

**Simulation and Modelling**  
**CSE3102**  
**Lab Submission – 6**

**Laboratory 6 - ROS Turtle Sim – Two Turtles**

Meher Shrishti Nigam  
20BRS1193

**Instructions:**

**Design two turtles namely "Dogobot" and "Catobot" which can be included within the Turtlesim and both the turtles should run concurrently in a random motion.**

You can use the Spawn module for creating another turtle.

Instructions:

Already we have tried creating a TurtleSIM which can move in different shapes and and through teleop key as well

```
$ roscore
```

```
$ rosrun turtlesim turtlesim_node
```

```
$ rosrun turtlesim turtle_teleop_key
```

```
$ rosrun ros_tut_1 circularmovement.py
```

Now, You need to define two turtles in a single turtlesim namely:

```
scp lab21@172.16.10.7:pradeep_ws/src/ros_tutorial1/script/twoturtles.py .
```

**Code:**

```
#!/usr/bin/env python3
```

```
import rospy
```

```
from geometry_msgs.msg import Twist
```

```
from turtlesim.srv import Spawn
```

```
import time
```

```
def create_turtle1():
```

```
    rospy.wait_for_service('spawn')
```

```
    spawner = rospy.ServiceProxy('spawn', Spawn)
```

```
    turtle1_x = 1
```

```
    turtle1_y = 1
```

```
    turtle1_theta = 0
```

```
turtle1_name = "turtle2"
spawner(turtle1_x, turtle1_y, turtle1_theta, turtle1_name)
```

```
def create_turtle2():
    rospy.wait_for_service('spawn')
    spawner = rospy.ServiceProxy('spawn', Spawn)
    turtle2_x = 9
    turtle2_y = 9
    turtle2_theta = 0
    turtle2_name = "turtle3"
    spawner(turtle2_x, turtle2_y, turtle2_theta, turtle2_name)
```

```
def move_turtle1():
    pub = rospy.Publisher('/turtle2/cmd_vel', Twist, queue_size=10)
    rate = rospy.Rate(10)
    twist = Twist()
    twist.linear.x = 2
    twist.angular.z = 2
    while not rospy.is_shutdown():
        pub.publish(twist)
        time.sleep(0.1)
        move_turtle2()
        #rate.sleep()
```

```
def move_turtle2():
    pub = rospy.Publisher('/turtle3/cmd_vel', Twist, queue_size=10)
    rate = rospy.Rate(10)
    twist = Twist()
    twist.linear.x = -2
    twist.angular.z = -2
    while not rospy.is_shutdown():
```

```
pub.publish(twist)
```

```
time.sleep(0.1)
```

```
move_turtle1()
```

```
#rate.sleep()
```

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

```
try:
```

```
    rospy.init_node('turtle_spawner')
```

```
    create_turtle1()
```

```
    create_turtle2()
```

```
    move_turtle1()
```

