TP2:

Controlling a Turtle (bis)

All the ROS environment is already setup into a Docker container. To start it, enter the following command line:

```
./run.sh
```

Run the roscore and turtlesim node (see TP1).

Forward then turn

In this exercice you have to make the turtle move following a square:

- go forward for 2 meters
- turn by pi/2
- go forward for 2 meters
- turn by pi/2
- go forward for 2 meters
- turn by pi/2
- go forward for 2 meters
- turn by pi/2

To reset the turtle position (if required) you can use the rosservice /reset (from the command line):

```
rosservice call /reset
```

Tips:

1. Define a subscriber that will get the pose of the turtle and save it to a variable (for example a global variable)
2. Define a publisher that will be used to send control command to the turtle
3. Define a function that will turn the robot until a specific angle is reached
4. Define a function that will move the robot will move the robot on a straight line based on a distance
5. Use the functions defined

The idea here is to get familiar with publishers/subscribers so we do not require that the turtle follow a perfect square.

```
import rospy
from turtlesim.msg import Pose  # import the turtlesim Pose message type
from geometry_msgs.msg import Twist   # import the Twist message type
import math

COMPLETE CODE HERE
```

Turn and move at the same time (more difficult)

Now you have to write a go to function that allows the turtle to go to a destination (X,Y) but doing the move and rotation at the same time.

You can use the following control law:

$$v = k_v * \sqrt{(X - cur_x)^2 + (Y - cur_y)^2}$$

$$\theta_{target} = tan^{-1}(y - cur_y)/(x - cur_x)$$

$$\theta_{velocity} = k_t * (\theta_{target} - \theta_{robot})$$

You can use different value for kv and kt, e.g kv=0.1, kt=1. You could also limit the min velocity that you send to the turtle, for instance a velocity bellow 0.1 meter/sec is maybee too slow.

For more information see : https://www.youtube.com/watch?v=Qh15Nol5htM&feature=youtu.be&list=PLSzYQGCXRW1HLWHdJ7ehZPA-nn7R9UKPa

```
import math

##############
#COMPLETE HERE

################

move_to(1, 1)
```

ROS TOOLS (RVIZ and Gazebo)

Launch the turtle bot gazebo with corridor world:

TURTLEBOT_GAZEBO_WORLD_FILE=/opt/ros/kinetic/share/turtlebot_gazebo/worlds/corridor.world roslaunch turtlebot_gazebo turtlebot_world.launch

Start rviz and visualize the robot and it's sensors.