The interactive simulations will be executed in the open-source MuJoCo physics engine. To define the behaviour of the simulated systems, we'll use the Python API of MuJoCo. Please follow these steps to ensure you are able to run and edit scripts during the summer school.

Setting up MuJoCo for forward dynamics

If you are already familiar with package/environment managers, you can use the manager of your choice.

If you don't have a package manager yet, we recommend getting Miniconda. A package manager will help us collect all the code we need (e.g. the physics engine) in one place.

Test your installation by opening up a terminal / command window and typing the command:

```
conda --version
```

You should see a response like "conda 22.11.1" (or your version of conda).

Download and extract the files from the following repository: Balint-H/ssnr_sim (github.com)

You can use Git to clone it, or click on the <> Code ▾ button on the Github page to download it.

Navigate to the extracted folder with a terminal. Create a new environment using the environment file in the folder "ssnr_environment.yml". If you are using conda, the command is:

```
conda env create -f ssnr_environment.yml
```

Activate the created environment:

```
conda activate ssnr_sim
```

From now on any installs made with this terminal will only affect this environment.

Test your installation. We'll run a python script that just opens a scene in MuJoCo's interactive viewer. If you see a human shape relaxing in a hammock, then you are ready for the workshop!

```
python welcome_scene/hello_ssnr.py
```

(Recommended) Install an Integrated Development Environment to efficiently edit and debug your scripts. We recommend getting Spyder or PyCharm. To install Spyder you just need to run the following command from your activated environment:

```
conda install spyder
```

(Suggested self guided study) Open the ".xml" files included in the xml folder with a code editor. You can use your IDE to do so. Inside you'll find an annotated MuJoCo scene, breaking down individual elements simulated, and making suggestions of things to consider to try out and explore. You can edit the "load_model.py" file to run the other scenes, or drag-and-drop the xml on to an already open MuJoCo viewer. If interested, looking at the official introductory MuJoCo tutorials is a good way to dig deeper before the workshop.

If you have further questions about the workshop and its contents, feel free to contact me at:

bkh16@ic.ac.uk

Balint Hodossy