

# JasonLLLL-第三章作业

## 1. 创建一个节点，在其中实现一个publisher和一个subscriber

### 1.1. 创建功能包 HW\_Chapter3

- 1.1.1. `cd ~/catkin_ws/src`
- 1.1.2. `catkin_create_pkg HW_chapter3 roscpp rospy std_msgs std_srvs`
- 1.1.3. 打开新Terminal `cd ~/catkin_ws/`
- 1.1.4. `catkin_make`(进行编译)
- 1.1.5. `source ~/catkin_ws/devel/setup.bash` (配置环境变量,也可将该句添加到 `.bashrc`文件中)

### 1.2. 创建publisher，发布速度指令，让海龟作圆周运动

- 1.2.1. 代码详情见文件: `turtle1_vel_publisher.py`
  - 1.2.1.1. `pwd: /catkin_ws/src/HW_Chapter3/src`
- 1.2.2. 添加文件权限 `chmod +x turtle1_vel_publisher.py`

### 1.3. 创建subscriber，订阅海龟位置信息，并在Terminal中打印输出

- 1.3.1. 代码详情见文件: `turtle1_vel_subscriber.py`
  - 1.3.1.1. `pwd: /catkin_ws/src/HW_Chapter3/src`
- 1.3.2. 添加文件权限 `chmod +x turtle1_vel_subscriber.py`

### 1.4. 打开新Terminal(只在cpp编写publisher和subscriber后使用，python编写不需要)

- 1.4.1. `cd ~/catkin_ws`
- 1.4.2. `catkin_make`(编译)

### 1.5. 运行文件

- 1.5.1. 启动ROS master
  - 1.5.1.1. 打开新Terminal
  - 1.5.1.2. `Roscore`
- 1.5.2. 打开海龟仿真器
  - 1.5.2.1. `roslaunch turtlesim turtlesim_node`
- 1.5.3. 运行 `turtle1_vel_publisher.py`
  - 1.5.3.1. 打开新Terminal
  - 1.5.3.2. `roslaunch HW_Chapter3 turtle1_vel_publisher.py`
  - 1.5.3.3. 运行结果

```
jingsheng@jslyuUB18: ~
File Edit View Search Terminal Help
[INFO] [1583957101.364390]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957101.464311]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957101.564347]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957101.664366]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957101.764455]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957101.864426]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957101.964393]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957102.064422]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957102.164395]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957102.264498]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957102.364675]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957102.464618]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957102.564511]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957102.664538]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957102.764533]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957102.864535]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957102.964534]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957103.064528]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957103.164579]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957103.264478]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957103.364641]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957103.464723]: Turtle velocity [0.50 m/s, 0.50 rad/s]
[INFO] [1583957103.564266]: Turtle velocity [0.50 m/s, 0.50 rad/s]
```

