

Turtlesim Noetic gotogoal_1 10_13_2022

Run Corrected python gotogoal_1; Make Executable

Alias foxy or noetic

```
harman@harman-VirtualBox:~$ noetic
```

ROS_DISTRO was set to 'foxy' before. Please make sure that the environment does not mix paths from different distributions.

```
harman@harman-VirtualBox:~$ cd Desktop/
```

```
harman@harman-VirtualBox:~/Desktop$ ls -la | grep go
```

```
-rw-rw-r-- 1 harman harman 2681 Oct 13 18:56 gotogoal_1.py
```

```
-rwxrwx--- 1 harman harman 2668 Sep 2 2019 gotogoal.py
```

```
harman@harman-VirtualBox:~/Desktop$ chmod +x gotogoal_1.py
```

```
harman@harman-VirtualBox:~/Desktop$ ls -la | grep go
```

```
-rwxrwxr-x 1 harman harman 2681 Oct 13 18:56 gotogoal_1.py
```

```
-rwxrwx--- 1 harman harman 2668 Sep 2 2019 gotogoal.py
```

DEMO START: RUN ROSCORE - Terminal 1

Alias foxy or noetic

```
harman@harman-VirtualBox:~$ noetic
```

```
harman@harman-VirtualBox:~$ roscore
```

RUN TURTLESIM_NODE – Terminal 2

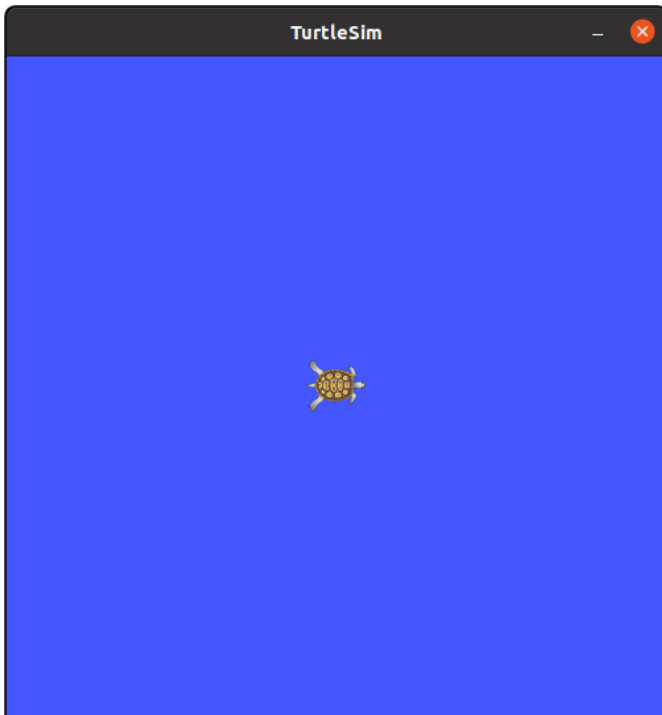
Alias foxy or noetic

```
harman@harman-VirtualBox:~$ noetic
```

```
harman@harman-VirtualBox:~$ rosruntime turtlesim turtlesim_node
```

```
[ INFO] [1665705763.615104048]: Starting turtlesim with node name /turtlesim
```

```
[ INFO] [1665705763.619462117]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]
```



