Lehrstuhl für Medieninformatik
Institut für Information und Medien, Sprache und Kultur
Universität Regensburg

Interaction Technology and Techniques
Assignment 7: Wiimote, PyQtGraph

Summer semester 2017

Submission due: Thursday, 29. June 2017, 23:55

Hand in in groups of max. two.

Your task is to get comfortable with the WiiMote and PyQtGraph

7.1: A short introduction to Digital Signal Processing

Skim the chapters of The Scientist and Engineer's Guide to Digital Signal Processing[1] so that you have a good overview of the topics covered by this guide.

Concisely answer the following questions:

- What is the defining property of Gaussian noise?
- What does a low-pass filter do in general?
- Is a moving average filter a low-pass or a high-pass filter? Why?

Points

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

7.2: A WiiMote Game

Install the package `python3-bluez_0.22-1_amd64.deb` from GRIPS and follow the instructions from the slide set to get your Wiimote working. Read the source code for `wiimote.py` and have a look at `wiimote_demo.py` to understand the API. Write a small Python application `wiimote_game.py` that takes the Bluetooth MAC address of a Wiimote as its only parameter. This application should implement a fun game that involves your WiiMote:

- The application should `import wiimote` similar to the `wiimote_demo.py` example (i.e., do not modify `wiimote.py` itself).
- On launch, print instructions for the game to `stdout` or show them in a Qt window.
- Automatically connect to the Wiimote with the given MAC address.
- Utilize at least one input modality and one output modality of the Wiimote
- If you want, you may also implement a graphical user interface for the game - but you can also just use the Wiimote without any display.

[1] http://www.dspguide.com/

If you are looking for inspiration on game concepts, check out e.g., Bop It[2] or ball-in-a-maze puzzles[3].

Hint: Activating the rumble motor will mess with the accelerometer values. You might want to wait for a short time until you read and interpret them again.

Hand in the following file:

wiimote_game.py: a Python script that implements your game

(Please also hand in the `wiimote.py` version you are using)

Points

- **1** The python script has been submitted, is not empty, and does not print out error messages.
- **1** The script is well-structured and follows the Python style guide (PEP 8).
- **2** The game is fun to play (at least a little bit)
- **1** The game utilizes at least one input and one output modality of the Wiimote

7.3: A custom PyQtGraph flowchart using the WiiMoteNode

Read the source code for `wiimote_node.py` (from the wiimote.py GitHub repository[4]) and the PyQtGraph documentation[5]. Install the PyQtGraph Debian package from the website (run `dpkg -i <package.deb>` as root). Write a small Python application `analyze.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,
- three `PlotWidgets` that plot the accelerometer data for each channel and another `PlotWidget` that displays the output of the `NormalVectorNode` (see below)
- a NormalVectorNode (to be implemented by you) that calculates the rotation around one axis from the accelerometer values of the other two axes and outputs a vector (i.e., two 2D points) that can be plotted by a `PlotWidget` to indicate the rotation (see video in GRIPS) - this node should accept accelerometer values on its two input terminals and provide a list/tuple of two tuples, such as ((0, 0),(1.0,1.0)) on its output terminal.
- a LogNode that reads values (e.g., accelerometer data) from its input terminal and writes them to `stdout`.

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

Hand in the following file:

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

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 a flowchart.
- **2** The script correctly reads accelerometer data from the Wiimote and plots it.
- **1** The script is well-structured and follows the Python style guide (PEP 8).
- **2** The script contains a working NormalVectorNode as described above.
- **1** The script contains a working LogNode as described above.

[2] https://en.wikipedia.org/wiki/Bop_It
[3] https://en.wikipedia.org/wiki/Labyrinth_(marble_game)
[4] https://github.com/RaphaelWimmer/wiimote.py
[5] http://pyqtgraph.org/documentation/

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. If you need to submit further supporting files, please add a comment describing their use.

Have Fun!