

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

#TODO 1 Modify this header so that the correct information is displayed
#Junhyung Park and Ryan Buckton
#turtlebot_controller.py
#For lab1, this will subscribe to mouse_client and publish to cmd_vel
#Will convert messages of type MouseController to Twist
#Deactivates when mouse wheel is scrolled up
#last modified 12 Feb 2023
#10 Feb 2023 Finished TODO 3
#12 Feb 2023 Fnished rest of TODO

import rospy
#TODO 2 Import the appropriate message types that we will need
from lab1.msg import MouseController
from geometry_msgs.msg import Twist

class Controller:
    """Class that controls subsystems on Turtlebot3"""
    def __init__(self):
        #TODO 3 initialize the appropriate Controller class attributes
        self.cmd = Twist()

        self.cmd.linear.x = 0.0
        self.cmd.linear.y = 0.0
        self.cmd.linear.z = 0.0
        self.cmd.angular.x = 0.0
        self.cmd.angular.y = 0.0
        self.cmd.angular.z = 0.0

        # self.rate = rospy.Rate(10)    # 10 Hz

        self.pub = rospy.Publisher('cmd_vel', Twist, queue_size = 1)

        rospy.Subscriber('mouse_info', MouseController, self.callback_mouseControl)

        self.ctrl_c = False
        rospy.on_shutdown(self.shutdownhook)

    def callback_mouseControl(self, mouseInfo):
        #TODO 4 Scale xPos from -1 to 1 to -.5 to .5
        scaled_xPos = -(mouseInfo.xPos)/2

        #TODO 5 set angular z in Twist message to the scaled value in the appropriate
direction
        self.cmd.angular.z = scaled_xPos

        #TODO 6 Scale yPos from -1 to 1 to -.5 to .5
        scaled_yPos = -(mouseInfo.yPos)/2

        #TODO 7 set linear x in Twist message to the scaled value in the appropriate
direction
        self.cmd.linear.x = scaled_yPos
```

```
#TODO 8 publish the Twist message  
self.pub.publish(self.cmd)
```

```
def shutdownhook(self):  
    print("Controller exiting. Halting robot.")  
    self.ctrl_c = True  
    #TODO 9 force the linear x and angular z commands to 0 before halting  
    self.cmd.linear.x = 0  
    self.cmd.angular.z = 0  
  
    self.pub.publish(self.cmd)
```

```
if __name__ == '__main__':  
    rospy.init_node('controller')  
    c = Controller()  
    rospy.spin()
```