Installing Required Software for *EYW: Computing* on Macs

(last updated 5/24/22)

This document contains everything that teachers and students need to download and install the software that is required to teach the 2022-23 *Engineer Your World: Computing* curriculum on Macs.

For PCs, please see the document 'EYW-Computing_Software_for_PCs_2022-23'. While the curriculum can be run on both PCs and Macs, if you have the choice between the two, we recommend using PCs because they tend to have fewer issues with software installation and OS updates that affect the functionality of the necessary software. Also, many of the lessons show screenshot examples from PCs.

***A note about the newer Macs with an M1 chip (versus an Intel chip). How do I know if I have an M1 chip? Click on the Apple icon in the upper left corner -> About This Mac. Next to Chip it will say something like "Apple M1". The M1 Macs cannot properly install PortAudio, so you should skip the steps below titled "Installing Homebrew and PortAudio" as well as "Installing the Pysine library". Fortunately, this software is only needed in a very brief (~1 week) introductory unit in the curriculum: Electronic Music. Every other unit works on an M1 Mac (with some small workarounds). We recommend students use an alternate computer for this one intro unit, or do a separate Python introductory tutorial. Please contact EYW for more info about this.*

Contents

What software do we need and why do we need it? ***1***

Software Installation ***2***

- **Installing Python 3.7 and Thonny (combined installation)** **2**
- **Installing the OpenCV and NumPy libraries (combined installation)** **3**
- **Installing Homebrew and PortAudio *(* skip this if you have an M1 Mac)*** **5**
- **Installing the Pysine library *(* skip this if you have an M1 Mac)*** **6**
- **Installing the CameraController App** **7**

End of software installation ***8***

Common Issues That Might Occur on Macs ***8***

Appendix for Macs with the newer M1 chip ***9***

What software do we need and why do we need it?

- **Python 3** - The programming language that students will use to write programs throughout the year. The Thonny IDE (below) comes with Python 3.7 built in, so Python and Thonny will be installed together.
- **Thonny** - An Integrated Development Environment (IDE) for Python that is designed for beginners. This is where students will compose their Python code throughout the year. Thonny offers both ease of use and consistency across the year, since this is the IDE that students will use to code on the Raspberry Pis in the spring semester.

- **OpenCV** (Open Source Computer Vision) **Library** - A library that contains copious functions for analyzing and manipulating digital images. It is needed for the *Images* and *Motion Capture* units.
- **Numpy Library** - A library that adds support for multidimensional arrays (*e.g.*, digital images) with a collection of mathematical functions to operate on these arrays. NumPy automatically comes installed with the OpenCV library (above). It is needed for the *Images* and *Motion Capture* units.
- **Homebrew** - A software package management system that assists with the installation of software on macOS. It supports the installation of the PortAudio Library (below) and is needed for the *Electronic Music* unit. *(* not useful on an M1 Mac)*
- **PortAudio Library** - A library that supports audio input and output. It is required by the Pysine Library (below) and is needed for the *Electronic Music* unit. *(* doesn't work on an M1 Mac)*
- **Pysine Library** - A library that contains useful functions for playing sound through the computer's internal speaker. It is needed in the *Electronic Music* unit. *(* doesn't work on an M1 Mac)*
- **CameraController App** (optional but recommended) - An app that offers extra functionality for controlling the webcams that are provided in the course equipment kit. Note that this app is not on the official Apple App Store, and hence may be a cause of concern for a school's IT department. Do not worry if you find yourself in this situation. Students can still complete the curriculum without this app, although with slightly increased difficulty in setting filters in the *Motion Capture* unit.

Software Installation

Install the following software on the teacher's computer and on all student computers prior to the beginning of the year.

