

Lab 03 - Feb 19

Announcement: Staff and students: you are welcome to edit and add materials into this document, or any of the setup (or other) docs, but please put it in blue like this, and then it will be turned into black by the instructors after the change is announced. This will avoid us getting confused regarding what is the starting version and what are the changes.

Reminder: Color key: blue is what students should read and what students edit; orange is important/needs attentions; red is urgent / past due; purple is the status survey; green is done (yay!)

Important Information:

- Main channel for this event is: **#lab03-feb19**
 - Other relevant channels:
 - #help-ros
 - #help-git
 - #help-network
- Important documents:
 - [Setting up Ubuntu laptops and the Duckietops](#)
 - [Setup Step 2.05 - RC control, launched remotely](#)
 - [Lab 03 status spreadsheet](#)

Lab Goals

Goal: 100%

- Everyone has a laptop with ROS operational
 - Can run `roscore`
- Everyone has the duckietown repo on their laptop
 - `catkin_make` succeed
 - `roslaunch` and `roslaunch` auto-complete working

- Everyone has launched a `publisher_node` (provided) under their namespace through a launch file (provided)
- Everyone has launched a `subscriber_node` node (provided) through a launch file to listen to a specific topic
- Everyone knows how to remap a topic using launch file
- Everyone knows how to setup `ROS_MASTER_URI` to connect to different masters through
 - `set_ros_master.sh`
 - proper `~/ .bashrc` setup

Stretch Goal (95%)

- Everyone knows the ethernet share trick and has used that to ssh into their robot
- Everyone writes a node that listens to a specific topic and republish it under a private namespace (combining the code in the publisher node and listener node)
- Everyone knows how to use `rosparam set` and `get`
- Everyone knows how to use `rosbag record` and `play`

Pre-lab (before coming to Beaverworks)

Things that the students should have done before coming to lab:

- Have working Ubuntu 14.04 on their laptop (either native, VirtualBox or VMWare) and add your name to one of the lists in Exercise 0.1
AC will support the virtual machine options - anyone having trouble should come to Beaverworks at 9am. Of course, if you come at 9:55 there might not be enough time for us to help you.
- Have installed ROS on some form of Ubuntu (Part of [Setting up your laptop](#))
 If you have problems please post to **#help-ros**.
- Have set up git on Ubuntu and checkout duckietown ([Setup Step 1.9](#))
 If you have problems post to **#help-git**.
- Read Tutorial on roslaunch <http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch>

Note: If you have not succeeded in setting up Ubuntu, VMWare, Virtualbox, etc., we will provide support in Beaverworks at **9am**. If your computer is not ready at **10am**, you will just follow along watching somebody else. (sorry, no exceptions)

- **Status update! Please edit column C in the [lab 03 status](#)**
 - Are you stuck? Please put your name and problem here:
 -
- **Suggestion box / tips and tricks** (please write your name and things that you found useful to get through the exercise):

Exercise 0.1:

Please add your name to one of these lists (**LP + AC finish the list**):

Ubuntu 14.04 (personal computer): {Liam, Sam, Catherine, Brandon, Teddy, Robert, Wyatt, Tristan, Michael, Giulia, Veronica, Amado, Sang, Guy, Ari, Valerio, Yajun}

Ubuntu 15.04 + Jade (personal computer): {Mrinal}

Ubuntu 14.04 (course duckietop): AC Hans

Mac + VMWare: {Jenny, Robert, Nick, Erlend}

Mac + VirtualBox: {, Joe, Takke, Victoria}

Windows + VMWare:

Windows + VirtualBox:

native Mac OSX {lapentab}

- **Status update! Please edit column D in the [lab 03 status c](#)**
 - Are you stuck? Please put your name and problem here:
-
- **Suggestion box / tips and tricks** (please write your name and things that you found useful to get through the exercise):
-

Exercise 0.5:

- If you have an ethernet cable: connect to ethernet.
- If you don't, connect to the "duckietown-5GHz".
- laptop \$ ping Wolverine.local
- **Status update! Please edit column E in the [lab 03 status](#)**
 - Are you stuck? Please put your name and problem here:
-
- **Suggestion box / tips and tricks** (please write your name and things that you found useful to get through the exercise):
-

Exercise 0.6:

