TurtleSim Demo.

https://www.youtube.com/watch?v=tNqYDqC6wo4&t=92s&ab_channel=ThomasCarstens

At this stage:

USING A STATE MACHINE TO CONTROL THE SEQUENCE

Splitting the robot motion into states made it possible to link ROS data to a state machine.

- MY ANNOTATED CODE

LINK TO THE TURTLESIM TUTORIALS

USING A STATE MACHINE TO CONTROL A SEQUENCE OF ACTIONS

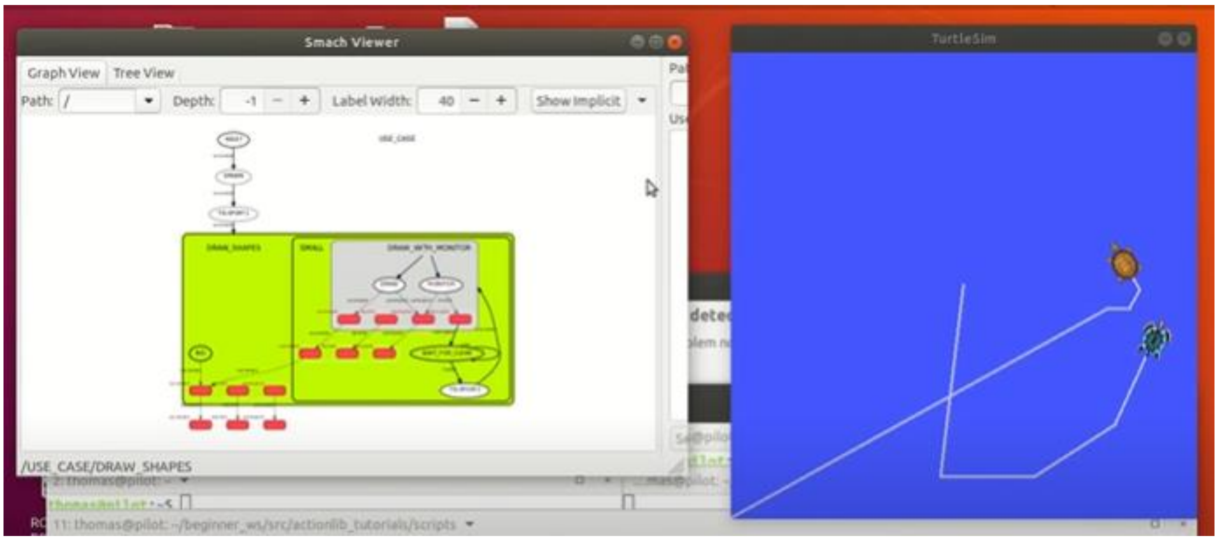
SETTING UP A STATE MACHINE IN A PROJECT

Choosing the state machine

My experience previously consisted of designing **digital electronics** state machines. I went on a search for the best framework for my particular case. The different programs that I reviewed were ROSPlan, RSM and Python Smach. The solution needed to work with ROS nodes, but also function independently for the sake of modularity. However, it was not necessary to have a performance-oriented state machine such as ROSPlan, rather one that would be useful for quick prototyping. An added plus would be integration into server-client setups for smooth demoing. The Python SMACH library promised all these elements.

Learning to use the state machine

I followed the official set of tutorials for the Smach documentation. These tutorials work in conjunction with TurtleSim, frequently used to test ROS functionality. The final result of this process is a simulation that is better viewed as a video rather than in pictures, please access it here.



Integrating the state machine into the behaviour script

The behaviour script is available in the github linked in the first page.

Using the behaviour script, we were able to create the demo which you will view on our video.