Three Alternatives for an Introductory Computer Graphics Sequence

John McDonald Stephen Luecking
School of Computer Science, Telecommunications and Information Systems
DePaul University
jmcdonald@cs.depaul.edu, sluecking@cs.depaul.edu

Abstract

Changes in the computer graphics marketplace over the last 10 years have made it necessary to revisit the traditional introductory computer graphics sequence offered in computer science departments. As there are now a wider range of career paths available to graduates in computer graphics, students should be offered a choice of several paths of study, ranging from technology oriented versions of the traditional design path offered by graphics design and art departments, to the computer graphics development route currently offered in computer science departments.

This paper outlines the efforts of several faculty members at DePaul University to build a degree structure that encompasses three possible routes in this spectrum. We outline the three introductory courses of study, including sample course descriptions and content, and discuss the faculty composition requirements and administrative support necessary to build such a degree.

Keywords

Graphics Education, Introductory Sequence.

1. INTRODUCTION

The last ten years has seen an incredible explosion in the field of computer graphics [Kent02] [Rhyne96]. Many industries are finding new and varied ways to use techniques of digital modeling, rendering and interaction. Among the most visible are the film and computer game industries. The computer game industry alone is expected to have a market capacity of over $10 billion during 2002 [Godinez02], while the market for computer generated feature films continues to blossom [Lasseter95].

However, the applicability of this relatively new field does not stop there. Consider that computer graphics (CG) techniques are currently being widely used in

- Industrial Design and Testing
- Scientific Simulation and Visualization
- Interactive and Web-based multimedia to support Education and Training
- Ergonomics Research
- Medical Imaging
- Research on Support for the Disabled [Wolfe99] [McDonald00]

For example, automobile (BMW, Ford) and aviation (Boeing) manufacturers routinely design and test products digitally [Potter00]. Many design houses offer a range of media including print and electronic formats, most notably in the form of web pages. This growth has created an increased demand for graphics professionals who are skilled both in art and in technology.

Even in the fine arts a rapid increase of creative investigations requiring high-end and innovative uses for technology has increased the need for technical collaborators keyed to the needs of artists [Wilson02]. The School of the Art Institute of Chicago has decided to pursue digital technology at a level deeper than commercial graphics packages. To this end our graphics students have teamed up with Art Institute students in a special workshop to apply simulation technology to creative projects.

Although job titles such as software developer place a higher priority on computing skills, such jobs in the computer graphics field also require a significant grounding in Art and Design. This fact is quite clearly demonstrated by the current trend in the CG industry to ask for a demo reel, even for software development positions. Many employers in these fields have noted that knowledge of both art and computer science is essential [Apodaca96] [Morie99].

Using digital technology as a medium for creative expression still places highest emphasis on visual and creative knowledge, but also requires technical skills as well. In DePaul University's Department of Art and Art History, for example, graphic design production is taught exclusively in computer labs. This reflects the fact that in today's market the vast majority of graphics design jobs require a technical background, including familiarity with graphics software packages, and knowledge of scripting and programming.

The fact CG professionals must be fluent in both of these fields cannot be overstated. Certainly, many technical fields require developers well grounded in mathematics and computer science, but even the most technical of these fields require developers who have well developed design skills. Consider that a common use of CG in scientific visualization is the use of color-scale images to bring forward subtle patterns in large complex datasets. This problem requires knowledge of color, human perception and formal composition, in addition to the more obvious need for knowledge of software development and data analysis techniques [Domik00] [Foley00].

On the other hand, jobs for professional designers increasingly demand fluency with digital technology. The field of graphics design is being inundated with increasingly complex software tools, whose power often lies in their programmability through an associated scripting language, such as Corel PhotoPaint™, Rhinoceros™, and 3D Studio Max™.

Thus career paths in computer graphics comprise a wide spectrum of different emphases, varying from creative designers to software developers. Therefore it is increasingly important that colleges and universities supply an environment in which interested students can gain the balance of skills required for their chosen career in this growing industry. This paper describes our experiences in creating a computer graphics curriculum to support the diverse range of paths that students in CG wish to follow.

