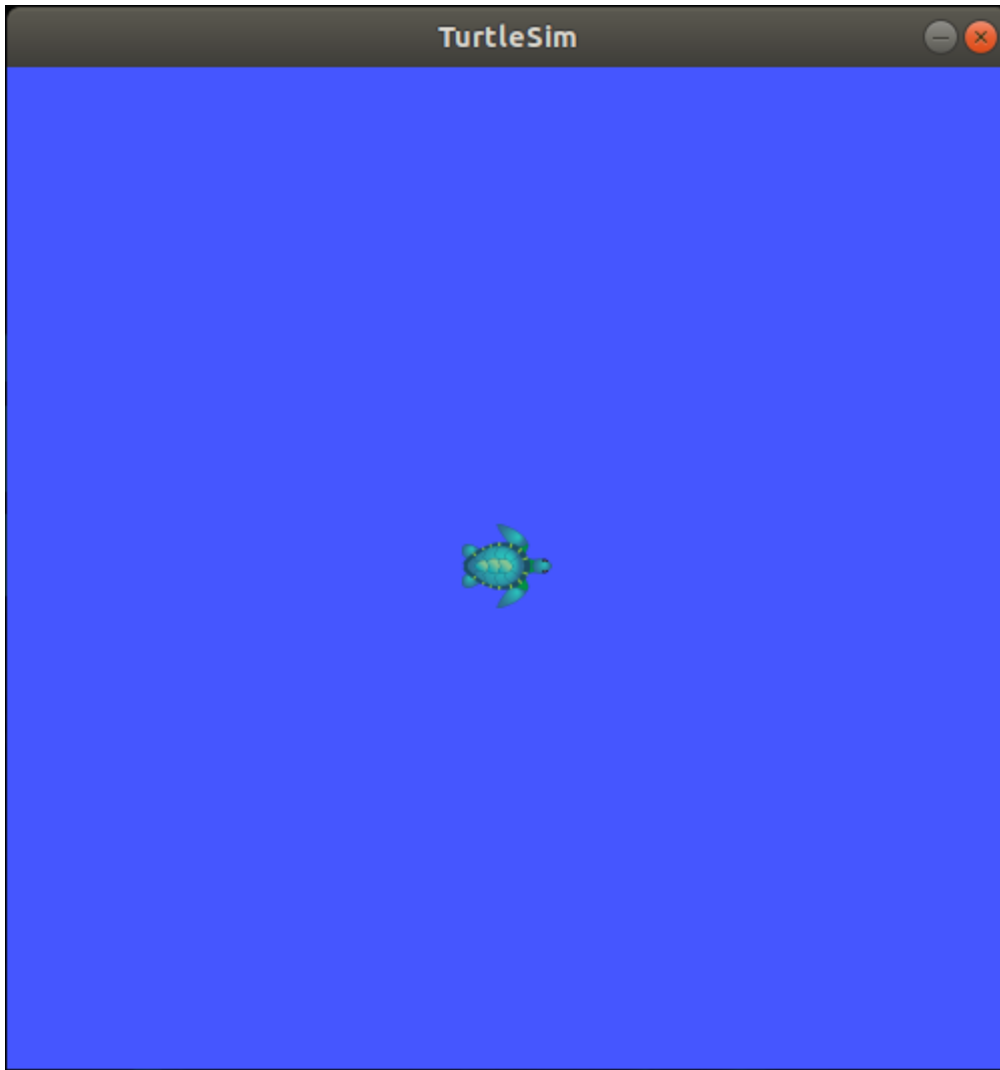


小乌龟节点启动



1. 启动小乌龟模拟器节点

```
1 | rosrn turtlesim turtlesim_node
```

2. 启动小乌龟键盘输入节点

```
1 | rosrn turtlesim turtle_teleop_key
```

!!!warning

启动完成后，可以通过键盘输入操控小乌龟移动。

```
1 | **键盘操控时，光标一定要在命令行上。**
```

小乌龟操控原理

1. 节点信息查看

小乌龟启动过程中，我们启动了两个可执行的程序：`turtlesim_node` 和 `turtle_teleop_key`。可以通过命令查看当前的启动的节点：

```
1 | rosnode list
```

