Lab05

Marker(測距離) & Tello EDU

1. Marker Detection (50%)

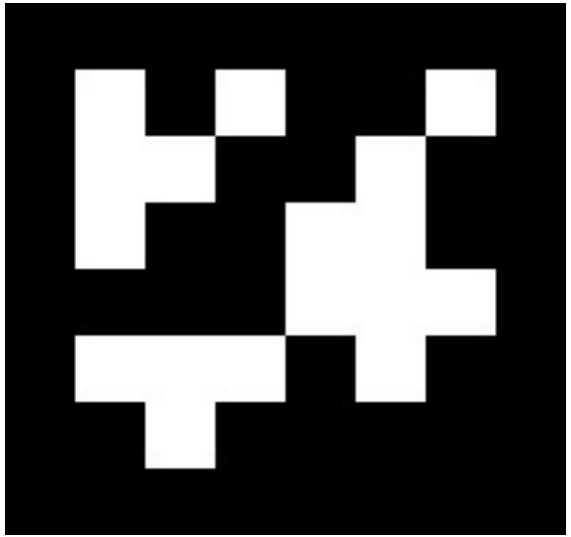
a. calibration

b. marker detection

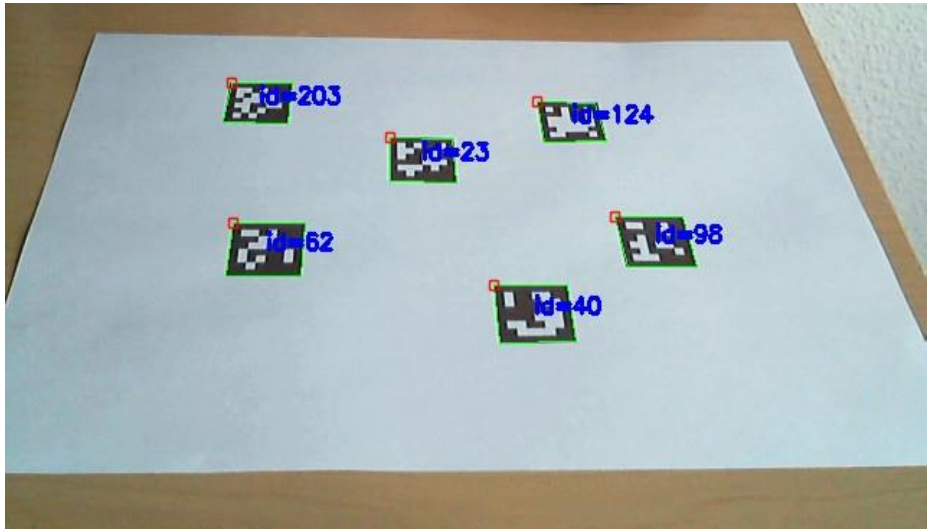
c. pose estimation

d. controlling

1. Marker Detection (50%)



aruco marker



marker detection

1. Marker Detection (50%)

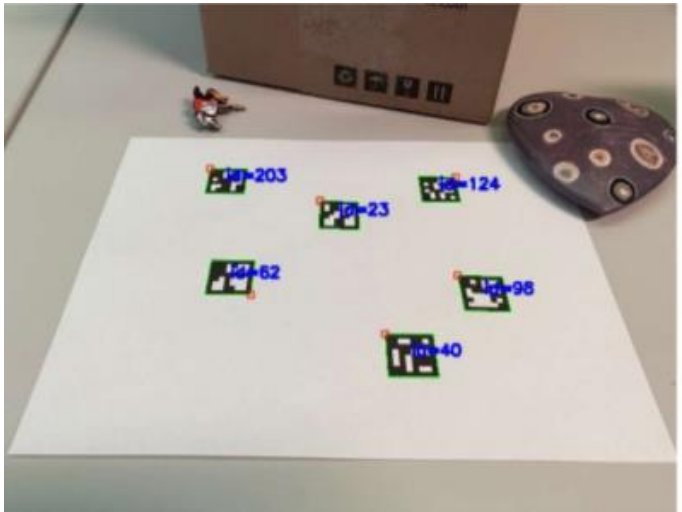
```
# Load the predefined dictionary
dictionary = cv2.aruco.Dictionary_get(cv.aruco.DICT_6X6_250)

# Initialize the detector parameters using default values
parameters =  cv2.aruco.DetectorParameters_create()
```

