Project Description: Multi-Robot Sequential Manipulation Simulation

# 1. Project Title

Simulation of Sequential Object Manipulation by Two Robots in MuJoCo

# 2. Objective

The objective of this project is to simulate a collaborative manipulation task between two robotic arms within a MuJoCo-based physics environment. The task involves sequential actions: one robot detects and relocates an object to a predefined platform, while a second robot retrieves the object from the platform and inserts it into a specific hole within its own workspace.

# 3. Simulation Scope

- Environment: MuJoCo (Multi-Joint dynamics with Contact) physics engine.  
- Robots:  
 - Robot A: Positioned on the left, responsible for object detection, grasping, and placement on a platform at a specific height.  
 - Robot B: Positioned on the right, responsible for retrieving the object from the platform and inserting it into a hole.  
- Object: A small movable cube that can be grasped and transported.  
- Platform: Elevated flat surface placed at an intermediate position between both robots.  
- Hole: Destination zone within Robot B’s workspace designed to accept the object.

# 4. Functionality

- Multi-Agent Coordination:  
 - Robot A and B operate in a strictly sequential order.  
 - Robot A places the object and only after completion does Robot B begin its task.  
- Kinematics:  
 - Each robot has at least two degrees of freedom (hinge joints) for movement and basic reachability.  
- Object Physics:  
 - The object is defined with a free joint to simulate translation and rotation from both robot interactions.  
- Platform Physics:  
 - Acts as an intermediate resting position with a fixed elevation.

# 5. Simulation Components

- MuJoCo XML Model (two\_robot\_scene.xml):  
 - Defines geometry, joint structures, body hierarchy, and actuator mappings.  
- Python Control Script (simulate\_two\_robots.py):  
 - Drives simulation via simple scripted logic using the mujoco Python API.  
 - Synchronizes robot behavior in phases to reflect sequential task logic.

# 6. Use Cases

- Robotics research: Multi-agent coordination and sequential task planning.  
- Reinforcement learning: Can be extended for agent training.  
- Simulation-based control development for pick-and-place tasks.  
- Human-in-the-loop robotics (with future integration of feedback).

# 7. Potential Extensions

- Upgrade robot models to real-world arms (e.g., UR5, KUKA).  
- Add sensors (e.g., simulated cameras, touch/contact).  
- Implement full inverse kinematics or trajectory planning.  
- Introduce reinforcement learning to allow robots to learn the sequence.  
- Add human feedback loops using frameworks like gym-hil.

# 8. Requirements

- Software:  
 - MuJoCo (v2.3+)  
 - Python 3.8+  
 - mujoco, numpy  
- Hardware:  
 - GPU recommended for visualization (or use software renderer like EGL).  
- OS Compatibility:  
 - Linux, macOS, Windows

# 9. Author / Maintainer

- Project Lead: Isaac Mendez (Simulation Engineer / Robotics Developer)  
- Robotics Developers: Ezequiel Ramirez, Oscar  
- Simulation Stack: MuJoCo + Python  
- Platform: Debian 12 Workstation

10. Github repo:

LeRobotWorldwideHackathonSV-team01-pr01

📅 Project Created on: 2025-06-14 11:42:45 CST