

Projects

Challenge levels and grades

Projects can be completed at three *Challenge levels*. The *Challenge level* determines the **best** grade that can be received to the project!

Challenge level	Best grade
Basic	3
Advanced	4
Epic	5

Tip

The projects are defined in a way that it is recommended to start with the **Basic** level, and then gradually work towards **Epic**.

The projects are graded based on the following aspects:

- Proved to be the student's own work
- Running results valid output
- Usage of versioning, usage of GitHub/GitLab/other repository
- Grading:
 - completeness of the solution
 - proper ROS communication
 - proper structure of the program
 - quality of implementation
 - documentation quality

Schedule

Week	Date	Event
7.	oct. 27	Announcement of project topics. Project lab I.
12.	dec. 1	Project lab II.
14.	dec. 15	Project presentations.

Grading

Personal attendance on the classes is mandatory (min 70%).

To pass the course, Tests and the Project must be passed (grade 2). One of the Test can be taken again.



Grade

$$\backslash(Jegy = (Test1 + Test2 + 2 \backslash times Project) / 4\backslash)$$

Project topics

1. Mobil robot

A. Playground Robot

- [Gazebo install](#)
- [Setting up a robot simulation \(Gazebo\)](#)



B. TurtleBot4

- [TurtleBot4 Simulator Tutorial](#)
- [TurtleBot4 GUI Docs](#)



C. PlatypOUs (ROS 1)

- [PlatypOUs GitHub](#)



D. Any mobile robot

1.1. Mobile robot obstacle avoidance

- **Basic:** Simulator setup, testing SLAM. Implementation of ROS node(s) to read the sensor data and move the robot.
- **Advanced:** Implementation of a ROS system to detect obstacle. Calculation and execution of a trajectory avoiding the obstacle in the simulator, using any sensor of the robot.
- **Epic:** Implementation and testing on the real robot/impress me!

1.2. Mobile robot path following

- **Basic:** Simulator setup. Implementation of ROS node(s) to read the sensor data and move the robot.
- **Advanced:** Implementation of a ROS system for path following in the simulator, using any sensor of the robot (e.g., driving next to the wall with given distance using LIDAR).
- **Epic:** Implementation and testing on the real robot/impress me!

1.3. Mobile robot object following

- **Basic:** Simulator setup. Implementation of ROS node(s) to read the sensor data and move the robot.
- **Advanced:** Implementation of a ROS system to detect an object and follow it in the simulator, using any sensor of the robot(e.g., visual servoing).
- **Epic:** Implementation and testing on the real robot/impress me!

1.4. Mobile robot action library

- **Basic:** Simulator setup. Implementation of ROS node(s) to read the sensor data and move the robot.
- **Advanced:** Implementation of a ROS action library containing simple actions and their execution (e.g., push object, move to object, turn around).
- **Epic:** Implementation and testing on the real robot/impress me!

2. Quadcopter

- [Gazebo install](#)
- [Setting up a robot simulation \(Gazebo\)](#)

```
ign gazebo -v 4 -r quadcopter.sdf
```



- **Basic:** Simulator setup. Implementation of ROS node(s) to read the sensor data and move the robot.
- **Advanced:** ROS system implementation to control velocity/position.
- **Epic:** Impress me!

3. Any Gazebo simulaion

- [Gazebo install](#)
- [Setting up a robot simulation \(Gazebo\)](#)
- [Gazebo World Examples](#)



Based on discussion.

