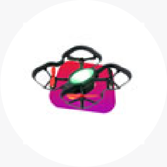
1.4 Flight Movements: Throttle and Yaw

Now that we know some basic movements, lets go a little deeper by making our drone go higher!

This lesson works with:



CoDrone EDU

CoDrone EDU (JROTC ed.)

Python

beginner | CDE | CoDrone EDU | flight movements | Python

Grade level:
6 - 12+

Approx. time required:
30 - 45 mins

Step 1

Goals and Steps to Success

By the end of this lesson, you should be able to:

- Describe how roll, pitch, yaw, and throttle change the drone's movement in the air.
- Use the roll, pitch, yaw, and throttle commands to make the drone navigate a path of your choosing.
- Incorporate your knowledge of creating a complete program from Island 1 to build a full flight path for the drone including takeoff and landing.

In this lesson, you will complete the following steps. Make sure that you demonstrate your completed code to your instructor after each step:

1. Getting Started
2. More Movements: Throttle and Yaw
    1. Yaw
    2. Throttle
3. Doing the Drone Tango: Putting Movements Together
4. Challenge: No One Puts Drone In a Corner

Step 2

Getting Started

Before we learn how drones move, we need to have a program created for testing movement. To do this, follow the steps below:

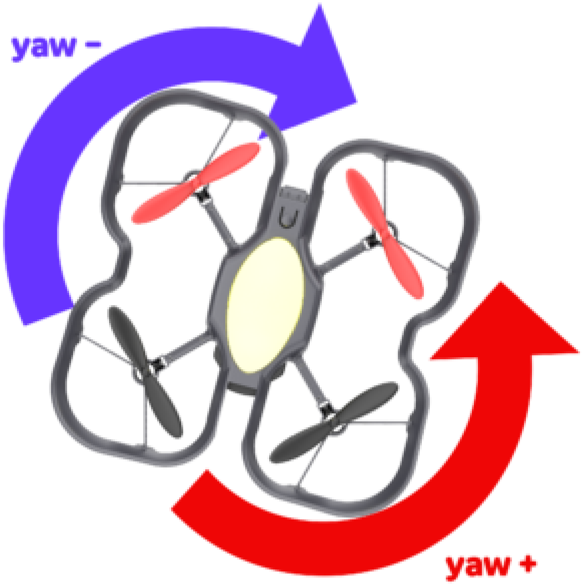
1. Open Pycharm
2. Right click on the 2_1 folder you created in 2.1 and click New -> Python File
3. Name the file *yaw*
4. Enter the commands you learned from the previous lesson to pair with the drone, take off, and land. (Hint: You can go back to the previous lesson's folder to look at what you wrote if you can't remember)

You are now ready to start testing some more movements that your drone can perform!

Step 3

Yaw

This is the CoDrone EDU's left and right rotation. Yaw is a little different than pitch and roll. While pitch and roll make the drone move left and right or up and down, Yaw makes the drone move around its axis. If you remember at the beginning of this lesson, we learned that drones movement can be tied to the X, Y, and Z axes. So far we have looked at X and Y movement with roll and pitch. Now, we are looking at the Z axis with Yaw. With Yaw, the drone spins on the Z axis to face a direction.



Think of Yaw as the ability to point the drone in a direction before making it go that way. If someone has ever told you to look at something, you probably rotated your body or head to face that direction. That would be an example of yaw. Yaw makes the drone spin in place to face a direction. Positive yaw will make the CoDrone EDU spin to the left, and negative yaw will make the CoDrone EDU spin to the right.

```
drone.set_yaw(power) # power represents power out of
100%

                     # power can be between -100 and 100
```

To test out this code, make a copy of your pitch.py file and call it yaw.py. Now change set_pitch to set_yaw. You can leave all other values the same. So your code should look like:

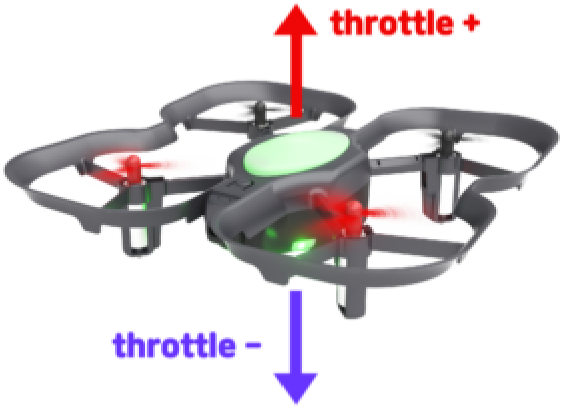
```
drone.set_yaw(25) # setting the yaw value to 25
drone.move(1)      # drone will yaw for 1 second
```

What did you observe? The drone should have spun slowly *left* for 1 second. Try making the drone spin right and slightly faster for a half second. What does your code look like?

