Intro to Drone Programming

CDA 4625 - U01 Introduction to Mobile Robotics

Paulo Padrao, Research Assistant

School of Computing and *Information Sciences*

ppadraol@fiu.edu

Some applications

Photography

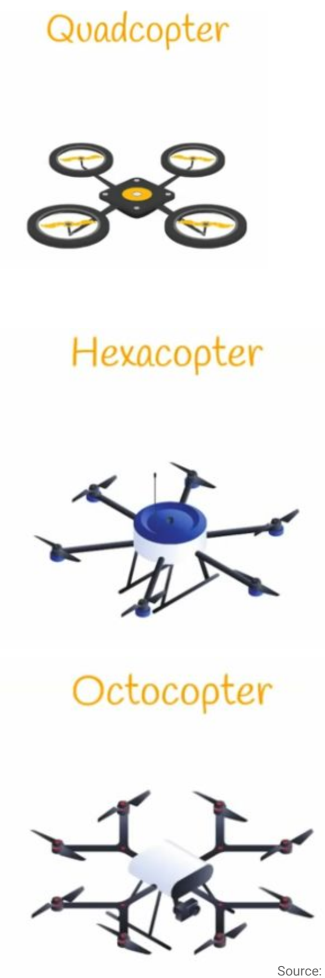
Film recording

Surveillance

Delivery

Source: Murtaza's Workshop - Robotics and AI

Types of Drones

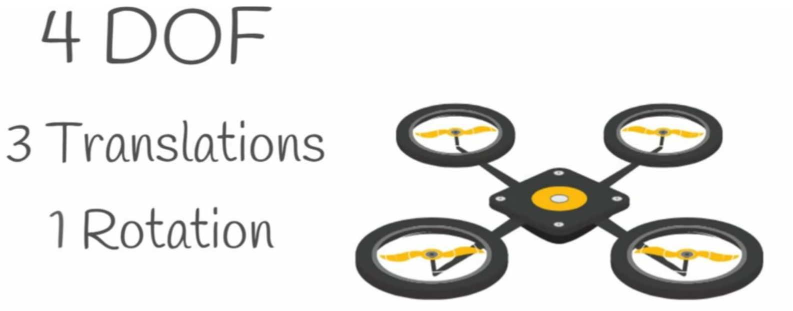


Source: Murtaza's Workshop - Robotics and AI

Main Components

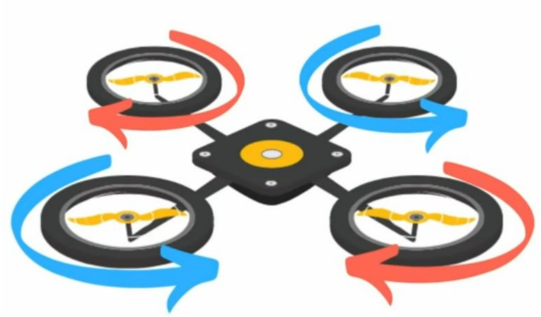
- Frame
- Motors
- Propellers (Blades)
- Electric Motor Controller
- Battery
- Power Distribution Board
- Flight Controller
- Camera
- Receiver
- Sensors (IMU, pressure sensor, etc)

How it flies



Source: Murtaza's Workshop - Robotics and AI

How it flies

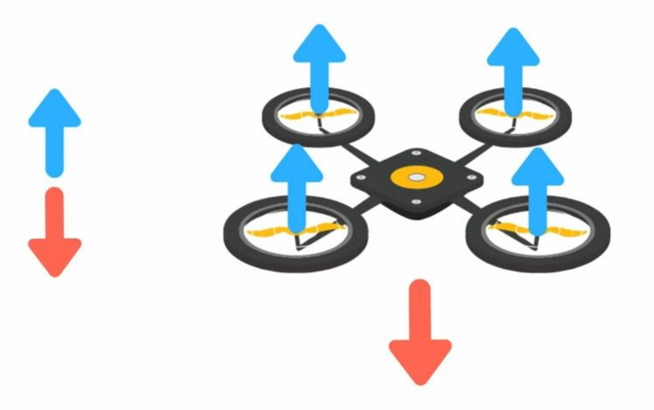


- 2 propellers rotate CW and 2 propellers rotate CCW
- Generate 0 angular momentum
- Keep it stationary instead of rotating in one direction

Source: Murtaza's Workshop - Robotics and AI

How it flies
(Hovering)

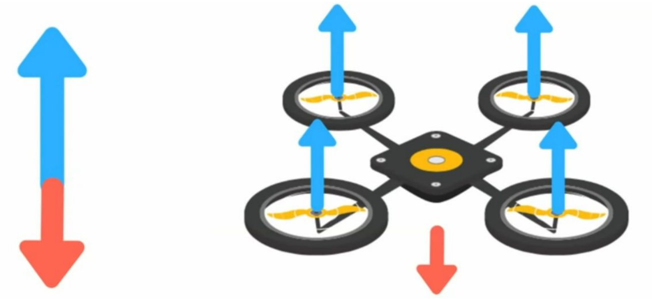
Hovering: Lift = Weight



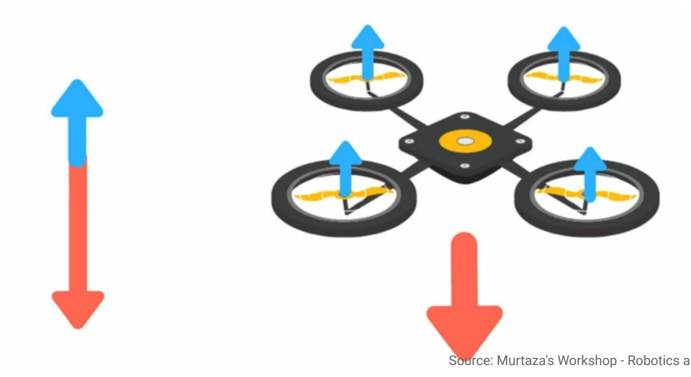
Source: Murtaza's Workshop - Robotics and AI

How it flies
(Translational movements)

