# **Mini Project 0: Simulator Setup** Due: Wednesday, September 3 11:59pm EST

The objective of MP 0 is to setup the simulator environment for the robotic manipulator. In this course, we will use [MuJoCo](https://mujoco.org/) to simulate the [AgileX Piper](https://global.agilex.ai/products/piper) robotic arm.

1. Linux Environment Setup

The majority of today’s robotics research and industry development happens in Linux (especially Ubuntu) environments. Some of the reasons for this include but not are not limited to:

* Strong support for open-source tools and libraries.
* Easier access to low-level hardware, e.g, serial ports, CAN buses, and custom drivers.
* Large developer and research community
* Frameworks such as ROS (Robot Operating System), Gazebo, MoveIt, and many hardware drivers are designed primarily for Linux.

Accordingly, to become a competent roboticist, you need to get familiar with working in a Linux environment. The hardware drivers and SDK (Software Development Kit) for AgileX Piper robotic arm that we will utilize in the lab are also designed for Linux. Therefore, the desktop computer connected to the robotic arm at each station runs Linux (Ubuntu). To minimize discrepancies between working in a simulator environment and at a real robot arm station, we strongly encourage you to setup a Linux environment on your computer.

If you currently have not setup a Linux environment on your computer, please choose from the two options below:

* Windows Users:
  1. [Dual Boot](https://ubuntu.com/tutorials/install-ubuntu-desktop#1-overview)
  2. [WSL (Windows Subsystem for Linux)](https://documentation.ubuntu.com/wsl/latest/howto/install-ubuntu-wsl2/)
  3. VM (Virtual Machine) – [Virtual Box](https://linuxhint.com/install_ubuntu_virtualbox_2004/)  
     Note: Running MuJoCo on a VM can be very slow
* Mac Users:  
  Currently, there is no consistent and stable way to install Linux on a Mac. Ubuntu does not officially support native install for Macs with Apple M-series chips (ARM64 architecture). Users of older Macs with Intel chips (x86\_64 architecture) can setup dual boot, but this may be somewhat difficult.
  1. [Dual Boot](https://linuxnewbieguide.org/how-to-install-linux-on-a-macintosh-computer/) (Intel Macs only)
  2. Keep using Mac (may face complications in the future)
  3. VM (Virtual Machine) – [Virtual Box](https://linuxhint.com/install_ubuntu_virtualbox_2004/)  
     Note: Running MuJoCo on a VM can be very slow

1. Docker Installation

Note: Mac users can skip this step. They can still install Docker on their machine, but they will not be able to see GUI from the container. Please follow instructions at the end of Section C. for local MuJoCo installation.

In this class, we will also heavily rely on Docker. Docker is a software platform that lets you package an application and all of its dependencies (OS, libraries, configuration, environment, etc) in a lightweight unit called a container. A container is just like a complete snapshot of a working instance of an application. The main advantages of using Docker are:

* Containerized software runs consistently across different environments, avoiding “it works on my machine” problem. If a container runs on my machine, it will run on your machine as well, if Docker is installed.
* Containers are lightweight. Unlike a virtual machine, containers share the host machine’s operating system kernel. Therefore, they are much lighter and faster to run.

Note: Docker is an instance of a container engine / runtime. There are alternatives including Podman and containerd, but Docker is the most popular one.

Note: The instructions below are for a Linux (Ubuntu) system.

Installing Docker:

# Docker installation using the convenience script

curl -fsSL https://get.docker.com -o get-docker.sh

sudo sh get-docker.sh

# Add user to the docker group

sudo usermod -aG docker $USER

Note: This allows you to run docker command without ‘sudo.’ However, this will not take effect until you restart.

# Verify Docker

docker run hello-world

1. MuJoCo Installation:

In this course, we provide a preconfigured docker container installed with MuJoCo 3.3.5.

# Clone

git clone <https://github.com/skim743/CS4803_ARM.git>

# Change the directory

cd CS4803\_ARM/docker

# Important: You will not see GUI if you don’t run this command before running the shell script.

xhost +

# Run the convenience script to run the container

./docker\_run.sh

# Test run MuJoCo. This will load an AgileX Piper arm and an optical breadboard. You can control each joint by clicking/dragging each bar under the ‘Control’ tab on the top right.

python3 -m mujoco.viewer -mjcf=./assets/scene.xml

Local MuJoCo installation for Mac users:

# Install dependencies. Please make sure you have Python 3.10

brew install python glfw

# Install MuJoCo

pip3 install mujoco

# Clone

git clone <https://github.com/skim743/CS4803_ARM.git>

# Change the directory

cd CS4803\_ARM

# Test run MuJoCo. This will load an AgileX Piper arm and an optical breadboard. You can control each joint by clicking/dragging each bar under the ‘Control’ tab on the top right.

python3 -m mujoco.viewer -mjcf=./assets/scene.xml

# Run a Python test file

cd <path-to-your-directory>/CS4803ARM\_Fall2025/user\_data

mjpython test.py