4. Gazebo simulation creation

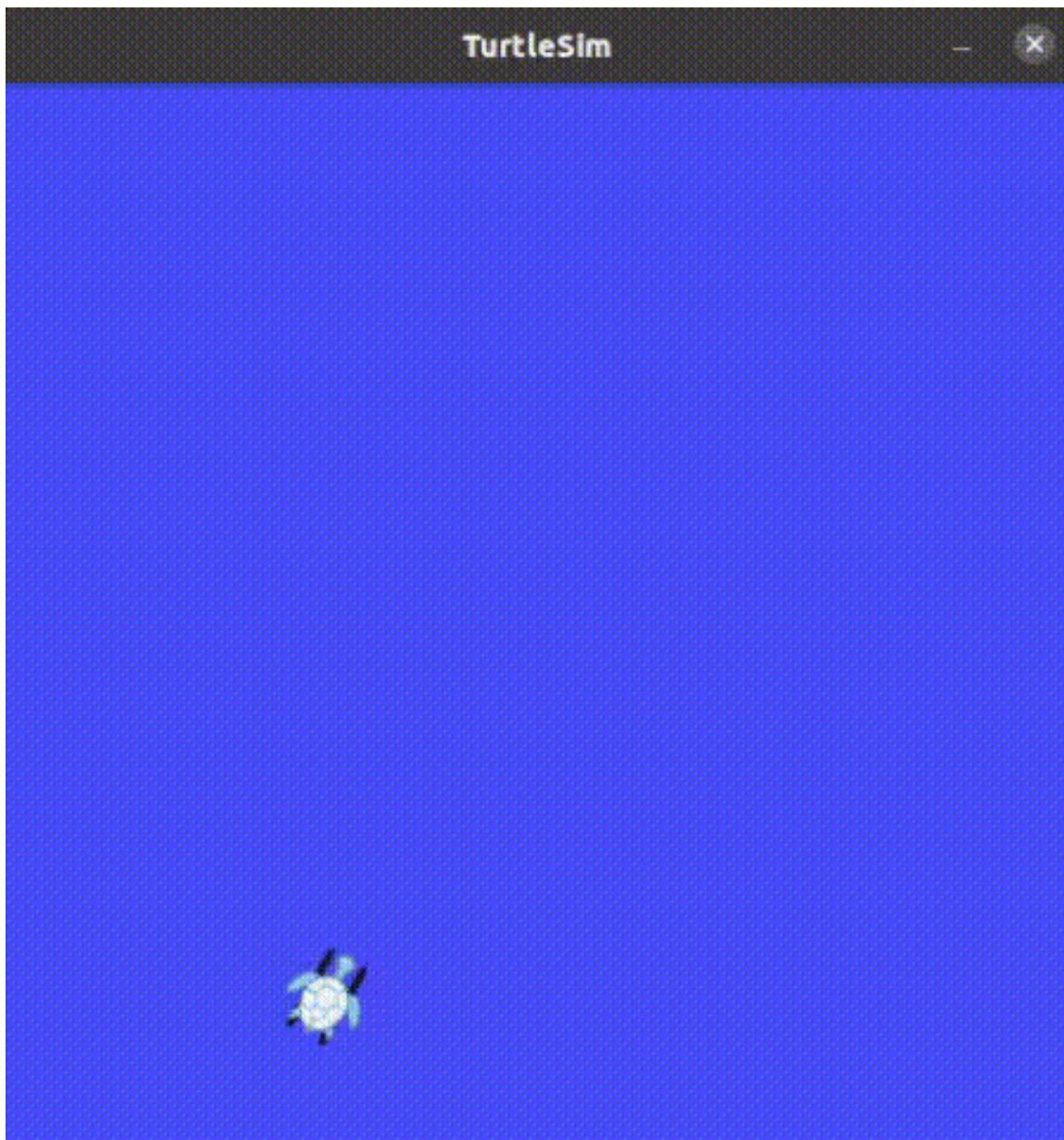
- [Gazebo install](#)
- [Setting up a robot simulation \(Gazebo\)](#)
- [Gazebo World Examples](#)



Based on discussion.

5. TurtleSim

- [Turtlesim Tutorial](#)
- [Koch snowflake](#)



5.1 Turtlesim Fraktál/Szöveg

- **Basic:** Implement a proportional controller.
- **Advanced:** Draw fractal/text.
- **Epic:** Impress me!

6. dVRK

- [Download and compile dVRK 2](#)
- [Marker examples](#)



6.1 DVRK Interactive Marker

Graspable, movable marker for the DVRK simulator.

7. YouBot (Windows)

||

- [YouBot controller GitHub](#)

7.1. YouBot ROS integration

- **Basic:** YouBot repo build.
- **Advanced:** ROS wrapper/interface implementation, move the simulated arm in joint space from ROS.
- **Epic:** Implementation and testing on the real robot/impress me!

X. Custom topic

Based on discussion.

Links

- [Gazebo install](#)
- [Setting up a robot simulation \(Gazebo\)](#)
- [Gazebo World Examples](#)
- [YouBot controller GitHub](#)
- [Download and compile dVRK 2](#)
- [Marker examples](#)
- [Turtlesim Tutorial](#)
- [Koch snowflake](#)