Of primary importance in such considerations is the introductory sequence encountered by students in the first two years of their degree. Much pedagogical and technical research has gone into the development of the introductory computer graphics sequence found in many computer science departments. For more information, see [Grissom95] [McGregor86] [Brown88] and [Larondo94] among others, and the ACM/IEEE curricular guidelines [ACM01]. In addition, curricular guidelines for design education can be found in the Handbook of the National Association of Schools of Art and Design [NASAD].

More recently there have been efforts to realign this introductory sequence to keep pace with this rapidly changing field. See [Hitchner99], [Cunningham00] and [Wolfe00]. Also, there has been a great deal of effort put into building complete curricula to support degrees in the field of computer graphics [Wolfe02]. The purpose of this paper is to build on these works by considering how several distinct introductory course sequences can support the diverse range of careers in CG.

2. THREE CAREER CATEGORIES

When building our introductory course sequences, we focused on three broad categories of careers in computer graphics that cover a wide range of possibilities available to graduates today. These were

1. Digital Media Content Designers
2. Technical Directors
3. Graphics Software Developers

Each of these career paths has distinct required skill sets and required levels of technical and artistic ability. Investigating each field closely reveals clearly how they are different and where they have commonalities.

2.1 Digital Media Content Designers

These are the most pure design oriented careers in computer graphics, as practitioners work mostly with digital media in the form of web sites, electronic publications and software interface designs. Much of their work is 2D, but they increasingly use a wealth of 3D techniques as screen environments feature more VR content [Anders98].

So, in addition to the traditional techniques taught in an art or design program, students wishing to enter this career path must be familiar with current web and computer graphics technologies. Moreover, they must understand the range of specific issues related to digital design, especially in regards to user interface and website design. Websites and electronic publications are very different from more traditional paper-based documents, and while design in this area shares most of the same base issues, it has many unique aspects and guidelines of its own which must be treated appropriately.

Therefore, it is necessary that practitioners in this area be familiar with computer graphics technologies. However, programming skills are secondary for this career path, except for knowledge of interface design. Nevertheless, students wishing to pursue this field should also be conversant enough in the concepts of programming techniques to work fluidly with programmers on their teams.

2.2 Technical Directors.

Jobs in this category require a balanced mix of design and technical background, however the programming background needed centers more on scripting than on software development. Their jobs focus on the creative aspects of 3D animation production. They rely heavily on packages built by software developers, and often work closely with the developers on the specifications of such software.

While their job descriptions do not often include building software, they are expected to be able to work with highly programmable scripting systems that allow them to customize software to achieve a visual goal. Also, they need knowledge of programming in general to work more efficiently in a team with developers. For more information on the skill needs of technical directors see [Morie99].

So, like media content designers, these practitioners, do not need to know many of the more technical issues of graphics algorithms. But, more so than their designer counterparts, they are expected to be able to use scripting languages and have basic programming skills necessary to work with programmers.

Designers of visualizations will often fall into this category, depending on how intimately they must work with the construction of the actual software to perform the visualization. If a visualization designer relies mostly on commercial software to render views of a scientist's data, he or she could be considered a technical director.

2.3 Graphics Software Developers.

These are among the most technical jobs available in this field. These jobs require intimate knowledge of graphics algorithms and software development techniques. Developers are responsible for building the software that allows directors and designers to realize their creative vision. Examples include the developers behind the numerous 3D graphics packages such as Maya™, 3D Studio Max™, RenderMan™, etc.

They must be intimately familiar with the full range of mathematics supporting 3D graphics: calculus, linear algebra, splines, computational geometry, and often physics and simulation techniques. They must also have well developed software development skills including a firm understanding of object oriented design, data structures, memory management and graphics API's.

In addition, however, they must have well developed design skills to facilitate their work with the technical directors and artists on their team. Developers will be far better able to build support software for designers and directors if they have a firm understanding of the kinds of design issues their software is expected to realize.

3. THREE INTRODUCTORY SEQUENCES

The range of careers in CG has become so broad that no single introductory sequence will suffice. In our bachelor's degree programs, we have identified three paths for introductory computer graphics courses. Students can choose from one of the following three tracks based on their aspirations.

1. Human Computer Interaction 270/271: Formatting Digital Pages I/II
2. Graphics 371/339: Survey of computer graphics and Advanced Rendering Techniques
3. Graphics 329/372: Computer Graphics I and Principles of Computer Animation

Note that these sequences mimic the three career paths described above.

