



COURSE DESCRIPTION CARD - SYLLABUS

Course name

Computer graphics [S2ETI1>GF]

Course

Field of study

Education in Technology and Informatics

Year/Semester

1/2

Area of study (specialization)

–

Profile of study

general academic

Level of study

second-cycle

Course offered in

polish

Form of study

full-time

Requirements

compulsory

Number of hours

Lecture

15

Laboratory classes

30

Other (e.g. online)

0

Tutorials

0

Projects/seminars

0

Number of credit points

3,00

Coordinators

dr hab. inż. Grażyna Sypniewska-Kamińska
grazyna.sypniewska-kaminska@put.poznan.pl

Lecturers

Prerequisites

1. Basic knowledge of mathematics and computer science corresponding to the core curriculum for the first-cycle studies. 2. The ability to think logically, skill for algorithmization and programming in C++. 3. The ability to self-study including the skill for search the needed information from various sources. 4. Awareness of the necessity to expand own competencies and develop skills.

Course objective

Learning the basics of computer graphics. Understanding the need and developing the ability to use computer graphics to present the results of engineering and research works.

Course-related learning outcomes

Knowledge:

1. The student who completed the course can explain the differences between the vector graphics and the raster graphics.
2. The student can explain the mechanism of event handling in the applications controlled by GUI.
3. He can explain elementary affine transformations and knows what is their importance in computer graphics.

Skills:

1. The student can construct the matrices of the affine transformations using the homogeneous coordinates.
2. He can create polygon meshes of solids and surfaces and save them as data structures.
3. He can create the algorithms intended for showing three dimensional objects employing planar projections and perspective drawings.
4. He can write a computer program showing an animation of a dynamical problem.

Social competences:

1. By preparing an independent project, the student shows himself the responsibility for the tasks entrusted to him.
2. The student, by creating and implementing algorithms, shows himself creativity, reliability and attention to detail.
3. He understands the importance of knowledge in the modern world. He is also well aware that the rapid development of knowledge causes the need for lifelong learning. He can organise the processes of learning and self-education.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Lectures: Credit basing on the final test in written that involves several theoretical questions and some practical tasks. Credit for a positive grade after obtaining at least 50% of the total points; linear grade scale.
Laboratory classes: Credit basing on the self-developed computer programs prepared during laboratory classes. Credit for a positive grade after obtaining at least 50% of the total points; linear grade scale.

Programme content

Lectures: Introduction - subject and applications of computer graphics. Vector and raster graphics. Graphical user interface (GUI) as an important application of computer graphics. Applications controlled by GUI - event handling. Graphical programming tools in C. Overview of the class that implements standard graphics functions. User coordinates definition rules. Representation of an object in the field of view. Rules for drawing objects with preserving the assumed proportions. Discussion of the details of the algorithm of the computer program used to draw a function of one variable. Two dimensional affine transformations. Three dimensional affine transformations. Homogeneous coordinates. Matrix notation of affine transformations in the homogeneous coordinates. Complex affine transformations. Overview of the class implementing the affine transformations. Polygon meshes of solids and surfaces. Introduction to projection - basic concepts. Planar geometric projections, parallel projections and perspective drawings. The parallel projection as the affine transformation in homogeneous coordinates. The matrix notation in the homogeneous coordinates for the isometric projection. One-point, two-point and three-point perspective drawings as the affine transformations in the homogeneous coordinates. Animation of images techniques. Animations in the real time and animations using given time scale factor. Principles of animation design - examples.

Laboratory classes: Vector and raster graphics. Graphical programming tools in C++ and Mathematica. The class that implements standard graphics functions. User coordinates definition rules. Representation of an object in the field of view. Rules for drawing objects with preserving the assumed proportions. Discussion of the details of the algorithm of the computer program used to draw a function of one variable. Affine transformations in 2D and 3D using a matrix notation in homogeneous coordinates. Creation of a class supporting affine transformations. Projecting 3D objects (parallel, isometric and perspective projection). Animation of images techniques.

Teaching methods

Lectures: lecture supported by multimedia presentations. The content that requires the implementation of calculations, or formulation of assumptions and the arrangement of the algorithm are discussing in detail at the blackboard.

Laboratory classes: Initially, the students are introduced to the graphical environment available in Visual Studio for creating applications in C++ and the graphical environment of Mathematica. Together with the teacher, simple computer programs are developed and discussed in detail, the main purpose of which is to familiarize students with the libraries responsible for handling graphics. Along with the following classes, the difficulty of the programs being developed increases and the teacher's direct contribution to the process

of their creation decreases.

Bibliography

Basic

1. J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes, R. L. Philips, Wprowadzenie do grafiki komputerowej, Warszawa, WNT 2001.
2. <https://learn.microsoft.com/en-us/windows/win32/api>.
3. I. Horton, Visual C++ 2005, Wyd. Helion, Gliwice 2006.
4. I. Horton, P. van Weert, Beginning C++17, Springer.

Additional

1. P. Kiciak, Podstawy modelowania krzywych i powierzchni. Zastosowania w grafice komputerowej, WNT, Warszawa 2000.
2. J. Matulewski, T. Dziubak, M. Sylwestrzak, R. Płoszajczak, Grafika. Fizyka. Metody numeryczne, PWN, Warszawa 2010.
3. A. Marciniak, Grafika komputerowa w języku Turbo Pascal, Wydawnictwo Nakom, Poznań 1998.

Breakdown of average student's workload

	Hours	ECTS
Total workload	80	3,00
Classes requiring direct contact with the teacher	45	2,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	35	1,00