Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Technical Aspects of Multimodal Systems
Department of Informatics

Robot Practical Course Master

Assignment #1

On this Assignment, we will setup the software which will be needed in this semester and start using the turtlesim. The goal will be drawing a ”Haus vom Nikolaus” (an old german drawing puzzle for kids) in the turtlesim.

Task 1.1 Set up ROS environment: Your system has to be configured, so that it will be simpler to use ROS later on.

1.1.1: Boot the lab computer to Ubuntu and login using your normal account (something like `11musterm`).

1.1.2: Create a folder which you will use as a ROS workspace.

```
mkdir ~/ros_workspace
```

1.1.3: To set up the ROS environment automatically, open your bashrc file (~/.bashrc) with your favorite text editor and enter

```
source /opt/ros/<ROSVERSION>/setup.bash
export ROS_PACKAGE_PATH=~/ros_workspace:$ROS_PACKAGE_PATH
export ROS_WORKSPACE=~/ros_workspace
```

1.1.4: To confirm that it is working go back to your terminal and run

```
echo $ROS_PACKAGE_PATH
```

You should see something like

```
/home/<USER>/ros_workspace:/opt/ros/<ROSVERSION>/share:/opt/ros/<ROSVERSION>/stacks
```

1.1.5: To go directly to your working directory, you can now use:

```
roscd
```

Task 1.2 Starting turtlesim: In this section you will learn to use the simulator for the turtle bot and basic ROS commands.

1.2.1: Open a shell with 4 tabs.
Start a roscore in the 1st tab:

```
roscore
```

Start Turtlesim in the 2nd tab:

```
rosrun turtlesim turtlesim_node
```

Start the keyboard control in the 3rd tab:

```
rosrun turtlesim turtle_teleop_key
```

Turtlesim should be started and you should be able to move the turtle around by using the arrow keys.
Now try some off these commands in the 4th tab to learn about your program.

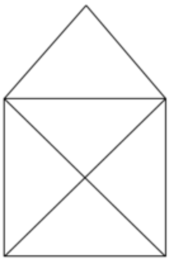
```
rosnode list
rostopic list
rostopic info TOPICNAME
rostopic echo TOPICNAME
rosrun rqt_graph rqt_graph
```

Task 1.3 Explain ROS: Explain to your supervisor the basic concepts of ROS. You should name:

- Nodes
- Communication
  - Topics
  - Services
  - Actions
- ROScore

1.3.1: Explain why you have to set up the environment.

Task 1.4 First Program for Turtlesim: Now you will write a program that moves the turtle so, that it will draw a ”Haus vom Nikolaus”. It looks like this and has to be drawn in one move.



1.4.1: Create a rospackage in your workspace. For this task, look up this tutorial:

```
http://wiki.ros.org/ROS/Tutorials/CreatingPackage
```

Dependencies are changeable later on but it is easier to use them from the start. Your program depends on the Turtle sim.

1.4.2: Look up the ROS tutorial for a simple publisher. There is one for C/C++ and one for Python. **You have to make sure that you are looking up the right version of ROS.**

```
C/C++: http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber(c++)
Python: http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber(python)
```

1.4.3: Now write a publisher (either in C/C++ or Python), which sends the right messages to draw a ”Haus vom Nikolaus”. In this tutorial you can look up which topic the turtle uses.

```
http://wiki.ros.org/ROS/Tutorials/UnderstandingTopics
```

Depending on your programming language ROS expects your code to be in different folders(C/C++: src; Python: scripts). Every program for ROS has to be compiled at least once, even when you use python, so the dependencies, messages etc are compiled. You can start you program with this command

```
rosrun <ROS_PACKAGE> <THE_PROGRAMM>
```

1.4.4: Read up on the Services avaiable by Turtlesim. After drawing the Image, clean the canvas and (optional) call other services, before redrawing the image. Present your loop to your supervisor.