Moving up: Lift > Weight



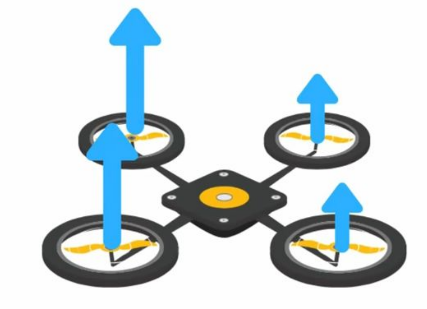
Moving down: Lift < Weight



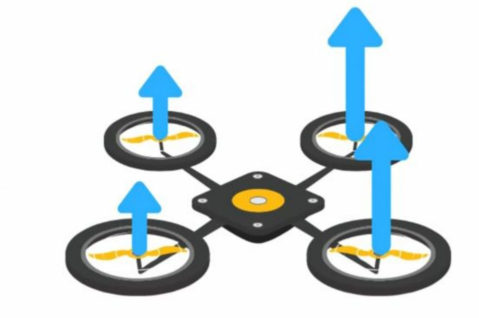
Source: Murtaza's Workshop - Robotics and AI

How it flies
(Translational movements)

Moving Right



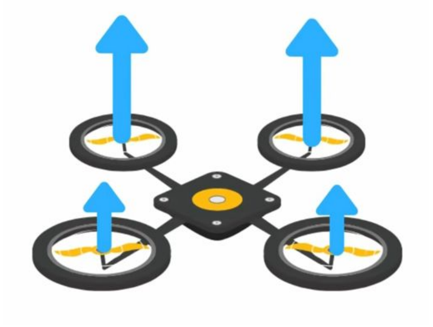
Moving Left



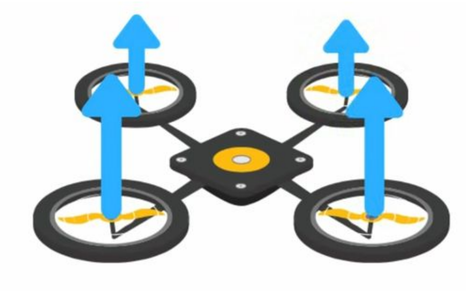
Source: Murtaza's Workshop - Robotics and AI

How it flies
(Translational movements)

Moving Forward



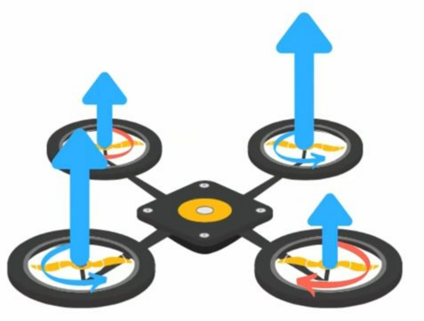
Moving Backwards



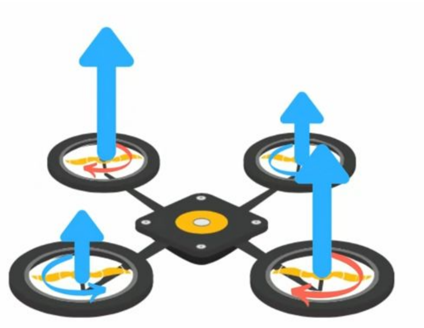
Source: Murtaza's Workshop - Robotics and AI

How it flies
(Rotational movements)

Moving CW

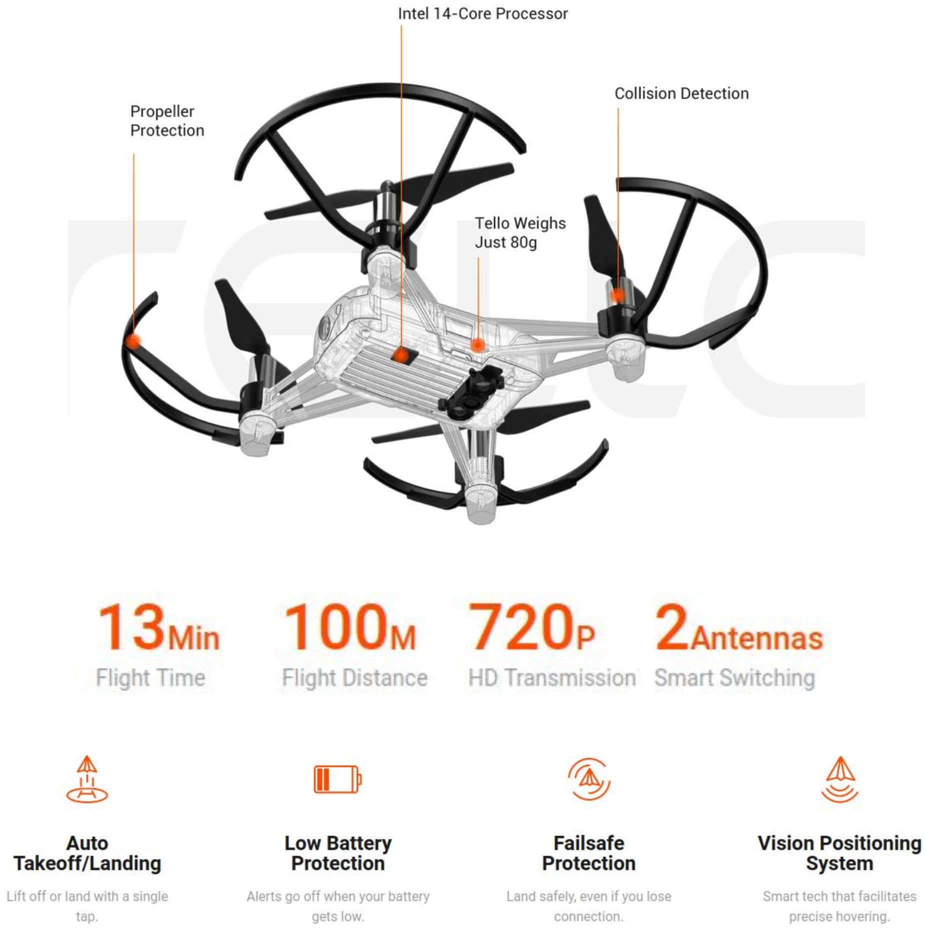


Moving CCW



Source: Murtaza's Workshop - Robotics and AI

Tello Drone



13Min Flight Time

100M Flight Distance

720P HD Transmission

2Antennas Smart Switching

Auto Takeoff/Landing

Lift off or land with a single tap.

