

An Introductory VR Course for Undergraduates Incorporating Foundation, Experience and Capstone.

Sharon Stansfield

Ithaca College

Ithaca, NY 14850

(607) 274-3630

sstansfield@ithaca.edu

ABSTRACT

This paper presents the structure, pedagogy and motivation for an introductory undergraduate course in Virtual Reality. The course is offered as an elective at the 400-level, hence students taking the course are juniors and seniors who have completed a substantial portion of their Computer Science curriculum. The course incorporates multiple components of VR theory and practice, including hardware and software survey and analysis, human perception, and applications. It also contains a semester-long, hands-on development component utilizing a specific virtual reality environment. In addition, because VR is a broad, multidisciplinary field of study, the course provides an ideal environment for incorporating capstone elements that allow undergraduate students to tie together many of the computing principles learned during their undergraduate academic careers.

Categories and Subject Descriptors

I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism; K.3.2 [Computer and Information Science Education]: computer science education, curriculum.

General Terms

None.

Keywords

Course design, virtual reality, capstone.

1. INTRODUCTION

In undergraduate education, it is important to take every opportunity to assist students in tying together the threads of their academic computer science experience. This is usually done in a one or two semester capstone course designed specifically for that purpose. But the incorporation of capstone elements into other upper-level computer science courses can serve to both reinforce

and broaden this experience. Because Virtual Reality (VR) is such a diverse and multidisciplinary area, an introductory course in VR lends itself naturally to the incorporation of capstone elements. In addition to the more traditional aims of a VR course, such as creating the next generation of VR researchers and end-users [4], such a course could seek to create a capstone experience that encompasses not only aspects of core CS knowledge, but also more general issues of importance to the computing professional, such as communication skills, teamwork, and application (as specified in Computing Curricula 2001 [2].) With this in mind, we have designed an introductory, undergraduate course in VR with the following goals:

1. That the student will gain an understanding of the breadth of the VR discipline, including hardware, software, human perception, and a range of applications.
2. That the student will gain an understanding of the underlying principals and design issues involved in constructing a virtual reality system/application.
3. That the student will gain hands-on experience by developing VR software utilizing an existing VR platform.
4. That the student will utilize a range of knowledge gained throughout his/her undergraduate curriculum in pursuing the above goals.
5. That the student will enhance his/her communication skills by writing and presenting a research paper.

2. OBJECTIVES AND ORGANIZATION

As indicated above, the course is structured around two synergistic aims: to introduce students, who may have a variety of interests, to the field of VR and to provide an opportunity for additional capstone experiences. Toward this end, the course has both a “traditional” lecture-based component and a hands-on experiential component. We have implemented this hands-on component as an on-going, semester-long team project. In our experience this works better than a “big bang” final project coming sometime after the bulk of the course material has been presented. Students develop each software component as the topic is presented in class. For example, the module on navigation in VR consists of a survey of hardware and software methodologies, a analysis of the trade-offs between these various devices and methods, and the incorporation of a navigation component into their VR programming project. The students choose and program the appropriate methodology for their application.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGCSE'05, February 23–27, 2005, St. Louis, Missouri, USA.

Copyright 2005 ACM 1-58113-997-7/05/0002...\$5.00.

The prerequisite for the course is our 300-level course in computer graphics. The primary reason for this prerequisite is the need for students to have a reasonable facility with the mathematics of computer graphics, such as coordinate frames, transformations and geometric object representation. While low-level graphics programming is not a component of the VR course, more advanced techniques, such as scenegraphs and stereo viewing, are presented. If desired, in depth coverage of these advanced graphics topics may be skipped – tools exist that permit one to develop virtual worlds without an in depth knowledge of the underlying mathematics and algorithms. In this case, the Computer Graphics requirement may be eliminated and additional introductory mathematics covering topics such as homogeneous transforms, coordinate systems, and geometric transformations, may be presented instead.

2.1 Objectives

After completing the course, students should be able to:

1. Explain the computational models used in VR systems (e.g. scenegraph representations, stereo viewing and collision detection.)
2. Describe the various hardware devices used in VR systems and compare and contrast the features of these devices used for similar purposes (e.g. magnetic tracking vs. optical tracking.)
3. Analyze a particular application specification and determine the best VR system configuration to meet the application's requirements.
4. Do the research and write a paper on a chosen VR topic.
5. Organize and do a presentation on this topic.
6. Program a VR environment utilizing a specific VR platform.
7. Demonstrate an ability to work as part of a programming team.

