Guide to PyQt Signals

ECE 1140

Good: <https://stackoverflow.com/questions/36434706/pyqt-proper-use-of-emit-and-pyqtsignal>

Basic Notes:

* **Signal**: emitted when a particular event occurs
  + when a signal is emitted, the slots connected to it are usually executed immediately
  + when this happens, the signals and slots mechanism is totally independent of any GUI event loop
  + execution of the code following the “emit()” statement will occur once all slots have returned
* **Slot**: a **function** that gets called by a signal
  + CAREFUL: signals and slots are type safe, meaning the signature of a signal must match the signature of the receiving slot (i.e., the slot’s function parameters). However, mismatches are detected at runtime, so an error will only occur when you try to emit a faulty signal.

Functions:

* Signal Declaration
  + signal\_name = pyqtSignal(data\_type)
  + Say for example I want a signal for the train controller to open doors on the train model. The signal declaration will look something like this
    - signals.train\_doors\_signal = pyqtSignal(bool)
* Signal Connection
  + in order for the signal to know where to go, it needs to be connected to its destination (a slot)
  + In the example above, I would want the target function to be in the train model class. Say this function is called train\_doors(bool).
  + To connect, I’d include the following line in my Train Controller \_\_init\_\_ function:
    - signals.train\_doors\_signal.connect({name of slot function})
* Signal Emit
  + Now that I’ve done all that nonsense, I can actually emit the signal.
  + Whenever I’m ready to send data, I’ll write the following line:
    - Signals.train\_doors\_signal.emit(data to send to slot function)

Note on our implementation:

* We will be placing all of our signal declarations into the SignalsList class within signals.py.
* This will ensure that all signals are easily viewable for everyone in the same place. It’s important to note that this is not necessarily how pyqtSignals have to be implemented, but it should make things easier if they’re all viewable in one big list.
* To access signals within your module,
  + First, “from signals import signals”
    - In the signals.py file, there is one instance of the signalsList class declared which we will all share
  + To declare,
    - In the signals list, “{signal\_name} = pyqtSignal(list of data types for your slot parameters)
  + To connect,
    - In your class \_\_init\_\_ method, “signals.{signal\_name}.connect({slot function name}).
  + To emit,
    - signals.{signal\_name}.emit({variable(s) to emit})