Task 0: Recording Logs

- ROS allows us to record a log of the messages that occurred in a given time period. This is like recording a data stream. The ROS utility which does this is called rosbag, and the command to capture the data is `rosbag record` .
- Create a folder called bag_files in your package (`pkg_task0/bag_files`) as a saving destination for the generated bag files.
- You can run the `rosbag record` command separately on the command line. But to not loose any data you will have to start recording precisely at the same moment your turtle starts moving. Hence it is preferred to include the `rosbag` recording node in your launch file itself.
- Add the following lines (if not already) to your launch file to have the `rosbag` record run parallel with your task. Some parameters are explained below:

```
<arg name="record" default="false"/>
<arg name="duration" default="20"/>
<arg name="rec_name" default="turtle_record.bag"/>

<group if="$(arg record)">

    <node name="rosbag_record_turtle" pkg="rosbag" type="record"
       args="record -O $(find pkg_task0)/bag_files/$(arg rec_name) --duration=$(arg durat

</group>
```

Note: Make sure you have added these line before `</launch>` syntax at the end of the code. And the 2 nodes, the `turtlesim` and your python node, should as well be within this `launch` syntax, for desire recording.

- The `arg_name` tags are `roslaunch` parameters, meaning they can be called while calling your `roslaunch` file, for example:

```
roslaunch pkg_task0 task0.launch record:=true rec_name:=my_turtle.bag
```

- This command will..
  - Start recording.
  - Should start `turtlesim` node.
  - Should start your python script node.
  - And name the resultant bag file as *my_turtle.bag*.

while still retaining the default value (in seconds) for the `duration` parameter.

Thus using these parameters along with your launch file will record a bag file of appropriate `duration` . Also, **make sure that the turtle has correctly covered the desired revolution pattern within the recording `duration` . If not then decrease the radius of curvature.**

- After the recording duration, a message similar as mentioned below will appear on your terminal within the `roslaunch` output window. You may also refer the cursor at the end of the Expected Output and verify.

```
[rosbag_record_turtle-4] process has finished cleanly
```

IMPORTANT: bag files with the same name will be overwritten by the `rosbag` utility without a prompt or warning. Make sure you provide proper name for each iteration if you want to save them all.

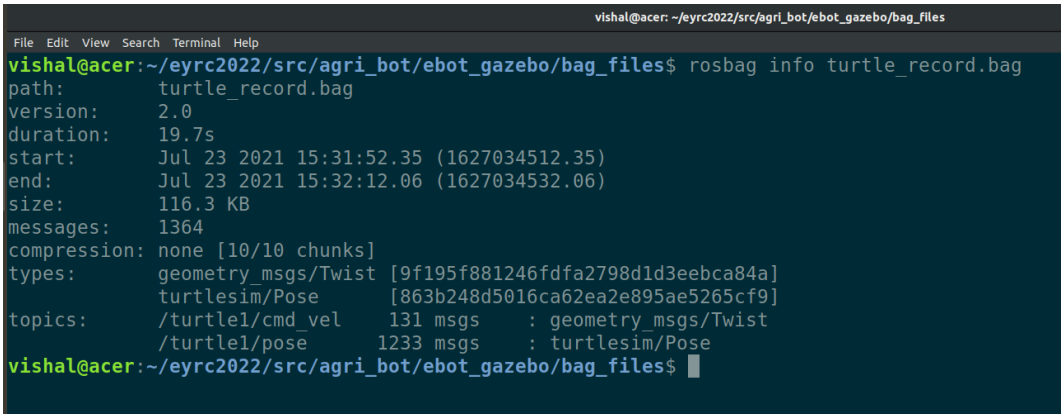
- Verify that your bag file is properly recorded by using the `rosbag info` command followed by the absolute or relative path of the file. To do so, enter the following command...
    - [OPTIONAL] To navigate to the `bag_files` folder.

      ```
      roscd pkg_task0/bag_files
      ```

    - To verify...

      ```
      rosbag info <NameOfBagFile>.bag
      ```

      Verify that the desire topics, like `/turtle1/cmd_vel` & `/turtle1/pose` are mentioned.

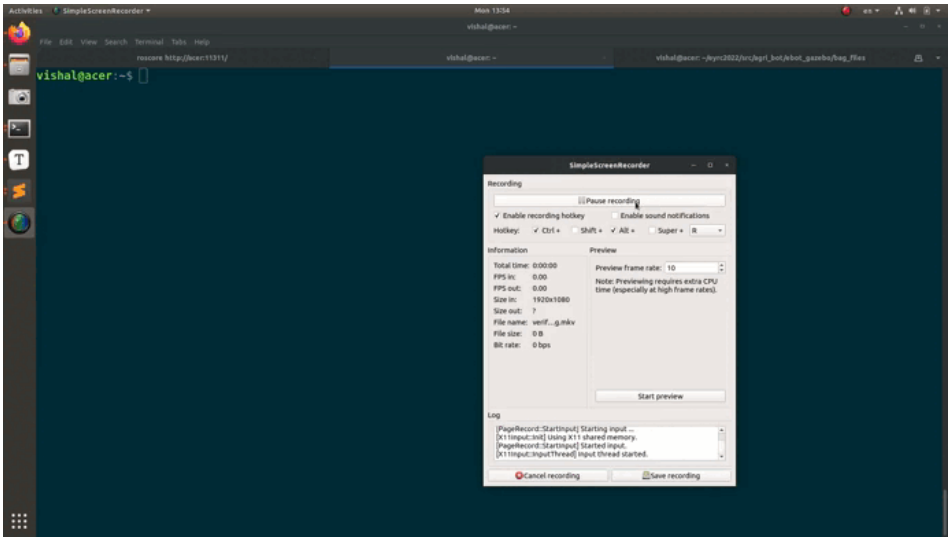


- You can use the `rosbag play` command to see how the messages were recorded. You can verify this by...
    - running the `turtlesim_node` in a new terminal, considering `roscore` is running separately

      ```
      rosrun turtlesim turtlesim_node
      ```

    - and then play your bag file in another terminal.

      ```
      rosbag play <NameOfBagFile>.bag
      ```



Next Read, Making the submission.