1. Marker Detection (50%)

```
# Detect the markers in the image
markerCorners, markerIds, rejectedCandidates =
cv2.aruco.detectMarkers(frame, dictionary, parameters=parameters)

frame = cv2.aruco.drawDetectedMarkers(frame,
markerCorners, markerIds)
```

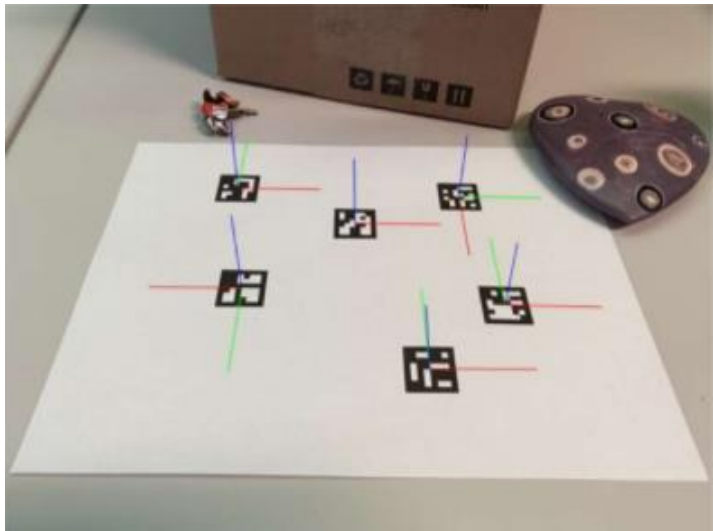


1. Marker Detection (50%)

#Pose estimation for single markers.

rvec, tvec, _objPoints = cv2.aruco.estimatePoseSingleMarkers(markerCorners, 15, intrinsic, distortion)

frame = cv2.aruco.drawAxis(frame, intrinsic, distortion, rvec, tvec, 0.1)

