Virtual Robot Task: Task01 - Move the Robot Forward

Objective

In this task, you will write code to move a virtual robot forward. The code will interact with the black box API that controls the robot's motor.

Understanding the `move_forward` Function

The `move_forward` function is used to send a command to the robot's motor to move forward. Here's how you can start writing it:

- The function takes one parameter: `speed`, which represents the speed at which the robot should move.
- The `speed` parameter is of type `u8`, which is an 8-bit unsigned integer, meaning it can have values from 0 to 255.

Coding Task

1. **Navigate to the virtual robot project**

   - In GitHub Codespaces, open the terminal (at the bottom of the screen).
   - Navigate to the `virtual-robot-maze/src` directory:

   ```
   cd ~/workspaces/virtual-robot-maze/src
   ```

2. **Edit the `main.rs` file**

   - Open the `main.rs` file located in the `src` directory.
   - Add the following function template:

   ```
   pub fn move_forward(speed: u8) {
       // Code to send forward command to the black box API
   }
   ```

   - Replace the comment with logic to interact with the black box API, using the `speed` parameter.

3. **Call the `move_forward` function**

   - In the `main` function of `main.rs`, call the `move_forward` function:

     ```
     fn main() {
         move_forward(5); // Example call to move the robot forward at speed 5
     }
     ```

   - This will send a command to move the robot forward at a speed of 5.

4. **Run the code**

   - In the terminal, run:

     ```
     cargo run
     ```

   - If implemented correctly, you should see output indicating that the command was sent to the virtual robot:

     ```
     Sending command: MOVE_FORWARD with speed 5
     ```

Switching Between Projects in GitHub Codespaces

As you progress, you will work on different Rust projects within the same repository. Here's how to switch between them:

1. **Switching to the Basic Programming Tasks**

   - If you are currently working in the `virtual-robot-maze/src` folder, but need to return to the basic programming tasks, navigate to the `basic-programming` directory:

     ```
     cd ~/workspaces/virtual-robot-maze/basic-programming/basic_syntax
     ```

   - Use `cargo run` to run the code within the `basic_syntax` project.

2. **Switching to the Virtual Robot Project (Task01)**

   - To switch back to the virtual robot task, navigate to the `virtual-robot-maze/src` directory:

     ```
     cd ~/workspaces/virtual-robot-maze/src
     ```

   - Use `cargo run` to execute the virtual robot code after making changes.

What to Expect

- After running the program, the output should confirm that the robot received a command to move forward at the specified speed.
- If there are errors, read the error messages carefully to understand what needs to be fixed.

Hints

- Make sure you're in the correct project directory before running any commands.
- Use the correct data type for the `speed` parameter (`u8`).
- If you're unsure how to interact with the black box API, refer to the provided API documentation.

Next Steps

Once you complete Task01, you will understand how to interact with the virtual robot's API to control its movements. In the next tasks, you will learn how to add more complex behaviors to the robot, such as turning and navigating through the maze.