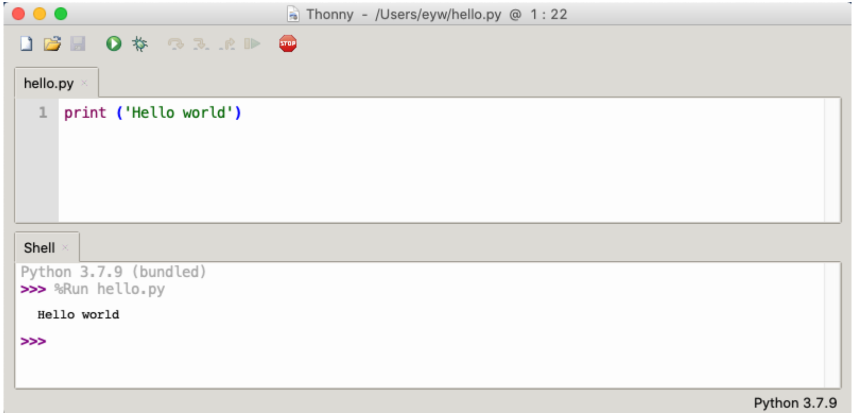
Installing Python 3.7 and Thonny (combined installation)

1. Go to https://thonny.org/
2. Click on the download link for the Mac installer. A pop-up window should appear asking what to do with the .pkg file.
3. Choose the Open with Installer (default) option. Click OK.
4. Continue and agree through all of the steps of the installation process. (We recommend that you install for all users on the computer, but if you are on a school computer your IT person may require the installation to be limited to a particular user or disk.)
5. Click Install. Enter the admin password if prompted. You should see a notice once the installation is successful.
6. Click Close. (At this point you can delete the .pkg file - simply drag the cardboard box icon into the trash bin.)
7. Start Thonny by opening a finder window, going into Applications, and clicking on the app icon. (Click 'Open' if you get a pop-up message about it being downloaded from the internet.)

8. Choose the standard initial settings and click ‘Let’s go!’
   You should see the Thonny app window appear; it will likely have two inset windows, <untitled> and Shell, as shown below, where <untitled> has now been replaced by the saved file name.
9. Test to see if everything works by writing a short program:
   a. In the upper <untitled> window type the following:

      ```
      print ('Hello world')
      ```

   b. Click the Run icon or press the F5 key.
   c. It will prompt you to name the file and choose its location. Choose anything you like (such as hello.py on the Desktop) and click Enter. Note that python code is saved with the extension .py. If you do not add this extension to your file name, Thonny will do it automatically for you.
   d. In the lower Shell window you should see a report that looks something like this:



```
print ('Hello world')
```

```
Python 3.7.9 (bundled)
>>> %Run hello.py

  Hello world

>>>
```

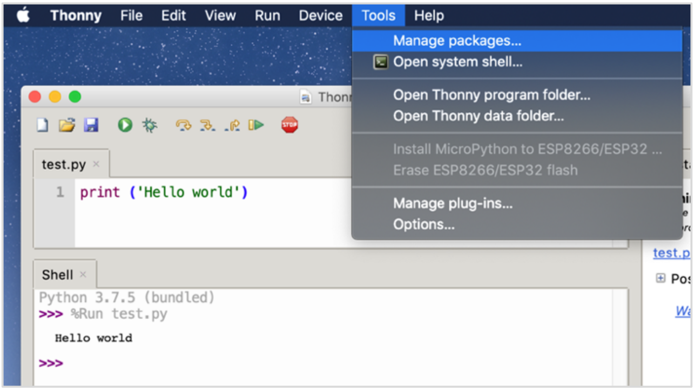
You just used Thonny to program in Python 3.7!

Installing the OpenCV and NumPy libraries (combined installation)

***If you have a newer Mac with an M1 chip (versus an Intel chip), then skip these next 3 sections and instead follow the instructions in the [appendix at the bottom of this document](#).*
How do I know if I have an M1 chip? Click on the Apple icon in the upper left corner -> About This Mac.
If you have an Intel chip, continue here…

1. In the Thonny app navigate to Tools > Manage Packages

   This will open up a new window titled ‘Manage packages for……’


Thonny
File
Edit
View
Run
Device
Tools
Help
Manage packages...
Open system shell...
Open Thonny program folder...
Open Thonny data folder...
Install MicroPython to ESP8266/ESP32 ...
Erase ESP8266/ESP32 flash
Manage plug-ins...
Options...
test.py
Shell

