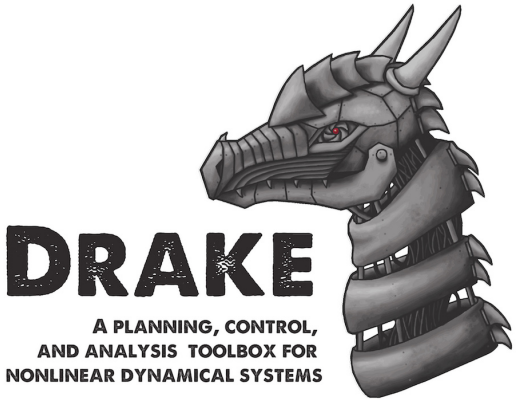
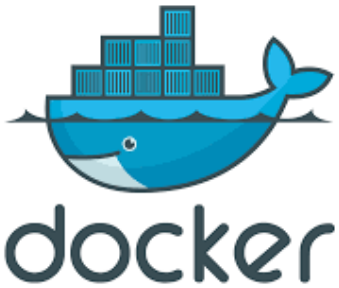
Installing Drake (via Docker)

These are our recommended setup instructions for the Fall 2018 version of MIT 6.881, Intelligent Robot Manipulation.



(http://drake.mit.edu)

via 

(http://www.docker.com)

Why Drake and Why in Docker?

For class, we want you to have access to all the power of Drake, and using Docker will help us all standardize the development environment for the class.

Here's some background on both of these tools:

- Drake (http://drake.mit.edu) (with Python bindings, `pydrake`) is a robotics toolbox developed by our lab (http://groups.csail.mit.edu/locomotion), with development now lead by the Toyota Research Institute (http://www.tri.global), that we hope can provide a basis for you to solve interesting problems in robotics without having to build many of the detail-intensive components -- for example, robot models, dynamics, simulators, and visualizers. Drake also strives to provide many features (particularly in exposing and preserving computational structure) that can't really be found elsewhere in a comparable tool.

- Docker (http://www.docker.com) is a virtual-machine-like container system that makes it easier to maintain a reproducible development environment. A Docker *container* functions like a separate OS that you can safely make modifications to without modifying your computer's native OS. We will distribute a Docker *image* that sets up this virtual OS to a configuration that we know will work well with Drake, so that you don't have to do that setup yourself -- and so that if things break, we have a better chance of reproducing your issues.

A big win of the dockerized setup is that we can all run identical Drake on essentially any common distribution of either Windows, Mac, or Linux! (At time of writing, Drake natively supports Ubuntu 16.04 and Mac (http://drake.mit.edu/from_source.html). We recommend starting with the pre-built Docker image described here, but if you're later interested in contributing back to Drake (https://github.com/RobotLocomotion/drake) then you'll want to be building from source.)

Another win is that regardless of the other software on your computer, as long as you can install Docker, then there should be no issues installing Drake.

Overview of Install

There are three overall steps:

1. Make sure your system meets Prerequisites
2. Install Docker
3. Pull down and run a Drake docker container

Prerequisites

- **You'll need to be on 64-bit Windows 10 or 7, Mac 10.10.3 Yosemite or higher, or Ubuntu 14.04 or higher.** MIT's Athena cluster machines running these OS's are only *experimentally supported* (athena_instructions.html), and offer a very poor user experience, so we recommend running your own machine. Other platforms that can run Docker (e.g. Debian, CentOS, Fedora) might work but we can't promise support. *If getting access to a suitable machine is difficult for you, let us know and we can work with you to find a solution.*
- **You'll need around 10GB free.** A single complete Drake image is around 2GB in size (since it contains an entire OS, and a lot of support software), and you may maintain duplicate images, so to be safe, keep around 10GB free.
- **You'll want to be on fast internet when you do an install.** You'll be sad if you're on slow internet and waiting for a couple gigabytes to download!
- **You'll need to use a terminal / command prompt occasionally.** If you haven't used one before, you will want to take a few minutes to learn the basics of using one (https://www.digitalocean.com/community/tutorials/an-introduction-to-the-linux-terminal).
- **You'll be interacting with Python code via a Jupyter notebook.** You can read up on details about what that means here (https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/), but you don't really need to know it in-depth. The gist of it is that we'll mostly interact

with PyDrake through a web page on which you can edit and run Python code interactively.

Install Docker

First you will need to install Docker, following the instructions below based on your OS.

Pick your platform (click on name of your OS):

Linux | Mac | Windows

Linux

Follow the instructions here (https://docs.docker.com/install/linux/docker-ce/ubuntu/) for your system. We will only officially support Ubuntu 14.04 and 16.04 but you can probably get other systems working too.

