Tag code positioning experiment

Apriltag is a coded mark commonly used in machine vision. It has a high recognition rate and reliability and can be used for various tasks, including augmented reality, robotics, and camera calibration. This apriltag tag code uses the TAG36H11 format. The factory has been equipped with relevant tag codes and attached to the building blocks. You need to take out the building blocks and place them under the camera screen for recognition.

Code path:

```
~/dofbot_ws/src/dofbot_apriltag/scripts/Apriltag_Position.ipynb
```

1. Main code

- Import header file

```
import cv2 as cv
import threading
import random
from time import sleep
import ipywidgets as widgets
from IPython.display import display
from apriltag_identify import ApriltagIdentify
from dofbot_utils.fps import FPS
from dofbot_utils.robot_controller import Robot_Controller
```

- Create an instance and initialize parameters

```
apriltag_Identify = ApriltagIdentify()
model = 'General'
robot = Robot_Controller()
robot.move_look_map()
fps = FPS()
```

- Main process

```
def camera():
    global HSV_learning,model
    # 打开摄像头 Open camera
    capture = cv.VideoCapture(0)
    capture.set(3, 640)
    capture.set(4, 480)
    capture.set(5, 30)
    # Be executed in loop when the camera is opened normally
    # 当摄像头正常打开的情况下循环执行
    while capture.isOpened():
        try:
            _, img = capture.read()
            fps.update_fps()
            img, msg = apriltag_Identify.getApriltagPosition(img)
```

```
            if len(msg):
                print(msg)
            if model == 'Exit':
                capture.release()
                break
            fps.show_fps(img)
            imgbox.value = cv.imencode('.jpg', img)[1].tobytes()
        except Exception as e:
            print("program end")
            print(e)
            capture.release()
```

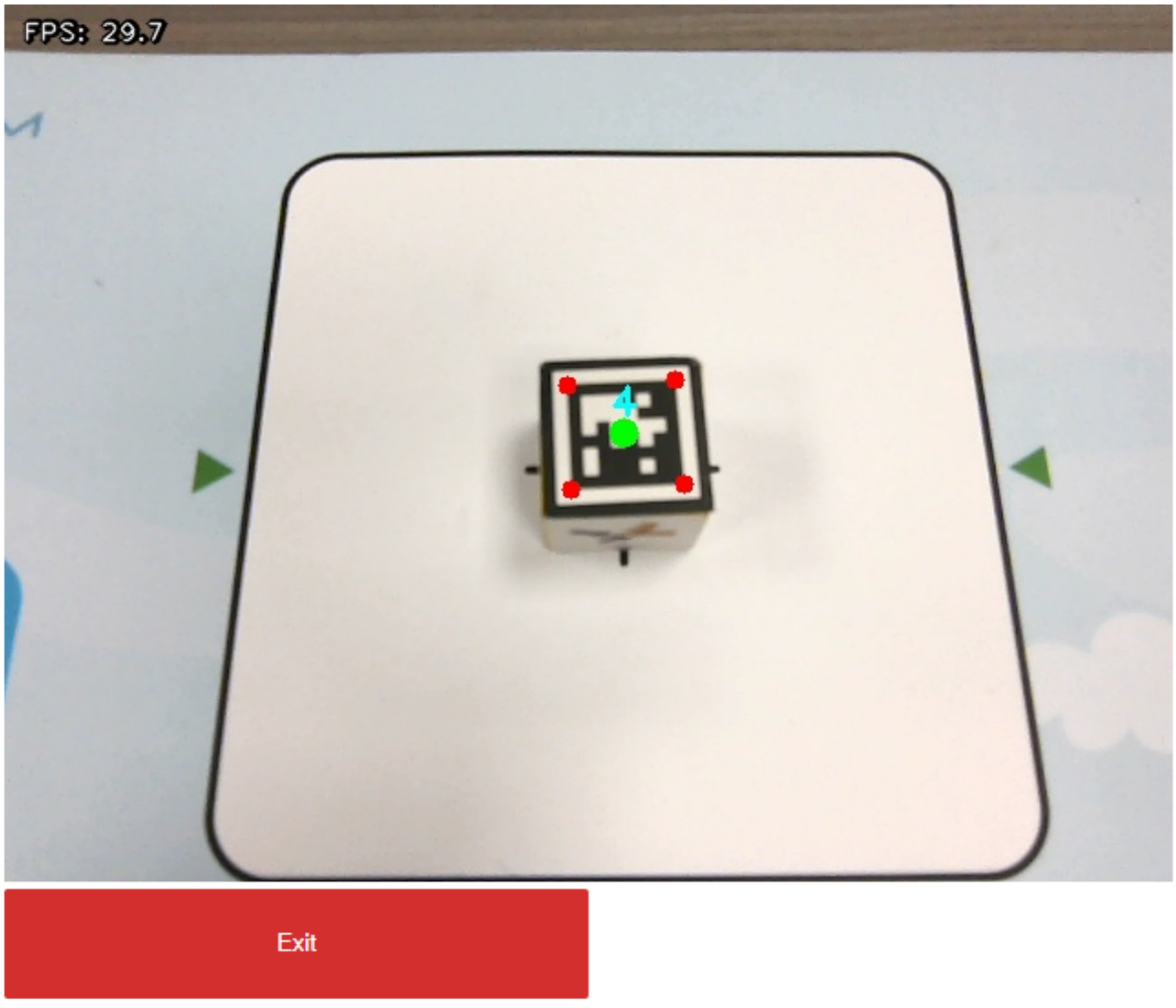
- start up

```
display(controls_box,output)
threading.Thread(target=camera, ).start()
```

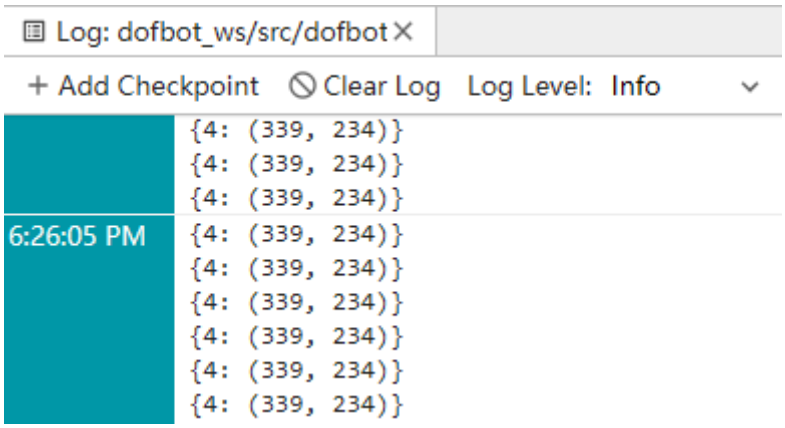
2. Run the program

Click the Run the entire program button on the jupyterlab toolbar, and then pull it to the bottom.

After the program starts, put the label code block into the camera screen, you can see the correct positioning of the label code and the ID number written on the label code. Move the label code and the recognition effect will move with it.



Open the Log information, select Log Level as Info, and you can see the ID number and location coordinates of the label code printed out.



If you need to end the program, please click [Exit] to avoid affecting other programs calling resources.