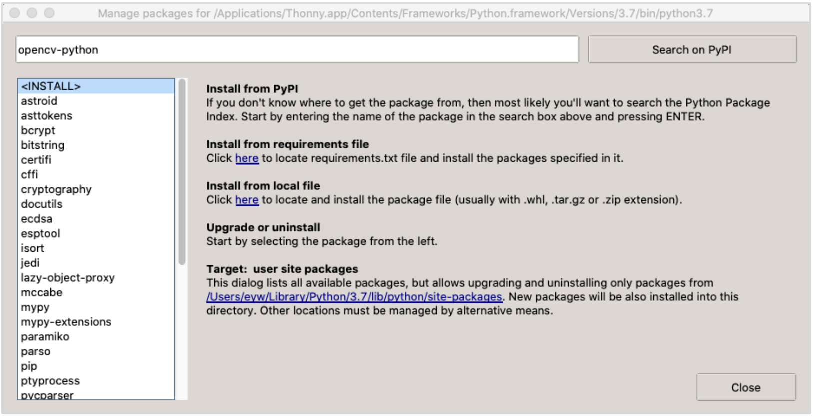
```
print ('Hello world')
```

```
Python 3.7.5 (bundled)
>>> %Run test.py

  Hello world

>>>
```

2. In the search bar at the top of the new window enter “opencv-python” (without the quotes) and click ‘Search on PyPI’.



Manage packages for /Applications/Thonny.app/Contents/Frameworks/Python.framework/Versions/3.7/bin/python3.7

opencv-python

Install from PyPI
If you don't know where to get the package from, then most likely you'll want to search the Python Package Index. Start by entering the name of the package in the search box above and pressing ENTER.

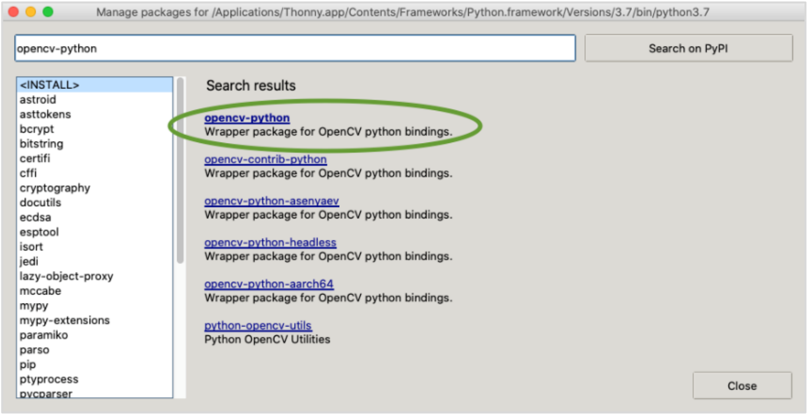
Install from requirements file
Click here to locate requirements.txt file and install the packages specified in it.

Install from local file
Click here to locate and install the package file (usually with .whl, .tar.gz or .zip extension).

Upgrade or uninstall
Start by selecting the package from the left.

Target: user site packages
This dialog lists all available packages, but allows upgrading and uninstalling only packages from /Users/eyw/Library/Python/3.7/lib/python/site-packages. New packages will be also installed into this directory. Other locations must be managed by alternative means.

3. It will find the latest stable packages that contain OpenCV for Python. You only need the top one: opencv-python. Click on that.



Manage packages for /Applications/Thonny.app/Contents/Frameworks/Python.framework/Versions/3.7/bin/python3.7

opencv-python

Search results

opencv-python
Wrapper package for OpenCV python bindings.

opencv-contrib-python
Wrapper package for OpenCV python bindings.

opencv-python-asenyaev
Wrapper package for OpenCV python bindings.