Student learning outcomes are measured through a midterm and final exam, written paper, presentation, and project demonstration.

2.2 Topics Covered

The course covers the basic hardware, software, methodologies, human factors, and applications relevant to the field and addresses the topics and learning objectives specified for the Virtual Reality elective in Computing Curricula 2001 [2].

VR topics may be organized so that they are somewhat independent of one another -- they are often organized around the various aspects of a VR system and this is the format used in our course. A survey of visual displays, for example, is preceded by a discussion of the human visual system and is followed by a discussion of the technical and human factors trade-offs involved in choosing a device. The module on interaction in VR includes a survey of tracking devices, along with their technical benefits and drawbacks and a presentation of the various models and algorithms for interaction within a VR (e.g. collision detection and voice recognition.) The course also includes assigned readings on and an open discussion of ethical issues, which often provokes the most spirited student participation of the semester.

The following is a complete list of the topics covered:

- Intro and overview of VR

- Human vision
- Visual displays
- 3D stereo viewing
- The scenegraph
- Introduction to the VR software used in class (in our case, VR_Station)
- Networked VR
- VR_station support for shared virtual environments
- Position and posture tracking
- VR_station support for tracking
- Human touch
- Haptic displays
- Interaction in virtual environments
- Navigation in virtual environments
- Human hearing
- Virtual sound
- Equilibrium and simulator sickness
- Autonomous and reactive worlds
- Ethical issues in virtual reality research and application
- Applications of virtual reality: student presentations

Presented materials may be reinforced by assigned readings from a textbook and of survey and/or research papers on the various topics. Two up-to-date and well-organized textbooks that may be used for an undergraduate course are Burdea and Coiffet [1] and Sherman and Craig [8].

In addition to the course materials, each student is expected to write a research paper on a chosen application of VR to a field outside of computer science. Topics chosen by students in the past have included such diverse applications as psychology, education, and archeology, and often reflect the student's interests in other disciplines. The assigned paper is not simply a literature review; in addition to a discussion of relevant work in the chosen area, students must compare and contrast the various methods used by individual researchers and provide an analysis of the approach and its potential. Pedagogically, the aim is to have the students research a topic as though they were preparing to do work in that same area. The last few weeks of the course are then dedicated to a "mini-conference" wherein each student prepares and presents a twenty-five minute presentation based on his/her paper. The goal in this case, of course, is to enhance technical communication skills.

3. Experiential Component

The on-going programming project is central to our VR course. Students utilize our VR laboratory to program a virtual reality environment. As topics are explored, this environment is extended to incorporate several of the major components of a VR system. Obviously, the available resources will influence what is possible – not all institutions will have the same equipment, and even a well-equipped VR research facility will likely not be available at all times to undergraduates enrolled in a VR course. Our own current model is to give the students access to the research facility, with the understanding that the research projects will get highest priority for using the equipment. However, as the popularity of the course (and hence the number of students enrolled) has increased, we have begun to reconsider this issue. Below we describe the VR platform we currently utilize for the programming project. We also consider the minimal

requirements to support the underlying pedagogy, for those who have limited access to VR facilities or who might prefer to establish a separate lab dedicated to the VR coursework

3.1 VR platform

VR Software: A reasonably fast PC with a good graphics card and a VR software platform might be considered the minimum requirements for programming Virtual Reality. We are currently using a set of software tools developed at Sandia National Laboratories for both our VR research and instructional environments. The software, which consists of a viewing component (VR_Station) and a network-based communication component (VR_Multicast), was obtained under a research licensing agreement and is fully described in [9]. This software provides support for shared virtual environments and multiple viewing methods (flat screen, head-coupled displays, etc.) It is an open platform that uses networking to support incorporation of independent modules to support such components as tracking and simulation. The course (and research) also utilizes the VR_Sim simulation engine. VR_Sim is an on-going software project to which a number of undergraduates at our institution have contributed as their senior, or capstone, project.