!!!tip

可以查看到启动程序对应的节点 `/turtlesim` 和 `/teleop_turtle`

查看小乌龟节点

通过命令可以查看 `/turtlesim` 节点的详情

```
1 | rosnode info /turtlesim
```

命令运行后，可以看到以下结果：

```
1 | -----  
2 |  
3 | Node [/turtlesim]  
4 | Publications:  
5 |   * /rosout [roscpp_msgs/Log]  
6 |   * /turtle1/color_sensor [turtlesim/Color]  
7 |   * /turtle1/pose [turtlesim/Pose]  
8 |  
9 | Subscriptions:  
10 |   * /turtle1/cmd_vel [geometry_msgs/Twist]  
11 |  
12 | Services:  
13 |   * /clear  
14 |   * /kill  
15 |   * /reset  
16 |   * /spawn  
17 |   * /turtle1/set_pen  
18 |   * /turtle1/teleport_absolute  
19 |   * /turtle1/teleport_relative  
20 |   * /turtlesim/get_loggers  
21 |   * /turtlesim/set_logger_level  
22 |  
23 | contacting node http://ubuntu:42049/ ...  
24 | Pid: 20218  
25 | Connections:  
26 |   * topic: /rosout  
27 |     * to: /rosout  
28 |     * direction: outbound  
29 |     * transport: TCPROS  
30 |   * topic: /turtle1/cmd_vel  
31 |     * to: /teleop_turtle (http://ubuntu:44433/)  
32 |     * direction: inbound  
33 |     * transport: TCPROS
```

!!!note

`rosnode info` 命令可以查看当前节点的一些信息：

- 1 * **Publications:** 此节点上定义的发布者
- 2 * **Subscriptions:** 此节点上定义的订阅者
- 3 * **Services:** 此节点上定义的服务
- 4 * 进程id, 占用的网络端口
- 5 * **Connections:** 此节点和其他节点间的连接信息

查看控制节点

同理, 我们也可以通过 `rostopic info` 查询 `/teleop_turtle` 节点的信息,

```
1 | rostopic info /teleop_turtle
```

结果如下:

```
1 | -----  
2 | Node [/teleop_turtle]  
3 | Publications:  
4 |   * /rosout [roscpp_msgs/Log]  
5 |   * /turtle1/cmd_vel [geometry_msgs/Twist]  
6 |  
7 | Subscriptions: None  
8 |  
9 | Services:  
10 |   * /teleop_turtle/get_loggers  
11 |   * /teleop_turtle/set_logger_level  
12 |  
13 |  
14 | contacting node http://ubuntu:44433/ ...  
15 | Pid: 20443  
16 | Connections:  
17 |   * topic: /rosout  
18 |     * to: /rosout  
19 |     * direction: outbound  
20 |     * transport: TCPROS  
21 |   * topic: /turtle1/cmd_vel  
22 |     * to: /turtlesim  
23 |     * direction: outbound  
24 |     * transport: TCPROS
```

现在我们大致可以搞清楚一些通讯的关系:

- `/teleop_turtle` 节点存在一个发布者, 往 `/turtle1/cmd_vel` 主题中发布数据。
- `/turtlesim` 节点存在一个订阅者, 去 `/turtle1/cmd_vel` 主题中获取数据。

2. 可视化工具查询节点关系

`rqt_graph` 工具提供了可视化的工具方便我们查看这种节点间的关系:

```
1 | roslaunch rqt_graph rqt_graph
```



!!!note

图像显示, `/teleop_turtle` 通过主题 `/turtle1/cmd_vel` 给 `/turtlesim` 进行数据传递。

```
1  `/teleop_turtle`为具备Publisher功能的节点。
2
3  `/turtlesim`为具备Subscriber功能的节点。
4
5  `/turtle1/cmd_vel`为publisher和subscriber通讯的主题。
```

