Eurographics / ACM SIGGRAPH Workshop on Computer Graphics Education (2004), pp. 1–5
Jiaoying Shi and Cary Laxer (Editors)

An Introductory Course on Graphical-Interactive Systems: Combining Human-Computer Interaction and Computer Graphics

Martin Rotard, Daniel Weiskopf, and Thomas Ertl

Visualization and Interactive Systems Group, University of Stuttgart, Germany
{rotard, weiskopf, ertl}@vis.uni-stuttgart.de

Abstract
In this paper we report on five years of teaching experience with our introductory course "Graphical-Interactive Systems", which covers basic Human-Computer Interaction (HCI) and Computer Graphics (CG) topics for students in the third year of their major in computer science. Although the two fields of HCI and CG are usually taught in different courses, we believe that they have much in common and can therefore be integrated into a combined course. "Graphical-Interactive Systems" builds a foundation for advanced courses in both disciplines. Due to synergy effects, our approach leads to an efficient use of teaching time.

Keywords: Curriculum, Computer Graphics, Human-Computer Interaction

1. Introduction

At many universities, students are taught in the fields of Computer Graphics (CG) and Human-Computer Interaction (HCI). These two fields are usually covered in separate courses and by different textbooks, and they are taught by different professors. Analogously, research in HCI and CG is conducted by largely disjoint communities with separate scientific journals, conferences, and organizations (such as ACM SIGCHI or ACM SIGGRAPH). At first glance, both fields seem to have only a small overlap in topics. A deeper look, however, shows that they share the same history: Ivan Sutherland developed a user interface that included a number of seminal CG ideas within his "Sketchpad" system as early as 1963 [Sut63]. And we believe that CG and HCI will have a shared future in many applications like virtual reality, 3D user interfaces, mobile computing, gesture recognition, haptic interfaces, user-centered computing, etc.

In this paper we report on five years of teaching experience with our introductory course Graphical-Interactive Systems, which covers basic HCI and CG topics for students in the third year of their major in computer science or a related major. The main contribution of this paper is a curriculum that is suitable for a one-semester course that gives a combined introduction to HCI and CG. We describe the structure of the course by an exhaustive list of topics and keywords so that others can adopt our curriculum. We discuss what typical elements are left out and why, compared to introductory courses that either cover HCI or CG only. In Section 3, we report on our experiences with the course.

2. Curriculum

In this section, we describe the structure and curriculum of the course. The proposed length is one semester with three hours of lecture and one hour of exercises per week. Prerequisites are a working knowledge in elementary undergraduate mathematics (in particular, on matrix and vector math) and basic programming skills in languages like C, C++, or Java. The goal of the course is to teach all relevant elements that are required to understand and realize basic concepts of HCI and CG all the way from the device level to the application level. In addition, students are equipped with proficiency in using the APIs OpenGL [OWN*99] and Qt [BS04] in conjunction with the C++ programming language.

The course topics correspond to established curricula for HCI, as devised by ACM SIGCHI [HBC*92], and for CG, as formulated for the course Cs255 on CG within the joint Computing Curricula by the IEEE Computer Society and ACM [ICA01]. Our course focuses on ten main topics that

