Localization - Knowing where the robot is on a map

In this first assignment, we will work on helping the robot to find it's own position on a map. So we will use the Adaptive Monte Carlo Localization (AMCL) to help localize the robot in our factory floor.

This first assignment is divided in three parts:

- Visualize the robot in the factory floor map.
- Provide a good estimation of the robot's initial position.
- Improve the quality of the localization with the AMCL algorithm.

Week 3 - Assignment 1 - part 1 (of 3) --- 1 Point

In this first is very short, you will prepare for navigation in our factory world by visualizing the robot on the factory floor.

To do this you need to modify the configuration of RVIZ to visualize the robot, You can do this with the following steps:

Step 1: Launch the hrwros factory simulation with:

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

Note: *This might not succeed at the first attempt or it may take a while to start.*

The factory environment has quite a lot of graphics to be rendered on Gazebo and depending on the processing power on your computer, it might take a few tries before you can see everything like it is shown in the picture of the previous unit. For example, you might see that the Turtlebot may not show up after multiple launch attempts. But, this is expected behavior. Normally, it should not require more than 5 re-starts.

Step 2: Launch AMCL with a map of the factory we created:

```
$ roslaunch hrwros_week3_assignment
amcl_navigation.launch
map_file:=$HOME/hrwros_ws/src/hrwros_week3/config/ma
p_factory_v1.yaml
```

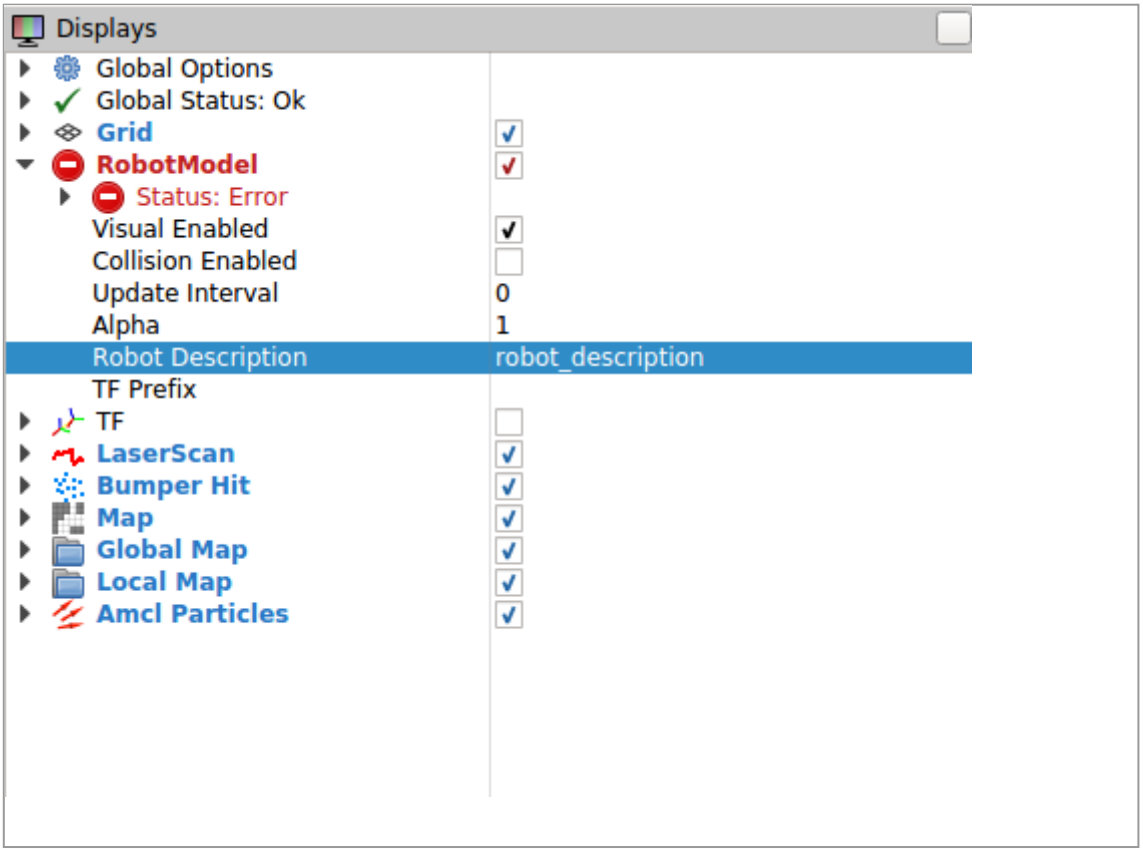
Step 3: Next, start the RViz navigation visualization:

```
$ roslaunch hrwros_week3_assignment
view_navigation.launch
```

