

QR Code Detection and Generation in Robotics

This Jupyter Notebook will guide you through generating and detecting QR codes using Python and mapping them to robot actions.

Part 1: Understanding QR Codes

1. What is a QR Code?

Write a brief explanation of what a QR code is and how it is used.

2. Why are QR Codes Useful in Robotics?

List three ways QR codes can be used in robotics.

Part 2: Generating a QR Code

Step 1: Run the QR Code Generator Script

Use the following script to generate a QR code:

```
In [ ]: # Import necessary libraries
import time
import random

import rclpy
from qr_code_tools import generate_qr_code
from joystick_control import JoystickController # Ensure this file exists
from omni_robot_controller import OmniWheelControlNode # Ensure this matches
from image_capture import ImageCaptureNode # Import image capture controller
rclpy.init()
node = OmniWheelControlNode() # Initialize the robot control node
image_node = ImageCaptureNode()
joystick = JoystickController() # Initialize joystick control
```

Prompt user for text input

```
In [ ]: # Prompt user for text input
text = input("Enter text for QR code: ")
generate_qr_code(text, "my_qr.png")

print("QR code generated and saved as my_qr.png. Open it to scan with your p
```

Step 2: View the Generated QR Code

Locate the generated **my_qr.png** file. Open it and scan the QR code with your phone.

Question: What data does it show?

Part 3: Detecting a QR Code

Step 1: Run the QR Code Detector

Use the following script to detect QR codes. Then turn the Mentorpi camera towards the qr code.

```
In [ ]: # Import the QR code detection module
from qr_code_tools import generate_qr_code
import rclpy

# Initialize ROS2 node
rclpy.init()
detector = QRCodeDetector()

# Run the QR code detection process
rclpy.spin_once(detector)
detected_qr = detector.get_detected_qr_code()

print("Detected QR Code:", detected_qr)
```

Step 2: Mapping QR Code Detection to a Joystick Button

Modify the joystick commands to add QR detection to the 'X' button.

```
In [ ]: # Function to detect QR code when pressing "X"
def detect_qr_code():
    rclpy.spin_once(detector)
    detected_qr = detector.get_detected_qr_code()
    print("Detected QR Code:", detected_qr)

# Map joystick button "X" to QR code detection
joystick.map_button("X", detect_qr_code)

print("Button mapping for QR detection set.")
```

Part 4: Applying QR Codes in a Robotics Scenario

Scenario:

Challenge 1- Create a function Access Control Verification

- A security robot scans a QR code at a restricted area.
- **If access is granted(qr code detected):**
 - The **RGB LED turns green**.
 - The **buzzer beeps once**.
- **If access is denied:**
 - The **RGB LED turns red**.
 - The **buzzer emits three short beeps** as a warning.

```
In [ ]: # Function to execute robot movement based on detected QR code
def qr_code_action():
    rclpy.spin_once(detector)
    detected_qr = detector.get_detected_qr_code()

    # now write your code logic here

# Map joystick button "X" to QR code detection and movement
joystick.map_button("x", qr_code_action)
```

Challenge 2- Create a function Package Sorting Confirmation

- A warehouse robot scans a QR code on a package.
- **If the package is correctly identified:**
 - The **RGB LED blinks blue**.
 - The **buzzer beeps twice**.
- **If an incorrect package is detected:**
 - The **RGB LED flashes red**.
 - The **buzzer emits a long error beep**.

```
In [ ]: def qr_code_challenge2():
    rclpy.spin_once(detector)
    detected_qr = detector.get_detected_qr_code()
    # now write your code logic here
joystick.map_button("y", qr_code_action)
```

Challenge 3- Create a function- Medical Robot Alert

- A hospital robot scans a QR code for patient room verification.
- **If the code matches the assigned room:**
 - The **RGB LED** turns green.
 - The **buzzer emits a soft chime**.
- **If the code is incorrect:**
 - The **RGB LED** turns red.
 - The **buzzer beeps loudly** to alert staff.

```
In [ ]: def qr_code_challenge3():
        rclpy.spin_once(detector)
        detected_qr = detector.get_detected_qr_code()
        # now write your code logic here
        joystick.map_button("square", qr_code_action)
```

```
In [ ]: print("Button mappings set.")

        print("Listening for button presses... Press Ctrl+C to stop.")
        try:
            joystick.listen() # This function should listen for button presses and
        except KeyboardInterrupt:
            print("Joystick listening stopped.")
```

Part 5: Reflection

What challenges did you face while implementing QR code detection and generation?

How can QR codes enhance robotics in real-world applications?

If you could improve this system, what features would you add?