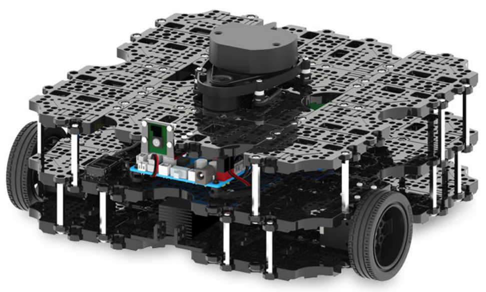
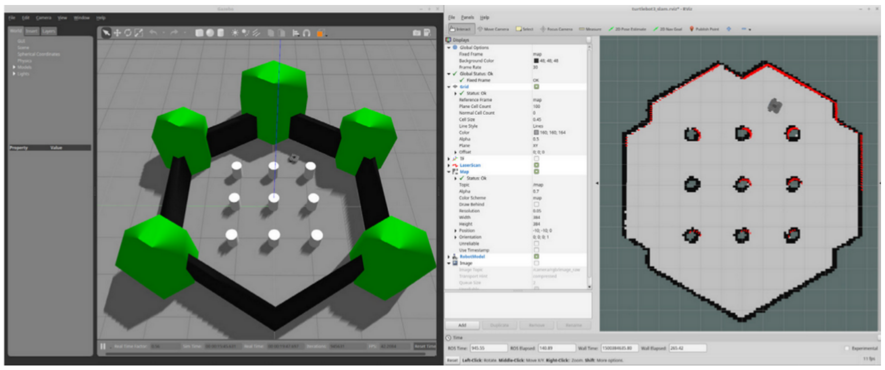
ROS - Navigation and The Turtlebot3 Simulator

ME 4140 - Introduction to Robotics - Fall 2020

What do we mean by navigation? This means different things in different places. Here, we mean the navigation stack in ROS melodic. This tutorial comes from here.

Overview:

After completing *Tutorial 5 - Turtlebot Simulator*, You have learned some basics of ROS, and you have a for a more advanced robot. Next you are going to learn to use the navigation stack with the turtlebot3 simulator. Read more here and here.

System Requirements:

- **ROS+OS**: This tutorial is intended for a system with ROS Melodic installed on the Ubuntu 18.04 LTS operating system. Alternate versions of ROS (i.e. - Kinetic, Noetic, etc.) may work but have not been tested. Versions of ROS are tied to versions of Ubuntu.
- **ROS:** Your computer must be connected to the internet to proceed. Update the system before you begin.
- **Workspace Setup:** The Turtlebot3 Simulator from tutorial 5 must be operational before completing tutorial 6.

Disclaimer:

- **Backup the System:** If you are using a virtual machine, it is recommend to make a snaphot of your virtual machine before you start each module. In the event of an untraceable error, you can restore to a previous snapshot.
- ROSLAUNCH: This tutorial involves using the roslaunch command which runs a muliple of nodes at once as described in the launch file. We will learn more about this later.
- Mouse for 3D viewing: This simulator view is much easier use if you have a three button mouse plugged in, but this is not required.

Part 1 - Install navigation and gmapping packges:

1. **Install the navigation and gmapping nodes if you have not already.**

```
sudo apt install ros-melodic-navigation ros-melodic-gmapping
```

2. **Set the robot model. This only needs to be done once. Modify the .bashrc file If you want to change models.**

```
echo "export TURTLEBOT3_MODEL=waffle_pi" >> ~/.bashrc
source ~/.bashrc
```

Part 2 - Generate a map of the virtual space:

1. **Start the turtlebot3 simulator.**

```
roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

2. **Next, start SLAM using the gmapping node.**

```
roslaunch turtlebot3_slam turtlebot3_slam.launch slam_methods:=gmapping
```

3. **Drive the robot around with the keyboard to collect pose and Lidar data**

```
roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

4. **When you are finished save the map. (-f allows the filename to be set)**

```
rosrun map_server map_saver -f map
```

Two map files (.pgm and .yaml) will appear in the current folder after Step 4. If you move the map to a new directory, keep both files together.

