Color Tracking

4.1 Introduction to the Gameplay

This course mainly uses the IMX219 camera to obtain the camera image, select a color, and then put the object of the selected color into the camera image. Use a circle to frame the color object in the image in real time. Move the color object and the circle will track the color object. Note that the car will not move during this process, but the camera image will draw a circle to track the color object.

4.2 Core Content Analysis

Color_HSV defines the HSV value of the color. Due to the color difference and light influence of the object color, if the recognition effect of a certain color is inaccurate, you can adjust the course according to the color HSV value. After adjusting the best effect, record the data and update the HSV value of the corresponding color.

```
Color_HSV = {
'Red': ([0, 43, 46], [10, 255, 255]),
'Green': ([35, 43, 46], [77, 255, 255]),
'Blue': ([100, 43, 46], [124, 255, 255]),
'Yellow': ([26, 43, 46], [34, 255, 255])
}
```

By default, track red objects.

```
color_lower = np.array(Color_HSV["Red"][0])
color_upper = np.array(Color_HSV["Red"][1])
```

Every time a key is pressed, update the color_lower and color_upper values of the color HSV value.

```
def on_button_clicked(b):
    global color_lower, color_upper, g_stop_program
    ALL_Uncheck()
    b.icon = 'check'
    with output:
        print("Button clicked:", b.description)
    if b.description == Name_widgets['Close_Camera'][g_ENABLE_CHINESE]:
        g_stop_program = True
        time.sleep(.1)
        g_camera.release()
        b.icon = 'uncheck'
    elif b.description == Name_widgets['Red'][g_ENABLE_CHINESE]:
        color_lower = np.array(np.array(Color_HSV["Red"][0]))
        color_upper = np.array(np.array(Color_HSV["Red"][1]))
```

```
    elif b.description == Name_widgets['Green'][g_ENABLE_CHINESE]:
        color_lower = np.array(np.array(Color_HSV["Green"][0]))
        color_upper = np.array(np.array(Color_HSV["Green"][1]))
    elif b.description == Name_widgets['Blue'][g_ENABLE_CHINESE]:
        color_lower = np.array(np.array(Color_HSV["Blue"][0]))
        color_upper = np.array(np.array(Color_HSV["Blue"][1]))
    elif b.description == Name_widgets['Yellow'][g_ENABLE_CHINESE]:
        color_lower = np.array(np.array(Color_HSV["Yellow"][0]))
        color_upper = np.array(np.array(Color_HSV["Yellow"][1]))
```

处理摄像头画面的程序功能，将读取摄像头的图像，然后进行HSV值解析，得到颜色物体的位置信息，并用圆框画出，最后通过图像控件显示。

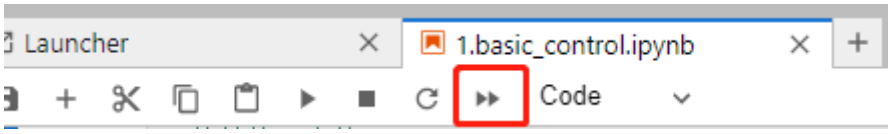
```
def task_processing():
    global color_lower, color_upper, g_stop_program
    t_start = time.time()
    m_fps = 0
    while g_camera.isOpened():
        if g_stop_program:
            break
        ret, frame = g_camera.read()
        hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
        mask = cv2.inRange(hsv, color_lower, color_upper)
        cnts = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)[-2]
        if len(cnts) > 0:
            cnt = max (cnts, key = cv2.contourArea)
            (color_x, color_y), color_radius = cv2.minEnclosingCircle(cnt)
            if color_radius > 10:
                cv2.circle(frame, (int(color_x), int(color_y)),
int(color_radius), (255,0,255), 2)
        m_fps = m_fps + 1
        fps = m_fps / (time.time() - t_start)
        if (time.time() - t_start) >= 2:
            m_fps = fps
            t_start = time.time() - 1
        cv2.putText(frame, "FPS " + str(int(fps)), (10,20),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,255,255), 1)
        image_widget.value = bgr8_to_jpeg(frame)
        time.sleep(.01)
```

4.3 How to play

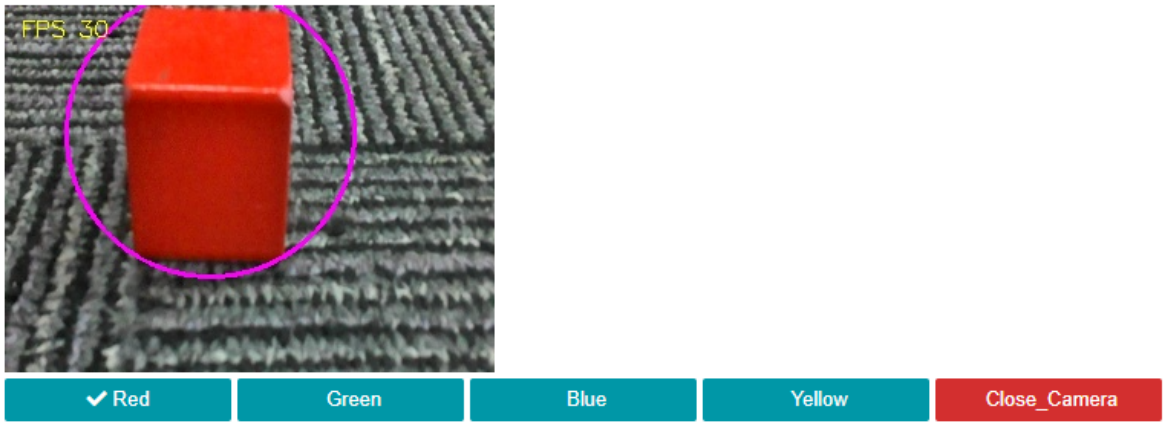
Open the jupyterLab client and find the code path:

```
/home/sunrise/sunriseRobot/Samples/2_AI/04_color_tracking/color_tracking.ipynb
```

Click to run all cells, then pull to the bottom to see the generated controls.



The camera will track the red object. If you need to track other objects, click the button below to switch.



Finally, click the Close_Camera button to close the camera.