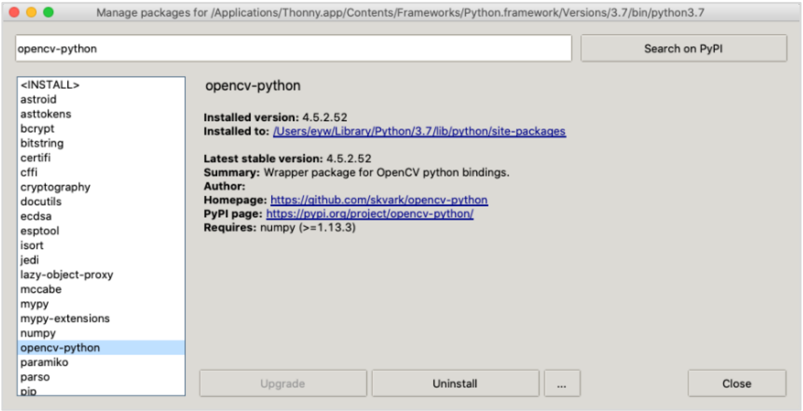
opencv-python-headless
Wrapper package for OpenCV python bindings.

opencv-python-aarch64
Wrapper package for OpenCV python bindings.

python-opencv-utils
Python OpenCV Utilities

4. On the next screen (not shown here) click ‘Install’.

5. Another window will pop up showing a log of the installation process. This pop-up window will close when it is finished. You’ll then see the new buttons ‘Upgrade’ (grayed out) and ‘Uninstall’ in the other window. Click Close.



Manage packages for /Applications/Thonny.app/Contents/Frameworks/Python.framework/Versions/3.7/bin/python3.7

opencv-python

opencv-python

Installed version: 4.5.2.52
Installed to: /Users/eyw/Library/Python/3.7/lib/python/site-packages

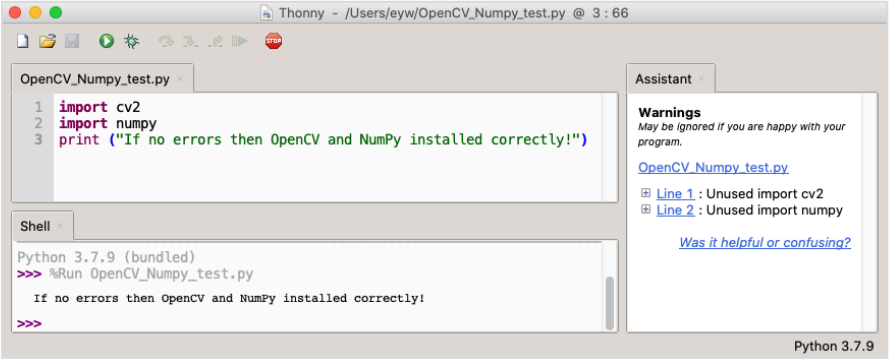
Latest stable version: 4.5.2.52
Summary: Wrapper package for OpenCV python bindings.
Author:
Homepage: https://github.com/skvark/opencv-python
PyPI page: https://pypi.org/project/opencv-python/
Requires: numpy (>=1.13.3)

6. You’ve just installed OpenCV, which comes with NumPy included. Note that OpenCV needs to be imported as “cv2” in your python programs, as in the example below. Now let’s test to make sure they’re working properly.
   a. Open a new file in Thonny with the 🗋 icon.
   b. Type the following in the upper window:

```
import cv2
import numpy
print ("If no errors then OpenCV and NumPy installed correctly!")
```

c. Click Run and save the file with whatever name you want (such as OpenCV_Numpy_test.py).

You should see something like this. You may not have an Assistant window on the right, but if you do, you can ignore the warnings. If you see the print statement, then everything is installed correctly.



```python
import cv2
import numpy
print ("If no errors then OpenCV and NumPy installed correctly!")
```

```
Python 3.7.9 (bundled)
>>> %Run OpenCV_Numpy_test.py

  If no errors then OpenCV and NumPy installed correctly!
>>>
```

Warnings
May be ignored if you are happy with your program.

OpenCV_Numpy_test.py

⊞ Line 1 : Unused import cv2
⊞ Line 2 : Unused import numpy

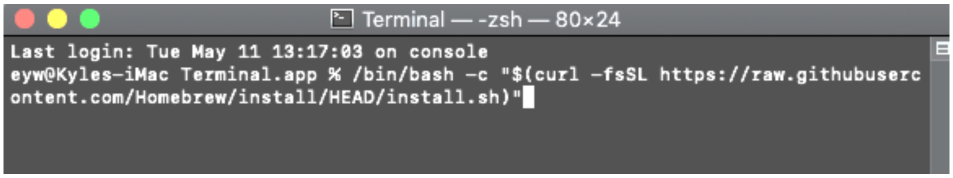
Was it helpful or confusing?

