87/100

Robot Design Lab
Tutor: Teena Hassan, Mihaela Popescu

WiSe 2022/2023
Group: 09

Maarten Behn
Joel Köper
Nhu Van Nguyen

Worksheet 01: Ubuntu and ROS 2

Hand-in date: November 1, 2022

1 Applications of Robotics

6 Points

11 Points

1. Name four application fields of robotics and describe for each of them a task that robots have to do. Which UN SDG goals are addressed by each of these applications?

   (a) Healthcare and therapy

   Robots can be a patient and reliable partner in the process of recovery. The can assist in regaining mobility by providing support in various physical exercises. Socially assistive robots can help in autism therapy.

   Addressed SDGs: 3 - Good Health

   (b) Industry

   Even nowadays Robots perform many dangerous or laborious tasks in industrial systems. Robots can work in extreme conditions and with far less errors than humans ever would. For example a robots work on a production line of car manufacturer.

   Addressed SDGs:

   8 - Good Job and economic growth
   9 - Innovation and infrastructure

   (c) Search and rescue

   Robots like Spot from Boston dynamics can traverse dangerous or extreme environments to search for victims of for example avalanches. They can be equipped with assistive sensors like thermal image cameras to aid in the search.

   Addressed SDGs:

   3 - Good Health
   16 - Peace an Justice

   (d) Research

   Robots can measures for example climate data to aid in the prediction of climate events. The quality of the seas or air can easy and much more reliable measured by robots.

   Addressed SDGs:

   13 - Climate Action
   14 - Life below water
   15 - Life on land

2. Briefly explain three developments to which the Shakey robot project contributed. 3 Points

   (a) The A* search algorithm is a search algorithm to find a short path from point A to B in a given map. It uses weighted graphs to traverse the area and find path with the smallest cost. This is done by maintaining a tree of paths originating at the start node and extending those paths one edge at a time until the end node is reached.

(b) STRIPS planner The Stanford Research Institute Problem Solver is an automatic planner developed by Richard Fikes and Nils Nilsson in 1971. The name STRIPS was later used to refer to the formal language that served as input to the planner and today provides the basis for describing most problem domains.

(c) Hough transform The Hough transformation is a robust global method for recognizing straight lines, circles or any other parameterizable geometric figures in a black-and-white image, after edge recognition. The method was patented by Paul Hough in 1962.

1 Point

3. Distinguish between teleoperation and full autonomy and give an example for each case.

To put it simply, teleoperation refers to remote work. A teleoperator is a technical device that is controlled remotely from a human control room. If this device is capable of autonomous work, it is called a telerobot. If it is fully autonomous, it is called a robot.

An standard RC car is teleoperated, where the mars rover curiosity is mostly autonomous.

4. Do you have a dream application for a robot that you would like to build yourselves in the future?

1 Point

I personally think that drones are very interesting, especially when they operate in swarm like formation. For example drone light shows did always fascinate me.

18.5 Points

2 Getting Started with Ubuntu

9 Points

1. Open a new terminal window, try out the following commands with suitable parameters of your choice and describe briefly what each of them does. (10%)

(a) pwd = print working directory

A: **Prints out the current shell working directory.**

(b) ls = list segment

A: **Lists the contents of the current directory.**

(c) cd = change directory

-1

A: **Allows to move files into different folders.**

(d) mkdir = make directory

A: **Creates an empty folder.**

(e) touch

A: **Has multiple purposes like generating files or sets of files, altering the modifications or access times of files etc.**

(f) echo

A: **Is used to display line of text/string that are passed as an argument.**

(g) cat = concatenate

A: **It can read, concatenate, and write file contents to the standard output.**

It can also view content of a file and redirect output in terminal or files.

