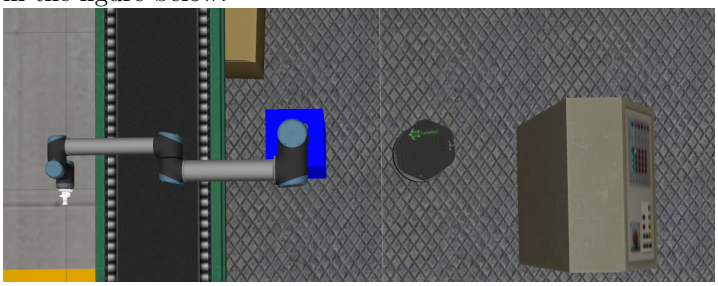
HRW ROS Assignment 3 Week 3 Part 1 (1)

The goal of this part of the assignment is to update two files: the ROS node implementation in `week3_assignment3_part1.py`, and the launch file to start this node, `week3_assignment3_part1.launch`.
The goal is to have the TurtleBot navigate to its first target location for the factory operation, as shown in the figure below:



You will work towards this goal using the following steps:

1. Terminate all the ROS nodes you started for Assignment2 with Ctrl+C in all the CCSs you opened so far. Remember, you do not have to close the CCS itself. This way, you will save yourself from having to source the setup files again in new CCSs.

2. Start the Factory simulation with:

   ```
   $ roslaunch hrwros_week3 hrwros_turtlebot_navigation.launch
   ```

3. Then, in a different CCS, start the `amcl_demo.launch` file with the correct arguments for the initial pose estimate of the TurtleBot using:

   ```
   $ roslaunch turtlebot_gazebo amcl_demo.launch
       map_file:=$HOME/hrwros_ws/src/hrwros_week3/config/map_factory_v1.yaml
       initial_pose_x:=<A> initial_pose_y:=<B>.
   ```

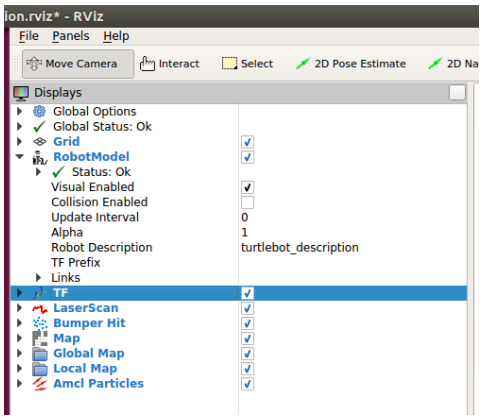
   Substitute A and B, with the correct values from the previous assignment.

4. Next, in another CCS, start the RViz visualization with:

   ```
   $ roslaunch hrwros_week3 hrwros_view_navigation.launch
   ```

   and follow the steps from Assignment 1 to change the robot description parameter to `"turtlebot_description"`. This way you can visualize the TurtleBot in the green cloud of arrows indicating its potential initial position.

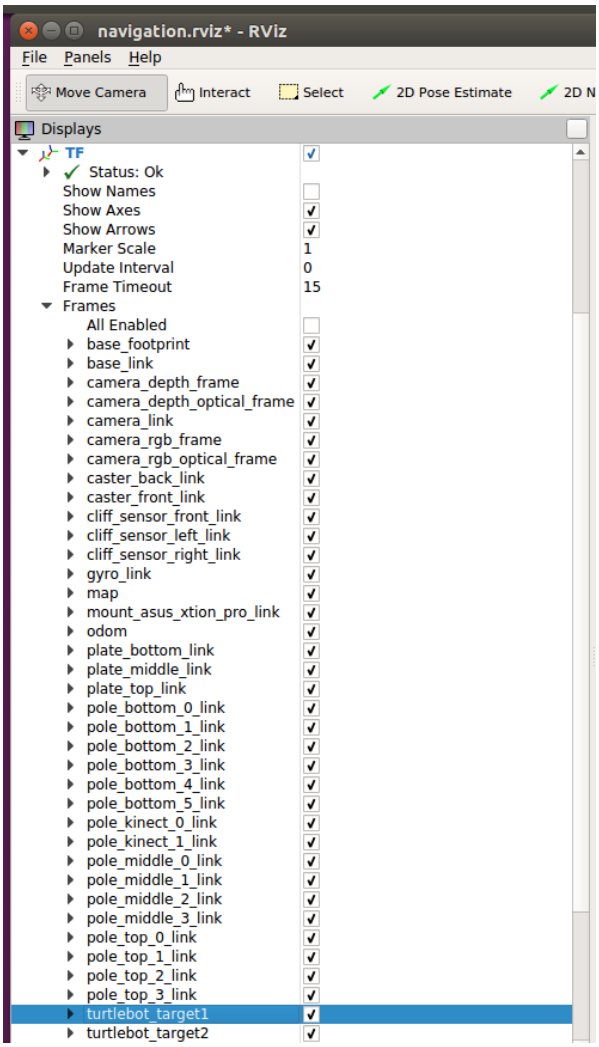
5. Now, in RViz, enable the TF display by clicking on the checkbox next to it like it is shown in the screenshot below:



6. Then, if you expand the TF display by clicking on the black triangle to the left of the letters TF, you will see several elements with a check mark against them under the category Frames. If you scroll down a bit, you will be able to see a frame called `turtlebot_target1` as shown in the figure on the next page.