Low Battery Protection

Alerts go off when your battery gets low.

Failsafe Protection

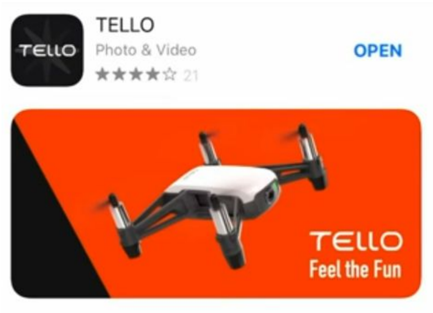
Land safely, even if you lose connection.

Vision Positioning System

Smart tech that facilitates precise hovering.

App Setup and Test Run

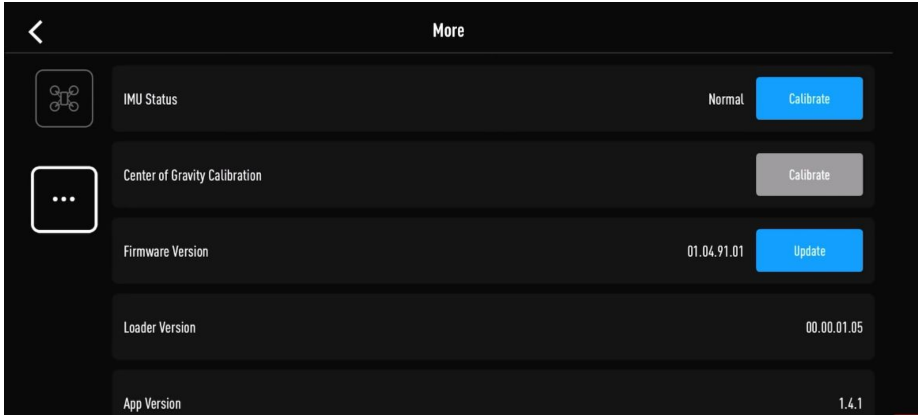
Step 0: Download the Tello app.



Step 1: Turn on the drone. When the light starts blinking with a single color, the drone is ready.

App Setup and Test Run

Step 2: In Tello App, go to Settings > More > Firmware Update.



IMU Status Normal

Center of Gravity Calibration

Firmware Version 01.04.91.01

Loader Version 00.00.01.05

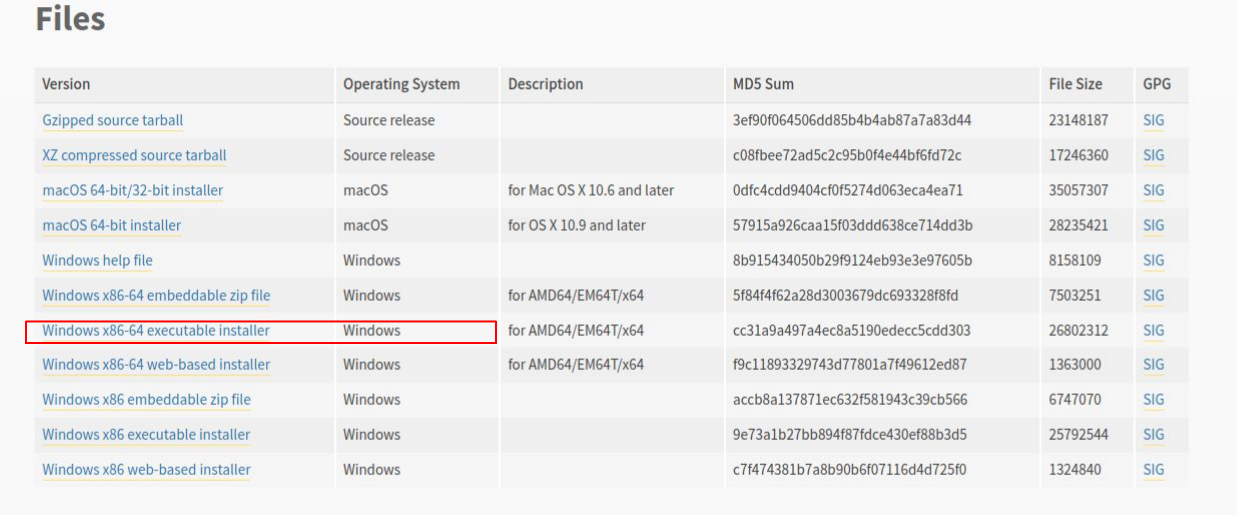
App Version 1.4.1

Note: We need to disconnect the drone first and then connect our phone to WiFi to download the firmware updates

Python Installation and Setup

Download Python 3.7.6.

https://www.python.org/downloads/release/python-376/

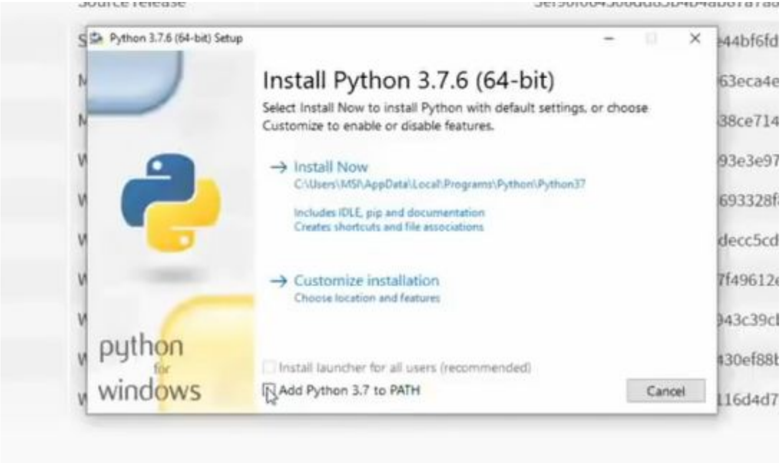


Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		3ef90f064506dd85b4b4ab87a7a83d44	23148187	SIG
XZ compressed source tarball	Source release		c08fbee72ad5c2c95b0f4e44bf6fd72c	17246360	SIG
macOS 64-bit/32-bit installer	macOS	for Mac OS X 10.6 and later	0dfc4cdd9404cf0f5274d063eca4ea71	35057307	SIG
macOS 64-bit installer	macOS	for OS X 10.9 and later	57915a926caa15f03ddd638ce714dd3b	28235421	SIG
Windows help file	Windows		8b915434050b29f9124eb93e3e97605b	8158109	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	5f84f4f62a28d3003679dc693328f8fd	7503251	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	cc31a9a497a4ec8a5190edecc5cdd303	26802312	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	f9c11893329743d77801a7f49612ed87	1363000	SIG
Windows x86 embeddable zip file	Windows		accb8a137871ec632f581943c39cb566	6747070	SIG
Windows x86 executable installer	Windows		9e73a1b27bb894f87fdce430ef88b3d5	25792544	SIG
Windows x86 web-based installer	Windows		c7f474381b7a8b90b6f07116d4d725f0	1324840	SIG