In RViz, you will see there are red mark in the `RobotModel` display tab to the left. Let's fix this!

It appeared in red because RViz is missing the `Robot Description` parameter. This was to be expected, because there are actually three robots present in the factory!

So let's tell RViz which one is the TurtleBot.



Step 4: Change the `Robot Description` field to `turtlebot_description`.

You should now see a green cloud of arrows where AMCL thinks the TurtleBot is.

This position estimation clearly is incorrect! However, we will solve that in Part 2 of this assignment.

Remember to save the RVIZ configuration with `File -> Save config` or `Ctrl+S`

Week 3 - Assignment 1 - part 2 (of 3) --- 2 Points

In this part, you will define the correct initial position of the robot in the map.

At the end of the first part, we noted that the pose of the TurtleBot in RViz is incorrect, it's further away from the obstacles in Gazebo than in RViz (it looks like it is in front of the conveyor belt).

This is because when we launch AMCL, it assumes a default initial pose at the origin, but the TurtleBot is actually spawned in a different position in Gazebo and the localization can't correct the pose estimation from that initial pose.

We can fix this by giving AMCL a better estimation of the initial pose, which can be obtained from the TurtleBot's position on Gazebo. The pose can be given, by passing two arguments to the `amcl_navigation.launch`:

- `initial_pose_x`
- `initial_pose_y`

The assignment is as follows:

Step 1: Find the pose of the TurtleBot in Gazebo from the World tab.

> - In the Hands-on practice 1 of the Recap module of this Week, you learned how to do this.

Step 2: Use the relevant coordinates from that pose to tell AMCL where the initial pose is using the command below.

> *Make sure you need terminate the* `amcl_navigation.launch` *launched previously (if it was still running).*
>
> ```
> $ roslaunch hrwros_week3_assignment amcl_navigation.launch map_file:=$HOME/hrwros_ws/src/hrwros_week3/config/map_factory_v1.yaml initial_pose_x:=<XCOORD> initial_pose_y:=<YCOORD>
> ```
>
> You should now see the 'green arrow cloud' and the TurtleBot shifted to the correct location.

Step 3: Now that you have found the correct initial position, you can set those changes on the `amcl_navigation.launch` file.

> Edit the relevant parts of the `amcl_navigation.launch` to change the default arguments `<arg>` tags.
>
> - Set the correct `map` and `initial pose` values.

Step 4: Relaunch the AMCL localization without the parameters to check that they have been set correctly.

```
$ roslaunch hrwros_week3_assignment
amcl_navigation.launch
```

Keep both Gazebo and RViz simulations running as you move on to the next part.

Week 3 - Assignment 1 - part 3 (of 3) --- 0 Points

Now that everything is fixed, that is, RViz knows the robot description, and AMCL also has a good estimation of the initial position of the TurtleBot, let's take it for a spin!

We will use the teleoperation for this, open another CCS and launch the teleoperation (as we did on the first part of this week).

```
$ roslaunch turtlebot_teleop keyboard_teleop.launch
```

Move the turtlebot around. You should see that, as the robot moves, the cloud of green arrows around it becomes less disperse and more dense centered in the robot. This means that the AMCL is improving the robot's estimation of its position.

This completes Assignment 1. Keep the Gazebo and RViz simulations, and the amcl launch running.