This part continues on the next page

HRW ROS Assignment 3 Week 3 Part 1 (2)



Your tasks for completing this part are as follows:

1. Expand the `turtlebot_target1` frame display by clicking on the small triangle to its left. Read the different kinds of information present there and update the relevant information in `week3_assignment3_part1.py`. *(Hint: you only have to change two lines of code in this file)*

2. Update `week3_assignment3_part1.launch` so that we can actually start the ROS node that will move the TurtleBot to the first target location as shown at the beginning of this assignment. (You only have to add the node type)

3. In a new CCS, source your workspace setup files and launch the file you just edited with:

   ```
   $ roslaunch hrwros_week3 week3_assignment3_part1.launch
   ```

   If everything is good, when you launch this, the TurtleBot will autonomously navigate to the first target location, wait until the turtlebot arrives to its goal. It may take some time, and the turtlebot may struggle to reach it, if it fails, just re-launch it.

To complete the assignment, upload a screenshot of the CCS where the `week3_assignment3_part1.launch` was launched. It should show the following messages: `"Goal sent to move_base action server."`, `"Goal position x: <> y: <>"`, `"Hooray! Successfully reached the desired goal"`

This assignment continues on the next page

HRW ROS Assignment 3 Week 3 Part 2

In this part, the goal is to navigate the TurtleBot to its second target location, following similar steps as you followed in the first part, but with an added difficulty: To visualize and become aware of the "unknown obstacle avoidance" feature of the ROS autonomous navigation package which we have learned this week.
You will accomplish these goals with the following steps:

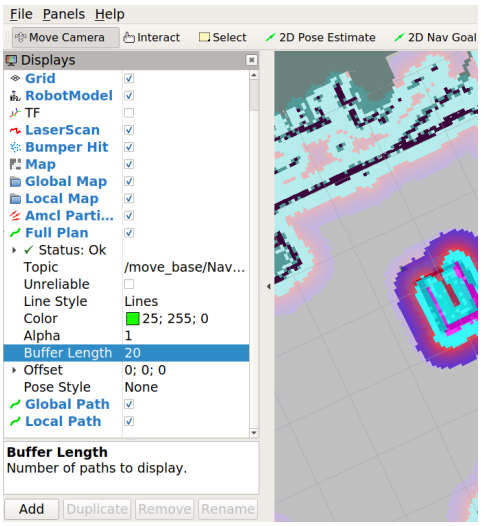
1. Following the same steps as in the Part 1, update the `week3_assignment3_part2.py` script with the second target location for the turtlebot.

   You will find the information under the `turtlebot_target2` frame, also in the TF Display in RViz. Once you have updated the relevant information in `week3_assignment3_part2.py`, uncheck the TF Display and collapse it.

2. Also following the same steps as in the Part 1, update the `week3_assignment3_part2.launch` with the corresponding type of node, but don't launch it yet.

3. Make sure you have all the Three Path Displays enabled, i.e., one for the overall global plan topic `/move_base/NavfnROS/plan`, one for the global plan of the Dynamic Window Approach (DWA) planner which is used for obstacle avoidance (topic: `/move_base/DWAPlannerROS/global_plan`) and one for the local plan of the DWA planner (topic: `/move_base/DWAPlannerROS/local_plan`).

4. You will upload a screenshot of RViz showing the modified global path around the obstacle, so you need to retain the global path by changing the "Buffer length" entry of the Full Path display to 20, as shown in the figure below:



5. Assuming you did not close or stop any ROS nodes from the previous part, you can start the `week3_assignment3_part2.launch` file in a new CCS with:

   ```
   $ roslaunch hrwros_week3 week3_assignment3_part2.launch
   ```

   If you did close or terminate the previous part, then make sure that the TurtleBot is at the first target location (following the steps in Part 1 of this assignment), and then start the `week3_assignment3_part2.launch` file.

6. A couple of seconds after the robot starts moving, you will notice that two new obstacles pop on Gazebo. These obstacles were not known to the global planner at the time of planning. So the initial path might actually collide with or go very close to the obstacles, however, as the TurtleBot approaches (one or both of) these obstacles, the global path gets modified.

Upload a screenshot clearly showing the TurtleBot near the obstacles, and the changes on Full Path as it avoids the obstacles.

This completes HRW ROS Assignment 3 Week 3