"So, why do we need teachers? Well, if you think that the fundamental job of a teacher is to transmit information from their head to their students, then you're right, they are obsolete... Luckily, the fundamental job of a teacher is not to deliver information. It is to guide the social process of learning. The job of a teacher is to inspire, to challenge, to excite their students to want to learn. Yes, they also do explain and demonstrate and show things, but fundamentally that is beside the point. The most important thing a teacher does is make every student feel like they are important – to make them feel accountable for doing the work of learning." – Derek Muller of the YouTube channel Veritasium, from the video "The Most Persistent Myth"

CS/ECE 4795 Syllabus

GPU Programming for Video Games, 3 Credits

Summer 2026

Asynchronous delivery: There will be no requirement for students to electronically "attend" lectures at a particular fixed time.

Instructor Information

Instructor: Aaron Lanterman

Email: lanterma@ece.gatech.edu

General Course Information

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.

Course 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

Course 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

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.

Prerequisites

ECE2035 or ECE2036 or CS2110 or CS2261 [min C on all]

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).

Course Materials

Primarily lecture slides and demo code; no required textbook is specified. 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.
- *Game Engine Architecture*, by Jason Gregory, 2009.

  However, the above books are not required.

Grading Policy

Grades are based on several major, intensive programming projects (heavily weighted, typically out of 100 points), and small**,** simple tasks to make sure you've watched the videos (lightly weighted, typically between 2 and 4 points), which will be described in the videos (so watch the videos!).

There will be no traditional paper-and-pencil exams, or traditional exams conducted online, 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!

Letter grades will be assigned based on traditional breakdowns: 90%-100%=A; 80%-89%=B; 65%-79%=C; 64% or below=D or F.

Description of Graded Components

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. Providing work you do for this course to others, particularly uploading it to homework solution websites, is prohibited.**

If you are unable to complete a homework by the given deadline, turn in what you have done for partial credit so you can move on to the next assignment. (In the case of extreme circumstances, contact me to discuss).

Course Logistics

Hardware Requirements (IMPORTANT!)

You will need access to a computer access to a computer capable of running the Unity game engine (any decent laptop built within the past 5 years with at least 8 GB of RAM should be usable) and a reasonably reliable internet connection. **During the first week of class, make sure that the latest Long Term Support version of Unity works on your computer, and contact me if you have any issues.** (The Unity Hub allows you to install multiple versions of Unity; if you have a game project in progress using an older version of the engine you will want to keep that older version around). You will not need to have a webcam or microphone, although such affordances may be handy for meeting over Zoom if desired.

Canvas

Canvas (http://canvas.gatech.edu) is the primary means of distributing information. Homework assignments and will be posted on Canvas. The following information will also be found on Canvas as it becomes available: (1) class grades, (2) this syllabus, (3) links to lecture videos, and (4) miscellaneous handouts. **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.**

On-Line Discussions

We will Discord facilitate class discussions. I usually have Discord running whenever I am using my laptop. Please post questions about anything related to the course material, and also answer other students' **general** questions, but **don't give away answers to homework problems** or post chunks of code that are more than a few lines. However, please **don't** publicly post questions about the **quizzes**, or answer such questions if you see them. Quiz questions should only be discussed with the instructor or the TA.

Course Policies

Attendance and/or Participation

Although an asynchronous class does not require class attendance in the usual sense, you will need to be self-motivating and disciplined to avoid running behind on watching the lecture videos.

Major Emergencies

If you have some sort of major life emergency – serious illness or injury, death in the family, house burns down or is flooded, etc. – that seriously impedes your progress in the class, please let me know as soon as possible so we can work something out. You will find professors can be quite reasonable if you keep us in the loop. Please don't disappear with no warning half way through, making us think that you dropped the class, and then reappear out of nowhere the week before finals asking what you can do to make things up.

Academic Integrity

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. Review Georgia Tech's Honor Code and the student Code of Conduct.

Any student suspected of cheating or plagiarism on a quiz, exam, or assignment will be reported to the Office of Student Integrity, who will investigate the incident and identify the appropriate penalty for violations.

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.

Accommodations for Students with Disabilities

If you are a student with learning needs that require special accommodation, contact the Office of Disability Services (404-894-2563) as soon as possible to make an appointment to discuss your special needs and to obtain an accommodations letter. Please also e-mail me as soon as possible in order to set up a time to discuss your learning needs.

Student-Faculty Expectations Agreement

At Georgia Tech, we believe that it is important to strive for an atmosphere of mutual respect, acknowledgement, and responsibility between faculty members and the student body. The Student-Faculty Expectations articulate some basic expectations that you can

have of me and that I have of you. In the end, simple respect for knowledge, hard work, and cordial interactions will help build the environment we seek. Therefore, I encourage you to remain committed to the ideals of Georgia Tech while in this class.

Campus Resources for Students

Student Well-Being:

At Georgia Tech, we are concerned about your overall physical, social, and mental well-being. A comprehensive list of wellness related resources has been compiled and maintained by the Office of the Vice President for Student Engagement and Well-being (student-resource-guide (gatech.edu)