Introductory Computer Graphics
CSC 461/561, Fall 2024
3 credit hours

Description

This is an introductory computer graphics course, focusing on three-dimensional imagery.

If you are interested in careers in the film and gaming industries (which are becoming increasingly similar), or simply interested in *how* the effects wizards do what they do, you should enjoy and be well served by this course. Graphics is also used much more widely today, with visualization becoming mainstreamed in our media, and highly visual interfaces in our phones and tablets — so even if you think entertainment is a waste of time, you should find something practical here.

Student learning outcomes

By the end of the course, you should be able to:

Create computer imagery, including interactive computer graphics using APIs such as WebGL and OpenGL, shading languages like GLSL, as well as software (and recently, hardware) graphics methods such as ray tracing.

Evaluate computer imagery. When you look at computer graphics in film and computer games, you should gain an appreciation for the successful use of computer graphics technique such as texturing and lighting, as well as the failures.

Analyze computer imagery. You should be able to recognize the techniques that are used to generate these images, including rasterization, ray tracing, modeling, texturing, shading, hidden surface removal and compositing.

Course structure

Each week of this course will focus on one graphics topic, as described by the course work schedule. Before that week, you will view recorded lectures available online on the topic page, and collected in this Google Drive folder — we will not lecture during class time.

This course adopts some of the studio model. (Only *some*, because it is difficult to adopt it fully in a class of this size). Class periods will be dedicated to lecture discussion, light programming, question sets, exercises and quiz reviews. Studio classes will take place in person.

Course prerequisites

For 461, CSC 316. For 561, graduate standing or instructor permission. This course requires significant programming; do not take it if you aren't prepared for that.

Instructor

Associate Professor Benjamin A. Watson. Office EBII 2280, telephone 513 0325, email bwatson@ncsu.edu.

Instructor Tianyu Wu. Office EBII 2241, email twu23@ncsu.edu.

Prof. Watson's office hours this semester are Tuesdays, at 10:30am. He will hold office hours remotely in Google Meet. Make sure you reserve your time during office hours: if no one reserves time on a given day, I will not hold office hours.

We receive a lot of email, and cannot examine all of it promptly. Include the word "461" or "561" in the subject for a quicker reply. You might get a more rapid response on our discord.

Reading material

There are no required textbooks. We will make any required readings available to you for free and online. We will often use ACM publications, most located at the ACM digital library (NCSU students have free access on campus or off-campus with login).

Nevertheless, for those of you who like a matching textbook, you may perform many if not most of your readings in Marschner & Shirley's Fundamentals of Computer Graphics. A limited number of copies are available free online. There are also many other books very relevant to this course. Programming references in particular will be quite useful. You can find a list of those on our course wiki.

Online

Moodle with grades and work schedule: https://cgclass.csc.ncsu.edu/p/wolfware.html

Website with course content and programming assignments: https://cgclass.csc.ncsu.edu

Forum for participation and reactions: https://go.ncsu.edu/cgclassdiscord

Log for reading reactions: https://go-vxlab.csc.ncsu.edu/cgclassreadinglog

Log for online participation: https://go-vxlab.csc.ncsu.edu/cgclassparticipationlog

Wiki with readings and more: https://sites.google.com/view/cgwiki-ncsu

Drive folder with lecture recordings: https://go-vxlab.csc.ncsu.edu/cgclasslectures

Activity and grading

The proportion of the final mark associated with the different components of the course is:

	461	*561*
Programming	50%	50%
Reading reactions	(5%)	10%
Exercises	10%	5%
Questions	10%	5%
Quizzes	15%	15%
Participation	15%	15%

All coursework must be performed individually, unless specifically noted otherwise. No exceptions will be made.

Programs and standups are scheduled on specific due dates. Quizzes, readings, exercises and questions are due weekly on Saturday. Participation through most of the semester. Details on when and how to turn in all this work is available at the turn-in page. At semester

end, we will stop accepting work after the final exam period, or a week before grades are due, whichever is later. This year, that is December (*461:* 13th; *561:* 11th).

