Project

Description

The project should be done using ROS2 and include self-created nodes (or at least a single node). The use of external packages is also possible (e.g. `usb_cam`) and recommended.

The goal of the project is to **create an interface for controlling the robot**. The example process should look as follows:

1. Launch a camera driver.
2. Start your node(s) to control the robot.
3. Launch the robot driver.
4. Demonstrate how the interface works.

The ROS package created for the project should be **published on GitHub**, with commits (more than one) from all project authors. The repository has to include a README.md file with short package documentation (how to install, run, use, etc.).

The **basic version** of the node (**grade = 3.0**):

- click on a point in the window, depending on whether the point pressed is above the center of the screen or below, the mobile robot (TurtleBot) moves forward or backward.

Possible ways to increase the grade:

- **+0.5**: using a **launch** file that runs all nodes at once automatically,
- **+0.5**: using **UR 5** robot instead of Turtlebot,
- **+0.5**: **dockerization** of the application (preparing a plug-and-play Docker setup for ROS with your package),
- **+0.5**: using **ArUco** to control the robot
  - in this option, you have to run a camera driver (laptop camera or external) or (if you don't have a camera) use data recorded with rosbag,
  - your node should process the camera image to find the pose of the ArUco tag (you can print it or display it on your smartphone),
  - depending on whether the center of the marker is above the center of the image or below, the robot moves forward or backward

Depending on the chosen robot, the control method changes. For the mobile robot, we publish commands on the `/cmd_vel` topic, while for the UR 5 robot, we publish commands on the `/scaled_joint_trajectory_controller/joint_trajectory` topic. For the manipulator, you can rotate the robot in the first axis (at the base) by a fixed angle, and for the mobile robot, move forward and backward.

Each group has to personally **demonstrate** how the interface works and upload a link to a GitHub repository.

Example demonstration (simplified version for UR 5 robot)