Computer Graphics Programming II

The Art Institute of Portland℠

Course Description

VGP352A

Winter 2008, 3 credits

Tuesday, 6:00PM - 9:45PM

Room #201

In this course students will learn and apply high-level shading language concepts using the OpenGL Shading Language. Students will gain experience working with advance lighting models and procedural texturing techniques.

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

- Develop and debug vertex and fragment shaders.
- Understand and implement advanced per-pixel lighting algorithms.
- Understand and implement procedural textures.
- Understand and implement render-to-texture type algorithms.
- Read, understand, and make use of information in academic papers.

The complete, up to date, course syllabus is also available on-line at the course website (http://people.freedesktop.org/~idr/2008Q1-VGP352/). The syllabus is available as both HTML and PDF (http://people.freedesktop.org/~idr/2008Q1-VGP352/VGP352.pdf).

Prerequisite

Successful completion of VGP351 or concent of instructor is required.

This course is both programming and math intensive. Some background in C or C++ programming is required. Familiarity with object oriented programming principles will be very helpful but is not strictly required. Working knowledge matrix math and trigonometry is required. Some background in basic calculus is helpful but not required.

Texts

Required text:

Tomas Akenine-Möller Eric Haines *Real-Time Rendering (2^{nd} Ed.)*. AK Peters, Ltd., July 2002. ISBN 1-56881-182-9.

The book also has a website (http://www.realtimerendering.com/), that includes lots of additional references and sample code.

Optional texts:

Randi J. Rost *OpenGL Shading Language (2^{nd} Ed.)*. Addison-Wesley Professional, January 25, 2006. ISBN 0321334892

Richard S. Wright Benjamin Lipchak Nicholas Haemel *OpenGL SuperBible: Comprehensive Tutorial and Reference (4th Ed.)*. Addison-Wesley Professional, June 2007. ISBN 0321498828.

Earlier editions of either book should be sufficient for this course. They are often available on eBay or Amazon at reduced prices. *There will not be any readings assigned from either book.*

OpenGL Shading Language has a website (http://3dshaders.com/), that includes example shaders and some references.

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 (256MB) 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. Each student will also be required to read an academic paper and present a summary of that paper to the class. The remainder of the student's grade will be based on bi-weekly quizes and a final exam.

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		50 pts.
	Subtotal	120 (63%)
Tests		

In-class quizzes		20 pts.
Final Exam		50 pts.
	Subtotal	70 pts. (37%)
	Total	190 pts. (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 opportunities. If you are unable to attend class on the due date for a assignment, please submit it by e-mail *before* class.

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.

AiPD Policies

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 ones own, is strictly prohibited at the Art Institute of Portland. Properly documented excerpts from others 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, 2008)

Lecture notes (GLSL Intro.pdf)

- Course road-map
- Introduction to OpenGL Shading Language
  - What fixed-function elements does GLSL replace?
  - Overview of shader syntax
  - Loading and using shaders
- Phong shading with GLSL
  - Shading in world-space
  - Shading in surface-space (aka tangent-space)
  - Bumpmapping in surface-space
- Homework assignments:
  - Read:

- *Real-Time Rendering*, section 5.7. Pay particular attention to sections 5.7.4 and 5.7.5. *Skip* the subsection titled Emboss Bump Mapping.
- Michael Toksvig *Mipmapping Normal Maps*. http://developer.nvidia.com/object/mipmapping_normal_maps.html

Week 2 (January 15th, 2008)

Lecture notes (GLSL, render-to-texture, env mapping.pdf)

- Render-to-texture techniques
  - Render to framebuffer, copy to texture
  - Framebuffer objects
- Environment mapping
  - Sphere
  - Paraboloid / dual-paraboloid
  - Cubic
  - Generating environment maps in real-time
- Homework assignments:
  - Read:
    - *Real-Time Rendering*, sections 6.1, 6.2, and 6.3. You can safely skim 6.2, but you'll probably want to read 6.3 twice.
    - Matthias Wloka *Fresnel Reflection*. http://developer.nvidia.com/object/fresnel_wp.html
    - 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
  - Programming assignment #1:
    - Environment mapped shininess in tangent-space. (http://people.freedesktop.org/~idr/2008Q1-VGP352/20080115_Assignment.pdf)
    - Base code. (http://people.freedesktop.org/~idr/2008Q1-VGP352/simple_phong.zip)

Week 3 (January 22nd, 2008)

Lecture notes (20080122 - BRDF, part 1.pdf)

- Improving the reflection model
  - Mipmapping of normal maps
  - Using environment maps as lights
  - Fresnel reflection
- BRDF Introduction
  - Bi-directional reflection distribution function
  - Common terminology and notation
- Cook-Torrance model
- Homework assignments:
  - Read:
    - Robert Cook Kenneth Torrance. 1982. *A Refectance Model for Computer Graphics*, ACM Trans. Graph. 1, 1 (Jan. 1982), pp 7-24. http://graphics.pixar.com/ReflectanceModel/

Week 4 (January 29th, 2008)

Lecture notes (20080129 - BRDFs, part 2.pdf)