#### 1.5.4. 运行turtle1\_vel\_subscriber.py

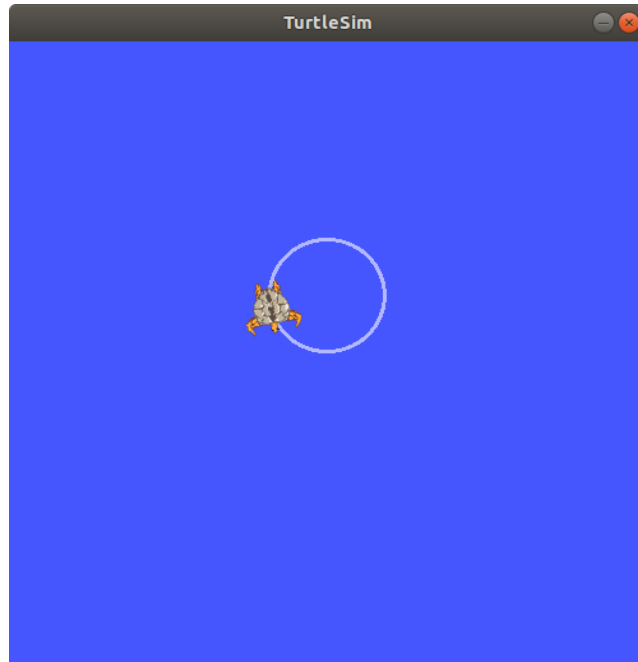
1.5.4.1. 打开新Terminal

1.5.4.2. rosrn HW\_Chapter3 turtle1\_vel\_subscriber.py

1.5.4.3. 运行结果

```
jingsheng@jslyuUB18: ~
File Edit View Search Terminal Help
[INFO] [1583957350.448536]: Turtle1 pose: x:6.030335, y:7.416226
[INFO] [1583957350.464210]: Turtle1 pose: x:6.023346, y:7.420117
[INFO] [1583957350.479840]: Turtle1 pose: x:6.016325, y:7.423952
[INFO] [1583957350.495663]: Turtle1 pose: x:6.009274, y:7.427731
[INFO] [1583957350.512336]: Turtle1 pose: x:6.002192, y:7.431454
[INFO] [1583957350.528662]: Turtle1 pose: x:5.995081, y:7.435119
[INFO] [1583957350.544468]: Turtle1 pose: x:5.987942, y:7.438727
[INFO] [1583957350.560154]: Turtle1 pose: x:5.980773, y:7.442279
[INFO] [1583957350.576196]: Turtle1 pose: x:5.973577, y:7.445773
[INFO] [1583957350.592108]: Turtle1 pose: x:5.966352, y:7.449209
[INFO] [1583957350.608046]: Turtle1 pose: x:5.959100, y:7.452587
[INFO] [1583957350.624086]: Turtle1 pose: x:5.951822, y:7.455907
[INFO] [1583957350.639623]: Turtle1 pose: x:5.944517, y:7.459169
[INFO] [1583957350.655464]: Turtle1 pose: x:5.937186, y:7.462372
[INFO] [1583957350.671201]: Turtle1 pose: x:5.929830, y:7.465517
[INFO] [1583957350.687804]: Turtle1 pose: x:5.922449, y:7.468603
[INFO] [1583957350.704183]: Turtle1 pose: x:5.915044, y:7.471629
[INFO] [1583957350.719529]: Turtle1 pose: x:5.907614, y:7.474596
[INFO] [1583957350.735586]: Turtle1 pose: x:5.900161, y:7.477504
[INFO] [1583957350.751857]: Turtle1 pose: x:5.892685, y:7.480351
[INFO] [1583957350.768413]: Turtle1 pose: x:5.885187, y:7.483140
[INFO] [1583957350.783970]: Turtle1 pose: x:5.877666, y:7.485868
[INFO] [1583957350.800796]: Turtle1 pose: x:5.870124, y:7.488535
```

