COMP 4501A - Advanced Facilities for Real-Time Games
Winter 2024

Carleton University
School of Computer Science
Course Outline
Last update: December 21, 2023

Course Information

Instructor: Oliver van Kaick
Contact: Oliver.vanKaick at carleton.ca
Classroom: Please check the public class schedule
Lectures: Tuesdays and Thursdays, 11:35am – 12:55pm
Student hours: Information on student hours can be found in Brightspace
Course Website: https://brightspace.carleton.ca/d2l/home/220996

For information about Carleton's academic year, including registration and withdrawal dates, see Carleton's Academic Calendar.

Teaching Assistants

A list of teaching assistants and their contact/office hours information will be posted to Brightspace once the course starts.

Course Summary

The course covers the use of game engines for the development of computer games, and advanced techniques relevant to games, such as 3D rendering, animation, and the simulation of physics. Assignments consist of programming/game development tasks.

Course Calendar Description

A practical course on the design and implementation of modern game engines and advanced facilities provided by these engines. Such facilities include systems for rendering 3D scenes; simulating physics; playing animations; game AI; and enabling multi-player games. Students will undertake a significant game development project.
Includes: Experiential Learning Activity
Prerequisite(s): COMP 3501.
Lectures three hours a week.

Topics Covered

- Architecture of games and game engines.
- Advanced rendering techniques: deferred rendering, global illumination heuristics, illumination models, programming of surface, vertex, and fragment shaders.
- Animation: key-frame animation, mesh animation, character animation, locomotion.
- Rigid-body physics: collision detection, animation based on physical simulation.
- Introduction to soft-body physics.
- Shape modeling and acquisition.
- Networking, AI, pathfinding.

Learning Outcomes

At the end of this course, students will be able to:

- Design the software architecture for a game of reasonable complexity, using a component-based architecture model.
- Summarize the main components that typically compose a game engine, explaining how these are integrated into a coherent software architecture, and how they can be used for game development.
- Explain the principles behind common techniques used for the creation of games, such as rendering, animation, and physical simulation. This includes the mathematical concepts and algorithms related to these techniques.
- Identify the most suitable techniques that can be used to add a specific functionality or effect to a computer game.
- Implement a game of reasonable complexity in the Godot engine, using 3D graphics.

Resources

We do not have an assigned textbook, as the course draws topics from a variety of areas. The following books are useful for the main topics discussed in the course:

- **Game engines:** Jason Gregory, “Game Engine Architecture”, Second Edition, CRC Press, 2015.
- **Computer graphics:** Peter Shirley, Steve Marschner, “Fundamentals of Computer Graphics”, Third Edition, CRC Press, 2009.
- **Real-time rendering, physically-based rendering:** Tomas Akenine-Möller, Eric Haines, and Naty Hoffman, “Real-time rendering”, Third Edition, A. K. Peters, 2008.
- **Animation:** Rick Parent, “Computer Animation: Algorithms and Techniques”, Third Edition, Morgan Kaufmann, 2012.

The programming assignments and course project will be based on the Godot Game Engine (https://godotengine.org/) using GDScript (https://docs.godotengine.org/en/stable/tutorials/scripting/gdscript/gdscript_basics.html). There is a wealth of books and on-line tutorials specific to programming in Godot. I would advise to start by checking the tutorials provided in Godot’s website.

Computer Requirement

For the programming assignments, you will need a computer that has a GPU suitable to run computer games and the Godot engine. Recent Windows or Linux computers with GPU support should work fine.

Assessment Scheme

Grading scheme (the specific deadlines and descriptions can be found in Brightspace):

- Assignments + course project: 60%
- Final (take home) exam: 40%.

Note that you need to obtain a passing grade on the final to pass the course.

Late Assignment Policy

Assignment deadlines are strict. The following scheme is applied to late submissions (which includes assignments and the final course project):

- 3 hours late: no penalty
- 3 to 12 hours late: -10%
- 12 to 24 hours late: -20%
- More than one day late: assignment receives a grade of zero.

