

COM 391, Computer Graphics

American University of Central Asia
Software Engineering Department

1 Course Information

Course Code

COM 391

Course ID

4954

Prerequisite

COM-223, Algorithms and Data Structures

Credits

6

Time and Place

Lecture: Tuesday 12:45–14:00, Room 410

Lab: Thursday 12:45–14:00, Room 410

Course Materials

<https://github.com/auca/com.391>

2 Contact Information

Professor

Dmitrii Toksaitov

toksaitov_d@auca.kg

Office

AUCA, room 315

Office Hours

By appointment throughout the work week (write to your professor to make an appointment for any day from Monday to Friday)

3 Course Overview

The course teaches students fundamentals of computer graphics through a process of testing a 3-D engine in a series of laboratory tasks throughout the course. Students will study on how to work with graphics accelerators with the help of the OpenGL ES API to deliver rich 3-D computer-generated images, animations, and interactive applications.

As a result, students should be able to research and analyze the functioning of a complex real-time computational system, improve their skills using programming languages for software design and development in alignment with the objectives of the AUCA Software Engineering Department and the 510300 IT competency standard.

4 Topics Covered

- Week 1–2: Introduction, Brief History, Dev. Environment (6 hours)
- Week 3–4: Vectors, Matrices, Scene Graph, Camera (6 hours)
- Week 5–6: Geometry, Buffer Objects, GPU Pipeline (6 hours)
- Week 7: Procedural Geometry (3 hours)
- Week 8–9: Materials, Shading, GLSL (6 hours)
- Week 10–11: Texturing and Mapping (6 hours)
- Week 12: Lambert, Phong, Blinn-Phong Shading (3 hours)
- Week 13–14: Instancing and other Rendering Optimizations (6 hours)
- Week 15–16: Real-time Graphics in Games, Building a Game with your Engine (6 hours)

5 Assignments and Exams

5.1 GitHub Checkpoints

Students are required to maintain private GitHub repositories provided by the instructor for all their assignments. They must periodically commit and push a specific number of lab and project solutions as directed by the faculty. Either the professors or teaching assistants will check the work regularly and assign points based on the completed assignments.

5.2 Labs and Projects

Throughout the course, students will be assigned several laboratory tasks. Additionally, they are required to develop one course project. Students are expected to defend their work to the instructor during both the Midterm and Final Examination periods.

6 Course Materials, Recordings and Screencasts

All course materials are available on GitHub at <https://github.com/auca/com.391>. By using GitHub, students will gain familiarity with the Git version control system and the widely-used GitHub service among developers.

Every class will be screencasted and uploaded to YouTube for student accessibility, though it's important to note that we do not guarantee every class will be recorded. Recordings will be done on a best-effort basis as time permits. Consider recording the class videos on your own computer if you need them to be available promptly. YouTube recordings can be located in the course repository at <https://github.com/auca/com.391>. While recordings provide flexibility, they should not be a substitute for attending classes. Active participation is crucial for success in this course. Each unexcused absence will result in a one-point deduction from your grade. Accumulating five or more unexcused absences may lead to an X grade. If overall attendance is poor, the instructor reserves the right to discontinue class recordings.

Access the course lectures remotely via Zoom at <http://com-391-zoom.auca.space>. When joining the Zoom session, students must identify themselves by providing their first and last names in Latin characters, properly capitalized.

7 Software

Students are recommended to install the following software on their machines.

- Git: <https://git-scm.com>
- Python 3: <https://www.python.org>
- CMake: <https://cmake.org>

On macOS install Xcode <https://developer.apple.com/xcode>. On Windows install Visual Studio 2022 Community <https://visualstudio.microsoft.com/vs> or any other edition if you can acquire it legally.

You can use any other IDEs (like CLion) or code editors (like VS Code, Vim, Emacs) that you like. Please note that we can only provide support for Visual Studio 2022 on Windows and Xcode on macOS.

You can also work on Linux, but you will need to manage the installation process for drivers, tools, and editors by yourself.

8 Hardware

We have made every effort to choose a graphics API that will likely work on a wide range of machines, from high-end personal computers to low-powered notebooks. However, each year we encounter students whose computers have outdated GPUs or buggy drivers that are unable to run our code. Unfortunately, this means you will need to invest some effort into finding an environment that is compatible with our course. Please be aware that we cannot provide extensions or preferential treatment if your machine is unable to handle our code. One option is to try a different native or virtualized environment (e.g., GNU/Linux) on your computer. Open-source drivers often perform better on older devices than those provided by the manufacturer. Additionally, you can reach out to the AUCA IT department for assistance. They may be able to provide a computer capable of running the programs, but please note that their resources are limited.

9 Reading

1. 3D Math Primer for Graphics and Game Development, Second Edition by Fletcher Done and Ian Parberry (ISBN-13: 978-1568817231, ISBN-10: 1568817231)

9.1 Supplemental Reading

1. Mathematics for 3D Game Programming and Computer Graphics, Third Edition by by Eric Lengyel (AUCA Library Call Number: QA76.76. C672 L46 2012, ISBN-13: 978-1435458864, ISBN-10: 1435458869)
2. Game Engine Architecture, Third Edition by Jason Gregory (ISBN-13: 978-1138035454, ISBN-10: 1138035459)
3. Game Programming Patterns by Robert Nystrom (ISBN-13: 978-0990582908, ISBN-10: 0990582906)

10 Grading

10.1 GitHub Checkpoints

Your instructor will announce a periodic review of your work. You will be awarded up to the following number of points for such checks.