You will post your reactions and much participation on the course forum. To permit us to locate your work there promptly, we require that you give us the matching URLs in our course reading and participation logs. Any reactions or online participation not listed there will not be graded.

Programming (50%, due dates on assignments). The first programming assignments will address basic elements of computer graphics programming. The last will be creating a simple computer game. These will be significant programming projects, though not as challenging as some systems or compiler assignments. If you don't have significant programming experience, you may wish to drop this class.

You will be using Javascript and the WebGL API for our assignments, which includes the shading language GLSL. You can find additional detail on these tools on our wiki.

We will weigh the programming assignments as follows:

Program 1 (ray casting)	10%
Program 2 (webgl intro)	5%
Program 3 (rasterization)	10%
Program 4 (texturing)	10%
Program 5 (game)	15%

Programs 1 and 4 must be performed individually. Programs 2 and 3 may be performed individually or in two-person teams. Program 5 may be performed individually, or in a small group if permission is sought from and given by Professor Watson. You will turn in and demo program 5 *during the final exam period*. If you miss this period and demo it earlier, you will lose 20% of your assignment mark (about 1 semester grade step). If you do not demo program 5 at all but do turn it in, you will lose 50% of your assignment mark (2 steps). Distance students must also demo, but can do so before the final period without penalty.

Further details on these programming assignments will arrive throughout the semester. We will mark the assignments primarily based on correct functionality, on a 100-point scale. We will penalize late assignments by the function (3^*n*)%, where *n* is *class* days late. You can find a description of how to turn in assignments on the class site.

We regularly use github for effort-only, in-class programming exercises, and to distribute assignment shells and solutions. However, *do not use github for your graded programming assignments before they are due*. It is too easy to accidentally publish your work this way, abetting academic dishonesty. If we find your repo with assignment code in it before all code is turned in, you will receive 0% for that assignment.

Students in 461 will find many extra credit opportunities on each assignment. For *students in 561*, only some of that extra credit will count.

Reading reactions (461: (5%) 561: 10%, due Saturday most weeks). Readings and videos are associated with most topics, and are listed in the class wiki.

You may react to these for course credit, in the manner described for good reactions on the class website. Reactions should be turned in on the course forum on the Saturday before the first class on the related topic, as listed on the course work schedule. This includes the final week of the course. Give us the URLs of your reactions in your reading log; only reactions

listed there will be graded. You may react at most once per topic, and may not turn in any readings more than four weeks late. You may only react once to a reading. Grading of reading reactions will be simple: 10 if it is good, 5 if it is only okay or late, 0 if you didn't do it.

Students in 461 may react 5 times for extra credit, with each counting as 1%. Only the best 5 will count. *Students in 561* must react 15 times, with the best 15 counting.

Exercises (461: 10% 561: 5%, due Saturday most weeks). The topic notes page lists coding and practical exercises that we will perform throughout the semester. These exercises introduce you to the real world phenomena we model, and to the coding APIs we use. We will often do these in class, making performance simple — but they can and often will be performed out of class. Please feel free to help one another in and out of class with them. Exercises are usually distributed and collected online, so bring your laptop to class if you can. You may perform exercises before we discuss the topic without penalty, but to avoid late penalties you should complete exercises by the last week the related topic is discussed in class, as listed on the course work schedule. You may not turn in exercises that are more than four weeks late. Exercises may occur during the course's final week. We grade exercises based on effort: 10 if they are prompt, 5 if late, 0 if you don't do the exercise.

Students must turn in at least (*461:* 10; *561:* 15) exercises. If they do more, only the best (*461:* 10; *561:* 15) count.