Installing Homebrew and PortAudio *(* skip this if you have an M1 Mac)*

1. Open the Terminal app. You can do this by searching in a finder window for "terminal app" (search 'This Mac'). The app will look like this:

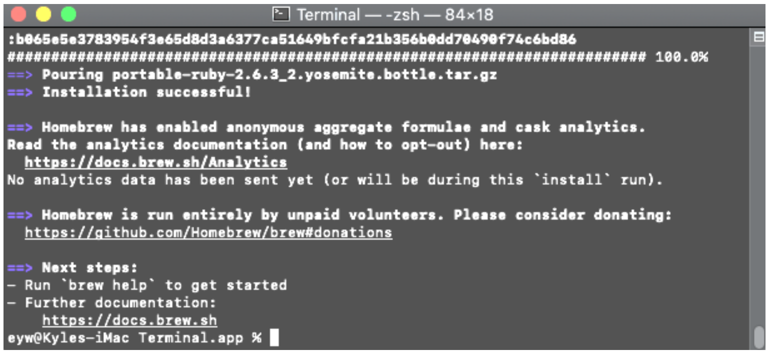
2. In the terminal, copy/paste the following line of code (taken directly from the Homebrew website) and press enter:

/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"



```
Last login: Tue May 11 13:17:03 on console
eyw@Kyles-iMac Terminal.app % /bin/bash -c "$(curl -fsSL https://raw.githubuserc
ontent.com/Homebrew/install/HEAD/install.sh)"
```

3. The script will execute and prompt you to press Return and/or enter your admin password. Accept any default prompts that may come up (by pressing Enter). The entire process will likely take several minutes. It might also need to install Xcode Command Line Tools on your computer; allow it to do so. Once it has finished it will return to the original prompt, which typically shows the name of your computer or your username followed by $ or %.



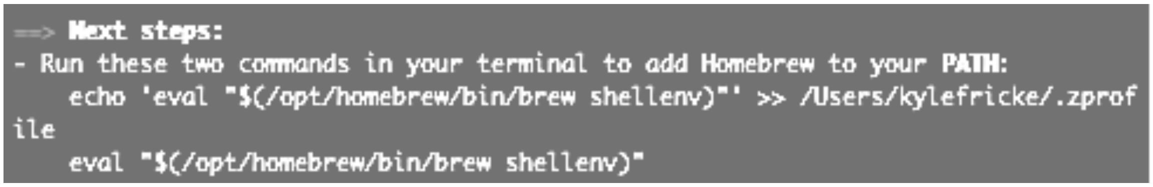
```
:b065e5e3783954f3e65d8d3a6377ca51649bfcfa21b356b0dd70490f74c6bd86
######################################################################## 100.0%
==> Pouring portable-ruby-2.6.3_2.yosemite.bottle.tar.gz
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics data has been sent yet (or will be during this `install` run).

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
  https://github.com/Homebrew/brew#donations

==> Next steps:
- Run `brew help` to get started
- Further documentation:
    https://docs.brew.sh
eyw@Kyles-iMac Terminal.app %
```

4. Look up several lines from the bottom to see if there is a section that says **==> Next steps:**
    - If so, follow those instructions. In this example case in the image immediately below, it requested that two more lines of code be run (which we did by copy/pasting those lines into the prompt at the bottom of the terminal window, running them one at a time). After you're done with the "Next steps" continue to step 5 below.



```
==> Next steps:
- Run these two commands in your terminal to add Homebrew to your PATH:
    echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> /Users/kylefricke/.zprof
ile
    eval "$(/opt/homebrew/bin/brew shellenv)"
```