Python Installation and Setup

Note 1: Make sure to add Python 3.7 to the Path



Python 3.7.6 (64-bit) Setup

Install Python 3.7.6 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

→ Install Now
C:\Users\MSI\AppData\Local\Programs\Python\Python37

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

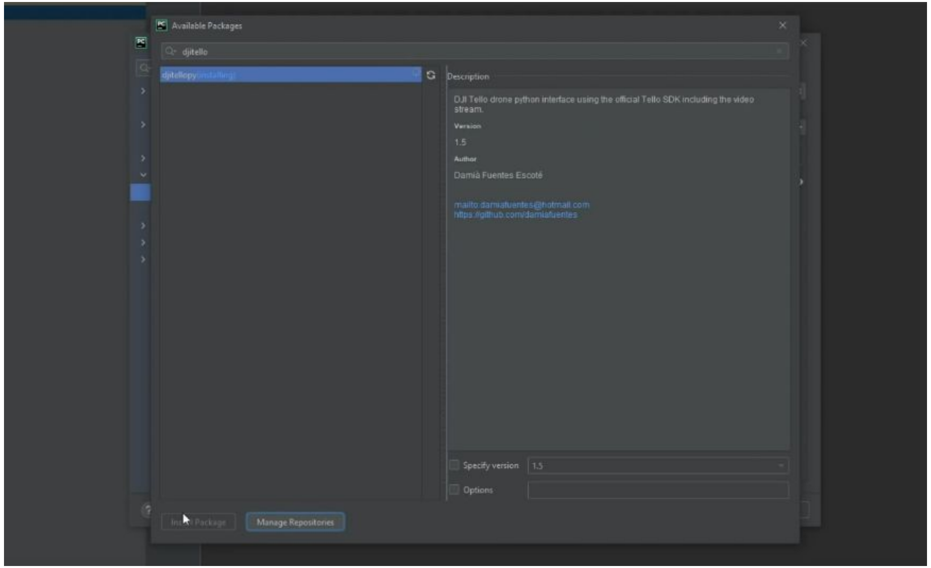
☐ Install launcher for all users (recommended)
☐ Add Python 3.7 to PATH

Note 2: When creating a new project in PyCharm, make sure to check if the interpreter is set to Python 3.7

Python Installation and Setup

Note 3: To install Tello Library, go to *File > Settings > Project > Project Interpreter > Add > Type djitello > Install*

Note 4: We also need to install opencv and pygame libraries



djitello

djitellopy (installing)

Description

DJI Tello drone python interface using the official Tello SDK including the video stream.

