

Assignment1 supplement material

1. People having a hard time to understand the catkin workspace

a. https://wiki.ros.org/catkin/conceptual_overview

- i. What is catkin: 'catkin' is the main build-system for ROS that is used to maintain workspaces.
- ii. This is what the name catkin comes from.



- iii. Everything related to the package or script registration to the ros needs to be recompile.
 - iv. After register the script change do not need to recompile again.
 - v. Using `CMakeList.txt` to do so, please read clearly on the guidance and tutorial on the website.
2. People have hard time understanding how to use talker talk to the turtle simulator

Talker: node → `turtle1/cmd_vel` : topic → turtlesim: node

- How do we create correct message to send to the connect topic?
 - Cmd: `rostopic list` : you will find the topic of `turtle1/cmd_vel`
 - Cmd: `rostopic info /turtle1/cmd_vel` : you will find the detail of what is this topic consuming

Type: geometry_msgs/Twist

Publishers: None

Subscribers:

* /turtlesim (<http://tams86:42129/>)

- Cmd: `rosmmsg show geometry_msgs/Twist` : This would show the detail of message you would like to use for talker to talk to the turtlesim

geometry_msgs/Vector3 linear

float64 x

float64 y

float64 z

geometry_msgs/Vector3 angular

float64 x

float64 y

float64 z

- To use the message please import it, you can find the path through following steps:
 1. `ls /opt/ros/noetic/lib/python3/dist-packages/ | grep msg` : List all kinds of messages
 2. Find the `geometry_msgs` and going into the directory you can see three items

`init .py msg pycache` and you want to find the usable `.py` file so you go further down.

```
ls /opt/ros/noetic/lib/python3/dist-packages/geometry_msgs/msg/
```

now you see the `_Twist.py`, the first letter "_" is to let python know that this `.py` file is only internal using not external.

So finally you can do `from geometry_msgs.msg import Twist` , and since each message type is a class you can init like `twist_msg = Twist()` and you can specify the number you want as `twist_msg.linear.x = 2.0` or `twist_msg.angular.z = 2`

Now you want to publish the message to the topic we talked before as `turtle1/cmd_vel` so we first use rospy to init node and publisher to the topic

with specific message type then **the most importantly do the `time.sleep(2)`, which would allow time for the publisher to set up.**

```
rospy.init_node('haus_vom_nikolaus', anonymous=True)
pub = rospy.Publisher('/turtle1/cmd_vel', Twist,
    queue_size=10)
time.sleep(2) # Allow time for the publisher to set up
```

Then you should be able to see the turtle move.

Hint: is also grateful to sleep a bit after each time you publish something. `time.sleep(duration)`