions, they are graded and the correct answers and explanations are published so that students can see what the right answers were, and ask the professors if they still do not understand the problems.

For the programming component an applet project is required to be completed by the students. The students are given a skeleton only including a simple program to build an applet using JOGL. In order to motivate the students, the project description is kept very open, so they can use their imagination to implement as many features as desired as long as they complete the minimum requirements (to include interaction, camera models, local illumination, modeling a few objects, etc). Unlike traditional programming projects where a complete description of the features to be implemented is given, this kind of project allows highly motivated students to implement interesting scenes with many different applications and GUIs. Through the years we have observed that this allows students to get more interested in their work, since they have the feeling of doing something they like and not something they have been “forced” to do. Since the key to a successful project is continuous work instead of trying to do everything in a rush at the last minute and thus getting frustrated with programming and compiling errors, we give the students the opportunity to submit several milestones of their project during the semester. These partial submissions are reviewed by the professor who will then send feedback regarding their progress, problems, things that could be further improved, etc. These partial submissions are not mandatory but we encourage the students to do them, since it is the only way the professor can actually help them with their particular problems. We find many times that students may think that they have implemented something correctly when they actually have not, so through this system we can warn them before they get a (surprising) low grade.

Some example projects are shown in Figure 6. The students also have the ability to choose from a set of extended problems that gives them the opportunity to improve their final grade.

The grades of these two parts (theory and project) are combined using Table 1. Note that the programming component is given more value, as the students need at least a C- for the project in order to pass the course.

		Theoretical component				
		A	B	C+	C-	D
Project	A	A	A	B	C+	C+
	B	A	B	B	C+	C+
	C+	B	B	C+	C+	C+
	C-	C+	C+	C-	C-	C-
	D	D	D	D	D	D

**Table 1:** Grades of the theoretical part and the project are combined to obtain a final grade.

## 6. Discussion & conclusion

The results obtained over the last 10 years are summarized in tables 2 and 3. For each term we draw the percentage of students who completed all exams with their achieved grades (table 2) and the percentage of students who submitted the final project (table 3). Note that the evaluation of the course was changed in 2004-2005. In previous courses the students were encouraged to do a series of exams and the project, however none of these were compulsory as they were allowed to do a final exam that determined their grade.

Since 2004-2005 the subject has been offered only once a year and the students have to present the final project to be able to pass the course. Optionally they can undertake continuous evaluation through the intermediate exams.

In this paper we have presented our experience teaching computer graphics at the distance teaching university Universitat Oberta de Catalunya (UOC). We have presented and discussed the results obtained for the last 10 years. Our results show that those students that participate through the virtual class, work on their homework assignments and programming exercises, manage to successfully pass the course with a relatively high percentage of good scores.

The biggest problem that we have observed in this course has been the percentage of people that drop the course. We have come to the conclusion that this is mostly due to the fact that the students attending this distance university are usually older, with full time jobs, and in many cases family responsibilities, and thus they are not fully dedicated to their university studies.

It is also important to notice that this course is offered in a public university so most of the enrolment fees are funded by the Catalan government. This makes us believe that the fact that they are not fully paying for their studies, makes it easier for them to drop their courses. We believe that this problem arises in most courses and so it is not exclusively a problem in Computer Graphics. What is more, since course 2009-10

