Obuda University John von Neumann Faculty of Informatics			*Institute of Biomatics and Applied Artificial Intelligence*	
Name and code: Programming robots in ROS			**Credits*: 4***	
Computer Science Engineering MSc			*2022/23 year I. semester*	
Subject lecturers: Péter Galambos, Tamás Nagy				
Prerequisites (with code):	Software Design and Development I. (NIXSF1EBNE)			
Weekly hours:	Lecture: 1	Seminar.: 0	Lab. hours: 2	Consultation: 0
Way of assessment:	Midterm grade			

Course description:

Goal:
The Robot Operating System (ROS) is a platform widely used in research and also in the industry. The students will learn how to develop ROS applications in Python programming language. The aim of the course is to get the students acquainted with ROS, and also to give them an opportunity to practice Python.

Course description:

ROS introduction, setting up the development environment. Implement ROS packages in Python. Basic ROS communication, implementing publishers and subscribers. Principles of robotics, programming a simulated robot in joint and workspace. Roslaunch, ROS parameter server. Acquisition and processing of sensory data in ROS. Programming da Vinci surgical robot in simulated environment. Programming humanoid robot. In simulated environment. Define custom messages. ROS service and ROS action. URDF, interfacing to web environment: RosBridge, RoslibJS. A galnce to ROS 2.

Lecture schedule

Education week	*Topic*
1.	Introduction to ROS. Introduction to Python.
2.	Setting up the development environment. Execution of ROS examples.
3.	Implement ROS packages in Python. Principles of ROS communication, imlementing publisher and subscriber.
4.	Chosing topics for project work. Practicing ROS communication.
5.	Principles of robotics. Programming a simulated robot in joint and cartesian space.
6.	Roslaunch, ROS parameter server. Rosbag.
7.	URDF. Web interafces: RosBridge, RoslibJS.
8.	Acquisition and processing of sensory data in ROS.
9.	Project work milestone.
10.	Programming a simulated da Vinci surgical robot.
11.	Programming a simulated humanoid robot.
12.	Defining ROS message types. ROS service.
13.	Ros action. A glance to ROS 2.
14.	Presentation of the project works.

Midterm requirements

Student participation in the lectures and labs is required.
All homeworks and the classroom test are required to complete during the midterm.

Homework schedule	
Education week	*Topic*
6.	Principles of ROS, publisher, subscriber. Principles of robotics.
13.	Roslaunch, ROS parameter server. URDF. RosBridge, RoslibJS. ROS service.

Final grade calculation methods

Students are to write 2 short classroom test during the semester, and are also obliged to finish a project work on the chosen topic. At the end of the semester, the projects will be graded based on the students' presentation. Requirents for the acceptance of the project work is to be the students own work, and running the code results in appraisable output. Further grading is given along the following aspects: completeness of the solution, proper ROS communication, proper structure of the software, quality of implementation, documentation. The requirements for the accpmlishment of the course: the results of both tests are at least satisfactory, and the project work also graded at least satisfacory. The final grade is the weighted average of the test grades and the project; the test with weight of 1-1, the project's grade with weight of 2.

Type of exam

None.

Type of replacement

At the last week, one of the two test can be replaced.

References

Obligatory:

- The slides from the lectures.

Recommended:

- ROS tutorial: http://wiki.ros.org/ROS/Tutorials
- M. Quigley et al., "ROS: an open-source Robot Operating System," in Proc. of the ICRA workshop on open source software, Kobe, Japan, 2009, vol. 3.