SYLLABUS

1. Course title:

Computer Graphics

2. Code:

RI205

3. Cycle of study:

1

4. ECTS credits:

6

5. Type of course:

(•) Mandatory ( ) Elective

6. Prerequisites:

[RI101] Introduction to Programing

7. Class restrictions:

8. Duration / semester:

1 6

9. Weekly contact hours:

9.1. Lectures:	3
9.2. Seminars:	1
9.3. Laboratory/Practice classes:	1

10. Faculty:

Faculty of Electrical Engineering

11. Department/study program:

Electrical Engineering and Computer Science

12. Lecturer:

PhD Emir Skejić, associate professor

13. Lecturer's e-mail:

emir.skejic@untz.ba

14. Web site:

www.fet.ba

15. Course aims:

This course gives an introduction to theoretical and practical concepts of computer graphics. It is implied that the student is well acquainted with the programming in the C / C ++ programming language and that he has the appropriate mathematical knowledge in the field of analytical geometry and linear algebra. The course also enables the development of programming skills in computer graphics through the OpenGL assignments/final project solving.

16. Learning outcomes:

- Obtain basic knowledge from theoretical and practical concepts of computer graphics;
- Improve programming skills in computer graphics;
- Have the ability to independently design OpenGL medium-level complexity applications;
- Get an insight into the problem of photorealistic rendering in real time.

17. Course content:

1. Hardware and software components of graphic systems. Output primitives. 2D and 3D geometric transformations.
2. 2D viewing: viewing pipeline, clipping, culling.
3. 3D viewing: viewing pipeline, viewing parameters, projection, view transformation, clipping, detection of visible surfaces.
4. Models of local illumination. Texture mapping.
5. Curves and curved surfaces. OpenGL API.

18. Learning methods:

- lectures with the use of multimedia resources;
- auditory exercises;
- laboratory exercises;
- independent solving of OpenGL programming assignments.

19. Assessment methods:

After half of the semester, students take a written test (the first colloquium) that covers previously discussed lectures and exercises. The student can score up to 20 points in the first midterm exam. After completing the semester, the students write a written test (second colloquium) that covers the topics covered by the lectures and exercises from the second part of the semester. The student on the second midterm exam can score up to 25 points. Both tests put all the students on the subject at the same time, thereby achieving the level of knowledge that is being tested and the conditions under which the student takes the exam. Also, for a continuous activity on lectures and exercises throughout the semester, the student can achieve 0 to 5 points.

The final exam is written and is realized in the form of a final project design. The project can be passed if the student successfully solves 50% of the tasks the project consists of. The maximum number of points a student can achieve from the final project is 50.

Checks on all forms of knowledge are recognized as a cumulative test if the result is positive after each individual check and is at least 50% of the total of the predicted and / or required knowledge and skills.

In order for a student to pass the subject must have at least 54 cumulative points, out of which at least 25 points on the final project.

20. Assessment components:

The assessment of the exam is based on the total number of points the student has obtained by fulfilling the prerequisites and passing the exam according to the quality of the acquired knowledge and skills, and it contains a maximum of 100 points and is determined according to the following scale:

Student Obligations	Points
- exercise activity	max. 5
- Colloquia /programming assignments	max. 45
- final project	max. 50

21. Required reading list:

1. Donald Hearn, M. Pauline Baker. Computer Graphics with OpenGL (4th Edition). Pearson, 2010.
2. Samuel R. Buss. 3-D Computer Graphics: A Mathematical Introduction with OpenGL. Cambridge University Press, 2003.

22. Web sources:

https://www.opengl.org/
https://developer.nvidia.com/opengl
https://www.khronos.org/opengles/

23. Applicable starting from the academic year:

2016/2017

24. Adopted in the Faculty/Academy session:

04.04.2016