## 1.6. 海龟圆周运动结果



## 2. 创建另外一个节点，在其中实现一个客户端

### 2.1. 创建新节点

2.1.1. touch newTurtle.py

2.1.2. 代码细节见: newTurtle.py

2.1.2.1. pwd: /catkin\_ws/src/HW\_Chapter3/src

2.1.3. 添加文件权限

2.1.3.1. chmod +x newTurtle.py

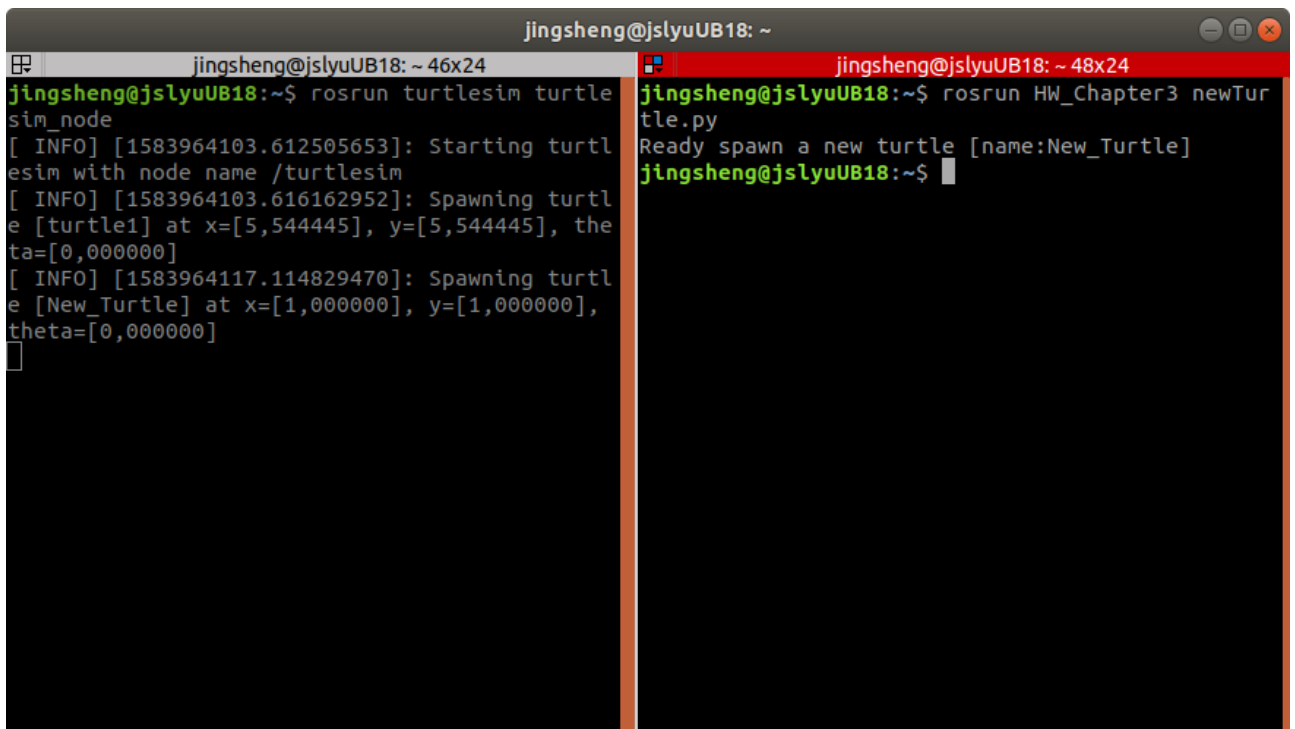
### 2.2. 打开海龟仿真器

2.2.1. rosrun turtlesim turtlesim\_node

### 2.3. 运行roslaunch

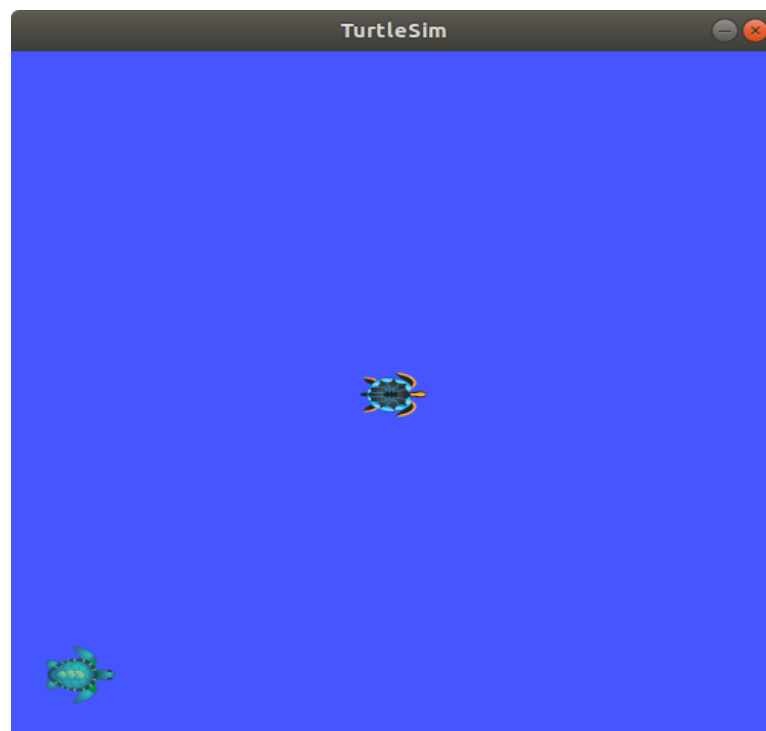
2.3.1. roslaunch HW\_Chapter3 newTurtle.py