Checkout the duckietown/software repo if you haven't ([Setup Step 1.9 - Github basics](#))

Switch to the lab03-feb19 branch

```
laptop $ cd ~/duckietown
```

```
laptop $ git fetch # added by AC
```

```
laptop $ git checkout lab03-feb19
```

(Update) Note that we are assuming that you followed step 2.x already. For example that implied running scrips such as `duckietown_install_car.sh` etc. Please refer to the documentation

Build the workspace

```
laptop $ cd ~/duckietown # added by AC
```

```
laptop $ source environment.sh # added by AC
```

```
laptop $ cd catkin_ws
```

```
laptop $ catkin_make
```

- **Status update! Please edit column F in the [lab 03 status](#)**
 - Are you stuck? Please put your name and problem here: Ari - opencv issues
 -
- **Suggestion box / tips and tricks** (please write your name and things that you found useful to get through the exercise):
-

Exercise 1.0: Be your own master

Make sure the following lines are in your `~/.bashrc` (This was a part of [Setup Step 2.05 - RC control, launched remotely](#) - Setting up Environment Variables)

```
source /opt/ros/indigo/setup.bash
# Set ROS_HOSTNAME using .local
export ROS_HOSTNAME=$HOSTNAME.local
# Default be your own ROS MASTER
export ROS_MASTER_URI=http://$HOSTNAME.local:11311/
# Allow remote roslaunch with machines that are not in the
know_hosts list.
export ROSLAUNCH_SSH_UNKNOWN=1
export DUCKIETOWN_ROOT=$HOME/duckietown
source $DUCKIETOWN_ROOT/catkin_ws/devel/setup.bash
```

Note: Nothing needs to be customized in the above lines. (If you have a python export path in the file, make sure you place this AFTER it)

Start byobu by

```
laptop $ byobu
```

Open a new Terminal by hitting F2. (~/.bashrc runs every time you open a new terminal)

You can check your current ROS_MASTER_URI of the terminal by

```
laptop $ echo $ROS_MASTER_URI
```

nanoYou should see something like

```
http://YOUR_HOST_NAME.local:11311/
```

Note that this can differ from terminal to terminal (Two terminals on your laptop can have different ROS_MASTER_URI).

Start a ros master on your machine by

```
laptop $ roscore
```

Pay attention to the printout, you should see this line

```
ROS_MASTER_URI=http://YOUR_HOST_NAME.local:11311/
```

This means that you've started a ros master with the ROS_MASTER_URI. Keep this terminal alive.

Open a new terminal and

```
$ rostopic list
```

You should see

```
/rosout
```

```
/rosout_agg
```

which are the topics that captures all the printouts in ROS.

- **Status update! Please edit column G in the [lab 03 status](#)**
 - Are you stuck? Please put your name and problem here:
 -
- **Suggestion box / tips and tricks** (please write your name and things that you found useful to get through the exercise):
-

Exercise 1.1: Start Talking

Start the publisher_node using roslaunch

```
laptop $ roslaunch lab03 publisher_node.launch
```

You should see error messages with the line

```
[/home/your_user_name/duckietown/catkin_ws/src/lab03/launch/publisher_node.launch] requires the 'veh' arg to be set
```

This is because the launch file requires the arg `veh` to be set. You can do so by

You should see output including the following lines

```
NODES
  /your_robot_name/
    publisher_node (lab03/publisher_node.py)
```

```
laptop $ roslaunch lab03 publisher_node.launch veh:=your_robot_name
```

Open a new terminal (In byobu, hit F2) and list all the nodes by

```
laptop $ rosnode list
```

You should see

```
/your_robot_name/publisher_node
/rosout
```

List all the topics by

```
laptop $ rostopic list
```

You should see

```
/your_robot_name/publisher_node/topic
```

Echo the topic by

```
laptop $ rostopic echo /your_robot_name/publisher_node/topic
```

You should see:

```
data: Hello Duckietown!
---
data: Hello Duckietown!
---
data: Hello Duckietown!
```

Now you're talking!