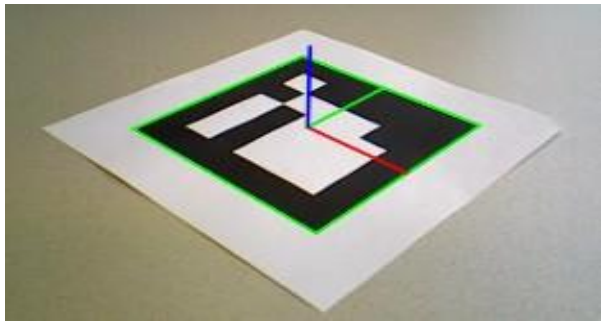


1. Marker Detection (50%)

a. calibrate the drone camera

b. marker detection by drone camera

c. pose estimation



x: 10.3478

y: 21.5618

z: 3.9908

2. Tello EDU (50%)

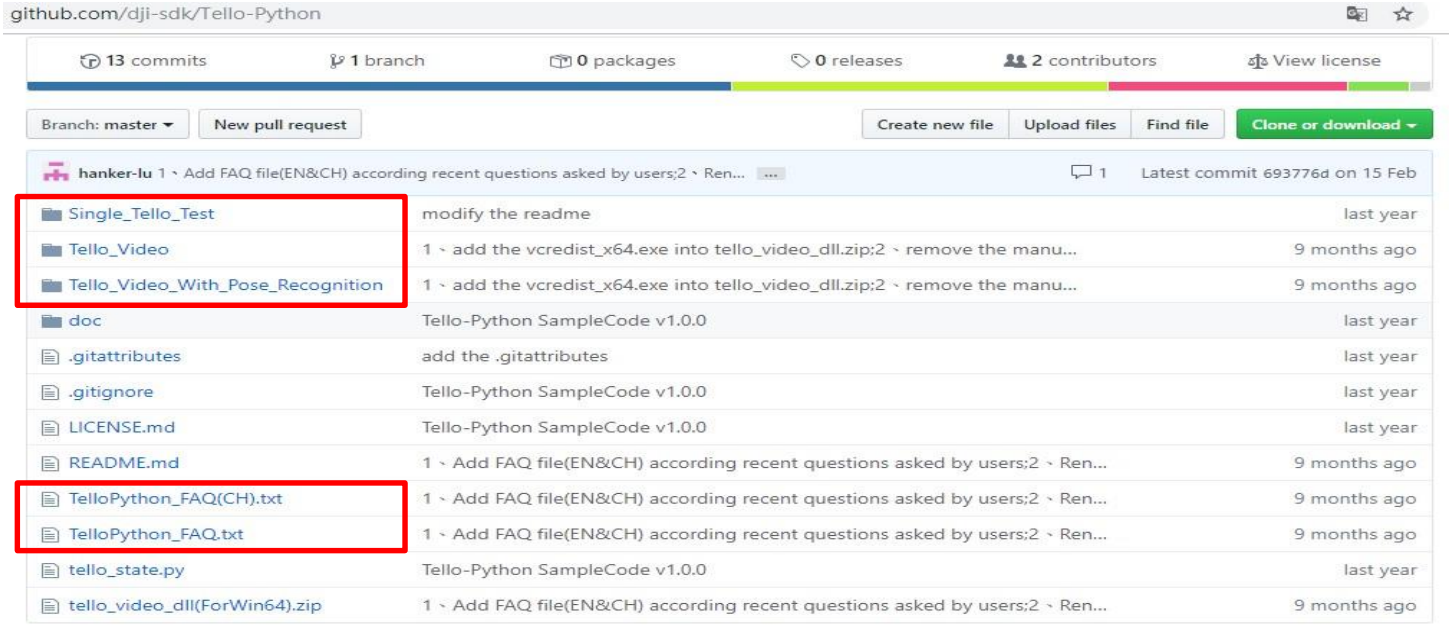
設備介紹

配件		數量
	飛行器	× 1
	螺旋槳（對）	× 4
	槳葉保護罩（套）	× 1
	電池	× 1
	Micro USB 傳輸線	× 1
	螺旋槳拆卸工具	× 1
	挑戰卡	× 4

電池管家

官方範例程式

- https://github.com/dji-sdk/Tello-Python



官方範例程式：Tello-Video

- https://github.com/dji-sdk/Tello-Python/tree/master/Tello_Video
- Written in Python 2.7
- 安裝：

- **Windows**

  Go to the "install\Windows" folder,select and run the correct "windows_install.bat" according to your computer operating system bits.

- **Linux (Ubuntu 14.04 and above)**

  Go to the "install\Linux" folder in command line, run

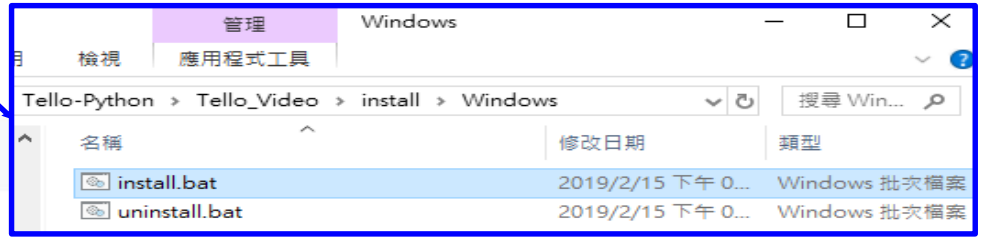
```
chmod +x linux_install.sh
./linux_install.sh
```

- **Mac**

  i. Make sure you have the latest Xcode command line tools installed. If not, you might need to update your OS X and XCode to the latest version in order to compile the h264 decoder module