3.1 HCI 270/271

This sequence parallels a beginning sequence in graphic design, but with very significant differences. These differences ensue from a focus on screen presentation. By eschewing user and production issues related to print media, the courses accelerate students' entry into web and multimedia environments. Unlike typical graphic design introduction these courses also reference issues related to ECT and HCI and encourage scripting in seeking design solutions.

The first of the pair HCI 270, Digital Page Formatting I applies introductory graphic design concepts to the design format of a single onscreen page. These concepts center on the effective integration of textual and visual information and include:

- Creation of symbols and their thematic transformations
- Typographic issues treating text as visual configurations
- Synergistically relating visual decisions to textual content
- Application of communication and sign theory to page construction

The second course HCI 271, Digital Page Formatting II, attends to multi-page formats such as a web site. The primary topics relate to temporal issues in design:

- Linear and hyper-linked information sequencing
- Use of visual design to enhance receptivity to information sequences
- Theme and variation in page-to-page layout, symbols and illustrations
- Function of multimedia in page and site formatting

3.2 Graphics 371/339

Students learn state of the art 3D rendering packages and their associated scripting languages. Using languages such as the RenderMan shading language, they learn how to use existing software to produce the effects necessary to realize an artistic vision. These courses are for students wishing to work with packages to support digital artists in a collaborative environment.

The first of these courses, GPH 371 teaches the fundamental ideas of 3D computer graphics including 3D modeling, transformations and texturing. The course focuses on a conceptual approach rather than an algorithmic one. Students use a free graphics package such as POV-RAY to explore the concepts and techniques of 3D graphics.

The second course, GPH 339, is a course in Advanced Rendering Techniques, focusing primarily on procedural shading and rendering techniques. Topics include

- Texturing with images including environment mapping
- Bump and displacement mapping including fractal noise generation of surfaces
- Volumetric rendering of atmospheric effects and 3D texturing
- Procedural lighting

The advanced rendering course focuses far more on the applications of graphics oriented scripting and programming languages such as the RenderMan shader language, and prepares students for more advanced courses in the use of CG software and scripting languages.

3.3 Graphics 329/372.

In these courses students develop graphics software, and learn the implementation details of a variety of algorithms for rendering and animation. Using a high level programming language such as C++ and a graphics application-programming interface (API) such as, students in these courses develop many of the same techniques that are found in current graphics software packages. These courses are for students who wish to develop graphics software or support and extend existing software.

The first of these, GPH 329, is a course teaching the fundamentals of programming in a low level graphics API such as OpenGL. The course focuses on OpenGL's immediate mode rendering issues such as

- Working with and specifying 3D transformations
- Building a scene graph
- Rendering and shading triangle meshes, including Gouraud and Phong shading
- Texturing, environment mapping, bump-mapping, and other basic surface rendering techniques
- Simple animation

The second course GPH 372, Principles of Computer Animation, digs deeper into the mathematical algorithms of computer animation, including

- Implementation of spline based key-frame animation, including velocity control
- Rotational interpolation using quaternion splines
- Particle systems
- Flocking behavior
- Physically based modeling (mass-spring systems, ropes, fabric, etc)

These courses prepare the students to take more advanced courses in software design for graphics and animation. The Appendix contains short course syllabi/outlines for each of these course sequences.

3.4 Other Issues

Naturally, the courses such students take after their chosen two-course sequence depend a great deal on which path they take from the beginning. Students in the developer track will very often take the advanced rendering course, for example, and design students who have had some programming experience will often venture to take GPH 329 to round out their skill sets. Therefore, in addition to the introductory courses for their primary track of study, students are also strongly encouraged to take courses from the other tracks.

Our programs also recognize that there are basic skill sets required by all three tracks. As we will see, students interested in graphics development are also required to take an introductory design sequence, to build the design skills necessary to support their development efforts. Developers will most likely work on teams with designers and must at least be conversant in the language and techniques of design.

Likewise, students who take the design track must also take at least two introductory programming courses. This is so that, even if they do not have to deal with scripting on a day-to-day basis, they will be at least familiar with programming and scripting techniques, and have the tools to learn such techniques if and when the need should arise.