## 2.4. Terminal运行结果



```
jingsheng@jslyuUB18: ~  
jingsheng@jslyuUB18: ~ 46x24  
jingsheng@jslyuUB18:~$ rosrn turtlesim turtle  
sim_node  
[ INFO] [1583964103.612505653]: Starting turtl  
esim with node name /turtlesim  
[ INFO] [1583964103.616162952]: Spawning turtl  
e [turtle1] at x=[5,544445], y=[5,544445], the  
ta=[0,000000]  
[ INFO] [1583964117.114829470]: Spawning turtl  
e [New_Turtle] at x=[1,000000], y=[1,000000],  
theta=[0,000000]  
█  
jingsheng@jslyuUB18: ~ 48x24  
jingsheng@jslyuUB18:~$ rosrn HW_Chapter3 newTur  
tle.py  
Ready spawn a new turtle [name:New_Turtle]  
jingsheng@jslyuUB18:~$ █
```

## 2.5. 海龟仿真器运行结果



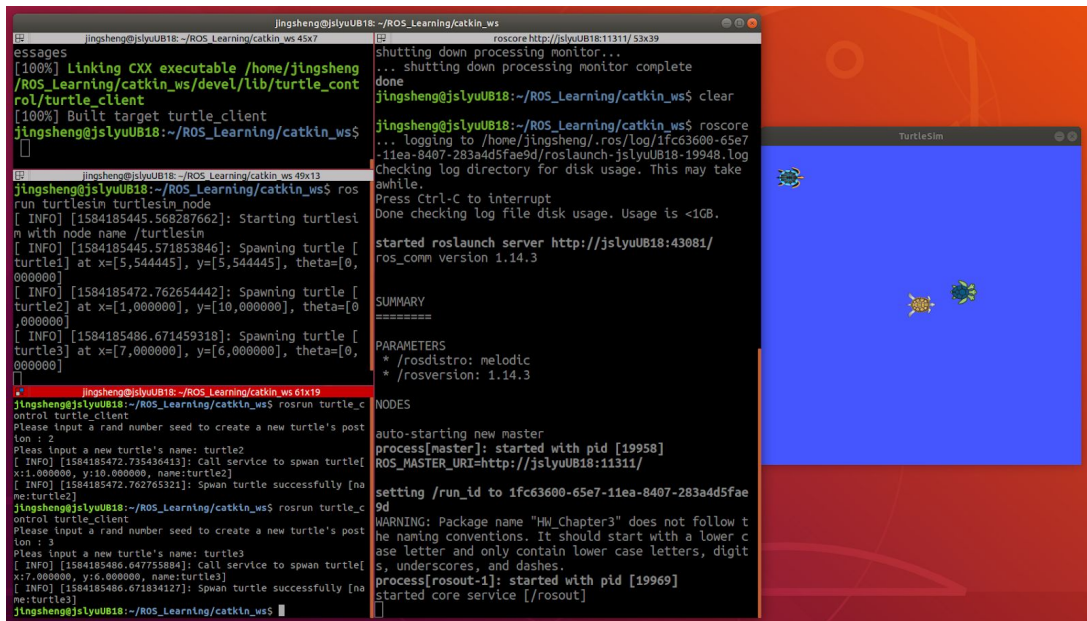
## 3. 综合运用

### 3.1. 创建海龟控制功能包

- 3.1.1. 打开新Terminal
- 3.1.2. 进入工作空间中的src目录 `cd ~/catkin_ws/src`
- 3.1.3. `catkin_create_pkg turtle_control roscpp rospy std_msgs std_srvs turtlesim`
- 3.1.4. 回到catkin\_ws目录 `cd ..`
- 3.1.5. 编译 `catkin_make`
- 3.1.6. 设置环境变量 `source devel/setup.bash`

### 3.2. 通过命令行产生新海龟，且位置不重叠

- 3.2.1. 新开Terminal, 进入海龟控制功能包
  - 3.2.1.1. `cd ~/catkin_ws/src/turtle_control`
- 3.2.2. 进入src目录，使用C++编写turtle\_client实现在命令行产生新海龟
  - 3.2.2.1. `touch turtle_client.cpp`
  - 3.2.2.2. 代码细节见: `turtle_client.cpp`
  - 3.2.2.3. `pwd: ~/catkin_ws/src/turtle_control/src`
- 3.2.3. 在CMakeList.txt中添加编译选项
  - 3.2.3.1. `add_executable(turtle_client src/turtle_client.cpp)`
  - 3.2.3.2. `target_link_libraries(turtle_client ${catkin_LIBRARIES})`
- 3.2.4. 新开Terminal, 编译
  - 3.2.4.1. `cd ~/catkin_ws`
  - 3.2.4.2. `catkin_make`
    - 3.2.4.2.1. 在catkin\_ws/devel/lib/turtle\_control目录下产生turtle\_client的可执行文件
- 3.2.5. 新开Terminal, 运行roscore
- 3.2.6. 启动海龟仿真器
  - 3.2.6.1. `roslaunch turtlesim turtlesim_node`
- 3.2.7. 运行turtle\_client
  - 3.2.7.1. `roslaunch turtle_control turtle_client`
  - 3.2.7.2. 输入随机数，产生随机坐标；输入海龟名字，产生新海龟
- 3.2.8. 最终结果如图



