Algorithmic Robotics
COMP/ELEC/MECH 450/550
OMPL Installation

For the project assignments you will use the Open Motion Planning Library (OMPL), developed by the Kavraki Lab. The library contains implementations of many sampling-based motion planning algorithms. In addition to the library, there is also a lightweight front-end named OMPL.app. The front-end consists of a simple GUI that allows you to use a variety of motion planning algorithms to plan motions for rigid bodies. Also included with OMPL.app are a number of command line programs that demonstrate how to integrate OMPL with libraries for reading graphical meshes and collision checking. These programs can form the starting point for completing some of the later projects. The main library is written in C++11.

The documentation for OMPL can be found at http://ompl.kavrakilab.org.

As part of the class, we provide some (hopefully) easy to use containers that contain both OMPL and OMPL.app preinstalled. The container also contains a built-in VNC server for visualization.

WARNING: be careful when copying and pasting from the PDF: sometimes extra spaces can be included that will affect the command be run.

Recommended: Using Docker

We provide a Docker image that can be used for development. You should first install Docker on your platform of choice, see Docker's documentation for more information. If on Linux, you should also follow the post-installation steps for you platform. This will enable you to run Docker not as the root user. **NOTE:** If you see permission denied errors, this probably relates to you not having permission to run Docker on your machine. See the post-installation steps or try to run as administrator.

Once Docker is installed, you should pull the image:

```
docker pull cronus.cs.rice.edu/comp450:2023
```

We recommend using Docker compose to simplify running the container. We provide a `docker-compose.yml` file that you should use in order to launch the environment. Download and put it into the directory you like to use for this class. **IMPORTANT**: make sure you edit the lines in the file that correspond to mounting your local directory.

In the directory where you have `docker-compose.yml`, you can run the container with the following (run it from the same directory that you have this file):

```
docker-compose up
```

This should start up the machine with the VNC server running. Navigate to a web browser and open http://localhost:6080. This should give you access to the desktop of the docker container. Try running a terminal and opening up the program:

```
ompl_app
```

You can also run the docker container in a non-desktop mode, with:

```
docker-compose run --rm --entrypoint bash ompl
```

Mac Users

If you are using a Mac, you will have to perform some different steps after installing Docker.

Note that if you do note start the Docker daemon on your machine, Docker will not be able to start. On Mac, launch the Docker application to start the Docker daemon. Additionally, you might run into issues finding the docker command from your terminal. If this happens, add the following line to your ∼/.zprofile file:

```
export PATH="\$PATH:/Applications/Docker.app/Contents/Resources/bin/"
```

Windows Users

If you are using Windows, follow the official instructions to install Docker Desktop https://docs.docker.com/desktop/install/windows-install/. You can verify if the installation is successful by launching the Docker Desktop. If you see an error about WSL needs an update, please go to the following page to download the update https://learn.microsoft.com/en-us/windows/wsl/install-manual#step-4---download-the-linux-kernel-update-package.

Note that always make sure the Docker engine is up. If it's not, simply launch the Docker Desktop application to start the Docker engine.

To launch the docker container that we had built up for you, first launch a command window shell (in modern Windows, you can search for Windows PowerShell in your search box and launch it). In the shell, go to the directory where you save the 'docker-compose.yml' file. Then, simply run

```
docker-compose up
```

The rest of the process should be similar to the instructions for Linux users, please check above.

IMPORTANT: Make sure you specify the volume path correctly in the docker-compose file. (the style of the paths in Windows is different than Linux!)

Using the Virtual Machine

The virtual machine is a VirtualBox image running Ubuntu Linux 18.04 with the latest version of OMPL and OMPL.app. VirtualBox runs on Windows, Mac, and Linux. Note that if you are using a newer Mac (with the M1 chip) you might experience issues with running the virtual machine. To install OMPL using the virtual machine:

1. Download and install **VirtualBox v5.1.4 or newer** from https://www.virtualbox.org/wiki/Downloads.
2. Download http://zeus.cs.rice.edu/OMPL_2021.ova. This file contains a virtual machine with Ubuntu 18.04, OMPL, and OMPL.app pre-installed.
3. Open `OMPL_2021.ova` with VirtualBox by double-clicking on the file. Complete the wizard to install the virtual machine. The wizard allows you to change some properties of the virtual machine to better match your host machine in terms of memory, number of cores, etc. You probably want to increase both the amount of memory and the number of cores allowed to the VM.

Now launch the virtual machine. **The username and password is *ompl***. Installation is successful if you can launch the VM, open the terminal, type `ompl_app`, and the OMPL.app GUI appears. You may see some OpenGL warnings in the terminal when launching the GUI. These are artifacts of hardware virtualization and are usually safe to ignore.

Graphical performance in the GUI can be improved by enabling 3D hardware acceleration in the virtual machine's graphics settings. Note: if you enable acceleration, the OpenGL canvas may appear on top of other windows in older versions of Ubuntu and VirtualBox - this is a limitation of VirtualBox.

Additionally, Python bindings for the main library have been generated and are available on the virtual machine. These are available with Python 3 only.

We recommend that you set up a shared directory with the VM and your host machine, so you can easily edit with your workflow of choice on your host machine and compile/run projects in the VM.

Installation from Source

Alternatively, you *can* find installation procedures for several operating systems at http://ompl.kavrakilab.org/installation.html. Note that the TAs can provide only limited support for users building from source or installing packages from the respositories. We will assume in the project documentation that you are using the provided virtual machine.

OMPL can be easily installed from source on Ubuntu Linux and Mac OS X. Source installation on Windows is possible, but is considered experimental.

Ubuntu Linux If you want to install OMPL and OMPL.app from source, you need to have Ubuntu Linux 14.04 (Trusty) or higher. Download the installation script and follow the instructions at http://ompl.kavrakilab.org/installation.html.

Instructions for the installation of packages are also available, but the version of OMPL provided in the Ubuntu repositories is not the latest release and can create conflicts with class instructions. Installation of the packages is only suggested if the user is an expert in C++ and open source build environments.

OS X Install MacPorts (http://www.macports.org). Type the following two commands in Terminal.app:

```
sudo port sync \; install ompl +app
```

OMPL.app is installed in `/opt/local`. Type "`port contents ompl`" to see what exactly got installed.

MS Windows Installation on MS Windows is more involved, simply because there is no package manager. See the installation web page for details. It is *substantially* easier to use the virtual machine.

Compiling your own code

The OMPL website has a page with directions for setting up your build system to compile code you write for your projects: http://ompl.kavrakilab.org/buildSystem.html. **NOTE:** we will provide Makefiles for each project.

Submission

The due date of this project is August 29th, 2023. Please compile and upload a short **pdf** file, specifying: 1, what operating system you are using; 2, how did you install it (docker, VM, from source, etc.); 3, how long did it take you to finish this project; 4, (optional) what were the difficulties you met during installation, if any.