While we have chosen to utilize the above software in our coursework and research, a number of other VR software platforms are also available from both commercial and research organizations that could serve as the basis for the laboratory component of a VR course. Some freely available VR software systems are DIVE, developed at the Swedish Institute of Computer Science [3], Java 3D, developed at Sun Microsystems [5], NPSNET, developed at the Naval Postgraduate School [6] and VR Juggler, developed at Iowa State University [10].

Modeling Software: While a modeling package is not strictly a requirement -- one can download models, or build them before hand and make them available to the class -- it can be a valuable experience for students to build their own VE models. Geometric models form the underpinning of most VR applications and can be one of the bottlenecks in developing a VR system. Some experience with model building is invaluable to any student of VR.

Hardware: Although the popular image of VR is of someone wearing a head-mounted display (HMD), many VR researchers no longer use these devices, due to their cost, lack of resolution and field-of-view, and general discomfort. Since our research is currently utilizing HMD devices, they are available to the students in the course. After an initial demo, however, we have found that the students develop and debug their VR programs primarily using the PC monitor display -- and this is probably adequate for an introductory course. If possible, a visit to the research lab or a near-by facility for a demo of some of the more traditional viewing devices is valuable -- experiencing an HMD, CAVE, or other immersive display provides a better example of the power of immersion. A dedicated device, however, does not appear to be a requirement. The same is true for other types of displays, such as tactile and force feedback. Sound, of course, is a freebie. Arguably, the one indispensable piece of VR hardware is some type of real-time position tracker. Again, these can be expensive, although simple devices with limited range are now available which could support a VR course. The ability to track the user's position is at the heart of both presence and interaction

-- two of the three required dimensions in Zeltzer's definition of VR [11]. We are currently using MotionStar® electro-magnetic trackers from Ascension Corporation. For the VR course, we use two of these six degree-of-freedom (DOF) trackers: a tracker worn on the head (or HMD) supports updating the user's view as they move and one worn on the hand permits integration of interaction techniques (touching and pointing.) Students are given the option of using more trackers if they would like to create more complex avatars and navigation. If the PC monitor is used for display, then a single six- or even three-DOF tracker would be adequate.

3.2 Programming project

The main focus of the project is to develop a virtual world that is immersive and interactive. In institutions with VR research programs, students in a VR course are often assigned to work on some aspect of a research project. We do not follow this model, since research projects tend to be large, complex, and on-going. Students are often limited to working on some small piece of the larger project. While this is no doubt a beneficial experience, it does not provide as broad an experiential activity as does having the teams build a full-up (albeit small) VR system. The components of the on-going project are discussed below:

Modeling and model hierarchies: Working in teams, students are given the task of building a simple virtual world (e.g. modeling a room and several different associated objects.) The concepts of object complexity, texture mapping, and scenegraphs are introduced. VR_Station is scenegraph-based, and builds its internal representation from a hierarchical data file. Students create this hierarchical representation of their world using the object geometries that they previously created. Using VR_Station, they are then able to view their room in the HMD. Completing this module gives students hands-on experience with modeling and with hierarchical scene representation.

Shared virtual worlds and object behaviors: Our VR software (VR_Station / VR_Multicast / VR_Sim) supports shared VR via network communication of joint values for objects whose state may change. Each viewer loads the same world hierarchy. Simulations programmed using the VR_Sim library then create object behaviors by changing and communicating joint values to all participating viewers, which update the states of these objects. Students utilize this method to add articulated objects to their worlds: each member of the team creates an articulated object and its associated "driver" or simulation. These objects are added to the team's world model hierarchy and world engine. Team members may now independently view a shared world consisting of multiple objects with autonomous behaviors. Completing this module gives students hands-on experience with networked VR and with creating animated virtual objects using joint sets.

Interaction: In the third extension of their virtual reality projects, student teams add interaction to their worlds: The behaviors developed for the second project are modified so that they are only invoked when a user interacts with the object. The project is divided into two components: teams create tracker-driven "avatars" that will represent users within the VR. They then develop the code to determine when an avatar has selected an object. Avatars may be driven using any number of trackers worn on the user's body. A single tracker on the hand or on a wand would represent the minimum configuration. Students may make

their avatars as simple or complex as they like in terms of geometry, but each of the avatar parts must move correctly in relationship to both the data input from the trackers and the representation (e.g. head and hands must move with the body as well as independently.) Interaction may consist of pointing or touching. For either method, students must program a simple collision detection method for determining when an object has been selected.