    - If there is no "Next steps" message, then continue below…

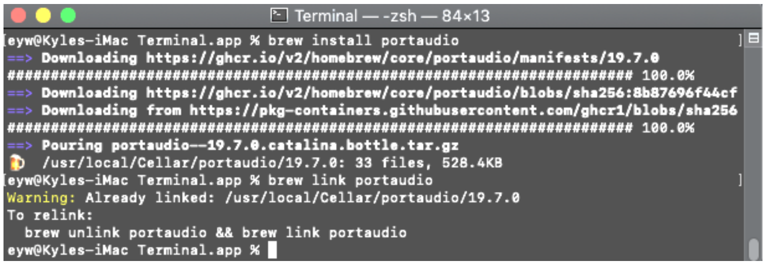
5. Next enter the following line into the terminal and press enter:

```
brew install portaudio
```

6. Then enter this line:

```
brew link portaudio
```

   Ignore any warning about it already being linked.



```
eyw@Kyles-iMac Terminal.app % brew install portaudio
==> Downloading https://ghcr.io/v2/homebrew/core/portaudio/manifests/19.7.0
######################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/portaudio/blobs/sha256:8b87696f44cf
==> Downloading from https://pkg-containers.githubusercontent.com/ghcr1/blobs/sha256
######################################################################## 100.0%
==> Pouring portaudio--19.7.0.catalina.bottle.tar.gz
🍺  /usr/local/Cellar/portaudio/19.7.0: 33 files, 528.4KB
eyw@Kyles-iMac Terminal.app % brew link portaudio
Warning: Already linked: /usr/local/Cellar/portaudio/19.7.0
To relink:
  brew unlink portaudio && brew link portaudio
eyw@Kyles-iMac Terminal.app %
```

You now have Homebrew and PortAudio installed and are ready to install Pysine. You can close the terminal window.

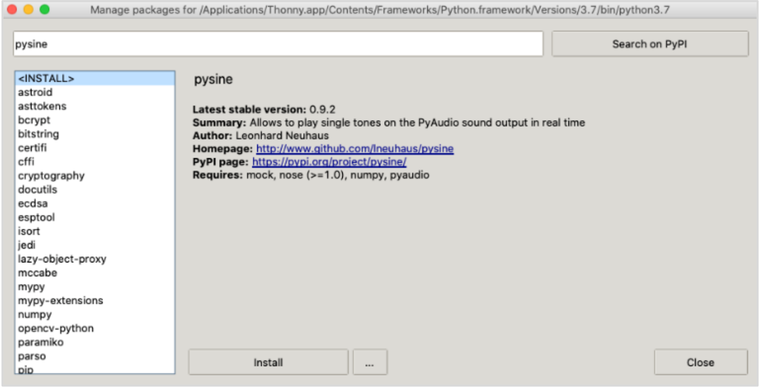
Installing the Pysine library *(* skip this if you have an M1 Mac)*

This process will be very similar to how you previously installed OpenCV. It can only be done after installing Homebrew and PortAudio.

1. In the Thonny app navigate to Tools > Manage Packages

   This will open up a new window titled 'Manage packages for……'

2. In the search bar at the top of the new window enter "pysine" and click the button 'Search on PyPI'.



pysine

Latest stable version: 0.9.2
Summary: Allows to play single tones on the PyAudio sound output in real time
Author: Leonhard Neuhaus
Homepage: http://www.github.com/lneuhaus/pysine
PyPI page: https://pypi.org/project/pysine/
Requires: mock, nose (>=1.0), numpy, pyaudio

3. It will find the latest stable version of Pysine. Click on the top result (pysine) then click 'Install'. After it's done click Close.

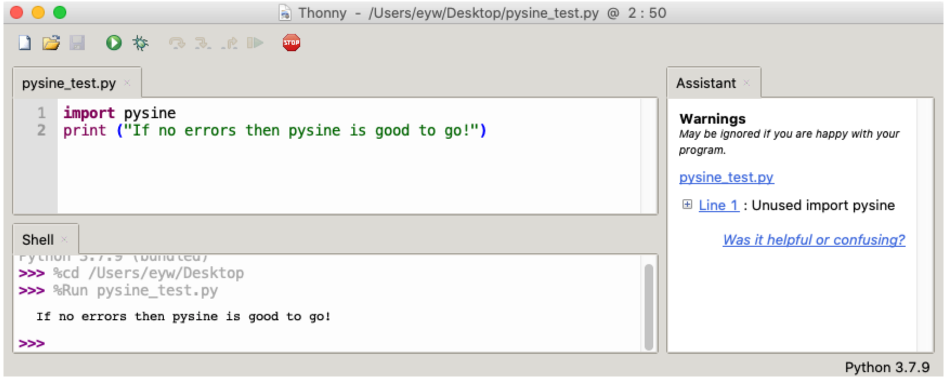
   If you have difficulties with this process, try installing pysine via the terminal instead.