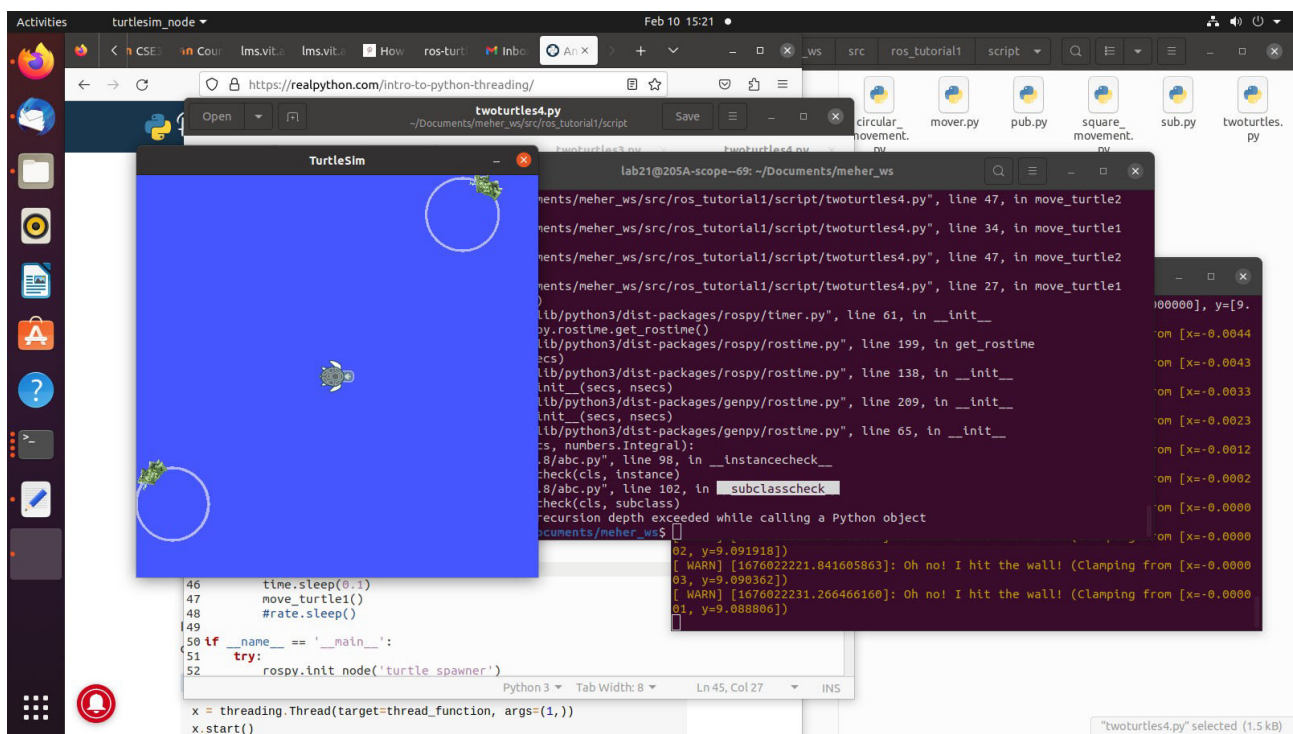
```
    #move_turtle2()
```

```
except rospy.ROSInterruptException:
```

