IUPUI Facemorph Python Project

This will try to walk you through environment set up.

**Step 1:** Install Python

Everything was developed in Win10 for Python 3.6 & The packages seem to expect Python to be the 32-bit version.

<https://www.python.org/downloads/release/python-368/>

I downloaded the Windows x86 executable installer

Make sure you install pip as well. I don't remember if they ask if you want it anymore, but we do.

When asked, add Python 3.6 to the PATH, if you don't already have a version of python installed. This will make sure the correct version of pip is being used later.

If you do have a version of python installed that you are already used to using, that's Ok. I'll tell you how to get around any issues that my come up.

**Step 2:** Install the required packages

**Step 2.1:** Verify pip

Open a command line terminal (Open the windows menu and type "cmd")

type pip --version to make sure python 3.6 is being used:



If it is, continue on to step 2.2 otherwise keep reading.

In the command prompt, you will need to navigate to the directory where you installed python 3.6

For me, it is in the Hidden folder called APPDATA:

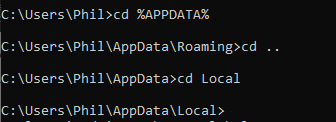
C:\Users\<Username>\AppData\Local\Programs\Python\Python36-32

To navigate to this folder, type the following commands:

cd %APPDATA%

If you are in the Roaming Directory (C:\...\AppData\Roaming) type this:

cd ..



The picture illustrates what it will look like if you are put in the Roaming Folder rather than just the AppData folder.

now you can navigate to the Python install

cd Local\Programs\Python\Python36-32

In this folder you will have an executable file called python.exe.

You can now move to Step 2.2

*Notes on command line:*

In windows, capitalization does not matter.

cd is most easily understood as "Change Directory"

cd .. means go to the parent directory.

If you want to see the folders/files in a directory, you can type this:

dir

**Step 2.2:** Installing Packages with Pip

If just using the pip command worked for you, type the following into the command prompt:

pip install numpy scipy matplotlib ipython jupyter pandas sympy nose pygame PyOpenGL numba

If you needed to navigate to the folder mentioned above type this instead:

python.exe -m pip install numpy scipy matplotlib ipython jupyter pandas sympy nose pygame PyOpenGL numba

The back half of the command is the same as above, but we needed to navigate to the correct folder to get the correct version of pip.

This may take some time, and there will be package dependencies that get downloaded as well.

*Notes on Command Line:*

pip is a built in python package manager.

The --version flag is common in most Command Line Interfaces for applications (at least on Linux Machines)

The -m flag is has a specific meaning for python. It means that a module is being included (in our case, the pip module)

The difference between -- & - in the flags is usually whether the flag is a single letter, or a full word.

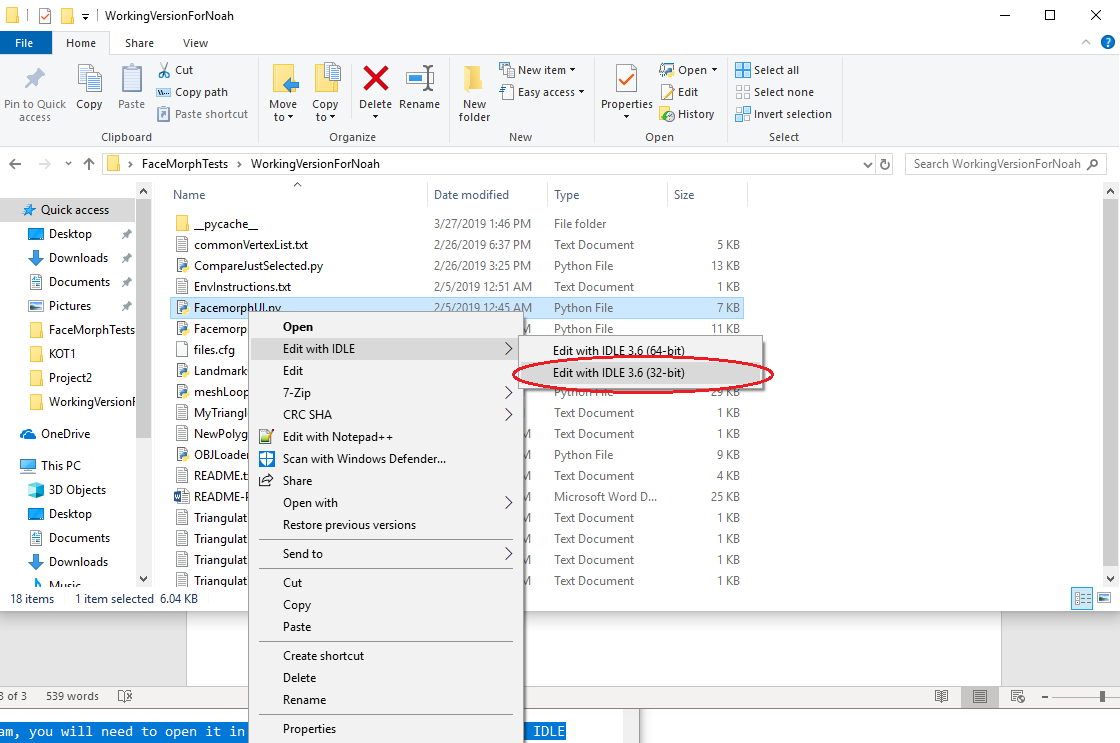
Step 3: Running the UI

To run the program, you will need to open it in the built-in python editor, called IDLE

Using the File System Explorer, navigate to the FacemorphUI.py file.

