The Art Institute of Portland℠

CG Programming II

Syllabus

VGP 352A

Winter 2007, 4 credits
Mondays, 6:00PM – 9:45PM
Room #202

Ian Romanick

Contact Information

e-mail: ir381@fac.aii.edu

Office Hours

By appointment.

Final Exam

Monday, March 19th, 5:30PM – 7:30PM.

Required Texts

Real-Time Rendering (2nd Ed.) by Tomas Akenine-Möller and Eric Haines. ISBN 1-56881-182-9. The book also has a website, http://www.realtimerendering.com/, that includes lots of additional references and sample code.

OpenGL Shading Language by Randi J. Rost. This book is *optional*, and either the 1st or 2nd edition should be sufficient. This book also has a web site, http://3dshaders.com/, that includes example shaders and some references. There will not be any readings assigned from this book.

OpenGL Programming Guide by Martin Ecker may also be useful. ISBN 0321335732. There will not be any readings assigned from this book. An older version of the book, which should be sufficient for this course, is available for download as a PDF from http://www.opengl.org/documentation/red_book/.

Additional materials will provided on-line at the course website http://people.freedesktop.org/~idr/2007-VGP352/.

Course Description

Programming advanced lighting using OpenGL and OpenGL Shading Language (GLSL). This includes per-pixel lighting, BRDFs, and multiple shadow rendering techniques.

Course Outcomes

By the end of the course, students will be able to:

- Develop and debug vertex and fragment shaders.
- Implement advanced per-pixel lighting algorithms.

- Implement multiple shadow rendering techniques including stencil-buffer shadows and shadow maps.
- Read, understand, and make use of information in academic papers.

Prerequisite

This course is both programming and math intensive. A strong background in C or C++ programming is required. Familiarity with basic graphics programming concepts (e.g., lighting, texturing, transformations, etc.) and OpenGL programming is also required. In addition, familiarity with *basic* matrix math and trigonometry are required.

Required Materials

In addition to paper and writing utensils, each student will need a removable storage device. The storage device will be used to both bring documents and sample code home from class and bring homework completed assignments to class. The storage requirements should be minimal, so a small USB flash-drive (32MB) should be sufficient.

Grading

Each student's grade in this course will be primarily based on a total of five single-week programming assignments and one four-week programming project. The remainder of the student's grade will be based on a mid-term exam, a final exam, and an in-class presentation.

Programming assignments will be graded first and foremost on whether or not correct output is produced. The remaining points are based on the style of the program. This includes, but is not limited to, algorithm selection, code formatting, and naming conventions. A detailed rubric will be provided with each assignment.

Programming Assignments		
In-class presentation		20 pts.
Homework programming assignments		50 pts.
Term project		70 pts.
	Subtotal	140 (70%)
Tests		
Mid-term Exam		20 pts.
Final Exam		40 pts.
	Subtotal	60 (30%)
	Total	200 (100%)

Some assignments *may* carry extra-credit opportunities, but they will be infrequent.

Grading Scale

A = 93% and above
A- = 90%-92%
B+ = 87%-89%
B = 83%-86%
B- = 80%-82%
C+ = 77%-79%

C = 73%-76%
C- = 70%-72%
D+ = 67%-69%
D = 60-66%

Late Work

I do not accept late work. If you miss a deadline, you will not earn the points for that activity. There are no make-up or extra-credit opportunities.

Attendance and Participation

If you are not in class for an in-class exercise, you cannot earn those points. If you miss an entire class, you are responsible for obtaining copies of handouts and other classroom materials from your classmates.

Lab Policies

Leave food and drink outside the class. Disciplinary action will be taken toward any student found using the equipment in an inappropriate manner, taking cell phone calls or surfing the web. Disruptive, disrespectful or rude behavior will not be tolerated.

Plagiarism

Presenting the writings, images or paraphrased ideas of another as one's own, is strictly prohibited at the Art Institute of Portland. Properly documented excerpts from other's works, when they are limited to an appropriate amount of the total length of a student's paper, are permissible when used to support a researched argument.

Students with Disabilities

It is AiPD policy not to discriminate against qualified students with a documented disability in its educational programs, activities or services. If you have a disability-related need for adjustments or other accommodations in this class, contact the Disability Services Coordinator.