- Labs (25%)
- Project (10%)

10.2 Exams

- Midterm Exam (30%)
- Final Exam (35%)

10.3 Totals

- 100% is formed from the GitHub submissions (35%) and the two exams (65%).

10.4 Scale

- [92%–100] %: A
- [85%–92) %: A-
- [80%–85) %: B+
- [75%–80) %: B
- [70%–75) %: B-
- [65%–70) %: C+
- [60%–65) %: C
- [55%–60) %: C-
- [50%–55) %: D+
- [45%–50) %: D
- [40%–45) %: D-
- Less than 40%: F

Please note that requests for a higher grade due to points being marginally close will be ignored. For instance, 91.99 is an A-, NOT an A. Similarly, requests for extra assignments to boost points will also be disregarded.

11 Rules

Students are required to follow the rules of conduct of the Software Engineering Department and the American University of Central Asia.

11.1 Participation

Active work during the class may be awarded with extra points at the instructor's discretion. Poor student performance during a class can lead to points being deducted from the final grade.

Instructors may conduct pop-checks during classes at random without prior notice. Students MUST be ready for every class in order not to lose points. Students absent without a good reason from such classes with graded work will also lose points unless it is force-majeure circumstances. Instructors must be notified in advance about why a student is absent not to lose points.

11.2 Questions

We believe that a question from one student is most likely a question that other students are also interested in. That is why we encourage students to use the online

discussion board of the LMS (Learning Management System) that you use (e.g., AUCA e-Course System) to ask questions in public that other students can see and answer. We discourage students from asking questions through E-mail. If it is a private matter, write direct messages to your instructor through the LMS system too. We will not be answering most E-mail messages this semester (unless it is a severe emergency) to consolidate all the course correspondence in one place.

Do not post the complete source code for any task on the LMS discussion board. You will get zero for that work for any such public post. Do not ask generic questions about your code to know why it does not work. Please spend some time thinking about your code, debugging it.

11.3 Late Policy

Late submissions and late exams are not allowed. Exceptions may be made at the professor's discretion only in force-majeure circumstances. If you got ill, got severe personal issues, got problems with your computer or the Internet, you **MUST** notify instructors at least 24 hours in advance. Otherwise, we will not give you an extension. We will consider that you were procrastinating until the very last day. We will also not be giving more than one emergency extension throughout the course.

Six hours before the deadline for any work on the course, instructors will go into a silent mode. No questions will be answered about the work that has to be submitted, no requests to have office hours will be considered. However, at any other work time before the deadline, we will try our best to answer your questions and help you through Zoom or in our office.

11.4 Exam and Task Submission Ceremonies

Students **MUST** follow exam and task submission ceremonies. It means they **MUST** strictly follow all the rules specified by the instructors in written or verbal form. Failure to do so will result in lost points. Throughout your career, you will have to work with various supporting documents (contracts, timesheets, etc.). It is a good idea to start learning to work with such documents accurately early. We will remove points for not following these rules or even refuse to accept your exam defense or tasks submitted to us. We will also give zero for not following deadlines or the strict exam timing rules.

11.5 Administrative Drop

Instructors have a right to drop a student from the course for non-attendance. If you have five classes or more missed without an excuse, the faculty may consider dropping you by giving you the *X* grade.

11.6 Incomplete Grade

Similar to the policy for late exams, the grade *I* may be awarded only in highly exceptional circumstances. Students **MUST** initiate a discussion about receiving an

I grade with the instructors well in advance and NOT during the last week before final exams.

11.7 Academic Honesty

Plagiarism is the act of copying or stealing someone else's words or ideas and presenting them as one's own. This definition encompasses various task elements, including but not limited to program code, comments, software documentation, abstracts, reports, diagrams, and statistical tables.

The following are examples of plagiarism in the context of a Software Engineering course:

- Presenting code written by others as your own
- Purchasing code, software, or any project-related content from online platforms or other sources and submitting it as your own creation
- Using algorithms, patterns, or architectural designs without acknowledging the source
- Incorporating code snippets, sentences, design patterns, or any intellectual content from sources, published or unpublished, without proper citation
- Modifying someone else's code or design (e.g., changing variable names, changing the structure of the code) and claiming it as original
- Utilizing graphics, data sets, audio, video, or other elements from external works without proper acknowledgment.

Engaging in plagiarism is not only unethical but also undermines the educational process. The consequences for plagiarism in this course for all parties involved are as follows:

- First instance: The students will receive a grade of zero for the plagiarized work, and a report will be filed with the Registrar's Office.
- Second instance: The students will receive an F grade for the entire course.

It's important to note that both parties involved in plagiarism—the one who plagiarizes and the one whose work was copied—will face equal consequences. This underscores the imperative for honest students to exercise caution in ensuring the security of their work. It is the student's responsibility to guarantee that their assignments, code, or any related content can only be accessed by them and the course instructors. Sharing, unintentionally exposing, or not securely storing one's work can lead to unintended consequences and sanctions.

Students are advised against rote memorization of code for examinations. Relying solely on memorization is an ineffective learning strategy in programming. Examinations in this course may contain open-ended questions targeting the student's analytical and design skills, and memorization may lead to answers that are off-target and of subpar quality.

In addition to the rules outlined in this syllabus, we abide by all global university policies concerning plagiarism. Should the global university rules evolve to be more consequential or stringent than what is stipulated here, those university-wide regulations will take precedence over our course-specific rules.