PIC 16, Winter 2018 – Preparation 5F

Assigned 2/7/2018. To be completed by class 2/9/2018.

Intended Learning Outcomes

By the end of this preparatory assignment, students should be able to:

- lay out GUIs, including custom widgets, in Qt Designer and save them to `.ui` files;
- translate the `.ui` files to Python code (`.py`) files using the command line tool `pyuic`; and
- use the class contained in the resulting `.py`file in the creation of a GUI.

Tasks

- ☐ Locate Qt Designer (*different from Qt Creator*), a piece of software that should have been installed with Anaconda, on your computer. On PIC computers, just enter `designer` at the Windows command prompt. If that doesn't work on a personal Windows / OS X machine, search in the start menu / perform a spotlight search for "Designer" or "Qt Designer". If that doesn't work, either, you may need to search your hard drive manually for a similarly named executable. If you really can't find it, follow the guidelines at the end of this document to install Qt.
- ☐ Check that you can run `pyuic`, `pyuic4`, or `pyuic5` on your computer. Try entering each of those at your operating system (not Python) command prompt / terminal until you find one that works. If you get `Error: one input ui-file must be specified`, that's a good thing! If you get`ImportError: No module named PyQt5` (especially on OS X), you need to locate the executable file (*not a* `.py` *file!*) in the Anaconda installation directory, open the file in a text editor, and replace "`pythonw2.7`" with "`pythonw`" (i.e. remove the "`2.7`"). If you get some sort of "command not found" error, you may need to locate one of these files on your system (search the Anaconda installation directory), add that folder to the system path, and then try again. If you're having trouble, ask for help *in person with your computer* or use a PIC computer!
- ☐ Follow Let's Learn Python #24 – UI with Python, PyQt & Qt Designer. Since you everything is already installed, you can skip ahead to minute 4:00.
- ☐ One limitation to this approach, as mentioned in the video, is that every time you modify your `.ui` and use `pyuic` to convert it to a `.py` file, you lose the code that makes the GUI *do* stuff. This next tutorial gets around this issue by importing and subclassing the UI class generated by `pyuic` in a separate `.py` file, then writing the functionality in the subclass. Also, remember the issue that the UI class inherited from `object` rather than `QWidget`? This is fixed by having your subclass inherit from `QWidget` (or `QMainWindow`) in addition to the UI class. Read PyQt: Getting started with PyQt and Qt Designer.
- ☐ Note that this tutorial uses a `QMainWindow` instead of a `QWidget`. There's not much of a difference; you just have to make sure that your subclass inherits from the appropriate class.
- ☐ If after that you're confident that you can lay out a GUI in Qt Designer and get it to do your bidding *without* referring to resources, great! I wasn't. So I created these videos to help you (and me!) get more comfortable.
    - Creating a GUI Programmatically – This was posted in the last reading, but you might want to watch it, especially if you didn't last time. I start by making the simplest Qt GUI I can (5 lines, including `import`) and work up to some fundamental features, like making a button do something and painting on a widget. Note that I pronounce "Qt" (should be pronounced "cute") incorrectly throughout. My bad. Also I misspoke when I said that the `main` function allowed us to run the file from the command line. Rather, wrapping

everything in a `main` function that only runs when `__name__ == "__main__"` allows us to *avoid* running code in the `main` function when we import the module into another.
    - Using Qt Designer for Layout – this provides a third perspective on the use of Qt Designer. I hope it's useful. The intent is for you to really understand what Qt Designer and `pyuic` have done for you, and why we use the output `.py` file the way we do.
    - These resources might have been created for a different version of pyqt/pyuic. The differences are superficial. Try to roll with is.
- ☐ Finally, watch Laying out Custom Widgets in Qt Designer; this isn't covered in the other references.
- ☐ If you're looking for some practice, try laying out some of the GUIs from Preparation 5W, Preparation 5F, Assignment 5W, and Assignment 5F using Qt Designer, generating the `.py` file for the layout using `pyuic`, then creating the program using the class generated by `pyuic`. For a little twist, try adding a single pause/play button to Assignment 5W (that pauses and plays the ball bouncing animation), or a slider that can adjust the size of the square in Assignment 5F.

Tips on installing Qt:
If you find that you need to install Qt, visit https://www.qt.io/download-open-source/.
Note that if you want to install Qt on a computer that's short on hard drive space, you can customize the Qt install, removing everything but Qt Creator, which includes the "Qt Designer" program we'll be using: