F1TENTH Autonomous Racing **(Due Date:)**

Lab 1: Introduction to ROS

Instructor: INSTRUCTOR *Name: Karel Smejkal, StudentID:* ID

Course Policy: Read all the instructions below carefully before you start working on the assignment, and before you make a submission. All sources of material must be cited. The University Academic Code of Conduct will be strictly enforced.

1 Workspaces and Packages

1.1 Written Questions

1. CMakeLists.txt is the main input for CMake file to build the package and calls catkin-specific functions. Specifically it contains a set of instructions describing the project's source files and targets (header files, executable, ...). As mentioned it is related to make file used for C++ objects. Main difference is in the "C" - as cross-platform, so CMakeLists.txt works across different operating systems.
2. No, when working simply with only python node you don't have to compile CMakelists.txt, python works as executable script so you need to make sure that the file is executable (chmod +x). So this makes executable python object.
3. catkin_make needs to be run in the workspace directory for the specific software.
4. Sourcing the setup files helps to set environment variables. The first command set variables for ROS environment, while the second command set variables for the workspace where the source command is executed.

2 Publishers and Subscribers

2.1 Written Questions

1. Nodehandle is roscpp's interface for creating subscribers, publishers, etc and is used for writing nodes. Yes, there can be more than one nodehandle objects in single node
2. Node handle object in python tells the name of the node to rospy. It is initialized with rospy.init_node("node_name"). Rospy communicate with this name with the master and node has to be unique. Or you can set parameter anonymous=True for creating node with unique name (adding unique string of number behind the name) for you.

3. ros::spinOnce() and ros::spin() is used not kill the node immediately after one run using. As name suggest ros::spinOnce() is executed only once while ros::spin() keeps the node running until it is killed. This is achieved by consuming some CPU cycles.

4. ros::rate() creates a Rate object class which keeps track of how much time since last rate.sleep() was executed and sleeps just long enough to maintain desired rate specified during creating of Rate object.

5. callback in python is invoked in Subscriber each time message is received. rospy.spin() is used to keep the node from exiting and you need it to receive the callback.

3 Implementing Custom Messages

3.1 Written Questions

1. ?

2. Header of the message is stamped data and it is used to communicate timestamped data in a particular coordinate frame. There are 3 different data in the stamp (seq, stamp, frame_id) with data type uint32, time, string respectively. Yes, you cna include it in custom defined message.

4 Recording and Publishing Bag Files

4.1 Written Questions

1. The bagfile get saved in the dir you can the command rosbag record. You can change it by switching to diffrent directory.

2. When launching rosbag from launch file it will be defaultly saved in /.ros dir. To change to different dir you can pass argument to launch command with the flag "-o /dir"