

QR code instructions

QR code instructions

[6.1 Introduction to gameplay](#)

[6.2 Core content analysis](#)

[6.3 How to play](#)

6.1 Introduction to gameplay

This course mainly uses the IMX219 camera to obtain the camera's image, identify the QR code information, and control the car's movement according to the QR code information.



6.2 Core content analysis

Import the QR code parsing library pyzbar

```
import pyzbar.pyzbar as pyzbar
from PIL import Image
```

If pyzbar is not installed on the system, open the terminal and run the following command to install it.

```
pip3 install pyzbar
sudo apt install libzbar-dev
```

Parse the grayscale image and extract the QR code information and image position in the image. If there is no QR code in the image, the information is None.

```

def detect_qrcode(image):
    # Convert to grayscale image
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    barcodes = pyzbar.decode(gray)
    for barcode in barcodes:
        # Extract the data of the QR code and the position of the bounding box The data
        # of the QR code and the position of the bounding box are extracted
        (x, y, w, h) = barcode.rect
        barcodeData = barcode.data.decode("utf-8")
        barcodeType = barcode.type
        # print("[INFO] Found {} barcode: {}".format(barcodeType, barcodeData))
        car_control(barcodeData)
    return barcodeData, (x, y, w, h)
    return None, (0, 0, 0, 0)

```

Control the car movement according to the string command of info. car_speed is the speed of the car, which can be adjusted according to actual needs.

```

car_speed = 0.3
def car_control(info):
    # print(info)
    global g_car_runtime
    if g_car_runtime >= 0:
        return
    g_car_runtime = 100
    if info == "forward": # forward
        g_car.set_car_run(1, car_speed*100)
    elif info == "back": # backward
        g_car.set_car_run(2, car_speed*100)
    elif info == "left": # left translation
        g_car.set_car_run(3, car_speed*100)
    elif info == "right": # right translation
        g_car.set_car_run(1, car_speed*100)
    elif info == "turnleft": # left rotation
        g_car.set_car_run(5, car_speed*100)
    elif info == "turnright": # Rotate right
        g_car.set_car_run(6, car_speed*100)
    elif info == "stop": # Stop
        g_car.set_car_run(7, 0)
    g_car_runtime = -1

```

Define a task_timeout task to make the car stop automatically, and do not execute other QR code instructions during the car's movement. g_car_runtime is in units of 10 milliseconds. When g_car_runtime is greater than 0, it will automatically decrease by 1 every 10 seconds, and the car will stop automatically when the timeout is reached.

```

g_car_runtime = -1

def task_timeout():
    global g_car_runtime
    while True:
        if g_car_runtime > 0:
            print("g_car_runtime:", g_car_runtime)
            g_car_runtime = g_car_runtime - 1
        elif g_car_runtime == 0:
            print("auto stop")
            g_car.set_car_run(7, 0)
            g_car_runtime = -1
            time.sleep(.01)

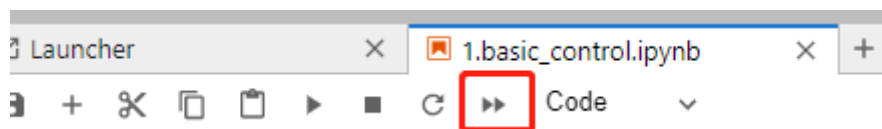
```

6.3 How to play

Open the jupyterLab client and find the code path:

```
/home/sunrise/sunriseRobot/Samples/2_AI/06_qrcode_movement/qrcode_movement.ipynb
```

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



Click the Start button below to start recognizing QR codes and execute commands. The QR code that can be recognized in the current routine is QRCode, and the information is "forward" for forward, "back" for backward, "left" for left translation, "right" for right translation, "turnleft" for left rotation, "turnright" for right rotation, and "stop" for stop.



Finally, click the Close_Camera button to close the camera.