Version

1.5

Author

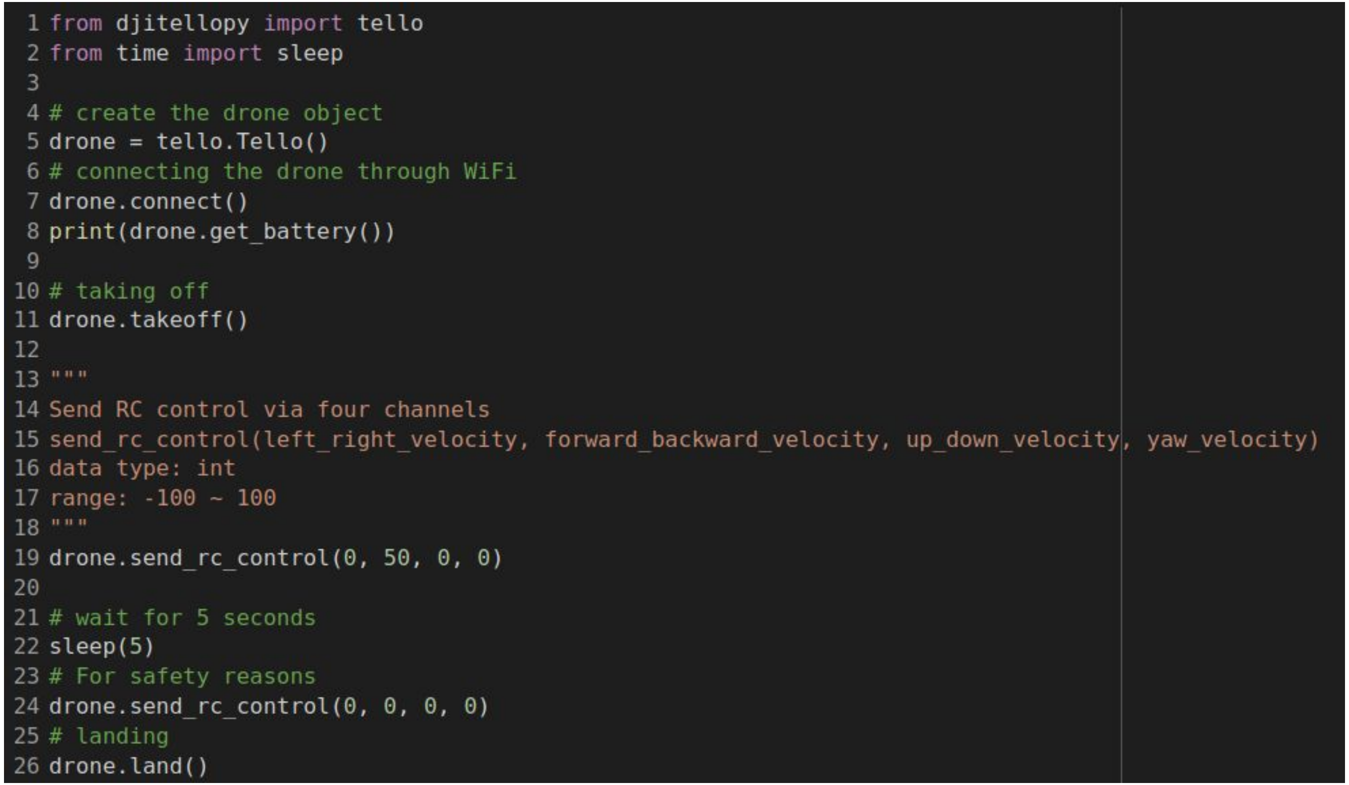
Damià Fuentes Escoté

mailto:damiafuentes@hotmail.com
https://github.com/damiafuentes

□ Specify version 1.5

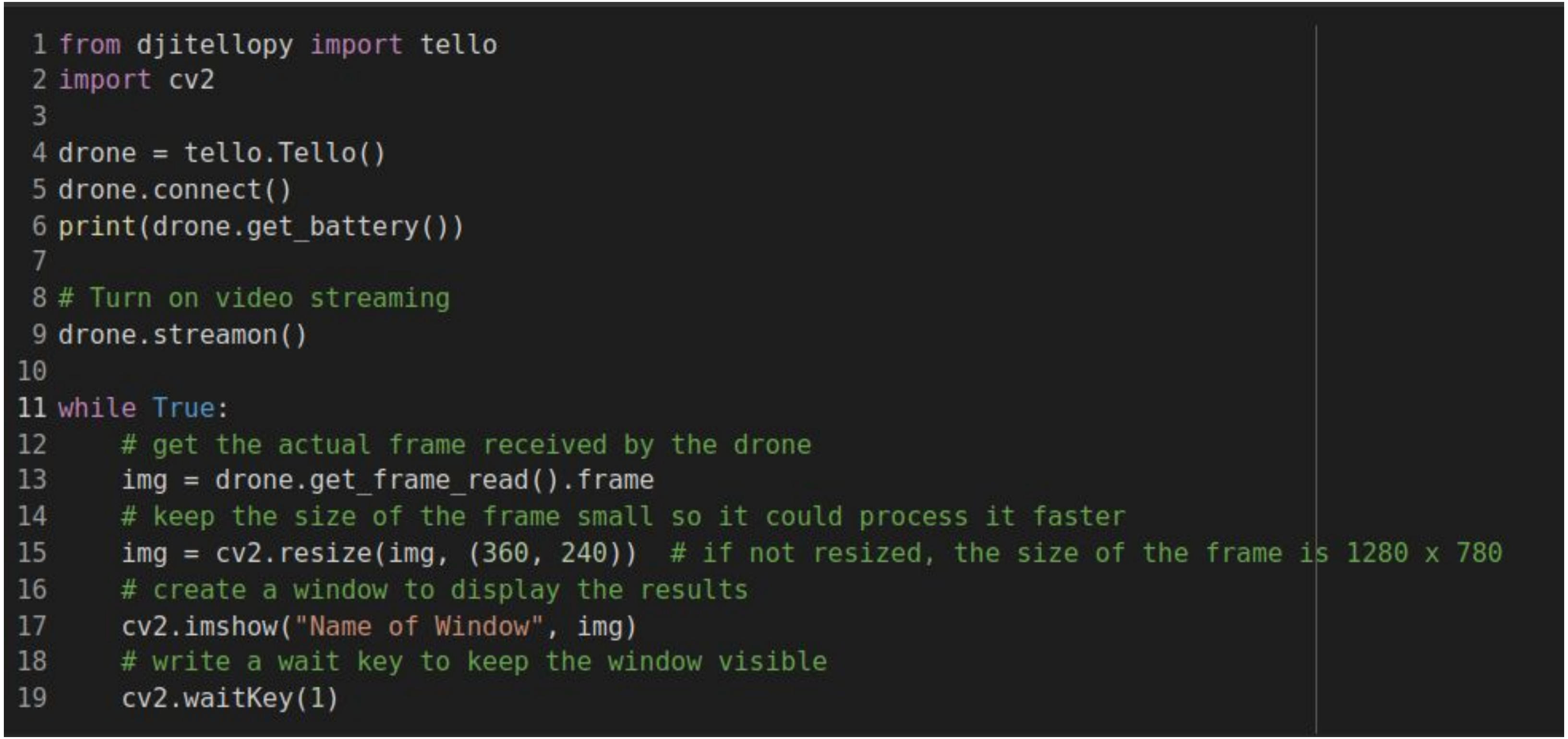
□ Options

Basic Movements



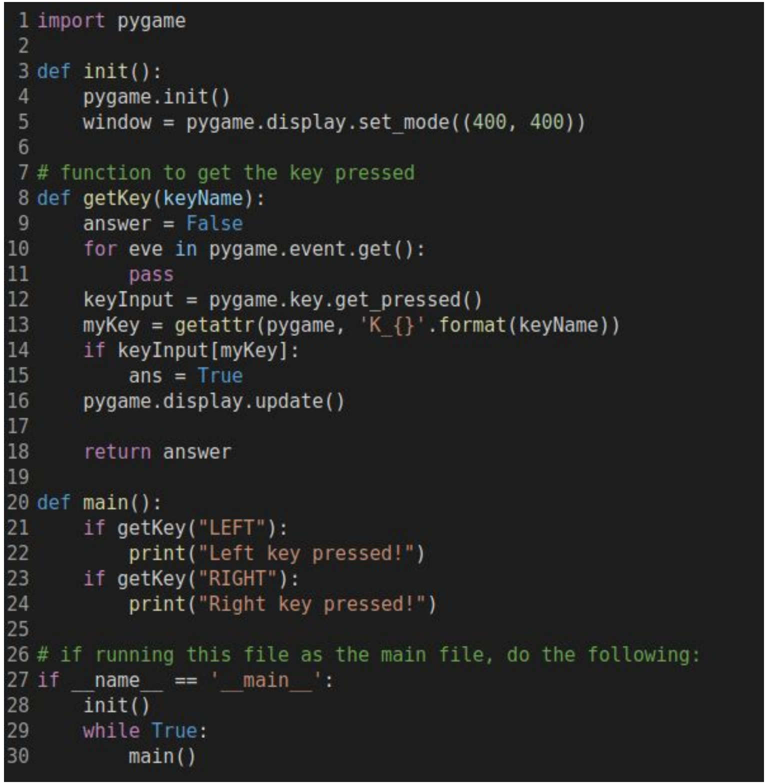
```python
from djitellopy import tello
from time import sleep

# create the drone object
drone = tello.Tello()
# connecting the drone through WiFi
drone.connect()
print(drone.get_battery())

# taking off
drone.takeoff()

"""
Send RC control via four channels
send_rc_control(left_right_velocity, forward_backward_velocity, up_down_velocity, yaw_velocity)
data type: int
range: -100 ~ 100
"""
drone.send_rc_control(0, 50, 0, 0)

# wait for 5 seconds
sleep(5)
# For safety reasons
drone.send_rc_control(0, 0, 0, 0)
# landing
drone.land()
```

