

# Exercise Session 1

## Theory

- ROS architecture
- ROS master, nodes, and topics
- Console commands
- Catkin workspace and build system
- Launch-files

## Exercise

Get to know ROS by inspecting the simulation of a Husky robot.

1. Setup the Husky simulation:

[http://wiki.ros.org/husky\\_gazebo/Tutorials/Simulating%20Husky](http://wiki.ros.org/husky_gazebo/Tutorials/Simulating%20Husky)

2. Launch the simulation and inspect the created nodes and their topics using:

```
roscpp list
rostopic list
rostopic echo [TOPIC]
rostopic hz [TOPIC]
rqt_graph
```

For more information take a look at the slides or:

<http://wiki.ros.org/rostopic>

<http://wiki.ros.org/rosnode>

3. Command a desired velocity to the robot from the terminal (`rostopic pub [TOPIC]`)

4. Use **teleop\_twist\_keyboard** to control your robot using the keyboard. Find it online and compile it from source! Use `git clone`.

For a short git overview see:

[http://rogerdudler.github.io/git-guide/files/git\\_cheat\\_sheet.pdf](http://rogerdudler.github.io/git-guide/files/git_cheat_sheet.pdf)

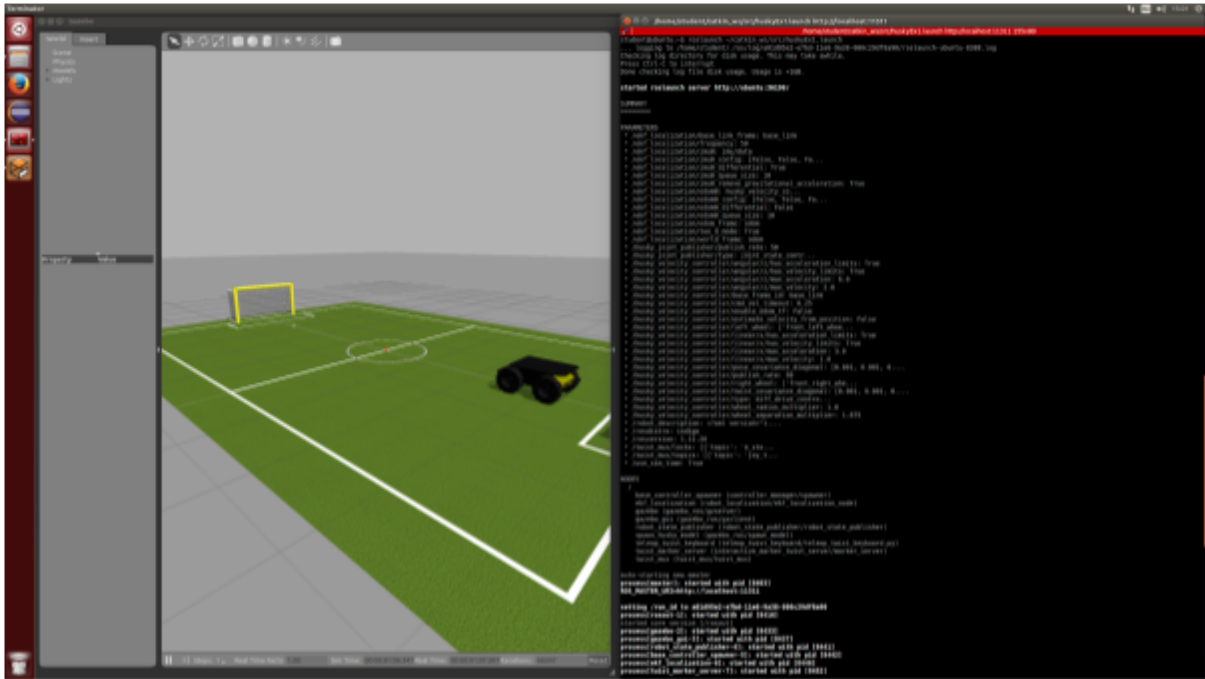
5. Write a launch file with the following content:

- husky simulation with a different world:

Include `husky_empty_world.launch` file and change the `world_name` Argument, e.g. `worlds/robocup14_spl_field.world` a world from the directory `/usr/share/gazebo-2.2/worlds`.

Note: the `world_name` is with respect to `/usr/share/gazebo-2.2/`

- `teleop_twist_keyboard` node



Left: Gazebo with Robocup14 World, Right: First lines of output when starting the launch file you have to set up

## Evaluation

- ❑ Check if teleop\_twist\_keyboard is compiled from source (roscd teleop\_twist\_keyboard should show the catkin\_ws folder) [40%]
- ❑ Start the launch file. This should bring everything up that's needed to drive Husky with the keyboard as shown in the above image. [60%]