Lesson 4 Athletics Performance

1. Getting Ready

Place the map on a flat and smooth surface and set the stair in the corresponding position on the map.

The specific placement method for the map and tool can refer to the video and file in “Athletics Sport Lessons/Lesson 1 Tools Assembly and Map Placement”.

2. Working Principle

As we all know, a robot is a device imitating human. In this lesson, we combine “Climb Stair” and “Hurdle” to simulate robots to learn human athletics sport.

The color is recognized through Lab color space firstly. Then convert RGB color into Lab color space, and proceed binarization, and dilation and erosion process to obtain the contour of the target color. Mark the target color with blue line, and obtain the coordinate parameter of the target to complete the color recognition.

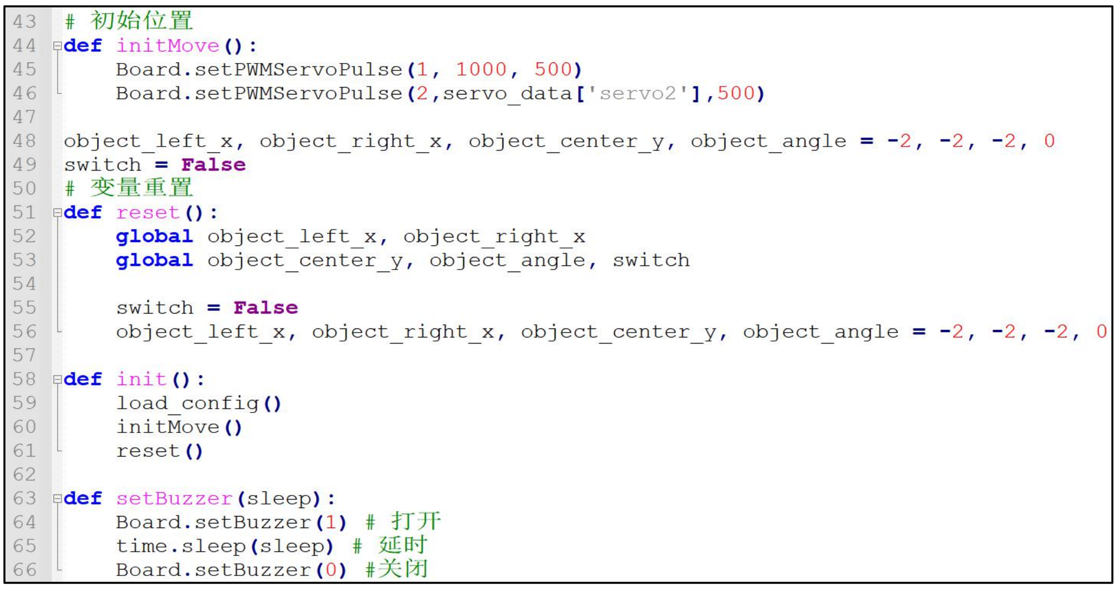
Then process the servo on the pan-tilt of the head after recognition, the x and y coordinates of the center point of the image are used as set values, and the currently acquired x and y coordinates are used as input values to update the pid.

Then calculate according to the feedback of the line position in the image, and finally make the robot move along the line through the change of position, so as to control the robot to walk along the black line.

Then, if you encounter step or hurdle in the middle, please refer to “Lesson 2 Climb step” and “Lesson 3 Hurdle”..

The source code of program is located in:

/home/pi/TonyPi/Extend/AthleticsPerform.py



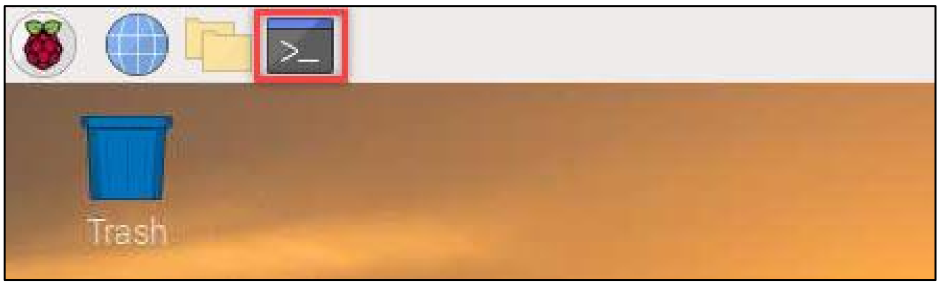
```
# 初始位置
def initMove():
    Board.setPWMServoPulse(1, 1000, 500)
    Board.setPWMServoPulse(2,servo_data['servo2'],500)

object_left_x, object_right_x, object_center_y, object_angle = -2, -2, -2, 0
switch = False
# 变量重置
def reset():
    global object_left_x, object_right_x
    global object_center_y, object_angle, switch

    switch = False
    object_left_x, object_right_x, object_center_y, object_angle = -2, -2, -2, 0

def init():
    load_config()
    initMove()
    reset()

def setBuzzer(sleep):
    Board.setBuzzer(1) # 打开
    time.sleep(sleep) # 延时
    Board.setBuzzer(0) #关闭
```

3. Operation Steps

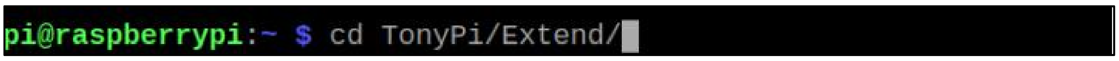
The entered command must pay attention to case sensitivity and space.

1) Turn on the robot and connect to Raspberry Pi desktop with VNC.

2) Click or press "Ctrl+Alt+T" to open LX terminal.



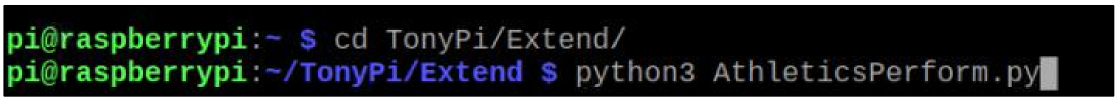
3) Enter "cd TonyPi/Extend/" and press "Enter" to come to the directory of game programmings.



```
pi@raspberrypi:~ $ cd TonyPi/Extend/
```

4) Enter "python3 Hurdles.py" command, and then press "Enter" to start the

game.



```
pi@raspberrypi:~ $ cd TonyPi/Extend/
pi@raspberrypi:~/TonyPi/Extend $ python3 AthleticsPerform.py
```

5) If want to exit the game, press “Ctrl+C” in the LX terminal. Please try multiple times if fail to exit.

4. Project Outcome

Note:

① In this game, the robot’s hands can not be replaced with novel robotic hands. Otherwise, it will change robot's center of mass and affect experience.

② The robot needs to be placed in a smooth and flat surface to avoid falling.

③ The background color does not be similar to red, black and blue to get better experience.

④ The height of the step and hurdle should be kept at 1cm, and the two ends of step need to be affixed with the red tape.

Place the map on a flat and smooth surface, and place step and hurdle on the corresponding position on the map. Then place the robot on the black line of map.

After starting this game, the robot will walk along the black line. When encounter a step, it will automatically go up and down step. When encounters a hurdle, it will go over it.

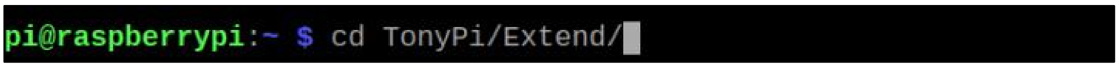
5. Function Extension

5.1 Modify Climbing and Hurdling Position

After the program is started, if the realization effect of going up and down step or hurdling is not good enough, we can change the distance where the robot starts performing action.

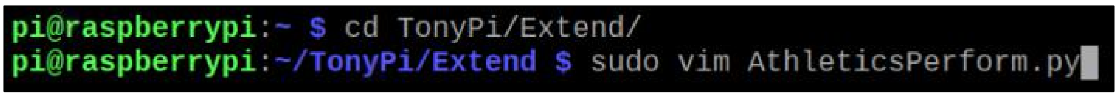
The program defaults that the robot starts going up and down step or hurdling at a distance of 1000 from the line. If the distance needs to be closer, the data should be decreased; if the distance needs to be farther, the data should be increased. This section takes robot starts climbing step or hurdling at a distance of 1020 as example. Please refer to the following steps：

1) Enter “cd TonyPi/Extend/” and press “Enter” to come to the directory of game programmings.