- Microfacet-based BRDFs
  - Normal distribution
  - Occlusion
- Homework assignments:
  - Read:
    - Michael Ashikhmin, Simon Premoze, Pete Shirley. *A Microfacet-Based BRDF Generator*, In *SIGGRAPH 2000 Conference Proceedings*. http://www.cs.utah.edu/vissim/papers/facets/
  - Programming assignment #2: Environment map based lighting with prefiltered environment maps. (http://people.freedesktop.org/~idr/2008Q1-VGP352/20080129_Assignment.pdf)

Week 5 (February 5th, 2008)

Lecture notes (20080204 - BRDFs, part 3.pdf)

- Anisotropic BRDFs
  - What is anisotropy?
  - Real surfaces that exhibit anisotropic reflaction

- Ward BRDF
- Ashikhmin BRDF

- Homework assignments:
  - Read:
    - Bruce Walter *Notes on the Ward BRDF*, Technical Report PCG-05-06. Cornell Program of Computer Graphics. April 29, 2005. http://www.graphics.cornell.edu/pubs/2005/Wal05.html
    - Dan Goldman *Fake Fur Rendering*, In *SIGGRAPH 97 Conference Proceedings*, pp. 127-134., 1997. http://www.cs.washington.edu/homes/dgoldman/fakefur/

Week 6 (February 12th, 2008)

Lecture notes (20080212 - Fur.pdf)

- Fur and hair
  - Goldman's "Fake Fur Rendering"
  - Shells and fins
  - Banks BRDF for individual hairs
- Homework assignments:
  - Read:
    - *Real-Time Rendering*, section 7.1.
    - Jerome Lengyel, Emil Praun, Adam Finkelstein, Hugues Hoppe. *Real-Time Fur over Arbitrary Surfaces*. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics I3D '01*. ACM, New York, NY, pp 227-232. http://research.microsoft.com/~hoppe/fur.pdf
    - John Isidoro, Jason Mitchell. *User customizable real-time fur*. In *ACM SIGGRAPH 2002 Conference Abstracts and Applications*. SIGGRAPH '02. ACM, New York, NY, pp 273-273. http://ati.amd.com/developer/SIGGRAPH02/SIGGRAPH2002_Sketch-Isidoro.pdf
  - Programming assignment #3: Anisotropic BRDF (http://people.freedesktop.org/~idr/2008Q1-VGP352/20080212_Assignment.pdf)

Week 7 (February 19th, 2008)

Lecture notes (20080219 - Nonphotorealistic Rendering.pdf)

- Non-photorealistic Rendering:
  - Cel (toon) shading

    - Silhouette edge rendering
    - Technical illustration
- Homework assignments:
    - Read:
        - *Real-Time Rendering*, sections 7.1 and 7.2.
        - Adam Lake, Carl Marshall, Mark Harris, Marc Blackstein. *Stylized Rendering Techniques for Scalable Real-Time 3D Animation*. In *Proceedings of the First International Symposium on Non-photorealistic Animation and Rendering (NPAR)*, pp 13-20. June 2000. http://softwarecommunity.intel.com/articles/eng/3085.htm
        - Drew Card, Jason L. Mitchell. *Non-Photorealistic Rendering with Pixel and Vertex Shaders* in Engel Wolfgang (editor) *ShaderX*, Wordware Publishing, Inc., May 2002. http://ati.amd.com/developer/shaderx/

Week 8 (February 26th, 2008)

Lecture notes (20080226 - Procedural Textures, part 1.pdf)

- Procedural textures, part 1
    - Wang tiles
    - Noise and noise based textures
- Homework assignments:
    - Read:
        - Li-Yi Wei. 2004. *Tile-based texture mapping on graphics hardware.* In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware* (Grenoble, France, August 29 - 30, 2004). HWWS '04. ACM, New York, NY, 55-63. http://graphics.stanford.edu/papers/tile_mapping_gh2004/
    - Term project: Fur shader or toon shader (http://people.freedesktop.org/~idr/2008Q1-VGP352/20080226_Assignment.pdf)

Week 9 (March 4th, 2008)

Lecture notes (20080304 - Procedural Textures, part 2.pdf)

- Procedural textures, part 2
- Homework assignments:
    - Read:

    - Guennadi Riguer, Natalya Tatarchuk, John Isidoro. *Real-time Depth of Field Simulation*, In *ShaderX2*, Wordware Publishing, Inc., October 25, 2003. http://ati.amd.com/developer/shaderx/

Week 10 (March 11th, 2008)

Lecture notes (20080311 - Post-processing.pdf)

- Texture rectangles
- Post-processing effects:
    - Filters
    - Separable filters
    - Depth-of-field
- Homework assignments:
    - Read:
        - J. D. Mulder, R. van Liere. *Fast Perception-Based Depth of Field Rendering*, In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (Seoul, Korea, October 22 - 25, 2000). VRST '00. ACM, New York, NY, 129-133. http://homepages.cwi.nl/~mullie/Work/Pubs/publications.html

Week 11 (March 18th, 2008)

- Final exam. TBD. **Do not be late today!**