(h) cp = copy

A: **Allows to copy files or directories.**

(i) mv = move

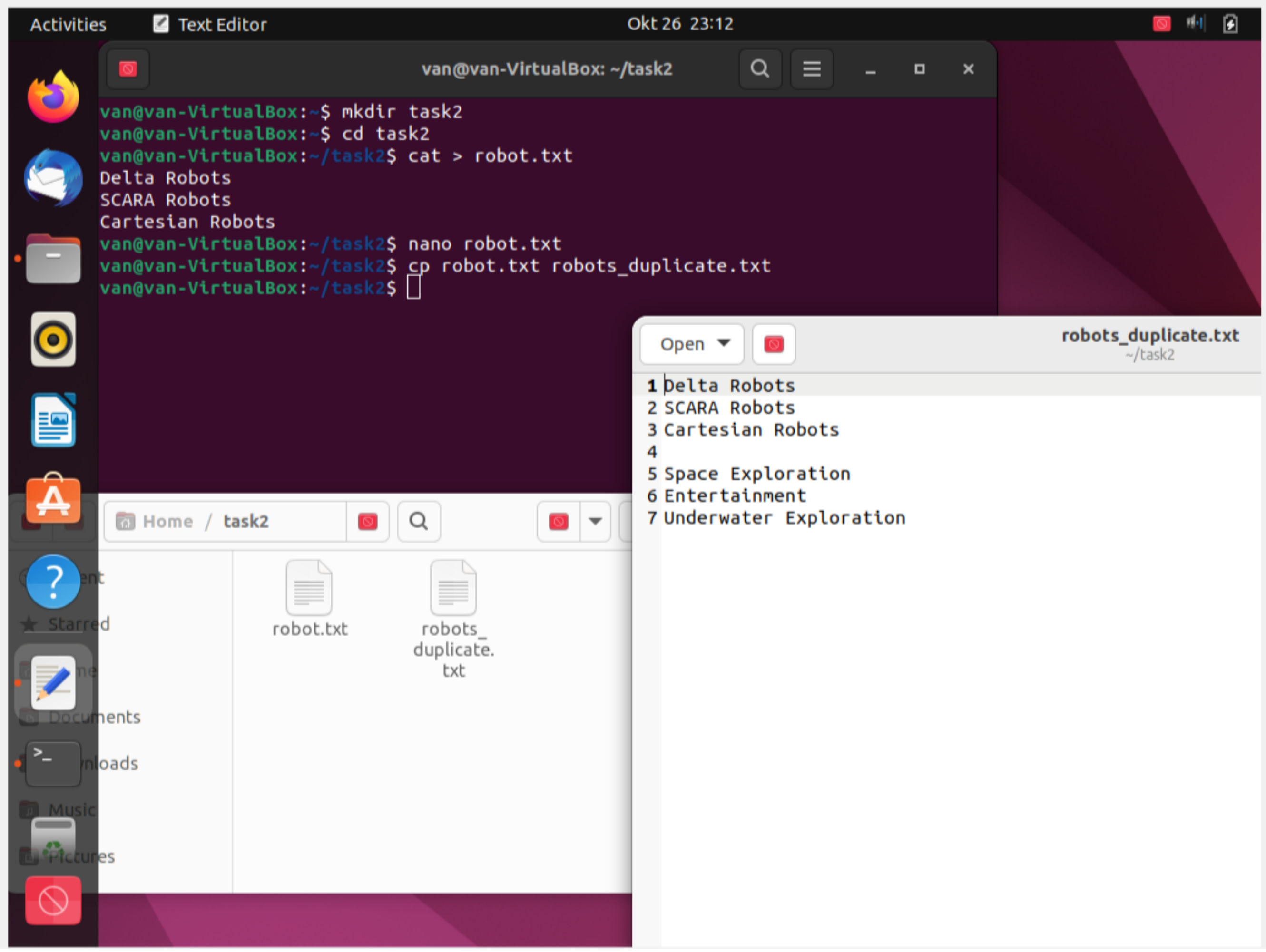
A: **The mv command moves or renames files and folders.**

(j) rm = remove

A: **Allows us to delete files.**

6 Points

2. Submit a screenshot of the terminal window with the used commands and the folder task2.



```
van@van-VirtualBox:~$ mkdir task2
van@van-VirtualBox:~$ cd task2
van@van-VirtualBox:~/task2$ cat > robot.txt
Delta Robots
SCARA Robots
Cartesian Robots
van@van-VirtualBox:~/task2$ nano robot.txt
van@van-VirtualBox:~/task2$ cp robot.txt robots_duplicate.txt
van@van-VirtualBox:~/task2$
```

```
Delta Robots
SCARA Robots
Cartesian Robots

Space Exploration
Entertainment
Underwater Exploration
```

3. What is root permission in Ubuntu and how can you execute a command with root permission? 2 points

A: The root user is the user which has administrative access to your system (Normal users do not have this access for security reasons) You can execute a command with root permission, when you type "sudo -i" at the command line

4. Which commands can you use to verify: 1.5 Points

(a) How much memory is free?