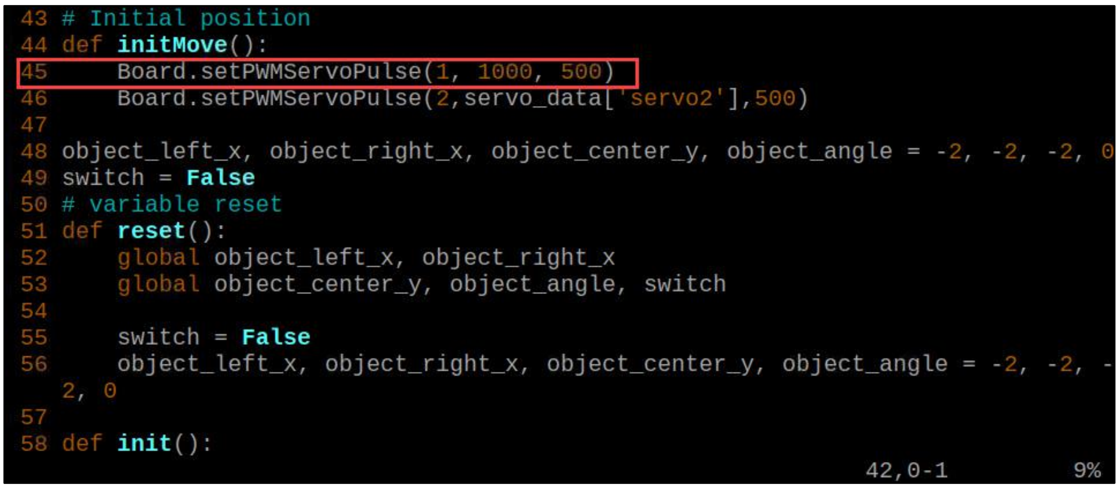
```
pi@raspberrypi:~ $ cd TonyPi/Extend/
```

2) Enter command “sudo vim AthleticsPerform.py” and press “Enter” to open the game programming file.



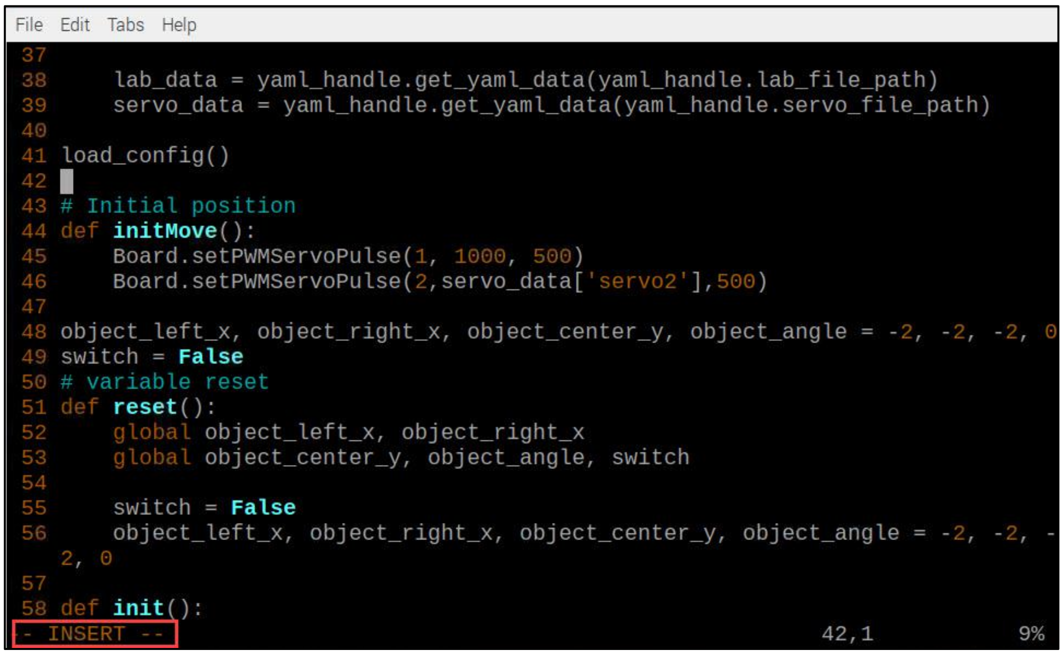
```
pi@raspberrypi:~ $ cd TonyPi/Extend/
pi@raspberrypi:~/TonyPi/Extend $ sudo vim AthleticsPerform.py
```

