COURSE DESCRIPTION CARD - SYLLABUS

Course name
Computer Graphics and Visualization

Course

Field of study
Computing

Year/Semester
2/4

Area of study (specialization)

Profile of study
general academic

Level of study
First-cycle studies

Course offered in
english

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture	Laboratory classes	Other (e.g. online)
16	16	
Tutorials	Projects/seminars	

Number of credit points

4

Lecturers

Responsible for the course/lecturer:
dr inż. Witold Andrzejewski
email: witold.andrzejewski@cs.put.poznan.pl
tel. (0-61) 665-2965
Instytut Informatyki
60-965 Poznan, ul. Piotrowo 2

Responsible for the course/lecturer:

Prerequisites

Knowledge:
Student starting this module should have basic knowledge regarding programming languages, geometry and computer system architectures.
Skils
He/she should have the skills to acquire knowledge from the designated sources of information and implement simple programs in C/C++.
Social competences:
Student should understand the need to extend his/her competences and be ready to work in a team.

Course objective

1. Teach students the basic concepts and definitions related to computer graphics.
2. Teach students the mathematical basics of 3D graphics.
3. Teach students the methods of 3D object animation.

4. Teach students the shading models and hidden surface removal methods.
5. Teach students different ways of 3d model representations.
6. Teach students basics of data visualization methods.
7. Develop students' computer graphics application programming skills by introducing popular computer graphics libraries.

Course-related learning outcomes

Knowledge

1. has an extended and in-depth knowledge of mathematics useful for formulating and solving complex computer science tasks related to computer graphics - [K1st_W1]
2. has a well-established theoretical knowledge regarding computer graphics - [K1st_W4]
3. knows the important directions and most important developments in the field of computer science and related research domains related to computer graphics - [K1st_W5]
4. knows the fundamental methods, techniques and tools employed to solve complex engineering tasks in the area of computer graphics - [K1st_W7]

Skills

1. can utilize multiple different data visualization methods at different stages of an it project implementation - [K1st_U2]
2. is able to formulate computer graphics algorithms and implement them using OpenGL - [K1st_U11]
3. can design data visualization methods processed in multiple different IT system types - [K1st_U14]

Social competences

1. understands that in the filed of computer graphics, knowledge and skills can quickly become obsolete - [K1st_K1]
2. understands the importance of knowledge in solving engineering problems - [K1st_K2]

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formative assessment:

a) lectures:
- based on answers to questions related to subjects covered during previous lectures,

b) laboratory classes:
- evaluation of correctness of implementation of assigned tasks (following provided lab. instructions),

Total assessment:

a) verification of assumed learning objectives related to lectures:
- evaluation of acquired knowledge on the basis of the written exam (a test, ~30 questions, total points achievable 30, 15 points needed to pass).

b) verification of assumed learning objectives related to laboratory classes:
- based on the project implemented by a team of students, each students' grade is evaluated based on the quality of his/hers part as well as answering to several project related questions.

Additional elements cover:

- discussing more general and related aspects of the class topic,
- ability to utilize knowledge covered in previous lectures
- showing how to improve the instructions and teaching materials.
- pointing out flaws in teaching materials and helping lecturer to improve them

Programme content

Lectures cover the following topics

Lecture 1. Basic concepts and definitions in the field of computer graphics. Introduction of: image buffering, hidden surface removal, basic texturing algorithms, basic texture types, several classes of special effects.

Lecture 2 and 3: Mathematical basics of 3D graphics. Short repetition of computational geometry. Introcution of homogenous coordinate system, geometrical transformations and their matrix representation, quaternions and their relation to 3D rotation, typical vertex processing pipeline in 3D application, view and projection matrices (perspective and orthogonal projection), geometrical transformations of normal vectors.

Lecture 4. Animation techniques. Vector graphics animation (per vertex animation, skeletal animation, inverse kinematics). Introduction to shading algorithms. Introduction of light abstraction types (point, directional, cone and surface lights).

Lecture 5. Shading models. Introduction of basic radiometry terms. Description of BRDF nad Schlicks BSF functions. Derivation of basic shading models: Lambertian diffuse model, Phong and Phong-Blinn model. Introduction of complex shading models such as: Cook-Torrance model.

Lecture 6. Introduction of raytracing rendering algorithm. Algorithms for detecting intersection of ray with a sphere, AAB bounding box and a triangle. Algorithms for generating main ray, shadow ray, reflected ray and transmitted ray.

Lecture 7. Description of 3d object representation methods including trimeshes, voxels, mathematically defined surfaces and particlesystems. Visualization techniques for such objects are provided as well.

Lecture 8. Data visualization. Introduction to data visualization process. Discussion of exemplary visualization methods for many different data types.

During laboratories students learn basics of computer graphics and implement simple exercises in OpenGL:

Laboratory 1: Introduction to OpenGL API. Discussion about the basic program structure and GLFW framework. Introduction to several basic topics related to drawing and animating of 3D models.

Laboratory 2: OpenGL exercises related to moving, rotating and animation 3D objects on a scene. Students gain the skills to correctly construct geometrical transformation matrices. Moreover introduction of methods allowing to draw arbitrary trimeshes.

Laboratory 3: Texturing and shading of objects in OpenGL via ready made shader programs.

Laboratory 4: Introduction to GLSL language. Simple exercises based on implementation of simple geometry transforming shaders and simple shading models.

Laboratory 5: Implementation of per-vertex shading models in GLSL, including Lambert and Phong models. Implementation of per-pixel shading models in GLSL including Phong and cell shading.

Laboratory 6: Texturing in GLSL. Multitexturing, Simple environment mapping.

Laboratory 7: Fur effect in GLSL as an illutration of instancing. Geometry shaders.

Laboratory 8: Normal mapping and Parallax mapping.

Teaching methods

1. Lectures: multimedia presentation, presentation illustrated with examples presented on black board, solving tasks, multimedia showcase.

2. Labs: solving tasks, practical exercises, discussion, teamwork, multimedia showcase.

Bibliography

Basic

1. Fundamental algorithms for computer graphics / ed. by Rae A. Earnshaw.
2. Mathematical Elements for Computer Graphics / Rogers David F., Adams J.Alan.
3. Computer graphics techniques : theory and practice / David F. Rogers, Rae A. Earnshaw (eds.).

Additional

1. OpenGL Superbible, fifth edition. Richard S. Wright, Jr., Nicholas Haemel, Graham Sellers, Benjamin Lipchak, Addison-Weslay Pearson Education.
2. Introduction to Computer Graphics, J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes, R.L. Phillips, Addison Wesley Longman

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,0
Classes requiring direct contact with the teacher	32	1,0
Student's own work (literature studies, preparation for laboratory classes, preparation for tests, project preparation) [1]	68	3,0

[1] delete or add other activities as appropriate