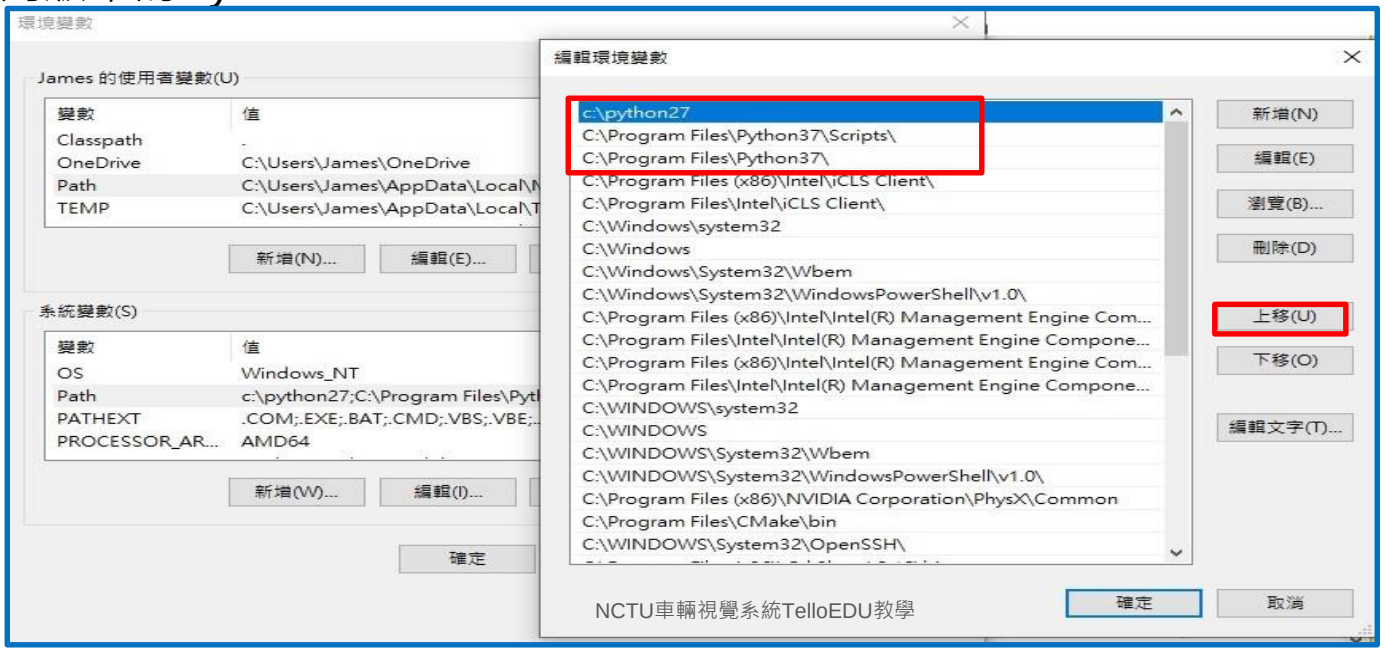
  ii. Go to the "install\Mac" folder folder in command line, run

```
chmod a+x ./mac_install.sh
./mac_install.sh
```



