

**teleop\_python.py**

```

1  #!/usr/bin/env python3
2  import rclpy
3  import os
4  import select
5  import sys
6  from rclpy.node import Node
7  from geometry_msgs.msg import Twist
8
9  if os.name == "nt":
10     import msvcrt
11 else:
12     import termios
13     import tty
14
15 BURGER_MAX_LIN_VEL = 0.21
16 BURGER_MIN_LIN_VEL = 0.01
17 BURGER_MAX_ANG_VEL = 2.50
18 BURGER_MIN_ANG_VEL = 0.1
19
20 msg = """
21 Control Your TurtleBot3!
22 -----
23 Moving around:
24         up
25 left    down    right
26
27 u : increase linear velocity and angular velocity
28 j : reset linear velocity and angular velocity
29 m : decrease linear velocity and angular velocity
30
31 CTRL-C to quit
32 """
33
34
35 def get_key(settings):
36     if os.name == "nt":
37         return msvcrt.getch().decode("utf-8")
38     tty.setraw(sys.stdin.fileno())
39     rlist, _, _ = select.select([sys.stdin], [], [], 0.1)
40     if rlist:
41         key = sys.stdin.read(1)
42         if key == "\x1b":
43             key += sys.stdin.read(2)
44     else:
45         key = ""
46
47     termios.tcsetattr(sys.stdin, termios.TCSADRAIN, settings)
48     return key

```

```
49
50
51 def print_vels_set(target_linear_velocity, target_angular_velocity):
52     print(
53         "setting :\tlinear velocity {0}\t angular velocity {1} ".format(
54             target_linear_velocity, target_angular_velocity
55         )
56     )
57
58
59 def print_vels_run(target_linear_velocity, target_angular_velocity):
60     print(
61         "currently :\tlinear velocity {0}\t angular velocity {1} ".format(
62             target_linear_velocity, target_angular_velocity
63         )
64     )
65
66
67 def main(args=None):
68     settings = None
69     if os.name != "nt":
70         settings = termios.tcgetattr(sys.stdin)
71     rclpy.init(args=args)
72     teleop_node = rclpy.create_node("teleop_publisher")
73     teleop_pub = teleop_node.create_publisher(Twist, "cmd_vel", 10)
74
75     current_speed = [0.0, 0.0]
76     max_speed_linear = 0.01
77     max_speed_angular = 0.1
78     twist = Twist()
79
80     print(msg)
81
82     while True:
83         try:
84             key = get_key(settings)
85             if key == "\x03":
86                 print("Exiting...")
87                 break
88
89             if key == "u":
90                 if max_speed_linear < BURGER_MAX_LIN_VEL:
91                     max_speed_linear += 0.01
92                 else:
93                     max_speed_linear = BURGER_MAX_LIN_VEL
94                 if max_speed_angular < BURGER_MAX_ANG_VEL:
95                     max_speed_angular += 0.1
96                 else:
97                     max_speed_angular = BURGER_MAX_ANG_VEL
98                 print_vels_set(max_speed_linear, max_speed_angular)
```

```

99     elif key == "m":
100         if max_speed_linear > BURGER_MIN_LIN_VEL:
101             max_speed_linear -= 0.01
102         else:
103             max_speed_linear = BURGER_MIN_LIN_VEL
104         if max_speed_angular > BURGER_MIN_ANG_VEL:
105             max_speed_angular -= 0.1
106         else:
107             max_speed_angular = BURGER_MIN_ANG_VEL
108         print_vels_set(max_speed_linear, max_speed_angular)
109     elif key == "j":
110         max_speed_linear = 0.01
111         max_speed_angular = 0.1
112         print_vels_set(max_speed_linear, max_speed_angular)
113
114     if key == "\x1b[A" or key == "\x1b[B" or key == "\x1b[C" or key ==
"\x1b[D":
115         if key == "\x1b[A":
116             current_speed[0] = max_speed_linear
117         if key == "\x1b[B":
118             current_speed[0] = -max_speed_linear
119         if key == "\x1b[C":
120             current_speed[1] = -max_speed_angular
121         if key == "\x1b[D":
122             current_speed[1] = max_speed_angular
123         print_vels_run(current_speed[0], current_speed[1])
124     else:
125         current_speed[0] = 0.0
126         current_speed[1] = 0.0
127
128     twist.linear.x = current_speed[0]
129     twist.angular.z = current_speed[1]
130
131     teleop_pub.publish(twist)
132 except KeyboardInterrupt:
133     break
134
135
136 if __name__ == "__main__":
137     main()
138

```