

IUCA-CG, Computer Graphics

International University of Central Asia
Information Technologies Program

1 Course Information

Course Code

IUCA-CG

Course ID

TBD

Prerequisite

Object Oriented Programming

Credits

6

Professors, TAs, Time, Place

Lecture (Dmitrii Toksaitov): Monday 10:50–12:05, Remotly

Friday (Dmitrii Toksaitov): Friday 12:45–14:00, Remotly

Course Repository

<https://github.com/iucau/iuca-cg>

Class Discussions

<https://piazza.com/iuca.kg/spring2020/iucacg>

2 Contact Information

Instructor

Toksaitov Dmitrii Alexandrovich

toksaitov_d@iuca.kg

Office Hours

By appointment throughout the work week

Remotely through Skype on Saturday and Sunday from 18:00 to 22:00

3 Course Overview

The course teaches students fundamentals of computer graphics through a process of developing 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, or 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 IUCA Information Technologies Program 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, First Program (6 hours)
- Week 3–4: Vectors, Matrices, Scene Graph, Camera (6 hours)
- Week 5–6: Geometry, Buffer Objects, GPU Pipeline (6 hours)
- Week 7–8: Materials, Shading, GLSL (6 hours)
- Week 9: Lambert, Phong, Blinn-Phong Shading (3 hours)
- Week 10–11: Texturing and Mapping (6 hours)
- Week 12: Procedural Geometry (3 hours)
- Week 13–14: Particle Systems, Rendering Optimizations (6 hours)
- Week 15–16: Real-time Graphics in Games, Building a Game with your Engine (6 hours)

4.1 Lectures

To ensure that students work actively and watch class recording, every week there will be a remote quiz on the topics discussed in the class video. Students have to submit it before 12:00 of every Sunday.

Students will have to take midterm and final theory examinations on the same topics discussed during lectures. Each examination is in the form of an electronic quiz with a set of open and multiple choice questions. Instructor will attend midterm and final examinations.

5 Labs

Students will have to finish 10 lab tasks. In every task students will build a functioning part of a 3-D graphics engine together with the instructor.

6 Course Materials, Recordings and Screencasts

Students will find all the course materials on GitHub. We hope 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.

This is a remote class. Every class is screencasted online and recorded to YouTube. Screencast sessions have live chat so that students can interact with the professor. Students **MUST** watch all the recordings in order to prepare for midterm and final examinations. Class discussions will go on Piazza. Office hours are to be conducted remotely through WebEx, Zoom or Slack.

7 Reading

1. 3D Math Primer for Graphics and Game Development, Second Edition by Fletcher Done and Ian Parberry (ISBN: 978-1-4398-6981-9)

7.1 Supplemental Reading

1. Mathematics for 3D Game Programming and Computer Graphics, Third Edition by Eric Lengyel (ISBN: 978-1435458864)
2. Game Engine Architecture, Second Edition by Jason Gregory (ISBN: 978-1568814131)
3. Game Programming Patterns by Robert Nystrom (ISBN: 978-0990582908)

8 Grading

- Weekly Quizzes (15%)
- Theory Midterm (15%)
- Theory Final (20%)
- Practice Midterm (20%)
- Practice Final (25%)
- Piazza Participation (5%)

9 Scale

- 92%–100%: A
- 85%–91%: A-

- 80%–84%: B+
- 75%–79%: B
- 70%–74%: B-
- 65%–69%: C+
- 60%–64%: C
- 55%–59%: C-
- 50%–54%: D+
- 45%–49%: D
- 40%–44%: D-
- Less than 40%: F

10 Rules

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

10.1 Participation

Active work during the class may be awarded with up to 5 extra points at the instructor's discretion.

10.2 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 Piazza to ask questions in public that other students can see and answer and NOT ask them through E-mail in private UNLESS the question itself is about private matters to discuss with the professor.

10.3 Late Policy

Late submissions and late examination defence are not allowed. Exceptions may be made at a discretion of the professor only in force-majeure circumstances.

10.4 Incomplete

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

10.5 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, abstracts, reports, graphs, statistical tables, etc.

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 IUCA Disciplinary Committee.

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

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

Students are not recommended to memorize before exams, as this is a difficult and inefficient way to learn; and since practice exams 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.