Kill the echo terminal when you're done by hitting Ctrl+C. (Keep the publisher terminal alive)

- **Status update! Please edit column H in the [lab 03 status](#)**
 - Are you stuck? Please put your name and problem here:
-
- **Suggestion box / tips and tricks** (please write your name and things that you found useful to get through the exercise):
 - **Teddy - Got a weird error 'Warning: error while crawling /home/teddy: boost::filesystem::status: Permission denied: "/home/teddy/.gvfs" ' Fixed with:**

```
#sudo umount ~/.gvfsr
#rm -rf .gvfs/
http://answers.ros.org/question/76896/permission-denied-gvfs/
```

- Can more descriptions been given on what those different commands actually do?
- Can we have a count as to how many terminals we should have at certain intervals? It seems that we sometimes have intervals that aren't doing anything and then we have to open more....
- Jacopo - Byobu hint: Use F8 to rename the open windows (every time you press F2) to keep track of what is what

Exercise 1.2: Listen!

Open a new terminal and launch the subscriber node by

```
laptop $ roslaunch lab03 subscriber_node.launch
veh:=your_robot_name
```

You should see printout including:

```
NODES
  /your_robot_name/
    subscriber_node (lab03/subscriber_node.py)
```

This means that your subscriber_node is up.

Open a new terminal, Let's check all the nodes and topics using rqt_graph

```
laptop $ rqt_graph
```

You should see two nodes:

```
/your_robot_name/publisher_node
/your_robot_name/subscriber_node
```

But no topics? Let's figure out why. Open a new terminal and list all the topics by

```
laptop $ rostopic list
```

You should see at least these two

```
/your_robot_name/publisher_node/topic
/your_robot_name/subscriber_node/topic
```

Let's look deeper into these topics:

```
laptop $ rostopic info /your_robot_name/publisher_node/topic
```

You should see

```
Type: std_msgs/String
Publishers:
  * /megaman/publisher_node (http://Wolverine.local:43041/)
Subscribers: None
```

Notice that there is no subscribers.

Let's look at the subscriber topic too by

```
laptop $ rostopic info /your_robot_name/subscriber_node/topic
```

You should see

```
Type: std_msgs/String
Publishers: None
Subscribers:
  * /megaman/subscriber_node (http://Wolverine.local:50693/)
```

Notice that there is no publisher

rqt_graph only shows “active” connections, since no one is listening to /your_robot_name/publisher_node/topic and no one is publishing to /your_robot_name/subscriber_node/topic, they aren’t shown in the graph.

Our subscriber_node is listening to the “wrong” topic. How do we fix that?

Kill the subscriber node by Ctrl+C in the subscriber terminal.

- **Status update! Please edit column I in the [lab 03 status](#)**
 - Are you stuck? Please put your name and problem here:
 -
- **Suggestion box / tips and tricks** (please write your name and things that you found useful to get through the exercise):
 - Use F3 (or Fn+F3) to switch screens in byobu

Exercise 1.3: Be a good listener (through remapping in launch files)

You can **remap** the topics a node is publishing/subscribing to through the `<remap>` tag in launch files.

Open the launch file we used to launch the subscriber by

```
laptop $ roscd lab03/launch
laptop $ nano subscriber_node.launch
```

(Of course you can use other editors too)

Notice the remapping line is commented out

```
<!-- <remap from="subscriber_node/topic" to="publisher_node/topic"/>
-->
```

Uncomment it by removing the `<!--` and `-->`

This will “remap” the topic the subscriber node is listening to **from**

```
/your_robot_name/subscriber_node/topic to
/your_robot_name/publisher_node/topic
```


Save your edits (Ctrl+O in nano) and then exit the editor (Ctrl+X in nano)

Now let's launch the subscriber node using the newly edited launch file:

```
laptop $ roslaunch lab03 subscriber_node.launch  
veh:=your_robot_name
```

You should start seeing printouts like these:

```
[INFO] [WallTime: 1455819181.415229] I heard: Hello Duckietown!  
[INFO] [WallTime: 1455819182.415265] I heard: Hello Duckietown!  
[INFO] [WallTime: 1455819183.415211] I heard: Hello Duckietown!
```

Open a new terminal and run `rqt_graph`

```
laptop $ rqt_graph
```

You should see now the `/your_robot_name/publisher_node` and `/your_robot_name/subscriber_node` are connected through the topic `/your_robot_name/publisher_node/topic`

Quit `rqt_graph` (close the window) and list all the topics by

```
laptop $ rostopic list
```

You should see:

```
/your_robot_name/publisher_node/topic  
/rosout  
/rosout_agg
```

Note that the `/your_robot_name/subscriber_node/topic` is not there anymore since it has been remapped.