Run Modified Python 3 gotogoal_1 script – Terminal 3

Alias foxy or noetic

harman@harman-VirtualBox:~\$ noetic

harman@harman-VirtualBox:~\$ cd Desktop/

harman@harman-VirtualBox:~/Desktop\$ **python3 gotogoal_1.py**

Set your x goal:1.0

Set your y goal:1.0

Set your tolerance:0.2

velocity = 9.64012816927244

Angular Velocity = -9.42477796076938

The answer is 4

x = 5.5444

y = 5.5444

velocity = 9.64012816927244

Angular Velocity = -9.42477796076938

The answer is 4

x = 6.3177

y = 5.0937

velocity = 10.066364994624426

Angular Velocity = -6.322979837749973

The answer is 4

x = 6.6004

y = 4.3484

velocity = 9.787573300874941

Angular Velocity = -4.768448004073219

The answer is 4

x = 6.501

y = 3.4221

velocity = 9.015932074527846

Angular Velocity = -3.4337797303370046

..... The turtle keeps coming

The answer is 4

x = 2.4139

y = 1.0471

velocity = 2.122026424199284

Angular Velocity = -0.17455209986889386

The answer is 4

x = 2.2446

y = 1.0355

velocity = 1.8676592763403073

Angular Velocity = -0.1378326458379604

The answer is 4

x = 2.0357

y = 1.0241

velocity = 1.5539705354349547

Angular Velocity = -0.09708479068966724

The answer is 4

x = 1.8867

y = 1.0179

velocity = 1.3303209857023228

Angular Velocity = -0.07212718246709926

The answer is 4

x = 1.759

y = 1.0135

velocity = 1.1386800746917458

Angular Velocity = -0.0540287531559418

..... The turtle keeps coming

The answer is 4

x = 1.4876

y = 1.0068

velocity = 0.7314711204141966

Angular Velocity = -0.02287698749800171

The answer is 4

x = 1.4057

y = 1.0053

velocity = 0.6086019265496946

Angular Velocity = -0.016155273992026764

..... The turtle keeps coming

The answer is 4

x = 1.2973

y = 1.0036

velocity = 0.4459826930498535

Angular Velocity = -0.008900994037956522

The answer is 4

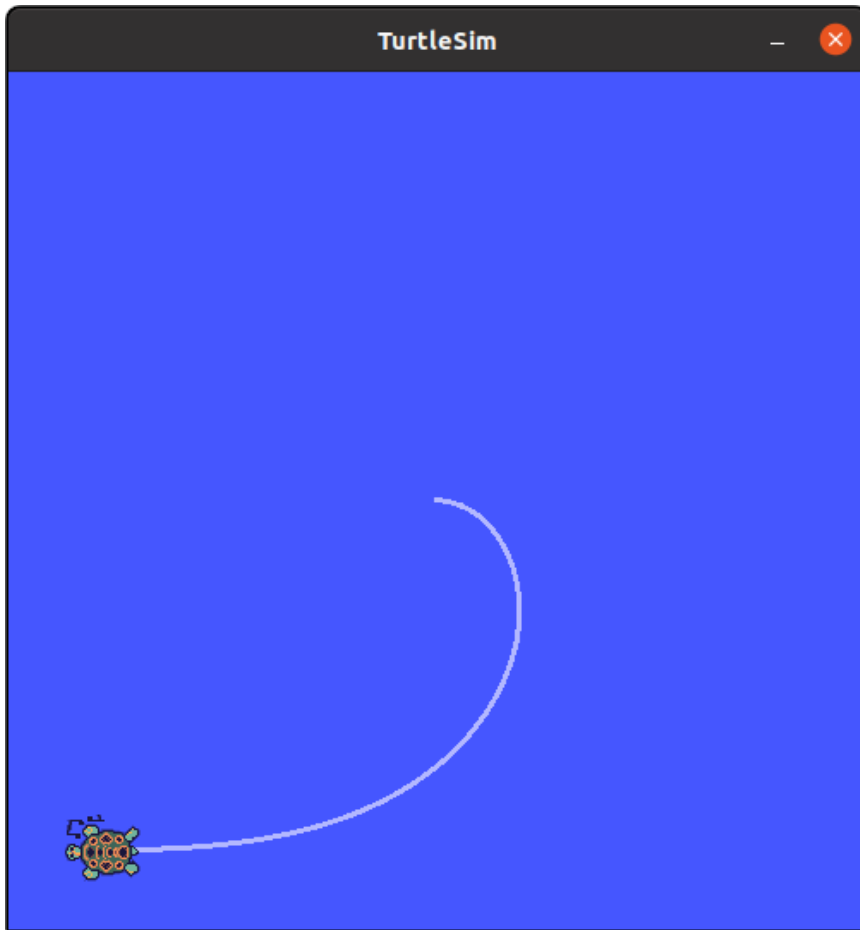
x = 1.2544

y = 1.003

velocity = 0.38162653209649866

Angular Velocity = -0.006748956466601896

```
The answer is 4  
x = 1.2178  
y = 1.0025  
velocity = 0.3267215213297098  
Angular Velocity = -0.005412828671746084  
The answer is 4  
x = 1.1917  
y = 1.0022          (PRETTY CLOSE!)  
^Z  
[1]+ Stopped          python3 gotogoal_1.py  
harman@harman-VirtualBox:~/Desktop$
```



