

## SPRAWOZDANIE

Rafał Litka

Marcin Michalski

Kod programu:

```
#!/usr/bin/env python
import click
import rospy
from geometry_msgs.msg import Twist
rospy.init_node('steering')
pub = rospy.Publisher('/turtle1/cmd_vel', Twist, queue_size=10)
while not rospy.is_shutdown():

    msg = Twist()
    key = click.getchar()
    if key == 'w':
        msg.linear.x = 1
    elif key == 'a':
        msg.angular.z = 1
    elif key == 's':
        msg.linear.x = -1
    elif key == 'd':
        msg.angular.z = -1
    pub.publish(msg)
```

Kod rozpoczyna się shebangiem, który sprawia, że program uruchamiany jest przez interpreter pythona.

Zaimportowane biblioteki to:

Click (biblioteka, której używamy to sprawdzenia wciśniętego przycisku)

Rospy (API ROS dla pythona)

Twist (z biblioteki `geometry_msgs.msg` pobieramy wiadomość Twist, która odpowiada za prędkość liniową i kątową żółwia)

Następnie inicjujemy nowy node o nazwie `steering` i tworzymy publishera o nazwie `pub`, który publikuje wiadomości na topic „/turtle1/cmd\_vel” zawierające żadaną prędkość żółwia.

Potem wchodzimy do pętli działającej tak długo jak aktywny jest nasz node, w pętli tworzymy pustą wiadomość Twist, którą wypełniamy po wciśnięciu przycisku(w/a/s/d) przez użytkownika żadaną prędkością i publikujemy wiadomość.