7、greet people

7.1 Introduction to gameplay

This course mainly uses the robot's camera. On the basis of face detection, it also adds a greeting action when a face is detected.

7.2 Core content analysis

This course mainly uses the robot's camera. On the basis of face detection, it also adds a greeting action when a face is detected.

```
from MutoLib import Muto
g_bot = Muto()
g_bot.Gimbal_1_2(90, 30)
```

The camera image is analyzed and processed to extract the position information of the face detected in the image.

```
def task_processing():
    global g_stop_program, g_start_function

    t_start = time.time()
    m_fps = 0
    fps = 0
    while g_camera.isOpened():
        if g_stop_program:
            break
        ret, frame = g_camera.read()
        if g_start_function:
            state, (face_x,face_y,face_w,face_h) = face_detect(frame)
            if state != None:
                cv2.rectangle(frame, (face_x, face_y), (face_x+face_w,
face_y+face_h), (0,255,255), 1)
                robot_action(face_x,face_y,face_w,face_h)
                # time.sleep(.01)

        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
        if g_start_function:
            cv2.putText(frame, "START " + str(int(fps)), (10,20), 
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0,255,255), 2)
        else:
            cv2.putText(frame, "FPS " + str(int(fps)), (10,20), 
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0,255,255), 2)

        # 图像传输给显示组件 The image is transmitted to the display component
        image_widget.value = bgr8_to_jpeg(frame)
```

The information of the face detected in the image is analyzed and processed. If the face is detected and the face position meets the requirements for triggering a greeting, the greeting action is executed. The default trigger condition is that the face position is not at the edge, and the width and height are greater than one-eighth of the image. The judgment parameters can also be adjusted according to the actual effect.

```
def robot_action(face_x,face_y,face_w,face_h):
    global g_time_pass
    # print("face:", face_x,face_y,face_w,face_h)
    if g_time_pass >= 0:
        return
    if int(img_width*0.1) <= face_x <= int(img_width*0.9) and \
        int(img_height*0.1) <= face_y <= int(img_height*0.9) and \
        face_w >= int(img_width/8) and face_h >= int(img_width/8):
        # 打招呼 greet
        g_bot.action(2)
        g_time_pass = 10
        print("greet:", face_x,face_y,face_w,face_h)
```

The movement time of the greeting action is timed out. Before the movement time is over, the greeting command will no longer be run to avoid frequently sending greeting commands.

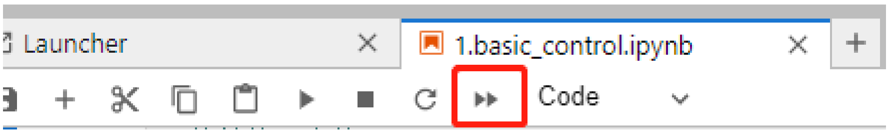
```
def task_time():
    global g_time_pass
    while True:
        if g_time_pass > 0:
            g_time_pass = int(g_time_pass - 1)
            if g_time_pass == 0:
                g_bot.zero_reset()
                time.sleep(.5)
                g_time_pass = -1
        time.sleep(.1)
```

7.3 operation

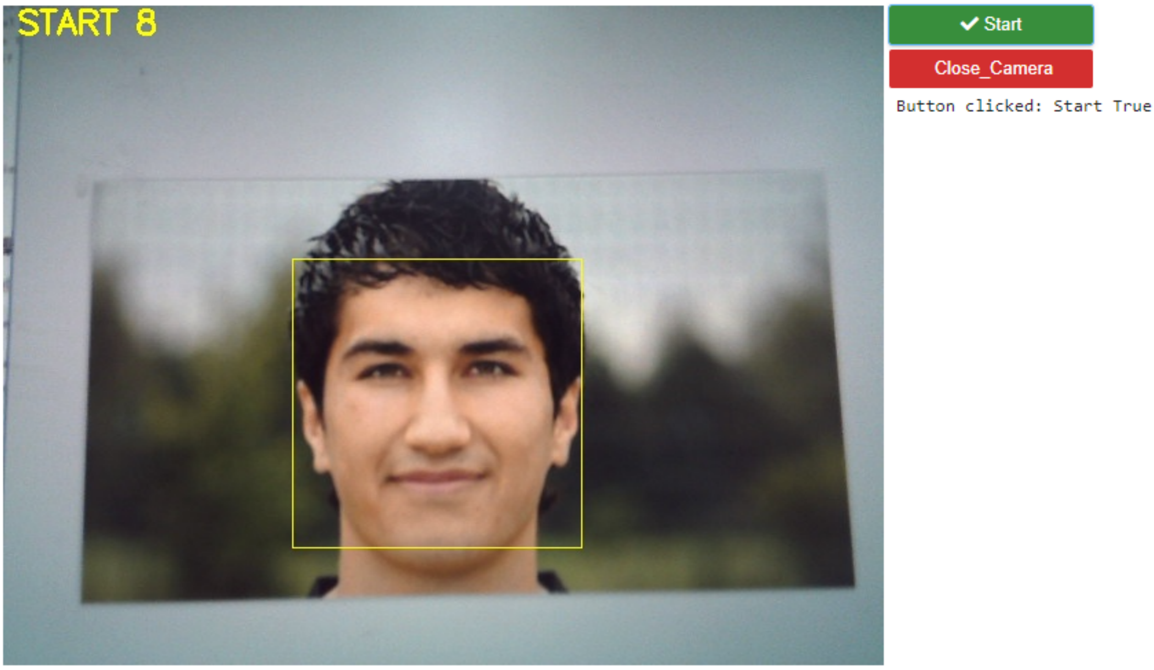
Open the jupyterLab client and find the code path:

```
muto/Samples/AI_Samples/07_greet/greet.ipynb
```

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



When the display control is started, it only displays the camera image. You need to click the Start button to turn on the face detection and robot action functions. When a face is detected in the image and the position and size of the face meet the requirements, the greeting action is triggered.



```
Button clicked: Start True
```

Finally click the Close_Camera button to close the camera.