Interaction Technology and Techniques
Assignment 8: DSP and Filtering

Summer semester 2015

Submission due: Sunday, 21. June 2015, 23:55

Hand in in groups of max. two.

Your task is to implement two PyQtGraph nodes and to measure the noise level of the WiiMote.

8.1: An FFT node

Read the source code for `wiimote_node.py` and the PyQtGraph documentation[1]. Write a small Python application `frequalyzer.py` that takes a Bluetooth MAC address as its only parameter. This application should generate a PyQtGraph flowchart with the following elements:

- a WiiMoteNode.
- a BufferNode (see `wiimote_node.py`) for each of the accelerometer channels,
- an FFTNode for each of the accelerometer channels (to be implemented by yourself) that takes an array of values as input and outputs a frequency distribution on two channels (X = frequency in Hz, Y = amount)
- a PlotCurve node for each of the accelerometer channels which takes the X/Y values from the FFTNode and transforms them so that a linked PlotWidget node can draw the frequency distribution

Your application should import `wiimote_node.py` and use the two nodes defined there.

Hand in the following file:

frequalyzer.py: a Python script that implements this flowchart.

(Please do not hand in `wiimote.py` or `wiimote_node.py`.)

Points

- **1** The python script has been submitted, is not empty, and does not print out error messages.
- **2** The script correctly implements the features above.
- **2** The flowchart correctly displays the frequency distribution
- **1** The script is well-structured and follows the Python style guide (PEP 8).

8.2: A convolution filter node

Read the source code for `wiimote_node.py` and the PyQtGraph documentation[2]. Write a small Python application `noisalyzer.py` that takes a Bluetooth MAC address as its only parameter. This application should generate a PyQtGraph flowchart with the following elements:

[1] http://pyqtgraph.org/documentation/

[2] http://pyqtgraph.org/documentation/

- a WiiMoteNode.
- a BufferNode (see `wiimote_node.py`) for one of the accelerometer channels,
- a StdDevNode (to be implemented) that takes an array as input and outputs the standard deviation
- a NumberDisplay node (to be implemented) that takes a number and outputs it on a connected Qt Widget (see e.g., the PlotWidget implementation). This can either be a custom widget or an existing widget class such as QLCDNumber.
- optionally (no bonus points): a PlotWidget showing the accelerometer data.

The flowchart should output the standard deviation of the sensor data received from the Wiimote. You may limit the flowchart to a single accelerometer axis.

Your application should import `wiimote_node.py` and use the two nodes defined there.

Measure the noise levels for:

- the Wiimote lying on a stationary surface
- the Wiimote being held by a user as steadily as possible
- the Wiimote being swung around wildly

Hand in the following files:

- **noisalyzer.py**: a Python script that implements this flowchart.
- **noise_levels.txt**: a short documentation of the measured noise levels..

(Please do not hand in `wiimote.py` or `wiimote_node.py`.)

Points

- **1** The python script has been submitted, is not empty, and does not print out error messages.
- **2** The script correctly implements and displays the required flowchart.
- **2** The script correctly displays the current noise levels (standard deviation)
- **1** The script is well-structured and follows the Python style guide (PEP 8).
- **1** The reported noise levels are realistic.

8.3: Read up on Gesture Recognition

Read the paper on the *$P Point-Cloud Recognizer*[3].

Concisely answer the following questions:

- What does the recognizer do?
- Name an advantage of the *$P* recognizer over the *$1* recognizer.
- What is the *minimum matching distance*?

Hand in the following file:

gesture-recognizer.txt: a plain-text file containing your answers

Points

- **2** Good answer to first question
- **2** Good answer to second question
- **2** Good answer to third question

[3] http://faculty.washington.edu/wobbrock/pubs/icmi-12.pdf

Submission

Submit via GRIPS until the deadline

All files should use UTF-8 encoding and Unix line breaks. Python files should use spaces instead of tabs.

`Have Fun!`