3) Find the code framed in the following figure:



```
# Initial position
def initMove():
    Board.setPWMServoPulse(1, 1000, 500)
    Board.setPWMServoPulse(2,servo_data['servo2'],500)

object_left_x, object_right_x, object_center_y, object_angle = -2, -2, -2, 0
switch = False
# variable reset
def reset():
    global object_left_x, object_right_x
    global object_center_y, object_angle, switch

    switch = False
    object_left_x, object_right_x, object_center_y, object_angle = -2, -2, -2, 0

def init():
                                                       42,0-1          9%
```

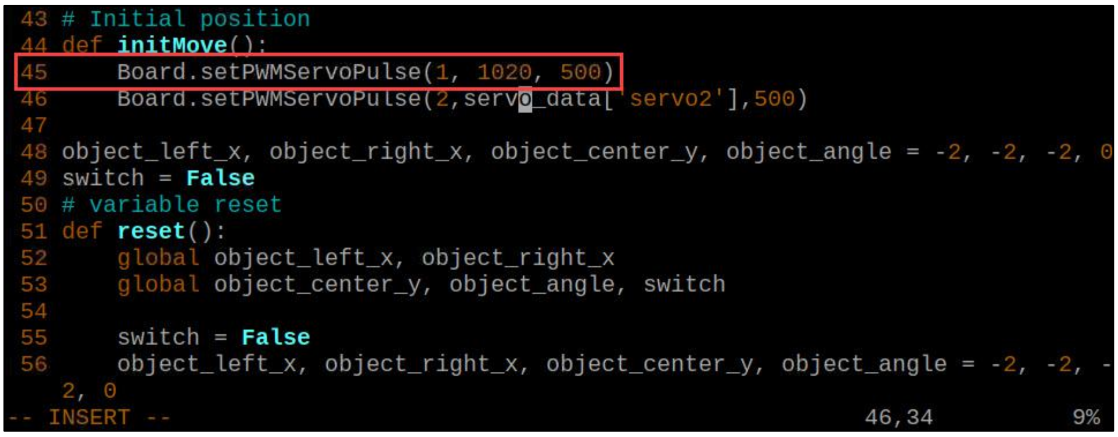
4) Press “i” on keyboard. When “Insert” appears in the lower left corner, which means it has entered the editing mode.