### 3.3. 通过命令行控制界面任意海龟的启动和停止，速度通过命令行控制

3.3.1. 新开Terminal，进入~/catkin\_ws/src/turtle\_control目录

3.3.2. 新建srv目录: mkdir srv, 然后进入srv目录: cd srv

3.3.3. 新建自定义 .srv文件

3.3.3.1. touch SpawnTurtle.srv

3.3.3.2. gedit SpawnTurtle.srv

3.3.3.3. 定义海龟名字，位置(x,y)，角度  
string tur\_name  
float64 pose\_x  
float64 pose\_y  
float64 tur\_theta

---

string result

3.3.4. 在package.xml中添加依赖

3.3.4.1. <build\_depend>message\_generation</build\_depend>

3.3.4.2. <exec\_depend>message\_runtime</exec\_depend>

3.3.5. 在CMakeLists.txt中添加编译选项

3.3.5.1. 在find\_package处添加message\_generation

```
10 find_package(catkin REQUIRED COMPONENTS
11   roscpp
12   rospy
13   std_msgs
14   std_srvs
15   turtlesim
16   message_generation
17 )
```

### 3.3.5.2. 添加其他编译选项

3.3.5.2.1. add\_service\_files(FILE SpawnTurtle.srv)

3.3.5.2.2. generate\_messages(DEPENDENCIES std\_msgs)

### 3.3.5.3. 打开注释, 在catkin\_package处添加message\_runtime

```
111 catkin_package(  
112#   INCLUDE_DIRS include  
113#   LIBRARIES turtle_control  
114   CATKIN_DEPENDS roscpp rospy std_msgs std_srvs turtlesim message_runtime  
115#   DEPENDS system_lib  
116)
```

### 3.3.6. 回到~/catkin\_ws

### 3.3.7. 编译: catkin\_make

3.3.7.1. 在~/catkin\_ws/devel/include/turtle\_control中产生头文件

3.3.7.2. SpawnTurtle.h, SpawnTurtleRequest.h和  
SpawnTurtleResponse.h

### 3.3.8. 进入~/catkin\_ws/src/turtle\_control/src, 进行turtle\_server.cpp编程

3.3.8.1. 代码细节见: turtle\_server.cpp

3.3.8.2. pwd: ~/catkin\_ws/src/turtle\_control/src

### 3.3.9. 配置CMakeLists.txt编译规则

3.3.9.1. add\_executable(turtle\_server src/turtle\_server.cpp)

3.3.9.2. target\_link\_libraries(turtle\_server \${catkin\_LIBRARIES})

### 3.3.10. 新开Terminal, 进入~/catkin\_ws进行编译: catkin\_make

### 3.3.11. 新开Terminal, 运行roscore

### 3.3.12. 新开Terminal, 打开海龟仿真器

3.3.12.1. rosrn turtlesim turtlesim\_node

### 3.3.13. 新开Terminal, 通过3.1问的turtle\_client添加新海龟

3.3.13.1. rosrn turtle\_control turtle\_client

### 3.3.14. 新开Terminal, 运行server

3.3.14.1. rosrn turtle\_control turtle\_server

### 3.3.15. 新开Terminal, 通过service服务, 来调用产生的server, Trigger数据类型为"{}"

3.3.15.1. rosservice call /turtle\_server "{}"