官方範例程式：Tello-Video

- 不同版本的Python

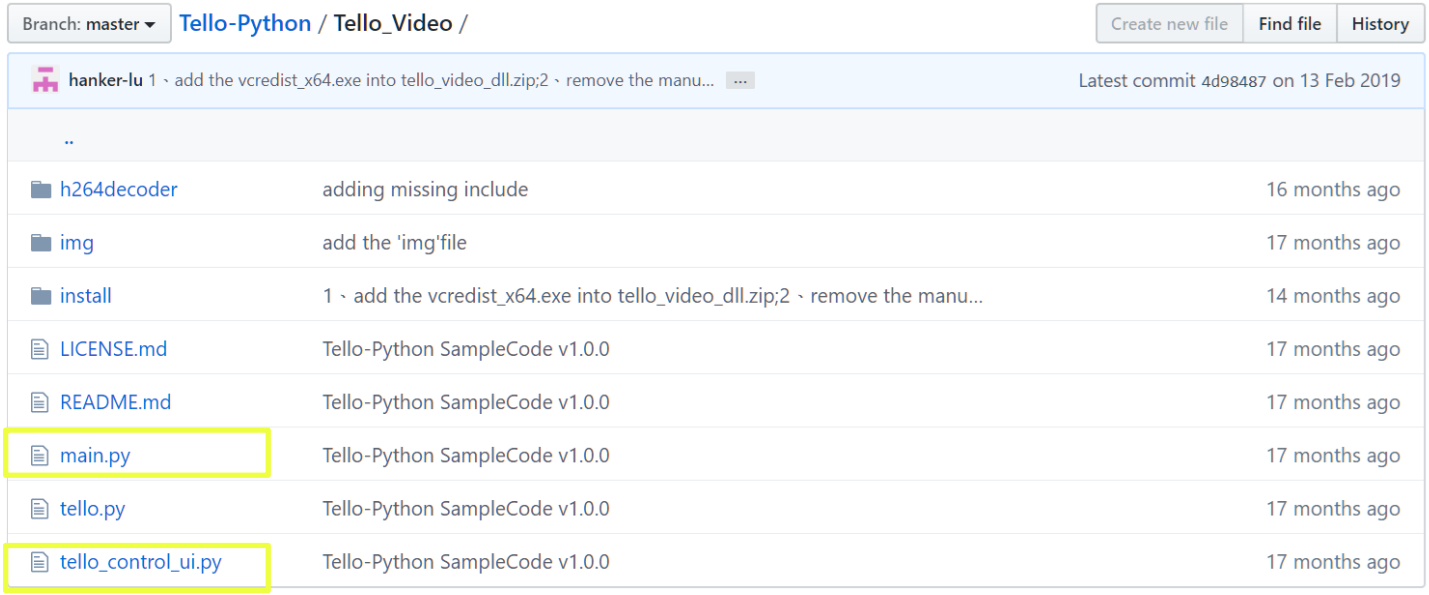


官方範例程式：執行 Tello-Video

- 第一步。打開Tello無人機，並透過Wi-Fi將筆電連接到Tello

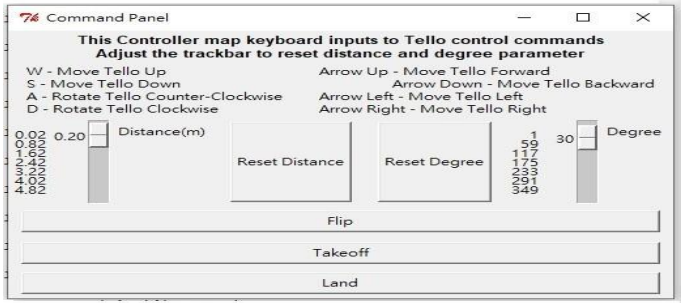
官方範例程式：執行 Tello-Video

- 第一步。打開Tello無人機，並透過Wi-Fi將筆電連接到Tello



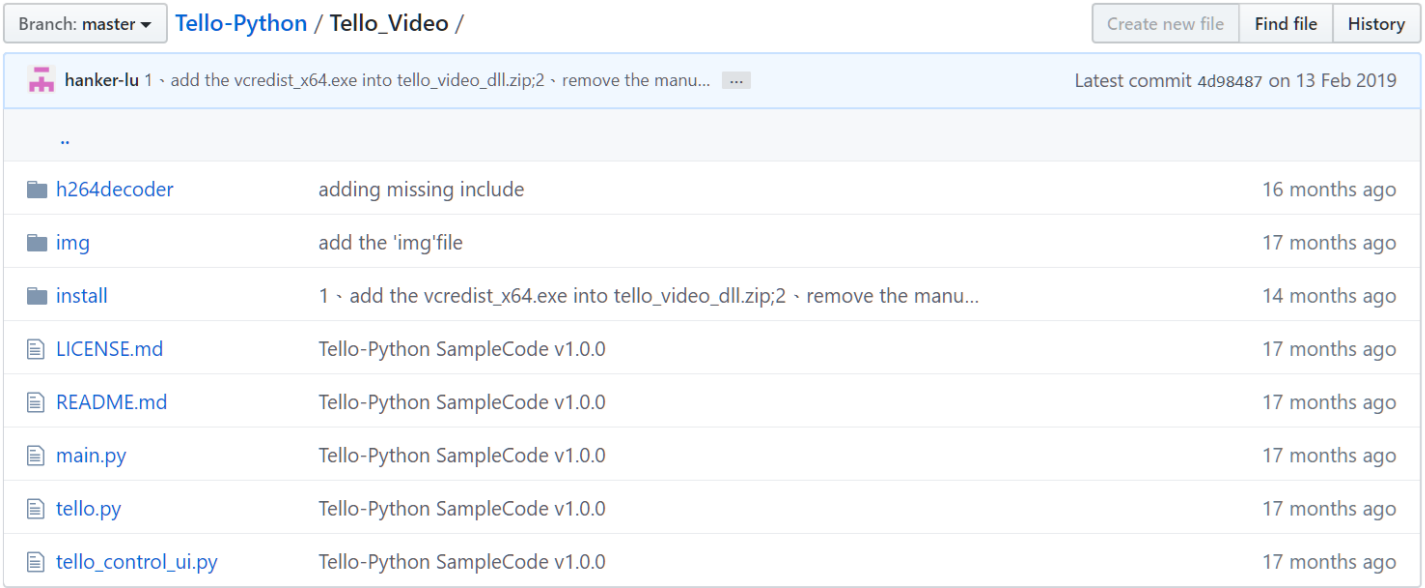
官方範例程式：執行 Tello-Video

- 在命令列打開程式所在資料夾，並輸入 python main.py, 將顯示一個使用者界面.



官方範例程式：執行 Tello-Video

- 第一步。打開Tello無人機，並透過Wi-Fi將筆電連接到Tello



放入lab05.py

tello.py

```
 5  # import libh264decoder
25              # self.decoder = libh264decoder.H264Decoder()
51              # self.socket_video.bind((local_ip, self.local_video_port))
```

註解掉def _h264_decode 跟 def _receive_video_thread

tello.py

```
def _receive_video_thread(self):
        video_ip = "udp://{}:{}".format("0.0.0.0", 11111)
        video_capture = cv2.VideoCapture(video_ip)
        retval, self.frame = video_capture.read()
        while retval:
            retval, frame = video_capture.read()
            self.frame = frame[..., ::-1] # From BGR to RGB
```

tello_control_ui.py

import Tkinter as tki

from Tkinter import Toplevel, Scale

改成

import tkinter as tki

from tkinter import Toplevel, Scale

tello_control_ui.py

except RuntimeError, e:

改成

```
100         except RuntimeError as e:
```

所有的print全部加括號

print('reset distance to %.1f' % self.distance)

取像控制

在Tello_Video內

放入lab05.py

```
import tello
import cv2
import time

def main():
    drone = tello.Tello('', 8889)

    time.sleep(10)

    while(True):
        frame = drone.read()
        frame = cv2.cvtColor(frame, cv2.COLOR_RGB2BGR)


        cv2.imshow("drone", frame)
        key = cv2.waitKey(1)


        if key!= -1:
            drone.keyboard(key)

    cv2.destroyAllWindows()

if __name__ == "__main__":
    main()
```

Tello API

```
self.tello.takeoff()
self.tello.land()
self.tello.flip('l')
self.tello.flip('r')
self.tello.flip('f')
self.tello.flip('b')
self.tello.rotate_cw(degree)
self.tello.rotate_ccw(degree)
self.tello.move_forward(distance)
self.tello.move_backward(distance)
self.tello.move_left(distance)
self.tello.move_right(distance)
self.tello.move_up(dist)
self.tello.move_down(dist)
```