Step 4

Throttle

As we saw with Yaw, the drone can pivot around the Z axis. However if we want the drone to move up and down the Z axis, we use throttle. Throttle controls the CoDrone EDU's vertical, or up and down, movement. Positive throttle will make the CoDrone EDU fly higher, and negative throttle will make the CoDrone EDU fly lower.



We have actually used throttle before, possibly without realizing it. Where do you think we have used throttle before? If you guessed on takeoff and land, you would be right! The takeoff and land functions use throttle to make the drone rise to a certain height by increasing the throttle and lowering to the ground by decreasing the throttle.

```
drone.set_throttle(power) # power represents power out of
100%

                          # power can be between -100 and 100
```

To test out this code, make a copy of your yaw.py file and call it throttle.py. Now change set_pitch to set_yaw. You can leave all other values the same. So your code should look like:

```
drone.set_throttle(25) # setting the throttle value to 25
drone.move(1)      # drone will change throttle for 1 second
```

What did you observe? What settings do you think are used for the takeoff and land commands related to throttle?

Be careful with yaw and roll! Yaw will make your CoDrone EDU turn left or right, while roll will move your CoDrone EDU left or right.

Step 5

Putting Movements Together

Now that we know how to use the different movement commands separately, lets see what happens when we use them together! Complete the steps below for our first combined program:

1. Make a copy of yaw.py or any of your exercise files to this point.
2. Enter a command to make the drone move forward for 1 second at 20% power
3. Enter a command to make the drone move to the left for 2 seconds at 20% power

Once you are done, run your code and see what happens! If you are having issues with your code running, here is some example code below:

```
from codrone_edu.drone import *
drone = Drone()
drone.pair()
drone.takeoff()
drone.set_pitch(20)
drone.move(1)
drone.set_roll(-20)
drone.move(2)
drone.land()
drone.close()
```

What did you observe when you ran the code? You may have noticed that the drone moved forward correctly but then it moved forward and left at the same time. This is because we never set the pitch back to 0. If you remember from earlier we learned that these set_movement commands were changing the value of a variable. If we don’t set the value back to 0, then it stays whatever value we set it as last. This can actually be valuable if we want to move in two directions at once like forward and to the right) but was not what we expected.

Try to think how you would reset the value of the pitch so that our drone only moves to the left. To do this, we just set the pitch to 0 after the drone has moved forward.

```
drone.set_pitch(0)
```

Add this new line of code to your program and test it again. This time the drone should move forward and then left only. Below is what the code should look like with the reset value(minus the importing, takeoff, etc..):

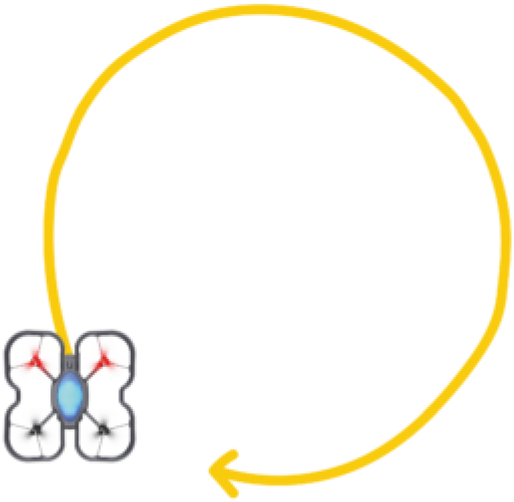
```
drone.set_pitch(20) # Moves the drone forward at 20% power
drone.move(1) # Moves for 1 second
drone.set_pitch(0) # Resets the pitch back to 0
drone.set_roll(-20) # Moves the drone to the left at 20% power
drone.move(2) # Moves for 2 seconds
```

Step 6

Flight Challenges

Below are a series of challenges to test your flight movement skills! These tasks may require combining movements together simultaneously.

1. Fly in a circle



2. Fly in the shape of a sine wave

Step 7

Lesson complete

In this lesson, you learned about how to fly the drone up and down and turn left and right. Next, you will learn about variables, and how to use them to write better and more efficient code.