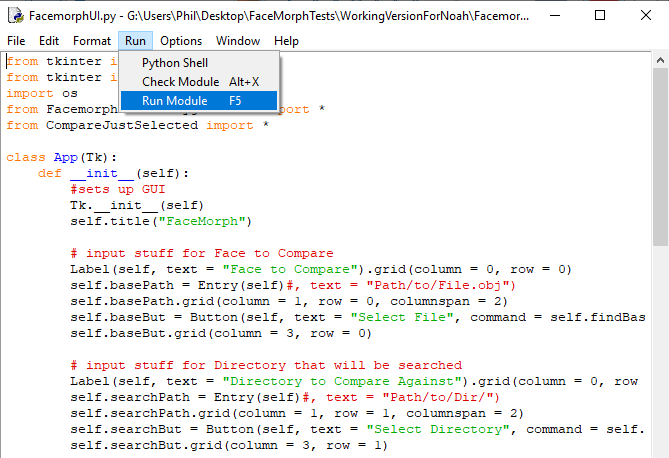
Right click the file and you should see an option that says “Edit With IDLE”.

Mouse over this option and select the 32-bit Version of Python 3.6:



This will open up a text editor with the source code in it.

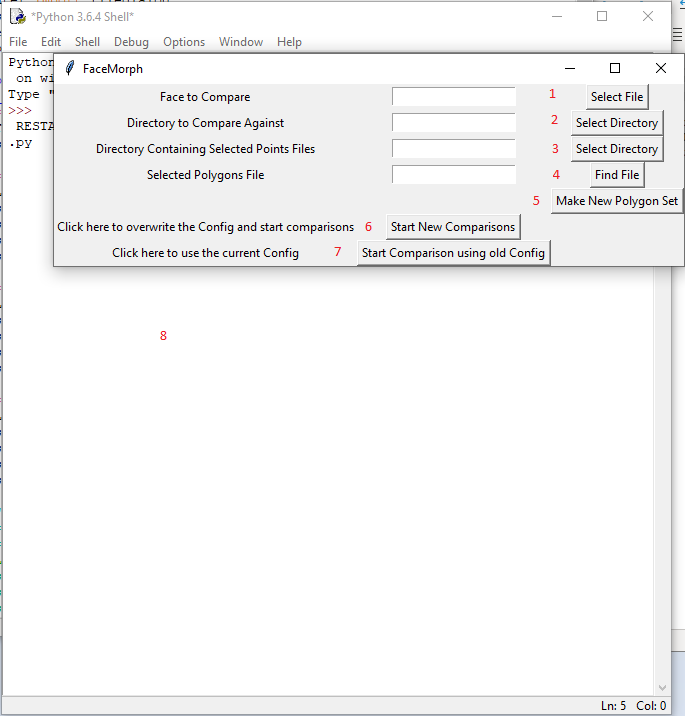
To run the code, either press the F5 key, or click the Run Module option in the Run Menu:



The Program is now running.

