Mapping PS2 Controller Buttons to Movement Functions

This Jupyter Notebook guides you through mapping PS2 controller buttons to control movement functions on a robot.

Step 1: Setting Up the Environment

Ensure your PS2 controller is connected and necessary libraries are installed.

Run the following command in the terminal to start Jupyter Notebook:

~~~~
cd RPI-Demos
code```

Now, proceed with the notebook to test the movement mappings.
~~~~

Step 2: Import Required Libraries

First, import the necessary modules for joystick control and movement functions.

In [ ]:

```python
# Import necessary libraries
import time
import random


import rclpy
from joystick_control import JoystickController  # Ensure this file exists a
from omni_robot_controller import OmniWheelControlNode  # Ensure this matche
from image_capture import ImageCaptureNode

# Initialize joystick, LED, and buzzer controllers
rclpy.init()
movement = OmniWheelControlNode()  # Initialize the robot control node
image_node = ImageCaptureNode()
joystick = JoystickController()  # Initialize joystick control

print("Joystick and Movement Controller initialized.")
```

Step 3: Define Movement Functions

These functions will be mapped to the PS2 controller buttons.

Movement Commands

The robot supports omni-directional movement using the following commands:

Forward Movement

Moves the robot forward at a specified speed for a given duration.

```
move_forward(0.5, 3.0)  # Moves forward at 0.5 m/s for 3 seconds
```

Backward Movement

Moves the robot backward.

```
move_backward(0.5, 3.0)  # Moves backward at 0.5 m/s for 3 seconds
```

Left Movement

Moves the robot to the left.

```
move_left(0.5, 2.0)  # Moves left at 0.5 m/s for 2 seconds
```

Right Movement

Moves the robot to the right.

```
move_right(0.5, 2.0)  # Moves right at 0.5 m/s for 2 seconds
```

Diagonal Movement

Moves the robot diagonally at a specified angle.

```
move_in_direction(45, 0.5, 2.0)  # Moves at a 45° angle at 0.5 m/s for 2 seconds
```

Rotating Left (Counterclockwise)

Rotates the robot to the left (CCW).

```
rotate_left(0.5, 2.0)  # Rotates CCW at 0.5 RPS for 2 seconds
```

Rotating Right (Clockwise)

Rotates the robot to the right (CW).

```
rotate_right(0.5, 2.0)  # Rotates CW at 0.5 RPS for 2 seconds
```

Stopping All Motors

Immediately stops all motors.

```
stop_all_motors()  # Stops all motor movement
```

In [ ]:

```
# Function to move forward
def move_forward():
    print("Moving forward")

# Function to move backward
def move_backward():
    print("Moving backward")

# Function to move left
def move_left():
    print("Moving left")

# Function to move right
def move_right():
    print("Moving right")

# Function to rotate left
def rotate_left():
    print("Rotating left")

# Function to rotate right
def rotate_right():
    print("Rotating right")

# Function to stop all movement
def stop_robot():
    print("Robot stopped")

# Function to move faster
def move_fast():
    print("Moving forward quickly")

# Function to move diagonally
def move_diagonal():
    print("Moving diagonally")
```

Step 4: Map Buttons to Functions

Use the `map_button` method to associate buttons with specific movement functions.

In [ ]:

```
# Map controller buttons to movement functions
joystick.map_button("cross", move_forward)
joystick.map_button("triangle", move_backward)
joystick.map_button("square", move_left)
joystick.map_button("circle", move_right)
joystick.map_button("l1", rotate_left)
joystick.map_button("r1", rotate_right)
joystick.map_button("start", stop_robot)
joystick.map_button("l3", move_diagonal)

print("Button mappings set.")
```

Step 5: Run the Joystick Event Loop

Start listening for button presses and trigger the corresponding movement functions.

In [ ]:

```
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.")
```

Step 6: Testing and Debugging

Press the following buttons to test the mappings:

- **Cross** → Robot moves forward.
- **Triangle** → Robot moves backward.
- **Square** → Robot moves left.
- **Circle** → Robot moves right.
- **L1** → Robot rotates left.
- **R1** → Robot rotates right.
- **Start** → Robot stops all movement.
- **L3** → Robot moves diagonally.

If anything doesn't work, check for errors and restart the script.