Computer graphics projects are built less and less by individual developers or artists, but rather by teams of people with distinct roles. Each person on the team must understand the language and capabilities of the others. For example, a designer on a team must know enough programming to know when to "call in the expert" on a problem, and reciprocally for the developer when encountering a design issue.

The developer must have empathy for the goals and visions of the designer so that they will have the proper mindset when working with the designer on realizing a desired effect. Reciprocally, the designer needs to know "what is possible" in graphics software design. Both groups need to know enough to respect the scope of the challenge that the other is facing.

4. A SUPPORTING DESIGN SEQUENCE

One common theme that permeates all of the career paths outlined above is that anyone working in the field of computer graphics must have well developed aesthetic and design skills. In fact, having a background in art is important, not only for jobs for digital effects developers, but even for those who design scientific visualizations. The purpose of a scientific visualization is often to display the evolution of a physical, biological or chemical system over time to facilitate interpretation, analysis and dissemination.

Scientific simulations must reveal subtle patterns in large data sets that can arise from a wide range of sources. Authors of scientific visualizations must therefore have enough expertise in visual design to enable them to create coherent visualizations that are engaging and communicate their intent efficiently and effectively [Domik00] [Foley00]. In fact, the goal of every visualization, whether for scientific, entertainment, advertisement or artistic purposes is to communicate. Thus it is imperative that any student of graphics studies the basic visual communication principles.

This fact became a design consideration when we built the introductory sequences for our graphics students. Every graphics student, whether designer, technical director, or developer, must complete a prerequisite sequence in the fundamentals of design. The content of these courses, outlined in the Appendix, is technology based, but nevertheless covers all of the fundamental principles of 2D and 3D design.

Despite its technology-oriented context this visual foundation sequence has remarkably similar goals to those of a foundation sequence in art: to lay down the visual basics and to instill the fundamental habits of work and thought necessary to convey visual messages effectively. In GPH 211 and 212 Principles of Digital Environments I & II, students learn the fundamental design principles for both 2D and 3D in a digital environment.

A second marked difference is the user-centered aesthetics that guides instruction. This is in contrast to the issues-oriented aesthetics under which artists push creative boundaries. While the user-centered approach chiefly acknowledges the Human Computer Interaction students in the courses, it also acknowledges the reality that technical students will most likely be working in user-targeted projects.

These courses focus on design while working with technology as a medium of communication. There are marked similarities to art and design pedagogy. Among these:

- Assignments structured as problem-based practicums addressing visual and communication principles
- Presentation lectures outlining the problem parameters, goals and principles to be applied
- One-on-one consultations as assignment progress
- In-progress and end group critiques of visual solutions

To this traditional art approach, we have added methodologies that help integrate aesthetic and technical learning. Their purpose is to imbue the time spent at the computer with the same creative engagement that traditional artists experience at the easel and to establish a strong sense of craft in the application of digital tools [McCullough98]. Some methods include:

- Principles-based tutorials designed to attach aesthetic decisions to tool selection and use.
- Re-formation of conventional art terminology into CG and HCI terminology
- Formalized approach to problem solving especially in parameterization and prototyping procedures.
- Text material more inclusive of scientific and digital approaches to form [Luecking02]

Working with current graphics software, students compose images and models, and explore ways of producing physical renderings of their work. These physical renderings include simple printouts, layered 3D sculptures and folded constructions. Students are able to execute all of the assignments at home using procedures derived from desktop publishing and desktop engineering [Luecking02a].

5. BUILDING THE NECESSARY FACULTY

One of the great problems in setting up such a program is that it does not fit into either a traditional computer science program (because of the art content) nor does it fit in a traditional art department (because of the computer science content). It is inherently an interdisciplinary degree.

We have found that such a program must be supported by a group of faculty with a wide range of backgrounds: computer science, mathematics and art, for example. Moreover these faculty members must be able to work as equal colleagues on the program without the divisions inherent in an interdepartmental degree. Here at DePaul, we have built a division consisting of seven professors, two who hold MFAs (one in sculpture and one in graphic design), two with PhDs in mathematics, two with PhDs in computer science, and one with a PhD in nuclear engineering.

