5. Color follow

5.1 Gameplay Introduction

This course mainly adds the function of PID control of car movement on the basis of color tracking. After opening the start switch, the car will adjust its position to keep the color object in the middle of the picture as far as possible.

5.2 Core Content analysis

The simplePID.py file is a simplePID controller, and the incremental function is an incremental PID algorithm function.

```
def incremental(self, current_value, limit=0):
        self.err = self.setPoint - current_value
        result = self.last_result + self.Kp * (self.err - self.err_next) +
self.Ki * self.err + self.Kd * (self.err - 2 * self.err_next + self.err_last)
        self.err_last = self.err_next
        self.err_next = self.err
        if limit > 0:
            if result > limit:
                result = limit
            if result < -limit:
                result = -limit
        self.last_result = result
        return result
```

Import and configure PID parameters, since the image pixels are 320*240, take the midpoint values of 160 and 120 as the target values. The PID parameter value can be adjusted according to the robot's motion effect.

```
from simplePID import PID

PID_X_Kp = 0.8
PID_X_Ki = 0
PID_X_Kd = 0.2

PID_Y_Kp = 0.8
PID_Y_Ki = 0
PID_Y_Kd = 0.2

PID_X = PID(160, PID_X_Kp, PID_X_Ki, PID_X_Kd)
PID_Y = PID(120, PID_Y_Kp, PID_Y_Ki, PID_Y_Kd)
```

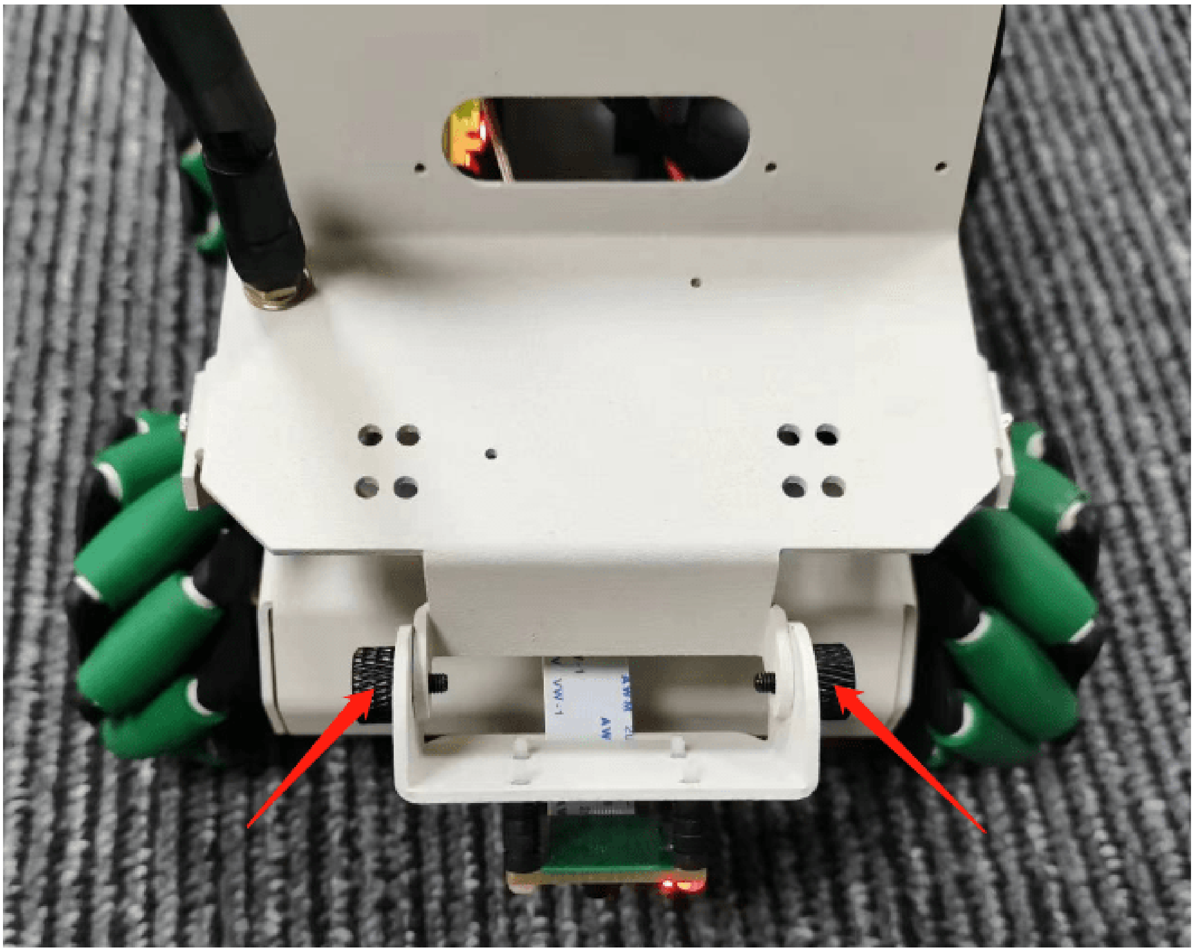
When the Start button is clicked, the car_control function is called to control the car movement. In order to keep the color object in the middle of the screen, the car moves with the color object.

```
from SunriseRobotLib import SunriseRobot
g_car = SunriseRobot()

def car_control(x, y, radius):
    global last_x, last_y, last_radius
    speed_x = PID_X.incremental(y)/100.0
    speed_y = PID_Y.incremental(x)/100.0
    speed_z = 0
    if -0.1 < speed_x < 0.1:
        speed_x = 0
    if -0.1 < speed_y < 0.1:
        speed_y = 0
    if speed_x > 0.5:
        speed_x = 0.5
    if speed_x < -0.5:
        speed_x = -0.5
    if speed_y > 0.5:
        speed_y = 0.5
    if speed_y < -0.5:
        speed_y = -0.5
    print("speed:", speed_x, speed_y, speed_z)
    g_car.set_car_motion(speed_x, speed_y, speed_z)
    last_x, last_y, last_radius = x, y, radius
```