### 3.3.16. 在server端输入想控制海龟的名字和速度

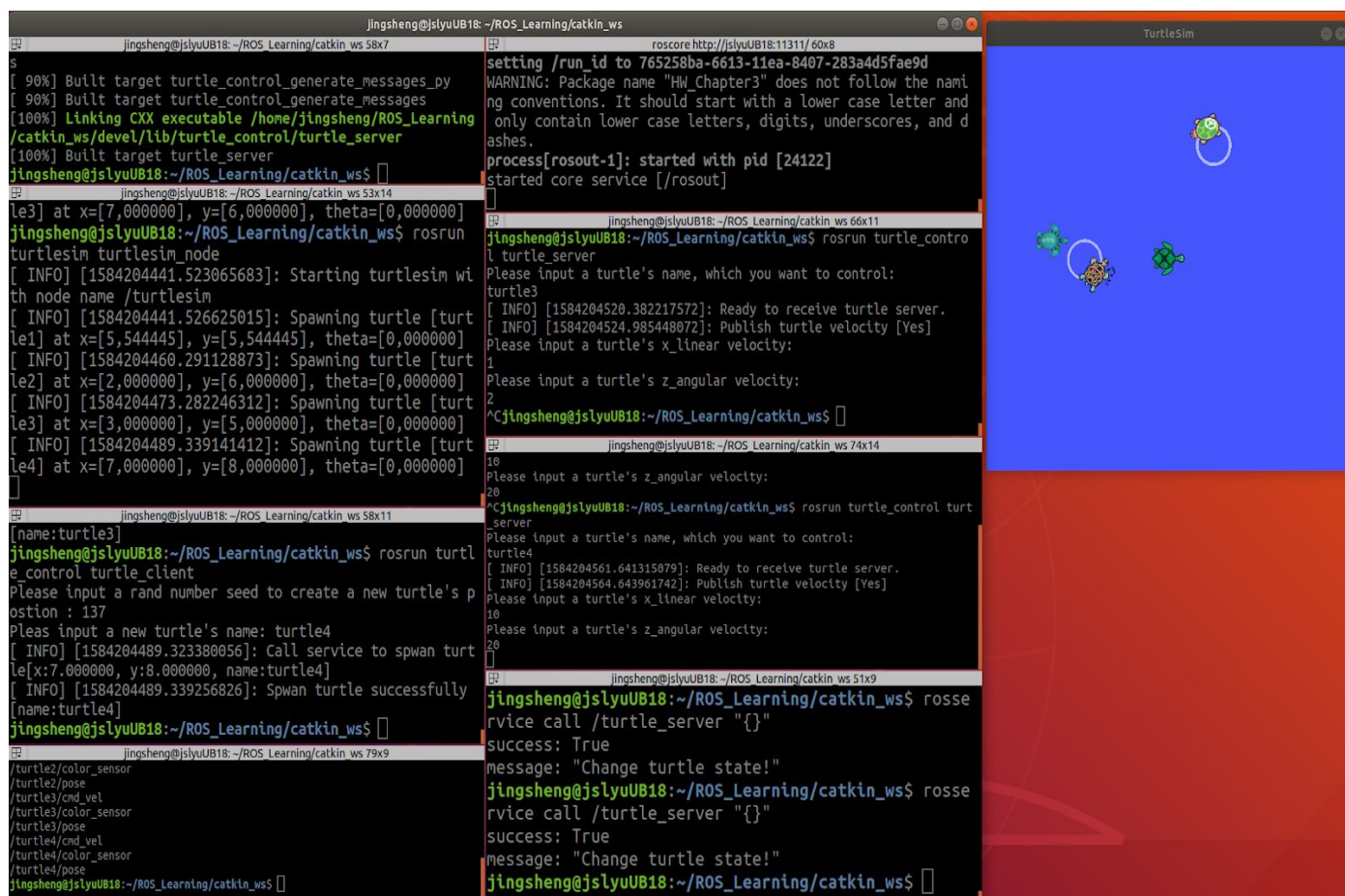
3.3.16.1. Ctrl + C 可以控制海龟的启动和停止

3.3.16.2. 在server端输入想控制的乌龟名字, 可以改变控制的乌龟

3.3.16.3. 通过改变x-axis的线速度和z-axis的角速度, 可以控制海龟进行  
圆周运动的速率

### 3.3.17. 最终结果如图





## 4. Reference

- 4.1. <http://docs.ros.org/melodic/api/rospy/html/>
- 4.2. <https://www.bilibili.com/video/av59458869>
- 4.3. 胡春旭 《ROS机器人开发实践》