- Introduction and History
  - Definition and Topics of HCI and CG
  - History
- Human Cognition
  - Norman's design concepts
  - Multi-store model for perception
- Human Visual System
  - The human eye (anatomy, visual path, receptive fields)
  - Optical illusions (Mach bands, Herrmann grid)
  - Shape laws by Max Wertheimer
  - Spatial perception (binocular disparity, monocular depth cues)
  - Color perception (color antagonism, luminance, color blindness, simultaneous contrast)
- Colorimetry and Color Systems
  - Grassmann laws
  - Color matching functions
  - CIE XYZ color space
  - Device-oriented color systems (RGB, CMYK)
  - Perception-oriented color systems (HSV, HLS)
  - Perception-uniform color systems (CIE Lab, MacAdams ellipses)
- Graphical Output Devices
  - Cathode ray tube (Gamma)
  - Liquid crystal displays
  - Framebuffer (double buffering)
  - Stereo displays (active, passive, autostereoscopic)
  - Virtual reality (projectors, CAVE, powerwall, AR)
- Raster Graphics
  - Scan conversion (DDA, Bresenham, midpoint, scan conversion of polygons, scanline interpolation, active edge table, filling convex polygons)
  - Text output
  - Halftoning (ordered dithering, error diffusion)
  - Image processing (filtering, morphological operators)
  - Compositing (α-blending, Porter-Duff operators)
  - Anti-aliasing
- 2D Geometry
  - World coordinates / device coordinates
  - Affine transformations (barycentric coordinates, homogenous coordinates)
  - Line clipping (Cohen-Sutherland, parametric line clipping)
  - Polygon clipping (Sutherland-Hodgeman)
- User Interactions and Usability
  - Dialogue techniques (selection, device and dialogue level, feedback)
  - Input devices
  - Interaction styles (direct manipulation, WYSIWYG, icons)
  - Menus and dialogues
  - WIMP paradigm
  - Models of the human-machine interface (Seeheim, model-view-controller, document-view architecture, PAC, sites-modes-trails)
  - Design criteria (ISO 9241 part 10, Shneiderman's 8 golden rules, style guides)
  - Metaphors
- GUI Implementation
  - Historical overview
  - Window systems (architecture, window hierarchies / management)
  - Toolkits (widgets, widget hierarchy, X-Window, Motif/Xt, Qt, Java AWT / Swing)
  - Event handling (event loop, de-multiplexing, callbacks, virtual methods, delegation)
  - Geometry management (implicit / explicit)
  - Platform-independent GUI development
- 3D Geometry
  - Graphics pipeline
  - Affine transformation (homogeneous coordinates, transformation of normal vectors)
  - Geometric representation of objects (BReps, triangle meshes, hierarchical modeling, scene graph)
  - Perspective and orthographic projections (matrix notation, normalization transformation, classification)
  - Clipping in homogeneous coordinates
- Image Synthesis
  - Visibility (image / object space approaches, BSP, Octree, z-buffer)
  - Local illumination (Lambert, specular reflection, Phong model)
  - Overview of global illumination techniques (ray tracing, radiosity)
  - Shading (flat, Gouraud, Phong)
  - Texturing (sampled textures, procedural textures, texture coordinates, forward and backward mapping, interpolation, filtering, mipmapping)

Figure 1: *Curriculum for a course on Graphical-Interactive Systems.*

are briefly outlined in what follows. Figure 1 shows a detailed list of all topics and subtopics.

The course starts with the definition of HCI and CG and a historical overview about the developments in both fields beginning in the 1960s. The following three chapters focus on basic knowledge about human aspects that play an important role in both HCI and CG. The first topic is on human cognition oriented along Donald A. Norman's design concepts [Nor88] and the multi-store model of perception. The subsequent chapter describes the human visual system and visual perception, with a focus on color perception. This topic is followed by a discussion of colorimetry and color systems that starts with the Grassmann's laws of additive color mixture and ends with perception-uniform color systems.

The next three chapters cover fundamentals of 2D graphics and therefore lay a basis for graphical user interfaces and 3D graphics. The first topic in this teaching block deals with graphical output devices such as cathode ray tubes, liquid crystal displays, and display systems for virtual reality. Based on the concept of a framebuffer, the following chapter comprises algorithms for 2D raster graphics, e.g., for scan conversion, halftoning, image processing, compositing, and anti-aliasing. On a higher level of abstraction, affine transformations and clipping approaches are introduced as basic concepts of 2D geometry.

These fundamental topics are followed by two specific HCI chapters. The first one covers basic interaction and dialogue techniques as well as human-machine interface models and rudimentary aspects of software ergonomics in the form of design criteria for user interfaces. The second chapter of this HCI block contains technical aspects on the implementation of graphical user interfaces, window systems, event handling, and programming toolkits.

The last two chapters comprise the fundamentals of 3D graphics, which is a classical part of a CG course. In an introduction to 3D geometry, we cover affine and projective transformations, the rendering pipeline, representation of geometric objects by triangle meshes, basic scene graph structures, and clipping in homogeneous coordinates. The final chapter on rendering describes algorithms for visibility computation, local illumination models, shading, and texturing.

With the conceptual knowledge about the topics of this lecture and the practical proficiency in OpenGL and Qt from the accompanying exercises, students have a profound basis of modern CG and HCI. Specialized courses may build upon this basis, cover advanced topics in either CG or HCI, and lead to a level of education that allows students to conduct state-of-the-art research for a M.S. or Ph.D. thesis in the corresponding fields. Figure 2 depicts the advanced courses offered at the University of Stuttgart.

