D158 Ubuntu Workstation
Login and Setup

Type in your Seminar Username and password.

When the Unity Desktop comes up, press Ctnl + Alt + T and bring up a terminal window. At the command prompt, type:

```
$ echo “source /opt/ros/kinetic/setup.bash” >>  ~/.bashrc
$ source  ~/.bashrc
```

You only need to do these commands the first time you login. These commands set up your account to recognize the ROS files on the system.

Turtlesim Exercise

Reference: Additional information on Turtlesim can be found in ROS Robotics By Example, Chapter 1 – Turtlesim pages 19 – 32.

Tip: Remember to use tab completion to see the fields of the message!

Output from the following commands can be found in the previously referenced section on Turtlesim. Refer to this section in the book for descriptions of the commands or if there are any difficulties or errors.

To start Turtlesim, first open a terminal window and start the ROS Master with the command:

```
$ roscore
```

Open a 2nd terminal window and start the Turtlesim node:

```
$ rosrun turtlesim turtlesim_node
```

Now it is time to examine the ROS components that were discussed in the lecture: nodes, topics, services and message. Open a 3rd terminal window and try the following commands to explore the ROS elements are the framework of Turtlesim:

```
$ rosnode list
```

```
$ rosnode info /turtlesim
```

```
$ rostopic list
```

```
$ rostopic type /turtle1/color_sensor
```

```
$ rosmsg list | grep turtlesim
```

```
$ rosmsg show turtlesim/Color
```

```
$ rostopic echo /turtle1/color_sensor
```

Use Ctrl+C to kill the process.

(page 26) The following commands will introduce how to move the turtle around in the window:

```
$ rostopic type /turtle1/cmd_vel
```

```
$ rosmsg show geometry_msgs/Twist
```

```
$ rostopic pub /turtle1/cmd_vel geometry_msgs/Twist -r 1 -- '[2.0, 0.0, 0.0]''[0.0, 0.0, 1.8]'
```

(page 28) The following command moves the turtle using the keyboard:

```
$ rosrun turtlesim turtle_teleop_key
```

Use Ctrl+C to kill the process.

(page 28) The following commands will introduce Turtlesim parameters and ROS service:

```
$ rosparam list
```

```
$ rosparam get /
```

```
$ rosparam set background_b 0
$ rosparam set background_g 0
$ rosparam set background_r 255
$ rosservice call /clear
```

(page 30) The following commands will find the **pose** of the turtle and introduce ROS service commands to move the turtle:

```
$ rostopic type /turtle1/pose
```

```
$ rosmsg show turtlesim/Pose
```

```
$ rostopic echo /turtle1/pose
```

Use Ctrl+C to kill the process.

```
$ rosservice call /turtle1/teleport_absolute 1 1 0
```

```
$ rosservice call /turtle1/teleport_relative 1 0
```

Trying more with Turtlesim

These exercises are in addition to the material explained in the referenced book.

More Turtlesim services

Try spawning a turtle using a rosservice call. This command requires arguments for the x, y, theta and name of the turtle:

```
$ rosservice call /spawn 8 8 0 my_turtle
        name: my_turtle
```

Change pen settings. This command requires arguments for the pen's color (r g b), width (width), and turning the pen on (0) or off (any other number):

```
$ rosservice call /my_turtle/set_pen 255 255 255 7 0
```

```
$ rosservice call /my_turtle/teleport_absolute 1 8 0
```

```
$ rosservice call /my_turtle/set_pen 100 100 100 15 0
```

```
$ rosservice call /my_turtle/teleport_absolute 1 1 3.14
```

```
$ rosservice call /my_turtle/set_pen 100 100 100 15 1
$ rosservice call /my_turtle/teleport_relative 5 3.14
```

```
$ rosservice call /my_turtle/set_pen 50 0 50 10 0
```

```
$ rosservice call /my_turtle/teleport_relative 3 1.57
```

Try killing a turtle using a rosservice call:

```
$ rosservice call /kill turtle1
```

ROS Control Turtlesim Guide

These exercises are from the Turtlesim Control Guide: http://sceweb.uhcl.edu/harman/A_CRS_ROS_I_SEMINAR/3_1ROS_I_Seminar_TurtlesimControl_1_23_2018a.pdf

Move Turtle in a Circle Once (page 10)

```
$ rostopic pub -1 /turtle1/cmd_vel geometry_msgs/Twist -- '[2.0, 0.0, 0.0]' '[0.0, 0.0, 1.8]'
```

Reset Turtle (page 12)

```
$ rosservice call /reset
```

Move Turtle in a Continuous Circle (page 12)

```
$ rostopic pub /turtle1/cmd_vel geometry_msgs/Twist -r 1 -- '[2.0, 0.0, 0.0]' '[0.0, 0.0, 1.8]'
```

ROS tool: rqt_plot (page 13-14)

```
$ rqt_plot /turtle1/pose/x:y:z
```

Ctrl + C to stop turtle

Clear the Turtlesim screen (page 21)

```
$ rosservice call /clear
```

Python script to move turtle forward (page 21)

Copy the turtlesim1.py script from http://sceweb.uhcl.edu/harman/A_CRS_ROS_I_SEMINAR/UNIT4_EXERCISES/turtlesim1_py.txt into an editor on your Ubuntu Desktop. (We use the gedit editor.) Save the file with the name turtlesim1.py. Change to the Desktop folder:

```
$ cd Desktop
```

Change permission on file to make it executable:

```
$ chmod +x turtlesim1.py
```

Run the program:

```
$ python turtlesim1.py
```

Python script to move turtle in a circle (page 21)

Copy the turtlesim2.py script from http://sceweb.uhcl.edu/harman/A_CRS_ROS_I_SEMINAR/UNIT4_EXERCISES/turtlesim2_py.txt into an editor on your Ubuntu Desktop. Save the file with the name turtlesim2.py. Change to the Desktop folder:

```
$ cd Desktop
```

Change permission on file to make it executable:

```
$ chmod +x turtlesim2.py
```

Run the program:

```
$ python turtlesim2.py
```

ROS tool: rqt_graph (page 23)

```
$ rqt_graph
```

Python Control of Turtlesim

These exercises are from the Turtlesim Python examples:
http://sceweb.uhcl.edu/harman/A_CRS_ROS_I_SEMINAR/3_2_PythonControlOfTurtlesim_Pose_GoToGoal.pdf

Python script to display the turtle's position

Copy the poseTurtlesimWithRounding.py script from
http://sceweb.uhcl.edu/harman/A_CRS_ROS_I_SEMINAR/UNIT4_EXERCISES/poseTurtlesimWithRounding_py.txt
into an editor on your Ubuntu Desktop. Save the file with the name poseTurtlesimWithRounding.py. Change to the Desktop folder:

```
$ cd Desktop
```

Change permission on file to make it executable:

```
$ chmod +x poseTurtlesimWithRounding.py
```

Run the program:

```
$ python poseTurtlesimWithRounding.py
```

Python script to move turtle to a goal with proportional control

Copy the gotogoal.py script from
http://sceweb.uhcl.edu/harman/A_CRS_ROS_I_SEMINAR/UNIT4_EXERCISES/gotogoal_py.txt
into an editor on your Ubuntu Desktop. Save the file with the name gotogoal.py. Change to the Desktop folder:

```
$ cd Desktop
```

Change permission on file to make it executable:

```
$ chmod +x gotogoal.py
```

Run the program:

```
$ python gotogoal.py
```