Intense teamwork is a must as is a strong intuitive understanding and empathy for the course content taught by fellow faculty members. This leads to cohesiveness in the program and strengthens individual course offerings. One example of the latter is the development of GPH 212, the 3D design component of the supporting design sequence. In this case our sculptor consulted with our geometer on the content of the latter's surface graphics course. This consultation significantly affected the instruction of surface modeling in the design course. Reciprocally, our geometer conferred with the sculptor on assessing the visual qualities of shaders scripted in a rendering course. The outcome was a marked increase in quality of student work in both the design and rendering courses.

Such interdisciplinary collaboration has also positively conditioned our program's cohesion by firming up our lineage of prerequisites and ensuring continuity of content. More holistically, too, this collaboration has resulted in a myriad of small links that stitch together content between courses in unpredictable ways. This is a program where one's understanding of the calculus of spline surfaces just might improve the visual effect of one's sculpture.

6. CONCLUDING REMARKS

The growth and diversification of the computer graphics job market it has become clear that the traditional introductory sequence in computer graphics offered by computer science departments is inadequate to support the different career paths available to graduates.

The three introductory sequences described above can give a degree in computer graphics the flexibility to support three separate but highly related tracks of study. This allows students to pursue their chosen career path while giving them sufficient experience in all of the skills used in computer graphics, technological, aesthetic and mathematical, to work with other CG professionals in a collaborative environment.

7. REFERENCES

[ACM01] ACM/IEEE-CS Joint Curriculum Task Force (2001). *Computing Curricula 2001*. Available online, http://www.computer.org/education/cc2001/.

[Anders98] Anders P; Envisioning Cyberspace, Designing 3-D Electronic Spaces; McGraw-Hill, 1998

[Apodaca96] Apodaca P. Effects industry decries lack of art education. *Los Angeles Times,* p. A-1, October 13, 1996.

[Brown88] Brown J.R., Burton R.P., Cunningham S, Ohlson M, Varieties of Computer Graphics Courses in Computer Science. *Papers of the Nineteenth SIGCSE Technical Symposium on Computer Science Education, p. 313, 1988.*

[Cunningham00] Cunningham S, Powers of 10: The Case for Changing the First Course in Computer Graphics. *Proceedings of the Thirty-First SIGCSE Technical Symposium on Computer Science Education.* 46-49, 2000.

[Domik00] Domik G. Do We Need Formal Education in Visualization? *IEEE Computer Graphics and Applications, 20(4),* 16-19, 2000.

[Ebert99] Ebert D.S., Bailey D, An Interdisciplinary Approach to Teaching Computer Animation to Artists and Computer Scientists. *Proceedings of the Computer Graphics and Visualization Education Workshop.* 99-104, 1999.

[Foley00] Foley J, Getting There: The Ten Top Problems Left. *IEEE Computer Graphics and Applications 20(1)*, 66-68, 2000.

[Grissom95] Grissom S, Bresenham J, Kubitz B, Owen, G.S., Schweitzer D, Approaches to Teaching Computer Graphics. *Proceedings of the Twenty-Sixth SIGCSE Technical Symposium on Computer Science Education.* 382-383, 1995.

[Hitchner99] Hitchner L, Cunningham S, Grissom S &, Wolfe R, Computer Graphics: The Introductory Course Grows Up. *Proceedings of the thirtieth SIGCSE Technical Symposium on Computer Science Education.* 341-342, 1999.

[Kent02] Kent S, Sony Riegns over Gaming. *Chicago Tribune,* April 22, 2002, p. CN-1.

[Larrondo94] Larrondo-Petrie M.M, Bresenham J, Laxer J, Lansdown J, Owen G.S., Approches to Teaching Introductory Computer Graphics. *SIGGRAPH 94, Computer Graphics Proceedings, Annual Conference Series*, pp 479-480, 1994.

[Luecking02] Luecking S; Principles of Three-Dimensional Design: Object, Space and Meaning; Prentice-Hall, 2002

[Luecking02a] Luecking S; The Digital Instruction of Three-Dimensional Design, prepared for Foundation Art and Theory Education (FATE) in Review

[McCullough98] McCullough M; Abstracting Craft: the Practiced Digital Hand; MIT Press, 1998

[McDonald00] McDonald J, Toro J, Alkoby K, Berthiaume A, Chomwong P, Christopher J, Davidson M.J., Furst J, Konie B, Lancaster G, Lytinen S, Roychoudhuri L, Sedgwick E, Tomuro N, Wolfe R, An Improved Articulated Model of the Human Hand. *Proceedings of the 8th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media*. 2000. 306 - 313.