In order for the car to better follow the color object, it is necessary to adjust the Angle of the camera bracket.

First, run all cells, loosen the screws on the left and right sides of the CSI camera bracket without starting the car movement, then adjust the Angle of the camera bracket, observe the picture captured by the camera, and tighten the screws at the appropriate position to fix the Angle of the camera bracket.

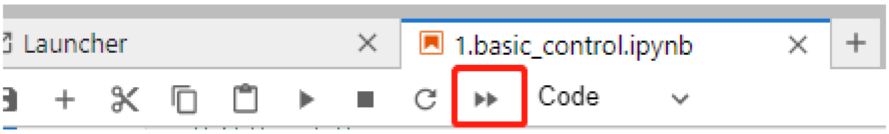


5.3 Gameplay operation

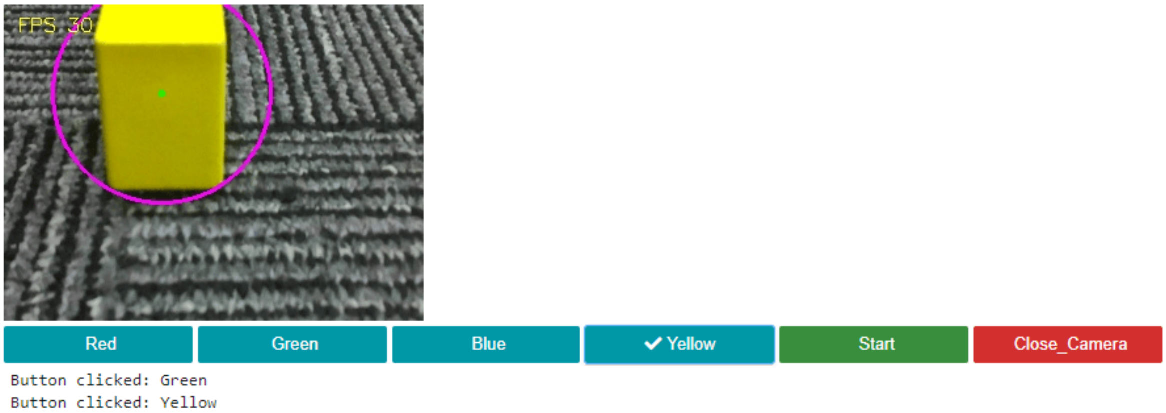
Open the jupyterLab client and find the code path:

```
/root/sunriseRobot/Samples/2_AI/05_color_following/color_following.ipynb
```

Click Run All Cells, and then drag to the bottom to see the generated controls.



The camera will track red objects, and if you need to track other objects, click the button below to switch. Click the Start button and the car will start to move, keeping the color object in the middle of the camera screen. At this time, moving the color object will move the car.



```
Button clicked: Green
Button clicked: Yellow
```

Finally click the Close_Camera button to close the camera.