distance和dist的單位為公尺！

所以在測試時，建議distance和dist的數字不要設超過**"1"**！

```
def keyboard(self, key):
    print("key:", key)
    distance = 0.9
    degree = 30

    if key == ord('1'):
        self.takeoff()
    if key == ord('2'):
        self.land()
    if key == ord('i'):
        self.move_forward(distance)
        print("forward!!!!")
    if key == ord('k'):
        self.move_backward(distance)
        print("backward!!!!")
    if key == ord('j'):
        self.move_left(distance)
        print("left!!!!")
    if key == ord('l'):
        self.move_right(distance)
        print("right!!!!")
    if key == ord('s'):
        self.move_down(distance)
        print("down!!!!")
    if key == ord('w'):
        self.move_up(distance)
        print("up!!!!")
    if key == ord('a'):
        self.rotate_cw(degree)
        print("rotate!!!!")
    if key == ord('d'):
        self.rotate_ccw(degree)
        print("counter rotate!!!!")
    if key == ord('5'):
        height = self.get_height()
        print(height)
    if key == ord('6'):
        battery = self.get_battery()
        print (battery)
```

- Note:
  往後撰寫自動飛行的程式碼時,
  一定也要有 keyboard control 功能,
  且要有最高優先權,
  確保自動飛行狀況不佳時仍能手動控制。

- 將keyboard加入 tello.py 中的Tello物件

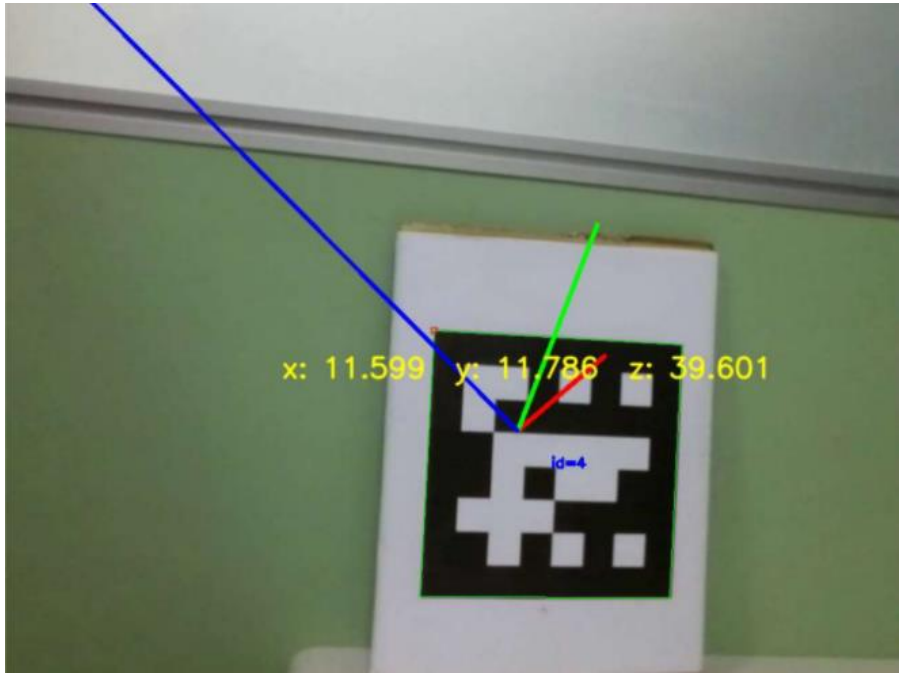
Keyboard control

- Note:
  往後撰寫自動飛行的程式碼時,
  一定也要有 keyboard control 功能,
  且要有最高優先權,
  確保自動飛行狀況不佳時仍能手動控制。

```
"*****************************************"
"*       CV Drone sample program         *"
"*           - How to play -             *"
"*****************************************"
"*                                       *"
"* - Controls -                          *"
"*    'Space' -- Takeoff/Landing         *"
"*    'Up'    -- Move forward            *"
"*    'Down'  -- Move backward           *"
"*    'Left'  -- Turn left               *"
"*    'Right' -- Turn right              *"
"*    'Q'     -- Move upward             *"
"*    'A'     -- Move downward           *"
"*                                       *"
"* - Others -                            *"
"*    'C'     -- Change camera           *"
"*    'Esc'   -- Exit                    *"
"*                                       *"
"*****************************************"
```

測距離

利用cv2.putText()



測距離

若找不到aruco module

pip install opencv-contrib-python