The following command using **cat** contains real-time information about the system's memory usage

```
cat /proc/meminfo
```

You can also use the **free** to view the amount of free and used memory in the system.

```
free -m
```

(b) The memory load of each running process?

A: ps (only listened in %)

```
ps -o pid,user,\%mem,command ax | sort -b -k3 -r
```

A: sudo

```
sudo pmap -r
```

12.5%

3 Quiz on ROS2

ros2 run launches an executable from a package and ros2 launch runs a launch file v

1.5% 1. What is the difference between ros2 run and ros2 launch? The answer is not very precise

The *ros2 launch* command launches a launch-file written in python (or XML), for example launching ros2 itself. The *ros2 run* command executes a command, script or executable.

0% 2. What does the command ros2 interface show std msgs/String do?

*ros2 interface show std_msgs/**msg**/String* displays a *message* from the module *std_msgs* of the type *String.* This command shows the structure of data in the message type String.

2% 3. How can you verify that a ROS2 node is subscribing to a topic?

ros2 node info ⟨node-name⟩

2% 4. Name two communication paradigms used in ROS2 to send and receive information between nodes.

Messages used within the *topic* architecture and *requests* as part *services.*

3% 5. Name three differences between ROS and ROS2.

Ros2 has

C ?

- A unified API for c, c++ and python
- Launch-scripts written in python rather than XML
- Asynchronous *services*

2% 6. Your robot has a camera that sends a video stream. Would you use a 'Service' or a 'Topic' to provide the video to a human that controls the robot with a joypad? Justify your answer.

When trying to get a flow of data like camera output it is advised to use a *Topic* due to its streamed nature which enables real-time data transmission without copious amounts of requests to a *service.*

2% 7. Your robot is moving towards location G. As it is approaching location G, you want it to take a picture. Would you use a 'Service' or a 'Topic' to obtain this picture? Justify your answer

When trying to do a one-time task a *service* would be the preferred method because it is less expensive in terms of bandwidth, compute and memory to request one image then picking the correct one out of an image stream.

