1 Docker setup

There are four ways to get the prerequisites for the assignment. In order of **decreasing** difficulty:

1. You're an expert hacker. Get on an Ubuntu machine and install ROS Kinetic, numpy, skimage, scipy, Pillow, cython, h5py, TensorFlow>=1.3.0, Keras>=2.0.8. Note you'll also have to set up a ROS workspace as in previous assignments, and possibly *source /opt/ros/melodic/setup.bash* a bunch of times. The docker instances has that in *.bashrc*. **We strongly recommend using docker.**

2. You're a lazy hacker: we provide a Docker image for you at `https://hub.docker.com/r/gtmobilemanipulation/mm8803-fetch-gazebo` which you can pull to your machine. It is assumed you know Docker and how to get the image running on your machine, with a GUI.

3. You prefer not to learn the ins/outs of Docker: we provide some handy dandy shell scripts that pull the Docker image and start/stop it etc. To get it, clone them from github.gatech.edu:
   `$ git clone https://github.gatech.edu/gtmobilemanipulation/mm8803-fetch-gazebo.git`

If you use our docker image on your own machine, you will need to create a directory ~/catkin_ws before continuing:

```
$ mkdir ~/catkin_ws
```

Easy Docker Scripts Overview

Here is an explanation of the scripts we provide, in the order you'll probably use them:

- *pull.sh* : This file is for pulling image from *Dockerhub*, it is the first step you should do if you don't have this Docker image locally on your computer. To use it,

  ```
  $ ./pull.sh
  ```

- *run.sh* : This file is for creating a Docker container with exactly one parameter as the name of your container. After this, let us also allow X-window server connections from the container:

  ```
  $ ./run.sh YOUR_USERNAME
  $ xhost + 127.0.0.1
  ```

- *connect.sh* : This file is for running the Docker container with one parameter, which is your container name specified in previous step. The color of the characters will change from green to grey on our lab computers.

  ```
  $ ./connect.sh YOUR_USERNAME
  ```

- *save.sh* : This is for saving your work (file changes or setting changes) in your container.

  ```
  $ ./save.sh YOUR_USERNAME
  ```

- *stop.sh* : This is the last step to fully stop the container and remove it (but if you saved it you will not lose data).

  ```
  $ ./stop.sh YOUR_USERNAME
  ```

- *restart.sh* : This is for retrieving your previous work in the saved image,

  ```
  $ ./restart.sh YOUR_USERNAME
  ```

Please note that all above commands should be executed on the host computer, not in the container.