Assignment 3: ROS Tutorials Part 2

References and Prerequisites

- Reading: PRR
  - Chapter 2: Review of the catkin workspace, roslaunch. We won't deal with coordinate frames (tf) yet, but we'll see it later in the course.
  - Chapter 3: Topics, review
  - Chapter 4: Services
- ROS Tutorials
  - Work through the following ROS Beginner Level Tutorials
    - 7. Understanding ROS Services and Parameters
    - 8. Using rqt_console and roslaunch
    - 9. Using rosed to edit files in ROS
      - Note - you can also set your editor to gedit by including this line in your .bashrc file: `export EDITOR='gedit'`
    - 10. Creating a ROS msg and srv
    - 17. Recording and playing back data
    - 18. Getting started with roswtf
- MATLAB Tutorials
  - Read through the Mathworks tutorial on Working with rosbag Logfiles

Git Repository

Just as in the last assignment, in this assignment will be submitted as a git repository on the NPS gitlab server. Create a new repository named "mrc_hw3" using the server's web interface at https://gitlab.nps.edu

When you create repository us on your local machine, do so in your catkin workspace (~/catkin_ws/src). You can do this by first

```
cd ~/catkin_ws/src
```

Then following the directions from the gitlab site which start with...

```
mkdir mrc_hw3
cd mrc_hw3
git init
...
git push -u origin master
```

Make it a ROS Package

As we did in Assignment 2, make your mrc_hw3 directory into a ROS package.

```
cd ~/catkin_ws/src
catkin_create_pkg mrc_hw3
cd ~/catkin_ws

catkin_make
source ~/catkin_ws/devel/setup.bash
```

You can check that this operation was successful by using roscd to move into the mrc_hw3 directory...

```
roscd mrc_hw3
```

Finally we want to make sure and add the files created by catkin_create_pkg to our git repository...

```
roscd mrc_hw3
git add CMakeLists.txt
git add package.xml
```

Exercise 1: Using Services and Parameters to Make Two Letters

This is an extension of the last exercise from Assignment 2. You will need to have the turtlesim node running as described in ROS Tutorial 6 Understanding ROS Topics.

Read the documentation for the turtlesim_node program...
http://wiki.ros.org/turtlesim#turtlesim_node
Note the services described in the documentation.

Make a **scripts** directory in your ROS package. Add a new script called **turtleletterstwo.sh** in the scripts directory that draws the first two letters of your first name in two different colors. The location of this file should be `~/catkin_ws/src/mrc_hw3/turtleletterstwo.sh`
Hint - you will need to use services.

For my first name it might look something like this...

Note: There are many ways to complete this exercise. The details are not terribly important (lower-case vs. uppercase, two-turtles vs. one-turtle, etc.); the point is to get some practice using ROS topics, services and parameters.

Submit your solution by committing and pushing your turtleletterstwo.sh script to the mrc_hw3 remote repository on the gitlab server.

Exercise 2: Launch Script

Add a new directory called **launch** to your ROS package, then add a new file called **turtle.launch**

This launch script should do three things...

1. Start **roscore** (roslaunch does this automatically)
2. Start a **turtlesim_nod**e
3. Start a **turtle_teleop_key** node as we did in this tutorial http://wiki.ros.org/ROS/Tutorials/UnderstandingTopics#turtle_keyboard_teleoperation with the output sent to the screen (output="screen")

You should be able call this launch file like this...

```
roslaunch mrc_hw3 turtle.launch
```

and then drive a turtle around with the keyboard.

Commit and push your turtle.launch file to your mrc_hw3 repository.

Exercise 3: Recording and Playing Back Data

This is an extension of the tutorial on recording and playing back ROS data: http://wiki.ros.org/ROS/Tutorials/Recording%20and%20playing%20back%20data

