Assignment 2:
Gazebo Environments and Turtle-Bot3

Due by 5.Feb.2024 23.59 IST

1. **Installation:** Install the Turtlebot packages and simulation on your computer or virtual machine (1) https://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/ You need to perform the PC setup and the installation of the turtlebot3_simulations catkin_ws. This step can be omitted on the lab PCs we provide.

2. **Loading Gazebo Simulation Environments (10):** Load the provided environment with the Turtlebot. This requires manipulation of launch and world files - found in catkin_ws/src/turtlebot3_simulation/turtlebot3_gazebo/launch and ../world folders. Tip: Should be just an issue of correct naming and calling. Load the world provided in: models/room1. Next, load the environment called: turtlebot3_house provided by the turtlebot3_simulations pkg. Task 7 should be performed in turtlebot3_house, the provided room1 is for debugging purposes for the robot skills as it loads faster -- and in case your pc hardware has considerable performance issues, you are allowed to use the provided map instead of turtlebot3_house for task 7.

3. **Localization and mapping (15):** Use the Turtlebot SLAM package to create a map of our environment - by teleoperation. Then export and save this map. Tip: Check out link (1), and follow the Turtlebot tutorials.

4. **Navigation (10) :** Use the map you created (load it when launching the navigation stack) and navigate through the environment

    1. Via rviz - clicking / publish 2d goal

    2. Via publishing in command line to ROS (move_base: simple move goal)

    3. Via publishing from python to ROS (requires having the python api rospy and a few other packages)

5. **Robot control (25):** Create five basic skills for the robot (see "turtlebot3_functions.py" for inspiration and also do the TODOs!)

    1. Navigate to a point using move_base - the function gets a goal: x, y, theta (i.e. 2D position and heading) and moves the robot from the current pose to the goal pose (you can use the existing topics that move_base exposes)

    2. Sense current robot pose - gps like, when it is called returns x,y and heading

    3. Sense objects - this function gets called and returns the location and name of each object within the map

    4. Pick object nearby - gets an object name and location and then picks it up (deletes the object, spawns it in the knapsack (outside the map))

    5. Place object nearby - gets an object name and a place location and then places the object (deletes the object from the knapsack, spawns it at the place location)

Note that you can do Task 5 in any language: py, cpp or directly via ROS Launch Files. We recommend python (also example code is given in py).

6. **ROS (10):** Wrap these skills in dedicated ROS services and launch them on corresponding nodes

7. **The Task (30):** Red balls are all over the room - and the Turtlebot needs to collect them. It can drive and navigate as much as it wants but it can only "carry" a single object at a time. It can sense the environment - its own position in it and the location of the objects it needs to interact with. There is a single point, where all the balls need to be collected at - the blue cube.

    1. You need to alter the environment_functions.py provided to you to randomly distribute 4 red balls and one blue cube in the turtlebot3_house (you can use the script directly or create a ROS service from it that sets up the environment)

    2. You need to add a function that checks if all red balls were collected correctly in the end (goal checker - which the robot can call)

    3. Use the robot skill services you created in order to solve this task - do this by creating a script (py, cpp or launch file) that calls the services (5) when needed (reactive or pre-planned approach are okay)