I’m Writing This Because Every Time I Install MuJoCo, I Face the Same Issues...

Every. Single. Time. So here’s my attempt to save someone (maybe even my future self) from the headache of getting MuJoCo up and running on Ubuntu 18.04 and 20.04. Whether you’re new to MuJoCo or you’ve been battling it for a while, this guide should help you dodge the common pitfalls. I’m even throwing in some tips for installing it with Conda because that’s where things often go sideways!

Step-by-Step Installation Guide for Ubuntu

Let’s get MuJoCo up and running on a regular Ubuntu setup. Then, we’ll dive into the Conda madness.

Step 1: Download the MuJoCo Software

1. Head to MuJoCo's website.
2. Grab the latest `mujoco210` package. You should get a `.tar.gz` file.
3. Unzip it to your home directory:

   ```bash unzip mujoco210-linux-x86_64.tar.gz -d ~/.mujoco ```

4. Now you should have MuJoCo sitting comfortably in `~/.mujoco/mujoco210`.

Step 2: Get the License Key (If Needed)

Older versions need a key, but the newer ones are free! If you need a key (`mjkey.txt`), place it in `~/.mujoco`.

Step 3: Install Dependencies

We need some dependencies to make sure MuJoCo plays nice with Ubuntu:

For **Ubuntu 18.04**: ```bash sudo apt update sudo apt install libosmesa6-dev libgl1-mesa-glx libglfw3 ```

For **Ubuntu 20.04**: ```bash sudo apt update sudo apt install libosmesa6-dev libgl1-mesa-glx libglfw3 patchelf ```

Step 4: Set Environment Variables

Open up your `.bashrc` file and add these environment variables: ```bash nano ~/.bashrc ```

Add: ```bash export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:~/.mujoco/mujoco210/bin export MUJOCO_PY_MUJOCO_PATH=~/.mujoco/mujoco210 export MUJOCO_PY_MJKEY_PATH=~/.mujoco/mjkey.txt ```

Update the terminal: ```bash source ~/.bashrc ```

Step 5: Install `mujoco-py` with pip

If you want to use Python, you’ll need `mujoco-py`: ```bash pip install mujoco-py ```

Step 6: Test It!

Run a quick Python test to make sure it's working: ```python python -c "import mujoco_py" ```

MuJoCo Installation with Conda

Now, let's get into the fun stuff: setting up MuJoCo with Conda. If you prefer using virtual environments (which you should), Conda is a solid choice, but it does come with its quirks.

Step 1: Create a Conda Environment

Create a new Conda environment for your project: ```bash conda create -n mujoco_env python=3.8 conda activate mujoco_env ```

Step 2: Install Dependencies via Conda

Conda makes dependency management a bit easier, so let's use it: ```bash conda install -c conda-forge glfw glew ```

Note: If you're on Ubuntu 20.04, you might still need `libosmesa6-dev`: ```bash sudo apt install libosmesa6-dev patchelf ```

Step 3: Download & Setup MuJoCo

Do the usual MuJoCo download: ```bash wget https://mujoco.org/download/mujoco210-linux-x86_64.tar.gz mkdir ~/.mujoco && tar -xvzf mujoco210-linux-x86_64.tar.gz -C ~/.mujoco ```

Step 4: Set Environment Variables for Conda

Add MuJoCo to the Conda environment: ```bash export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:~/.mujoco/mujoco210/bin ```

Or, if you want it to persist in your Conda environment, create an `activate.d` script: ```bash mkdir -p ~/anaconda3/envs/mujoco_env/etc/conda/activate.d nano ~/anaconda3/envs/mujoco_env/etc/conda/activate.d/env_vars.sh ```

Add: ```bash export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:~/.mujoco/mujoco210/bin export MUJOCO_PY_MUJOCO_PATH=~/.mujoco/mujoco210 export MUJOCO_PY_MJKEY_PATH=~/.mujoco/mjkey.txt ```

Step 5: Install `mujoco-py` in Conda

Use pip to install `mujoco-py` in the Conda environment: ```bash pip install mujoco-py ```

Step 6: Test It in the Conda Environment

Do the same quick test: ```python python -c "import mujoco_py" ```

Common Issues and Debugging (Conda & Native)

Here's a refresher on some common headaches and their fixes.

1. Conda Environment Not Recognizing MuJoCo

Problem: MuJoCo binaries aren't getting picked up in your Conda environment.

Solution: Make sure your `LD_LIBRARY_PATH` is set properly in the environment variables. Verify with: ```bash echo $LD_LIBRARY_PATH ``` And ensure it includes the MuJoCo path.

2. Issues with OpenGL or GLFW

Problem: If MuJoCo crashes with rendering errors in Conda, you probably have an OpenGL or GLFW problem.

Solution: Use Conda to manage GLFW: ```bash conda install -c conda-forge glfw ``` And make sure your system has the right graphics drivers.

3. Segmentation Faults Inside Conda

Problem: You get mysterious segmentation faults when trying to run simulations in Conda.

Solution: Check if you need to install additional dependencies via apt: ```bash sudo apt install libosmesa6-dev patchelf ``` And always make sure your Conda environment's Python version matches what `mujoco-py` likes (Python 3.6 to 3.8 is usually safe).

Wrapping It Up

MuJoCo can be finicky, but once you've got it going, it's a powerhouse for simulations. Whether you're running native or in a Conda environment, I hope this guide helps you skip the drama and get straight to coding those fancy RL models.

Good luck, happy simulating, and may your virtual robots stay upright! 🤖