

Robot Simulator – Documentation Report

Project Description

The Robot Simulator project is a Python-based interactive simulation that allows users to control a robot on a 2D grid. It supports:

- Basic directional movement.
- Diagonal movement.
- Obstacle detection and handling.
- Battery simulation.
- Grid expansion.
- Grid visualization.
- Command history.

This simulator is ideal for learning object-oriented programming (OOP), custom exception handling, and building stateful command-driven systems.

Summary of the Approach

The simulator was implemented using Python OOP design, centering around a class called `RobotSimulator`. The approach includes:

- Handling robot state (position, direction, battery).
- Using enumerations for directions (Enum).
- Command parsing and execution (`execute_command` method).
- Grid boundary checks and auto-expansion logic.
- Obstacle management via a set of coordinates.
- Error handling using custom exception classes.
- Visual feedback through command-line grid rendering.

Two running modes are provided:

- Demo Mode: Runs predefined commands.
- Interactive Mode: Accepts real-time user input via terminal.

Lessons Learned

- The importance of modular design for easier testing and debugging.
- How to use custom exceptions to manage logical flow.
- The challenge of simulating a real-time robot-like behavior.
- How simple visual feedback (like ASCII grid rendering) can significantly improve usability.
- Managing edge cases, like moving out of bounds or battery depletion, requires robust validation logic.