Flight Movements

In this lesson, you'll learn how to program your CoDrone to move up, down, forwards, backwards, and side to side. You'll also program it to fly in a shape!

Roll, pitch, yaw, throttle

Air vehicles move a little differently than land vehicles. While land vehicles can move forward, backward, and sometimes side-to-side, air vehicles can do all of that and move up and down. To keep track of how a CoDrone moves in a three dimensional space, you can use the terms below:

Roll controls the CoDrone's horizontal, or side to side, movement. Positive roll will make the CoDrone move to the right, and negative roll will make the drone move to the left.

```
drone.set_roll(power) # power represents power out of 100%
                      # power can be between -100 and 100
```

Pitch is the CoDrone's forward and backward tilt. Positive pitch will make the CoDrone tilt and move forward, and negative pitch will make the CoDrone tilt and move backwards.

```
drone.set_pitch(power) # power represents power out of 100%
                       # power can be between -100 and 100
```

Yaw is the CoDrone's left and right rotation. Positive yaw will make the CoDrone turn to the right, and negative yaw will make the CoDrone turn to the left.

```
drone.set_yaw(power) # power represents power out of 100%
                     # power can be between -100 and 100
```

Throttle controls the CoDrone's vertical, or up and down, movement. Positive throttle will make the CoDrone fly higher, and negative throttle will make the CoDrone fly lower.

```
drone.set_throttle(power) # power represents power out of 100%
                          # power can be between -100 and 100
```

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

Blind drone

One of the best ways to learn new words is to see them in action, so instead of you just memorizing all of the flight movements, you're going to do an obstacle course!Have a friend sit down on a chair and then blindfold them (or just have them keep their eyes closed). Next, guide them through an obstacle course to have them sit down on another chair at the end using only flight movement words. Make sure that anything dangerous is out of their way first!

- Positive throttle: stand up
- Negative throttle: sit down
- Positive pitch: move forward
- Negative pitch: move backward
- Positive yaw: turn right
- Negative yaw: turn left
- Positive roll: move right
- Negative roll: turn left

For example, if you wanted to tell your friend to move forward, you could say, "Positive pitch 2 steps." That would let your friend know that you want them to move forward by 2 steps. Once your friend successfully completes the obstacle course, switch roles so they can give you directions!

Using flight commands

Your first challenge: have your CoDrone take off, go forward, then turn left, and land. First, you need to import the CoDrone library and pair your drone:

```
import CoDrone

drone = CoDrone.CoDrone()
drone.pair(drone.Nearest)
# drone.pair() if paired to CoDrone before
drone.takeoff()
```

Next, you can add the new functions that you just learned!

```
import CoDrone

drone = CoDrone.CoDrone()
drone.pair(drone.Nearest)
# drone.pair() if paired to CoDrone before

drone.takeoff()       # takeoff for 2 seconds

drone.set_pitch(30)    # Set positive pitch to 30% power
drone.move(2)         # forward for 2 seconds

drone.set_yaw(-50)    # Set negative yaw to 30% power
drone.move(2)         # turn left for 2 seconds

drone.land()          # lands the CoDrone
drone.close()         # disconnects CoDrone
```

Wondering why you need two lines of code for one movement? `drone.set()` will prepare the CoDrone to move in a certain direction and speed, while `drone.move()` will actually move the CoDrone in the air.

In the last step, each flight direction was used separately. You probably saw your CoDrone go straight up, fly straight forward, turn to the left, and then land. However, you can also have your CoDrone fly in multiple directions at once. We know this sounds crazy but it just means you will be able to do stuff like fly diagonally or spin while flying in another direction.

Run this code and see what happens!

```
import CoDrone

drone = CoDrone.CoDrone()
drone.pair(drone.Nearest)
# drone.pair() if paired to CoDrone before

drone.takeoff()       # takeoff for 2 seconds

drone.set_pitch(30)    # Set positive pitch to 30% power
drone.set_roll(-30)    # Set negative roll to 30% power
drone.move(2)         # forward and right for 2 seconds

drone.land()          # lands the CoDrone
drone.close()         # disconnects CoDrone
```

Flight commands can be confusing. Try `drone.go()` instead! This is what the function looks like:

```
drone.go(direction, duration, power)
```

This is what each part means:

- Direction: where the drone will fly. You can use FORWARD, BACKWARD, LEFT, RIGHT, UP, and DOWN, but make sure you type in all caps like you see here.
- Duration: how long the movement will happen in seconds. You can leave it as blank, zero, or any positive number. If duration is zero, it runs indefinitely. If duration has no number, it automatically sets to 1. If you put any positive number, duration is equal to that value.
- Power: the speed of the drone. It can by any number between 0-100, with 0 being no power and 100 being full power. If power has no number, it will default to 50.

Here's some examples:

```
drone.go(UP, 3, 75) # go up for 3 seconds at 75% power
drone.go(FORWARD)   # go forward for 1 second at 50% power
drone.go(LEFT, 6)   # go left for 6 seconds at 50% power
```

If you want to use drone.go() in your programs, you'll need to include this line at the top, right underneath `import CoDrone`:

```
from CoDrone import Direction
```

After that, you're good to go! Try using `drone.go()` in the program you wrote for Step 3 in place of `drone.set()` and `drone.move()`. What happened?

Step 7 - Programming assignment

Program your CoDrone to fly in a square using the CoDrone directions you recently learned. You can use `drone.set()` and `drone.move()` or `drone.go()`.

Using drone.set() and drone.move():

```
import CoDrone
drone = CoDrone.CoDrone()
drone.pair()              # pairs to previous drone

drone.takeoff()           # takeoff for 2 seconds
drone.hover(3)            # hover for 3 seconds

drone.set_pitch(30)       # set positive pitch at 30% power
drone.move(2)             # forward for 2 seconds

drone.set_roll(-30)       # set negative roll at 30% power
drone.move(2)             # left for 2 seconds

drone.set_pitch(-30)      # set negative pitch at 30% power
drone.move(2)             # backward for 2 seconds

drone.set_roll(30)        # set positive roll at 30% power
drone.move(2)             # right for 2 seconds
```

```
19
20 drone.land()              # lands the CoDrone
21 drone.close()             # disconnects CoDrone
```

Using drone.go():

```
1  import CoDrone
2  from CoDrone import Direction
3
4  drone = CoDrone.CoDrone()         # easier to type drone
5  drone.pair(drone.Nearest)         # pairs to nearest drone
6
7  drone.takeoff()                   # takeoff for 2 seconds
8  drone.hover(3)                    # hover for 3 seconds
9
10 drone.go(FORWARD, 2, 30)          # moves the drone forward for 2 seconds at
   30% power
11 drone.go(LEFT, 2, -30)            # moves the drone left for 2 seconds at 30%
   power
12 drone.go(BACKWARD, 2, -30)        # moves the drone backward for 2 seconds at
   30% power
13 drone.go(RIGHT, 2, 30)            # moves the drone right for 2 seconds at
   30% power
14
15 drone.land()                      # lands the CoDrone
16 drone.close()                     # disconnects CoDrone
```

Check yourself

Let's see what you know!

What will your CoDrone do when it sees this line? drone.go(BACKWARD, 2.5)