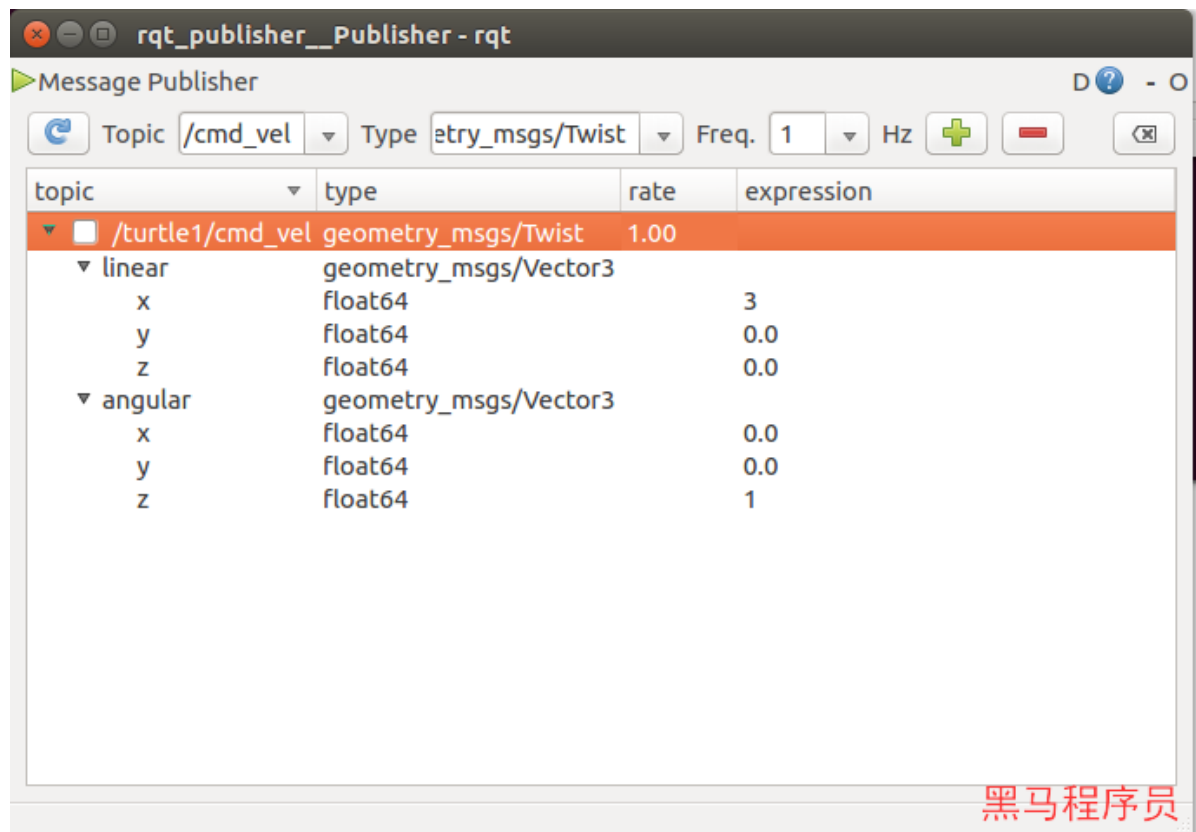
调试工具操控小乌龟

1. `rqt_publisher` 模拟数据发送

启动 `rqt_publisher` 工具

```
1 | rosrn rqt_publisher rqt_publisher
```

通过图形化配置参数:



2. 通过命令行模拟数据发送

```
1 | rostopic pub /turtle1/cmd_vel geometry_msgs/Twist "linear:
2 |   x: 1.0
3 |   y: 0.0
4 |   z: 0.0
5 | angular:
6 |   x: 0.0
7 |   y: 0.0
8 |   z: 3.0"
```

小乌龟坐标系相关

面板

- 面板的坐标原点在左下脚，即左下角为(0,0)
- 面板的X轴是自左向右，数值是0开始正向增长
- 面板的Y轴是自下向上，数值是0开始正向增长
- 面板的宽度和高度相同，值为11.088899

小乌龟

- 小乌龟的坐标原点为小乌龟的中心点。

小乌龟移动指令参数

我们通过rostopic命令可以获得小乌龟的移动数据类型为 `geometry_msgs/Twist`

```
1 | rostopic type /turtle1/cmd_vel
```

通过rosmmsg命令可以查看数据的详细格式:

```
1 | rosmmsg show geometry_msgs/Twist
```

输出的格式为:

```
1 | geometry_msgs/Vector3 linear
2 |   float64 x
3 |   float64 y
4 |   float64 z
5 | geometry_msgs/Vector3 angular
6 |   float64 x
7 |   float64 y
8 |   float64 z
```

!!!tip

`geometry_msgs/Vector3 linear` 指的是线速度，小乌龟只用到了 `float64 x`，代表着乌龟向前进的线速度

```
1 | `geometry_msgs/Vector3 angular` 指的是角速度，小乌龟只用到了 `float64 z`，代表着乌龟旋转的角速度
```