[McGregor86] McGregor J.D., An Introductory Course in Graphics. *Papers of the Seventeenth SIGCSE Technical Symposium on Computer Science Education,* pp 222-224, 1986.

[Miller00] Miller A.I.; Insights of Genius; MIT Press; 2000

[Moire99] Morie J.F., Training in Computer Graphics for Entertainment Production: What Future TDs Need to Know. *Computer Graphics*, 33(4), 21-23, 1999.

[NASAD] The NASAD Handbook, National Association of Schools of Art and Design, Samuel Hope, Executive Directory, Reston, VA.

[Rhyne02] Thyne T.M., Collaborative Computing and Integrated Decision Support Tools for Scientific Visualization. *SIGGRAPH 96 Course #16 Notes for Visualizing Scientific Data and Information.* August 1996.

[Wilson02] Wilson S; Information Arts: Intersections of Art, Science and Technology; MIT Press; 2002

[Wolfe99] Wolfe R, Alkoby K, Barnett J, Chomwong P, Furst J, Honda G, Lancaster G, Lavoie T, Lytinen S, McDonald J, Roychoudhuri L, Taylor C, Tomuro N, Toro J. An Interface for Transcribing American Sign Language. *SIGGRAPH 99 Sketches and Applications*, August, 1999.

[Wolfe00] Wolfe R, Bringing the Introductory Computer Graphics Course Into the 21st Century. *Computers and Graphics, 24(1),* 151-155, 2000.

[Wolfe02] Wolfe R, Luecking S, McDonald J, Furst J, Sedgwick E, Carter R, Ougouag-Tiouririne N, Curricular Considerations for Supporting Careers in Computer Graphics, To Appear.

8. APPENDIX: COURSE DESCRIPTIONS

In this appendix, we detail the course content of each of the sequences described in the body of the paper.

8.1 GPH 211 Perceptual Principles for Digital Environments I

Text: *Appearance and Reality: A visual handbook for artist, designers, and makers*, Stephen Hogbin, Cambium Press, 2000.

Description: An introduction to the visual, non-verbal principles incorporated in the effective presentation of on-screen environments. This course emphasizes the use of two-dimensional elements and their organization.

Objectives: By the end of the course you will be able to:

- Set occupational goals for yourself
- Work on projects with and within a team structure as well as work on individual projects
- Recognize visual communication concepts and theories at work in everyday life.
- Create a visual work that communicates specific design concepts.
- Use design terminology and discuss design concepts.
- Work comfortably and productively in the digital environment.

Topics:

- Visual Contrast and Design Principle and Elements
- Pictorial Composition: 2D and 3D
- Expression and Emotion: Iconic Representation and Idea
- History, Culture, and Organization of Pattern
- Image Transformation and Digital Techniques
- Idea Development and Critique Methods

Class Format:

- Critical group discussions and progress reviews of project assignments.
- Informal group discussions about design concepts based on reading and homework assignments
- Short in-studio exercises coordinated with reading and project assignments.

8.2 GPH 212 Perceptual Principles for Digital Environments II

Text: Principles of Three-Dimensional Design, Stephen Luecking, Prentice-Hall, 2002

Description: This course introduces the basic visual principles guiding communication and production decisions in three dimensional computer and physical environments. Students are led through a series of problems with both on screen and handmade components.

Topics:

Students will learn:

- basic formal principles unique to communications incorporating 3-D elements.
- to relate graphic tools and principles to corresponding concepts of visual form and aesthetics
- basic modeling of NURBS surfaces.
- to translate virtual objects into actual environments using proto-CAM techniques for constructing solid models.
- to apply visual principles introduced in GPH 211 to 3-D formations on and off screen.
- to digitally manipulate geometric primitives using transformations and Boolean edits.

Class Format: Lecture/Demo/Discussion, with in-class lab time.

8.3 HCI 270 Digital Page Formatting I

Text: *Don't Make Me Think: A common sense approach to web usability*, Steve Krug, Circle.com, 2000

User-Centered Web Design, John Cato, Addison Wesley, 2000.

