Project Intelligent Robotics
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.

NOTE: We will use the ROS distribution "Indigo" during the project. So always replace "<ROSVERSION>" with "indigo"!!!

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. In a terminal type:

```
mkdir -p ~/pr16_ws/src
```

1.1.3: To set up the ROS environment, initialize your catkin workspace:

```
source /opt/ros/<ROSVERSION>/setup.bash
catkin_init_workspace pr16_ws/src
cd pr16_ws
```

1.1.4: To initially build the empty workspace call:

```
catkin_make
```

1.1.5: Complete setting up your workspace by adding the following line to your `~/.bashrc`:

```
source ~/pr16_ws/devel/setup.bash
```

1.1.6: and restart your bash:

```
exec bash
```

Task 1.2 Set up a git: Working in groups or projects requires the use of a version control system. We're going to use git in this lecture. If you're already familiar with git you can quickly go through this task. If not please learn it at home, because we can't do a full tutorial here. A good tutorial can be found at *www.vogella.com/tutorials/Git/article.html*

1.2.1: Login to the Redmine (redmine.mafiasi.de) with your Mafiasi account (something like **11**musterm). If you don't have an account yet, create one at mafiasi.de. Make sure that you can access the group "2016-Masterproject Intelligent Robotics" in Redmine. If not tell your tutor.

1.2.2: Create a ssh key, if don't already have one.

```
ssh-keygen
```

1.2.3: Copy the **public** key to your Redmine account by clicking on the Redmine website on "my account" → "my public keys".

1.2.4: Go to `~/pr16_ws/src` and clone the existing git project from redmine:

```
git clone ssh://gitolite@git.mafiasi.de/masterproject-intelligent-robotics\
/2016-masterproject-intelligent-robotics/tutorial.git
```

1.2.5: Create a new file ⟨YourName⟩.txt in the repository folder, commit and push it to the server.

```
git add <YourName>.txt
git commit -m "Added name file"
git push
```

1.2.6: Pull the commit of the other groups, add your name as a line to one of the other files and commit the changes.

Task 1.3 Starting turtlesim: In this section you will learn to use the simulator for the turtle bot and basic ROS commands.

1.3.1: 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 1.4 Explain ROS: Explain to your supervisor the basic concepts of ROS. You should name:

- Nodes

- Communication

    - Topics

    - Services

    - Actions

- ROScore

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

Task 1.5 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.
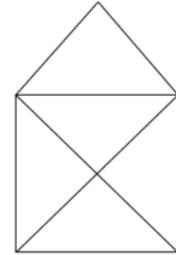
1.5.1: Create a rospackage with your groups name in the git. For this task, look up this tutorial:

```
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.

1.5.2: 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)
```

1.5.3: 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>
```

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

Task 1.6 Using the git: Commit your program and get the programs of the others.

1.6.1: Open a shell and change the directory to your git folder.

```
cd ~/pr16_ws/src/tutorial
```

1.6.2: Look up if your repository has changed. It should have, because you added a program.

```
git status
```

1.6.3: Now add your rospackage and commit it. Get the programs of the others and push your program so the others can get it.

```
git commit -m "message"
git pull
git push
```

1.6.4: Try out the other programs and compare them to your own.