COURSE DESCRIPTION CARD - SYLLABUS

Course name
Rich Internet Applications

Course

Field of study
Computing

Year/Semester
2/3

Area of study (specialization)
Games and Internet Technologies

Profile of study
general academic

Level of study
Second-cycle studies

Course offered in
Polish

Form of study
full-time

Requirements
elective

Number of hours

Lecture
15

Laboratory classes
15

Other (e.g. online)
0

Tutorials
0

Projects/seminars
0

Number of credit points

2

Lecturers

Responsible for the course/lecturer:
Dr. Eng. Andrzej Urbański

Responsible for the course/lecturer:

Prerequisites

Learning outcomes from first-cycle studies defined in the Resolution of the Senate of the PP, in particular the effects K_W1-2, K_W4, K_W6-15, K_U1-2, K_U4, K_U7-8, K_U14-20, K_U22-23, K_U26, K_K1-9, verified in the recruitment process for second-cycle studies - these effects are presented on the faculty website www.fc.put.poznan.pl

The student starting this course should have basic knowledge of OpenGL programming. He should have the ability to solve basic problems in the field of computer game programming and graphics, and the ability to obtain information from the indicated sources. He should also be ready to cooperate as part of the team.

In addition, in terms of social competences, the student must present attitudes such as honesty, responsibility, perseverance, cognitive curiosity, creativity, personal culture, respect for other people.

Course objective

1. Provide students with basic knowledge of 3D graphics programming in the field of webGL programming for various needs

2. Developing students' ability to solve problems related to 3D programming

3. Shaping students' teamwork skills in the field of 3D programming

Course-related learning outcomes

Knowledge

1. has a structured, theoretically founded general knowledge in the field of network technologies, languages and graphic programming paradigms

2. has theoretically founded detailed knowledge related to selected issues in the field of computer science, such as: programming graphics, programming games, programming multimedia websites

3. knows the basic methods, techniques and tools used to solve complex engineering tasks in the field of graphics and game programming

Skills

1. can determine the directions of further learning and implement the self-education process in the field of programming in the webGL environment

2. is able to evaluate the usefulness and the possibility of using new achievements (methods and tools) and new IT products

3. is able to propose improvements (improvements) to existing technical solutions

Social competences

Social competence

1. understands that in computer science knowledge and skills very quickly become obsolete

2. knows the examples and understands the causes of malfunctioning information systems, which have led to serious financial and social losses, or to a serious loss of health and even life

3. is able to properly define the priorities for the implementation of the tasks set by himself or others

Methods of verification of learning outcomes and assessment criteria

The learning outcomes presented above are verified as follows:

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formative assessment:

a) in the field of lectures:

• on the basis of answers to questions about the material discussed in previous lectures,

b) in the field of laboratories:

• based on the assessment of the current progress in the implementation of tasks,

Summative assessment:

a) in the field of lectures, verification of the assumed learning outcomes is carried out by:

• assessment of the knowledge and skills shown in the problem-based written test (the student may use any teaching materials) 5 questions, 10 points, minimum 5 points for a satisfactory grade

• discussion of the results of the test,

b) in the field of laboratories, verification of the assumed learning outcomes is carried out by:

• evaluation of the report prepared partly during the classes and partly after their completion; this assessment also includes the ability to work in a team,

• assessment and "defense" by the student of the report on the implementation of the project,

Obtaining additional points for activity during classes, especially for:

• discuss additional aspects of the issue,

• the effectiveness of applying the acquired knowledge while solving a given problem,

• remarks related to the improvement of teaching materials,

• identifying students' perceptual difficulties enabling the ongoing improvement of the teaching process.

Programme content

The lecture program covers the following topics:

A reminder of the OpenGL programming news. Schematic diagram of a computer game, and the skeleton of its code in webGL. Programming in pure webGL with manual graphics coding. Preparing graphics in programs and webGL code for its import. Using the Coppercube graphic editor. Adding event code in JavaScript. Three.JS environment and its use in programming computer games. Computer camera programming for game control via gestures.

Laboratory classes are conducted in the form of fifteen-hour laboratory exercises, preceded by a 1-hour instructional session at the beginning of the semester. Classes are carried out by 2-person teams of students. The laboratory program covers the following topics:

WebGL programming. Creating graphics in the WebGL environment for given simple examples. Enriching the game mechanics with code in JavaScript.

Teaching methods

1.Lecture: multimedia presentation, presentation illustrated with examples given on the blackboard.

2. Laboratory exercises: problem solving, practical exercises, discussion, team work.

Bibliography

Basic

Jacob Seidelin HTML5. Tworzenie gier Helion, Gliwice, 2012

Karl Bunyan HTML5 : tworzenie gier z wykorzystaniem CSS i JavaScript, Helion, 2016.

Additional

Tony Parisi Aplikacje 3D : przewodnik po HTML5, WebGL i CSS3 Helion, 2015.

Sobiesiak, Karol., Sydow, Piotr.Shadery : zaawansowane programowanie w GLSL PWN, 2015.

Breakdown of average student's workload

	Hours	ECTS
Total workload	50	2,0
Classes requiring direct contact with the teacher	30	1,5
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests/exam, project preparation) [1]	20	0,5

[1] delete or add other activities as appropriate