Amber Perrin
Disabilities Services Coordinator
The Art Institute of Portland
1122 NW Davis Street
Portland, OR 97209-2911
503-382-4836 aperrin@aii.edu

Course Calendar

Week 1 (January 8th)

- Introduction to course.
- OpenGL lighting refresher.
  - How OpenGL performs lighting calculations.
  - How OpenGL shades polygons, and what the pitfalls of this approach are.
- Phong shading and per-pixel lighting.
  - Outline features of OpenGL 1.3 that allow true Phong shading.
- Homework assignments:
  - Read:

        - Chapter 3 and the first half of chapter 4 (sections 4.1, 4.2, and 4.3). We'll discuss quaternions (section 3.3) *briefly* next week.
        - J. Kautz, W. Heidrich, and H.-P. Seidel, Real-Time Bump Map Synthesis. In *Proceedings of Graphics Hardware '01,* 2001. http://www.cs.ubc.ca/~heidrich/Papers/
        - M. Toksvig, Mipmapping Normal Maps. http://developer.nvidia.com/object/mipmapping_normal_maps.html
    - Per-pixel lighting using DOT3 texture environment. (10 points)

Week 2 (January 15th)

- Martin Luther King Day. *No class this week.*

Week 3 (January 22nd)

- Discuss last week's homework assignment.
    - What is the main performance problem with our approach?
- Introduction to the OpenGL Shading Language.
    - What fixed-function elements does GLSL replace?
    - Overview of vertex and fragment shaders.
- Homework assignments:
    - Read:
        - The first third of chapter 6 (sections 6.1, 6.2, and 6.3). You can safely skim section 6.2, but you'll probably want to read section 6.3 twice.
        - W. Heidrich and H.-P. Seidel, Realistic, Hardware-accelerated Shading and Lighting. In *SIGGRAPH '99 Conference Proceedings,* 1999. http://www.cs.ubc.ca/~heidrich/Papers/
        - M. Wloka, Fresnel Reflection. http://developer.nvidia.com/object/fresnel_wp.html
    - Re-implement last week's assignment using vertex and fragment shaders. (10 points)

Week 4 (January 29th)

- BRDFs part 1
    - Isotropic BRDFs
- Real-time generation of environment maps.
- Homework assignments:
    - Read:
        - Section 6.4.
        - M. Ashikhim, S. Premoze, P. Shirley, A Micro-facet Based BRDF Generator. In *SIGGRAPH 2000 Conference Proceedings,* 2000. http://www.cs.utah.edu/vissim/papers/facets/
        - K. Kautz, P. Vazquez, W. Heidrich, and H. Seidel, A Unified Approach to Prefiltered Environment Maps. In *Proceedings of Eurographics Rendering Workshop*. 2000. http://www.mpi-sb.mpg.de/~jnkautz/projects/unifiedenvmaps/
    - Implement metallic surfaces using BRDFs. (10 points)

Week 5 (February 5th)

- BRDFs part 2
    - Anisotropic BRDFs
- Homework assignments:

- Prepare for mid-term.
- Read:
  - Dan B Goldman, Fake Fur Rendering, In *SIGGRAPH 97 Conference Proceedings*, pp. 127-134., 1997. http://www.cs.washington.edu/homes/dgoldman/fakefur/
- Add anisotropy to last week's BRDF to give the appearance of brushed metal. (10 points)

Week 6 (February 12th)

- Mid-term. *Do not be late today!*
- Wrap up BRDFs.
- Shadows, part 1
  - The stencil buffer
  - Simple shadows using the stencil buffer
- Homework assignments:
  - Read:
    - Sections 6.12.1, 6.12.2, and 6.12.3.
    - C. Evertt and M. Kilgard, Practical and Robust Stenciled Shadow Volumes for Hardware-Accelerated Rendering, 2002. http://developer.nvidia.com/object/robust_shadow_volumes.html

Week 7 (February 19th)

- Shadows, part 2
  - Improved stencil buffer shadows
- Homework assignments:
  - Read:
    - Rob Jones, OpenGL Framebuffer Object 101, 2006. http://www.gamedev.net/reference/programming/features/fbo1/
    - Simon Green, The OpenGL Framebuffer Object Extension, Game Developer's Conference '05, 2005. http://developer.nvidia.com/object/gdc_2005_presentations.html

Week 8 (February 26th)

- Off-screen rendering
  - Framebuffer objects.
- Homework assignments:
  - Read:
    - Section 6.12.4 and 6.12.5.
    - S. Brabec, T. Annen, and H.-P. Seidel, Practical Shadow Mapping, Journal of Graphics Tools, Vol. 7, Number 4, 2003. http://www.mpi-sb.mpg.de/~tannen/

Week 9 (March 5th)

- Shadows, part 3
  - Shadow map theory
  - Depth textures
  - Implementing shadow maps using depth texture and framebuffer objects.
- Homework assignments:
  - Read:

- W. Reeves, D. Salesin, and R. Cook, Rendering Antialiased Shadows with Depth Maps, In *Proceedings of SIGGRAPH '87.* 1987. http://graphics.pixar.com/ShadowMaps/
- R. Fernando, Percentage-Closer Soft Shadows, In *Proceedings of SIGGRAPH 2005.* 2005. http://developer.nvidia.com/object/siggraph_2005_presentations.html

Week 10 (March 12th)

- Shadows, part 4
  - Percentage-closer filtering and percentage-closer soft shadows.
  - Improved robustness of shadow maps
- Homework assignments:
  - Prepare for final.

Week 11 (March 19th)

- Final exam. 5:30PM – 7:30PM. *Do not be late today!*