In the previous lessons, you have learned a lot about navigation in ROS. Now it's time to put all you have learned into practice. Following the tutorial, you will be guided to make the turtle bot navigate autonomously. Good luck and have fun.

Path planning tutorial

In previous tutorials, we have seen how to create a map and localize the robot properly in the environment. In this section we will take it a step further, and make the robot navigates to a goal position autonomously.

First of all, let's open the TurtleBot in Gazebo. In the **first CCS** enter:

```
$ roslaunch turtlebot_gazebo turtlebot_world.launch
```

Right now, we will need to open up the nodes responsible for localization and navigation, on a **second CCS** execute:

```
$ roslaunch turtlebot_gazebo amcl_demo.launch
```

The `amcl` stands for Adaptive Monte Carlo Localization. This is a probabilistic localization system for a robot moving in 2D which uses a particle filter to track the pose of a robot against a known map (the default one is none is specified). This launcher also starts the `move_base` node, which is responsible for planning and controlling the movements of the robot.

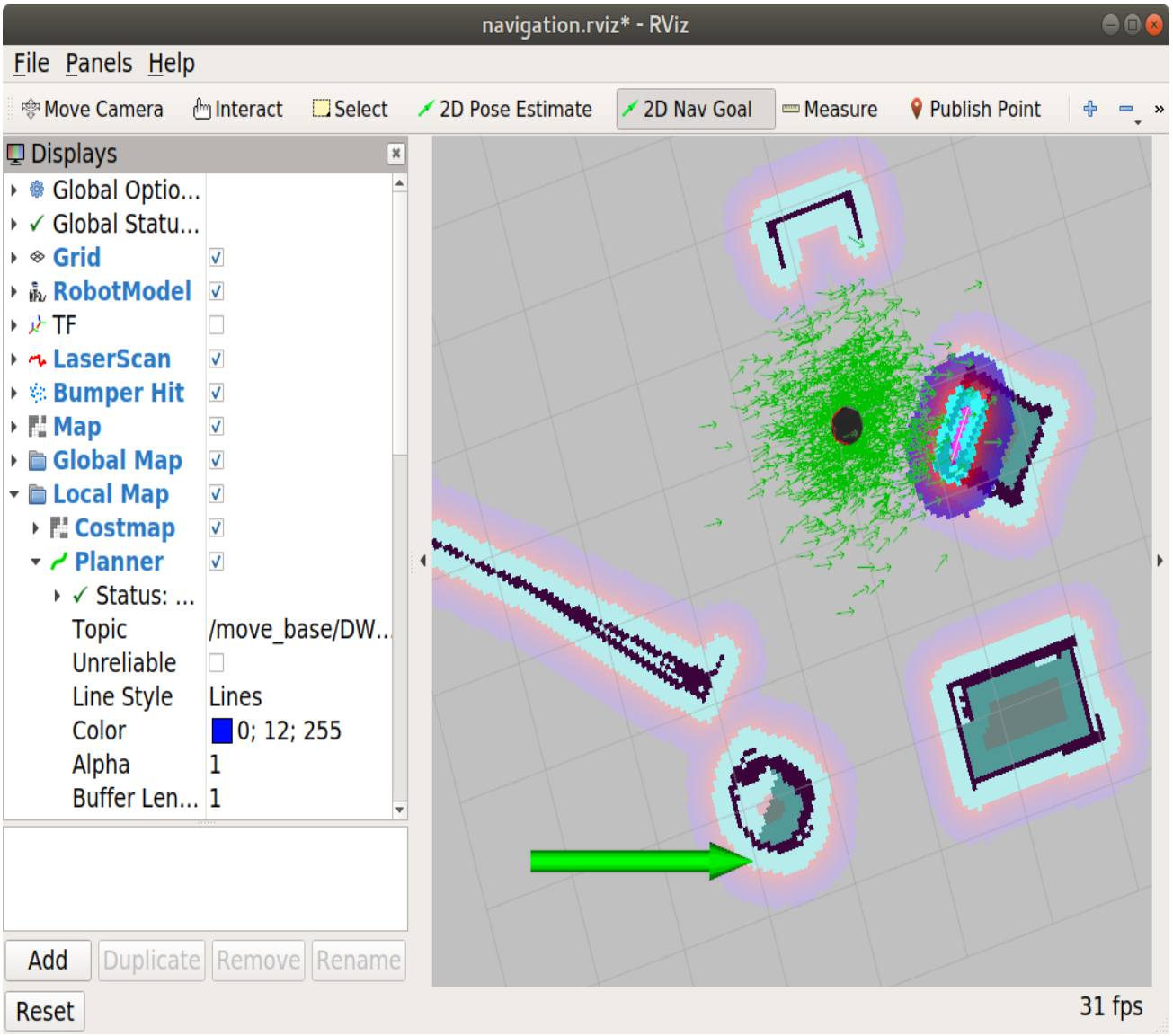
Finally in a **third CCS** we will open up RViz to visualize the navigation.

```
$ roslaunch turtlebot_rviz_launchers view_navigation.launch
```

Right now we are ready to go.