The easiest of install methods, if you trust Docker, is to just run these scripts:

```
curl -fsSL get.docker.com -o get-docker.sh
sudo sh get-docker.sh
```

Now you can go to the *After Docker Installation* section!

After Docker installation: Pull down and run a Drake docker container

- First let's test Docker. Open your favorite terminal (unless you're on Windows 7 – y'all have to use the Docker Quickstart terminal) and type:

```
docker run hello-world
```

You should see output indicating that Docker downloaded an image named `hello-world` and launched it to say ”Hello from Docker!”

- Provided here are utility scripts (download and unzip) (drake_docker_utility_scripts.zip) for setting up a convenient developer environment. Both of these scripts rely on the idea of "mounting" a shared directory -- that is, sharing files between your host OS and the OS within the docker container. Specifically, both scripts take as their second argument a directory that you can edit from both outside the container, as well as inside it at path `/notebooks`. This way, work you do inside the container within that directory is saved. You'll usually want to point this path to whatever directory contains your homework code.

  There are two types of scripts:

  1. *For using Jupyter notebooks to interact with the class code.*

     docker_run_notebook scripts : These launch the Drake Docker container, using 1) a problem-set-specific *tag*, which will be provided with each problem set, and 2) a *notebook directory*, which should be a relative path to where put your problem-set-specific code. E.g.:

     ```
     ./docker_run_notebook.sh drake-20190308 ./6881_pset1_code
     ```

     A batch version is provided for Windows 10 users. Everyone else (including Docker Toolbox users on Windows 7) should use the .sh files.

  2. *For using a bash terminal to run scripts, full bash control, etc.*

docker_run_bash scripts: These launch the Drake Docker container, using 1) a problem-set-specific *tag*, which will be provided with each problem set, and drops you in an interactive bash terminal inside of the container, and 2) a *notebook directory* to mount inside the container at `/notebooks`. You can use this to, e.g., run python scripts from the command line. A MinGW-specific version is provided for Docker Toolbox users, and a batch file is provided for Windows 10 users. These both require a 3rd argument: your computer's IP, as you can find from *ipconfig*.

These scripts vary more for different platforms:

```
# use the script that matches your platform
./docker_run_bash_linux.sh drake-20190308 .
./docker_run_bash_mac.sh drake-20190308 .
./docker_run_bash_mingw.sh drake-20190308 .
./docker_run_bash_win10.sh drake-20190308 .
```

- Run the notebook script from your favorite terminal, using a tag (e.g. *drake-20190308*) and your current directory:

```
# on most platforms
./docker_run_notebook.sh drake-20190308 .
```

```
# on Windows 10
./docker_run_notebook.bat drake-20190308 .
```

You'll see a relatively large download happen, and eventually you should see output including:

```
...
[I 16:00:11.467 NotebookApp] The Jupyter Notebook is running at:
[I 16:00:11.467 NotebookApp] http://(e8b19aaed3d0 or 127.0.0.1):8080/
...
```

which means that everything launched OK!

- Open a web browser and point it to the appropriate hostname based on your OS:
    - **Linux and Mac:** *http://127.0.0.1:8080/*
    - **Windows 10:** *http://127.0.0.1:8080/*
    - **Windows 7:** Open a new Docker Quickstart terminal and run `docker-machine ip` to get an IP address. Use *http://[that IP]:8080/*

  You should get a login prompt for a Jupyter notebook. Log in in with password **mit6881**.

- Start a new Python 2 notebook, and type in the first cell:

```
import pydrake
print "Hello world!"
```

Press *Control+Enter* to run the cell, and you should see "Hello world!" printed below the cell. If this is true, everything works, and you can use PyDrake from this notebook! (Go do a problem set if you want to see something more exciting than text output.)

- Also here is how to run an example with graphics using the bash script.

```
## shown for linux below, or use the docker_run_bash script for your system
./docker_run_bash_linux.sh drake-20190308 .
cd /drake/examples/manipulation_station
python end_effector_teleop_sliders.py --meshcat --setup=clutter_clearing
```

Open the printed URL (something like http://127.0.0.1:7000/static/) in a browser to open meshcat. You should also see a slider window pop up! (If it doesn't please re-read the Docker install instructions, including XQuartz / X Server items, and make sure you have the latest scripts (drake_docker_utility_scripts.zip)). If you move the sliders, you should see the robot moving in meshcat.

All done!

Web design with Bootstrap | Contact: manipulation-tas [AT] mit [dot] edu