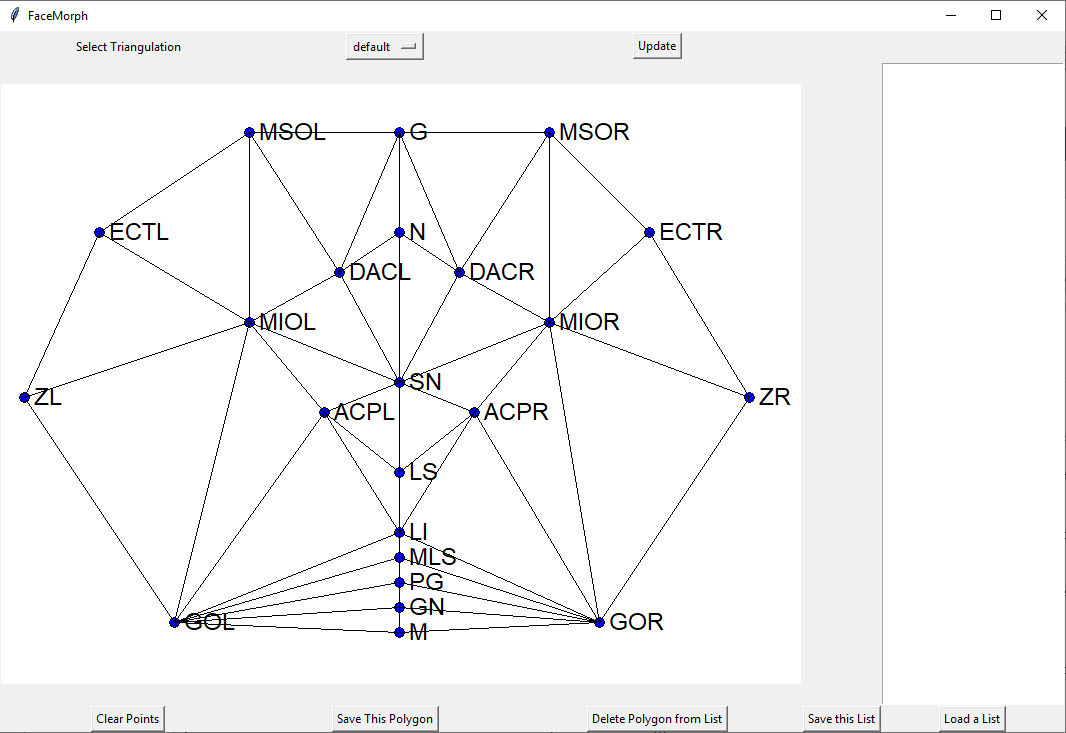
**Step 4:** Operating the UI

Here is what will open up when you run the code:



I have added numbers next to each button, I will describe the function of each of the buttons below:

1. General Notes:
   1. The buttons will open up file dialogs.
      1. On the first run, this may take a second or two.
   2. The text boxes will display the selected file/folder path.
      1. You can also type in the path if you know it, but be careful, incorrect paths will cause errors later on.
      2. Using the buttons & File dialogs is the suggested method
2. Face to Compare:
   1. What will be selected: A single .obj File representing a Face.
      1. This Face file will be compared to all of the other face files in the directory or sub directories of the folder selected in (2)
3. Directory to Compare Against:
   1. What will be selected: The directory containing all of the other .obj files to compare with the face selected in (1)
   2. The other .obj files can be in their own directories. The program will search through all sub-folders in the selected directory.
4. Directory containing Selected Points Files:
   1. What will be selected: A directory containing .txt files with Names that match exactly one .obj file each
   2. This directory will not have its sub-directories looked through.
   3. It is expected that all “Selected points” files are sitting directly in the selected directory.
   4. This may require some manual movement of files.
5. Selected Polygons File:
   1. What will be selected: A single file containing a list of polygons that should be compared
   2. This file can and most likely will be created using the tool launched by (5)
   3. After creating the file using (5) you will still have to select it using this button.
   4. The integration of this feature with the comparison side of the program is still being worked on.
6. Make New Polygon Set
   1. This will open another window of the program that looks like this:



* 1. This window shows the relative positions of the landmarks that are used in the program.
  2. At the top, there are a few different triangulations of the landmarks you can pick between.
     1. If you have selected any points on the canvas, you will not be able to change the triangulation without clearing the selected points or saving the selected points as a polygon.
     2. To change Triangulations, select the one you want, the click the Update Button.
  3. To select points, simply click on them in the canvas portion of the window.
     1. Selected points will be Red, not blue.
  4. Save this polygon will put the points in the textbox on the right.
  5. In the Textbox on the right, you can click on any of your previous polygons to select them.
     1. Selecting Polygons will highlight the vertexes on the canvas.
  6. Using the Delete Polygon from List button, you can remove any selected polygon from the list on the right.
  7. Once you have all of the polygons you want, you can save the list.
     1. The list will be saved as a text file called “NewPolygonSelection.txt”
     2. Clicking Save again will overwrite this file, so if you want to keep both files, either move or rename the first one to something else.
  8. You can also Load an old List using the Load a List button.
     1. This will overwrite the list you are currently working on.
     2. Changes to the loaded list will still need to be saved using the same button as above, so the same file name concerns still apply.
        1. I Might change the save button to use file dialogs at a later date.

1. Start New Comparison:
   1. This will write the values in the textboxes associated with (1-4) to a file called “files.cfg”
   2. This will overwrite the previous config file.
   3. This also launches the Comparison script, whose results are currently only getting written to (8)
2. Compare using Old Configuration:
   1. If you already have a files.cfg file, and do not want to change the old information, click here.
   2. This will launch the Comparison script using the old parameters.
   3. The textboxes in (1-4) can be left blank if using this button.
3. Python STD Out window:
   1. This is the window where all “print()” statements get written to.
   2. For development, I am writing the results to this window.
   3. The visualization scripts do not get run at the end of the comparison script right now.

**Final Notes:** The state of the program and next steps

I have been working on getting the “selected polygons” functionality to work but have spent a lot of time trying to automate the process. In the interest of time, this process is currently being switched to a manual method of entry that may eventually be added to the UI. The 3D visualization takes roughly 20 minutes to complete, but once that initial processing time is completed, two selected faces can be interacted with (Rotated) in real time. The Next Steps for the foreseeable future is getting the selected polygons to be the only ones displayed on the 3D face.