gotogoal_1 Python3 P control 10_13_2022 Corrected

```
#!/usr/bin/env python          gotogoal_1.py Python3 turtlesim_cleaner/src/gotogoal.py GitHub
                                # Added float - float(input("Set your x goal:"))

import rospy
from geometry_msgs.msg import Twist
from turtlesim.msg import Pose
from math import pow,atan2,sqrt

class turtlebot():

    def __init__(self):
        #Creating our node,publisher and subscriber
        rospy.init_node('turtlebot_controller', anonymous=True)
        self.velocity_publisher = rospy.Publisher('/turtle1/cmd_vel', Twist, queue_size=10)
        self.pose_subscriber = rospy.Subscriber('/turtle1/pose', Pose, self.callback)
        self.pose = Pose()
        self.rate = rospy.Rate(10)

    #Callback function implementing the pose value received
    def callback(self, data):
        self.pose = data
        self.pose.x = round(self.pose.x, 4)
        self.pose.y = round(self.pose.y, 4)

    def get_distance(self, goal_x, goal_y):
        distance = sqrt(pow((goal_x - self.pose.x), 2) + pow((goal_y - self.pose.y), 2))
        return distance

    def move2goal(self):
        goal_pose = Pose()
        goal_pose.x = float(input("Set your x goal:")) # Added float to there 10/13/22
        goal_pose.y = float(input("Set your y goal:"))
        distance_tolerance = float(input("Set your tolerance:"))
        vel_msg = Twist()

        while sqrt(pow((goal_pose.x - self.pose.x), 2) + pow((goal_pose.y - self.pose.y), 2)) >=
distance_tolerance:

            #Proportional Controller
            #linear velocity in the x-axis:
            vel_msg.linear.x = 1.5 * sqrt(pow((goal_pose.x - self.pose.x), 2) + pow((goal_pose.y -
self.pose.y), 2))
            vel_msg.linear.y = 0
            vel_msg.linear.z = 0

            #angular velocity in the z-axis:
```

```
    vel_msg.angular.x = 0
    vel_msg.angular.y = 0
    vel_msg.angular.z = 4 * (atan2(goal_pose.y - self.pose.y, goal_pose.x - self.pose.x) -
self.pose.theta)
```

```
    #Publishing our vel_msg
    self.velocity_publisher.publish(vel_msg)
    self.rate.sleep()
```

```
    print("velocity =", vel_msg.linear.x)
    print("Angular Velocity =", vel_msg.angular.z)
    print("The answer is", 2*2)           # 4 !
    print("x =", self.pose.x)
    print("y =", self.pose.y)
```

```
#Stopping our robot after the movement is over
```

```
    vel_msg.linear.x = 0
    vel_msg.angular.z = 0
    self.velocity_publisher.publish(vel_msg)
```

```
    rospy.spin()
```

```
if __name__ == '__main__':
```

```
    try:
```

```
        #Testing our function
        x = turtlebot()
        x.move2goal()
```

```
    except rospy.ROSInterruptException:
```

```
        pass
```

```
#RESULT:
```

```
# harman@D104-45931:~/Desktop$ python gotogoal.py
# Set your x goal:1.0 Note - input gets string, change to float for the math! 10/13/2022
# Set your y goal:1.0
# Set your tolerance:0.2
```

ERROR FROM PYTHON 2 TO PYTHON 3

ALWAYS, ALWAYS KEEP TRACK OF YOUR ERRORS!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Correction Python2 to Python3

```
def move2goal(self):          # Convert string to float for Python 3
    goal_pose = Pose()
    goal_pose.x = float(input("Set your x goal:"))
    goal_pose.y = float(input("Set your y goal:"))
    distance_tolerance = float(input("Set your tolerance:"))
    vel_msg = Twist()
```

harman@harman-VirtualBox:~/Desktop\$ **python3 gotogoal_1.py**

Set your x goal:1.0

Set your y goal:1.0

Set your tolerance:0.2

Traceback (most recent call last):

File "gotogoal_1.py", line 70, in <module>

x.move2goal()

File "gotogoal_1.py", line 36, in move2goal

while sqrt(pow((goal_pose.x - self.pose.x), 2) + pow((goal_pose.y - self.pose.y), 2)) >= distance_tolerance:

TypeError: '>=' not supported between instances of 'float' and 'str'

(Error running Python 2 program with Python 3)