than

37%

4 Write ROS2 Nodes to Operate on Numbers

5%

1. What is the type of data that is published and what is the name of the topic on which it is published?

   Type: String

   Topic: topic

1%

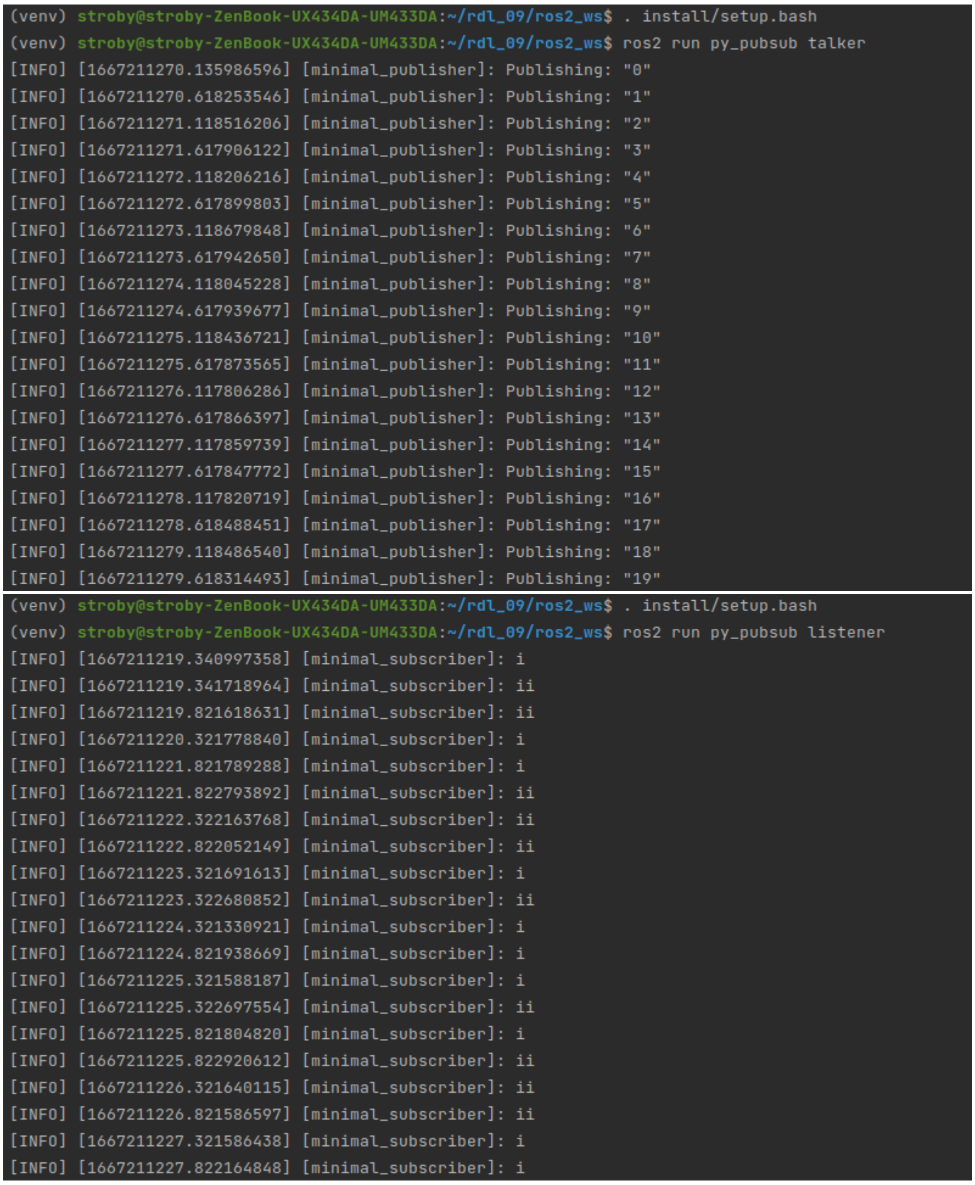
2. How many times per second does the publisher publish the data? How can you change this so that four messages are published per second? To send four messages per second, timer_period should b

   Every 0.5 seconds. You can set this in the first argument in the create_timer function.

   2 messages will be published in a second.

5%

3. Run the publisher and subscriber code. Provide screenshots of the outputs generated by the subscriber.



```
(venv) stroby@stroby-ZenBook-UX434DA-UM433DA:~/rdl_09/ros2_ws$ . install/setup.bash
(venv) stroby@stroby-ZenBook-UX434DA-UM433DA:~/rdl_09/ros2_ws$ ros2 run py_pubsub talker
[INFO] [1667211270.135986596] [minimal_publisher]: Publishing: "0"
[INFO] [1667211270.618253546] [minimal_publisher]: Publishing: "1"
[INFO] [1667211271.118516206] [minimal_publisher]: Publishing: "2"
[INFO] [1667211271.617906122] [minimal_publisher]: Publishing: "3"
[INFO] [1667211272.118206216] [minimal_publisher]: Publishing: "4"
[INFO] [1667211272.617899803] [minimal_publisher]: Publishing: "5"
[INFO] [1667211273.118679848] [minimal_publisher]: Publishing: "6"
[INFO] [1667211273.617942650] [minimal_publisher]: Publishing: "7"
[INFO] [1667211274.118045228] [minimal_publisher]: Publishing: "8"
[INFO] [1667211274.617939677] [minimal_publisher]: Publishing: "9"
[INFO] [1667211275.118436721] [minimal_publisher]: Publishing: "10"
[INFO] [1667211275.617873565] [minimal_publisher]: Publishing: "11"
[INFO] [1667211276.117806286] [minimal_publisher]: Publishing: "12"
[INFO] [1667211276.617866397] [minimal_publisher]: Publishing: "13"
[INFO] [1667211277.117859739] [minimal_publisher]: Publishing: "14"
[INFO] [1667211277.617847772] [minimal_publisher]: Publishing: "15"
[INFO] [1667211278.117820719] [minimal_publisher]: Publishing: "16"
[INFO] [1667211278.618488451] [minimal_publisher]: Publishing: "17"
[INFO] [1667211279.118486540] [minimal_publisher]: Publishing: "18"
[INFO] [1667211279.618314493] [minimal_publisher]: Publishing: "19"
```

```
(venv) stroby@stroby-ZenBook-UX434DA-UM433DA:~/rdl_09/ros2_ws$ . install/setup.bash
(venv) stroby@stroby-ZenBook-UX434DA-UM433DA:~/rdl_09/ros2_ws$ ros2 run py_pubsub listener
[INFO] [1667211219.340997358] [minimal_subscriber]: i
[INFO] [1667211219.341718964] [minimal_subscriber]: ii
[INFO] [1667211219.821618631] [minimal_subscriber]: ii
[INFO] [1667211220.321778840] [minimal_subscriber]: i
[INFO] [1667211221.821789288] [minimal_subscriber]: i
[INFO] [1667211221.822793892] [minimal_subscriber]: ii
[INFO] [1667211222.322163768] [minimal_subscriber]: ii
[INFO] [1667211222.822052149] [minimal_subscriber]: ii
[INFO] [1667211223.321691613] [minimal_subscriber]: i
[INFO] [1667211223.322680852] [minimal_subscriber]: ii
[INFO] [1667211224.321330921] [minimal_subscriber]: i
[INFO] [1667211224.821938669] [minimal_subscriber]: i
[INFO] [1667211225.321588187] [minimal_subscriber]: i
[INFO] [1667211225.322697554] [minimal_subscriber]: ii
[INFO] [1667211225.821804820] [minimal_subscriber]: i
[INFO] [1667211225.822920612] [minimal_subscriber]: ii
[INFO] [1667211226.321640115] [minimal_subscriber]: ii
[INFO] [1667211226.821586597] [minimal_subscriber]: ii
[INFO] [1667211227.321586438] [minimal_subscriber]: i
[INFO] [1667211227.822164848] [minimal_subscriber]: i
```