Part 3 - Navigate the virtual space using the map and RVIZ:

Now that navigation is installed and there is a map saved to file, the robot can perform autonomous point to point navigation with dynamic obstacle avoidance.

1. **Start the turtlebot3 simulator.**

```
roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

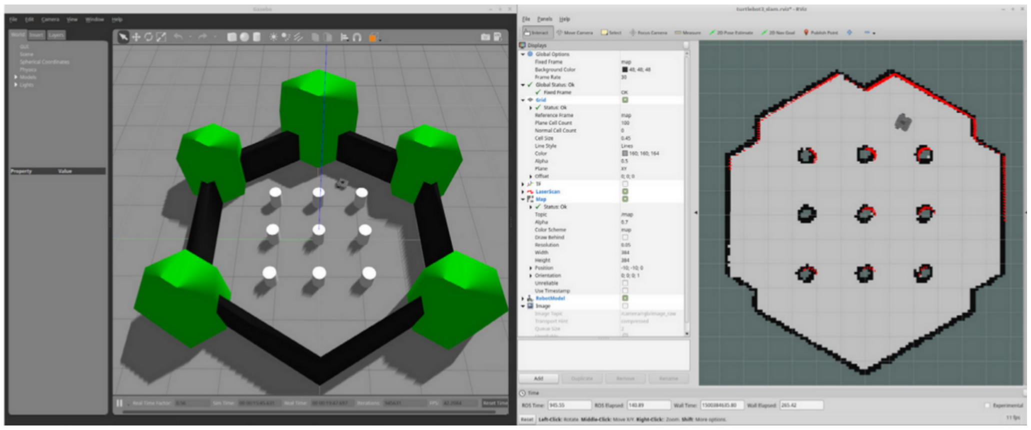
2. **Turn on the navigation nodes and RVIZ. Use the name of the map you created.**

```
roslaunch turtlebot3_navigation turtlebot3_navigation.launch map_file:=map.yaml
```

The gazebo window will open containing your robot, and you will also see the rviz window open separately. Find and test the following features of navigation in RVIZ.

1) Pose Estimate - Click and drag the direction to set the current pose estimate of the robot.

2) 2D Nav Goal - Click and drag the direction to define a goal point for the robot in the map.



3. **Check the available topics with rostopic.**

```
rostopic list
```

```
rosrun rqt_graph rqt_graph
```

Part 4 - Issues Loading Maps Files:

You may have run into an issue in which *turtlebot3_navigation* cannot load the map file. A typical error message is shown below, along with with the preferred solution.

```
[ERROR] [1604667817.760623311]: Map server could not open /map.yaml.
```

- Leave the top line of *map.yaml* as is when the file is generated. Do not add the full path to map file. The filename is sufficient.
- Copy *map.yaml* and *map.pgm* to the maps/ directory of a package in the ROS workspace.
- Ensure that the workspace will compile and build with *catkin_make*. You should see in the terminal output, that your package was successfully built.
- Finally, use the *find* command to point the node to the maps directory of your package.

Now repeat the commands from before with the addition of the find command in the map argument.

1. **Start the turtlebot3 simulator.**

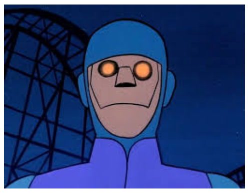
```
roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

2. **Turn on the navigation nodes and RVIZ. Use the name of the map you created.**

```
roslaunch turtlebot3_navigation turtlebot3_navigation.launch \
map_file:='$(find <YOUR PKG>)\maps\<YOUR MAP>.yaml'
```

Tutorial Complete:

After completing *Tutorial 6 - Turtlebot3 Simulator*, you are ready to learn about ... more ROS!



NOW, YOU KNOW ABOUT ROS! GOOD JOB!