Step 4.1 - Integrating your New Package into the Duckietown Software Infrastructure

Packages

Make the package according to Setup Step 4 - Creating your own ROS package. We assume here that <package_name> and <package_path> are set according to the conventions in Setup Step 4.

Nodes

node names should end with **_node**
Note: a package can contain multiple nodes that should be connected within a theme.

As a general rule, for each node there should also be a test stub node: e.g.
<tester_node_name> **=** <package_name>**_tester_node**
the default node (i.e. if you have a package with just one node) name should be
<node_name> = <package_name>**_node**

Launch Files

Let’s define three paths:

- <package_launch_path>: is the path to the launch folder **within a package.** The launch files in here are “elemental” - they only launch the nodes in the package.
- <duckietown_launch_path>: is the path to the launch folder that contains **system level launch scripts**. The launch files in here call a bunch of different “elemental” launch files to perform a certain functionality:
    - example: the joystick.launch file in the duckietown runs 4 nodes: joy, joy_mapper_node, wheels_trimmer_node, and wheels_driver_node
- <duckietown_test_launch_path>: is the path to the **unit test** that tests the functionality of the node.

These paths are set differently depending on whether you are doing core development or completing a student module for the class:

Core development:

```
    <package_launch_path> = $(DUCKIETOWN_ROOT)/catkin_ws/src/<package_name>/launch
    <duckietown_launch_path> = $(DUCKIETOWN_ROOT)/catkin_ws/src/duckietown/launch
    <duckietown_test_launch_path> = $(DUCKIETOWN_ROOT)/catkin_ws/src/duckietown_unit_test
```

Student modules:

```
    <package_launch_path> = $(DUCKIETOWN_ROOT)/catwin_ws/src/spring2016/<handle>/<package_name>_<handle>/launch
    <duckietown_launch_path> = $(DUCKIETOWN_ROOT)/catwin_ws/src/spring2016/<handle>/duckietown_<handle>/launch
    <duckietown_test_launch_path> = $(DUCKIETOWN_ROOT)/catwin_ws/src/spring2016/<handle>/duckietown_unit_test_<handle>/launch
```

These paths are assumed to be set in the following.

Each **node** ideally has a minimum of **4** associated launch files if you are doing things right:

1) **The elemental launch file:** the one that actually calls the node:
   `<package_launch_path>/<node_name>.launch`

Rules:

- no tunable parameters
- **No remapping** in elemental launch file
- Should have this at the top (after the <launch> tag):

```
<arg name="pkg_name" value=”<package_name>”/>
<arg name="veh"  doc="Name of vehicle. ex: megaman"/>
<arg name="local" default="true" doc="true for running everything you possibly can on laptop. false for everything running on vehicle."/>
<arg name="config" default="baseline" doc="Specify a config."/>
<arg name="param_file_name" default="default" doc="Specify a param file. ex:megaman." />
```

- Should have a <rosparam> tag in the <node> tag to load the parameters (see section in this doc on Configurations):

```
<rosparam command="load" file="$(find
<duckietown_package_name>)/config/$(arg config)/$(arg pkg_name)/$(arg
node_name)/$(arg param_file_name).yaml"/>
```

where <duckietown_package_name> = duckietown for core development and
<duckietown_package_nmae> = duckietown_<handle> for student modules

Example:

```
      $(DUCKIETOWN_ROOT)/catkin_ws/src/lane_filter/launch/lane_filter
_node.launch
```

2) The **test stub** launch file: feeds the necessary inputs to the node to test basic functionality

<package_launch_path>/<tester_node_name>.launch

Note:

- if fake stimulus is not available for your node, you can skip this test

Example:

```
      $(DUCKIETOWN_ROOT)/catkin_ws/src/lane_filter/launch/lane_filter
_tester_node.launch
```

3) The **system** launch file

```
      <duckietown_launch_path>/<node_name_without_node>.launch
```

(Note: if node name is <package_name>_node then <node_name_without_node> is simply <package_name>)

Rules:

- When you run it the something actually works
- launches node + every dependencies below it
- no <node> tags are allowed - only <include>
- at a minimum it should include <package_launch_path>/node_name.launch

Remappings:

- Node subscriptions and publications are done relatively with a ~. By default this will subscribe or publish to <what_ever_namespace_the_node_is_under>/node_name/topic_name.
- By convention, in general we remap the subscribers. As a rule, for every topic subscription in your node you should have one line like:

```
<remap from="<subscribing_node>/<topic_name>" to="<publishing_node>/<topic_name>"/>
```

- In general <topic_name> will be the same in both cases but not always.
- The only exception of the “remap the subscriber” rule is topics for driver nodes. For example: wheels_driver/wheels_cmd should never be remapped.

Example:

```
    $(DUCKIETOWN_ROOT)/catkin_ws/src/duckietown/launch/lane_filter.launch
```

4) The **unit test** launch file: the file that uses a logged bag to test the functionality of the node `<duckietown_test_launch_path>/<node_name_without_node>_test`**`X`**`.launch` where X is the test number in the Testing everything document (if it's not in the document then just omit).

Rules:

- Should only have one include which is the system launch file
- should accept arguments **bagin** and **bagout**

Example:

```
$(DUCKIETOWN_ROOT)/catkin_ws/src/duckietown_unit_tests/launch/lane_filter_test2.launch
```

Messages

Core development:

All messages should go in:

```
$(DUCKIETOWN_ROOT)/catkin_ws/duckietown_msgs/msg
```

Student modules:

in a "private" duckietown_**<handle>**_msg package, in

```
$(DUCKIETOWN_ROOT)/catkin_ws/src/spring2016/<handle>/duckietown_msg_<handle>/msg
```

Rules:

- Every message should start with a header:
  `Header header`

Configurations

Core development:

```
    <duckietown_config_path> =
$(DUCKIETOWN_ROOT)/catkin_ws/src/duckietown/
```

Student modules:

```
    <duckietown_config_path> =
$(DUCKIETOWN_ROOT)/catwin_ws/src/spring2016/<handle>/duckietown_<hand
le>/
```

All configurations should go in:

```
<duckietown_config_path>/config/baseline
```

Car related parameters (e.g. calibrations) should go in:

```
<duckietown_config_path>/config/baseline/calibration
```

For nodes that have parameters to be loaded (which should be almost every one), they should go in:

```
<duckietown_config_path>/config/baseline/<package_name>/<node_n
ame>
```

In this folder there should be at least one file: default.yaml

Logs

- It is not strictly enforced, but the convention is that the logs are in $(DUCKIETOWN_ROOT)/duckietown-data (symlink)

```
$ cd $(DUCKIETOWN_ROOT)
$ ln -s <path_to_dropbox_logs_folder> duckietown_data
```