4. You can test to see if pysine is working properly by writing a short program like you did for OpenCV:

   a. Open a new file in Thonny with the 🗋 icon.
   b. Type the following in the upper window:

```
import pysine
print ("If no errors then pysine is good to go!")
```

   c. Click Run and save the file with whatever name you want (such as pysine_test.py).



```
import pysine
print ("If no errors then pysine is good to go!")
```

```
Python 3.7.9 (bundled)
>>> %cd /Users/eyw/Desktop
>>> %Run pysine_test.py

  If no errors then pysine is good to go!

>>>
```

Warnings
May be ignored if you are happy with your program.

pysine_test.py

⊞ Line 1 : Unused import pysine

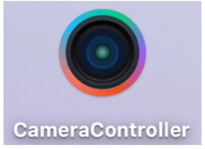
Was it helpful or confusing?

Installing the CameraController App

Note that this app is not on the official Apple App Store, and hence may be a cause of concern for your school's IT department. Do not worry if you find yourself in this situation. Students can still complete the curriculum without this app, although with slightly increased difficulty in setting filters in the *Motion Capture* unit.

1. Go to this link: https://github.com/itaybre/CameraController/releases/tag/v1.4.0
   *(**Why are we getting an app from github rather than the app store?** It is the only free app that offers the desired functionality. The app can disable the webcam's default settings to auto-adjust White-Balance, Exposure, Backlight Compensation, and Focus. EYW has encountered zero issues with using this app.)*
2. Scroll down to the bottom, look under Assets, and click the file CameraController.zip.

3. Download and save the file to your Desktop.
4. Go to your Desktop and open the file. You should see the icon for the appear:
5. Click on that app icon. You'll may get a pop-up warning about the app being downloaded from the internet, click open.
6. The app should open. Test to see if it's working by choosing a camera in the top pulldown menu and make sure that you see an image from the camera.



End of software installation

At this point your Mac computer has everything it needs to complete the *Engineer Your World: Computing* curriculum. All the software needed for setting up and communicating with a Raspberry Pi is already standard on all Macs. You do not need to install anything else.

Common Issues That Might Occur on Macs

- In the Images unit, it won't let the user type in the name of their picture.
  This issue tends to occur on newer Macs. The workaround is to run the program from the terminal. To do this in Thonny, go to 'Run' at the top of the screen, then choose 'Run current script from Terminal'. The first time you do this, you might be prompted to allow a few things. Allow all of them. You may have to try a few times before it works.
- In the Images and MoCap units, once the Esc key is pressed the program freezes.
  The program can be ended if you force-quit Python via option+command+Esc.
  One potential fix is to add a cv2.waitKey(1) both before and after the cv2.destroyAllWindows() at the end of the program:
  ```
      cv2.waitKey(1)
      cv2.destroyAllWindows()
      cv2.waitKey(1)
  ```
  Simply put, OpenCV performs its processing during waitKey() so adding those in help it to get over humps that cause freezes. You sometimes might need to put multiple waitKey(1) commands in a row. This doesn't always work but it is the only known potential fix.
- The trackbars don't work in the Images unit.
  The fix is to use cv2.waitKey(500) instead of cv2.waitKey(1).
- The order of the trackbars is seemingly random in each trackbar window.
  There is no known fix yet.
- Python crashes as soon as OpenCV tries to read the webcam, making it impossible to get started in the initial MoCap programs.
  This issue is known to occur in both macOS 10.14 Mojave, macOS 10.15 Catalina, macOS 11 Big Sur, and macOS 12 Monterrey. It is due to Mac's tightened privacy protections that are

intended to have an app prompt the user before it can access the camera. Most apps haven't been updated to properly ask for this permission yet. The workaround is to run the program from the terminal, which does ask for permission. To do this in Thonny, go to 'Run' at the top of the screen, then choose 'Run current script from Terminal'. The first time you do this, you'll be prompted to allow a few things ("bash" access to the terminal, terminal access to the desktop, and terminal access to the camera). Allow all of them. You may have to try a few times before it works.