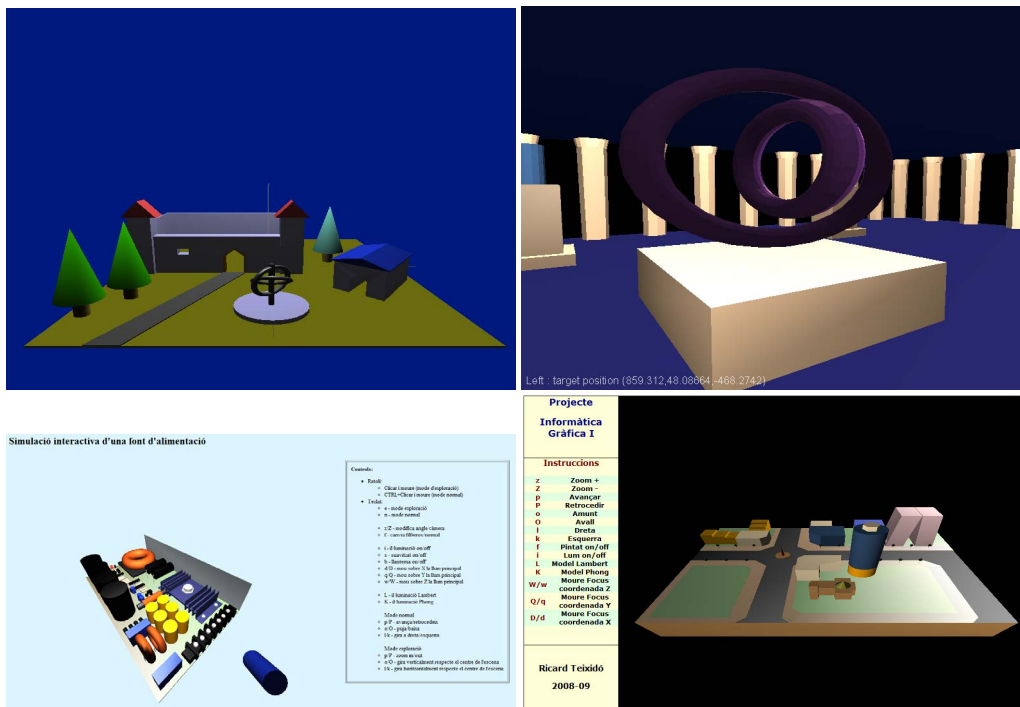


Figure 6: Screenshots of some projects presented by students.

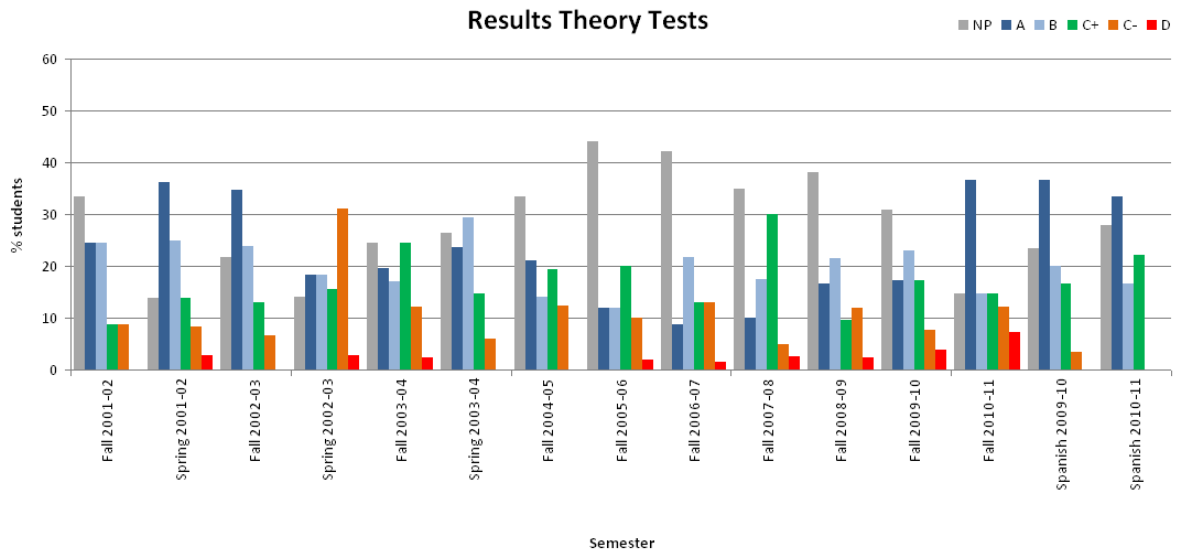
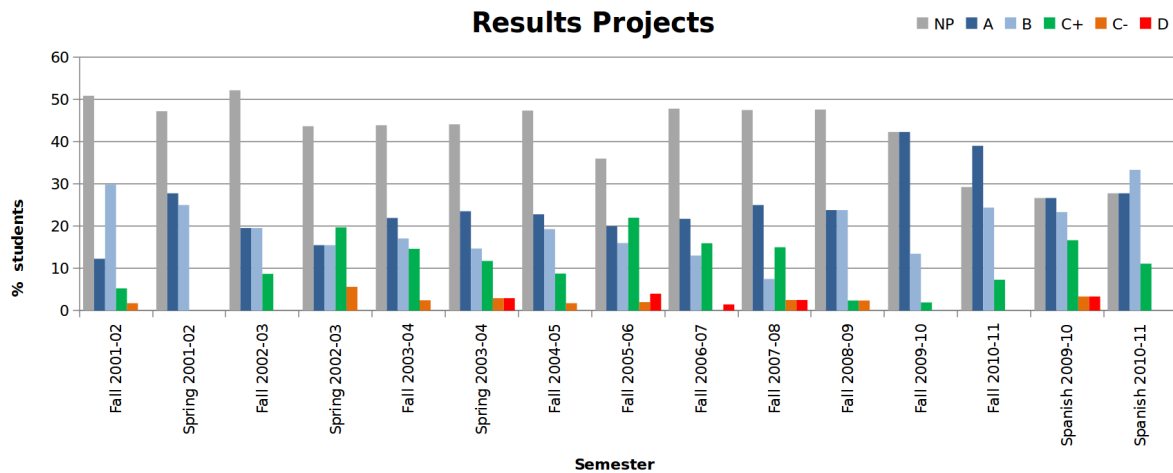


