UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

COURSE SYLLABUS

Computer Graphics

2526-3-E3101Q134

Aims

The student will acquire skills for the design and the implemention of 2D and 3D computer graphics algorithms.
In particular it will be able to apply the acquired skills to develop applications based on the OpenGL rendering API and
exploiting the potential of a programmable rendering pipeline through the use of the graphic shaders.

Knowledge and understanding
Knowledge of the fundamental concepts of 2D and 3D computer graphics.
Knowledge of the techniques and algorithms underlying the representation and management of 2D and 3D data.

Applied knowledge and understanding
Ability to apply knowledge to the creation of rendering programs
Ability to use graphics APIs for the development of graphics applications

Judgment and decision-making
Ability to choose the most suitable techniques for the application context

Communication skills
Ability to present and discuss the development of computer graphics projects, justifying design choices

Learning skills
Ability to independently explore advanced aspects of computer graphics

Contents

Tools for construction, transformation and presentation of geometric models in a 3D world. Basic knowledge of

standard 3D computer graphics pipelines and API. Creation of computer graphics applications using state-of-the-art techniques and the most widespread reference standards.

Detailed program

Introduction to the 3D rendering pipelines
- Fixed
- Programmable

OpenGL
- Introduction to the API
- Working logic
- The shaders
- GLSL language (OpenGL Shading Language)

Development tools
- G++
- GLEW
- FREEGLUT
- GLM
- ASSIMP

Mathematics for computer graphics

The rendering process
- 3D modeling
- Geometric transformations
- Change of reference systems
- Room transformation
- Projection transformation
- Viewport transformation
- Clipping
- Hidden surface removal
- Depth test

Approximating the light
- Local illuminance models
- Shading algorithms

Giving the details
- Texture Mapping
- Bump Mapping
- Shadow Mapping
- Environment Mapping

Physical Based Rendering
- Global illumination models
- Ray Tracing

Examples of 3D modeling software
- Blender
- POVray

Prerequisites

The student must necessarily have a good knowledge of at least one programming language (preferably C++), of linear algebra, and geometry.

Teaching form

Teaching given in Italian.

The course is structured as follows:

40 hours of face-to-face lectures in delivery and interactive in-presence mode
20 hours of exercises in delivery and interactive in-presence mode
12 hours of laboratory in delivery and interactive in-presence mode

Textbook and teaching resource

Steven K. Feiner, Andries van Dam, John F. Hughes, Morgan McGuire, David F. Sklar, James D. Foley, Kurt Akeley, Computer Graphics: Principles and Practice, Third Edition, Addison-Wesley Professional

Graham Sellers, Richard S. Wright Jr., Nicholas Haemel, "OpenGL Superbible: Comprehensive Tutorial and Reference" 7th edition, Addison-Wesley.

Slides and handouts.

Semester

III° year. Second Semester.

Assessment method

The assessment includes a written test, a project and an oral.

The written test consists of open questions and questions with multiple choices on topics presented in the course. Some questions can be replaced by brief exercises.

The project aims to make you familiar with the topics seen in class by applying them in new contexts. The project can be carried out in one of the following ways:

1. create 3D models complete with textures.

2. analyze and implement advanced rendering techniques in small demo applications.

The oral exam consists in a questions about the technical and theoretical choices made in the project, and the topics of the lectures.

At the moment, there are no ongoing tests.

Office hours

By appointment

Sustainable Development Goals

INDUSTRY, INNOVATION AND INFRASTRUCTURE