Questions (461: 10% 561: 5%, due Saturday most weeks). The topic notes page also lists practice questions we will perform. We will often do these in class, making performance simple. These questions are low-stress learning opportunities, great for quiz preparation, graded only for effort with answers given as you perform them. Please feel free to help one another in and out of class with them. We distribute and collect questions online, so bring your laptop to class if you can. You may perform questions before we discuss the topic without penalty, but to avoid late penalty you should complete questions by the last week the related topic is discussed in class, as listed on the course work schedule. You may not turn in questions more than four weeks late. Some questions may occur during the course's final week. We will grade questions based on effort: 10 if you turned them in promptly, 5 if late, 0 if you didn't do the questions.

Students must turn in at least (*461:* 10; *561:* 15) question sets. If they do more, only the best (*461:* 10; *561:* 15) count.

Quizzes (15%, due Saturday most weeks). There will be no midterm or final exam (but *we will use the final exam period*, see programming above). Instead you will perform multiple choice quiz questions for each topic listed on the topic notes page. Quiz questions will be quite similar to the practice questions, and are also listed on the course work schedule. Quizzes are open book and take-home, and are available most of the preceding week. You may *not* help each other with quizzes. We expect that at least one quiz will occur during the course's final week. We will grade each quiz on a 10-point scale for correctness. We will discuss the quiz results in class, and email you the correct answers with your automatically generated score (make sure you log out of any google account except NCSU to see answers). If you want partial credit, please show your work in text fields, and request a partial credit regrade by emailing our TAs.

Students in 461 must perform all but 5 quizzes, *students in 561* all but 2 quizzes. If they do *n* more, the lowest *n* scores will not count.

Participation (15%, due most any time). Learning happens when you engage with the course, and part of your grade measures this participation. We will provide plenty of opportunity to participate in class, but in a class of this size and variety, we also need to offer ways of engaging online. All are equally weighted:

- *In class verbal discussion:* view lectures and examine any topic material already online before class, and discuss it verbally in class.
- *Online forum discussion:* pose questions, thoughts or results in the course forum, including discussion of course topics, programming, assignments, and more. Record your online discussion in our participation log — only what is listed there will be counted.
- *Attendance:* during class, we will take attendance. If you miss more than two classes without excuse, your grade will suffer. Distance students are excused from this requirement. You may request an excused absence online.
- *Stand-ups:* in front of the class, share your work to receive or offer help. This is typically a 5-minute walkthrough of your work, complete or incomplete. Each student must share twice this way during the semester, once on the studio work you did that day, and once on your programming work. Regular students must do this in class. Distance students (only) must make a short video and post it on our forum to our #standups channel. You can find when your standup is scheduled here for 461, and here for 561.

Attendance and standups each count for 5% of the weight for participation in this class. We will grade your remaining participation discussions on a 10-point scale, with each counting as 1%. Thus you must participate online or verbally at least 5 times in class or online throughout the semester, not counting your standups.

General. We use the standard university grading scale:

A+: 97%+	B+: 87%+	C+: 77%+	D+: 67%+	F: all others
A: 93%+	B: 83%+	C: 73%+	D: 63%+	
A-: 90%+	B-: 80%+	C-: 70%+	D-: 60%+	

Pass/fail students must earn at least 60% to pass the course, which means they must participate very meaningfully in projects. Audits must demonstrate regular attendance by earning at least 20%, which they can easily achieve via participation, questions, quizzes and a few readings.

We will hear appeals on grades for two weeks after you receive them. You should appeal to the course TA first, and if you are unsatisfied, ask him or her to forward the details to me.

Course topics

Below are the possible course lecture topics in rough order. These may be updated as new topics emerge:

Introduction. To the course and to the field of computer graphics.

Ray tracing. The light transport problem including path notation. The ray casting algorithm, ray object intersections, computing shadows, specular reflections and refractions, speed improvements including bounding hierarchies, quality improvements including

supersampling.

Rasterization. The rasterization and GPU pipelines. The basics of turning a triangle into fragments. Polygon fill, interpolation.