Navigation: The final component of the project is to add a navigation technique to the virtual reality. Teams choose and implement a navigation method using either a position sensor or mouse as input. For simplicity, navigation is implemented by manipulating the model, rather than view (which is controlled by the VR viewing software.)

By the end of the course, students will have built a shared, inhabited VR with behavior and interaction – the foundation of any VR application.

4. CAPSTONE ELEMENTS

The definition of a capstone course varies from institution to institution, however, all definitions have the same fundamental elements. The following definition is taken from [7]:

“Capstone courses generally target undergraduate students who are nearing completion of their studies. They are designed to build on skills acquired in earlier courses and emphasize situations and challenges that exist in the "real world." Specific learning goals and course objectives vary across disciplines and institutions but most capstone courses provide an opportunity for students to demonstrate a range of professional competencies and communication skills.” (p. 1)

In general, capstone courses are not advanced courses on a particular topic, such as VR. Most programs, including ours, have a specific one or two semester course dedicated to the capstone experience. However, the incorporation of capstone elements into other upper-level computer science courses serves to both reinforce and broaden this experience. A VR course, such as the one presented here, provides many opportunities for students to utilize and build on previously acquired skills. The on-going, team project requires design and implementation, as well as the very “real-world” task of understanding and using code developed by others. (This is a good argument for using one of the VR platforms described above. None is a commercial product: they are the result of on-going research efforts, with all of the “real world” connotations that implies.) Specific methods and technologies from earlier courses must be applied to the design and implementation of the project, as well -- object-orientation, systems analysis and design, networked communication, and real-time data acquisition, among them. For students who may not have been previously exposed to some of these topics, the platform and project provide an opportunity to gain some experience with them in the context of an actual application. The paper and presentation not only reinforce communication skills, but also require students to understand, analyze, and critique the work of others. (A reason why we have chosen to have students write and present a topic paper, rather

than their own project work.) Project demonstrations are also scheduled throughout the semester – another all too familiar real-world task.

Of course, VR is not the only topic which lends itself to the incorporation of capstone elements: robotics and artificial intelligence are examples of others for which such a structure should work.

5. ACKNOWLEDGEMENTS

The author would like to acknowledge the support of Ithaca College and of the Mathematics and Computer Sciences Department, with special thanks to Pat Woodworth, without whose support and effort neither the course nor the lab would have been possible. We would also like to acknowledge the National Science Foundation, which supported the purchase of the VR equipment used for this course under NSF grant EIA0116295.

6. REFERENCES

- [1] Burdea, G. and Coiffet, P. *Virtual Reality Technology*. John Wiley and Sons, Inc. 2003.
- [2] Computing Curricula 2001 Final Report. IEEE Computer Society Press. 2002
- [3] The DIVE Homepage. Retrieved from <http://www.sics.se/dive> on September 1, 2004
- [4] IEEE VR Panel on Teaching VR. R. Darken, moderator, IEEE Virtual Reality Conference. March 29-31, 2004. Chicago, IL.
- [5] Java 3D API. Retrieved from <http://java.sun.com/products/java-media/3D/> on September 1, 2004
- [6] NPSNET-V. Retrieved from <http://www.npsnet.org/~npsnet/v/> on September 1, 2004.
- [7] Rhodus, T. and Hoskins, J. Toward a Philosophy for Capstone Courses in Horticulture. *Horticulture Technology*. Vol 5, No. 2, (April-June 1995) p. 175-178
- [8] Sherman, W. and Craig, A. *Understanding Virtual Reality: Interface, Application, and Design*. Morgan Kaufman Publishers. 2003.
- [9] Stansfield, S., Shawver, D., Sobel, A., Prasad, M., and Tapia, L. Design and Implementation of a Virtual Reality System and Its Application to Training Medical First Responders. *Presence: Teleoperators and Virtual Environments*. MIT Press Journals, Vol. 9, No. 6. (Dec. 2000) p. 524-556.
- [10] VR Juggler - Open Source Virtual Reality Tools. Retrieved from <http://www.vrjuggler.org> on September 1, 2004
- [11] Zeltzer, D. Autonomy, Interaction and Presence. *Presence: Teleoperators and Virtual Environments*, Vol. 1, No. 1. (1992) p. 127-132.