Due to time limitations, a combined HCI and CG course cannot allow for all topics that are contained in a lecture that just covers either HCI or CG. We choose to leave out geometric modeling (e.g. B-splines), global illumination techniques, and image processing, which are often taught in an introductory CG course. Likewise, our treatment of HCI does not take into account details of usability and toolkit programming. Our decision is based on the following considerations. First, the topics of our curriculum form a self-contained course that can stand on its own. For example, students learn enough to develop a simple, yet complete 3D OpenGL-based interactive graphics application with scene objects represented by triangle meshes, which is possible without the knowledge about advanced geometric modeling. Second, the course is efficient in the sense that the first chapters up to "Raster Graphics" serve as a common basis for HCI and CG topics, i.e., we make use of a large overlap between both fields and avoid redundancies. Third, this introductory course directly leads to advanced lectures in which the aforementioned missing topics are taught to students who specialize in these fields.

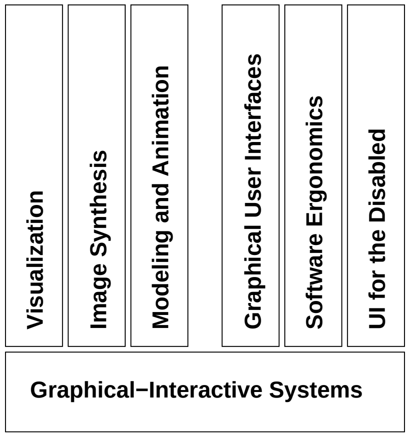

Figure 2: *Structure of the complete course program in CG and HCI at the University of Stuttgart.*

3. Experiences and Teaching Practice

We have been teaching this course on Graphical-Interactive Systems at the University of Stuttgart since 1999. The course takes place each fall semester and comprises three hours of lecture and one hour of accompanying exercises for a time span of thirteen weeks. A course is typically attended by 50–100 students. Most students have their major in computer science, software engineering, or information technology, while some students take this course as a selective subject in a related major (e.g., cybernetics or mathematics). We have experience in teaching the course in German and in English. From one year to the other, we alternate between the German and the English version.

3.1. Lecture

The lecture is mainly based on a presentation of Powerpoint slides with a video projector. We have prepared some 500 slides for the whole course, both in German and English versions. Where possible and adequate, additional teaching material and styles are employed. We make frequent use of interactive Java applets to illustrate complex concepts. Because the development of good applets is a time-consuming task, we use well-tested existing software by the University of Tübingen [HK97, HS02, HS03] and Brown University [Bro04, SS00]. Electronic films are another modern teaching element and mainly used to motivate topics of 3D graphics. Moreover, we employ a two-wall passive stereo Powerwall installed within our lecture hall to demonstrate modern virtual reality technologies during the lecture. In addition to these new media, we still rely on traditional chalk-on-blackboard techniques to develop mathematical derivations or draw 2D illustrations and sketches. This combination of different teaching styles is supported by the technical environment of our lecture hall, which allows us to simultaneously use the blackboard and a slide projection. We have good experience with using different presentation styles because they can be adapted to the different types of contents to be taught.

As a combined course on HCI and CG is not yet an established concept, there is no single textbook available for the lecture. Copies of the slides are provided as a direct source of information for the students. Standard textbooks on CG and HCI are recommended as additional reading material. References are included on the slides to point students to the corresponding chapters in the additional reading material for detailed background information.

3.2. Exercises

One exercise hour per week accompanies the lecture. Students are distributed over several exercise groups so that each group has approximately 20 students. In this way, a personal relationship between tutors and students can be established and students can be encouraged to participate actively. Homework assignments are handed out each week; handed-in assignments are checked by the tutors and returned to the students just before the exercise hour. During the exercise hour, students present their solutions in front of the class. A discussion about the results is explicitly wanted and encouraged by the tutors. We allow students to solve and present the solutions of the assignments in groups.

The exercises target different goals: They may repeat conceptual and theoretical aspects of the lecture in more detail; they may encourage students to independently apply their new knowledge to a specific problem; or they may equip students with practical programming skills. Different types of assignments are employed to achieve these different goals. Pen-on-paper assignments are used to practice mathematical transformations or to describe models of human-machine interface. Reading and web research assignments encourage students to perform independent learning. Finally, programming assignments allow students to improve their practical coding skills. We use C++ as programming language; Qt is the API for GUIs and OpenGL is the API for CG assignments. To keep the homework assignments interesting to the students, different types of assignments are combined on each homework sheet.

3.3. Resources on Course Web Page

All relevant information on our course is distributed to students via a web page. This web page contains copies of the lecture slides, homework assignments, and up-to-date announcements. In addition, an electronic forum is established to facilitate discussions between students and teachers and amongst students.