3D modeling and transforms. Polygons and triangles. Basic file formats. 3D math, including vectors, dot and cross products. 3D transformations, including scale, rotation, translation and matrices.

Projections and viewing. Different types of projections, specifying views, building a view transformation matrix.

Shading and illumination. Local vs. global illumination. Basic physics of reflection. The Phong illumination model. Gouraud shading, Phong shading, and programmable shading.

Texture mapping. Mapping textures to polygonal models. Interpolating the map within triangles. Determining fragment color, including filtering alternatives and mipmapping.

Hidden surface removal. Backface removal, depth buffering, and binary space partition trees.

Hierarchy and other structures. Using hierarchies for grouping, organizing, animating, rendering and replicating. Other higher level structures, including display lists, triangle strips, vertex caches and vertex arrays.

Images and display devices. Raster vs. vector displays, scanned vs. random displays, CRTs vs. LCDs. Frame buffers, stencil buffers, depth buffers, accumulation buffers. Color mapping and compression. Alpha blending and compositing.

Color models. Tristimulus theory. The CIE color space. Additive vs. subtractive spaces. Device spaces, including RGB, CMY, and HSV.

The eye and color. The retina, rods and cones, color opponency and color deficiency. Hue, saturation and lightness/brightness vs. wavelength, purity and luminance.

Aliasing. Basic terminology and concepts, including sampling and reconstruction, the Nyquist rate, aliasing, pre- and postfiltering, supersampling, filtering, support, convolution.

Virtual reality. 3D display, tracking and input devices. The history of work in this area, and its open problems.

Class evaluations

Student evaluations are now online. Please do your best to complete these. They will be available for you to complete during the last two weeks of class; you will receive an email message directing you to a website where you can login using your Unity ID and complete evaluations. All evaluations are confidential; I will never know how any one of you responded to any question, and you will never know the ratings for any particular instructors.

Academic integrity

See the university policy on academic integrity. If you cheat you will fail the course, and your case will be referred to the university Office of Student Conduct. If this is not your first offense at the university, there is a chance they will place you on suspension. If you are on a student visa, this will put your visa and academic career in jeopardy.

We regularly find copied code, and have reported dozens of students to Student Conduct.

When a student copies code, we recommend that Student Conduct's penalty is two times the assignment's weight (-200%). When a student allows their code to be copied, we recommend a zero (-100%).

I encourage you to consult with other students during programming, and use utility code you find on the net. After all, code reuse is the hallmark of good programming. Nevertheless, for instructional reasons, the core code you hand in must be your own work: e.g. in a ray tracing assignment, you cannot download or copy the central ray tracing and illumination routines, but you can download and share matrix, math, and model loading routines. To avoid any problems, please feel free to forward any code you would like to reuse to our TAs. If they approve, post the code to the forum for sharing with the entire class. In any case, you must cite the author or source of the code you reuse.

Students with disabilities

Reasonable accommodations will be made for students with verifiable disabilities. In order to take advantage of available accommodations, students must register with the Disability Resource Office at Holmes Hall, Suite 304, Campus Box 7509, 919-515-7653. For more information on NC State's policy on working with students with disabilities, please see the Academic Accommodations for Students with Disabilities Regulation (REG02.20.01).

Non-discrimination policy

NC State provides equal opportunity and affirmative action efforts, and prohibits all forms of unlawful discrimination, harassment, and retaliation ("Prohibited Conduct") that are based upon a person's race, color, religion, sex (including pregnancy), national origin, age (40 or older), disability, gender identity, genetic information, sexual orientation, or veteran status (individually and collectively, "Protected Status"). Additional information as to each Protected Status is included in NCSU REG 04.25.02 (Discrimination, Harassment and Retaliation Complaint Procedure), NC State's policies and regulations covering discrimination, harassment, and retaliation. Any person who feels that he or she has been the subject of prohibited discrimination, harassment, or retaliation should contact the Office for Equal Opportunity (OEO) at 919-515-3148.