Figure 1: Screenshots of the outputs generated by the subscriber

4. Talker code

10%

```
def __init__(self):
    super().__init__('minimal_publisher')

    # Using Topic circle_radius for the data transfer.
    self.publisher_ = self.create_publisher(Int32, 'circle_radius', 10)

    timer_period = 0.5  # seconds

    self.timer = self.create_timer(timer_period, self.timer_callback)
    self.i = 0

def timer_callback(self):
    msg = Int32()  # Use It32 as data type for number transfer
    msg.data = self.i
    self.publisher_.publish(msg)
    self.get_logger().info('Publishing: "%d"' % msg.data) # Use %d for numbers
    self.i += 1
```

5. Listener code

16%

```
def __init__(self):
    super().__init__('minimal_subscriber')
    self.subscription = self.create_subscription(
        Int32, # Use It32 as data type for number transfer
        'circle_radius', # Using Topic circle_radius for the data transfer.
        self.listener_callback,
        10)
    self.subscription  # prevent unused variable warning

def listener_callback(self, msg):
    circle_radius = msg.data    self.get_logger().info('I heard: "%s"' % msg.data)

    # Calculating the area with pi*r^2
    circle_area = circle_radius * circle_radius * math.pi  ✓

    # Calculating the circumference with 2*pi*r
    circle_circumference = 2 * circle_radius * math.pi  ✓

    # For debugging
    # self.get_logger().info('Radius: "%d" Area: "%s" Circumference: "%d"' % (circle_radius, circle_area % 2, circle_circumference))

    if int(circle_area % 2) == 0: # Check if even
        self.get_logger().info('i')✗ You have to check whether the radius is even or odd and then calculate the area and the circumference according to the condition

    if int(circle_circumference % 2) == 1: # Check if odd.
        self.get_logger().info('ii')✗
```

You can use the print() function to print the area and circumference of the circle.

8%

5 Teleoperation of TurtleBot3

1. Move the robot around with keys and identify which keys perform which movement. Fill up Table 1 and provide sequences of images showing your robot at different positions and orientations.

Good!

a	Increase rotational velocity to the left.	Steer to left
s	Reset all velocities.	force stop
d	Increase rotational velocity to the right.	Steer to right
w	Increase forward velocity.	Move forward
x	Increase backward velocity.	Move backwards

Table 1: Table showing robot steering commands.

