



← Project review - ROS. Day02

 Type of project	Individual
 Duration	30 min
 Passed Peer Reviews	0/2

Git project



ssh://git@repos-ssh.21-school.ru:2289/students/ROS._Day02.ID_568829/mjuli_python_ds/R...

Copy link

Open

Student



mjuli-python-ds

level 0

About



Introduction

The methodology of School 21 makes sense only if peer-to-peer reviews are done seriously. Please read all guidelines carefully before starting the review.

- Please, stay courteous, polite, respectful and constructive in all communications during this review.
- Highlight possible malfunctions of the work done by the person and take the time to discuss and debate it.

- Keep in mind that sometimes there can be differences in interpretation of the tasks and the scope of features. Please, stay open-minded to the vision of the other.
- If you have not finished the project yet, it is compulsory to read the entire instruction before starting the review.

Guidelines

- Evaluate only the files that are in src folder on the GIT repository of the student or group.
- Ensure to start reviewing a group project only when the team is present in full.
- Use special flags in the checklist to report, for example, an “empty work” if repository does not contain the work of the student (or group) in the src folder of the develop branch, or “cheat” in case of cheating or if the student (or group) are unable to explain their work at any time during review as well as if one of the points below is not met. However, except for cheating cases, you are encouraged to continue reviewing the project to identify the problems that caused the situation in order to avoid them at the next review.
- Doublecheck that the GIT repository is the one corresponding to the student or the group.
- Meticulously check that nothing malicious has been used to mislead you.
- In controversial cases, remember that the checklist determines only the general order of the check. The final decision on project evaluation remains with the reviewer.

Main part



Exercise 00 - Introduction to tf2.

- Does frames.pdf contain a tf tree?
- Does the transform.txt file contain the coordinate transformation between turtle1 and turtle2 frames?
- In the last_transform.txt file, are the X, Y offsets and the rotation angle in degrees along the Z axis correctly written out from the last coordinate transformation recorded in the transform.txt file?

Exercise 01 - Writing a tf2 broadcaster and listener.

- Does the package compile successfully?
- launch file starts successfully?
- After running the launch file, open rviz by loading the settings from the carrot.rviz file. Are the TF frames world, turtle1, turtle2, carrot visible in rviz? If you move the first turtle with the keyboard, do the TF frames in rviz move the same as in the simulator?
- Change the arguments in the launch file to radius=0.5 and direction_of_rotation=1. Run the launch file and control the first turtle from the keyboard. Does the second turtle follow the carrot frame rotating behind the first turtle at a distance of 0.5 and clockwise?
- Change the arguments in the launch file to radius=1.5 and direction_of_rotation=-1. Run the

e launch file and control the first turtle from the keyboard. Is the second turtle following the carrot frame rotating behind the first turtle at a distance of 1.5 and counterclockwise?

Exercise 02 - Time travel with tf2.

- Does the package compile successfully?
- launch file starts successfully?
- Change the argument in the launch file to delay=9. Run the launch file and control the first turtle from the keyboard. Is the second turtle following where the first turtle was 9 seconds ago?
- Change the argument in the launch file to delay=1. Run the launch file and control the first turtle from the keyboard. Is the second turtle following where the first turtle was 1 second ago?

Feedback



Fails

Comment