Keep the `subscriber_node` alive

- **Status update! Please edit column J in the [lab 03 status](#)**
 - Are you stuck? Please put your name and problem here:
-
- **Suggestion box / tips and tricks** (please write your name and things that you found useful to get through the exercise):
 - Why do we map from subscriber to publisher, but the `rqt-graph` shows an arrow the other way around? Could we instead change the publisher launch file and do the remapping there?
 - Generally, you don't want to do that, in the case where you have multiple subscribers, you want to know where the publisher is going, so if you keep that consistent (And just change the subscribers as necessary).. it's better. imo

Exercise 1.4: Say Something Else (through editing publisher_node.py)

Now that we're comfortable with launch files and remapping, let's start looking into the source code.

Open the `lab03/src/publisher_node.py` in your favorite editor by

```
laptop $ roscd lab03/src
laptop $ nano publisher_node.py
```

At line 11, you should see

```
msg.data = "Hello Duckietown!"
```

Change this line to

```
msg.data = "Hello your_name!"
```

Save and exit the editor. (Ctrl+O and then Ctrl+X in nano)

Check that both the publisher node and the subscriber node are still there by:

```
laptop $ rosnode list
```

You should see

```
/your_robot_name/publisher_node
/your_robot_name/subscriber_node
/rosout
```

Check that the subscriber node is still printing "I heard: Hello Duckietown!".

Open a new terminal and launch the now edited `publisher_node.py` by:

```
laptop $ roslaunch lab03 publisher_node.launch
veh:=your_robot_name
```

Switch to the subscriber terminal, you should see that it's now printing out "I heard: Hello your_name!"

Let's check again what nodes are running by:

```
laptop $ rosnode list
```

You should see

```
/megaman/publisher_node
/megaman/subscriber_node
/rosout
```

What happened? I thought I launched another publisher_node? Why is there still only one publisher node? This is because the new publisher node you just launched has exactly the same name as the old one (/your_robot_name/publisher_node). In ROS when two nodes have the same name, the old one will be terminated and replaced with the new one.

- **Status update! Please edit column K in the [lab 03 status](#)**
 - Are you stuck? Please put your name and problem here:
-
- **Suggestion box / tips and tricks** (please write your name and things that you found useful to get through the exercise):
 - **Why does the old publisher node print out not indicate that it was terminated?**

Exercise 1.5 Record a bag

rosviz is the tape recorder equivalent in ROS. You can record topics and play them back in the future.m

Bag files can be large and we don't want large files to get into the git repo by accident. Open a new terminal and create a bags folder under your home folder by:

```
laptop $ cd ~  
laptop $ mkdir bags  
laptop $ cd bags
```

Record the topic of interest, in this case:

```
laptop $ rosviz record /your_robot_name/publisher_node/topic
```

You should seepu

```
[ INFO] [1455822038.395319908]: Subscribing to /megaman/publisher_node/topic  
[ INFO] [1455822038.399604517]: Recording to 2016-02-18-14-00-38.bag.
```

The bag is now recording.

Wait for 30sec, and then stop the recording by Ctrl+C.

Now there should be a bag file in your current folder (~/.bags/). You can check by:

```
laptop $ ls
```

You should see something like:

```
2016-02-18-14-00-38.bag
```

Now let's see what's in the bag by:

```
laptop $ rosviz info XXXXX.bag
```

where XXXX is the output from ls.

You should see something like:

```
path:          2016-02-18-14-00-38.bag
version:       2.0
duration:      1:27s (87s)
start: Feb 18 2016 14:00:39.49 (1455822039.49)
end:          Feb 18 2016 14:02:06.49 (1455822126.49)
size: 11.4 KB
messages: 88
compression: none [1/1 chunks]
types: std_msgs/String [992ce8a1687cec8c8bd883ec73ca41d1]
topics: /your_robot_name/publisher_node/topic 88 msgs : std_msgs/String
```

- **Status update! Please edit column L in the [lab 03 status](#)**
 - Are you stuck? Please put your name and problem here:
 -
- **Suggestion box / tips and tricks** (please write your name and things that you found useful to get through the exercise):
 -

Exercise 1.6 Play A Bag

Now let's see how we can play a bag.