Table 2: Summary of the students results in exams for the last 10 years.

the subject is also taught in Spanish without the government funding. Even though we have only been able to gather data from two years, we can already observe in tables 2 and 3

(Spanish groups are showed at the end) a clear reduction in the number of people dropping the course.

Strategies to improve student motivation, like game-based teaching, could be applied. Such techniques use immersive



**Table 3:** Summary of the students results in projects for the last 10 years.

3D environments to allow students a better understanding of course topics. Unfortunately, these require support for additional software and we were limited by the online learning environment provided by UOC.

In order to evaluate the level of satisfaction by the students with this distance course, we have run a survey that 21 students completed. From this survey we observed that most students found useful the use of applets to study the different topics presented in the book (19% found the applets to be very useful, 38.1% useful, 33.3% neutral, 9.6% not useful). We also asked the students how helpful they found the final project to better understand the concepts, and the results again indicate that most students found it very helpful (9.5% found the applets to be very useful, 38.1% useful, 42.9% neutral, 9.5% not useful). Even though students found the interactive book useful, several students mentioned that they would also appreciate to have a printed version of the book to be able to study without needing a computer at all times. Given the high percentage of people dropping the course, we also asked the students which were the main reasons for this decision, and the results shown on the one hand lack of the mathematical background required to study computer graphics, and on the other hand the fact that a large number of students from this distance university are people that are already working full time, and thus do not have the necessary time to properly work on the assignments and project.

The conclusions given on this paper are based on the quantitative results obtained during the last ten years teaching computer graphics at a distance university, as well as the observed differences and challenges that professors have found between the distance university course, and a similar course taught at a traditional presence university. We would have liked to run a quantitative comparison between the two models of learning, but since the courses are not exactly the

same, and the background of the students is quite different, this kind of analysis would not have been fair.

## References

- [AP10] ANDERSON E. F., PETERS C. E.: No more reinventing the virtual wheel: Middleware for use in computer games and interactive computer graphics education. In *Eurographics Education Papers* (2010). 2
- [BCFH06] BOURDIN J.-J., CUNNINGHAM S., FAIREN M., HANSMANN W.: Report of the cge 06. In *Computer Graphics Education Workshop* (2006). 2
- [CFSN09] CZANNER S., FERKO A., STUGEL J., NUNUKOVA P.: Applet competition as an educational tool in creating novel e-textbook. In *Eurographics Education Papers* (2009). 2
- [PJS03] PETERSON L. W., JENSEN N., S. SEIPEL: A virtual laboratory for computer graphics education. In *Eurographics Education Papers* (2003). 2
- [Sch05] SCHWEITZER D.: Designing computer graphics courses for distance learning. In *ACM SIGGRAPH 2005 Educators program* (New York, NY, USA, 2005), SIGGRAPH '05, ACM. 2
- [STB08] SOBczyk D., TOUZEAU M.-S., BOURDIN J.-J.: Computer graphics curriculum: a programming approach. In *Eurographics Education Papers* (2008). 2
- [TRE07] TARAS C., ROTARD M., ERTL T.: An e-learning course on scientific visualization. In *Eurographics Education Papers* (2007). 2