

## تمرین 1:

در این بخش کد را به صورت زیر تغییر می دهیم. یعنی ابتدا 6 Subscription را به ازای هر لاک پشت Spawn شده ساخته و position هر کدام را set می کنیم:

```
def __init__(self):
    super().__init__("controller")
    self.chaser_x_ = 0
    self.chaser_y_ = 0
    self.chaser_angle_ = 0
    self.spawned_x_ = []
    self.spawned_y_ = []

    self.i = 0

    self.turtle_num = 6

    for i in range(self.turtle_num):
        self.spawned_position_subscriber_ = self.create_subscription(Pose, "spawned_"+str(i)+"/pose", self.set_spawned_position, 10)

    self.chaser_position_subscriber_ = self.create_subscription(Pose, "turtle1/pose", self.set_chaser_position, 10)
    self.velocity_publisher_ = self.create_publisher(Twist, "turtle1/cmd_vel", 10)

    self.kill_subscriber_ = self.create_subscription(Empty, "turtle1_killed", self.next, 1)
```

سپس هنگام استفاده در Chaser\_to\_turtle موقعیت مناسب هر کدام را به تابع می دهیم: (این موقعیت توسط یک Subscription آپدیت می شود).

```
distance = math.sqrt(((self.spawned_x_[self.i]-self.chaser_x_)**2) + ((self.spawned_y_[self.i]-self.chaser_y_)**2))
desired_angle = math.atan2(self.spawned_y_[self.i]-self.chaser_y_, self.spawned_x_[self.i]-self.chaser_x_)
```

## تمرین 2:

در این بخش فقط کافیست یک Publisher را به ازای هر لاک پشت کشته شده Publish کرده و کد را به صورت زیر تغییر می دهیم. همچنین یک topic با نام death\_note ساخته و ترتیب کشته شدن لاک پشت ها را نمایش می دهیم.

```
def __init__(self):
    super().__init__("spawner")

    self.turtle_num = 6
    self.name_prefix = "spawned_"
    self.turtle_map = dict()
    self.spawn_turtle()

    self.order = []

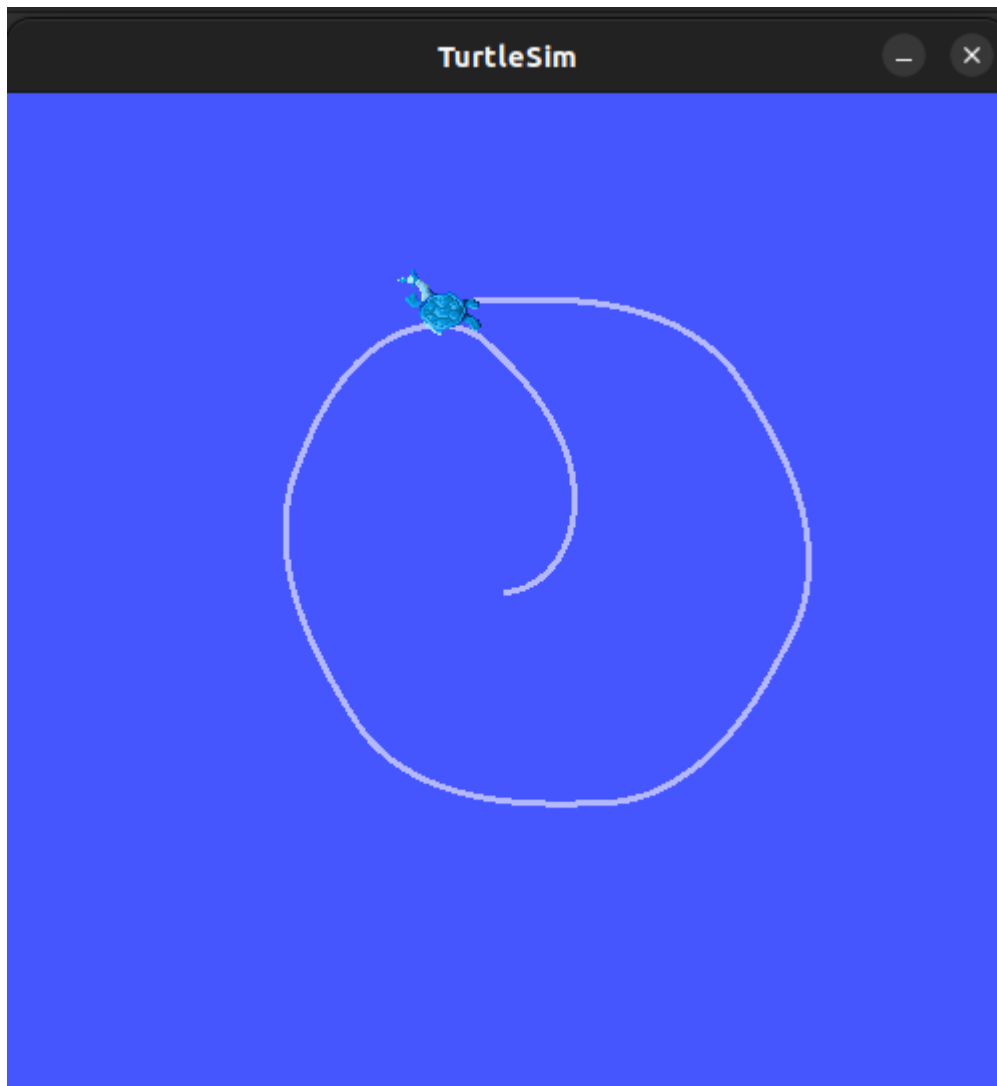
    self.chaser_position_subscriber_ = self.create_subscription(Pose, "turtle1/pose", self.callback_chased, 10)
    self.pose_publisher_ = self.create_publisher(Empty, "turtle1_killed", 1)
    self.death_publisher_ = self.create_publisher(String, "death_note", 1)

def callback_chased(self, chaser_position):
    for name in list(self.turtle_map.keys()):
        pos = self.turtle_map[name]
        if abs(chaser_position.x - pos[0]) < 0.5 and abs(chaser_position.y - pos[1]) < 0.5:
            self.kill_turtle(name)
            st = Empty()
            self.pose_publisher_.publish(st)
            message = String()
            self.order.append(name)
            message.data = name + " Has Been Killed! ("
            for i in range(len(self.order)):
                message.data += str(i)
                if i != len(self.order) - 1:
                    message.data += "->"
            message.data += ")"
            self.death_publisher_.publish(message)

            del self.turtle_map[name]

            time.sleep(0.1)
```

مشاهده نتایج:



```
data: spawned_0 Has Been Killed! (0)
---
data: spawned_1 Has Been Killed! (0->1)
---
data: spawned_2 Has Been Killed! (0->1->2)
---
data: spawned_3 Has Been Killed! (0->1->2->3)
---
data: spawned_4 Has Been Killed! (0->1->2->3->4)
---
data: spawned_5 Has Been Killed! (0->1->2->3->4->5)
```