First, kill the publisher node by

```
laptop $ rosnode kill /your_robot_name/publisher_node
```

You should see

```
killing /megaman/publisher_node
killed
```

(Don't worry, it's in a better place now)

Switch to the subscriber terminal, check that it's not printing out new messages anymore.

Now, go to the bags folder and play the bag:

```
laptop $ cd ~/bags
laptop $ rosbag play XXXXX.bag
```

where XXXX is the output from ls.

You should see something like:

```
[ INFO] [1455822770.954076757]: Opening 2016-02-18-14-00-38.bag
Waiting 0.2 seconds after advertising topics... done.
Hit space to toggle paused, or 's' to step.
[RUNNING] Bag Time: 1455822047.188216 Duration: 7.701535 / 87.000534
```

Your bag is now playing

Switch again to the subscriber terminal, you should see that it is printing out “I heard: Hello your_name!” again. The messages published to the /your_robot_name/publisher_node/topic during bag recording is now being played back to the same topic in real time.

The playback will finish in about 30sec (that’s how long you waited during recording)

Fun extra:

Play the bag again and launch rqt_graph. Who (which node) is publishing to /your_robot_name/publisher_node/topic exactly?

Note that the nodes were not recorded in the bag files, just the topics.

- **Status update! Please edit column M in the [lab 03 status](#)**
 - Are you stuck? Please put your name and problem here:
-
- **Suggestion box / tips and tricks** (please write your name and things that you found useful to get through the exercise):
-

Exercise 1.7 Clean up

We are done with Exercise 1 at this point. Now would you kindly:

Kill all the nodes and roscore (by Ctrl+C in each terminal)

Kill all the terminals (by Ctrl+D repeated in byobu)

Pat yourself on the back before you move on to Exercise 2

- **Status update! Please edit column N in the [lab 03 status](#)**
 - Are you stuck? Please put your name and problem here:
-
- **Suggestion box / tips and tricks** (please write your name and things that you found useful to get through the exercise):
-

Exercise 2.0: There can only be one (master)

In ROS, multiple machines can communicate by connecting to the same master.

The ~/.bashrc setting in Exercise 1.0 defaults your ROS_HOSTNAME to be /your_host_name.local and ROS_MASTER_URI to be http://your_host_name.local:11311/

Open a terminal and check your current ROS_MASTER_URI by:

```
laptop $ echo $ROS_MASTER_URI
```

You should see

```
http://your_host_name.local:11311/
```

Now, let's all switch to a new master. The script file `~/duckietown/set_ros_master.sh` provides a simple way to switch your ros master of the current terminal. Let's all use Wolverine as the ros master by

```
laptop $ source ~/duckietown/set_ros_master.sh Wolverine
```

You should see

```
Setting ROS_MASTER_URI...
```

```
ROS_MASTER_URI set to http://Wolverine.local:11311/
```

Now let's see what are the nodes and topics on this rosmaster by:

```
laptop $ rqt_graph
```

You should at least see

```
/megaman/publisher_node
```

Let's see what it has to say by:

```
laptop $ rostopic echo /megaman/publisher_node/topic
```

Ctrl+C to kill the echo when you're done.

- **Status update! Please edit column O in the [lab 03 status](#)**
 - Are you stuck? Please put your name and problem here:
 -
- **Suggestion box / tips and tricks** (please write your name and things that you found useful to get through the exercise):
 - roscore should be running when you call `rqt_graph`
 - (ari)it's running on wolverine. you don't need to run it too

Exercise 2.1: Be heard

Open a new terminal. Remember to set Wolverine as your master by

```
laptop $ source ~/duckietown/set_ros_master.sh Wolverine
```

If you are tired of having to switch the master to Wolverine at every new terminal, you can change the line in `~/.bashrc` file from:

```
export ROS_MASTER_URI=http://$HOSTNAME.local:11311/
```

To:

```
export ROS_MASTER_URI=http://Wolverine.local:11311/
```

This way, your default ros master is Wolverine. You will have to **remember to revert this change** when you're done with the lab today. Or else your ROS is going to complain about not being able to reach master when you try to run roscore at home.