Image Capturing



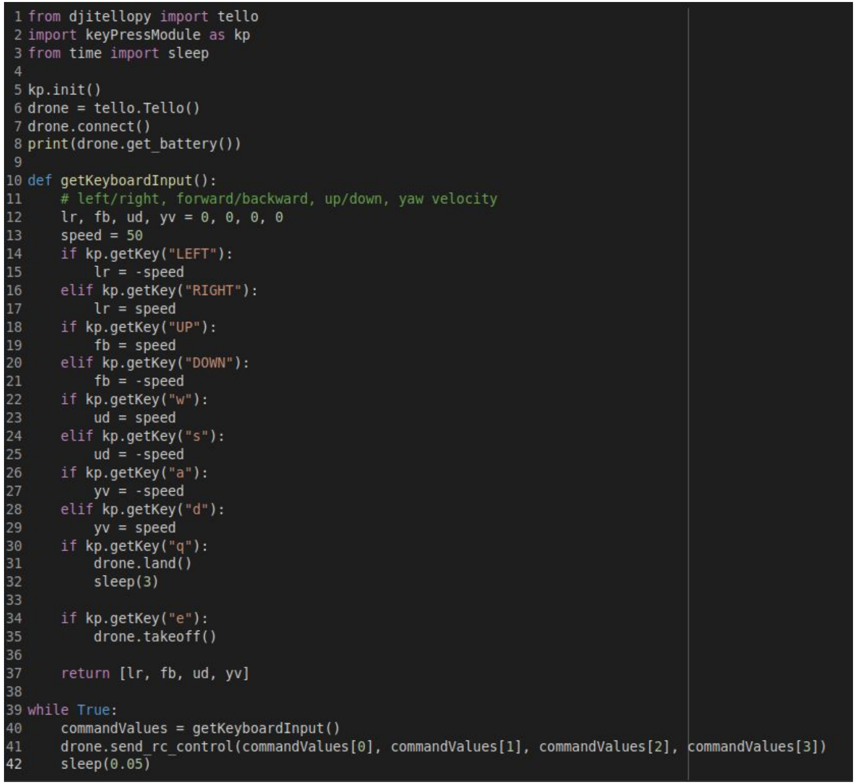
```python
from djitellopy import tello
import cv2

drone = tello.Tello()
drone.connect()
print(drone.get_battery())

# Turn on video streaming
drone.streamon()

while True:
    # get the actual frame received by the drone
    img = drone.get_frame_read().frame
    # keep the size of the frame small so it could process it faster
    img = cv2.resize(img, (360, 240))  # if not resized, the size of the frame is 1280 x 780
    # create a window to display the results
    cv2.imshow("Name of Window", img)
    # write a wait key to keep the window visible
    cv2.waitKey(1)
```

Key Press Module



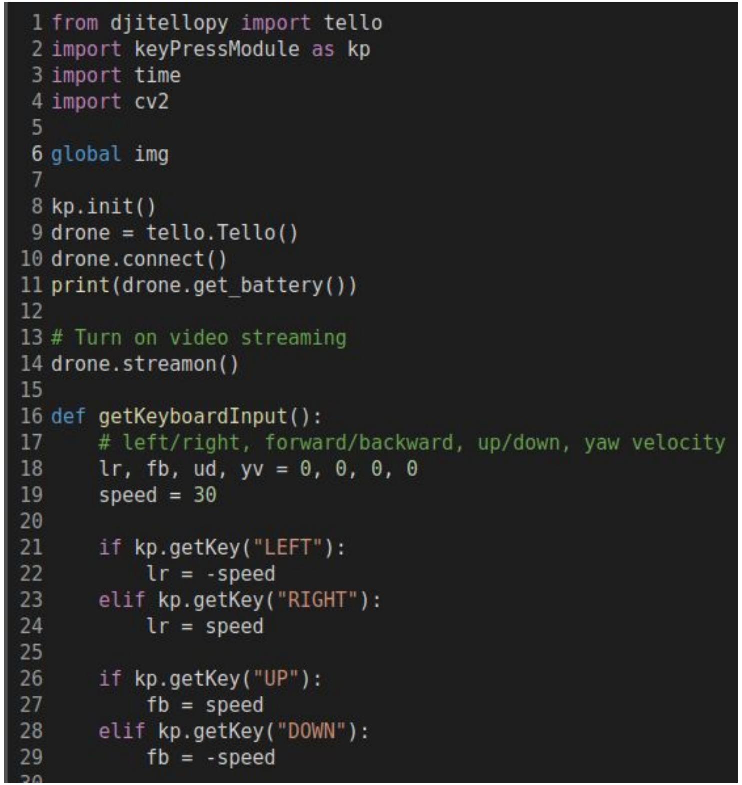
```python
import pygame

def init():
    pygame.init()
    window = pygame.display.set_mode((400, 400))

# function to get the key pressed
def getKey(keyName):
    answer = False
    for eve in pygame.event.get():
        pass
    keyInput = pygame.key.get_pressed()
    myKey = getattr(pygame, 'K_{}'.format(keyName))
    if keyInput[myKey]:
        ans = True
    pygame.display.update()

    return answer

def main():
    if getKey("LEFT"):
        print("Left key pressed!")
    if getKey("RIGHT"):
        print("Right key pressed!")

# if running this file as the main file, do the following:
if __name__ == '__main__':
    init()
    while True:
        main()
```