File Edit Tabs Help

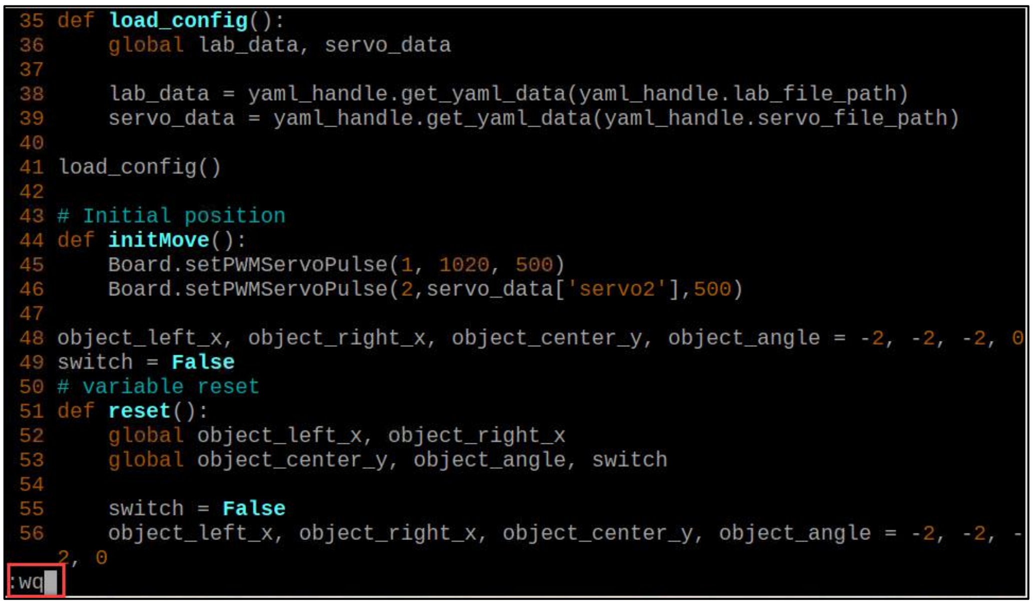
```
    lab_data = yaml_handle.get_yaml_data(yaml_handle.lab_file_path)
    servo_data = yaml_handle.get_yaml_data(yaml_handle.servo_file_path)

load_config()

# Initial position
def initMove():
    Board.setPWMServoPulse(1, 1000, 500)
    Board.setPWMServoPulse(2,servo_data['servo2'],500)

object_left_x, object_right_x, object_center_y, object_angle = -2, -2, -2, 0
switch = False
# variable reset
def reset():
    global object_left_x, object_right_x
    global object_center_y, object_angle, switch

    switch = False
    object_left_x, object_right_x, object_center_y, object_angle = -2, -2, -
2, 0

def init():
-- INSERT --                                                  42,1          9%
```

5) Modify “1000” in “Board.setPWMServoPulse(1, 1000, 500)” to “1020”, as the figure shown below:



```
# Initial position
def initMove():
    Board.setPWMServoPulse(1, 1020, 500)
    Board.setPWMServoPulse(2,servo_data['servo2'],500)

object_left_x, object_right_x, object_center_y, object_angle = -2, -2, -2, 0
switch = False
# variable reset
def reset():
    global object_left_x, object_right_x
    global object_center_y, object_angle, switch

    switch = False
    object_left_x, object_right_x, object_center_y, object_angle = -2, -2, -
2, 0
-- INSERT --                                                  46,34         9%
```

6) After modification, press “Esc” and then enter “:wq” (Please note that the colon is in front of wq). Then press “Enter” to save and exit the modified content.

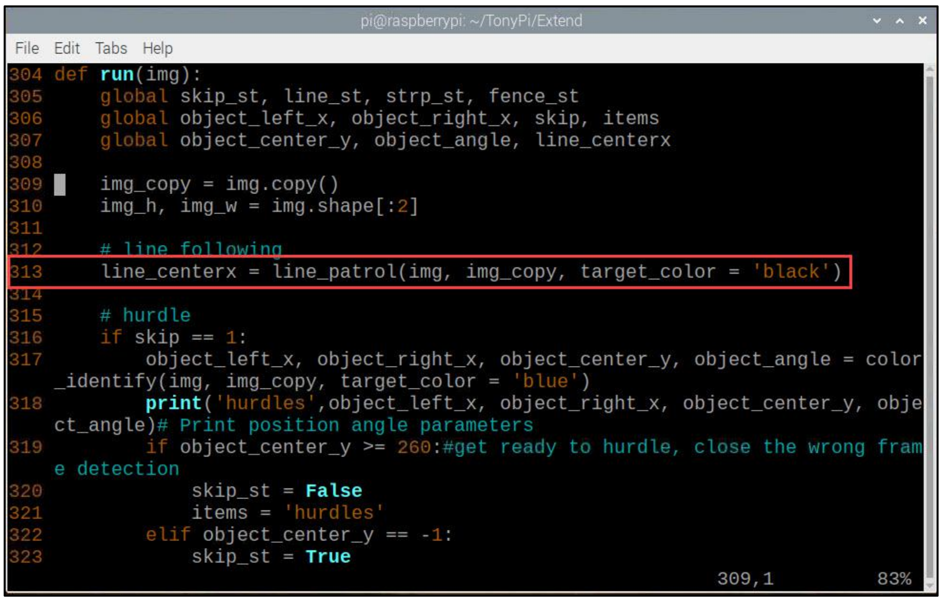


```
def load_config():
    global lab_data, servo_data

    lab_data = yaml_handle.get_yaml_data(yaml_handle.lab_file_path)
    servo_data = yaml_handle.get_yaml_data(yaml_handle.servo_file_path)

load_config()

# Initial position
def initMove():
    Board.setPWMServoPulse(1, 1020, 500)
    Board.setPWMServoPulse(2,servo_data['servo2'],500)

object_left_x, object_right_x, object_center_y, object_angle = -2, -2, -2, 0
switch = False
# variable reset
def reset():
    global object_left_x, object_right_x
    global object_center_y, object_angle, switch

    switch = False
    object_left_x, object_right_x, object_center_y, object_angle = -2, -2, -
2, 0
:wq
```

