COMP 3501

Contact

Instructor: Dr. David Mould
Email: mould@scs.carleton.ca
Office: 5346 HP
Course webpage: Brightspace

Overview

This is a course about the fundamentals of real-time computer graphics. We will use OpenGL and C++. The course provides both mathematical foundations and implementation practice. You should have some OpenGL knowledge from COMP 2501 and this course will build on that.

Modality

This course will be held in person.

Textbooks and Resources

The official textbook is *Computer Graphics Programming in OpenGL with C++* by V. Scott Gordon and John Clevenger. The textbook is "recommended", not "required"; it is a useful resource (with several practical code examples) that offers an alternative perspective on the course material. Morgan McGuire's *Graphics Codex* is an outstanding resource that will reward study; you can find it at www.graphicscodex.com.

You may also find various online tutorials and code fragments useful. You are free to make use of such material provided you credit the source. In particular, models and images found online are fair game. In past years, TurboSquid has been a source of free models for students to incorporate into their assignments and project.

We will be writing code using Visual Studio. Please ensure you have access to a computer capable of running VS; be aware that, regrettably, VS is not supported under MacOS.

Topics

The course's main topics include the following:

- Mathematical foundations: coordinate systems, vectors, matrices, quaternions
- Review of real-time rendering: the Z-buffer; pixel and vertex shaders
- Texture: texture mapping and texture synthesis
- Camera: translation, rotation, perspective, and camera control
- Illumination: the 3-term lighting model and alternatives
- Physical simulation: use of physics for animation of natural phenomena, particle systems, rigid-body motion
- Screen-space special effects

Additional topics, such as raytracing, shadows, and spline interpolation, will be undertaken as time permits.

Learning Outcomes

On successful completion of this course, students will be able to build 3D interactive applications using hardware acceleration. They will be able to deploy building blocks including pixel and vertex shaders, texture mapping, lighting calculations, and particle systems. They will have an understanding of a mathematical model of image creation and the organization of a program that uses that model to create animations.

Grading Scheme

Assignments: 30%
Midterm and in-class quizzes: 15%
Course project: 20%
Final exam: 35%

Course Project

One of the main components of the course is a large project, to be undertaken in a group of size 2 to 4. Your final submission will include your implementation, adequately documented, and a written report of approximately 10-12 pages. The project will involve creating an interactive application in which a player can navigate a detailed 3D environment; optionally, you can add conventional game elements, such as enemies to shoot or treasures to collect. We will discuss the precise requirements in class and on the LMS during the term.

Dates & Deadlines

Sept 6: Classes begin.
Oct 18: Midterm exam.
Oct 23-27: Fall break, no classes.
Nov 1: Project interim report due.
Dec 8: Final day of classes – although this is a Friday, it follows a Monday schedule, so we will meet.
Dec 8: Final project due.
Dec 22: End of exam period. (Our exam will be scheduled somewhere between Dec 10 and Dec 22 inclusive.)

Assignments

We will have several assignments throughout the term – approximately weekly, with some gaps. I expect every student to complete every assignment.

Assignment submissions are handled electronically, so assignment deadlines are firm. Do not wait until the last minute; rather, plan to submit at least 30 minutes in advance of the deadline, and ideally a day in advance.

For each assignment you will be submitting one or more files that contain source code. These files must be compressed into a "zip" file – never "rar" or other format.

Use good programming practices: thorough comments; good use of whitespace; breakdown into suitable functions; descriptive variable names. Plan your approach before writing any code (e.g., write some pseudocode, make some drawings). You may be asked to show your pseudocode to the instructor or TA before receiving help.

University Policies

Academic Accommodation

If you find yourself unable to complete an assignment by the deadline, please write to the professor to request accommodation. A typical accommodation will involve either a short extension on the assignment, if practical, or shifting the weight of the assignment to the final exam.

Please review the university's policies on academic accommodation, available here:

`https://students.carleton.ca/course-outline/`

Academic Integrity

Academic integrity is central to the University's mission. Allegations of academic dishonesty are taken seriously and in this course will be handled by the Office of the Dean of Science. A finding of academic dishonesty will result in sanctions, ranging from a grade of zero on the affected assignment or exam, to a failing grade in the course, and even suspension or expulsion from the university.

You can read about the policies of the Faculty of Science here:

`https://science.carleton.ca/academic-integrity/`

Additional information about academic integrity can be found here:

`https://carleton.ca/registrar/academic-integrity/`