CS 7638 Robotics:AI Techniques - Environment Setup

Introduction

This document will walk through setting up your local development environment or a virtual machine. Setting up your environment will ensure that you have an appropriate version of python installed along with the necessary libraries used in this course. If you are totally new to Python, you can refer to https://docs.python.org/3/tutorial/index.html to understand the basics of Python. All problem sets and projects in this course are designed to use Python 3, and we recommend Python 3.10. The autograder tool, Gradescope, uses at least version 3.10.12.

If you are comfortable setting up your environment on your own, feel free to reference the yaml file for the necessary packages to install. You will also need to install the course specific package (rait), see manual installation instructions below.

Download Setup files

On Canvas, click on `Files` in the left side menu, select `Python Environment Setup`, and download the following:

- `cs7638_env_setup.pdf` (this file)
- `rait_env.yml`
- `test_env_setup.py`
- `rait/`

Note that you can download a zip of this folder by: https://guides.instructure.com/m/4212/l/41998-how-do-i-download-a-folder-in-zip-format-as-a-student

Place these files into a directory (you can call it `student_env_setup`).

Environment setup

These steps will create an environment that you can use specifically for this course. This will isolate the libraries and versions used inside the `rait_env` environment so as to not conflict with any other python installations you may have now or create in the future.

1. Download and install Miniconda here https://docs.conda.io/en/latest/miniconda.html. Use the latest python version available. What is Conda? Conda is an open source package and environment management system. We recommend using Conda since it makes it easy to install and manage different versions of libraries without messing up other environments. You are welcome to use Anaconda instead of Miniconda, see **this page** for more details on the differences:
2. On Windows, open the installed "Conda prompt" to run your commands. On MacOS and Linux, you can just use a terminal window. Change directory (using `cd`) to the location of the directory `student_env_setup` (the directory that contains the files `rait_env.yml`, `test_env_setup.py`, `cs7638_env_setup.pdf`, and folder `rait/` you downloaded from Canvas).
3. Create a conda environment by running the following command in the "Conda Prompt" (Windows) or Terminal (MacOS/Linux):

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

4. This should create an environment named `rait_env`. Activate it using the following Windows command: `activate rait_env` or the following MacOS / Linux command: `conda activate rait_env`

Since you may have multiple environments installed on your computer, you will need to remember to activate this specific one (`rait_env`) any time you wish to work on code related to this course. If you are working on files in this course while a different environment is active then you may encounter errors because certain packages are missing or their version is incompatible.

Manual Installation of 'rait' package

If you are not using our instructions above to install your environment, but instead are installing it on your own, then you will need to manually install the rait module. You can do this by downloading the 'rait/' folder. Assuming the 'rait/' folder is saved in a folder named 'student_env_setup' then you will run the following command inside the 'student_env_setup' folder:

```
pip install rait/
```

Check your installation

Now that you've installed python and created your environment, you're ready to verify that you have everything set up correctly! You can run the script `test_env_setup.py` using your newly installed version of python. It is a simple script that checks the following:

- installed python version
- installed necessary libraries (including the rait specific package)
- unicode characters display correctly (arrows helpful for debugging later in the course)
- GUI library tkinter properly working (a window should pop up with 2 buttons: `Click me!` and `QUIT`. The `Click me!` button should add an extra set of square brackets around the `Click me!` text each time it is clicked. The `QUIT` button should make the window disappear.)

You can run it by: `python test_env_setup.py`

Note: Be sure to activate your rait_env before running the above command.

Installation Troubleshooting

- Arrows not displaying

  It is likely the case that you do not have a font on your machine that is able to display these specific unicode characters. You can install the `symbola` font which is able to display these characters. (Mac & Windows: You can download the font **here** or find it through a google search).

  Linux: `sudo apt install fonts-symbola`

  Mac: **How to install a font on Mac**

  Windows: **How to install a font on Windows**

- Arrows (still) not displaying

  Windows: **Switch your font** to MS Gothic and try again (you'll only need these for the Search project so feel free to switch at that time since the MS Gothic font may introduce some other annoying characters in your prompt)

- Tkinter not working correctly on Mac

  Mac: `brew install python-tk@3.10`

- Buttons on GUI pop-up are all white (no text)

  Mac: This is likely due to being in dark mode, switch to the mac light theme and see if it resolves the issue

- Using WSL on Windows (GUI issues)

  See **this first** guide and **this second** guide to install an x-server (don't forget to add the last line to your .bashrc, this is the step that fixes the issue with $DISPLAY environment variable). **This third** guide may be helpful as well (the `display id` may be different for some machines)

RAIT Virtual Machine (VM) Image

The recommended method to execute the code in this course is through your native OS (Windows/Mac/Linux). However, an alternative is to use a virtual machine (VM). You are fine to use the VM to run your code but know that it is a little less of a seamless process. If you are not able to get python successfully installed using the steps listed above then you may choose to use the optional VM image located in `Canvas->Files->Optional Class VM Image`. Instructions to help you install it are included in a PDF document along with the .ova image file. Note that this image has everything you need already pre-installed. It does NOT use conda, so you will not need to activate your environment inside the VM. It is recommended that you perform the `check your installation` step above in the VM once you have it installed and running.

PyCharm Setup

You may choose to use any Python IDE including PyCharm, Visual Studio Code, Sublime, Atom, VIM, etc, or you may also use just a plain text editor and a command line. Below are the steps to setup PyCharm and configure it to use the conda environment that we created above.

Please note that the instructions below are for high-level guidance specific to Linux for PyCharm Professional 2020.2. The exact paths or options may differ for you based on your system. You may refer to the provided PyCharm links in the steps if your operating system or PyCharm version is different.

1. Download and setup PyCharm https://www.jetbrains.com/pycharm/download.
2. Open the directory `student_env_setup` in PyCharm https://www.jetbrains.com/help/pycharm/opening-reopening-and-closing-projects.html#opening_projects.
3. Configure PyCharm to use the conda environment created above https://www.jetbrains.com/help/pycharm/conda-support-creating-conda-virtual-environment.html.

   a: Press Ctrl+Alt+S to open the project Settings/Preferences.

   b. In the Settings/Preferences dialog, select `Project <project name> | Python Interpreter`. Click the icon next to the Python Interpreter dropdown and select Add.

   c. In the left-hand pane of the Add Python Interpreter dialog, select Conda Environment.

   d. Select Existing Environment.

   e. Click Select an interpreter and specify a path to the Conda executable in your file system. To see the path of the conda environment in your system, run the command `conda info --envs` and note the path of the `rait_env`. In the Interpreter path on PyCharm, add that path to the rait_env environment, followed by `bin/python`. An example path would be `/home/user/anaconda3/envs/rait_env/bin/python`.

   f. Select the checkbox Make available to all projects

   g. Apply the changes.
4. Now, to run the file test_env_setup.py, right click on it in the Project window in the left and select Run.

Windows users may refer to `https://www.youtube.com/watch?v=1gtHso20YMQ` for installing Miniconda and PyCharm if you face issues.

MacOS users may refer to `https://www.youtube.com/watch?v=yQo1kb0_8EI` for installing Miniconda and PyCharm if you face issues.