Keyboard Control

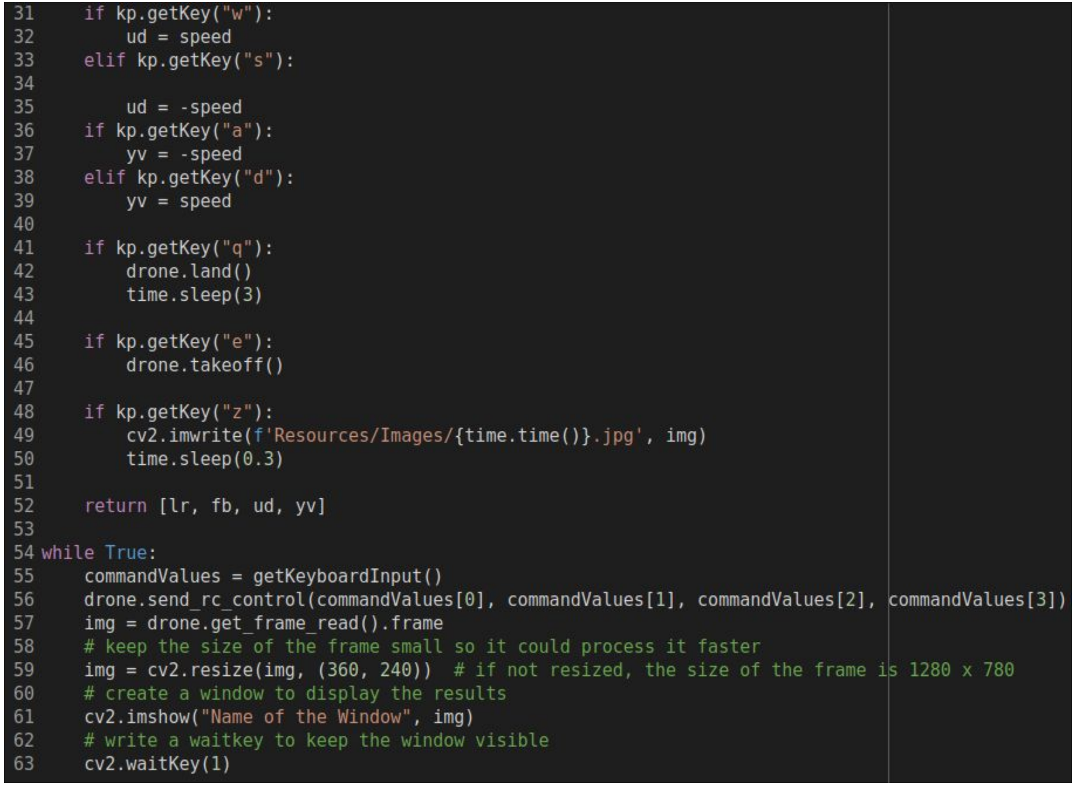


```python
from djitellopy import tello
import keyPressModule as kp
from time import sleep

kp.init()
drone = tello.Tello()
drone.connect()
print(drone.get_battery())

def getKeyboardInput():
    # left/right, forward/backward, up/down, yaw velocity
    lr, fb, ud, yv = 0, 0, 0, 0
    speed = 50
    if kp.getKey("LEFT"):
        lr = -speed
    elif kp.getKey("RIGHT"):
        lr = speed
    if kp.getKey("UP"):
        fb = speed
    elif kp.getKey("DOWN"):
        fb = -speed
    if kp.getKey("w"):
        ud = speed
    elif kp.getKey("s"):
        ud = -speed
    if kp.getKey("a"):
        yv = -speed
    elif kp.getKey("d"):
        yv = speed
    if kp.getKey("q"):
        drone.land()
        sleep(3)

    if kp.getKey("e"):
        drone.takeoff()

    return [lr, fb, ud, yv]

while True:
    commandValues = getKeyboardInput()
    drone.send_rc_control(commandValues[0], commandValues[1], commandValues[2], commandValues[3])
    sleep(0.05)
```

Surveillance



```python
from djitellopy import tello
import keyPressModule as kp
import time
import cv2

global img

kp.init()
drone = tello.Tello()
drone.connect()
print(drone.get_battery())

# Turn on video streaming
drone.streamon()

def getKeyboardInput():
    # left/right, forward/backward, up/down, yaw velocity
    lr, fb, ud, yv = 0, 0, 0, 0
    speed = 30

    if kp.getKey("LEFT"):
        lr = -speed
    elif kp.getKey("RIGHT"):
        lr = speed

    if kp.getKey("UP"):
        fb = speed
    elif kp.getKey("DOWN"):
        fb = -speed
```



```python
    if kp.getKey("w"):
        ud = speed
    elif kp.getKey("s"):

        ud = -speed
    if kp.getKey("a"):
        yv = -speed
    elif kp.getKey("d"):
        yv = speed

    if kp.getKey("q"):
        drone.land()
        time.sleep(3)

    if kp.getKey("e"):
        drone.takeoff()

    if kp.getKey("z"):
        cv2.imwrite(f'Resources/Images/{time.time()}.jpg', img)
        time.sleep(0.3)

    return [lr, fb, ud, yv]

while True:
    commandValues = getKeyboardInput()
    drone.send_rc_control(commandValues[0], commandValues[1], commandValues[2], commandValues[3])
    img = drone.get_frame_read().frame
    # keep the size of the frame small so it could process it faster
    img = cv2.resize(img, (360, 240))  # if not resized, the size of the frame is 1280 x 780
    # create a window to display the results
    cv2.imshow("Name of the Window", img)
    # write a waitkey to keep the window visible
    cv2.waitKey(1)
```