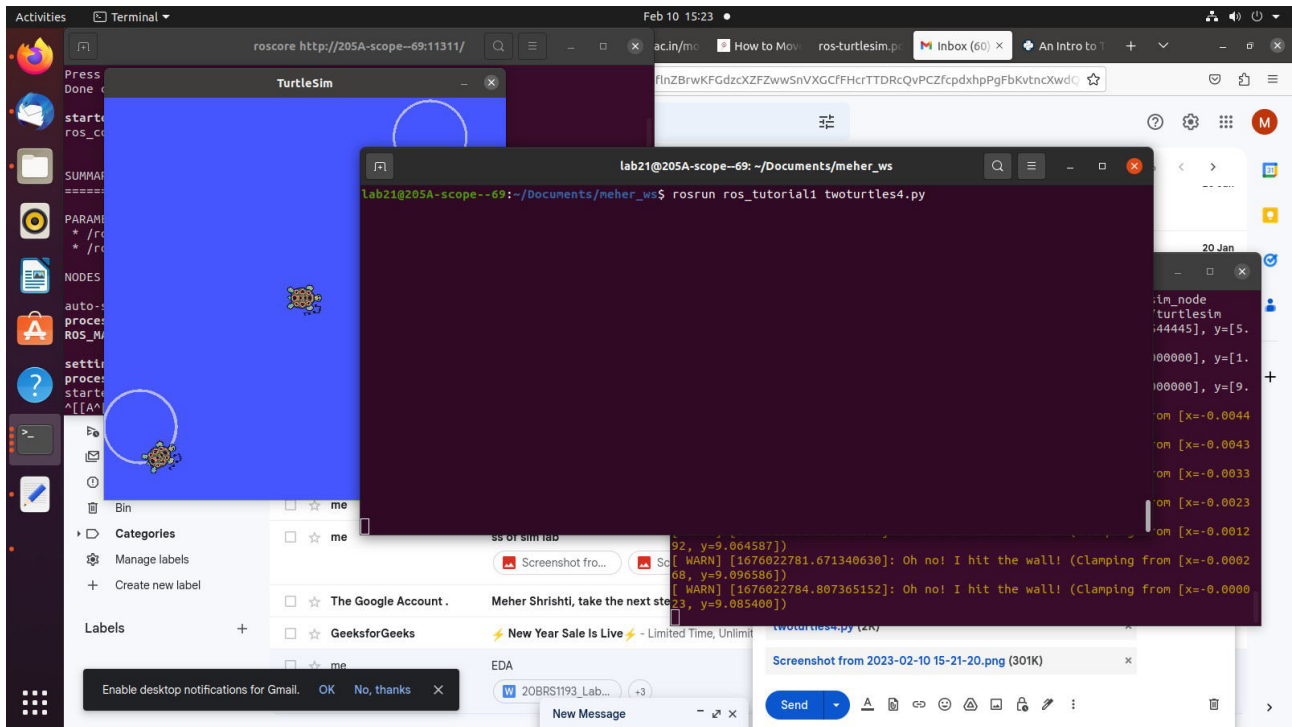
```
    pass
```

**Output:**

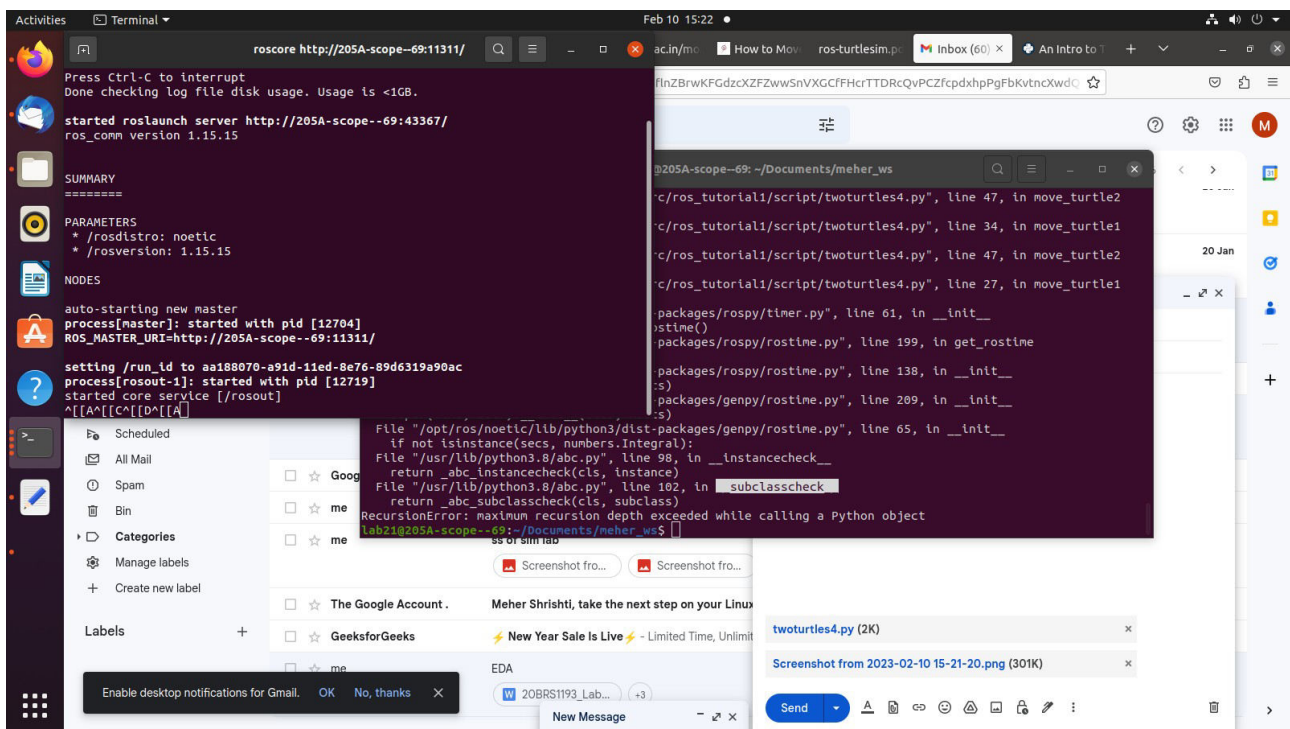
**Turtles spinning simultaneously**



roslaunch ros\_tutorial1 twoturtles4.py



roscore



roslaunch turtlesim turtlesim\_node

