# 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 12:45–14:10, Room 207
Lab: Thursday 12:45–14:00, Lab 432
Lab: Thursday 14:10–15:25, Lab 432

**Course Repository**
`https://github.com/auca/com.391`

## 2 Contact Information

**Professor**
Dmitrii Toksaitov
toksaitov_d@auca.kg

**TAs**
Pavel Ges
ges_p@auca.kg

**Office**
AUCA, room 315

**Office Hours**
By appointment throughout the work week (write to your professor or TA to make an appointment)

# 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 accord to the goals of the AUCA Software Engineering Department and the 510300 IT competency standard (including competency elements OK 1–7, 1–7, 1–15).

# 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 will have to maintain a personal private GitHub repository with all their works shared with their instructor. Students have to periodically commit and push a specific number of lab and project solutions as told by the instructor. The instructor and TAs will regularly check the work and give points for the accomplished work.

## 5.2 Labs, Course Project

Students will have a number of laboratory tasks and have to finish one course project developing a clone of a popular computer game. Students will have to defend their work to the instructor during midterm and final examination sessions.

# 6 Course Materials, Recordings and Screencasts

Students will find all the course materials on GitHub. We hope that by working with GitHub, students will become familiar with the Git version control system and the popular (among developers) GitHub service. Though version control is not the focus of the course, some course tasks may have to be submitted through it on the GitHub Classroom service.

Every class is screencasted online and recorded to YouTube for students' convenience. The course Zoom link is `http://bit.do/auca-com-391`. YouTube links can be found on the course repository at `https://github.com/auca/com.391`. An ability to watch a class remotely at any time MUST NOT be a reason not to attend at least the online Zoom session. Active class participation is necessary to succeed in this course.

# 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 2019 (or 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 2019 (and 2022) on Windows and Xcode on macOS.

You can also work on Linux, but you have to figure out the driver/tools/editor-installation process on your own.

# 8 Hardware

We did our best to select a graphics API that will most likely work on your machine, whether it is a top-of-the-line expensive personal computer or a low-powered notebook. Nevertheless, every year we have students with computers where GPUs are too old, or the drivers are too buggy and don't run our code. Unfortunately, you will have to put some effort into acquiring an environment that will work on our course. We will not give you extensions or preferential treatment if your machine can't handle our code. Your first option is to try a different native or virtualized environment (e.g., GNU/Linux) on your computer. Some open-source drivers are known to work better on old devices than those one provided by the manufacturer. You can also contact the IT department of AUCA to offer you a computer where you can run the programs. Note that they have a limited amount of machines.

# 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 (15%)
- Project (20%)

## 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 to award a better grade if the number of points is close to such a grade will be ignored. For example, 91.99 is A-, NOT A. Likewise, requests to get extra assignments to increase the number of points will also be overlooked entirely.

# 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 up to 5 extra points at the instructor's discretion.

Poor student performance during a class can lead to up to 5 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. TAs and 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 Canvas online discussion board 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 TAs or your instructor there on Canvas 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 on our LMS (Learning Management System).

Do not post the complete source code for any task on the Canvas 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 me 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. Usually, it will be Saturday and Sunday (which are not working days for us anyway). 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 Ceremonies

Students MUST follow exam ceremonies. It means they MUST prepare task list forms with all points appropriately calculated. They MUST submit them correctly. They must bring task list forms to the exam. 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. We will give zero for not following the strict exam timing rules.

## 11.5 Incomplete Grade

As with late exams, the grade *I* may be awarded only in exceptional circumstances. The student must start a discussion on getting the grade *I* with the instructors in advance and not during the last week before the final exams.

## 11.6 Academic Honesty

Plagiarism can be defined as "an act or an example of copying or stealing someone else's words or ideas and appropriating them as one's own". The concept of plagiarism applies to all tasks and their components, including program code, comments, documentation, abstracts, reports, graphs, statistical tables, etc.

The following are examples of some common acts of plagiarism:

1. Representing the work of others as their own
2. Using other people's ideas or phrases without specifying the author
3. Copying code snippets, sentences, phrases, paragraphs or ideas from other people's works, published or unpublished, without referring to the author
4. Replacing selected words from a passage and using them as your own

5. Copying from any type of multimedia (graphics, audio, video, Internet streams), computer programs, graphs or diagrams from other people's works without representation of authorship

6. Buying work from a website or from another source and presenting it as your own work

In addition to being unethical, this indicates that the student has not studied the given material. Tasks written from somewhere for 5% or less will be assessed accordingly or will receive a 0 at the discretion of the teacher. If plagiarism is more than 5%, the case will be transferred to the AUCA Disciplinary Committee.

In this course, teamwork is NOT encouraged. The same blocks of code or similar structural pieces in separate submissions will be considered academic dishonesty, and all parties will get zero for the task.

Students are not recommended to memorize lab and project code before exams, as this is a difficult and inefficient way to learn; and since practice exams may consist of open questions designed to test a student's analytical skills, memorization invariably leads to the fact that the answers are inappropriate and of poor quality.