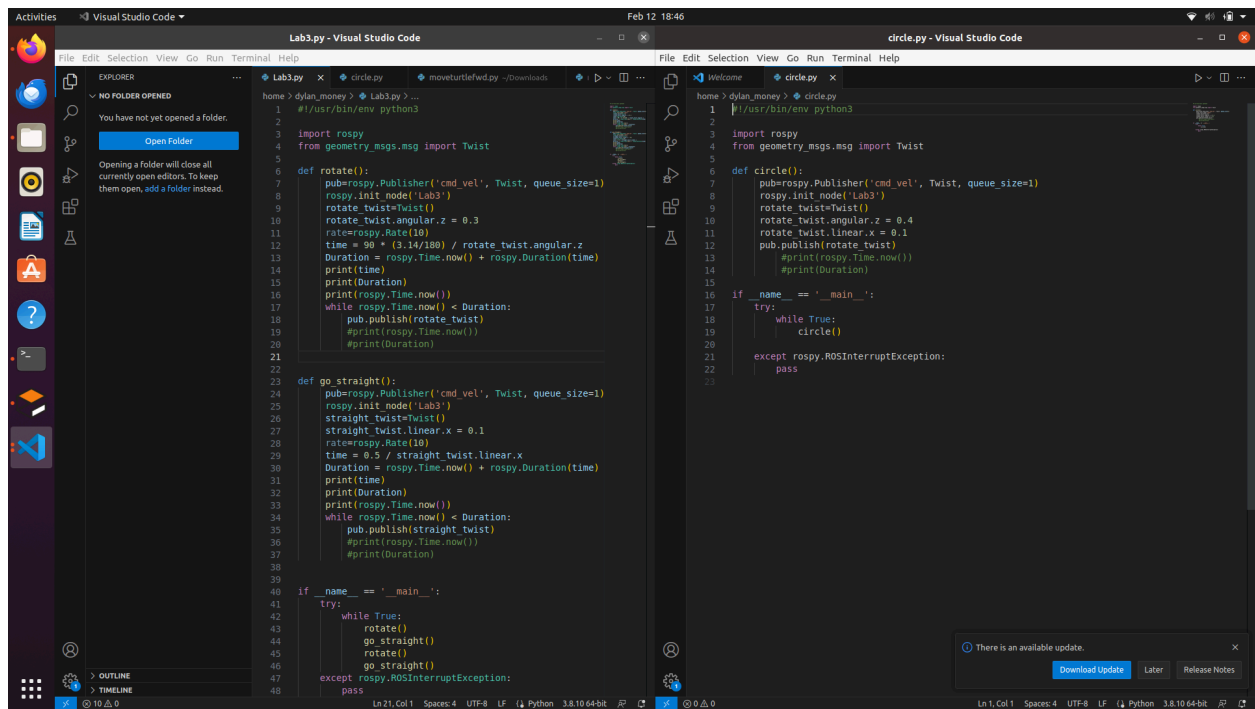


# Lab 3 Report: Dylan Money

Lab 3 was a bit more tricky than our previous labs. Loading up the turtle sim in gazebo I was trying to figure out how to run the python code to make the turtle move and as it turns out it's the exact same as running it from Lab 2 so that was a waste of time. Then I couldn't figure out why I couldn't get the turtle bot to move but I realized I was publishing the linear and angular velocity values to the wrong topic. After those unfortunate mistakes I managed to get my turtle moving. Having gotten my turtle moving I then messed around with the values to see how they affected the turtle. For the square shape I created 2 functions, one that turned the turtle and one that sent it in a straight line. Using the 2 functions in succession you can create a square shape. For the circle shape I made a function called circle that gave the robot both angular and linear velocity. The function would then be called repeatedly and created the circular motion.



```
Lab3.py
1 #!/usr/bin/env python3
2
3 import rospy
4 from geometry_msgs.msg import Twist
5
6 def rotate():
7     pub=rospy.Publisher('cmd_vel', Twist, queue_size=1)
8     rospy.init_node('Lab3')
9     rotate_twist=Twist()
10    rotate_twist.angular.z = 0.3
11    rate=rospy.Rate(10)
12    time = 90 * (3.14/180) / rotate_twist.angular.z
13    Duration = rospy.Time.now() + rospy.Duration(time)
14    print(time)
15    while rospy.Time.now() < Duration:
16        pub.publish(rotate_twist)
17        #print(rospy.Time.now())
18        #print(Duration)
19    print(rospy.Time.now())
20
21 def go_straight():
22     pub=rospy.Publisher('cmd_vel', Twist, queue_size=1)
23     rospy.init_node('Lab3')
24     straight_twist=Twist()
25     straight_twist.linear.x = 0.1
26     rate=rospy.Rate(10)
27     time = 0.5 / straight_twist.linear.x
28     Duration = rospy.Time.now() + rospy.Duration(time)
29     print(time)
30     print(Duration)
31     while rospy.Time.now() < Duration:
32         pub.publish(straight_twist)
33         #print(rospy.Time.now())
34         #print(Duration)
35
36 if __name__ == '__main__':
37     try:
38         while True:
39             rotate()
40             go_straight()
41             rotate()
42             go_straight()
43     except rospy.ROSInterruptException:
44         pass
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
circle.py
1 #!/usr/bin/env python3
2
3 import rospy
4 from geometry_msgs.msg import Twist
5
6 def circle():
7     pub=rospy.Publisher('cmd_vel', Twist, queue_size=1)
8     rospy.init_node('Lab3')
9     rotate_twist=Twist()
10    rotate_twist.angular.z = 0.4
11    rotate_twist.linear.x = 0.1
12    pub.publish(rotate_twist)
13    #print(rospy.Time.now())
14    #print(Duration)
15
16 if __name__ == '__main__':
17     try:
18         while True:
19             circle()
20     except rospy.ROSInterruptException:
21         pass
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```