3.4. Exams

Student assessment is based on one exam at the end of this course. One option is a written exam that takes 60 minutes. As an alternative, students that take an advanced course in CG or HCI have to pass a 45 minutes oral exam that covers both the introductory and the advanced course.

3.5. Evaluation

All courses at the University of Stuttgart are regularly evaluated via a university-wide evaluation system. In five questionnaires, students are asked for their opinions with respect to the contents of the lectures and the exercises, the teaching methods, the teaching environment, and their satisfaction with the course. There is also a qualitative part in which constructive critics can be given in plain text. We have been incorporating students' feedback to improve our course. Today the course is in a stable state although we keep on including (small) changes. After the first year of teaching this course in 1999, the evaluation showed that students complained about the almost pure chalk-on-blackboard style used at that time. Therefore, we restructured the teaching material and converted most parts of the lecture to its current state with Powerpoint-based slides. Otherwise the evaluation documents that students are satisfied with both the contents and the teaching methods used in the course.

4. Conclusions

We have proposed a curriculum for an introductory course on Graphical-Interactive Systems that combines basic HCI and CG topics. Our experiences show that such a combination is feasible and, in fact, exploits much overlap between both fields to achieve time-efficient teaching. We believe that this efficiency aspect is very important and will become even more important in the future. Computer science is a rapidly growing discipline and therefore an increasing number of relevant topics have to be taught—without increasing the overall course load. Moreover, we think that HCI and CG have much in common and thus should no longer be separated in such a strict way as it is usually done at universities today. Recent developments in mixed realities, 3D user interfaces, mobile computing, and user-centric computing can only be understood with a background in both fields. The latest technical trends in GUIs support our point of view: Apple's Aqua desktop environment makes use of graphics hardware support via "Quartz Extreme" [App04]. Similarly, Microsoft's future operation system "Longhorn" builds upon "Avalon", a unified graphics representation layer, to include typical computer graphics capabilities into a mainstream API for user interfaces [Mic04].

References

[App04] APPLE: Mac OS X Panther System Architecture, 2004. `http://www.apple.com/macosx/architecture`.

[Bro04] BROWN UNIVERSITY'S COMPUTER GRAPHICS RESEARCH GROUP: Exploratories project, 2004. `http://www.cs.brown.edu/exploratory`.

[BS04] BLANCHETTE J., SUMMERFIELD M.: *C++ GUI Programming with Qt 3*. Prentice Hall, 2004.

[HBC*92] HEWETT T. T., BAECKER R. ANDCARD S., CAREY T., GASEN J., MANTEI M., PERLMAN G., STRONG G., VERPLANK W.: *ACM SIGCHI Curricula for Human-Computer Interaction*. ACM Press, New York, 1992.

[HK97] HANISCH F., KLEIN R.: Computer-Graphik spielend lernen: 70 interactive visualizations covering basics of computer graphics, 1997. `http://www.gris.uni-tuebingen.de/projects/grdev`.

[HS02] HANISCH F., STRASSER W.: Highly interactive web-based courseware. In *Proc. of CGE02 - Eurographics/SIGGRAPH Workshop on Computer Graphics Education* (2002), pp. 31–35.

[HS03] HANISCH F., STRASSER W.: Adaptability and interoperability in the field of highly interactive web-based courseware. *Computer & Graphics 27*, 4 (2003), 647–655.

[ICA01] IEEE-CS, ACM: Cs255 computer graphics. Computing Curricula 2001: Computer Science Volume. Chapter 9. Completing the Curriculum, 2001. `http://www.sigcse.org/cc2001/255.html`.

[Mic04] MICROSOFT: Longhorn Developer Center, 2004. `http://msdn.microsoft.com/longhorn`.

[Nor88] NORMAN D. A.: *The Design of Everyday Things*. Doubleday Press, New York, 1988.

[OWN*99] OPENGL ARCHITECTURE REVIEW BOARD, WOO M., NEIDER J., DAVIS T., SHREINER D.: *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2*, third ed. Addison-Wesley, Reading, MA, USA, 1999.

[SS00] SPALTER A. M., SIMPSON R. M.: Integrating interactive computer-based learning experiences into established curricula. In *Proceedings of ACM ITICSE 2000* (2000), vol. 5, pp. 116 – 119.

[Sut63] SUTHERLAND I. E.: *SketchPad: A man-machine graphical communication system*. Dissertation, MIT, Jan. 1963.