STUDY MODULE DESCRIPTION FORM

Name of the module/subject	Code
Rich Internet Applications	**1010512311010513877**

Field of study	Profile of study (general academic, practical)	Year /Semester
Computing	**general academic**	**1 / 1**
Elective path/specialty: **Games and Internet Technologies**	Subject offered in: **Polish**	Course (compulsory, elective): **obligatory**
Cycle of study: **Second-cycle studies**	Form of study (full-time,part-time): **full-time**	
No. of hours: Lecture: **15** Classes: **-** Laboratory: **15** Project/seminars: **-**		No. of credits: **2**
Status of the course in the study program (Basic, major, other): **major**	(university-wide, from another field): **from field**	
Education areas and fields of science and art: **technical sciences** **Technical sciences**		ECTS distribution (number and %): **2 100%** **2 100%**

Responsible for subject / lecturer:

dr inż. Andrzej Urbański
email: andrzej.urbanski@put.poznan.pl
tel. +48616652984
Faculty of Computing
ul. Piotrowo 3 60-965 Poznań

Prerequisites in terms of knowledge, skills and social competencies:

1	**Knowledge**	Learning objectives of the first cycle studies defined in the resolution of the PUT Academic Senate, especially 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 that are verified in the admission process to the second cycle studies ? the learning objectives are available at the website of the faculty www.fc.put.poznan.pl Student starting this module should have basic knowledge regarding OpenGL programming.
2	**Skills**	He/she should have skills allowing solving basic problems related to computer games and graphics programming. Student should understand the need to extend his/her competences and to work in a team.
3	**Social competencies**	In addition, in respect to the social skills the student should show attitudes as honesty, responsibility, perseverance, curiosity, creativity, manners, and respect for other people.

Assumptions and objectives of the course:

1. Provide students knowledge regarding 3D graphics programming in the area of webGL programming for different needs.
2. Develop students? skills in solving problems related to 3D programming
3. Develop students? skills to work in a team in the area of 3D programming.

Study outcomes and reference to the educational results for a field of study

Knowledge:

1. 1. have sorted, on theoretical basis general knowledge in the area of network technologies, languages, paradigm of graphics programming - [K2st_W4 ++]
2. 2. have wide and in-depth knowledge on selected basic topics in computer science especially: graphics programming, games programming, multimedia web pages programming - [K2st_W5 +]
3. 3. know basic methods, techniques and tools used in solving complex engineers tasks in the area od games and graphics programming - [K2st_W8 ++]

Skills:

1. 1. s able to plan and arrange self-education process in particular covering issues of webGL. - [K2st_U5 +]
2. 2. can evaluate applicability of new achievements(methods and tools) and new IT products - [K2st_U13 ++]
3. 3. is able to propose enhancements of existing technical solutions - [K2st_U21 ++]

Social competencies:

1. 1. understands that knowledge and skills related to computer science and data mining quickly becomes non relevant - [K2st_K1 +]
2. 2. knows examples and understands reasons for bad working IT systems, which lead to serious financial and social loses, and even to peoples death - [K2st_K4 +]
3. 3. is able to correctly assign priorities to his own or others tasks - [K2st_K6 +]

Assessment methods of study outcomes

Formative assessment:

a) lectures:

based on answers to question in the written exam,

b) laboratory classes:

evaluation of doing correctly 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.

discussion of correct answers in the exam

b) verification of assumed learning objectives related to laboratory classes:

evaluation of lab reports (partly started during classes, finished after them) also ability to work in a team,

evaluation and "defence" by student of his report,

Additional points for the activity during course cover:

discussing more general and related aspects of the class topic,

efficiency of the usage of acquired knowledge during solving given problem,

showing how to improve the instructions and teaching materials.

showing perception problems of students enabling current improvements of didactic process.

Course description

The lecture should cover the following topics:

Recall of OpenGL programming knowledge. Scheme of game program and skeleton of OpenGL code. Programmimng in pure webGL with handwritten graphics. Preparing graphics in programs and webGL code to import it. Use of graphics editor in Coppercube. Addition of events code in JavaScript. Three.JS environment and its usage in computer games programming. Programming computer webcamera to make gesture driven software.

The lab course will have fifteen meetings. First is instructional session at the beginning of the semester. Exercises are performed by two student teams. Laboratory exercises themes: webGL programming, creating graphics in webGL environment for a given simple examples. Enhancing games mechanics using JavaScript code.

Learning 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, competitions or case studies

Basic bibliography:

1. Wprowadzenie do HTML5. Nauka HTML5 i JavaScriptu na przykładzie gier, Altmeyer Jeannine, Helion, 2017.

2. Jacob Seidelin "HTML5. Tworzenie gier", Helion, Gliwice, 2016

Additional bibliography:

1. Tony Parisi "WebGL: Up and Running", O'Reilly, 2016.

2. Richard S. Wright, Jr., Nicholas Haemel, Graham Sellers, Benjamin Lipchak "OpenGL. Księga eksperta.", Helion, Gliwice, 2016.

Result of average student's workload

Activity	Time (working hours)

1. 1. participating in laboratory classes	15
2. 2. preparing to laboratory classes:	7
3. 3. finishing reports from laboratory classes (in addition to laboratory classes):	15
4. 4. consulting issues related to the subject of the course; especially related to t laboratory classes and projects,	8
5. 5. writing a program	4
6. 6. preparing to assessment tests	4
7. 7. participating in lectures	15
8. 8. studying literature / learning aids	3
9. 9. preparing to and participating in exams	4

Student's workload

Source of workload	hours	ECTS
Total workload	76	2
Contact hours	38	1
Practical activities	52	1