5.2 Modify Line Color

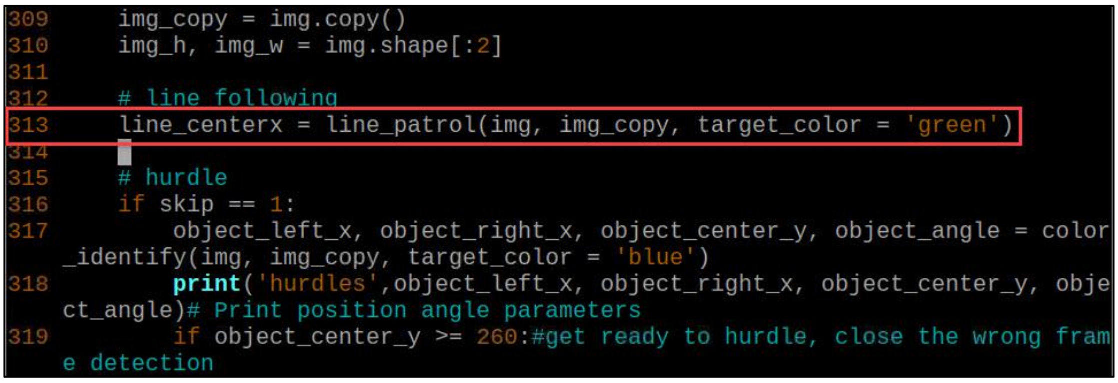
It is black line on the provided map. After starting program, the robot will walk along the black line. You can change the line with other colored tape, for example, green tape is used to affix on the map and the robot is set to walk along the green line. The specific steps are as follows:

1) Refer to steps (1), (2), (4) in "Modify Climbing and Hurdling Position" to enter the program editing interface and find the code framed in the figure below:



```
def run(img):
    global skip_st, line_st, strp_st, fence_st
    global object_left_x, object_right_x, skip, items
    global object_center_y, object_angle, line_centerx

    img_copy = img.copy()
    img_h, img_w = img.shape[:2]

    # line following
    line_centerx = line_patrol(img, img_copy, target_color = 'black')

    # hurdle
    if skip == 1:
        object_left_x, object_right_x, object_center_y, object_angle = color
_identify(img, img_copy, target_color = 'blue')
        print('hurdles',object_left_x, object_right_x, object_center_y, obje
ct_angle)# Print position angle parameters
        if object_center_y >= 260:#get ready to hurdle, close the wrong fram
e detection
            skip_st = False
            items = 'hurdles'
        elif object_center_y == -1:
            skip_st = True
                                                      309,1          83%
```

2) Press “i” to enter the editing mode. Modify “black” in line_centerx = line_patrol(img, img_copy, target_color = 'black') to “green”.



```
    img_copy = img.copy()
    img_h, img_w = img.shape[:2]

    # line following
    line_centerx = line_patrol(img, img_copy, target_color = 'green')

    # hurdle
    if skip == 1:
        object_left_x, object_right_x, object_center_y, object_angle = color_identify(img, img_copy, target_color = 'blue')
        print('hurdles',object_left_x, object_right_x, object_center_y, object_angle)# Print position angle parameters
        if object_center_y >= 260:#get ready to hurdle, close the wrong frame detection
```

3) After modification, refer to step (6) in “5.1 Modify Climbing and Hurdling Position” to save the modified code.