

Nama : Aryaka Leorgi
NRP : 5025231018

Laporan Penugasan 3

1. Sama seperti penugasan 2 kita karena sudah membuat package kita tinggal hanya perlu copy paste package sebelumnya ke folder baru
2. Kita sekarang hapus kode publisher dan bikin file baru yang bernama turtle sim
3. Setelah itu kita hanya tinggal perlu memasukkan kode turtlesim ke kode file tersebut
4. Untuk kodenya

```
def reset_turtle(self):  
    self.teleport_absolute(7.5, 5.0, 0.0)  
    self.clear_turtlesim()
```

Disini untuk mengreset turtle ke tempat yang sudah ditentukan yaitu 7,5 dan 5,0

```
def clear_turtlesim(self):  
    clear_service = self.create_client(Empty, '/clear')  
    while not clear_service.wait_for_service(timeout_sec=1.0):  
        self.get_logger().info('Waiting for the clear service...')  
    empty_request = Empty.Request()  
    clear_future = clear_service.call_async(empty_request)  
    rclpy.spin_until_future_complete(self, clear_future)  
    self.get_logger().info('Clearing complete.')
```

Bagian ini untuk menghilangkan garis garis yang sudah dibuat turtle sebelumnya.

```
def teleport_absolute(self, x, y, theta):  
    teleport_service = self.create_client(TeleportAbsolute,  
    '/turtle1/teleport_absolute')  
    while not teleport_service.wait_for_service(timeout_sec=1.0):  
        self.get_logger().info('Waiting for the teleport_absolute  
service...')  
  
    request = TeleportAbsolute.Request()  
    request.x = x  
    request.y = y  
    request.theta = theta  
  
    future = teleport_service.call_async(request)  
    rclpy.spin_until_future_complete(self, future)  
    self.get_logger().info('Teleportation complete.')
```

Ini untuk mentelportasi yang digunakan pada command resetnya

```
def timer_callback(self):
    msg = Twist()

    if self.count == 0:
        msg.angular.z = math.radians(180)
        self.get_logger().info('Rotating...')

    elif self.count % 2 == 1 and self.count < 7 and self.count > 0:
        msg.linear.x = 4.0
        self.get_logger().info('Moving Forward...')

    elif self.count % 2 == 0 and self.count < 6 and self.count > 0:
        msg.angular.z = math.radians(-120)
        self.get_logger().info('Rotating...')

    elif self.count == 7:
        msg.angular.z = math.radians(-30)
        self.get_logger().info('Rotating...')

    elif self.count > 7 and self.count % 2 == 0:
        msg.linear.x = 6.2
        msg.angular.z = math.radians(-178)
        self.get_logger().info('Moving in a curved path for Half
Circle...')

    elif self.count == 9:
        msg.angular.z = math.radians(60)
        self.get_logger().info('Rotating...')

    elif self.count == 11:
        msg.angular.z = math.radians(60)
        self.get_logger().info('Rotating...')

    self.publisher.publish(msg)
    self.count += 1

    if self.count == 13:
        self.get_logger().info('Shape completed.')
        self.timer.cancel()
```

Disini adalah command command yang digunakan agar turtle nya membuat shape yang sudah ditentukan pertama ini fungsinya dengan menambahkan 1 turn di setiap command yang dilakukan dengan kita juga menentukan turn berapa ingin kita akhiri command tersebut

Selanjutnya moving forward yaitu turtle nya gerak lurus seberapa banyak jauhnya
Terus ada rotating yaitu untuk memutar turtlenya untuk menentukan arah yang ingin dituju
Terus terakhir moving in curved itu untuk gerakan yang membuat setengah lingkaran