ECE4xxx - GPU Programming for Video Games (2-3-3)

Prerequisites: ECE2035 [min C] or ECE2036 [min C] or CS2110 [min C] or CS2261 [min C]
Co-requisites: None

The prerequisites ensure familiarity with C, assembly language-level concepts (registers, op codes, etc.), and object-oriented programming, as well as programming experience beyond first-year introductory course sequences. They provide natural routes into the class for Computer Engineering and Electrical Engineering majors (ECE2035 or ECE2036), Computer Science (CS2110) majors, and Computational Media majors participating in the Media or Intelligence threads (CS2261).

Instructor: Prof. Aaron Lanterman, office Van Leer W431, phone 404-385-2548, e-mail lanterma@ece.gatech.edu (please put GPU somewhere in the subject line)

Course Description: 3-D graphics pipelines. Physically-based rendering. Game engine architectures. GPU architectures. Graphics APIs. Vertex and pixel shader programming. Post-processing effects. Deferred rendering.

Textbook(s): No textbook specified.

Learning Objectives: As part of this course, students:

1. Apply mathematics to describe how surfaces and cameras respond to light
2. Write shader programs to perform geometric transformations, lighting calculations, and image processing on GPUs
3. Use real-time game engines as a host for their shader code, developing up-to-date industry-relevant experience

Learning Outcomes: Upon successful completion of this course, the student will be able to:

1. Write shader code to process 3-D geometry, calculate lighting, and postprocess images
2. Articulate the advantages of physically-based rendering
3. Summarize the operation of GPU architectures *in their native application of computer graphics*
4. Recognize the conceptual components common in most real-time game engines and graphics APIs
5. Make tradeoffs between per-pixel vs. per-vertex lighting and forward vs. deferred rending

Topical Outline:
Introduction and historical context
Classic 3-D rendering pipeline: geometry transformation, lighting, texturing
Physically-based rendering (Cook-Torrance BRDFs, linear vs. gamma space lighting)
Overview of 3-D APIs
Simulation loops and game engine components
Object-oriented and component-oriented game engines
GPU architectures and GPU assembly code
Introduction to shading languages (HLSL/Cg), vertex and pixel shaders
Per-pixel vs. per-vertex lighting
Advanced 3-D shading effects (ex: bump mapping, environment mapping)

Postprocessing effects (ex: bloom, motion blur)
Deferred rendering
Screen space techniques (ex: ambient occlusion)

Note that the course does not cover OpenCL or CUDA. OpenCL, CUDA, and general multicore programming are well covered in many other classes in ECE and CoC, whereas the use of GPU architectures *for their native application of computer graphics* is not extensively covered in many other classes, either here at Georgia Tech or other schools.

Course Materials
Primarily lecture slides and demo code. We draw material from textbooks such as:
Real-Time Rendering, Third Edition, by Tomas Akenine-Moller, Eric Hanes, and Naty Hoffman.
Mathematics for 3D *Game Programming and Computer Graphics*, Third Edition, by Eric Lengyel, 2011.
Unity Shaders and Effects Cookbook, by Kenny Lammers, 2013.
Game Engine Architecture, by Jason Gregory, 2009.
However, the above textbooks are not required and should not be ordered by the bookstore.

Workload and Grading: Grades are based on several major, intensive programming projects, as well as a few smaller "warm-up" assignments designed to get students comfortable with various toolsets and "pencil-and-paper" assignments exploring the underlying mathematical foundations of physically-based rendering. You will undertake several projects to gain **industry-relevant programming experience** using C# within the Unity engine, with emphasis on HLSL/Cg vertex and pixel shader programming. A couple of these assignments, particularly an initial one in which you will code a basic graphics pipeline *without* the aid of a 3-D API (to ensure that you understand what the GPU does for you), will be due before "drop day" to provide meaningful feedback early in the course. Each individual assignment will list its relative weight compared with the other assignments. **The use of "backfiles" – i.e. solutions from previous offerings of this class – is prohibited.**

There will be no traditional paper-and-pencil exams, either midterms or finals. This is a class about programming; my philosophy is that any time you would spend taking or studying for such exams is better spent in front of a computer actually programming!

On-Line Discussions: We will use Piazza to facilitate class discussions. I will try to check Piazza at least once a day. Please post questions about anything related to the course material, and also answer other students' questions, as long as you don't "give away the answer" or post chunks of code that are more than a few lines.

Class Website: I will set up a public website that will contain homeworks and up-to-date scheduling information; the URL for this website will be provided in class. I will also use Canvas (http://canvas.gatech.edu) for posting grades and for providing information that I do not want the public at large to access. **Providing private information from the Canvas website to other individuals, particularly students taking future offerings of this class, or uploading such information to websites or shared Google drives, etc., is strictly prohibited.**

Attendance & Absences: Students with medical or family emergencies should contact the Dean of Students. See http://catalog.gatech.edu/rules/4/ for an articulation of the Institute rules. Students with excused absences will be allowed to make up the work, normally within two days. Assignments turned in late without an excused absence *may* be accepted up to a certain time

subject to a certain penalty; the availability and parameters of such late turn-in options will be listed on each individual assignment, and may vary from assignment to assignment.

Honor Code: Georgia Tech aims to cultivate a community based on trust, academic integrity, and honor. Students are expected to act according to the highest ethical standards. This course will be conducted under the rules and guidelines of the Georgia Tech Honor Code; infractions will be reported to the Dean of Students. For information on Georgia Tech's Academic Honor Code, please visit http://www.catalog.gatech.edu/policies/honor-code/ or http://www.catalog.gatech.edu/rules/18/. The "ground rules" for each assignment, which may vary from assignment to assignment, will be given in each assignment description. Please ask for clarification if any aspects of the given "ground rules" seem unclear.

To reiterate: The use of backfiles – i.e., solutions from previous offerings of this class – is strictly prohibited. Providing private information such as homework solutions to other individuals, particularly students taking future offerings of this class, or uploading such information to websites or to shared Google drives, etc., is strictly prohibited.

Office of Disability Services: If you are a student registered with the Office of Disability Services (ODS), please make sure the appropriate forms and paperwork are completed by your instructor. We will abide by all accommodations required by ODS. If you are a student with learning needs that require special accommodation, contact the Office of Disability Services at (404) 894-2563 or http://disabilityservices.gatech.edu and http://disabilityservices.gatech.edu/content/welcome-accommodate as soon as possible, to make an appointment to discuss your special needs and to obtain an accommodations letter.

Teacher Commitment: The instructor commits to dedicating time and energy to ensure that you have a productive learning environment for this course.

Student Commitment: As the student, you agree to commit your time and energy to learn the material by completing all assignments in a timely manner, attending all class sessions, and seeking help when you require it.