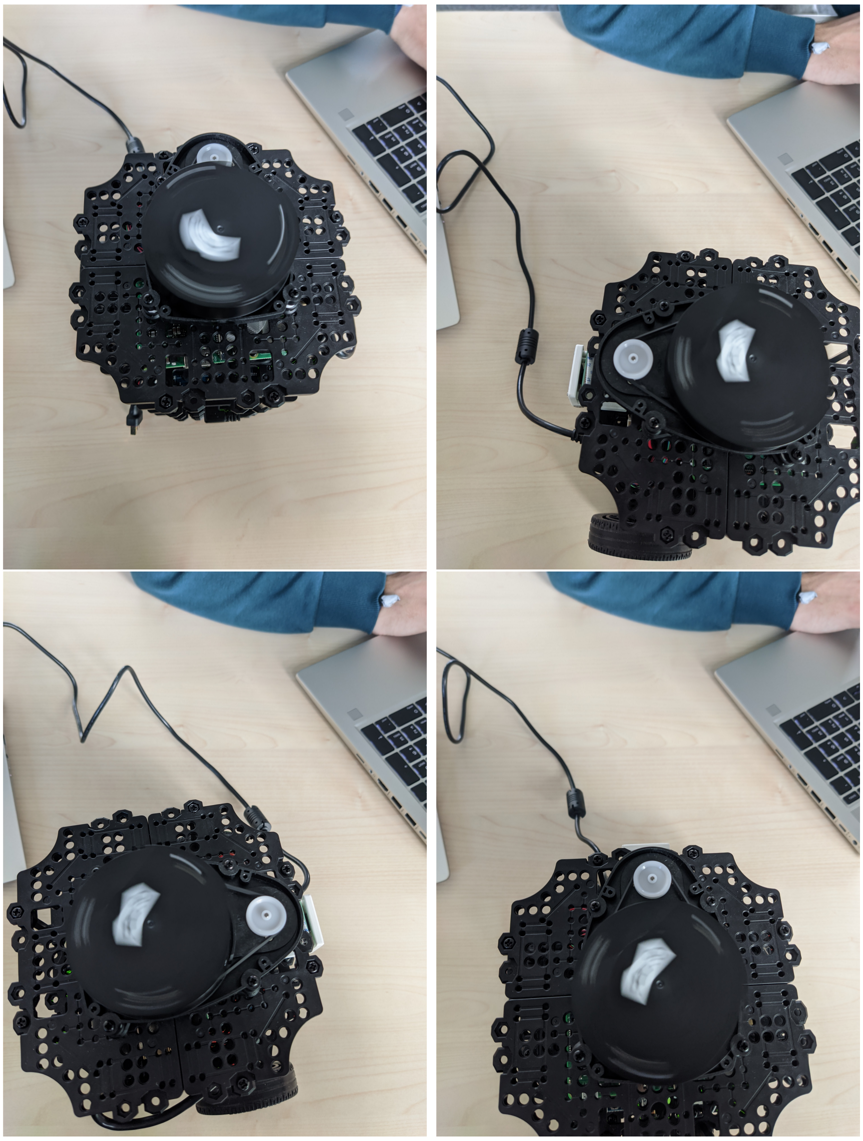


Figure 2: Pictures of robot while moving.

6 Feedback

1. How much time did you spend on doing this sheet per person? Anonymize your answer!

   Person1: 30 minutes (linux was already set up and excluding ros setup on a non Ubuntu distro).

   Person2: About 3 hours.

   Person3: I would say about 3.5 hours.

2. Was it too easy, easy, ok, hard, too hard

   Person1: the difficulty of the assignment was ok.

   Person2: It was a bit difficult but manageable, because i wasn't used to all the new operating systems.

   Person3: It wasn't particular hard I had just some problems with LaTeX.

3. Please tell us what you liked in this exercise sheet.

   Person2: I liked the structure of this exercise sheet.

   Person3: Controlling the Robot i really enjoyed. Also I think that one can easily follow the instruction's in the tasks.

4. Did you face any difficulties? What should be improved?

   Person1: a lot of difficulty setting up ros on Arch this however isn't a problem with the worksheet.

   Person2: Installation and setup in general.

   Person3: I had a hard time setting up the ros environment in PyCharm.