# Instrument and GUI

Feb 25, 2024

We are using PyQT. A convenient way to install it consists in installing pyqtgraph:

```
pip install --user pyqtgraph
```

## 1 Threads

Threads allow you to execute several functions at the same time. They are fundamental with GUI (for example you want to interact with buttons even if a function is running in the background).

Bellow is a simple example of a thread

```
import threading
from time import time, sleep

class MyThread(threading.Thread):
    def __init__(self, parameter):
        threading.Thread.__init__(self)
        self.parameter = parameter

    def run(self):
        for i in range(10):
            print(self.parameter, i)
            sleep(.3)


thread = MyThread('Hello')
thread.start()
sleep(2)
print('Bonjour !!!')
thread.join()
```

1/ Modify this thread, so that parameter there is a callback function that will be called in the loop. Try with

```
def callback(i):
    print('Hello', i)
```

2/ We want to run it in an infinite loop, but we have to find a way to terminate the loop. One way consists in using an attribute (`want_to_terminate`), set to False and then use a `while not self.want_to_terminate:` loop. To terminate the loop we simply have to set the attribute to True. Implement this method. Replace the index `i` by the time elapsed since the start of the thread.

## 2 Stopwatch

The stopwatch will be implemented during the lecture

The minimum interface is the following

```
import time
from pyqtgraph.Qt import QtCore, QtGui

class StopWatchWindows(QtGui.QWidget):
    def __init__(self, args):
        self.app = QtGui.QApplication([])
        QtGui.QWidget.__init__(self)
        self.main_layout = main_layout = QtGui.QVBoxLayout()
        self.setLayout(main_layout)

if __name__ == "__main__":
    main = StopWatchWindows([])
    main.show()
    exit(main.app.exec())
```

Look at the following resource in order to get common widgets : www.pythonguis.com/tutorials/pyqt-basic-widgets/

## 3 Display Scope Output

1/ Starting from the stopwatch application. Replace the label with a MatplotlibWidget. In the callback method, get data from the scope and plot them. Here are some code to plot a graph inside a MatplotlibWidget

```
from pyqtgraph.widgets.MatplotlibWidget import MatplotlibWidget
plot_widget = MatplotlibWidget()

fig = plot_widget.getFigure()
fig.clf()
x = np.linspace(0, 1, 1001)
y = np.sin(2*np.pi*x)
ax = fig.subplots()
ax.plot(x, y)
fig.canvas.draw()
```

2/ Add a save button to the scope, that will open a `QtGui.QFileDialog.getSaveFileName` dialog box. Ideally, the default directory should be something like `data/yyyy/mm/dd` where yyyy, mm and dd are the year number, month and day ! Save the waveform as txt file (np.savetxt).