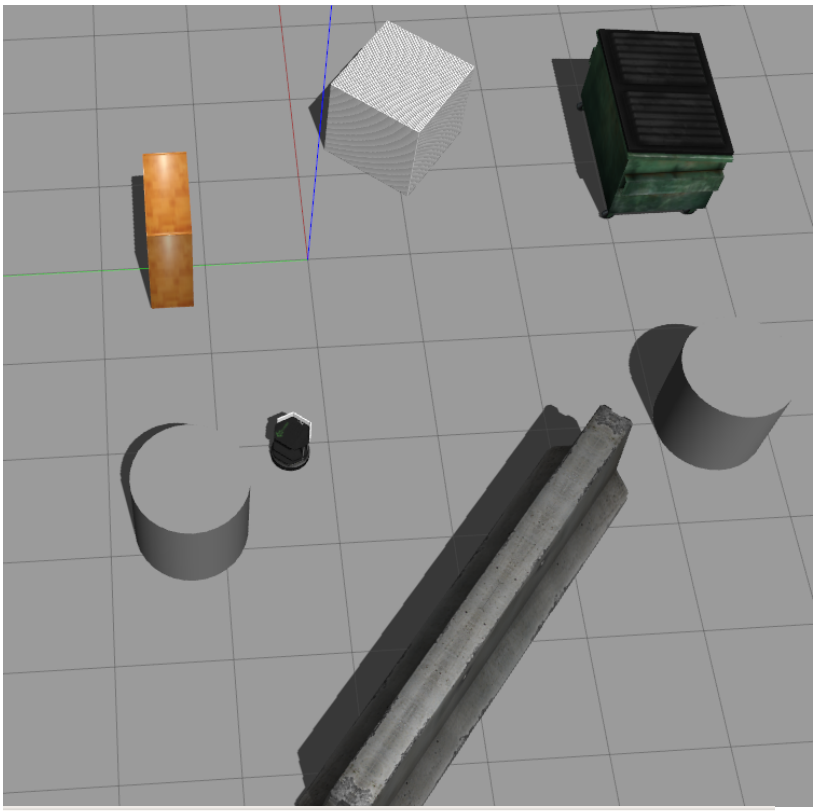
Use the *2D post estimate* button on the top of your RViz screen if you want to change the initial localization of the robot on the map.

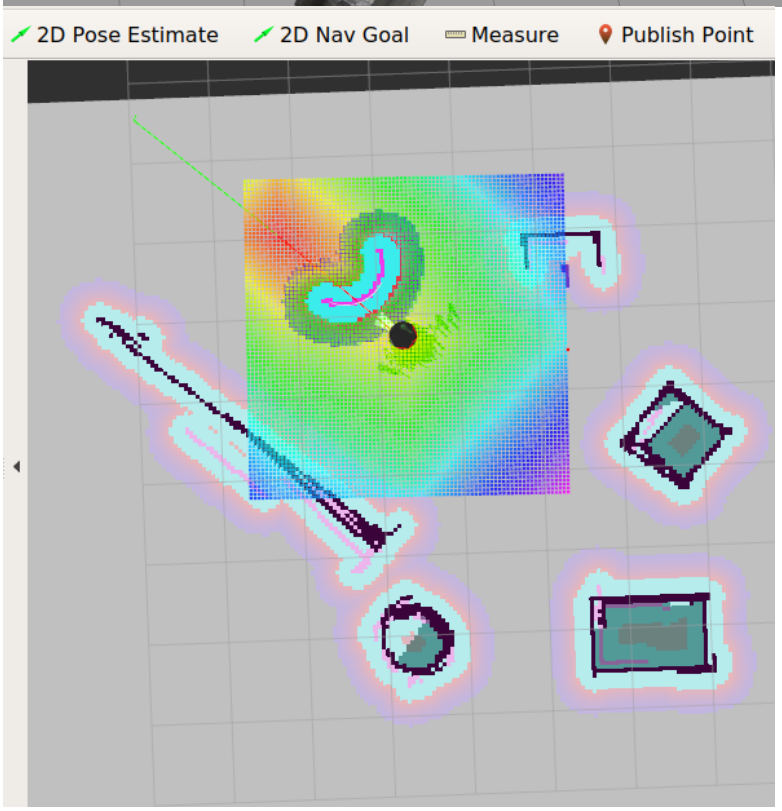
Afterwards, use the *2D Nav goal* to give the robot a target position to move to. Then, watch the TurtleBot as it generates a path and tries to follow it.



Now, to test the unknown obstacle avoidance functionality. Give the robot a target position far away. When you see the generated map, add an extra obstacle in *Gazebo*. This can be easily done using the top menu.

Watch how the TurtleBot maneuvers around the obstacle then gets back to the original path. This is done using the *local planner*.




2D Pose Estimate
2D Nav Goal
Measure
Publish Point