Test Plan for Moving2 Project

1. Test Objectives

The purpose of this test plan is to ensure that the Moving2 robot `Wall-k` can move autonomously using Reinforcement Learning (RL). Specific objectives include validating the hardware and software components, as well as the RL algorithms.

2. Test Scope

2.1 Hardware Components

- Verification of the robot's physical components (motors, sensors, controllers).
- Testing hardware integration and interactions with the software.

2.2 Software Components

- Testing the RL code, including `main.py`, `env.py`, and `test.py`.
- Ensuring the functionality of the RL environment and the accuracy of the Q-table.

2.3 Reinforcement Learning

- Validating the RL model's learning process.
- Ensuring that the robot can achieve the desired autonomous movement through learned behavior.

3. Test Strategy

3.1 Unit Testing

- **RL Code**: Test individual functions and modules within the RL scripts (`main.py`, `env.py`, `test.py`).
- **Hardware Interaction Code**: Test code that interacts directly with the robot's hardware components(`drive_all_test.cpp`,`agent_rpi.cpp`).

3.2 Integration Testing

- **Hardware and Software Integration**: Ensure that the hardware components work seamlessly with the software. Ensure that the data collected is usable by the software.
- **RL Model and Environment**: Test the integration of the RL model with its environment.

3.3 System Testing

- Test the entire system to ensure that the robot can perform autonomous movement as expected.
- Perform end-to-end tests covering all functionalities.

3.4 Performance Testing

- Evaluate the performance of the RL algorithms in terms of speed and accuracy.
- Assess the robot's response time and precision in executing movements.

4. Acceptance Criteria

- **Hardware**: All physical components should function correctly without errors.
- **Software**: No critical bugs in the RL scripts, and all unit tests should pass.
- **RL Model**: The robot should demonstrate improved movement through learned behavior with a success rate of at least 90%.

5. Test Methodology

5.1 Test Environment

- Set up the robot in a controlled environment with predefined obstacles and goals.
- Use the project structure and CI pipeline for testing and validation.

5.2 Test Procedures

- **Hardware Tests**: Check connections, calibrate sensors, and verify motor functionality.
- **Software Tests**: Run automated tests using `test.py` and ensure code quality with Black.
- **RL Tests**: Train the RL model, observe its behavior, and validate its learning outcomes.

6. Resources Needed

- **Hardware**: Robot components (motors, sensors, controllers).
- **Software**: Python environment, RL libraries, testing frameworks.
- **Personnel**: Developers, testers, and project managers.
- **Time**: Estimated 2 weeks for comprehensive testing.

7. Test Schedule

Task	Duration	Responsible
Hardware Setup	2 days	Hardware Team
Software Testing	3 days	Software Team
RL Training & Testing	5 days	RL Team
Integration Testing	2 days	Integration Manager
Performance Testing	2 days	Quality Manager
Final Review	1 day	Project Manager

8. Risks and Contingencies

- **Risk**: Hardware malfunctions.
  - **Mitigation**: Have spare parts and tools for quick repairs.
- **Risk**: Software bugs.
  - **Mitigation**: Regular code reviews and automated testing.

9. Validation and Reporting

- Document all test results, including any bugs found and fixed.
- Provide detailed test reports to stakeholders.
- Validate the final system performance against the acceptance criteria.