Now. Let's launch the publisher node by

```
laptop $ roslaunch lab03 publisher_vnode.launch veh:=your_robot_name
```

Use rqt_graph to see your node on the graph.

- **Status update! Please edit column P in the [lab 03 status](#)**
 - Are you stuck? Please put your name and problem here:
-
- **Suggestion box / tips and tricks** (please write your name and things that you found useful to get through the exercise):
 - If you open a new byobu tab for rqt_graph and didn't change your ~/.bashrc \$HOST_NAME, then you need to call `$ source ~/duckietown/set_ros_master.sh Wolverine`
 - **Teddy:** The graph might take a long time to draw, have patience and you will persevere in the end!

Exercise 2.2: Listen to megaman

Open a new terminal with Wolverine as master.

In this exercise the goal is to remap the subscriber node to subscribe to /megaman/publisher_node/topic (instead of /your_robot_name/publisher_node/topic)

Open the file subscriber_node.launch. Recall that the line

```
<remap from="subscriber_node/topic" to="publisher_node/topic"/>
```

remaps from /your_robot_name/subscriber_node/topic to /your_robot_name/publisher_node/topic.

Without a leading "/" in the topic names, remapping are done relatively. To remap /your_robot_name/subscriber_node/topic to /megaman/publisher_node/topic, we need to change that to

```
<remap from= "subscriber_node/topic" to="/megaman/publisher_node/topic" />
```

Note that there is a leading "/" in the **to**, but not in the **from**. This is important.

Save the now edited launch file and launch it by:

```
laptop $ roslaunch lab03 subscriber_node.launch
veh:=your_robot_name
You should see printouts of what megaman has to say.
```

Visualize the node/topic graph by running `rqt_graph`:

```
laptop $ rqt_graph
You should see that your_robot_name/subscriber_node is connected to the
/megaman/publisher_node/topic
```

- **Status update! Please edit column Q in the [lab 03 status](#)**
 - Are you stuck? Please put your name and problem here:
-
- **Suggestion box / tips and tricks** (please write your name and things that you found useful to get through the exercise):
 - **If you got “badly formed xml” it’s because it was using the wrong type of quote marks and an equals sign was missing :| I changed it up there^**

Exercise 2.3 Repeat after me

The original `subscriber_node` subscribes to a topic and prints out the message. In this exercise we will write a `repeater_node` that subscribes to a topic, prints out the message, and republishes the topic right away.

Partial codes of `repeater_node` are provided to you in `lab03/src/repeater_node.py`. Open this file.

In line 13, the publisher, which will publish to “`repeater_node/topic_out`” is already setup for you :

```
publisher = rospy.Publisher("~topic_out",String,queue_size=1)
```

In line 15, the subscriber, which default subscribing to “`repeater_node/topic_in`” is setup for you

```
subscriber = rospy.Subscriber("~topic_in", String, callback)
```

The callback function in line 6~8 will be invoked when receiving a msg

```
6 def callback(msg):
7     rospy.loginfo("I heard: %s" %(msg.data))
8     # TODO: Publish the received message using publisher
```

Your mission, should you choose to accept it, is to edit line 8 so that the msg is published. (Hint look at `publisher_node.py` to see how this is done).

Partial codes of a launch file is also provided as lab03/launch/repeater_node.launch. Your mission, should you accept it, is to edit line 3 and 4:

```
<remap from="repeater_node/topic_in" to=""/>
<remap from="repeater_node/topic_out" to=""/>
```

so that when you

```
laptop $ roslaunch lab03 repeater_node.launch
veh:=your_robot_name
```

The `/your_robot_name/repeater_node` node subscribes to `/megaman/publisher_node/topic` and republishes the message to `/your_robot_name/repeater_node/topic` (instead of `/your_robot_name/repeater_node/topic_out`)

When you've successfully launched the `repeater_node` with the desired behavior, send a message to @liu on slack at #lab03-feb19 stating the name of your robot (`your_robot_name` in this document) and that you're all done and he will check your work.