- Thonny gives a message saying "Backend terminated or disconnected. Use 'Stop/Restart' to restart" but clicking that button doesn't work.
  This issue typically occurs after you update your Mac's OS while Thonny is already installed. The best-known solution is to completely uninstall Thonny (in the Finder, go to Applications and drag the Thonny icon into the Trash, then restart computer). Also enter this line in the terminal to delete the Library/Thonny folder:
  ```
  rm -r ~/Library/Thonny
  ```
  Next reinstall Thonny again (no need to reinstall OpenCV, Homebrew, or Pysine). However, before starting up this new Thonny install in the usual way, open it from the terminal instead with this line:
  ```
  open -a Thonny
  ```
  This should work. If you get the error again, just close Thonny and enter the two terminal commands above.
- The trackbars don't work. They appear fine, but sometimes flicker and you're unable to adjust them. This typically occurs on a newer Mac with an M1 chip. The fix is to uninstall OpenCV from Thonny (In Thonny, go to Tools -> Manage Packages -> scroll down to opencv-python, click Uninstall), then follow the directions in the appendix below.

Appendix for Macs with the newer M1 chip

If you have a newer Mac with Apple's M1 chip (versus an Intel chip), you'll need to install OpenCV using a different method than the one listed above. This is necessary to prevent an issue with trackbars being glitchy and slow to respond.

Here's what you need to do:

1. Open the Terminal app. You can do this by searching in a finder window for "terminal app" (search 'This Mac'). The app will look like this:

2. In the terminal window, type the following lines and press Enter after each one:

```
pip3 install --upgrade pip
pip3 install opencv-python opencv-python-headless
```

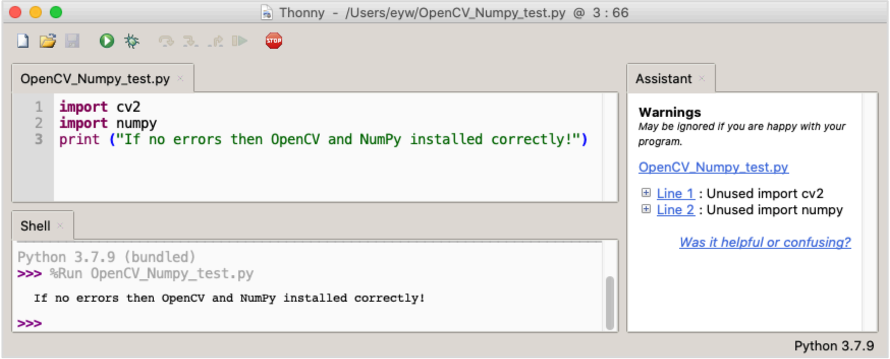
If prompted, enter your admin password and/or accept any default prompts that may come up (by pressing Enter).

3. You’ve just installed OpenCV, which comes with NumPy included. Note that OpenCV needs to be imported as “cv2” in your python programs, as in the example below. Now let’s test to make sure they’re working properly.
   d. Open a new file in Thonny with the 🗋 icon.
   e. Type the following in the upper window:

```
import cv2
import numpy
print ("If no errors then OpenCV and NumPy installed correctly!")
```

   f. Click Run and save the file with whatever name you want (such as OpenCV_Numpy_test.py).

You should see something like this. You may not have an Assistant window on the right, but if you do, you can ignore the warnings. If you see the print statement, then everything is installed correctly.



```
import cv2
import numpy
print ("If no errors then OpenCV and NumPy installed correctly!")
```

```
Python 3.7.9 (bundled)
>>> %Run OpenCV_Numpy_test.py

  If no errors then OpenCV and NumPy installed correctly!

>>>
```

Warnings
May be ignored if you are happy with your program.

OpenCV_Numpy_test.py

⊞ Line 1 : Unused import cv2
⊞ Line 2 : Unused import numpy

Was it helpful or confusing?

Continue to the section above called Installing the CameraController App.