Assignment submissions are handled electronically (i.e., through Brightspace). Technical problems do not exempt you from submitting on time. So, if you wait until the last minute and then have issues with your connection, you will receive a deduction according to the scheme above. Consequently, you are advised to:

- Periodically upload you progress (e.g., upload your progress to Brightspace after each major change; we will only grade your last submission).
- Submit your final work at least one hour in advance of the due date and time.
- Store backups of your assignments in the cloud, e.g., OneDrive, Dropbox, a private GitHub repository. However, your assignment has to be submitted to Brightspace so that we have a timestamped submission. Urls to the cloud will not be accepted as assignment submission.

The assignments consist of programming/game development tasks. If a project does not run, it will receive a mark of zero. Consequently, after you upload your submission to Brightspace, you should re-download it immediately and ensure that the project runs fine.

You are expected to demonstrate good programming practices at all times and your code may be penalized if it is poorly written. You are also expected to do the necessary preparatory work (i.e., devising an algorithm) before you start coding.

Academic Integrity and core reuse

If you are unsure about the expectations regarding academic integrity (how to use and cite references, how much collaboration with classmates is appropriate), ask your instructor. Sharing assignment specifications or posting them online (to sites like Chegg, CourseHero, OneClass, etc.) is considered academic misconduct. You are never permitted to post, share, or upload course materials without explicit permission from your instructor. Academic integrity offences are reported to the office of the Dean of Science. Information, the process, and penalties for such offences can be found on the ODS webpage: https://science.carleton.ca/students/academic-integrity/.

You are free to make use of art assets found online provided that their license allows you to freely use the assets and you credit the source. Code fragments that are not of your own authorship are allowed under the following conditions: 1. The code should not be implementing the main tasks required for an assignment, but rather serve for adding additional features to the projects; 2. Provide credit to the original author of the code and make sure that you understand what the code is doing.

Use of tools based on artificial intelligence

Many of the assessed activities in this course were designed to be completed by an individual working alone. The use of artificial intelligence tools such as ChatGPT, Copilot, etc., is allowed. However, if you use any of these tools, you have to disclose this in a radme.txt file or report when submitting the activity, just as you would do when reusing code from other sources as explained in the previous section. Explain what portions of the assignment were created solely by you and what portions were created with the aid of the tool. The use of any of these tools without disclosure will be considered academic misconduct. An exception to this rule is made for automated grammar and punctuation checking tools (such as Grammarly). In addition, do not fully trust these tools and double check any generated code, as AI tools are known to produce incorrect outputs.

Undergraduate Academic Advisors

The Undergraduate Advisors for the School of Computer Science are available in Room 5302HP; or by email at scs.ug.advisor@cunet.carleton.ca. The undergraduate advisors can assist with information about prerequisites and preclusions, course substitutions/equivalencies, understanding your academic audit and the remaining requirements for graduation. The undergraduate advisors will also refer students to appropriate resources such as the Science Student Success Centre, Learning Support Services and Writing Tutorial Services.

SCS Computer Laboratory

Students taking a COMP course can access the SCS computer labs. The lab schedule and location can be found at: https://carleton.ca/scs/tech-support/computer-laboratories/. All SCS computer lab and technical support information can be found at: https://carleton.ca/scs/tech-

support/. Technical support staff may be contacted in-person or virtually, see this page for details: https://carleton.ca/scs/tech-support/contact-it-support/.

University Policies:

Academic Accommodations: Carleton is committed to providing academic accessibility for all individuals. Please review the academic accommodation available to students here: https://students.carleton.ca/course-outline/.

Student Academic Integrity Policy. Every student should be familiar with the Carleton University Student Academic Integrity policy. A student found in violation of academic integrity standards may be sanctioned with penalties which range from a reprimand to receiving a grade of F in the course, or even being suspended or expelled from the University. Examples of punishable offences include plagiarism and unauthorized collaboration. Any such reported offences will be reviewed by the office of the Dean of Science. More information on this policy may be found on the ODS Academic Integrity page: https://carleton.ca/registrar/academic-integrity/.

Plagiarism. As defined by Senate, "plagiarism is presenting, whether intentional or not, the ideas, expression of ideas or work of others as one's own". Such reported offences will be reviewed by the office of the Dean of Science. More information and standard sanction guidelines can be found here: https://science.carleton.ca/students/academic-integrity/.

Unauthorized Collaboration. Senate policy states that "to ensure fairness and equity in assessment of term work, students shall not co-operate or collaborate in the completion of an academic assignment, in whole or in part, when the instructor has indicated that the assignment is to be completed on an individual basis".