Description: Problem-based applications of perceptual and communication principles to the presentation of on-line and off-screen pages. Includes experience with industry standard vector, raster and formatting software.

Objectives: By the end of the course you will be able to:

- Recognize visual communication concepts and theories at work in everyday life.
- Create a visual work which communicates specific design concepts.
- Use design terminology and discuss design concepts.
- Work comfortably and productively in the digital environment.

Topics:

- Image and Color
- The Mechanics of Page Layout and Formatting
- Elements and Principles of Visual Communication
- Problem Solving and Communicating Ideas
- The Creative Process and Tools
- Type and Layout

8.4 HCI 271 Digital Page Formatting II

Text: The Art and Science of Web Design, Veen, J. New Riders, 2001.

Description: This course prepares the learner to identify, analyze, and create applications in multi-page screen environments. Learners will explore challenges in design of multi-page interfaces and synthesize the principles of HCI with design considerations.

Topics:

- Principles of Multi-page Design
- Information Architecture Analysis
- Linear and hyper-linked information sequencing
- Use of visual design to enhance receptivity to information sequences
- Digital issues for color including the internet.
- Design for hand-held applications
- Prototype testing

8.5 GPH 371 Survey of Computer Graphics

Text: 3D Graphics: A Visual Approach, R. J. Wolfe, Oxford, 1999.

Description: Survey of Computer Graphics is designed to give the student an understanding of and experience with various topics in computer graphics. You will gain practical experience in image creation, visual analysis, model conversion, and compression utilities.

Topics:

- Basic 3D concepts
- Transformations and projections
- Setting up 3D views and cameras
- Visual effects of rendering techniques
- Texturing, Bump and Environment-Mapping
- 3D modeling
- 3D rendering
- Basic animation concepts
- Image & animation formats.

8.6 GPH 339 Advanced Rendering Techniques

Text: Advanced Renderman: Creating CGI for Motion Pictures, Apodaca & Gritz, Morgan Kaufmann.

Description: This course covers advanced rendering techniques primarily using Pixar's RenderMan standard. You will learn how to create many different types of procedural shaders and integrate them into full-scale scene or animation design in a final project for the course.

Topics:

- Object primitives and modeling in Renderman
- 2D and 3D texture mapping
- Procedural textures
- Antialiasing
- Procedural lighting
- Volumetric modeling of atmospheric effects
- Fractal noise for texture and object generation
- Using radiosity

Class Format:

- Most of the classes will be in lecture/demo format
- Each project will be followed by an in-class critique of submitted work
- In class midterm exam
- The final will consist of a comprehensive project which will use the techniques we covered in the course.

8.7 GPH 329 Computer Graphics I

Text: *Interactive Computer Graphics: A top-down approach with OpenGL*, Angel, Addison Wesley, 1997.

Description: This is an introductory course in 3D graphics software development. In this course you will learn how to build software to render 3D scenes and animation, including the use of various texturing, lighting and shading techniques using OpenGL. In this class, we assume that you are familiar with object oriented programming in C++ and the basic principles of differential calculus.

Topics:

- Basic graphics architecture.
- Coordinate systems.
- Clipping.
- Three-dimensional transformations.
- Simple visible-surface algorithms.
- Illumination models.
- Gouraud and Phong shading.
- Texture Mapping and Environment Mapping Algorithms
- OpenGL extensions and the advanced capabilities of graphics hardware

8.8 GPH 372 Principles of Computer Animation

Text: *Computer Animation: Algorithms and Techniques*, Parent, Morgan Kaufmann

Physics for Game Developers, Bourg, O'Reily

Description: This course will use C++ and Open GL to explore techniques for animating 3D scenes. We will explore the fundamentals of key-frame animation and interpolation and expand on these techniques with procedurally based techniques such as animations of deformations and physically based modeling and flocking behaviors.

Topics:

- Splines for animation including timing reparameterizations.
- Data structures for spline animation
- Quaternions and quaternion splines
- Particles systems
- Physically based modeling using interacting particles and mass-spring systems
- Collision response
- Flocking behavior
- Animating hierarchical models with forward kinematics
- Inverse kinematics and other topics covered as time allows