ROB314 – Session 1 - Exo 2

Configuration

You will use the package *turtlesim* that should be already installed.

You will need to use 4 terminals. With *terminator*, it is easier : you can split it in 4.

Console Tab Nr. 1 – Starting a *roscore*

- Start a roscore with

```
> roscore
```

- Take the time to look at what's on display.

Console Tab Nr. 2 – Starting a *turtlesim* node

- Run a talker demo node with

```
> rosrun turtlesim turtlesim_node
```

- The node *turtlesim_node* of the package turtlesim is launched.
- His work is to :
  - launch the turtlesim windows

Console Tab Nr. 3 – Analyze *turtlesim* node

- See the list of active nodes

```
> rosnode list
```

- We find the turtlesim node in the list

- Show information about the *talker* node

```
> rosnode info /turtlesim
```

- We see that the node /*turtlesim* have several possible *publications and several subscriptions*

Console Tab Nr. 4 – Starting a *turtle_teleop_key* node

- Run a *turtle_teleop_key* demo node with

```
> rosrun turtlesim turtle_teleop_key
```

- This node permit to move and control, with the keyboard, the turtlesim in the windows.
- You must be careful to click on the terminal before using the keyboard arrows !

Console Tab Nr. 3 – Analyze

- See the new *turtle_teleop_key* node with

```
> rosnode list
```

- We have a new element */turtle_teleop*

- Show the connection of the nodes over the /turtle1/cmd_vel topic with

```
> rostopic info /turtle1/cmd_vel
```

- We see the *publishers* of this topic : here the node */teleop_turtle*
- We see that *subscribers* of this topic : here the node */turtlesim*

Console Tab Nr. 3 – rqt_graph

- The tool rqt_graph provides a visualization of the ROS computation graph. It is useful to understand what happens in our ROS project.

```
> rqt_graph &
```

Console Tab Nr. 3 – Publish my own message from Console

- For example, to make the turtle move forward at a 0.2m/s speed, you can publish a *cmd_vel* message to the topic */turtle1/cmd_vel*:

```
> rostopic pub /turtle1/cmd_vel geometry_msgs/Twist '{linear: {x: 0.2, 
y: 0, z: 0}, angular: {x: 0, y: 0, z: 0}}'
```

- Check the result in the turtlesim windows
- We can have the same result by specifying only the linear x velocity:

```
> rostopic pub /turtle1/cmd_vel geometry_msgs/Twist '{linear: {x: 0.2}}'
```

- Some of the messages like *cmd_vel* have a predefined timeout
- If you want to publish a message continuously use the argument -r with the loop rate in Hz

- For example, to make the turtle turn in circles continuously, type:

```
> rostopic pub /turtle1/cmd_vel -r 10 geometry_msgs/Twist '{linear: {x: 
0.8}, angular: {z: 0.5}}'
```