ROS2 Humble Commands & Turtlesim

Common CLI commands and launching the TurtleSim demo

Introduction to ROS 2 Humble

- Second generation of Robot Operating System with improved performance and security
- Humble Hawksbill is a long-term support (LTS) release (May 2022) widely used for robotics research & development
- Provides command-line tools (`ros2`) for nodes, topics, services, actions and more
- Forms the foundation for exploring simulations like turtlesim

ROS2 CLI Structure & Help

- CLI entry point: `ros2` with structure `ros2 [command] [verb] [args]`
- Use `ros2 --help` to list all available subcommands ([Introspection with command line tools])
- Common commands include `action`, `bag`, `component`, `daemon` and others for interacting with actions, bags, component containers and the ROS daemon ([Introspection with command line tools])
- For detailed help on a command or verb, run `ros2 [command] --help`

Core ROS2 Commands

- List running nodes: `ros2 node list`
- Show available topics: `ros2 topic list` (Understanding topics — ROS 2 Documentation: Humble …)
- Inspect a topic's data: `ros2 topic echo /turtle1/pose` (Understanding topics — ROS 2 Documentation: Humble …)
- List services: `ros2 service list` to view services like /clear, /spawn and /turtle1/set_pen (ROS 2 Commands Cheat Sheet)
- Publish velocity commands: `ros2 topic pub /turtle1/cmd_vel geometry_msgs/msg/Twist "{linear: {x: 0.5, y: 0.0, z: 0.0}, angular: {x:0.0, y:0.0, z:0.5}}"` (ROS Cheat Sheet)

Introducing Turtlesim

- Lightweight simulator illustrating basic ROS 2 concepts (Introspection with command line tools)
- Spawns a turtle with a random design; control it via keyboard teleop or topic publishing
- Ideal for practicing nodes, topics, services and introspection with `ros2` CLI
- rqt provides GUI plugins to spawn additional turtles and call services (Introducing turtlesim and rqt)

Launching Turtlesim

- Start simulator: `ros2 run turtlesim turtlesim_node` spawns the `/turtlesim` node and opens a window with a turtle (Using turtlesim, ros2, and rqt, ROS Cheat Sheet)
- Control via keyboard: in a separate terminal run `ros2 run turtlesim turtle_teleop_key` to create the `/teleop_turtle` node and drive the turtle with arrow keys (ROS Cheat Sheet)
- Keep the teleop terminal focused to ensure key presses are captured
- Use `ros2 node list` or `ros2 topic list` after launch to inspect active nodes and topics
- Press Ctrl+C in each terminal to stop the nodes and close the simulator

Conclusion & Next Steps

- ROS2 CLI offers a consistent interface for managing nodes, topics, services and other entities
- Use turtlesim as a sandbox to practice launching, controlling and introspecting nodes
- Explore additional commands for parameters, actions, bags and tools like `rqt_graph`
- Write your own launch files to coordinate multiple nodes and build repeatable demos
- Progress from turtlesim to real robots or Gazebo simulations for deeper learning