- **Status update! Please edit column R in the [lab 03 status](#)**
 - Are you stuck? Please put your name and problem here:
-
- **Suggestion box / tips and tricks** (please write your name and things that you found useful to get through the exercise):
 - It would be nice to see how to check the intended behavior (what should `rqt_graph` look like?)
 - if you see this an error that looks like this:

```
File "[...]b03/src/repeater_node.py", line 9
#whatever you wrote in line 9#
      ^
```

IndentationError: unindent does not match any outer indentation level

check the settings of your editor for tab-stops and rewrite that line.

In VI you can fix it by typing:

```
:set noexpandtab
```

If you are using any other editor, you can fix that by installing VI using:

```
$> sudo apt-get install vim
```

Exercise 2.4 (Optional) Glorious `rqt_graph`

Let's create an interesting `rqt_graph` (Think of this as a modern art project)

Keep your `repeater` node from exercise 2.3 alive.

Check and see what are the live topics on the master by

laptop \$ rostopic list

Pick a topic that is **not** /megaman/publisher_node/topic

Now edit repeater_node.launch so that `repeater_node/topic_in` is remapped to a live topic of your choice (as long as it's **not** /megaman/publisher_node/topic)

Launch a new repeater node under the namespace `your_name` (Note: not `your_robot_name`) by
laptop \$ roslaunch lab03 repeater_node.launch veh:=your_name

Open rqt_graph by:

laptop \$ rqt_graph

And see how it looks.

Timeline:

Thu 2pm:

- LP: announcement sent out regarding the prelab
- LP: call John Vivilecchia and get access to BW

@everyone There is lab tomorrow in Beaverworks as usual **10am-1pm**. The document for the lab is

(<https://docs.google.com/document/d/1fJe0lZXczCmrez8LEv3s5BxbIO0owAwpc3gZul1hhec/edit?usp=sharing>) and the channel for the lab is #lab03-feb19. This lab has a "pre-lab." *Everyone* please:

- Bring your laptop
- Have ROS indigo installed on your laptop (for instructions see <http://drive.google.com/open?id=1uYgTz74Pqs4Kiwd7KxhHMHpQH9uuTBCMiPqGc9Qd1Tk>)
- Have git set up on your laptop (for instructions see http://drive.google.com/open?id=1inbwS7PNHY_-VI0iLWQZi5AKT4xT7YVtPLcQ2hTOml8)
- Read this tutorial <http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch>

If you are having trouble with any of these please come to Beaverworks before the lab (*9am*) and *the CTO himself* will be there to help.

Fri 9am:

There are two teams: one led by AC (the laptop support team) and one led by SY the lab setup team

members of Team AC: AC

members of Team SY: SY, LP, MN, HZ, ?

- **9am: Team AC** is at Beaverworks ready for students to show up who need help
 - **9:15am LP has brought router, switch, and plenty of ethernet cables to BW. Optional: one person joins team AC.**
 - For team AC the objectives should be that every laptop
 - 1) can ping google.com
 - 2) can ping the router
 - 3) can ping a robot or laptop with \$hostname.local
 - 5) Can git clone (or git pull) the duckietown repo
 - 4) run roscore & rviz and see rviz open.
- **9am: Team SY** arrive at 32-226
 - Go through the checklist
 - Move stuff to BW

Fri 9:30

Team SY arrives at Beaverworks.

- ?HZ - set up the network
- SY - prepare Megaman and Wolverine
- LP - ?
- MN - ? (Grab remaining Duckietops? + Kitt?)

Fri 10:00am

- Lab starts; anyone without a functional laptop should find a friend

Fri 10:10am

- SY gives a quick overview of the lab

Fri 10:25am - 12:30pm

- Lab starts. Students start going through the exercise
- Students progress through lab

Fri 12:15

- pizza arrives

12:30pm

- Drop everything and have lunch break

12:45pm

- Everyone (Staff and student) pack and clean up
- One staff goes back to 226 to prepare for receiving equipment

1:00 pm

- Everybody is out of Beaverworks

Equipment checklist

- Network
 - 2x 16-port switches
 - 2x smaller switches
 - 5 super long (orange?) ethernet cable
 - 30 long (orange) ethernet cable
 - 30 short (green) ethernet cable
 - 1x airport express
 - 1x extra airport express
- Power
 - 4x Power strips ()
 - 4x Power extender
- Duckietops for loan
- zipties and duck tape to manage ethernet cords
- **Robots? Megaman + Pontiac (Kitt is available if needed)**