Here is a ROS bag file that you can download: remapped_turtle.bag

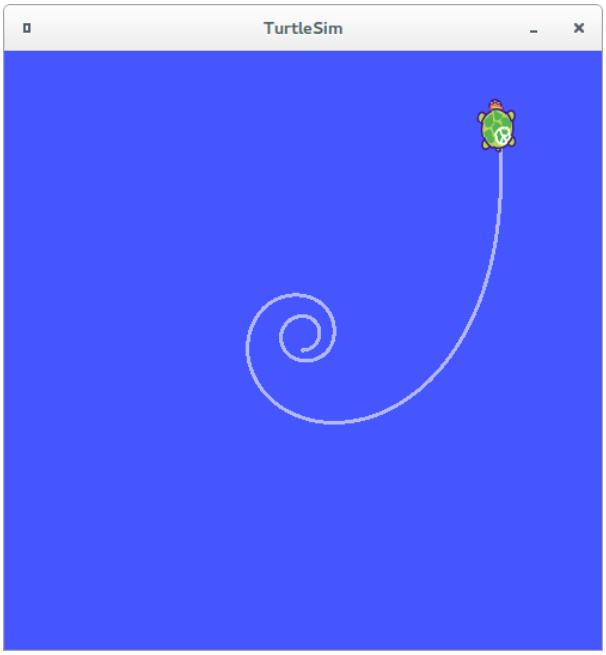
Use the rosbag program to inspect the file. What topics are included in the file?
I generated this file based on the tutorial, but I remapped the velocity command from the /turtle1/cmd_vel to a new topic name. You should be able to determine that new topic name.

Now write a new launch file called **turtle_remap.launch** in the **launch** directory of your package. This launch file will start the turtlesim_node and use the remap function to have that node listen to the topic in the remapped_turtle.bag file. That launch file will be as shown below, but you will need to replace the "XXXXXX" with the appropriate topic.

```
<launch>
<node pkg="turtlesim" type="turtlesim_node" name="simulated_turtle">
        <remap from="turtle1/cmd_vel" to="XXXXXX" />
</node>
</launch>
```

- Open a terminal and use roslaunch to execute your launch file.
- Open another terminal and start recording all the ROS data to a new bag file called **myturtle.bag**
- Open a third terminal and plot the rosgraph
- Open yet another terminal and playback the **remapped_turtle.bag file.**

You should see your turtle do something like this...



- Stop your data logging by using Cntrl-C in the terminal where you started the rosbag program.
- Move the **myturtle.bag** file to a new directory in your package called **data**
- Save your rosgraph to a SVG file called **rosgraph.svg** and put it in your project directory**.**

To complete the assignment, make sure that your **myturtle.bag** and **rosgraph.svg** files are added to your git repository, commit and push.\

Exercise 4: Plotting rosbag Data with MATLAB

This exercise uses the tools described in the Mathworks tutorial on Working with rosbag Logfiles

When you first open MATLAB, type the command rosinit in the command window. You should see something like this...

```
>> rosinit
Initializing ROS master on http://aku:11311/.
Initializing global node /matlab_global_node_12356 with NodeURI http://aku:56098/
```

If not, the ROS toolbox may not be installed - see the instructor!

Use MATLAB to generate two figures based on the recorded data from Exercise 3:

1. A plot the X and Y location of your turtle.
2. A plot of the heading angle (Theta) of your turtle as a function of time.

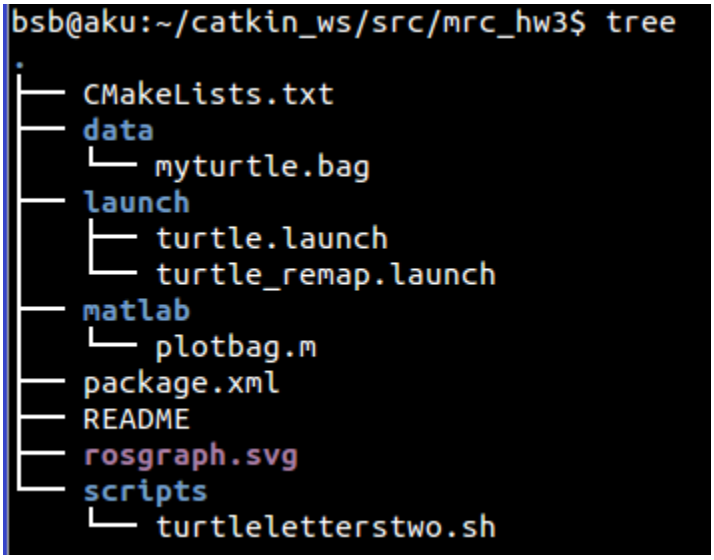
Following the Working with rosbag Log Files example, develop a script called **plotbag.m** and put it in a directory of your package called **matlab**. Your script will need to do the following...

1. Load the **myturtle.bag** file
2. Select the messages on the turtle1/pose topic
3. Convert these pose messages to a timeseries
4. Generate the two figures. (Make sure to include labels, a title and units for your figures.)

To complete the exercise, add your **plotbag.m** file to your git repository, commit and push.

Summary

After completing all the exercises, your ROS package should look something like this...



It is fine if there are other, extra files, but the files illustrated above are what you are asked to submit via git.