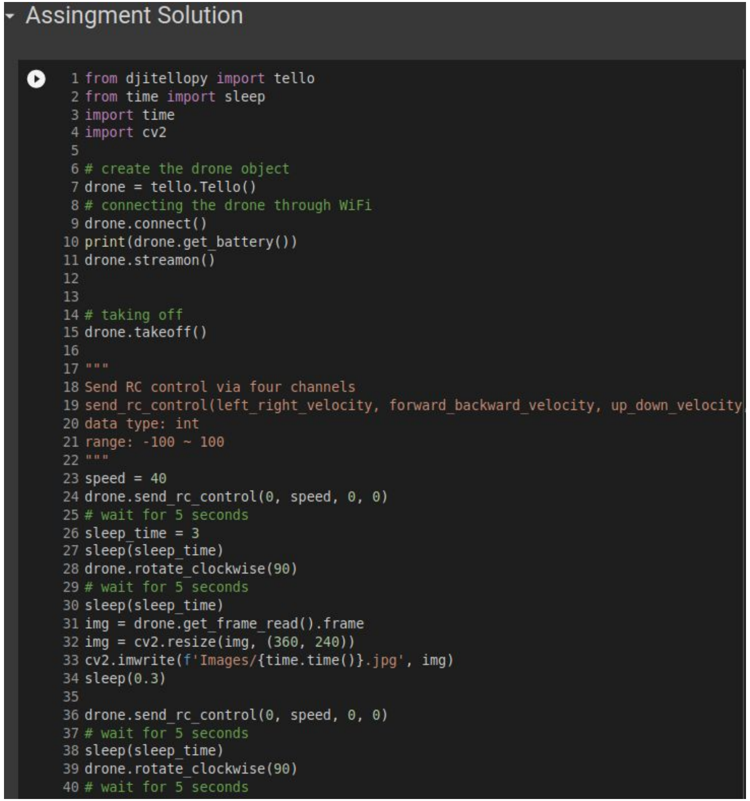
Task

Move your drone with a square trajectory and take 1 picture at each corner of the square.

At each side of the square, you must go forward 4 seconds.

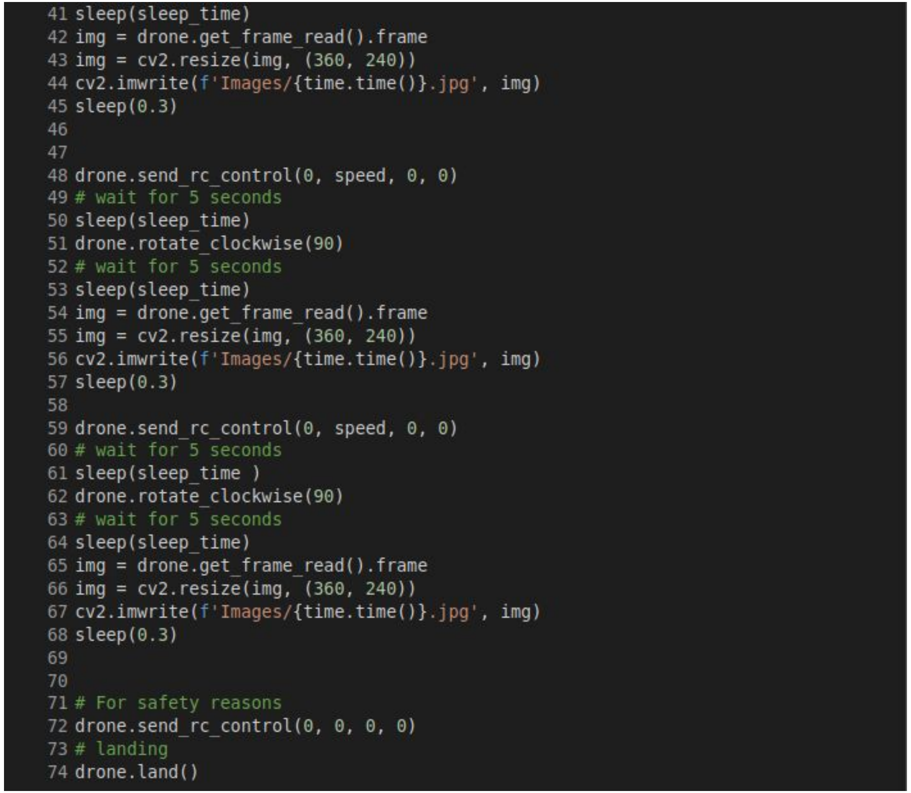
You must rotate your drone at each corner

Solution



Assingment Solution

```python
from djitellopy import tello
from time import sleep
import time
import cv2

# create the drone object
drone = tello.Tello()
# connecting the drone through WiFi
drone.connect()
print(drone.get_battery())
drone.streamon()


# taking off
drone.takeoff()

"""
Send RC control via four channels
send_rc_control(left_right_velocity, forward_backward_velocity, up_down_velocity,
data type: int
range: -100 ~ 100
"""
speed = 40
drone.send_rc_control(0, speed, 0, 0)
# wait for 5 seconds
sleep_time = 3
sleep(sleep_time)
drone.rotate_clockwise(90)
# wait for 5 seconds
sleep(sleep_time)
img = drone.get_frame_read().frame
img = cv2.resize(img, (360, 240))
cv2.imwrite(f'Images/{time.time()}.jpg', img)
sleep(0.3)

drone.send_rc_control(0, speed, 0, 0)
# wait for 5 seconds
sleep(sleep_time)
drone.rotate_clockwise(90)
# wait for 5 seconds
```



```python
sleep(sleep_time)
img = drone.get_frame_read().frame
img = cv2.resize(img, (360, 240))
cv2.imwrite(f'Images/{time.time()}.jpg', img)
sleep(0.3)


drone.send_rc_control(0, speed, 0, 0)
# wait for 5 seconds
sleep(sleep_time)
drone.rotate_clockwise(90)
# wait for 5 seconds
sleep(sleep_time)
img = drone.get_frame_read().frame
img = cv2.resize(img, (360, 240))
cv2.imwrite(f'Images/{time.time()}.jpg', img)
sleep(0.3)

drone.send_rc_control(0, speed, 0, 0)
# wait for 5 seconds
sleep(sleep_time )
drone.rotate_clockwise(90)
# wait for 5 seconds
sleep(sleep_time)
img = drone.get_frame_read().frame
img = cv2.resize(img, (360, 240))
cv2.imwrite(f'Images/{time.time()}.jpg', img)
sleep(0.3)


# For safety reasons
drone.send_rc_control(0, 0, 0, 0)
# landing
drone.land()
```